



Atatürk Üniversitesi
Açıköğretim Fakültesi

Animasyon Tasarımı



Bu kitabın, basım, yayım ve saĖş hakları Atatürk Üniversitesi'ne aī Ė r. Bireysel öğrenme yaklaşımıyla hazırlanan bu kitabın bütün hakları saklıdır. Atatürk Üniversitesi'nin izni alınmaksızın kitabın tamamı veya bir kısmı mekanik, elektronik, fotokopi, manyeak kayıt veya başka şekillerde çoĖalĖlamaz, basılamaz ve dağıĖlamaz.

Copyright ©2014

The copyrights, publicaēons and sales rights of this book belong to Atatürk University. All rights reserved of this book prepared with an individual learning approach. No part of this book may be reproduced, printed, or distributed in any form or by any means, technical, electronic, photocopying, magneac recording, or otherwise, without the permission of Atatürk University.



ATATÜRK ÜNİVERSİTESİ
AÇIKÖĖRETİM FAKÜLTESİ

Animasyon Tasarımı

ISBN: 978-975-442-571-0

ATATÜRK ÜNİVERSİTESİ AÇIKÖĖRETİM FAKÜLTESİ YAYINI

ERZURUM

İÇİNDEKİLER

1. Animasyon Temelleri	4
Öğr. Gör. MUSTAFA AKBUĞA	
2. Grafikler	25
Öğr. Gör. MUSTAFA AKBUĞA	
3. Metin İşlemleri	49
Öğr. Gör. MUSTAFA AKBUĞA	
4. Semboller	70
Öğr. Gör. MUSTAFA AKBUĞA	
5. Zaman Çizelgesi	86
Öğr. Gör. MUSTAFA AKBUĞA	
6. Ses ve Video İşlemleri	102
Öğr. Gör. MUSTAFA AKBUĞA	
7. Animasyon	126
Öğr. Gör. MUSTAFA AKBUĞA	
8. Optimizasyon	146
Öğr. Gör. SUAVİ SERTCAN	
9. Yayınlama Ayarları	157
Öğr. Gör. SUAVİ SERTCAN	
10. Actionscript	177
Öğr. Gör. MUSTAFA AKBUĞA	
11. Temel Programlama Yapıları I	202
Öğr. Gör. MUSTAFA AKBUĞA	
12. Temel Programlama Yapıları II	224
Öğr. Gör. MUSTAFA AKBUĞA	
13. Dinamik Veriler	251
Öğr. Gör. ARİF DAŞ	
14. Eylem Kodları	277
Öğr. Gör. ARİF DAŞ	

Editör

Doç. Dr. İBRAHİM YÜCEL ÖZBEK

Öğr. Gör. MUSTAFA AKBUĞA

ANİMASYON TEMELLERİ



İÇİNDEKİLER

- Giriş
- Animasyon Yazılımı Yükleme
- Açılış ekranı ve Menüler
- Animasyon Belgesi oluşturmak
- Paneller ve Özellikleri
- Yardım Menüsünü Kullanmak



HEDEFLER

- Bu üniteyi çalıştıktan sonra;
- Animasyon hazırlamak için kullanılacak yazılımı seçebilecek,
- Seçilen animasyon yazılımında ki temel menü seçeneklerini açıklayabilecek,
- Animasyonun temel kavramlarını tanımlayabileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

ANİMASYON TASARIMI

Okt.
Mustafa AKBUĞA

ÜNİTE 1

GİRİŞ

Web sayfaları tasarlanırken görsellerin yanında, ses ve müzik içeren animasyonlar da kullanılır. Animasyon hazırlamak için çeşitli programlar (Adobe Flash, FotoMorph, Pivot Animation, Blander vb.) mevcuttur. Bu programlardan Adobe Flash programı animasyon tasarım programları içerisinde grafik, ses ve video bileşenlerini en iyi kontrol eden ActionScript programlama dilini desteklemektedir. Bu sebeple animasyon hazırlanmada en çok Adobe Flash programı, Web sayfası içerisinde gösterilmesinde ise Adobe Flash Player bileşeni tercih edilir.

ANİMASYON YAZILIMINI KURMAK

Animasyon tasarımı yapmak ve bu tasarımın kullanıcı ile iletişimini sağlamak amacıyla Adobe Flash Programı ve onun içerisinde kullanılan ActionScript kodlama dili kullanılmaktadır. Bu tasarım altyapısına sahip olmak için programın bilgisayarınıza yüklenmesi gerekmektedir. Bilgisayar sistemlerine güncel programların kurulması her zaman mümkün olmayabilir. Kurulumun başarıyla gerçekleşmesi programın teknik gereksinimlerinin karşılanmasına bağlıdır.

Adobe Flash Programın kurulabilmesi ve verimli çalışabilmesi için sistem gereksinimleri aşağıda verilmiştir.

Windows için;

- Intel® Pentium® 4 veya Intel Core™ Duo (veya uyumlu) işlemci
- Microsoft® Windows® 7 (64 bit), Windows 8.1 (64 bit)
- 2 GB RAM (4 GB önerilir)
- Kurulum için 2,5 GB kullanılabilir sabit disk alanı; kurulum sırasında ek boş alan gerekir (çıkarılabilir flash depolama cihazlarına yüklenemez)
- 1024x768 ekran (1280x800 önerilir)
- Gerekli yazılım etkinleştirilmesi, abonelik doğrulaması ve çevrimiçi hizmetlere erişim için İnternet bağlantısı ve kayıt gerekmektedir.

Mac OS için ;

- Multicore Intel® işlemci
- Mac OS X v10.7 (64 bit) veya 10,9 (64 bit) olması yeterlidir. Diğer teknik özelliklerin Windows ile aynı olması yeterlidir.

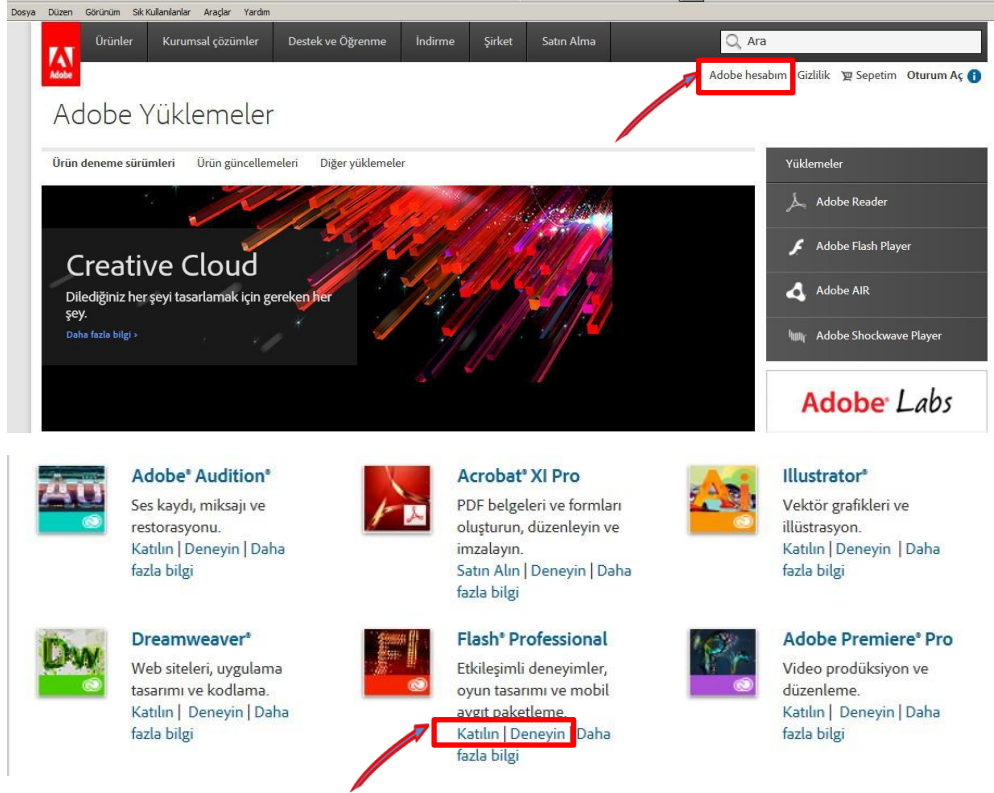
Grafik Tasarım dersi için Adobe Fireworks programını bilgisayarınıza kurarken, Flash programını da beraberinde kurduysanız bu programla animasyon tasarımı yapmaya hemen başlayabilirsiniz. Eğer bilgisayarınızda Flash programı kurulu değilse aşağıdaki adımları izleyerek programı bilgisayarınıza kurabilirsiniz



Adobe Flash
Professional CSS
sürümünün kurulumu
için mutlaka 64 bit
teknolojisine sahip
bilgisayar sistemi
gerekir.

1-Adım:

Aşağıda adresi verilen Flash programının internet sayfasına girin Bu adrese girdiğinizde Şekil 1.1. görülen sayfa ekrana gelecektir. Adobe Flash programını bilgisayarınıza yükleyebilmek için öncelikle Adobe ID numarası almanız gerekmektedir. Şayet Adobe ID numaranız var ise 2- Adımı atlayarak 3-Adım yardımıyla Flash programını bilgisayarınıza kurabilirsiniz. Şayet Adobe ID numaranız yok ise 2-Adım da verilen işlem basamaklarını uygulayın.



Şekil 1.1 Adobe Türkiye Sitesi Tarafından Sağlanan Ücretsiz Deneme Sürümlerini İndirme Sayfası

2. Adım:

Şekil 1.1' de sayfanın üst tarafında kırmızı ok ile gösterilen "**Adobe Hesabım**" bölümüne tıklayın. Şekil 1.2 de görülen sayfa karşınıza gelecektir. Bu şekilde kırmızı dikdörtgen çerçeve içerisindeki "**Adobe kimliğiniz yok mu?**" bölümüne tıklayın. Bu işlem sonucu Şekil-1.3 sayfası karşınıza gelecektir. Burada belirtilen yerleri doldurunuz. E-posta adresi olarak aktif olarak kullandığınız bir eposta adresinizi ve şifre olarak unutmayacağınız bir şifre belirleyerek yazınız. Bu bölümlere yazdığınız e-posta adresiniz ve şifreniz sizin Adobe ID nizi oluşturacak ve kurulum esnasında gerekli oldukça bu bilgileri girmeniz istenecektir. "**Adobe Kullanım Koşulları ile Gizlilik İlkesini okudum ve kabul ediyorum**" bölgesini

Animasyon Temelleri

işaretledikte sonra en altta bulunan **Oluştur** bölgesine tıklayınız. Bu işlemleri yaptıktan sonra sisteme girdiğiniz e-postanızı kontrol ediniz. Adobe tarafında size bir e-posta gönderilmiş olacak gelen e-postanın bir örneği Şekil 1-4 de görülmektedir. Gelen e-postada “**doğrulama bağlantısını**” bölmesine tıklayınız. Bu işlem sonunda Adobe ID hesap oluşturma işlemi başarıyla tanımlanmış olacaktır. Simdi Aşağıdaki adrese tekrar girerek diğer adımlarda verilen işlem basamaklarıyla flash programını indiriniz.

Adobe hesabım

Adobe kimliğinizi kullanarak ücretsiz deneme sürümleri indirebilir, ürün satın alabilir, siparişlerinizi yönetebilir, Adobe® Creative Cloud™ ve Acrobat.com gibi çevrimiçi hizmetlere erişim sağlayabilirsiniz. Ayrıca, giderek büyüyen çevrimiçi Adobe topluluğuna katılabilirsiniz.

Oturum Aç

Adobe Kimliği

Şifre

☐ Beni hatırla

Oturum aç

Adobe kimliğiniz yok mu?

Oturum açarken sorunlarla mı karşılaşıyorsunuz?

Şekil 1.2 Adobe Türkiye Sitesi Tarafından Sağlanan Ücretsiz Deneme Sürümünü İndirme Sayfası

Adobe hesabım

Adobe kimliğinizi kullanarak ücretsiz deneme sürümleri indirebilir, ürün satın alabilir, siparişlerinizi yönetebilir, Adobe® Creative Cloud™ ve Acrobat.com gibi çevrimiçi hizmetlere erişim sağlayabilirsiniz. Ayrıca, giderek büyüyen çevrimiçi Adobe topluluğuna katılabilirsiniz.

Bir Adobe kimliği oluşturun

< Zaten bir Adobe kimliğim var

Adobe ID (E-posta Adresi)

Ad

Şifre

Soyadı

Şifreyi Yeniden Yazın

Ülke/Bölge

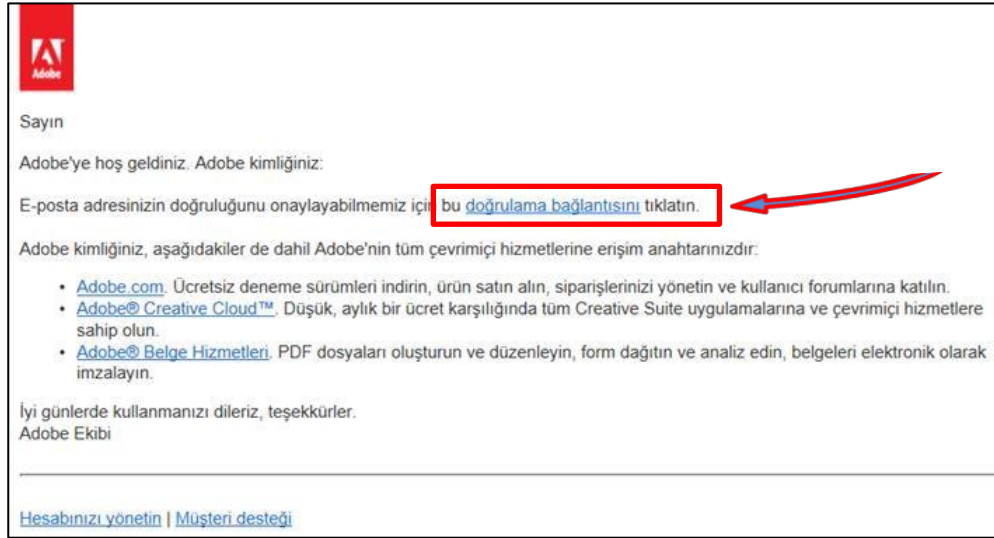
Türkiye

☐ E-posta yoluyla Adobe ürünleri ve hizmetleri hakkında bilgi almak istiyorum. Daha fazla bilgi.

☐ Adobe Kullanım Koşulları ile Gizlilik İlkesini okudum ve kabul ediyorum.

Oluştur

Şekil 1.3 Adobe Türkiye Sitesi Tarafından Sağlanan Ücretsiz Deneme Sürümünü İndirme Sayfası



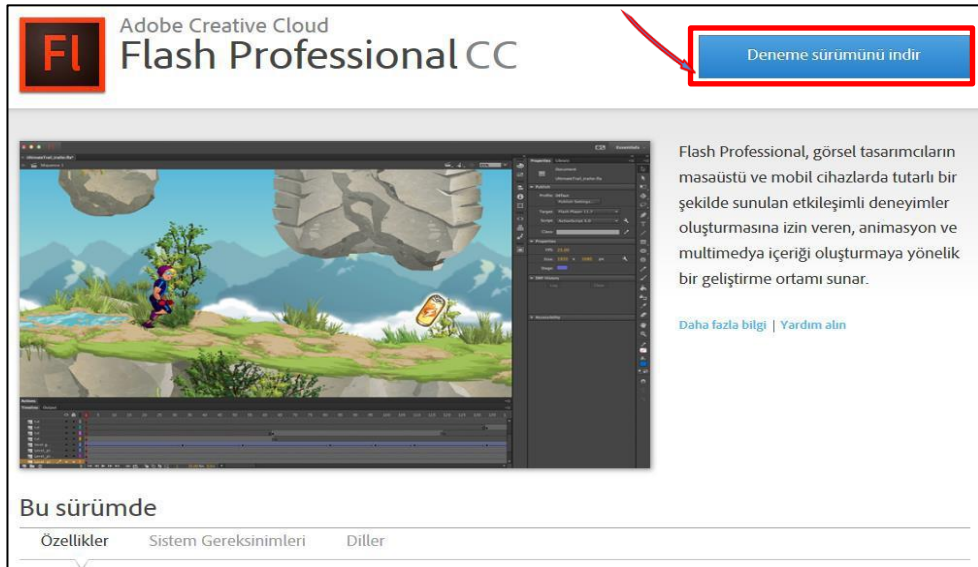
Şekil 1.4 Adobe Türkiye Sitesi Üzerinden ID Doğrulama İşleminin Başlatılması

3-Adım:

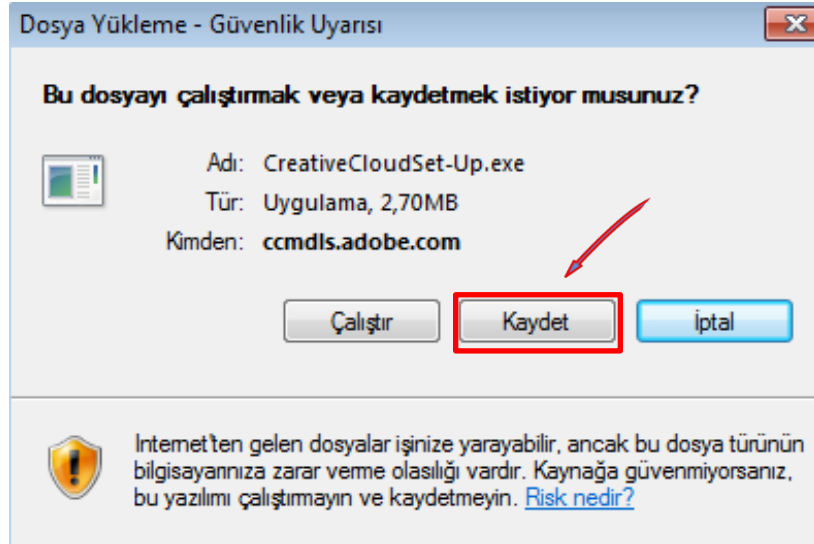


Bir Adobe ID hesabı için geçerli bir e-posta hesabı gereklidir. Bu e-posta hesabı şifre unutulduğunda gerekli olabilir.

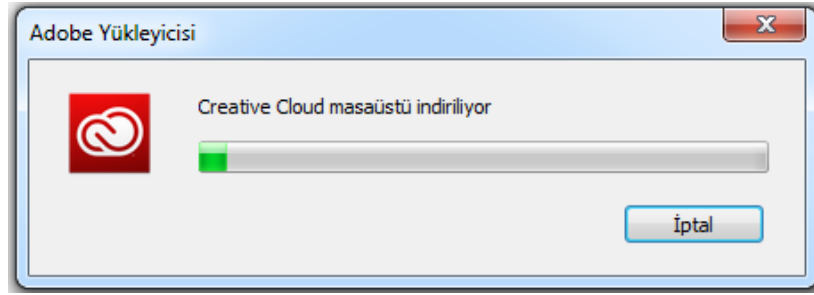
Şekil 1.1' de verilen sayfanın orta bölümünde bulunan Flash Pro. amblemine tıklayınız. Bu tıklama sonucunda Şekil 1.5' te görülen sayfa açılacaktır. Ekranın sağ üst tarafında bulunan ve Şekil 1.5.'te kırmızı kutucuk ile işaretlendirilmiş “*Deneme sürümünü indir*” bölümüne tıklayınız. Bu tıklama sonucunda Şekil 1.6' da görülen ekran karşınıza gelecektir (Bu işlem öncesinde e-postanız ve şifreniz sorulur ise bu bilgileri giriniz.) Burada *Kaydet* bölümüne tıklayarak programın *Adobe Creative CloudSet-Up. exe* dosyasını bilgisayarınızın masa üstüne indirin. İndirilen Adobe Creative CloudSet-Up. exe dosyasını çalıştırın çalıştırma sonrası Şekil 1.7' de görülen yükleme ekranı ekrana gelecektir.



Şekil 1.5 Flash Professional CC Versiyonun Deneme Sürümünün İndirilmesi

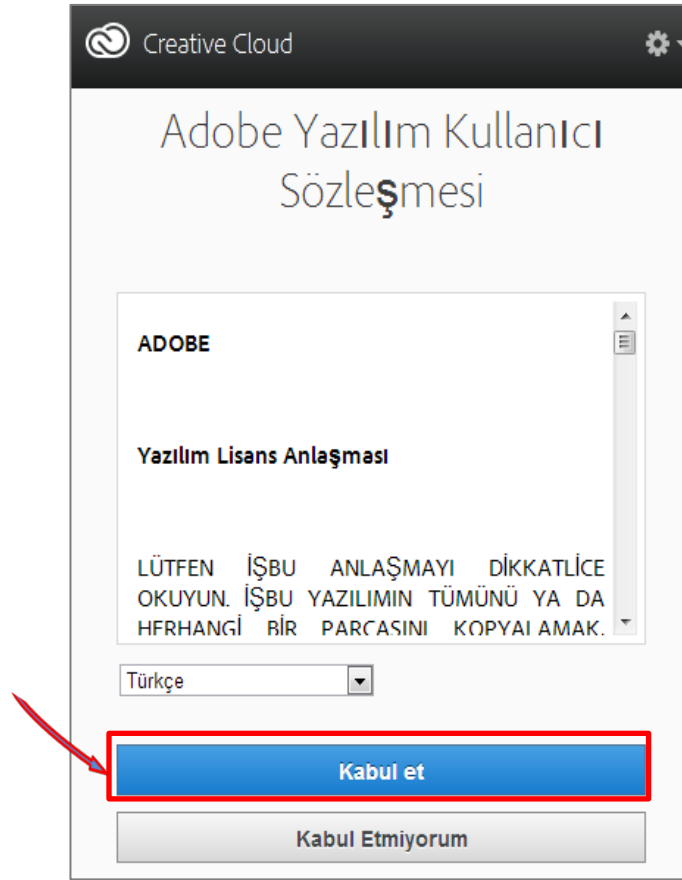


Şekil 1.6 CreativeCloudSet-Up Dosyasının Bilgisayarda Çalıştırılması

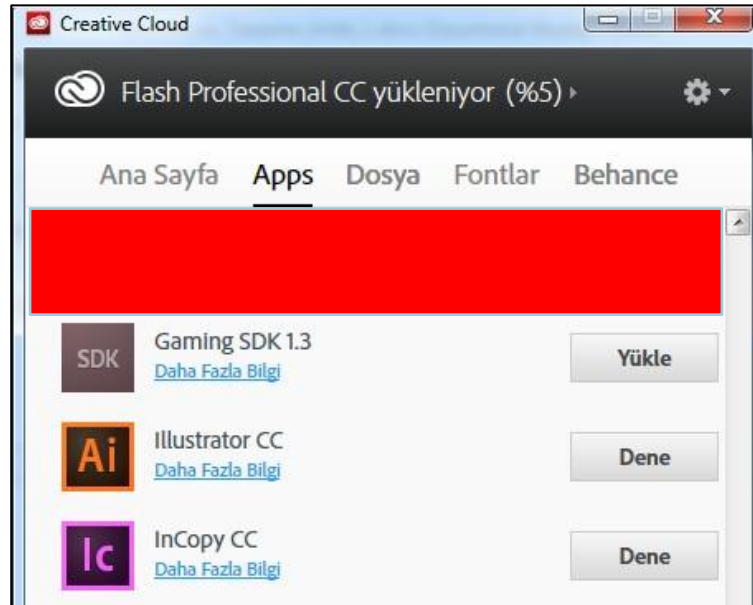


Şekil 1.7 Adobe Yükleyici ile Creative Cloud Yazılımının Kurulumu

Bu işlemler sonrasında Adobe ID'nizi tekrar girmeniz gerekecektir. Adobe ID'nizi doğruladıktan sonra Şekil 1,8'de görülen sözleşme karşınıza gelecektir. Bu sözleşmeyi kabul ettiğinizi belirtiniz. Sözleşme kabulünden sonra Program bilgisayarınıza yüklenmeye başlanacak ve Şekil 1,9'da görülen görüntü ekrana gelecektir.

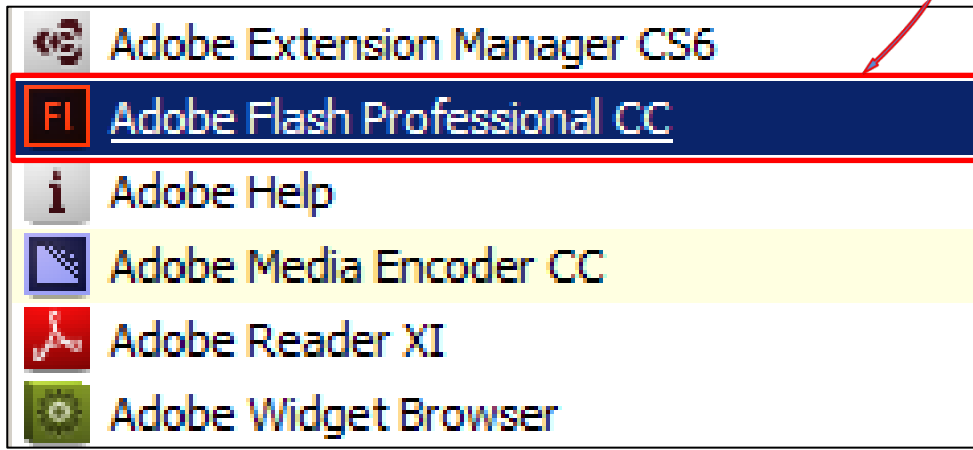


Şekil 1.8 Kullanıcı Yazılım Lisans Sözleşmesinin Yapılması

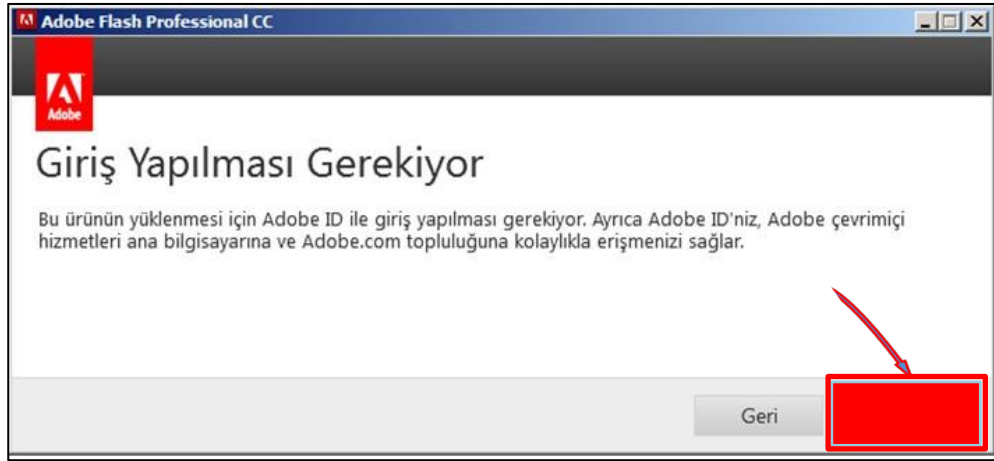


Şekil 1.9 Creative Cloud ile Flash Professional CC'nin Bilgisayara İndirme İşleminin Başlatılması

Bu işlemlerden sonunda Flash programı bilgisayarınıza yüklenmiş olacaktır. Başlat menüsünden Tüm programlar bölümüne giriniz. Bu durumda Şekil 1.10 ekrana gelecektir, buradan Adobe Flash Professional CC'yi seçerek *Giriş yap* bölümüne tıklayın. Bu durumda Şekil 1.11 ekrana gelecektir.

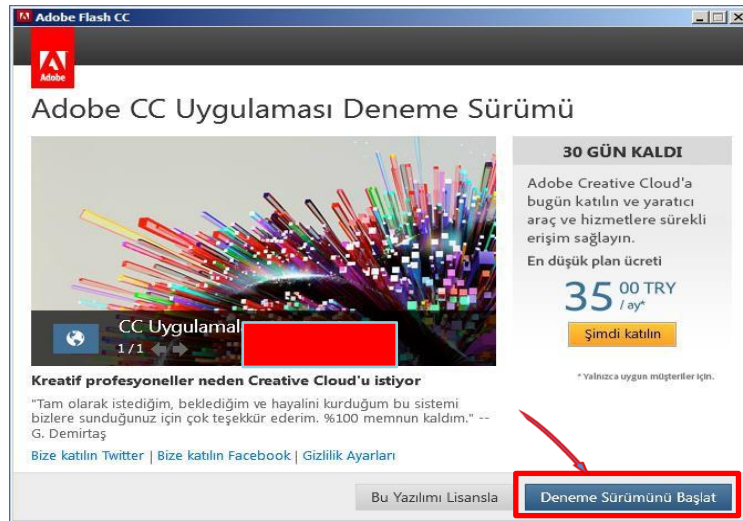


Şekil 1.10 Programın Başlat Menüsündeki Görünümü



Şekil 1.11 Kurulumun başlatılabilmesi için Adobe ID ile giriş gerekir.

Şekil 1.12'deki Deneme sürümünü başlatma tıkladıktan sonra Adobe Flash programı artık kullanılmaya hazırdır.



Şekil 1.12 Adobe Flash Programının Lisanslama İşlemi veya Demo Olarak Kullanılması



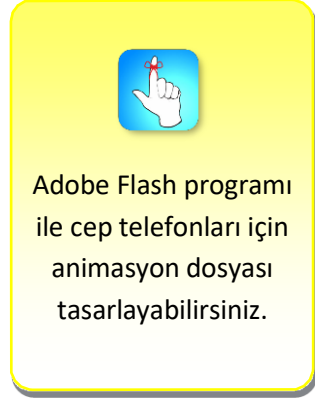
Şekil 1.13 Adobe Flash Professional CC bütün eklentileriyle demo olarak çalışmaya başlaması.

AÇILIŞ EKRANI VE MENÜLER

Bilgisayara Adobe Flash Professional CC'in demo ya da lisanslı sürümünü kurduktan sonra *Başlat > Programlar > Adobe Flash Professional CC* komutu verildiğinde (Şekil1.14) deki gibi bir açılış ekranı karşımıza gelecektir.

Yeni Bir Belge Oluşturma

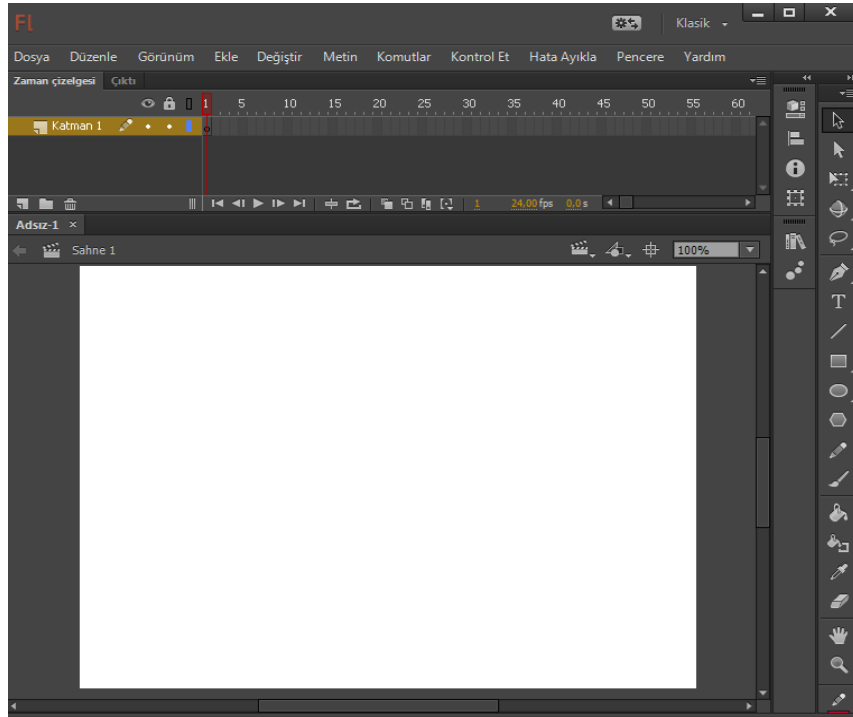
Şekil 1.14 ekranında *Yeni Oluştur > ActionScript 3.0* veya diğer seçeneklerden biri seçilerek çalışmaya başlayabilirsiniz. Yeni bir Flash dosyası oluşturmak veya mevcut bir Flash dosyası açmak içinse Son Kullanılan Öğeyi Aç bölmesinden Aç komutu verilir. Bu sayfada ayrıca son kullandığınız dosyalara ait bir listede görüntülenir. Bu dosyaları açmak için dosya ismine tıklamanız yeterli olacaktır.



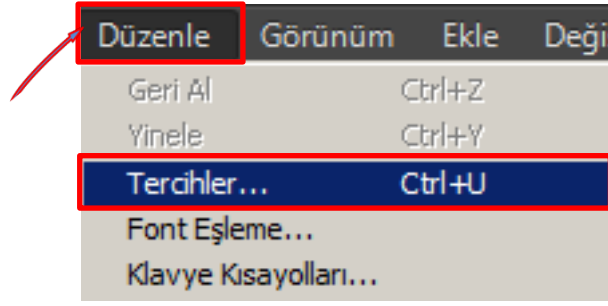
Şekil 1.14 Açılış Ekranı

Yeni bir çalışma dosyası oluşturduktan sonra koyu bir tasarıma sahip program ekranıyla karşılaşacaksınız (Şekil 1.15). Programın en sağında araç paneli sola gittikçe diğer paneller ve beyaz çalışma sahnesiyle vardır.

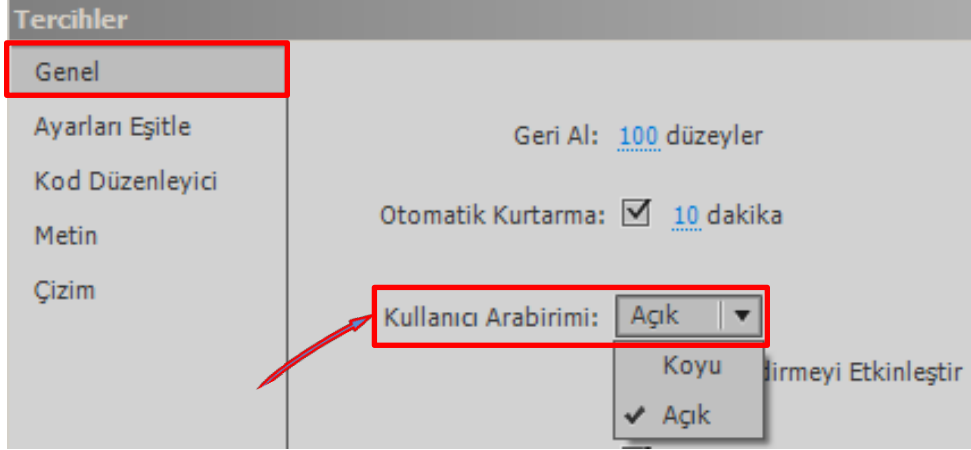
Bu ekranın önceki Adobe Flash versiyonlardaki gibi açık bir temaya sahip olması için Şekil 1.16-17 deki gösterildiği gibi **“Düzenle>Tercihler”** bölümden **Genel>Kullanıcı Arabirimi >“Açık”** Komutunu verebilirsiniz.



Şekil 1.15 Programın İlk Açılış Ekranı

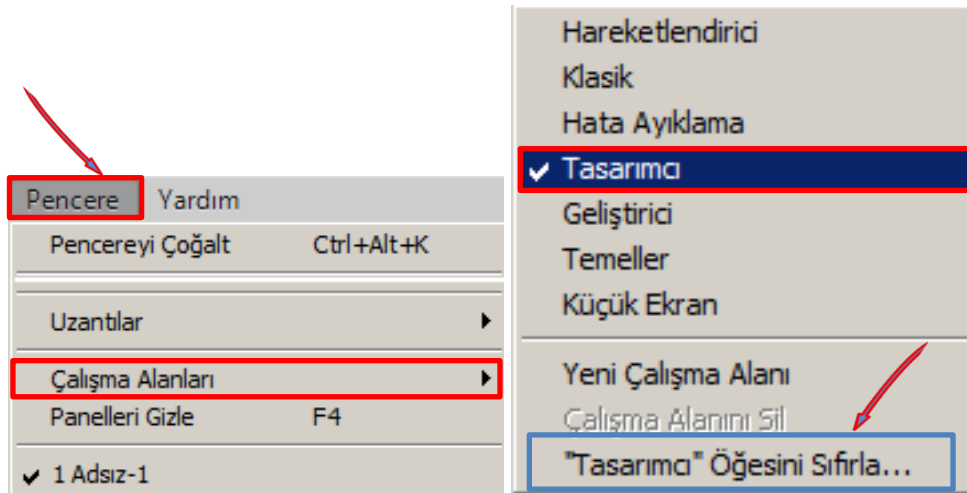


Şekil 1.16 Programın Açık Tema Seçeneğinin Aktifleştirilmesi

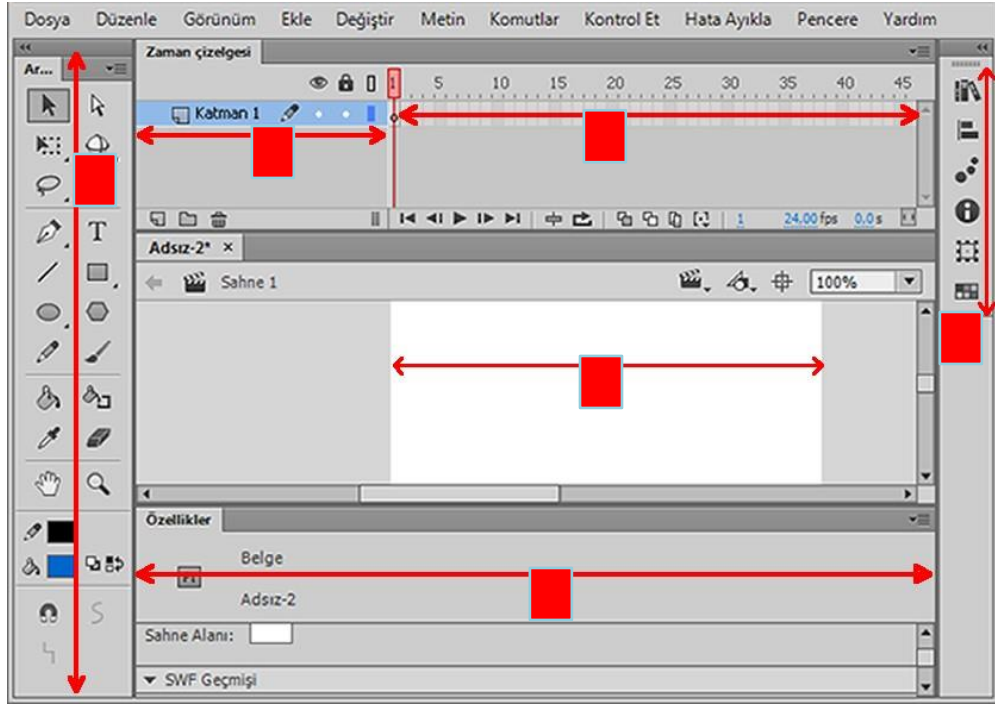


Şekil 1.17 Programın Açık Tema Seçeneğinin Aktif Edilmesi

Adobe Flash'ın önceki alışılmış ekran görüntüsüne ulaşmak için Paneller üzerindeki taşıma noktalarından tutularak çalışma alanı üzerinde istediğiniz bölgeye yerleştirebilirsiniz veya Şekil 1.18'deki gösterildiği gibi "Pencere>Çalışma Alanları" bölümden Tasarımcı ara birimini seçerek Şekil 1.19'da gösterilen tasarım alanına ulaşabilirsiniz.



Şekil 1.18 Programın Açık Tema Seçeneğinin Aktif Edilmesi



Şekil 1.19 Boş Çalışma Sayfasıyla Ekrana Gelen Flash CSS Editörünün Görüntüsü

Kullanıcı panelleri hatalı bir şekilde taşır yada çalışma ortamından kaldırılırsa Şekil 1.18'deki gösterilen Mavi çerçeve içerisindeki bölümden **Pencere > Çalışma Alanları > "Tasarımcı" Öğesini Sıfırla..** Komutlarını vererek programın başlangıç ayarlarına tekrar döndürebilir.

Çalışma alanı öğelerini; şekil 1.19'de yapılmış olan numaralandırmaya göre açıklayabiliriz.

1. Zaman doğrusu: Kare kare canlandırma hazırlanırken zaman doğrusu üzerindeki çerçeveler kullanılır.

2. Katmanlar: Flash film dosyalarında kullanılan katmanları gösterir. Katmanlarla ilgili işlemler buradan yapılır.

3. Araç Kutusu: Flash'ta grafik çizimleri için gerekli olan araçların bulunduğu araç kutusudur.

4.Özellikler paneli: Flash çalışma alanında ve panellerinde yer alan tüm nesne ve araçların sahip oldukları özelliklerin gösterildiği özellikler panelidir.

5. Panel gurupları: Grafikler üzerinde çeşitli renk işlemlerinin yapıldığı Renk Karıştırıcısı ve Renk Pastelleri panelleridir.

6.Çalışma Alanı: Çalışma sayfası olarak da adlandırılır. Zaman çizelgesindeki her bir Anakare bir çalışma sayfası olarak kullanılır. Grafik öğelerinin ve nesnelerin yerleştirildiği alandır.

ANİMASYON BELGESİ OLUŞTURMAK

Adobe Flash boş çalışma dosyası dışında çeşitli tiplerde reklam bantları,



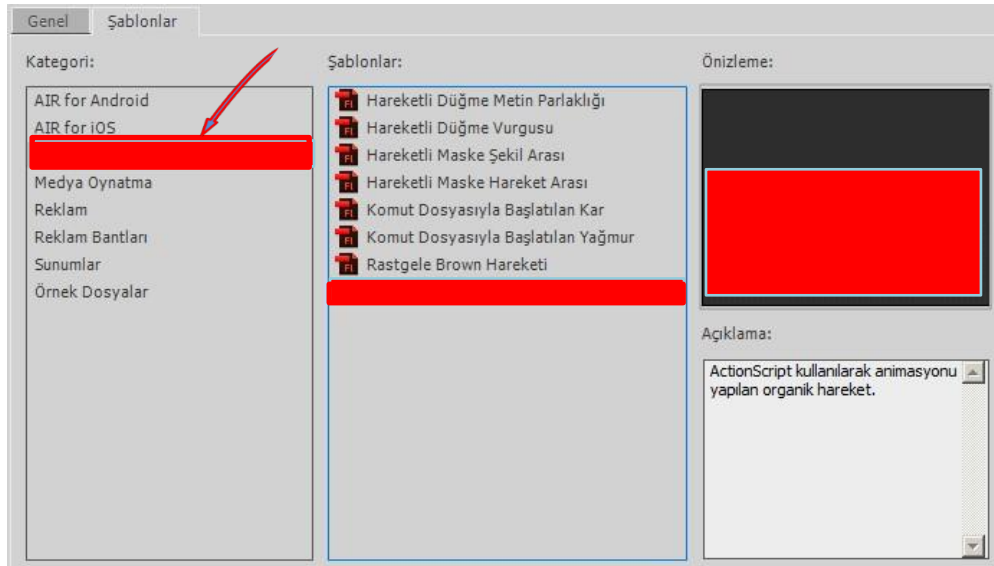
Program açıldığında görülmeyen paneller varsa Pencere menüsü kullanılarak görünür hale getirilir.

sahiptir. Bu şablonlar kullanılarak web sayfasına uygun flash dosyası hazırlanabilir.

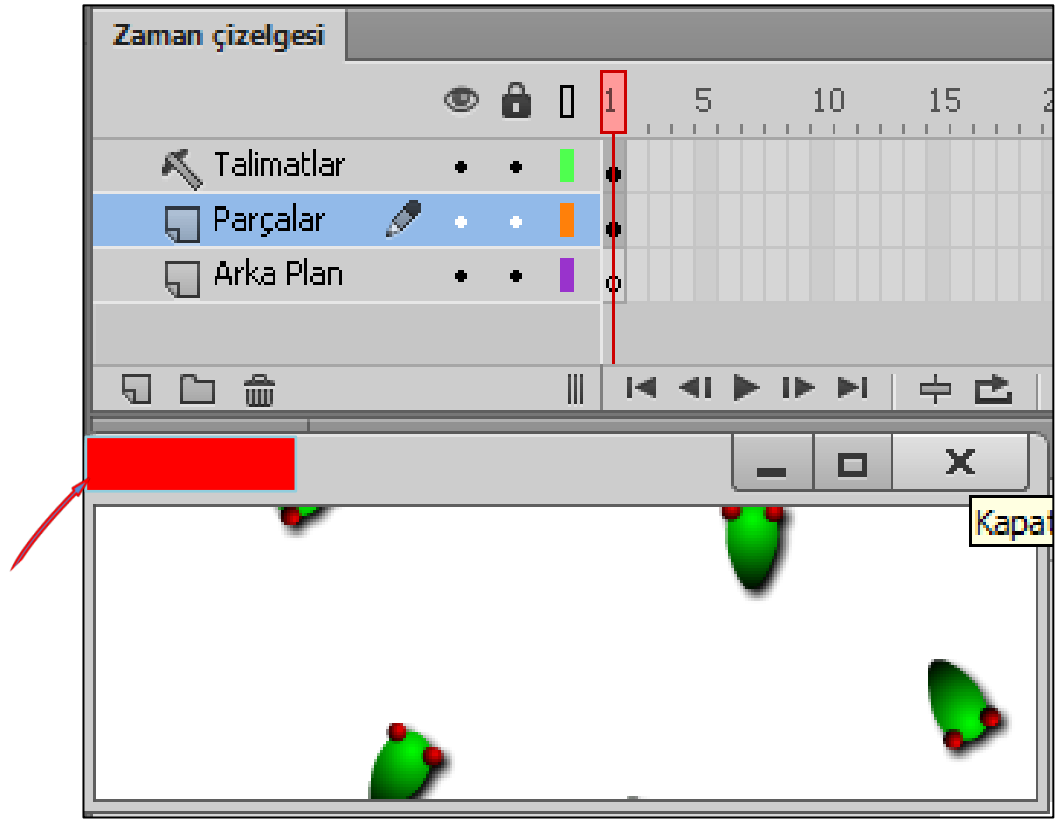


Örnek

- 500x500 boyutlarında Animasyon şablonunun kullanılır hâle
- getirilmek için;
- *Dosya > Yeni > Şablonlar > Animasyon > Rasgele organik hareket'i*
- komutunu vererek "tamam"ı işaretleyiniz (Şekil 1.20)
- *Kontrol Et > Test Et'i* (Şekil 1.21) seçerek oluşturulan animasyonu izleyebilirsiniz.
- *Pencere > Eylemler* (Şekil 1.22) seçerek çalışma alanında oluşturulan animasyonun hareket etmesini sağlayan ActionScript kodlarını görebilirsiniz.



Şekil 1.20 Şablonlar Sekmesinden Animasyon Dosyasının Seçilmesi



Şekil 1.21 Seçilen Animasyonun “Test Et” Komutuyla Önizlenmesi



Şekil 1.22 Animasyona Hareket Kazandıran Actionscript Kodlarının Görüntülenmesi

PANELLER VE ÖZELLİKLERİ

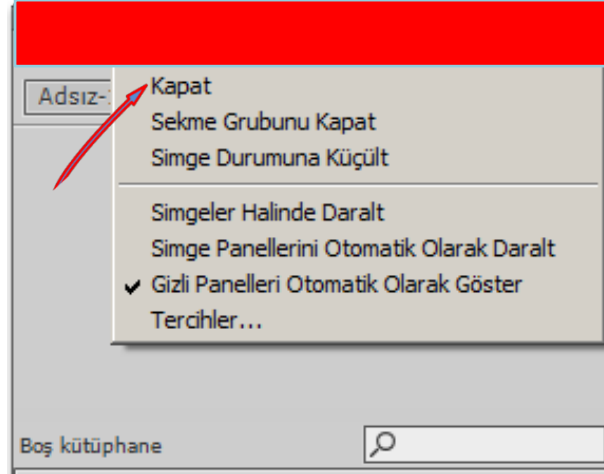
Flash MX den sonraki Flash sürümleriyle kullanıcı dostu, kullanımı kolay, istenildiği gibi hareket ettirilebilen her biri kendine özgü özellikleri barındıran *paneller* kullanılmaya başlanmıştır.



Şekil 1.23 Özellikler Panelinin Ekran Üzerinde Taşınması

Panellerin üzerinde hareket işlemlerinin yapılması ve bazı özelliklerinin değiştirilmesi;

1. Şekil 1.23 'te gösterilen **Özellikler** panelini bulunuz ve farenin sol tuşuyla sürükleme yapılacak alana (kırmızı daire içinde gösterilen alan) tıklayınız.
2. Tuşu basılı tutarak fareyi çalışma alanına doğru sürükleyiniz.
3. Araç kutusunun üstünde gözüken aşağıdaki alanda çarpı işaretine tıklayarak araç kutusunu kapatınız.
4. **Pencere > Özellikler** Komutuyla araç kutusunu yeniden görünür hâle getiriniz.



Şekil 1.24.Kütüphane Panelinin Ekran Üzerindeki Konumunu Ayarlanması

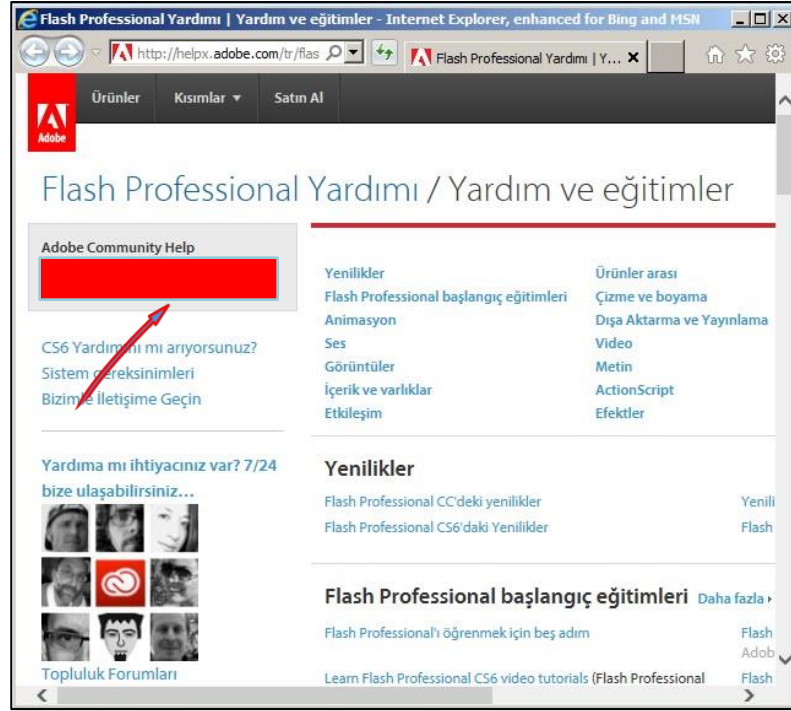
- 5- Şekil 1.24'de gösterilen **Kütüphane Panelini** bulunuz ve işaret edilen şekle farenin sağ tuşu ile tıklayınız. Bu durumda yapılabilecek işlemler menüsü görülecektir. Menüdeki seçenekleri seçerek panelde ne tür değişiklikler oluştuğunu gözlemleyiniz.
- 7-

YARDIM MENÜSÜNÜ KULLANMAK

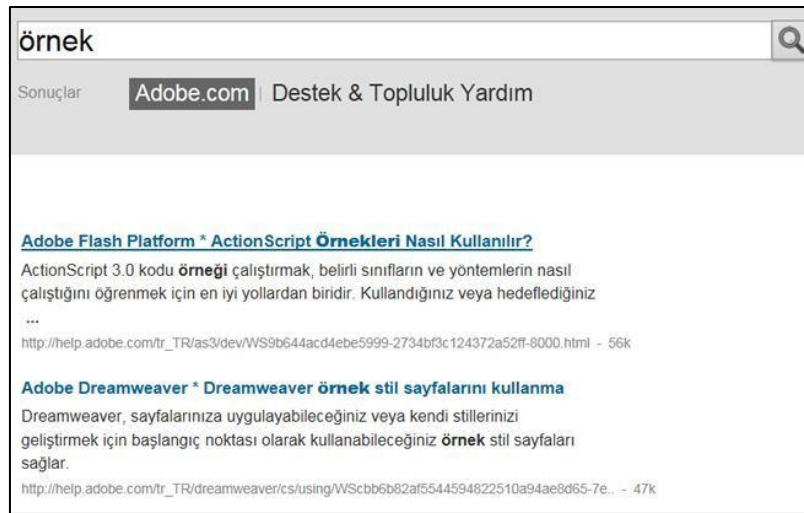
- 8- Flash'ta **F1** tuşuna basarak ya da menü çubuğunda **Yardım > Flash Yardım** komutuyla yardım menüsüne ulaşabilirsiniz (Şekil 1.25). Yardım menüsü geniş bir içeriğe sahiptir. (yabancı dil bilmeniz durumunda) Flash'la ilgili öğrenmek istediğiniz pek çok konuya rahatlıkla ulaşabilirsiniz.



Şekil 1.25 Yardım Panelinin Açılması



Şekil 1.26 Flash Yardım Merkezinin Görüntüsü



Şekil 1.26 Flash Yardım Merkezinde Bir Konunun Aranması

Flash'ın yardım menüsünde, hakkında yardım alabileceğiniz konu sınıfları vardır (Şekil 1.27). Özellikle örnek konusunu seçecek olursanız, Flash'ta çeşitli çalışmaları nasıl yapabileceğiniz hakkında geniş bilgiler ve örnekler bulabilirsiniz.



Özet

- Web tasarımında kullanılan sesli animasyonlar Adobe flash CSS programı veya önceki sürümleri kullanılarak oluşturulabilir ve düzenlenebilir.
- Adobe Flash CSS programının kurulması için gereken (setup.exe) dosyası Adobe program yükleme merkezinden indirilerek kaydedilmeli ve yönetici konumunda çalıştırılmalıdır.
- Kurulum programı çalıştırıldıktan sonra kurulum dosyalarının hazırlanması ve kurulum sihirbazının ekrana gelmesi için ön yükleme işlemi yapılır. İşlem başarıyla tamamlandıktan sonra kurulum sihirbazı ekrana gelir. İleri butonuna tıkladığınızda Lisans Anlaşması ekrana gelir. Anlaşmayı KABUL ET düğmesine tıklayarak onayladığınızda lisans seri numarası ya da demo kurulum seçeneği seçilir ve kurulum tamamlanır.
- Adobe Flash CSS programının çalışma ekranında, zaman doğrusu, katmanlar, paneller, araç kutusu, özellikler penceresi ve diğer paneller yer alır.
- Araç kutusu: Flash'ta grafik çizimleri için gerekli olan araçların bulunduğu bölümdür.
- Özellikler penceresi: Flash çalışma alanında ve panellerinde yer alan tüm nesne ve araçların sahip oldukları özelliklerin gösterildiği paneldir.
- Çalışma alanı, grafik öğelerinin ve nesnelerinin yerleştirildiği alandır. Flash'ta F1 tuşuna basarak ya da menü çubuğunda Yardım > Flash Yardım komutuyla yardım menüsüne ulaşabilirsiniz.
- Program ara yüzünde bulunan araç kutusu, panel gurupları, çalışma alanı, özellikler penceresinin bilinmesi yapılan çalışma verimi için son derece önemlidir.

DEĞERLENDİRME SORULARI

1. Web tasarımında kullanılan animasyon kaynakları aşağıda verilen programlardan hangisiyle düzenlenebilir?
 - a) Word
 - b) Powerpoint
 - c) Excel
 - d) Winrar
 - e) Flash

2. Yardım menüsüne klavyeden hangi tuş yardımıyla ulaşılır?
 - a) F10
 - b) F2
 - c) F1
 - d) F20
 - e) F39

3. Adobe Flash Professional CC yazılımının kurulması için hangi sistem gereksinimine kesinlikle ihtiyaç vardır?
 - a) Microsoft Windows XP işletim sistemine
 - b) 1GB RAM modülüne
 - c) 800X600 Ekran çözünürlüğüne
 - d) Mac OS X (32bit) işletim sistemine
 - e) Microsoft Windows 7 (64 bit)

4. Seçeneklerden hangisi Adobe Flash yazılımının kurulumunu başlatan dosyadır?
 - a) Deploy
 - b) Packages
 - c) Payloads
 - d) Set-up
 - e) Setup

5. Kare kare canlandırma yapılırken seçeneklerde verilen alanlardan hangisinden yararlanılır?
 - a) Zaman dogrusu
 - b) Katmanlar
 - c) Araç Kutusu
 - d) Özellikler paneli
 - e) Panel gurupları

6. Birden fazla görüntü ve ses dosyası içeren yapılara ne denir?
- a) Müzik
 - b) Slayt
 - c) Animasyon
 - d) Resim
 - e) Kitap
7. Seçeneklerden hangisi hareketli görsellerde kullanılan kodlama dilidir?
- a) C#
 - b) Visual Basic
 - c) Visual C
 - d) Delphi
 - e) ActionScript
8. Grafik çizimleri için gereken nesneler aşağıdakilerden hangisinde bulunur?
- a) Zaman doğrusu
 - b) Katmanlar
 - c) Araç Kutusu
 - d) Özellikler paneli
 - e) Panel gurupları
9. İstenen paneli program içerisine eklemek için aşağıdakilerden hangisi yapılır?
- a) Pencere menüsünden ilgili panelin yanındaki işaret kaldırılır.
 - b) Pencere menüsünden ilgili panelin yanına işaret eklenir.
 - c) Özellikler panelinden görünürlük değerleri değiştirilir.
 - d) Panel başlık çubuğuna sağ tıklanır ve kapat komutu verilir.
 - e) Panel başlık çubuğuna sağ tıklanır ve gizli panelleri otomatik göster işareti kaldırılır.
10. Bir animasyon dosyası hangi seçenekle izlenir?
- a) Dönüştür
 - b) İçer Aktar
 - c) Dışa Aktar
 - d) Yayınla
 - e) Kontrol Et

Cevap Anahtarı

1.e, 2.c, 3.e, 4.d, 5.a, 6.c, 7.e, 8.c, 9.b, 10.e

YARARLANILAN KAYNAKLAR

[1]Adobe Fireworks CS6, Yardımcı El kitabı. <http://helpx.adobe.com/tr/support>

[Erişim tarihi: 10 Kasım 2013].

[2]AKBUĞA, M. (2011). Grafik Animasyon I Erzurum: Atatürk üniversitesi ATAUZEM.

GRAFİKLER



İÇİNDEKİLER

- Grafikler
- Kılavuzlar ,Izgaralar ve Koordinatlar
- Araçlar Paneli



HEDEFLER

- Bu üniteyi çalıştıktan sonra;
- Kılavuzları,ızgaraları ve koordinatları çalışma sahnesine ekleyebilecek,
- Animasyon hazırlamak için gerekli olan araçları araç kutusundan seçebilecek,
- Tasarımlarda kullanılan grafikler üzerinde istenen düzenlemeleri kolayca yapabileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

ANİMASYON TASARIMI

Okt.
Mustafa AKBUĞA

ÜNİTE 2

GİRİŞ

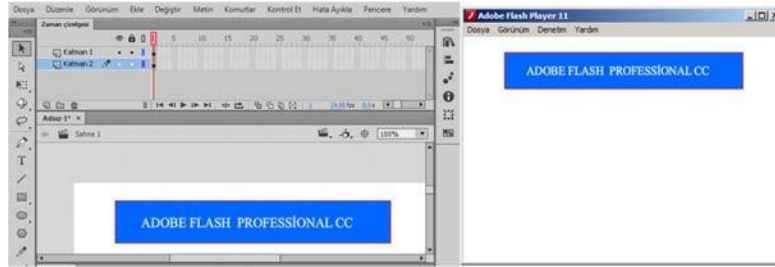
Adobe Flash Professional CC ya da önceki sürümlerini kullanmaya başladığınızda en fazla araç paneline ihtiyaç duyarsınız. Çalışma alanına eklenen Grafik, Düğme, Film Klibi gibi sembollerin tasarımını yapabilmek veya düzenlemek için Araç panelinde yer alan seçim araçları, çizim veya yazı araçları, boyama araçları gereklidir.

Araç panelinde bulunan bir aracı seçtiğinizde, seçim yapılan aracın özelliklerinin yer aldığı özellikler paneli, gerekli olur. Bu panelin çalışma ekranında bulunmadığı durumlarda, Pencere> Özellikler komutunu verilerek ekrana gelmesi sağlanmalıdır.

GRAFİKLER

Adobe Flash Pro yazılımı ile yapılan grafik tasarımlarında kullanılan gereçler Araç panelinde yer alır. Birçok aracın bulunduğu bu panelde yer almayan araçlar gerekli olursa Adobe Fireworks tasarım ortamı da kullanılabilir. Flash'ı Web için önemli bir geliştirme aracı hâline getiren özelliklerden biri de vektörel grafik modunu kullanmasıdır. Vektörel grafikler, matematiksel denklemlerle oluşturulmuş nesnelerdir. Bu grafikler nesnenin boyu, şekli, rengi, dış hattı ve konumu hakkında bilgi içeren vektörlerdir. Çözünürlükten bağımsızdırlar, büyütüldüğünde kalite kaybına uğramazlar.

Adobe Flash'la geliştirilen içerikler iki dosya yapısı kullanılarak oluşturulur ve yayınlanır. Bir kaynak ve diğeri ise flash filmi adı ile bilinen bu dosyanın sıkıştırılmış hâlidir. Kaynak dosya “*.fla” uzantısına sahiptir ve kaydedilerek daha sonra üzerinde değişiklik yapılabilir (Şekil 2.1). Bu dosya istenen hedef dosya hâlini alınca onu flash filmine dönüştürmek için ihraç (export) edilir ve dosyanın uzantısı “*.swf” hâlini alır (Şekil 2.1). Kaynak dosyalardan dışarıya QuickTime filmleri, animasyon (animated gif) ya da hareketsiz grafikler hâlinde de ihraç edebilir.



Örnek. fla(5Kb)

Örnek. swf(3Kb)

Şekil 2.1 Adobe Flash'ın Kullandığı Dosya Kayıt Türleri

Yeni Bir Dosya Oluşturmak

Adobe Flash Pro CC programında istediğimiz tasarımı yapabilmek için ilk önce yeni bir dosya oluşturmak gerekir. Bu nedenle dosya işlemlerini anlamak tasarımın sağlıklı geliştirilmesi için gereklidir.

Yeni bir dosya oluşturmak için iki yöntem kullanılır. Birincisi Dosya menüsünden yeni seçeneğini tıklamak, ikincisi ise yöntem Başlat Menüsü üzerinden Adobe Flash simgesine tıklamaktır. Birinci yöntemi istediğimizde karşımıza “Yeni belge” penceresi gelir ve buradan belgemizin tipini seçmemiz gerekir. Simgesinden boş belge açıldığında ise “Yeni oluştur” penceresinden uygulama tipi seçilmelidir (Şekil 2.2).



Adobe Flash ile tasarlanan dosyalar *.fla dosya uzantısıyla kaydedilmelidir. Farklı dosya yapılarında katmanlar ve zaman çizelgesi öğeleri kaybolur.



Yeni Adobe Flash dosyası oluşturmak için “Dosya” menüsünden “Yeni” seçeneği tıklanmalıdır.

. Birincisi Dosya menüsünden yeni seçeneğini tıklamak, ikincisi ise yöntem Başlat Menüsü üzerinden Adobe Flash simgesine tıklamaktır. Birinci yöntemi kullanarak yeni boş bir belge açmak istediğimizde karşımıza “Yeni belge” penceresi gelir ve buradan belgemizin tipini seçmemiz gerekir. Simge durumundan boş belge açıldığında ise “Yeni oluştur” penceresinden uygulama tipi seçilmelidir (Şekil 2.2).



Şekil 2.2. Yeni Dosya Oluşturma Komutları

Kayıtlı Bir Flash Belgesi Açmak

Önce bilgisayara kaydedilen bir flash belgesini tekrar açmak için iki yöntem kullanılır. Birinci yöntem, Dosya menüsünden “Aç” seçeneğine tıklamak, ikinci yöntem ise “Son Kullanılan Öğeyi Aç” bölümünden “Aç” komutu vermektir (Şekil 2.3).



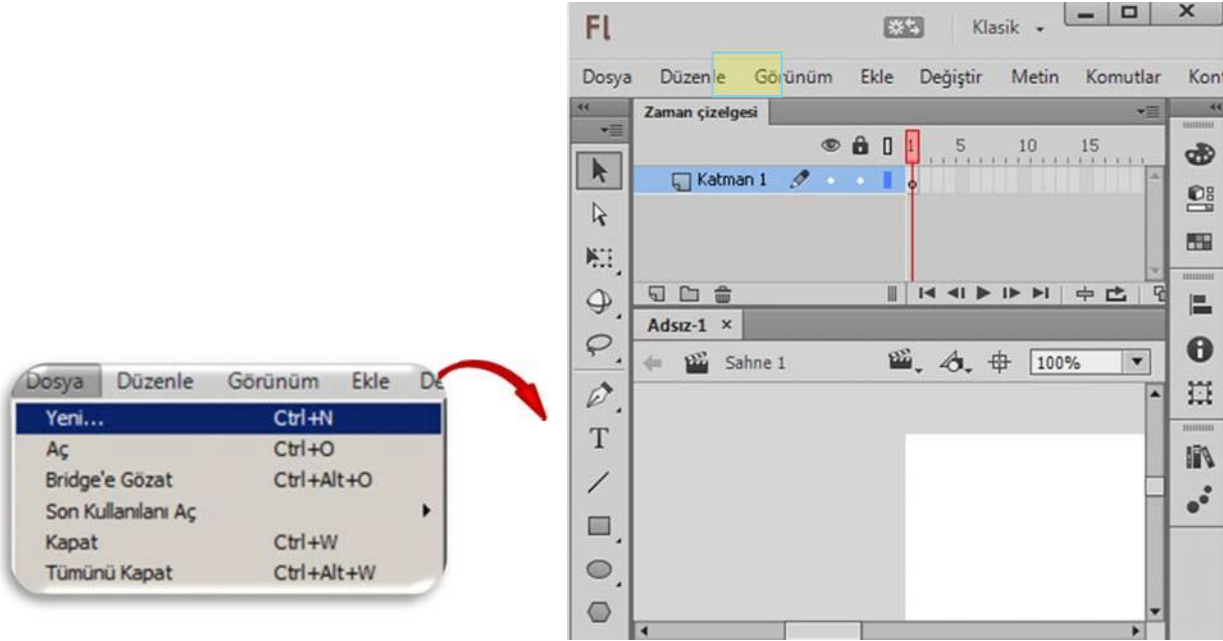
Şekil 2.3 Kaydedilmiş Bir Dosyanın Çağırılması

KILAVUZLAR, IZGARALAR VE KOORDİNATLAR

Adobe Flash çalışma ortamı, kullanıcının hassas ve ölçekli çizimler gerçekleştirmesi için gerekli özellikleri sunar. Izgaralı çalışma alanı sayesinde sayfa içindeki nesnelerin yerleştirilmesi ve konumları daha kolay belirlenir. Yatay ve dikey cetveller ile ölçekli nesneler çizebilirsiniz. Kılavuz çizgileri yardımıyla da çalışma ortamı içine yerleştirdiğiniz nesnelere referans noktaları atayabilirsiniz.

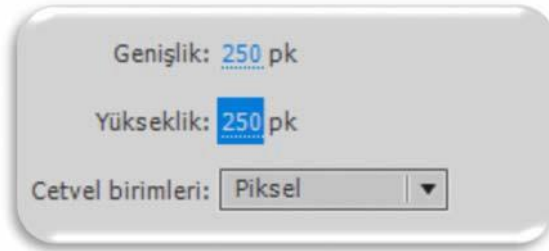
Izgaranın, cetvellerin ve kılavuzların görünür hâle getirilmesi için aşağıda verilen talimatları yerine getirilmelidir.

- 1- Dosya >Yeni komutunu vererek yeni bir çalışma sayfası açınız (Şekil 2.4).



Şekil 2.4 Dosya>Yeni Talimatının Yerine Getirilmesi.

- 2- Çalışma sayfasının boyutlarını 250pk – 250pk'e ayarlayınız (Şekil 2.5).



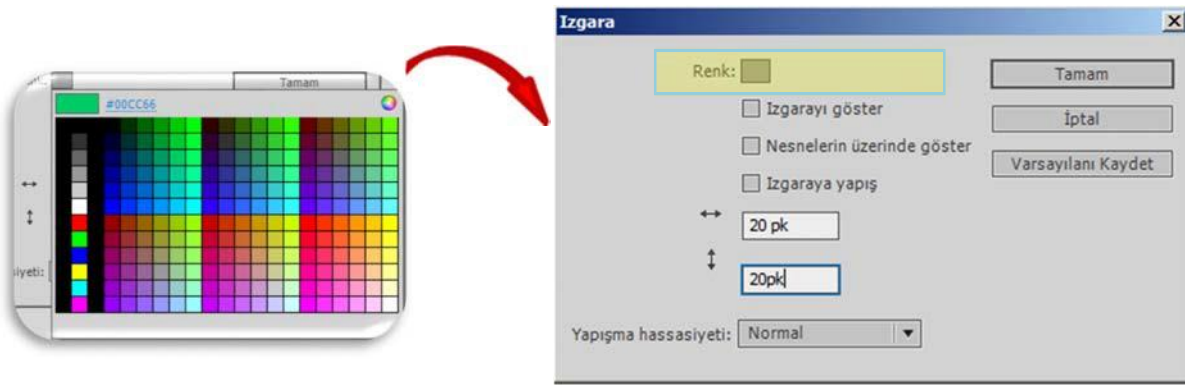
Şekil 2.5 Çalışma Alanı Boyutunun Belirlenmesi.

- 3- Görünüm > Izgara >Izgarayı göster komutunu vererek çalışma alanında birbirini kesen düşey ve yatay çizgilerin görünmesini sağlayınız (Şekil 2.6).
- 4- Görünüm>Izgara>Izgarayı düzenle komutuyla ızgara düzenleme penceresini açınız (Şekil 2.7).



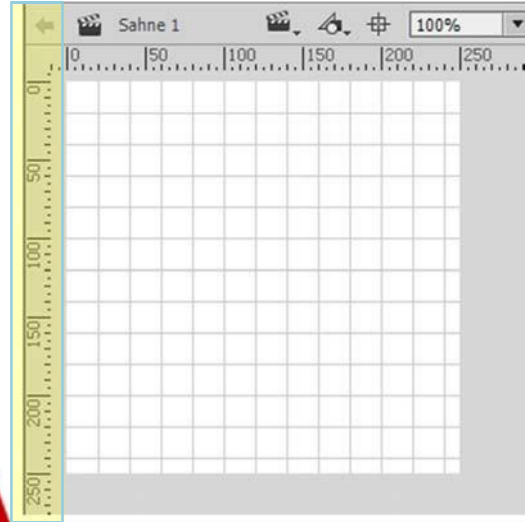
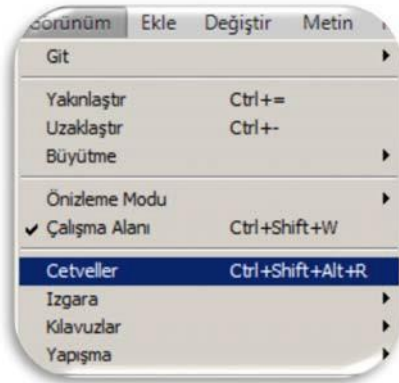
Şekil 2.6 Çalışma Alanı İçerisinde Izgaranın Gösterilmesi

- 5- Izgara genişliğini ve yüksekliğini 20 pk olarak belirleyiniz.



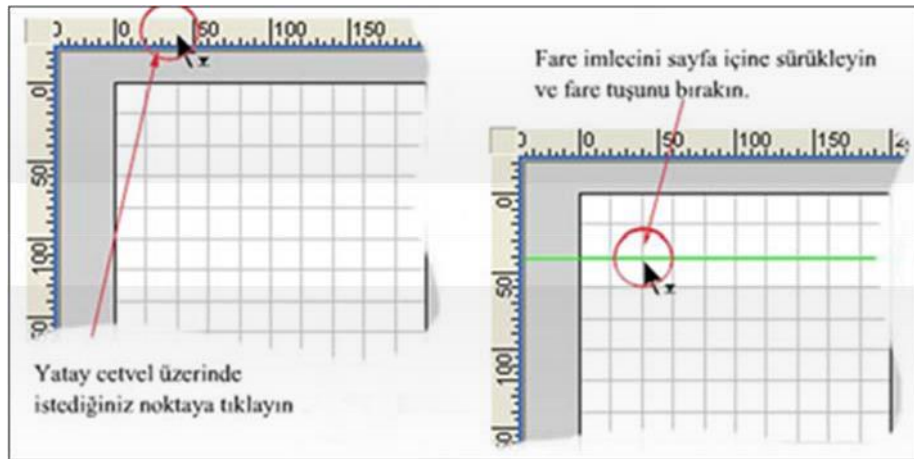
Şekil 2.7 Sahne İçerisindeki Izgaranın Görünüm Ayarları

- 6- Renk kısmından renk paletini açınız. Fare imleci tüp şeklini alınca renk paletinden veya Flash editörünün herhangi bir yerinden renk belirleyiniz. Tamam, düğmesine tıklayınız ve iletişim penceresini kapatınız. Izgara renginin değiştiğini göreceksiniz.
- 7- Görünüm > Cetveller komutuyla cetvelin gözükmelerini sağlayın (Şekil 2.8).



Şekil 2.8 Sahne İçerisinde Cetvel'in Gösterilmesi

- 8- Fare imlecini yatay cetvelin istediğiniz bir noktasına iken tıklayınız. (Fare imlecini şekli alması gerekir.) Farenin sol tuşuna parmağınızı basılı tutarak fare imlecini çalışma sayfasının istediğiniz bir noktasına sürükleyiniz. Farenin tuşunu bıraktığınızda Şekil 2.9'te görüldüğü gibi renkli yatay kılavuz çizgisi belirir.



Şekil 2.9 Sahne Üzerine Klavuz Çizgilerinin Getirilmesi



Adobe Flash'ın Araç panelindeki araçlarla grafik sembollerini düzenlemek zordur. Bu işlem için Adobe Fireworks kullanılmalıdır.

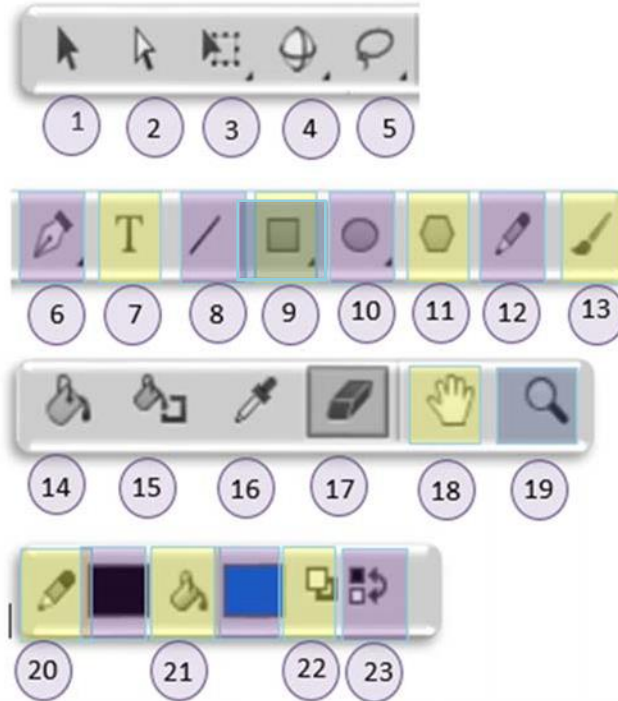
ARAÇLAR PANELİ (TOOLS PANEL)

Flash içinde grafiksel nesneler yaratılırken Araçlar (Tools) çubuğu kullanılır. Bu araç çubuğu iki alandan oluşur. Araç çubuğu düğmeleri ve seçenekleri bir araca tıkladığınızda aktif hâle geçer. Pek çok araç, bir aracın düğmesine tıkladığınızda araç çubuğunun alt tarafında görünen Seçenekler (Options) bölümüne sahiptir.

Her aracın üzerine geldiğinizde o aracın ismini ve parantez içinde de o araca ait harf görürsünüz. Bu harf o aracın klavye kısa yolunu temsil eder. Araçlar arasında hızlı geçiş yapmak için klavyeden bu kısa yollara basmanız yeterlidir.

Araçlar panelindeki araçlar, Sahne Alanı'nın görünümünü değiştirmenin yanı sıra, resim oluşturma, boyama, seçme ve değiştirmeyi sağlar. Araçlar paneli (Şekil 2.10) gösterildiği gibi dört bölüme ayrılmıştır):

- ✓ Araçlar alanı çizim, boyama ve seçim araçlarını içerir.
- ✓ Görünüm alanı, uygulama penceresinde yakınlaştırma ve kaydırmaya yarayan araçlar içerir.
- ✓ Renkler alanı kontur ve dolgu renkleri için değiştiriciler içerir.
- ✓ Seçenekler alanı geçerli araç için değiştiriciler içerir. Değiştiriciler aracın boyama veya düzenleme işlemlerini etkiler.



Şekil 2.10 Araç Panelindeki Araç Türlerinin Özelliklerinin Açıklanması

Araçlar paneli üzerinde yer alan ve Şekil 2.10 ile gösterilen araçlar ekranda birçok ilave seçeneğe sahiptir. Bu seçeneklerin ayrıntıları aşağıda açıklanmıştır.

1. Seçim aracı (Selection Tool): Adobe Flash CSS yazılımında en sık kullanılan araçlardan biridir. Çalışma alanındaki nesneler bu araç yardımıyla seçilerek, şekilleri düzenlenebilir veya yeri değiştirilebilir. Birden fazla işlem yapabilme yeteneğine sahiptir. Bu araç seçiliyken imleci bir nesnenin kenarına getirdiğinizde imleç iki şekilden birini alır:

Eğer yaklaşılacak kenar yumuşak kıvrımlardan oluşuyorsa ya da tamamen düzse



şeklini alır.



2. **Alt Seçim aracı (Subselection Tool):** Alt seçim aracıyla seçilen bir çerçevenin içinde kalan alan da seçilir. Ayrıca seçilen çerçeve üzerinde çeşitli tutma noktaları oluşturularak şeklin bu noktalardan değiştirilmesi sağlanır.

3. **Serbest Dönüştürme aracı (Free Transform Tool):** Bu aracı nesneleri, metinleri, grup altına alınmış öğeleri ve Flash'ın örnek öğelerini istenildiği gibi döndürmek ve boyutlandırmak amacıyla kullanılır.

4. **3B Döndürme aracı:** Bu araç nesneleri, kendi etrafında 3 boyutlu olarak döndürmek ve taşımak amacıyla kullanılır.

5. **Kement aracı (Lasso Tool):** Serbest seçim yapmak için kullanılır. Fare imlecini seçime başladığınız noktaya kadar sürüklemenize gerek yoktur. Başladığınız noktaya sürüklemeyi bitirdiğiniz nokta arasını seçili duruma getirir.

6. **Kalem aracı (Pen Tool):** Bu araç serbest çizim sağlar. Nokta nokta tıklayıp eğip bükme suretiyle çizimler yapılabilir.

7. **Metin aracı (Text Tool):** Resimler ne kadar bilgilendirici olursa olsun, yine de düğmeler için etiketler, reklam başlıkları için metin oluşturmak gerekmektedir. Çalışma dosyasına metin aracı kullanarak metin eklenebilir. Simge seçildiğinde özellikler penceresinde tüm metin biçimlendirme denetimleri görüntülenir. Metni yazmadan önce veya çalışma alanına yerleştirdikten sonra biçimlendirme değişiklikleri yapılabilir.

8. **Çizgi aracı (Line Tool):** Vektör araçlarından biri olan çizgi aracı çalışma alanı içerisine çizgi eklemek için kullanılır.

9. **Dikdörtgen aracı (Rectangle Tool):** Vektör araçlarından biri olan dikdörtgen aracı araç kutusunda dikdörtgen çizmek için kullanılır. Dikdörtgen yerine kare çizilmek isteniyorsa **Shift** tuşuna basılarak (basılı tutularak) çizim yapılmalıdır. Merkezden dikdörtgen çizilmek isteniyorsa **Alt** tuşu kullanılır. Merkezden ve kare çizilmek isteniyorsa **Shift ve Alt** tuşları birlikte kullanılmalıdır.

10. **Oval aracı (Oval Tool):** Vektör araçlarından biri olan oval aracı araç kutusunda elips çizmek için kullanılır. Daire çizilmek isteniyorsa **Shift** tuşu kullanılır. Merkezden daire çizilmek isteniyorsa **Alt** tuşu kullanılır.

11. **ÇokgenYıldız aracı (PolyStar Tool):** Vektör araçlarından biri olan ÇokgenYıldız aracı çokgen çizmek için kullanılır. Yıldız çizilmek isteniyorsa özellikler panelinden, seçenekler alt menüsüne girilerek stil, Kenar sayısı, Yıldız nokta boyutu ayarlanmalıdır.

12. **Kurşun kalem aracı (Pencil Tool):** Bu araç kalem aracı ve çizgi aracı gibi çizgi çizmeyi sağlar. Bu aracın çizgi çizmesi için farenin sol tuşuna basılı tutulması gereklidir. Çizilen çizgi sadece dış çizgi rengindedir.

13. **Fırça aracı (Brush Tool):** Bu araç aynı kurşun kalem aracı gibi çizgi çizmeyi sağlar. Bu aracın çizdiği çizgiler dolgu rengine sahiptir ve kurşun kalem kalındır. Çizginin kalınlığı çizgi ve kurşun kalemin ayar seçenekleriyle ayarlanabilir.

14. **Kova Boyası aracı (Paint Bucket):** Grafik öğelerinin çerçeve renklerini değiştirmek için kullanılır. Çerçevesi olmayan grafik öğelerine de çerçeve ekler.



Adobe Flash Araç panelinde Flim Klibi sembollerini kendi x,y,z eksenleri etrafında döndürmek amacıyla 3B Döndürme aracı bulunur.



Araç panelindeki bazı araçların özellik paneli üzerinden düzenleme seçenekleri yoktur. Bu araçlara ait seçenek modları yine Araç paneli üzerinde yapılır.

1. **Mürekkep Şişesi aracı (Ink Bottle Tool):** Grafik öğelerinin dış çizgi özelliklerinden Kontur özelliği kullanılarak, rengini, boyunu ve stilini değiştirmeyi sağlar.

2. **Damlalık aracı (Eyedropper Tool):** Bu araç ile bir bölgeye tıklandığında o bölgeyi kendi rengine dönüştürür. Bu seçim ile renk kopyalama yapabilirsiniz. Nesnelerin rengini değiştirmek istediğinizde kullanılır.

3. **Silgi aracı (Eraser Tool):** Çizilen grafik öğelerini farklı biçimlerde silmek için kullanılır. Metin ve Flash bileşenleri gibi nesneleri silmez.

4. **El aracı (Hand Tool):** Yapılan çalışmada bazen çalışma alanı gereğinden çok fazla büyütülür. Çok büyütülmüş alanlarda nesnelerin yeri bulmak ve çalışmak oldukça zorlaşır. El aracı kaydırma çubuğunda yatay ve dikey ayarları tek başına yaparak bu işlemleri kolaylaştırır.

5. **Yakınlaştırma aracı (Zoom Tool):** Bir öğeyi yakınlaştırmak için Yakınlaştırma aracı kullanılır. Yakınlaştırma aracında yakınlaştırma ve uzaklaştırma arasında geçiş yapmak için Büyüt ve Küçült değiştiricilerini (Yakınlaştırma aracı seçiliyken, Araçlar panelinin seçenekler alanında bulunur) kullanın ya da Alt tuşuna basıp tıklayın.

19. **Dış çizgi rengi ayarlama (Stroke Color):** Kurşun kalem ile çizilen çizgilerin rengini gösterir. Bu araç sayesinde şekillerin dış çerçeve rengini değiştirebilirsiniz.

20. **Dolgu Rengi ayarlama (Fill Color):** Fırça aracı ile çizilen çizgilerin rengini gösterir. Bu araç sayesinde şekillerin dolgu rengini değiştirebilirsiniz.

23. **Siyah-Beyaz (Black and White):** Bu da şekillerimizin rengini siyah & beyaz yapar.

24. **Renk Değiştir aracı (Swap Colors):** Seçtiğimiz şeklin iç dolgu ve dış çerçeve renklerinin yerini değiştirir. Yani dış çerçeve rengi iç dolgu rengi olur, iç dolgu rengi de dış çerçeve rengi olur.

Yeni bir grafik oluşturma ve düzenleme

Aşağıda konu anlatımında Adobe Flash uygulamasındaki araç kutusundaki düzenleme araçlarının nasıl kullanılacağı göstermektedir.



Seçim aracı (Selection Tool)

Seçme (Selection) aracı, grafiklerinizi seçmek, taşımak için kullanılan bir araçtır. Bir nesne üzerinde herhangi bir işlem yapabilmek için önce onu seçmemiz gerekir. Bu anlamda önemli ve sık kullandığımız bir araçtır.

Flash programında çizdiğimiz grafiksel öğeler 2 bölümden oluşur; çizgi (stroke) ve dolgu (fill). Bir şekli seçerken dikkatli olmamız gerekmektedir. Sadece dolguyu veya sadece çizgiyi seçebiliriz. Ama şeklin üzerine tek tıklamak yerine çift tıklarsak her iki bölümü de seçerek işlem yapabiliriz. Ya da şekli bir kare içine alacak şekilde de seçebiliriz (Şekil 2.11).



Şekil 2.11 Seçim Aracının Şekil Üzerindeki Etkisinin Gösterilmesi

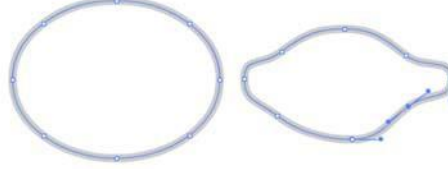


Seçim aracı ile bir grafik nesnesi çerçeve ve dolgu yapısıyla birlikte seçilmelidir. Tam seçilmeyen nesnenin taşınma esnasında şekil yapıları bozulabilir.

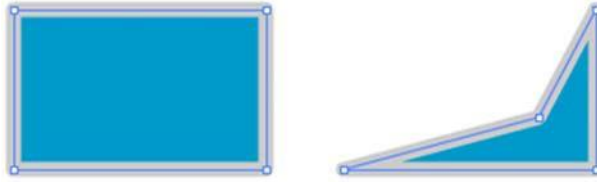


Alt Seçim aracı (Subselection Tool),

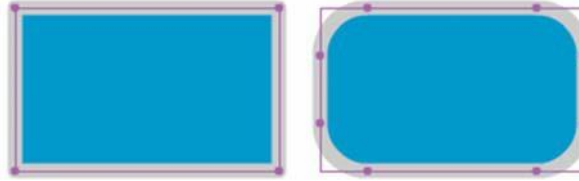
Seçme (Selection) aracına benzer, ancak farkı tüm nesneyi seçmek yerine şeklin yolu üzerindeki temel noktaları seçmesi ve onlarla çalışmayı sağlamasıdır. Şekillerin türlerine göre özellikleri değişir. Temel öge araçlarıyla çizilen şekiller üzerinde geometrik değişiklik yapmaz (Şekil 2.12-13-14).



Şekil 2.12 Alt Seçim Aracının Oval Şekil Üzerindeki Etkisinin Gösterilmesi



Şekil 2.13 Alt Seçim Aracının Dikdörtgen Şekli Üzerindeki Etkisinin Gösterilmesi



Şekil 2.14 Alt Seçim Aracının Dikdörtgen Temel Öge Aracı Üzerindeki Etkisinin Gösterilmesi

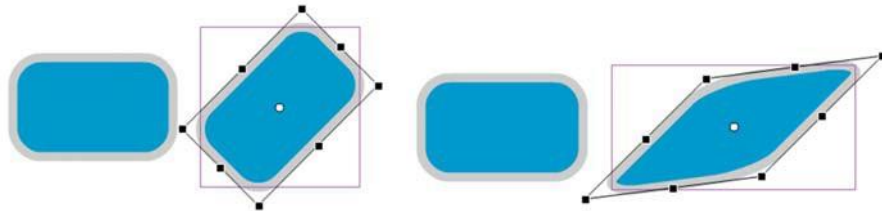


Serbest dönüştürme aracı ile her türlü sembol büyültülüp küçültülebilir.



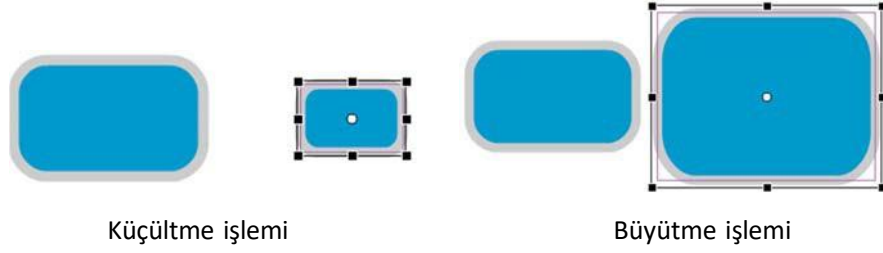
Serbest dönüştürme aracı (Free Transform Tool): Bu aracı, çizilen şekilleri yada çalışma sayfasındaki bir nesneyi büyültüp küçültmek, uzatıp eğmek, çevirmek, şeklini değiştirmek için kullanabilirsiniz. Bu araç da özellikle büyültüp küçültmek için çok sık kullanılan bir araçtır. Bu aracı seçtiğimizde de Seçenekler (Options) bölümünde yapabileceğimiz 2 işlem vardır. Şimdi onları tek tek tanıyalım.

1. Döndür ve Uzat (Rotate and Skew): Şekilleri döndürmek ve uzatmak için kullanabilir (Şekil 2.15).



Şekil 2.15 Serbest Dönüştürme Aracıyla Şekilleri Döndürme ve Uzatma İşleminin Gösterilmesi

2. Büyüt-Küçült (Scale): Bu aracı seçilen şekli büyütüp küçültmek için kullanabilir (Şekil 2.16).



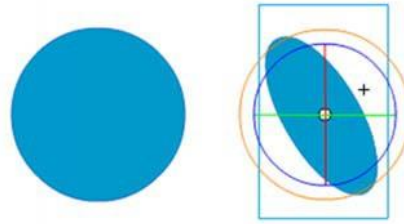
Şekil 2.16 Serbest Dönüştürme Aracıyla Şekilleri Büyütüp Küçültme İşleminin Gösterilmesi



3D Döndürme aracı,

Araç kutusundaki bu öğeyi kullanabilmek için önce bu nesneyi seçilerek film klipi sembolüne çevirmelisiniz. Bu işlem sonrası Çevirme ve 3D Döndürme araçlarını kullanarak onları x ya da y ekseninde hareket ettirebilir veya döndürebilirsiniz. Bir film klibine 3B perspektif efekti eklemek için önce bu efektlerden herhangi birini bir film klibine uyguladıktan sonra, Flash Professional uygulaması bunu bir 3B film klipi olarak değerlendirir ve film klipi her seçildiğinde, renklendirilmiş eksen göstergesi film klipinin üzerini kaplamış şekilde görüntülenir.

Seçildiğinde, renklendirilmiş eksen göstergesi film klipinin üzerini kaplamış şekilde görüntülenir (Şekil 2.17).



Şekil 2.17 3D Döndürme Aracıyla Daire Üzerindeki Etkisinin Gösterilmesi

Sahne Alanı'nda 3B nesneleriniz varsa, FLA dosyanızın Perspektif Açısı ve Ufuk Noktası özelliklerini ayarlayarak bu nesnelerin tümüne grup olarak bazı 3B efektler ekleyebilirsiniz. *Perspektif Açısı özelliği*, Sahne Alanı'nın görünümünü yakınlaştırma efektine sahiptir. *Ufuk Noktası özelliği*, Sahne Alanı'ndaki 3B nesneleri yatay kaydırma efektine sahiptir. Bu ayarlar, yalnızca 3B dönüştürme veya çevirme uygulanmış film kliplerinin görünümünü etkiler.

Seçim araçları, Şekillerin belirli bölgelerini seçmek için kullanılabilir.

Seçenekler (Options) bölümünde 3 seçenek aktif hale gelir. Bunlar;

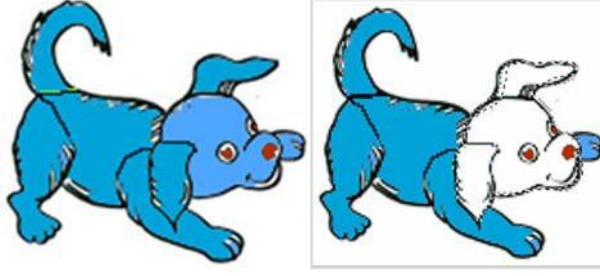
Kement aracı (Lasso Tool): Şekli belirsiz bölgeleri seçerken yapabileceğimiz ayarları gösterir (Şekil 2.18).

Sihirli Çubuk (Magic Wand) : Tek renk olan bölgeyi farenin sol tuşu ile tek bir tıklama yaparak seçmeye yarayan bir araçtır (Şekil 2.19).

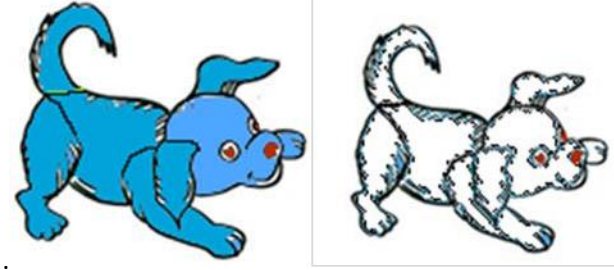
Çokgen aracı (Polygon Mode) : Bu seçenekte de yapacağımız seçimi noktadan noktaya düz bir şekilde seçmemizi sağlar (Şekil 2.20).



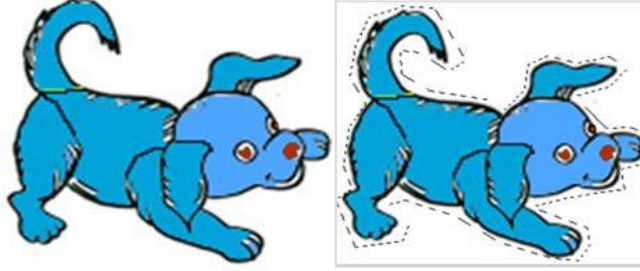
Grafiklerin aynı renk tonlarına ait bölgelerinin seçiminde sihirli çubuk aracı diğer araçlardan daha başarılı seçim yapar.



Şekil 2.18 Kement Aracı ile Görüntüde Bir Bölgenin Seçilerek Renginin Değiştirilmesi



Şekil 2.19 Sihirli Çubuk Aracı ile Görüntü Üzerinde Aynı Renk Bölgelerinin Seçilmesi ve Düzenlenmesi

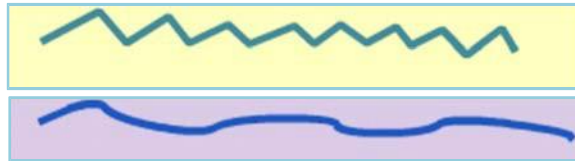


Şekil 2.20 Çokgen Aracı ile Görüntüde Daha Keskin Çizgiler Yardımı ile Seçim Yapılması

Dolmakalem aracı (Pen Tool),

Dolmakalem aracı (Pen tool) ile hassas bir şekilde çizgi çizerek çeşitli düz veya eğri çizgi dilimleri ve kapalı şekiller oluşturabiliriz (Şekil 2.21).

Şekil 2.21 Dolmakalem Aracı ile Yapılan Çizgi Çalışması



Metin aracı (Text Tool),

Metin aracında bir yazının font ayarlarını değiştirmek için Aile özelliğinden faydalanılır.

Metin (Text) aracı ile Adobe Flash programında yazı yazılabilir ve yazıları düzenleyebiliriz. Adobe Flash programında yazı yazmak için Metin (Text) aracı seçilir. Çalışma alanına bir kutu çizilir ve bu kutu içerisine istenen yazı yazılır. Metin aracından çıkarak, Araç kutusundaki başka bir araç kullanmak zorunda kalınırsa (örneğin yazıyı taşımak için) tekrar önceki yazı yazdığımız kutunun içine dönebiliriz. Bunun için Seçim (Selection) aracı ile aynı yazı tekrar seçilerek kaldığımız yerden devam edebiliriz. Bunun için kutunun üzerine farenin sol tuş ile 2 kez tıklamanız veya tekrar Text (Yazı) aracını seçerek yazının içine tıklamanız yeterli olacaktır. Ayrıca Flash program pencerenizin alt bölümündeki Özellikler (Properties) bölümünü

kullanarak yazının özelliklerini ayarlayabilirsiniz (Şekil 2.22-24).

Atatürk Üniversitesi

Atatürk Üniversitesi
Atatürk Üniversitesi
Atatürk Üniversitesi
 Atatürk Üniversitesi
 Atatürk Üniversitesi



Metin aracı ile oluşturulan Dinamik Metin türü, animasyonun çalışması esnasında içeriğinin değiştirilmesinde kullanılır.

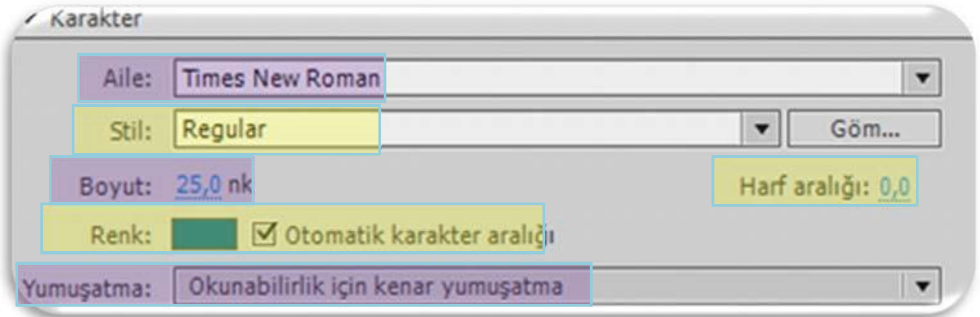
Şekil 2.22 Metin Aracıyla Yazılmış Yazıların Özelliklerinin Uygulanması

T

Statik Metin

Şekil 2.23 Metin Aracının Türünün Seçimi

Özellikler panelinde yer alan Yazı Bloğu Seçenekleri, Static Metin, Dinamik metin, Giriş metni olarak bu bölümden seçilir. Bu seçenekler 3.ünite de ayrıntılı olarak incelenecektir.



Şekil 2.24 Metin Aracının Karakter Özellikleri

Aile: Yazı Tipi, Seçeneklerden istenen font seçilir.

Stil: Yazı Stili, kalınlık, italik düz stillerinden biri

seçilir. **Boyut:** Yazı boyutu, yazının büyüklüğü ayarlanır.

Renk: Yazı rengi, yazının tasarıma uygun rengi ayarlanır.

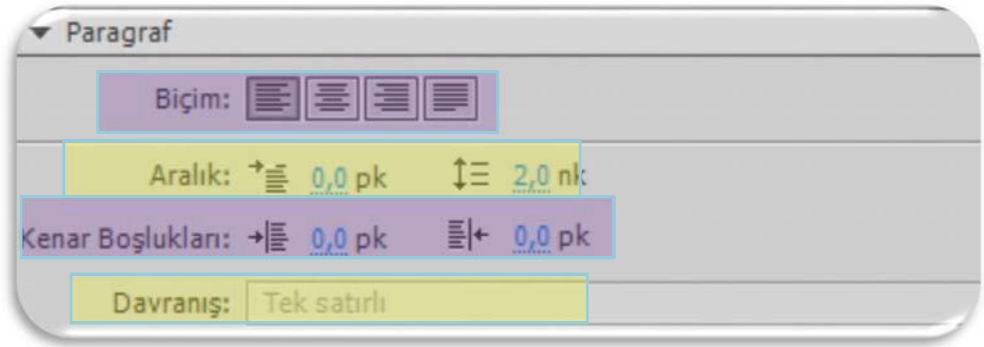
Harf Aralığı: Harfler arası mesafe bu bölümden ayarlanır.

Yumuşatma: Ekrada gösterilen harflerin kenar düzenlemeleri bu bölümden ayarlanır, bu bölüm seçenekleriyle istenen her fontun kenar düzeltmeleri kolayca yapılır. Aygıt Fontlarını kullan, Bitmap metni vb. seçeneklere sahiptir.



Sahne içerisine bir metin eklemeyen önce özellikler penceresinden gerekli ayarlamalar yapılmalıdır.

Adobe Flash programı içerisinde metin aracıyla yazılmış yazıların, metin zarfı içerisindeki kenar boşlukları ve konumları paragraf bölümünden ayarlanır (Şekil 2.25).



Şekil 2.25 Metin Aracının Özellikler Panelinde Yer Alan Paragraf Seçenekleri

Biçim: Hizalama, Sola Hizala, Ortaya hizala, sağa Hizala, Yasla seçenekleri kullanılır.

Aralık: Satır başı ve satır aralıkları buradan ayarlanır.

Kenar Boşlukları: Kenar boşlukları sağdan ve solda bu kısımdan ayarlanır.

Davranış: Yazı üzerine davranış belirlemek amacıyla kullanılır.



Çizgi aracı (Line Tool),

Çizgi aracı çizgi çizmemizi sağlar. Bu araç ile çalışırken özellikler (properties) bölümünde aracının özelliklerini değiştirerek çalışabiliriz (Şekil 2.26).



Şekil 2.26 Çizgi Aracı ile Kontur Özelliklerinin Değiştirilmesiyle Oluşan Çizgiler

Dikdörtgen aracı (Rectangle Tool),

Dikdörtgen aracı araç kutusunda dikdörtgen çizmek için kullanılır. Kare çizilmek isteniyorsa **Shift** tuşu kullanılır. Merkezden dikdörtgen çizilmek isteniyorsa **Alt** tuşu kullanılır. Merkezden ve kare çizilmek isteniyorsa **Shift ve Alt** tuşları birlikte kullanılır. İki seçeneği vardır (Şekil 2.27). Birinci seçenekte Dikdörtgen aracı varken ikinci seçenekte ise Dikdörtgen Temel Öğe Aracı vardır. Dikdörtgen temel aracının köşeleri istenirse özelliklerinden yuvarlatılabilir (Şekil 2.28). Her iki seçenekte verilen araçların özellikleride çizgi aracının özellikleriyle aynıdır.



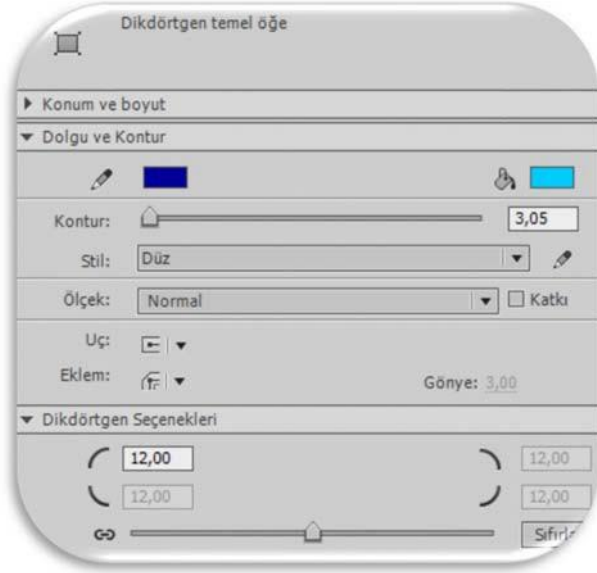
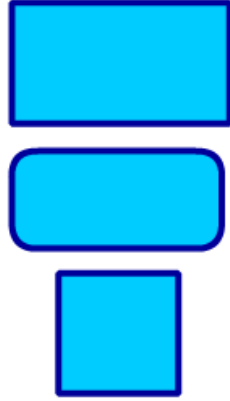
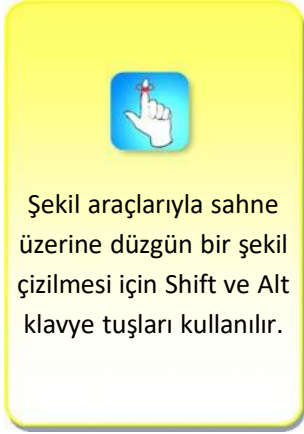
Şekil 2.27 Dikdörtgen Aracı Menüsünde Yer Alan Araçlar



Metin aracının Özellikler panelindeki yapılan değişiklikler Sahne içerisindeki yazı nesnesini etkilemez.



Özellikler panelinde yapılan değişiklikler ancak nesnenin önceden seçilmesiyle etkin olur.



Şekil 2.28 Dikdörtgen Temel Öge Aracındaki Özellikler ve Değişimler ile Oluşturulan Şekiller

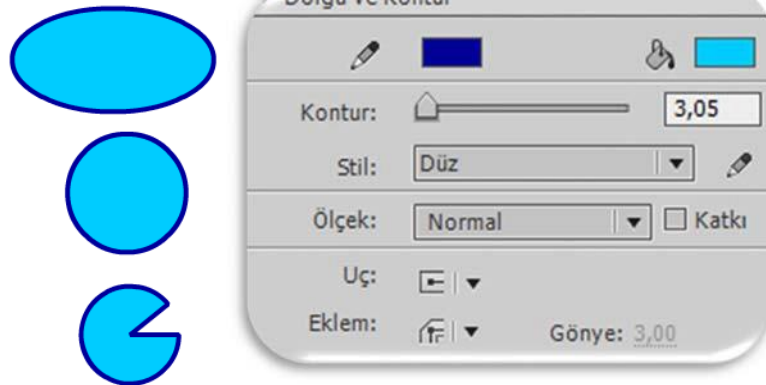


Oval aracı (Oval Tool),

Oval aracı araç kutusunda elips çizmek için kullanılır. Daire çizilmek isteniyorsa **Shift** tuşu kullanılır. Merkezden daire çizilmek isteniyorsa **Alt** tuşu kullanılır. İki seçeneği vardır. Birinci seçenekte Oval aracı varken ikinci seçenekte ise Oval Temel Öge Aracı vardır (Şekil 2.29). Oval temel aracının köşeleri istenirse özelliklerinden yuvarlatılabilir (Şekil 2.30). Her iki seçenekle verilen araçların özellikleride çizgi aracının özellikleriyle aynıdır.

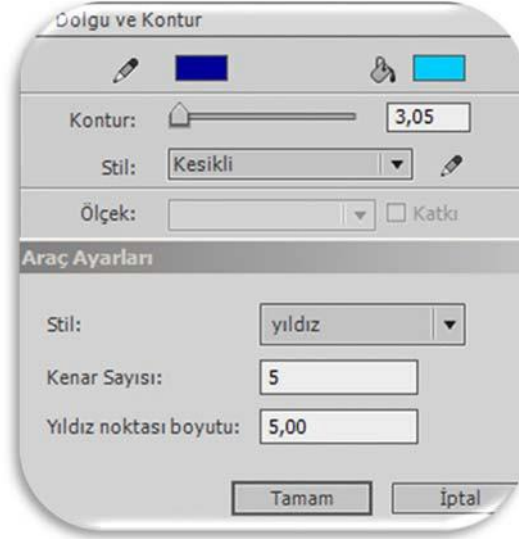
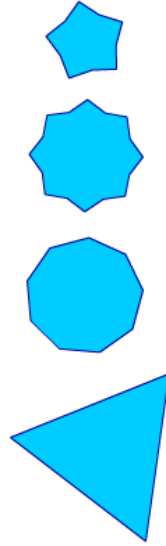
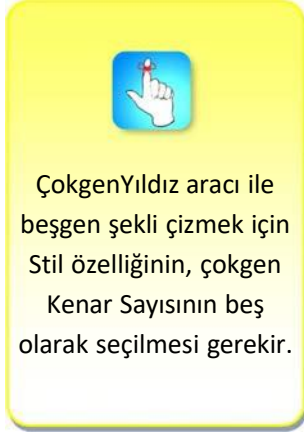


Şekil 2.29 Oval Aracı Menüsünde Yer Alan Araçlar



Şekil 2.30 Oval Temel Öge Aracındaki Özellikler ve Değişimler ile Oluşturulan Şekiller

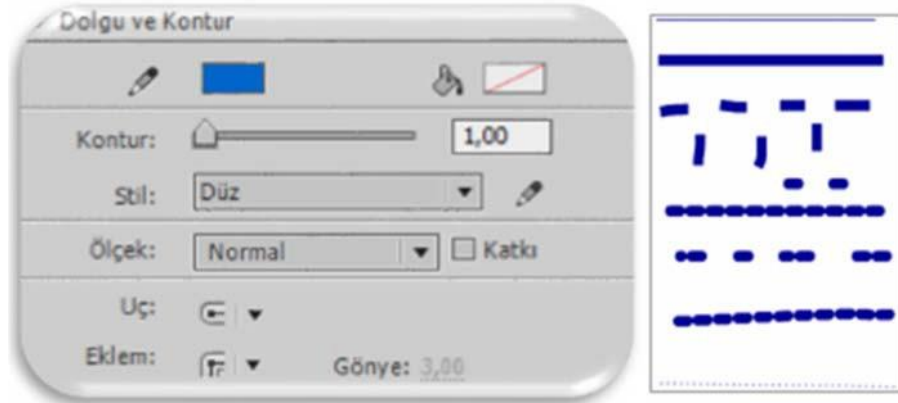
ÇokgenYıldız aracı (PolyStar Tool), Vektör araçlarından biri olan ÇokgenYıldız aracı araç kutusunda çokgen çizmek için kullanılır. Yıldız çizilmek isteniyorsa özellikler panelinden, seçenekler alt menüsüne girilerek stil, Kenar sayısı, Yıldız nokta boyutu ayarlanmalıdır (Şekil 2.31).



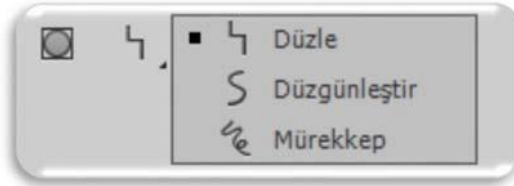
Şekil 2.31 Çokgenyıldız Aracındaki Özellikler ve Değişimler ile Oluşturulan Şekiller

Kalem aracı (Pencil Tool),

Kurşun kalem (Pencil) aracı ile Özellikler (Properties) bölümündeki seçenekleri kullanarak fare yardımıyla düz veya eğimli çizgiler çizebilirsiniz (Şekil 2.32-33)



Şekil 2.32 Kalem Aracındaki Özellikler ve Değişimler ile Oluşturulan Şekiller

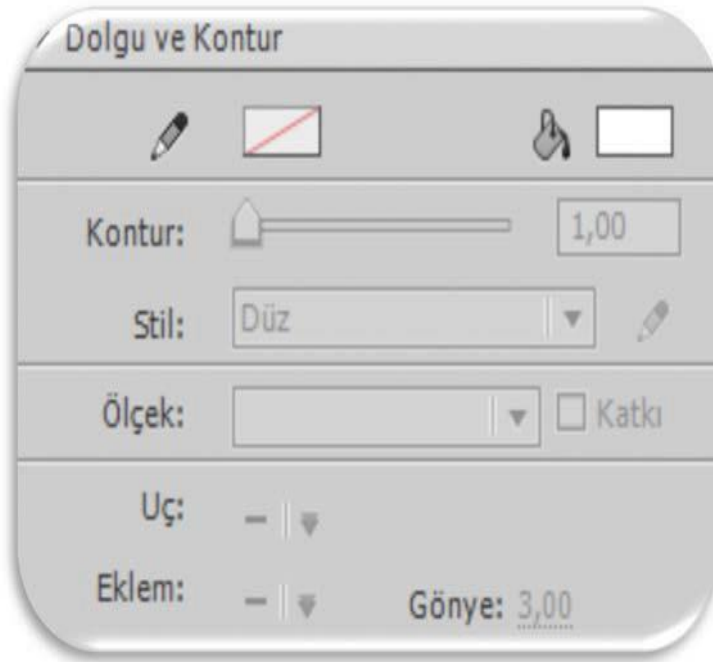


Şekil 2.33 Kalem Aracı ile Oluşturulan Çizgilerin Çizim Modları

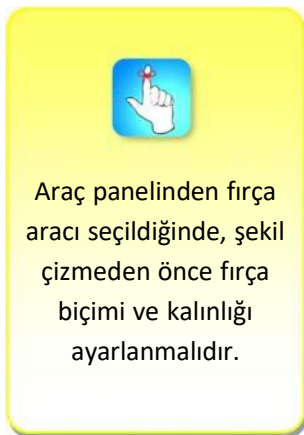


Fırça aracı (Brush Tool),

Fırça (Brush) aracını alanları düz renkle boyamak için kullanabilirsiniz. Bu araçta bulunan seçenekler araç panelinden yapılır. Özellikler panelindeki seçenekler pasif haldedir (Şekil 2.34). Seçenekler (Options) bölümünde 5 ayrı seçeneğe sahiptir (Şekil 2.35-36).



Şekil 2.34 Fırça Aracının Özellikler Panelindeki Seçeneklerin Pasif Durumu



Şekil 2.35 Fırça Aracının Araç Panelindeki Fırça Modları ve Şekil Üzerindeki Etkileri

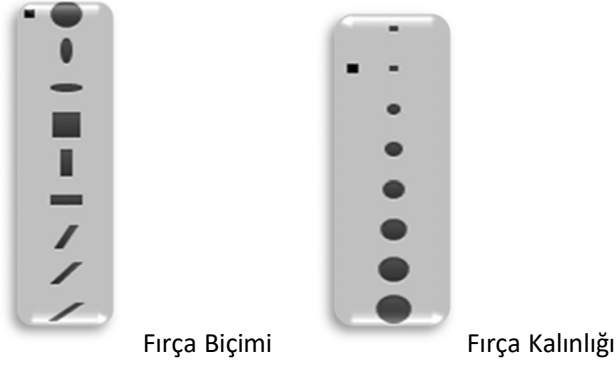
Normal Boya(Paint Normal) : Normal Boyar

Dolguları Boya(Paint Fills) : Dolguları boyar, Çizgileri boyamaz.

Arkayı Boya (Paint Behind) : Şeklin çevresini boyar, üzerini boyamaz ve arkasını boyamış görüntüsü yaratır.

Seçimi Boya(Paint Selection) : Seçili alanı boyar, onun dışında bir yeri boyamaz.

İçini Boya(Paint Inside) : Dolgulu alanın yalnız içini boyar.



Şekil 2.36 Fırça Aracının Araç Paneli Üzerinde Gösterilen Fırça Biçim ve Kalınlık Çeşitleri



Kova boyası aracı (Paint Bucket),

Bir şeklin dolgu rengini değiştirmekte amacıyla kullanılır. Options (Seçenekler) bölümünde Paint Bucket (Boya Kutusu) için 4 ayrı seçenek vardır(Şekil 2.37-39).



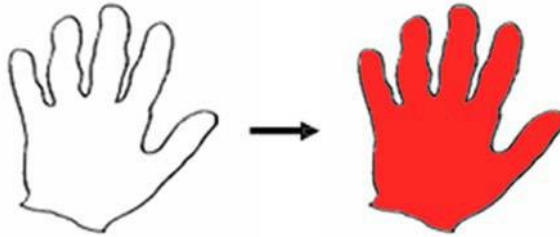
Şekil 2.37 Kova Aracının Araç Paneli Üzerindeki Boşluk Boyutu

Arkalı Kapatma (Don't Close Gaps) : Boşlukları boyama

Küçük Aralıkları Kapat (Close Small Gaps) : Küçük boşlukları boyama

Orta Aralıkları Kapat (Close Medium Gaps) : Orta büyüklükte boşlukları kapat

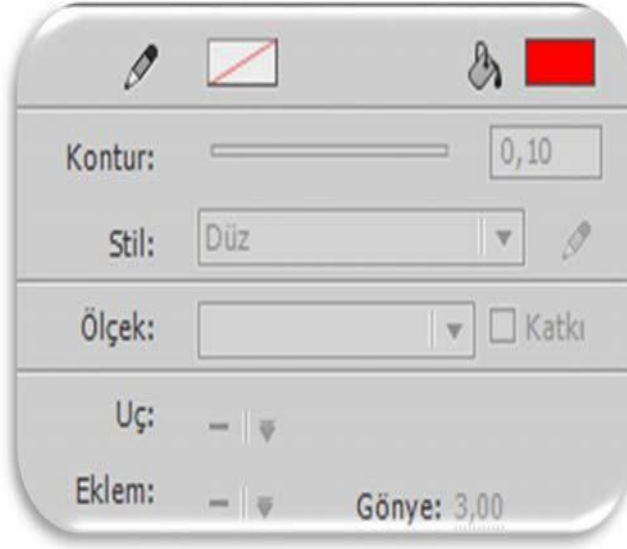
Büyük Aralıkları Kapat(Close Large Gaps) : Büyük boşlukları kapat



Şekil 2.38 Kova Boyası Aracıyla Şeklin Dolgu Renginin Ayarlanması



Kova aracı ile çizilmiş nesnelerin dolgu rengi ayarlanırken yeni renk seçeneği, kova sembolünün üzerindeki ok şeklinin işaret ettiği alanda etkilidir.



Şekil 2.39 Kova Boyası Aracının Özellikler Panelindeki Seçeneklerin Pasif Durumu



Mürekkep Şişesi aracı (Ink Bottle Tool),

Grafik öğelerinin dış çizgilerinin görünümünü, Mürekkep şişesi aracı yardımıyla, rengini, boyunu ve stilini değiştirmeyi sağlar(Şekil 2.40-41)



Şekil 2.40 Mürekkep Şişesi Aracıyla Oluşturulan Çizgi Şekilleri



Damlalık aracı (Eyedropper Tool),

Damlalık (Dropper) aracı çalışma alanında bulunan bir şeklin dolgu veya çizgi stilini almanızı ve başka bir şekle uygulamanızı sağlar. Bu aracın özellikler panelinden ayarlanan seçeneği bulunmamaktadır (Şekil 2.42).



Şekil 2.41 Damlalık Aracının Şekiller Üzerindeki Etkileri



Silgi aracı (Eraser Tool),

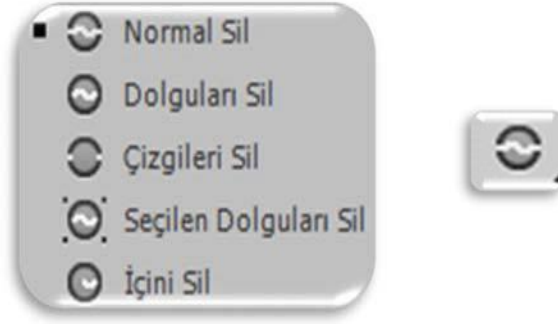
Çizilen grafik öğelerini farklı biçimlerde silmek için kullanılır. Metin ve Flash bileşenleri gibi nesneleri silmez. Çizgileri, dolguları, basit şekilleri tamamen veya kısmen silebilir. Seçenekler (Eraser Options) bölümünde Flash'ın çizimindeki alanları nasıl sileceğini belirleyen beş seçeneğe sahiptir (Şekil 2.43).



Mürekkep Şişesi aracı ile sahne üzerindeki çizilmiş nesnelerin çizgi rengi ve stili yeniden düzenlenebilir.



Sahne üzerindeki bir bölgenin renk tonu damlalık aracıyla kolayca kopyalanabilir.



Şekil 2.42 Silgi Aracının Araç Panelindeki Silgi Modları

Normal Sil (Erase Normal) : Çizgileri ve dolguları siler

Dolguları Sil (Erase Fills): Sadece dolguları siler ve çizgileri etkilemez.

Çizgiler Sil (Erase Lines) : Sadece çizgileri siler, dolguları etkilemez.

Seçilen Dolguları Sil (Erase Selected Fills) : Sadece seçili durumdaki dolguları siler, çizgileri değiştirmez.

İçini Sil (Erase Inside) : Dolgulu bir alanın içini silmeye başlamanızı ve daha sonra herhangi bir çizgiyi etkilemeden çizgi içinde kalan dolgulu alanın içini silmenizi sağlar.

Musluk (Faucet): Bir çizgi ve dolgu üzerine herhangi bir yere tıklayarak bir çizgiyi veya dolguyu silmenizi sağlar. Bir çizgi veya dolguyu seçtikten sonra Delete tuşuna basmışsınız gibi çalışır (Şekil 2.44).



Şekil 2.43 Silgi Aracının Araç Panelindeki Musluk Özelliği

Silgi Şekli (Eraser shape) : Silginin büyüklük ve şeklini aşağıda verilen daire ve kare hâline getirir (Şekil 2.45).



Şekil 2.44 Silgi Aracının Araç Panelindeki Silgi Şekilleri



El aracı (Hand Tool),

Bu araç çalışma sayfanızı tutarak istediğiniz gibi hareket ettirmenizi sağlar. Yapılan çalışmada bazen çalışma alanı çok fazla büyütülür. Çok büyütülmüş alanlarda nesnelerin yeri bulmak ve çalışmak oldukça zorlaşır. El aracı kaydırma çubuğunda yatay ve dikey ayarları tek başına yaparak bu işlemleri kolaylaştırır.



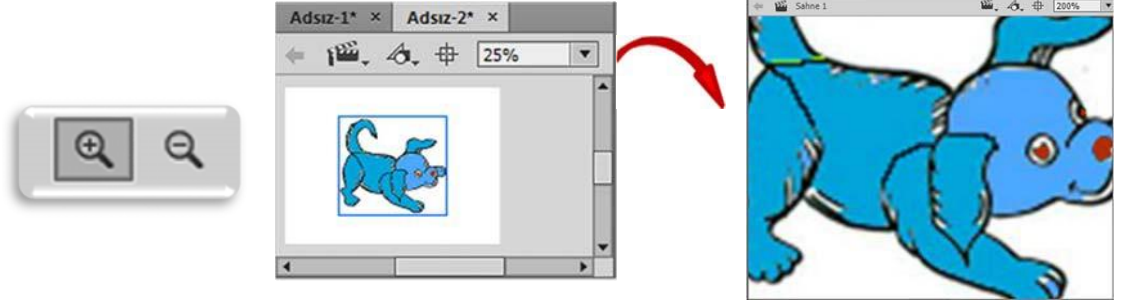
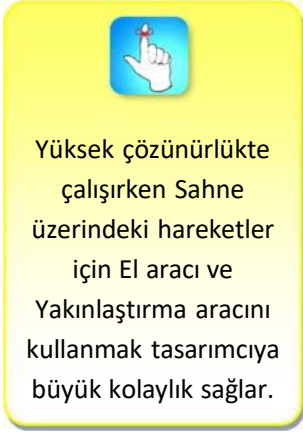
Yakınlaştırma aracı (Zoom Tool),

Çalışma sayfanızda çalıştığınız alanı yakınlaştırıp uzaklaştırmanızı sağlar.

Grafikler

Böylece detaylar üzerinde yakınlaştırarak daha kolay çalışabilirsiniz. Bu seçeneği seçmek için farenin sol tuşu ile büyüteç resmine tıkladığınızda Seçenekler (Options) bölümünde 2 seçenek çıkar.

Artı olan büyüteci seçerek çalışmamızı yakınlaştırabilir, eksi olan büyüteci seçerek çalışmamızı uzaklaştırabiliriz (Şekil 2.46-47).



Şekil 2.45 Yakınlaştırma Aracının Sahne Üzerindeki Etkisi



Dış çizgi rengi ayarlama (Stroke Color): Çizdiğiniz şeklin çizgi rengini bu bölümden değiştirebilirsiniz.



Dolgu rengi ayarlama (Fill Color): Fırça aracı ile çizilen çizgilerin rengini gösterir. Bu araç sayesinde şekillerin dolgu rengini değiştirebilirsiniz.



Siyah-Beyaz (Black end White): Bu da şekillerimizin rengini siyah & beyaz yapar.



Şekil 2.46 Yakınlaştırma Aracının Sahne Üzerindeki Etkisi



Özet

- Adobe Flash çalışma ortamı, kullanıcının hassas ve ölçekli çizimler gerçekleştirmesi için gerekli özellikleri sunar. Izgaralı çalışma alanı sayesinde sayfa içindeki nesnelerin yerleşimi ve konumları daha kolay belirlenir. Yatay ve dikey cetveller ile ölçekli nesneler çizebilirsiniz. Kılavuz çizgileri yardımıyla da çalışma ortamı içine yerleştirdiğiniz nesnelere referans noktaları atayabilirsiniz.
- Seçim aracı, çalışma alanındaki nesneler bu araç yardımıyla seçilerek, şekilleri düzenlenebilir ya da yeri değiştirilebilir. Alt Seçim aracıyla, seçilen bir çerçevenin içinde kalan alan da seçilir. Serbest dönüştürme aracı, nesneleri, metinleri, grup altına alınmış öğeleri ve Flashın örnek öğelerini istenildiği gibi döndürmek ve boyutlandırmak amacıyla kullanılır. 3B Döndürme aracı, nesneleri, kendi etrafında 3 boyutlu olarak döndürmek ve taşımak amacıyla kullanılır. Kement aracı, serbest seçim yapmak için kullanılır. • Kalem aracı: Bu araç serbest çizim sağlar. Nokta nokta tıklayıp eğip bükme suretiyle çizimler yapabilirsiniz. Metin aracı, resimler ne kadar bilgilendirici olursa olsun, yine de düğmeler için etiketler, reklam başlıkları için metin oluşturmak gerekmektedir. Çalışma dosyasına metin aracı kullanarak metin eklenebilir. Simge seçildiğinde özellikler penceresinde tüm metin biçimlendirme denetimleri görüntülenir. Metni yazmadan önce veya çalışma alanına yerleştirdikten sonra biçimlendirme değişiklikleri yapılabilir. Çizgi aracı, Vektör araçlarından biri olan çizgi aracı çalışma alanı içerisine çizgi eklemek için kullanılır. Dikdörtgen aracı, Vektör araçlarından biri olan dikdörtgen aracı araç kutusunda dikdörtgen çizmek için kullanılır. Kurşun kalem aracı, Bu araç kalem aracı ve çizgi aracı gibi çizgi çizmeyi sağlar. Bu aracın çizgi çizmesi için farenin sol tuşuna basılı tutulması gereklidir. Çizilen çizgi sadece dış çizgi rengindedir. Fırça aracı, bu araç aynı kurşun kalem aracı gibi çizgi çizmeyi sağlar. Bu aracın çizdiği çizgiler dolgu rengine sahiptir ve kurşun kalem kalındır. Çizginin kalınlığı, çizgi ve kurşun kalemin ayar seçenekleriyle ayarlanabilir.
- Kova boyası aracı: Grafik öğelerinin çerçeve renklerini değiştirmek için kullanılır. Çerçevesi olmayan grafik öğelerine de çerçeve ekler. Damlalık aracı ile bir bölgeye tıklandığında o bölgeyi kendi rengine dönüştürür. Bu seçim ile renk kopyalama yapabilirsiniz. Silgi aracı: Çizilen grafik öğelerini farklı biçimlerde silmek için kullanılır.
- El aracı: Yapılan çalışmada bazen çalışma alanı gereğinden çok fazla büyütülür. Çok büyütülmüş alanlarda nesnelerin yerini bulmak ve çalışmak oldukça zorlaşır. El aracı kaydırma çubuğunda yatay ve dikey ayarları tek başına yaparak bu işlemleri kolaylaştırır. Bir öğeyi yakınlaştırmak için Yakınlaştırma aracı kullanılır. Yakınlaştırma aracında yakınlaştırma ve uzaklaştırma arasında geçiş yapmak için Büyüt ve Küçült değiştiricilerini kullanın veya Alt tuşuna basıp tıklatın.

DEĞERLENDİRME SORULARI

1. Fare imleci yandaki şekli () ne zaman alır?
- a) Fare imleci bir grafik ögesinin keskin köşeli kısmına yaklaştırıldığı zaman
 - b) Fare imleci bir grafik ögesinin yumuşak eğimli bir köşesine yaklaştırıldığı zaman
 - c) Fare imleci grup altına alınmış bir nesnenin köşesine yaklaştırıldığında
 - d) Seçili bir grafik ögesi taşınacağı zaman
 - e) Seçili bir grafik ögesi döndürüldüğü zaman



2. Fare imleci yandaki şekli () ne zaman alır?
- a) Fare imleci bir grafik ögesinin keskin köşeli kısmına yaklaştırıldığı zaman
 - b) Fare imleci bir grafik ögesinin yumuşak eğimli bir köşesine yaklaştırıldığı zaman
 - c) Fare imleci grup altına alınmış bir nesnenin köşesine yaklaştırıldığında
 - d) Seçili bir grafik ögesi taşınacağı zaman
 - e) Seçili bir grafik ögesi döndürüldüğü zaman



3. Serbest dönüştürme () aracının işlevi nedir?
- a) Seçili grafik ögesini yalnızca döndürmek için kullanılır
 - b) Seçili grafik ögesinin boyutlarını ölçekli olarak değiştirmek için kullanılır
 - c) Üzerine tıklanan grafik ögesini seçer ve boyutlandırma ve döndürme işlemleri yapabilir
 - d) Grafik ögesinin şeklini değiştirmek için tutamaçlar oluşturur. Tutamaçlardan şeklin biçimi değiştirilir.
 - e) Grafik üzerinde renk düzenlemesi yapabilir



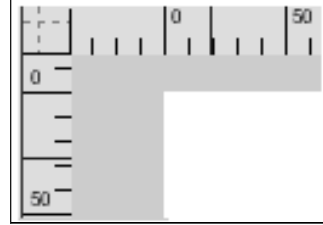
4. Yandaki resimde gösterilen düğme ne işe yaramaktadır?
- a) Üzerine tıklandığı grafik ögesinin dolgu rengini değiştirir.
 - b) Metin rengini değiştirmek için kullanılır.
 - c) Zemin rengini değiştirmek için kullanılır.
 - d) Üzerine tıklandığı grafik ögesinin çerçeve niteliklerine ayarlı değerleri atar.
 - e) Üzerine tıklandığı grafik ögesinin dolgu niteliklerine ayarlı değerleri atar.
5. İçerisinde ses, resim ve video barındıran görsele ne denir?
- a) Film
 - b) Slayt
 - c) Animasyon
 - d) Resim
 - e) Kitap

6. Hareketli görselleri oluşturmak için hangi dosya formatı seçilmelidir?

- a) JPEG
- b) WAW
- c) SWF
- d) PNG
- e) TIF

7. Yandaki şekilde verilen kenar çizgilerine ne denir?

- a) Kılavuz
- b) Seçim
- c) Kement
- d) Ölçekleme
- e) Cetvel



8. “Kalem Aracı” Araç çubuğu nesnesi aşağıdakilerden hangisinin görevini yapar?

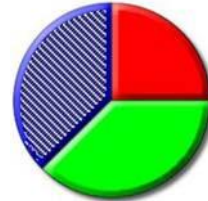
- a) Yazı ekler
- b) Çizgi çizer
- c) Düz /Kavisli çizgi çizer
- d) Çizgileri parçalara ayırır
- e) Kare şekli çizer

9. Yeni bir çalışma sayfası hangi seçenekteki kısayol ile sağlanır?

- a) Alt + Enter + X
- b) Ctrl + N
- c) Ctrl + V
- d) Ctrl + Shift + X
- e) Ctrl + Shift + C

10. Yandaki şekilde verilen kesikli çizgiler hangi işlem sonucu oluşur?

- a) Seçim
- b) Alt Seçim
- c) Sihirli değnek
- d) Serbest dönüştürme
- e) 3B Döndürme



YARARLANILAN KAYNAKLAR

[1]Adobe Flash Pro CC, Yardımcı el kitabı. <http://helpx.adobe.com/tr/support>

[Erişim tarihi: 30 Ekim 2013].

[2]SCHULZE Pati.(2004) Macromedia Fireworks 8 Kaynağından Eğitim, Medyasoft Yayınları.

[3]AKBUĞA, M. (2011). Grafik Animasyon- II Erzurum: Atatürk üniversitesi. ATAUZEM.

METİN İŞLEMLERİ



İÇİNDEKİLER

- Metin Oluşturmak
- Tasarım İçerisine Font (Harf) eklemek
- Metin Bulma ve Değiştirme
- Metin Özellikleri



HEDEFLER

- Bu üniteyi çalıştıktan
- Her ortamda kusursuz çalışan metinler oluşturabilecek,
- Metin içerisinde arama ve değiştirme işlemi yapabilecek,
- Metinlerin bütün özelliklerini kullanabilecek,
- Her çeşit metne filtre uygulayabileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

ANİMASYON TASARIMI

Okt.
Mustafa AKBUĞA

ÜNİTE 3

Giriş

Bu ünite de bir Flash tasarımına nasıl metin ekleneceği, eklenen metnin font, boyut, tür, renk ve satır aralığı, yatay veya dikey olarak döndürme özelliklerinin nasıl değiştirileceği hakkında detaylı bilgi verilecektir.

Ayrıca bu ünite de tasarımdaki metinlerin internet üzerinden başka bilgisayarlarda sorunsuz olarak görüntülenebilmesi için tasarımda kullanılan fontların çalışmanın içerisine nasıl yerleştirileceği (gömüleceği) anlatılacaktır.

METİN OLUŞTURMAK

Metinlerinizi doğrudan Flash'ta oluşturabilir veya başka bir dosyadan aktarabilirsiniz. Metnin yönünü yatay veya dikey olarak ayarlayabilirsiniz. Metinleri: metin boyu, yazı stili, paragraf hizalaması ve satır boşluğu gibi özellikleri ayarlayarak biçimlendirebilirsiniz. Bir metin alanı döndürülebilir, yeniden boyutlandırılabilir veya bir sembole dönüştürülebilir. Çalışma içerisine dâhil edilen Metinlere Alt gölge ve Işıma gibi filtreler uygulanabilir.

Metin aracıyla 3 tür metin oluşturulabilir.

- **Dinamik Metin:** Dinamik olarak güncellenen metin görüntüleyen bir alan oluşturur.
- **Giriş Metin:** Kullanıcıların metin girebileceği bir alan oluşturur.
- **Statik Metin:** Dinamik olarak kendini güncelleyemeyen bir alan oluşturur.

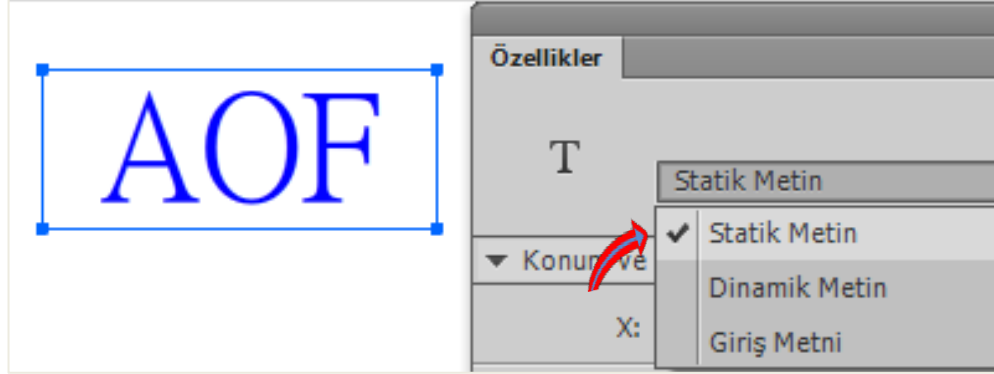
Çalışma alanı içerisine Statik bir Metin eklemek;



- Araç panelinden, Metin aracını () seçin.
- Metin aracı türlerinde Statik Metin türünü seçin. Eğer Dinamik metin'i seçerseniz, metin filmde görünmeyecektir. Giriş Metin türü metnin etrafını belirginleştirir ve çalışma anında dışarıdan bilgi girişine izin verir.
- Sahne üzerinde metin yazmak istediğiniz yere tıklayınız.
- Oluşan metin alanı içerisine "AOF" yazın (Şekil 3.1).
- Özellikler panelinden istediğiniz özelliklerini ayarlayınız.



Çalışma alanına eklenen Dinamik metin türü swf dosyası gösterilirken ActionScript dili yardımıyla değiştirilebilir.



Şekil 3.1 Çalışma alanı içerisine metin ekleme işlemi



Yayınlanmış SWF dosyalarına Font karakterlerini dâhil ederek, font'un dosyayı oynatan bilgisayardan bağımsız hâle getirilmesi sağlanabilir.

TASARIM İÇERİSİNE FONT EKLEME

Tasarım esnasında kullanılan özel fontların tasarım içerisine gömülmediği durumlarda ilgili çalışma içerisindeki metin nesnesi, başka bilgisayarlarda eksik ya da yanlış görülebilmektedir. Örnek olarak, tasarımı yapılan bilgisayarda Şekil 3.2 deki gibi doğru bir şekilde oluşturulan metin nesnesi, Font gömme işlemi yapmayınca ilgili fontun ya da fontların olmadığı başka bir bilgisayar da Şekil 3.3'teki gibi eksik bir şekilde görüntülenir. Bu tür olumsuzlukların yaşanmaması için, tasarım esnasında kullanılan özel fontlar, mutlaka tasarıma dâhil edilmelidir ve bu sayede ilgili metin nesnesinin doğru görüntüleneceği garanti altına alınmış olur.



Şekil 3.2 Açıköğretim yazısının tasarımı yapılan bilgisayarda açıldığında oluşan görünümü



Şekil 3.3 Açıköğretim yazısının 'ç, ö, ğ, i' fontlarının olmadığı farklı bir bilgisayarda açıldığında oluşan görünümü.

Font bir tasarım içerisine bir kez gömülünce, onu SWF dosyanızın içindeki herhangi bir yerde kullanabilirsiniz.

SWF dosyasına font gömerek doğru metin görünümü sağlamanın gerekli olduğu dört özel durumu vardır:



Eski bir FLA dosyasını açtığınızda Flash Professional CC eskiden gömülmüş fontları düzenlemenize izin verir.

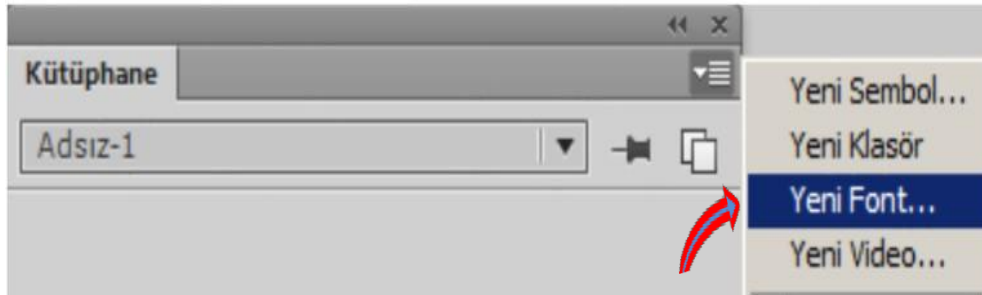
- FLA dosyanızda tutarlı metin görünümü gerektiren durumlarda tasarımın parçası olan metin nesneleri oluştururken.
- Aygıt Fontlarını Kullan dışında bir kenar yumuşatma seçeneği kullanırken fontları gömmeniz gerekir. Aksi takdirde, metin yok olabilir veya yanlış görüntülenebilir.
- FLA dosyanızda ActionScript ile dinamik olarak metin oluştururken. ActionScript'te hangi fontu kullanacağınızı belirtmeniz gerekir.
- SWF dosyanız metin nesneleri içerip; gerekli olan fontların gömülü olmadığı başka bir SWF dosyası tarafından yükleneceği zaman.

Bir çalışma içerisine Font eklemek için aşağıdaki maddelerin sırayla uygulanması gerekir.

- Metin menüsünden Font Gömme'yi seçin (Şekil 3.4) veya Kütüphane panel seçenekleri menüsünden Yeni Font seçeneğini kullanın (Şekil 3.5).



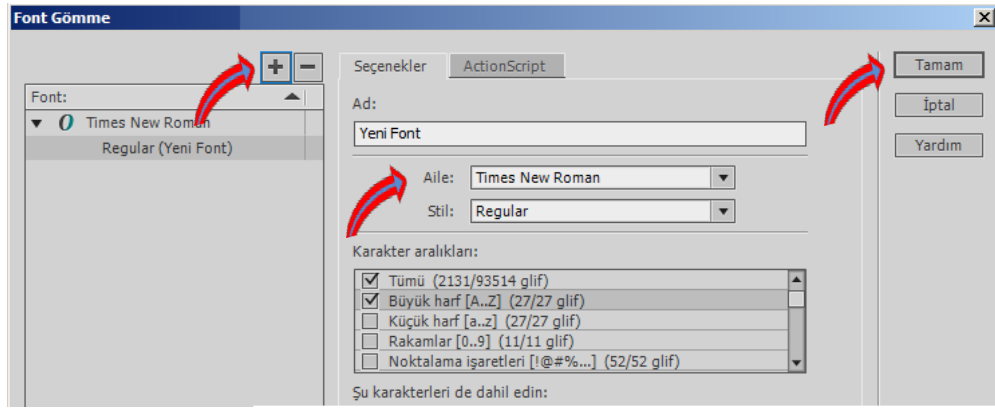
Şekil 3.4 Adobe Flash' ile Metin menüsünden çalışma dosyasına font ekleme



Şekil 3.5 Adobe Flash ile kütüphane seçeneklerinden font ekleme

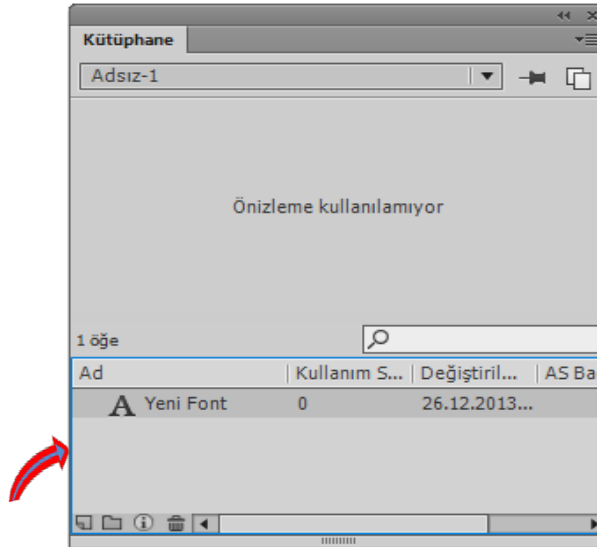
Bir çalışma içerisine ne kadar karakter gömerseniz yayınlanmış SWF dosyanızın boyutu o kadar artacaktır.

- Fontunuz Font Gömme iletişim kutusunda seçili değilse, yeni bir gömülü fontu FLA dosyanıza eklemek için Ekle (+) düğmesini tıklayın. (Şekil 3.6). Font Gömme iletişim kutusunu Kütüphane'den veya Metin Özellik Denetçisinden açtığınızda iletişim kutusunda otomatik olarak bir font ögesi belirir. Seçenekler sekmesinde gömmek istediğiniz fontun Aile ve Stilini seçin. Eğer Font Gömme iletişim kutusunu Metin Özellik Denetçisinden veya Kütüphane panelinden açtıysanız, geçerli seçim tarafından kullanılan font otomatik olarak iletişim kutusunda belirir (Şekil 3.6).
- Karakter Aralığı bölümünde gömmek istediğiniz karakter aralığını seçin. Herhangi bir karakteri ek olarak gömmek isterseniz, bunları "Bu karakterleri de ekle" alanına girin.



Şekil 3.6 Font Gömme iletişim ekranı ile font ekleme işlemi

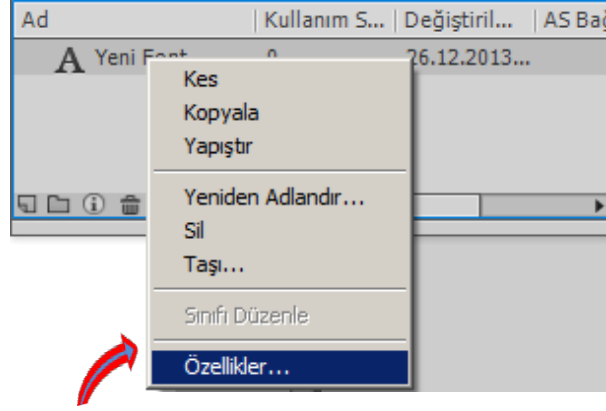
- Kütüphane panelinden çalışma içersine eklemiş olduğunuz Fontları görebilirsiniz (Şekil 3.7).



Şekil 3.7 SWF çalışması içersine alınacak fontun kütüphane panelinde görüntülenmesi

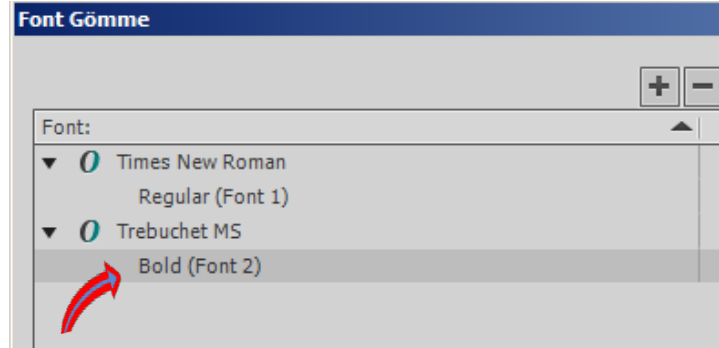
Daha önce gömmüş olduğunuz bir fontun parametrelerini düzenlemek için ise aşağıdaki işlem basamaklarını takip etmelisiniz.

- Kütüphane'deki font sembolünü sağ tıklayın ve Özellikler'i seçin (Şekil 3.8) veya Kütüphane'deki font sembolünün simgesini çift tıklayın.



Şekil 3.8 SWF çalışması içerisine gömülmüş fontun özelliklerinin değiştirilmesi

- Açılan Font gömme iletişim kutusunun solundaki ağaç görünümünde düzenlemek istediğiniz font sembolünü seçin (Şekil 3.9).



Şekil 3.9 SWF çalışması içerisine gömülmüş fontlardan birinin seçimi

- Font Gömme iletişim kutusundaki seçenekler içerisinden gerekli değişiklikleri yapın ve Tamam'ı tıklayın (Şekil 3.6).

METİN BULMA VE DEĞİŞTİRME

Adobe Flash kullanarak kısa süreli sınavlar veya okuma metinleri oluşturulabilir. Çalışma içerisinde çok miktarda metin var ve bunların aranarak bulunması ve yenileriyle değiştirilmesi gerekiyorsa Düzen Menüsünde Bul ve Değiştir komutunun verilmesi gerekir. Bu işlemin yapılması için:

- Düzen > Bul ve Değiştir'i seçin (Şekil 3.10).

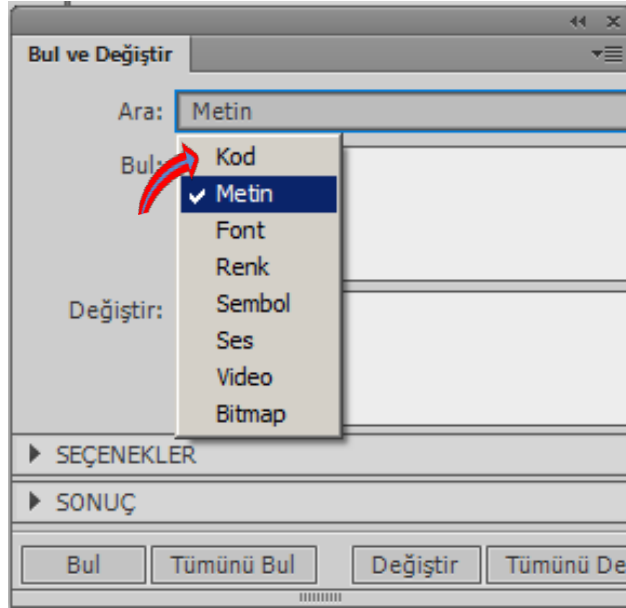


Birden fazla katman ve Sahnenin olduğu çalışmalarda Bul ve Değiştir komutu aradığınızı daha kolay bulmanızı sağlayabilir.



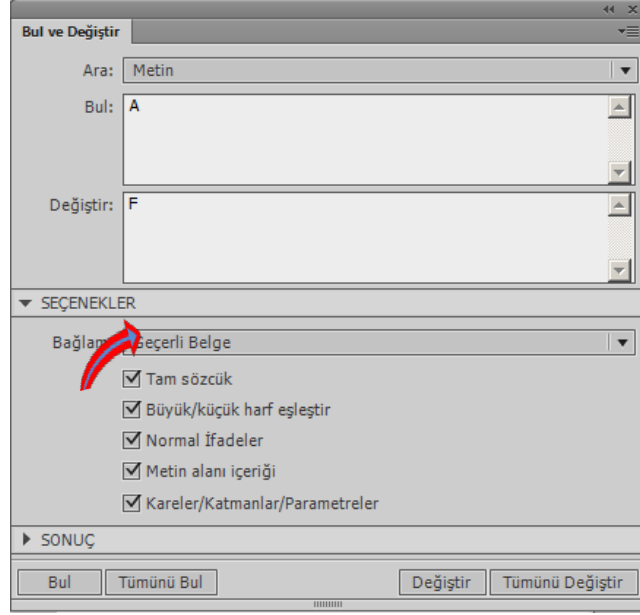
Şekil 3.10 Düzenle menüsünden Bul ve Değiştir komutunun verilmesi

- Düzenle> Bul ve Değiştir > Ara menüsündeki arama türlerinden Metin'i seçin (Şekil 3.11). Bul metin kutusuna, bulunacak metni girin. Değiştir metin kutusuna, varolan metinle değiştirilecek metni girin.



Şekil 3.11 Bul ve Değiştir penceresinde bulunan arama seçenekleri

- Metin arama için seçenekleri seçin. Değiştir ya da Tümünü Değiştir komutunu verilerek işlemi yapıldını görün (Şekil 3.12).



Şekil 3.12 Bul ve Değiştir penceresinde bulunan arama seçenekleri

Bul ve Değiştir penceresinde aşağıdaki seçenekler bulunur.

- **Tam sözcük:** Belirtilen metin dizesini sadece, iki taraftan da boşluklar, tırnaklar veya benzer işaretlerle sınırlanmış bir tam sözcük olarak arar.
- **Tam sözcük seçili değilken:** Yer için bir arama bariyer, yerleştirme gibi kelimeleri ortaya çıkarır.
- **Büyük/küçük harf eşleştir bulurken ve değiştirirken:** Belirtilen metnin büyük/küçük harf biçimine tam olarak uyan metinleri arar.
- **Normal ifadeler:** ActionScript içerisinde normal ifadeler kullanarak metin arar.
- **Metin Alanı İçeriği:** Bir metin alanının içeriğini arar.
- **Kareler/Katmanlar/Parametreler Kare etiketleri:** Katman adları, sahne adları ve bileşen parametrelerini arar.

Tasarım içerisinde metin bulmak için iki seçenekten biri seçilir.

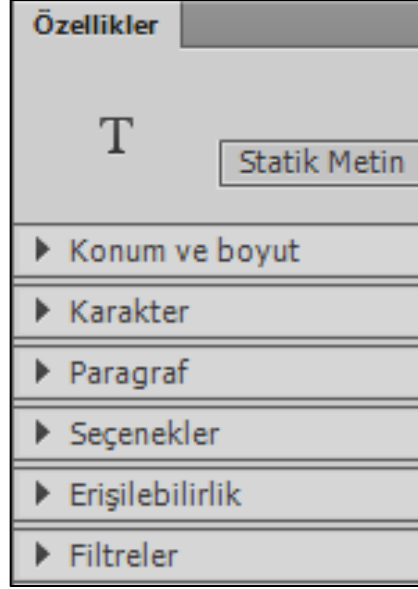
1. Belirtilen metnin bir sonraki tekrarlamasını bulmak için Sonrakini Bul'u tıklatın.
2. Belirtilen metnin tüm tekrarlamalarını bulmak için Tümünü Bul'u tıklatın
Tasarım içerisinde metni değiştirmek için iki seçenekten biri seçilir.
3. Belirtilen metnin bulunan sonuçlarından sadece birini değiştirmek için Değiştir'i seçin.
4. Belirtilen metnin tüm tekrarlamalarını değiştirmek için Tümünü Değiştir'i seçin.



Metin eklendiğinde özellikler paneline dikkat edilirse Zarfın Yükseklik değeri pasif durumda olduğu görülür.

METİN ÖZELLİKLERİ

Bir metnin Şekil 3.14'te gösterilen ve aşağıda ayrıntılı olarak anlatılan özellikleri ayarlanabilir.



Şekil 3.13 Metine ait özellik seçenekleri

Konum ve boyut bölümü: Şekil 3.15'te görülen metin özelliklerini barındırır:

- **X**, Metnin çalışma alanı içerisindeki X eksenindeki konumu gösterir.
- **Y**, Metnin çalışma alanı içerisindeki Y eksenindeki konumu gösterir.
- **G**, Metin zarfının genişliğini gösterir. Metin büyüklüğü değişse de zarfın değeri değişmez.

Y, Metin zarfının yükseklik değerini gösterir. Metin zarfının yüksekliği için kullanıcı sayısal bir değer giremez. Kullanıcı Metin girerken Enter tuşuna bastıkça ya da zarfa yeni metin karakteri girildikçe bu değer ancak Flash tarafından değiştirilir (Şekil 3.15).



Şekil 3.14 Metine ait özellikler paneli

- **Aile:** Çalışma alanında yazılan metne ait fontun adı.
- **Karakter bölümü:** Şekil 3.16'da görüntülenen metin özelliklerini barındırır.
- **Stil:** Normal, Kalın veya İtalik. Sahte italik ve sahte kalın stilleri TLF metin nesnelerinde bulunmaz. Bazı fontlar Siyah ve Kalın İtalik gibi ek stiller de içerebilir.



Adobe Flash programı içerisinde kullanılan fontların değişimi Aile özelliği kullanılarak yapılır. Genellikle stil özelliği ile bu özellik karıştırılmaktadır.

- **Boyut:** Karakterlerin piksel olarak boyutlarını belirler.
- **Harf Aralığı:** Harflerin arasındaki yaslamaların dağıtımını yapar.
- **Renk:** Metnin renginin seçimini belirler.
- **İzleme:** Seçili karakterlerin aralarındaki boşluk miktarını ayarlar.
- **Vurgu:** Vurgu rengini ayarlar.
- **Karakter Aralığı:** Belirli karakter eşlerinin aralarındaki boşluğu çıkartır veya aralarına boşluk ekler. TLF metni, çoğu fontun içine yerleştirilmiş karakter aralığı bilgisini kullanarak karakter aralığını otomatik olarak belirler.

Otomatik: Latin karakterleri fontu içine yerleştirilmiş karakter aralığı bilgisini kullanır. Asya karakterleri için, karakter aralığını yalnızca içine karakter aralığı bilgisi yerleştirilmiş karakterlere uygular. Karakter aralığı bilgisi bulunmayan Asya karakterleri Kanji, Hiragana ve Katakana dilleridir.

Açık: Karakter aralığını her zaman açar.

Kapalı: Karakter aralığını her zaman kapatır.

- **Kenar Yumuşatma** Üç farklı kenar yumuşatma modu arasından seçim yapılabilir (Şekil 3.16).

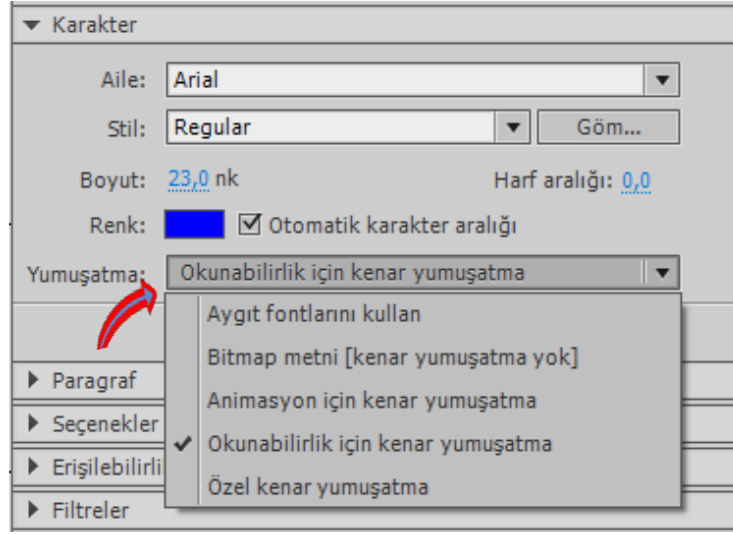
Aygıt Fontlarını Kullan: Fontları görüntülemek için SWF dosyasının yerel bilgisayarda yüklü bulunan fontları kullanmasını belirtir. Tipik olarak, aygıt fontları çoğu font boyutunda okunaklıdır. Bu seçenek SWF dosyasının boyutunu artırmaz. Ancak sizi font görüntüleme amacıyla kullanıcının bilgisayarında yüklü bulunan fontlara bağlı kalmaya zorlar. Aygıt fontlarını kullanırken, yalnızca genel olarak yüklü bulunan font ailelerini seçin.

Okunabilirlik: Özellikle küçük boyutlardaki fontların daha rahat okunabilmesine olanak verir. Bir metin bloğu için bu seçeneği kullanmak üzere, metin nesnesi tarafından kullanılan fontu gömün.

Animasyon: Hizalama ve karakter aralığı bilgisini yok sayarak daha yumuşak bir animasyon oluşturur. Bir metin bloğu için bu seçeneği kullanmak üzere, metin bloğu tarafından kullanılan fontu gömün.



Yumuşatma özelliği içerisinde Aygıt Fontlarını Kullan seçeneği kullanılırsa fontların düzgün çalışması için dosyaya gömülmesine gerek kalmaz.



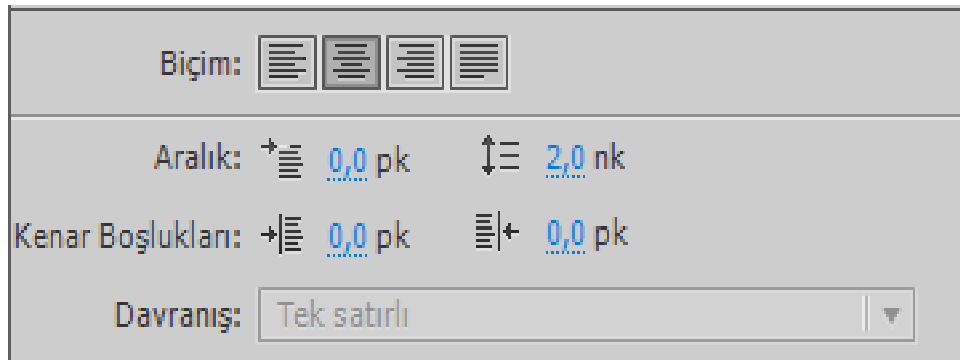
Şekil 3.15 Metine ait karakter bölümündeki seçenekler.

Paragraf bölümü: Şekil 3.17’de görülen metin özelliklerini barındırır.

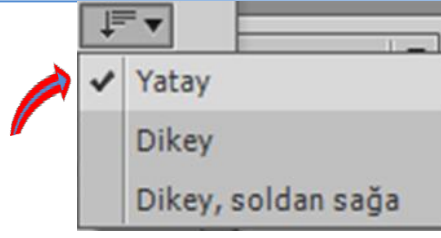
- **Biçim:** Bu özellik yatay veya dikey metinle kullanılabilir. Başlangıca Hizala, metni kabın başlangıç kenarı boyunca hizalar (soldan sağa metin için sol taraf). Bitişe Hizala, metni kabın bitiş kenarı boyunca hizalar (soldan sağa metin için sağ taraf). Geçerli seçimin paragraf yönü sağdan sola olduğu zaman hizalama simgelerinin görünümü doğru yönü gösterecek şekilde ters tarafa döner.
- **Aralık:** Seçili paragraftaki ilk sözcük girintisini piksel olarak belirtir.
- **Satır Aralığı:** Metnin satır aralarındaki dikey boşluk miktarını ayarlar. Satır aralığının yüzde olarak ifade edildiği varsayılır, ancak bunun yanı sıra nokta olarak da ifade edilebilir.
- **Kenar Boşlukları:** Metnin zarf içerisindeki sağ ve sol kenar boşluklarının genişliğini piksel olarak belirler. Varsayılan değer 0’dır.



Araç kutusundan Metin aracı seçildiğinde özellikler panelinden yön ayarı da yapılarak metin zarfı oluşturulmalıdır. Sonradan değişen yön, zarfın metne küçük gelmesine neden olur.



Şekil 3.16 Metine ait Paragraf bölümündeki seçenekler



Şekil 3.17 Metin yönlendirilmesi ve seçenekleri

Yön: Paragraf yönünü belirtir. Yön ayarları ancak Tercihlerdeki Dikey

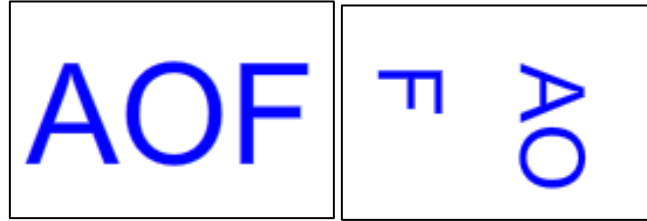


seçenekleri açıksa kullanılabilir (Şekil 3.18). Karekter özelliklerindeki Sembolü metni döndürme işlemini gerçekleştirir. Bu ayar yalnızca metin kabının içindeki, seçili durumda olan paragraf için geçerlidir. Metin yönü değiştirildiğinde Şekil 3.19- 20’de görüldüğü gibi metin zarfında ortaya çıkan taşma, zarfın genişletilmesiyle düzeltilebilir.

Yön şu değerleri alabilir:

Yatay: Normal bir metin yazma işleminde kullanılır. Biçim özelliklerinde seçilen değere göre yaslama işlemi yapılır.

Dikey: Sağdan sola, yazılan metni dikey duruma getirir. Eğer metin zarfında yer olmaz ise harfler dağılır. Bu durumda zarfın mouse ile genişletilmesi gereklidir. Dikey, soldan sağa, yazılan metnin dikey olarak soldan sağa doğru sıralanması gereklidir.



Şekil 3.18 Yatay yazılmış bir metnin yönü Dikey olarak seçildiğinde oluşan etki



Şekil 3.19 Dikey konuma alınan metnin zarfa sığmaması sonucu dağılan harflerin Zarf genişletilerek düzeltilmesi



Seçenekler bölümünde yer alan bağ özelliğine yazılacak adrese “http” protokolü mutlaka eklenmelidir.

- Paragraf bölümünde yer alan Davranış seçeneği, Statik metin işlemlerinde kullanılmaz. Bu bölüm için Giriş metni ya da Dinamik metin seçeneklerinden birinin seçili olması gereklidir.

Seçenekler bölümü: Şekil 3.21’de görülen metin özelliklerini barındırır.

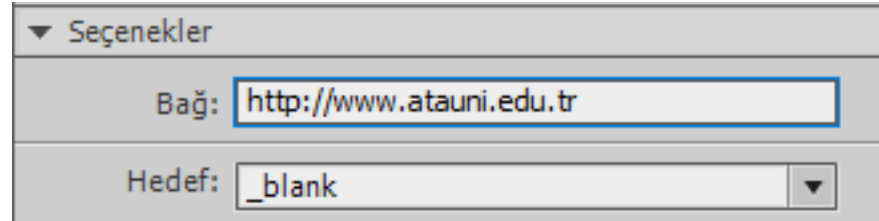
- Bağ:** Bu bölüme önceden hazırlanmış swf videolarının ya da sayfalarının bağlantı adresleri yazılmalıdır.
- Hedef:** Tıklama sonrası açılacak sayfanın açılma şeklini belirler. Bu seçeneğin alt seçenekleri:

blank: Bağlantıyı yeni bir pencerede açar.

self: Bağlantıyı aynı pencerede açar.

top: Bağlantıyı aynı pencere en üstten itibaren açar.

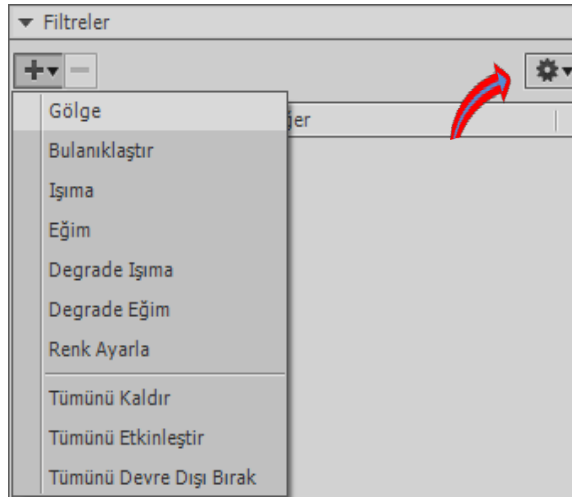
parent: Bağlantı, o anda açık olan internet sayfası yerine Bağ seçeneğine girilen adresteki sayfa açılır.



Şekil 3.20 Metne ait Seçenekler bölümü

Erişebilirlik bölümü, Bu bölümde yer alan özellikler Statik metin türünde açılmaz. Bu bölüm özelliklerinin açılması için Giriş metni ya da Dinamik metin seçeneklerinden birinin seçili olması gereklidir. Eylem kodları ve Temel programlama yapılarında ele alınacaktır.

Filtreler bölümü: Şekil 3.22’de görülen metin özelliklerini barındırır.



Şekil 3.21 Metne ait Filtreler bölümü



Seçilmiş metin üzerine Filtre uygulandığında sonuç; çalışma ekranında canlı olarak görüntülenir, böylece değerler üzerinde kolayca oynayabilirsiniz.

Metin aracı kullanılarak yazılmış yazıları seçerek, bu bölümden Filtreler ekleyebilirsiniz. Metin nesnesinde, metnin tamamını üzerinde değilde, metinde bulunan her bir harfin üzerinde teker teker işlem yapmak istendiğinde, metinleri grafik nesnesine çevirerek her bir harfi birbirinden bağımsız hâle getirmek gerekir.

Bu işlemler için;



Araç panelinden, Metin aracını () seçin. Çalışma alanına Şekil 3.23'teki gibi "BİLGİSAYAR" yazınız.

- Metni seçiniz ve Değiştir menüsünden parçalan komutunu iki kere uygulayınız.
- Bir bütün olan metin harflerine ayrılır ve her bir harf birbirinden bağımsız hâle gelir. Bu durum Şekil 3.24 de gösterilmiştir.
- Parçala komutuyla parçalara ayırdığınız yazı, metin özelliğini kaybetmiş olduğundan her bir harf üzerinde istediğiniz grafik işlemini yapabilirsiniz (Şekil 3.25).



Seçili bir metin için
Parçala komutu
verildiğinde her bir harf
kendi metin zarfına
kavuşur. Bağımsız her
bir harf yeni birer metin
özeliliği gösterir.



Şekil 3.22 Normal yazılmış Metin görünümü



Şekil 3.23 Her bir harfi birbirinden bağımsız hâle getirilmiş grafikler harfleri



Şekil 3.24 Harflerin kenar noktalarından tutularak grafik işlemlerinin yapılması



Seçilmiş bir Metin zarfına ya da Grafiğe çevrilmiş Metin harfine, Filtre ekleme için (+) Sembolü, eklenmiş filtreyi silmek içinse (-) sembolü kullanılır

. Bu sembollerin konumları Şekil 3.22’de sağ taraftaki okla gösterilmiştir.



Bu seçenek metine eklenmiş filtreleri kopyalayarak başka metinlere aktarma seçeneklerini barındırır. Bu sembolün konumu Şekil 3.22’de sol taraftaki okla gösterilmiştir. Aşağıdaki seçeneklere sahiptir.

Seçilen Filtreleri Kopyala: Tek bir filtreyi seçerek bütün ayarlarıyla kopyalar.

Tüm Filtreleri Kopyala: Metne uygulanmış bütün filtreleri seçerek ayarlarıyla kopyalar

Filtreleri Yapıştır: Önceki seçeneklerle kopyalanan filtreleri yeni metinde uygular

Filtreleri Sıfırla: Metindeki eklenmiş bütün filtre etkilerini sıfırlar

Ön ayarak olarak kaydet: Yapılan filtre ayarlarının bir isim altında kaydedilmesini sağlar.

Şekil 3.22’de gösterilen filtrelerin özellikleri ve etkilerini aşağıdaki örnekler ile inceleyelim.

Gölge filteri: Yüzeye gölge bırakan bir nesnenin görünüşü etkisi yapar. Şekil 3.23’teki metine bu filtre uygulandığında Şekil 3.26’daki görünüm elde edilir.



Seçili bir Metin için iki kez parçala komutu verildiğinde her bir harf grafik nesnesine dönüşür. Metin özelliklerini kaybeder. Bu grafik nesnelere Filtre uygulanamaz.



Şekil 3.25 Metnin üzerine Gölge filteri uygulanmış hâlinin görünümü



Şekil 3.26 Metnin üzerine Bulanıklaştır filteri uygulanmış hâlinin görünümü

Bulanıklaştır filteri: Bir nesneye bulanıklaştırma uygulamak, onun sanki başka nesnelerin arkasındaymış gibi görünmesini sağlar. Şekil 3.23’teki metne bu filtre uygulandığında Şekil 3.27’deki görünüm elde edilir.



Şekil 3.27.Metnin üzerine Işıma filtresi uygulanmış hâlinin görünümü

Işıma filtresi: Bir nesnenin kenarlarının etrafına bir renk uygulamanızı sağlar. İç, Dış, Tam olmak üzere üç türü vardır. Bu türlerin etkisi Şekil 3.23'teki metne bu filtre uygulanmasıyla Şekil 3.28'deki gibi görülmektedir.



Şekil 3.28 Metnin üzerine Eğim filtresi uygulanmış hâlinin görünümü

Eğim filtresi, Eğim uygulama, nesneye arka plan yüzeyi üzerinde eğimli görünmesini sağlayan bir vurgu uygular. Şekil 3.23'teki metne bu filtre uygulandığında Şekil 3.29'daki etki elde edilir.



Şekil 3.29 Metnin üzerine Degradé ışıma filtresinin iç türü uygulanmış hâli



Şekil 3.30 Metnin üzerine Degradé ışıma filtresinin Dış türü uygulanmış hâli



Şekil 3.31 Metnin üzerine Degradé ışıma filtresinin Tam türü uygulanmış hâli



Tasarım içerisinde, Metinler üzerindeki yapılan değişiklikleri görmek için, Metinler önce Film klibi sembolüne çevrilmelidir. Sonrasında editör içerisindeyken filtre seçenekleri uygulanmalıdır.

Degrade Işıma: Degrade Işıma filtresi uygulanmış bir yazının kenarları etrafına seçilen bir rengin farklı tonlarından oluşan degrade rengi uygulamanızı sağlar. Degrade rengi; Bir rengin bir başka renge dönüşmesi esnasında oluşan renk geçişine denir. Örneğin Siyah bir rengin Beyaz bir renge dönüşmesi arasındaki oluşan farklı tonlara degrade rengi denir. Şekil 3.23'teki metne bu filtre uygulandığında Şekil 3.30-31-32'deki etkiler görülmektedir.



Şekil 3.33 Metnin üzerine Degrad Eğim filtresi uygulanmış hâlinin görünümü

Degrad Eğim: Degrad Eğim filtresi uygulanmış bir yazının arka planına bir eğim etkisi verilerek degrade renginin nesneyi kabartılmış gibi görünmesini sağlar.

Degrad Işıma filtresinin özelliklerinde yer alan türler bu efektde yer alır. Şekil 3.23 'teki metine bu filtre uygulandığında Şekil 3.33'teki etki elde edilir.

Renk Ayarla filtresini uygulama:

Renk Ayarla filtresi, seçilen nesnenin Kontrast, parlaklık, doygunluk ve ton gibi renk niteliklerini kontrol etmenize olanak sağlar. Bu değerler 0 ile 100 arasında değiştirilebilir (Şekil 3.34-35-36).



Şekil 3.34 Metnin üzerine Renk filtresi değerleri sıfır olarak girilmiş hâli



Şekil 3.35 Metnin üzerine Renk filtresi değerleri değiştirilerek etkisinin görünümü



Şekil 3.36 Metnin üzerine Renk Ayarla olarak Tondeğeri 100 olarak girilmiş hâlinin görünümü



Özet

- Metinlerinizi doğrudan Flash'ta oluşturabilir veya başka bir dosyadan aktarabilirsiniz.
- Tasarımı tamamlanmış SWF dosyaları içerisindeki fontlar başka bilgisayarlar içerisinde bulunmayabilir, çalıştıran bilgisayar içerisinde metinlerin sorunsuz görüntülenmesi amacıyla fontları çalışma içerisine dâhil etmek gerekir. Bu işleme font gömme adı denir.
- Seçilen metnin yönünü yatay veya dikey olarak ayarlayabilirsiniz.
- Metinleri; metin boyu, yazı stili, paragraf hizalaması ve satır boşluğu gibi özellikleri ayarlayarak biçimlendirebilirsiniz. Bir metin alanı döndürülebilir, yeniden boyutlandırılabilir veya bir sembole dönüştürülebilir.
- Dinamik Metin, dinamik olarak güncellenen metin görüntüleyen bir alan oluşturur.
- Giriş Metin, kullanıcıların metin girebileceği bir alan oluşturur.
- Statik Metin, dinamik olarak kendini güncelleyemeyen bir alan oluşturur.
- Metinlerinize animasyon ve alt gölge veya ışıma gibi bir filtre uygulayabilirsiniz.
- Gölge filtresi, yüzeye gölge bırakan bir nesnenin görünüşünü simüle eder.
- Bulanıklaştır filtresi, nesnelerin kenarlarını ve ayrıntılarını yumuşatır.
- Bir nesneye bulanıklaştırma uygulamak, onun sanki başka nesnelerin arkasındaymış gibi görünmesini sağlayabilir veya onu hareketliymiş gibi gösterebilir.
- Işıma filtresi, bir nesnenin kenarlarının etrafına bir renk uygulamanızı sağlar.
- Eğim filtresi, nesneye arka plan yüzeyi üzerinde eğimli görünmesini sağlayan bir vurgu uygular.
- Degrade Işıma uygulamak, ışıma yüzeyi boyunca bir Degrade rengi bulunan bir ışıma görünüşü üretir.
- Dinamik metinler ve Giriş metinleri ilerleyen ünitelerde incelenecek olup bu bölümde ele alınmamıştır.
- Seçili metinlerin Renk Ayarla filtresi ile seçilen nesnenin kontrast, parlaklık, doygunluk ve ton gibi renk seçeneklerini değiştirebilirsiniz.

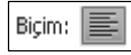
DEĞERLENDİRME SORULARI

1. Adobe Flash ile yapılan tasarımlar, tasarımın yapıldığı bilgisayarda normal olarak görüntülenirken, aynı tasarım internet üzerinden farklı bilgisayarlar üzerinde görüntülendiğinde font ve karakter sorunları ortaya çıkarmaktadır. Seçeneklerden hangisinin yapılması bu sorunları ortadan kaldırır?
 - a) Tasarımlarda sembollerin kullanılması
 - b) Tasarımda kullanılan metinlerin renginin siyah seçilmesi
 - c) Çalışma içerisinde nesnelere uygulanan filtrelerin kaldırılması
 - d) Çalışma içerisinde ActionScript dilinden yararlanılması
 - e) Tasarımda kullanılan Fontların çalışma içerisine gömülmesi

2.  simgesinin işlevi nedir?
 - a) Metin boyutunu ayarlar
 - b) Metnin nasıl konumlanacağını ayarlar
 - c) Metnin harfleri arasındaki mesafeyi ayarlar
 - d) Metnin ekranda nasıl görüneceğini değiştirir
 - e) Metinler arasındaki yüksekliği ayarlar

3.  simgesinin işlevi nedir?
 - a) Metni hizalar (sağa, sola, ortaya, vb.)
 - b) Metnin harfleri arasındaki mesafeyi ayarlar
 - c) Yazı türünü değiştirir
 - d) Metin türü belirlenir (Static Text, Dynamic Text, vb.)
 - e) Kullanılan harflerin büyüklüğünü ayarlar

4. Metin kutusunda karakterlerin font ayarları nereden ayarlanır?
 - a) Aile
 - b) Still
 - c) Boyut
 - d) Renk
 - e) Yumuşatma

5.  simgesinin işlevi nedir?
 - a) Metni hizalar (sağa, sola, ortaya, vb.)
 - b) Metnin harfleri arasındaki mesafeyi ayarlar
 - c) Yazı türünü değiştirir
 - d) Metin türü belirlenir
 - e) Kullanılan harflerin büyüklüğünü ayarlar

6. Nesnelerin kenarlarını ve ayrıntılarını yumuşatır. Seçili metin nesnesine bu filtre uygulamak, onun sanki başka nesnelerin arkasındaymış gibi görünmesini sağlayabilir. Aşağıdaki Filtrelerden hangisi bu etkiyi sağlar?

- a) Eğim
- b) Kabartma
- c) Bulanıklaştır
- d) Keskinleştir
- e) Renkli dolgu

7. “Bir seçili metin nesnesinin etrafına istenen bir renk uygulamanızı sağlar.” Aşağıdaki Filtrelerden hangisi bu etkiyi sağlar?

- a) Eğim
- b) Kabartma
- c) Bulanıklaştır
- d) Işıma
- e) Renk ayarla



8. Yandaki metne hangi filtre uygulanmıştır?

- a) Eğim
- b) Gölge
- c) Degrade Işıma
- d) Işıma
- e) Renk ayarla



9. Yandaki metne hangi filtre uygulanmıştır?

- a) Bulanıklaştır
- b) Gölge
- c) Degrade Işıma
- d) Işıma
- e) Renk ayarla



10. Yandaki metne hangi filtre uygulanmıştır?

- a) Bulanıklaştır
- b) Gölge
- c) Degrade Eğim
- d) Işıma
- e) Renk ayarla

Cevap Anahtarı

1.b, 2.c, 3.e, 4.c, 5.a, 6.c, 7.d, 8.c, 9.a, 10.c

YARARLANILAN KAYNAKLAR

[1]Adobe Flash Pro CC, Yardımcı el kitabı. <http://helpx.adobe.com/tr/support>

[Erişim tarihi: 30 Ekim 2013].

[2]SCHULZE Pati.(2004) Macromedia Fireworks 8 Kaynağından Eğitim, Medyasoft Yayınları.

SEMBOLLER



Atatürk Üniversitesi
Açıköğretim Fakültesi



İÇİNDEKİLER

- Sembolleri Oluşturmak
- Kütüphane Paneli
- Grafik Sembolü
- Düğme Sembolü
- Film Klibi Sembolü



HEDEFLER

- Bu üniteyi çalıştıktan sonra;
- Sembol oluşturabilecek ,
- Semboleri düzenleyebilecek,
- Sembolleri Kütüphane Paneli yardımıyla organize ederek,
- Tasarımlarda her çeşit sembolü uygulayabileceksiniz.

ANİMASYON TASARIMI

Okt.
Mustafa AKBUĞA

ÜNİTE 4



Tek nesnenin çok yerde kullanılması kavramı, sembollerin arkasında yatan temel prensiptir.

Giriş

İnternet ortamı için yapılan tasarımlarda, çoğu zaman bir içeriğin çok sayıda kullanılması gerekebilir. Kullanılan içeriğin yeniden çizilerek tasarlanması hem zaman kaybına neden olur, hem de dosya boyutunu artırır. Yapılan çizimler sembolere dönüştürüldüğünde kütüphane paneline yerleşir ve ihtiyaç duyulduğunda bu panelden alınarak istenen yerlerde yansımaları (örneği) kullanılır. Kütüphanedeki sembol düzenlendiğinde yapılan çalışmalar içerisine aktarılan yansımaları aynı anda değişeceğinden, tasarımların güncellenmesi de son derece kolaylaşır.

Bu ünite de bir Flash tasarımında kullanabilecek sembol türleri ve bu türlerin nasıl oluşturulabileceği hakkında detaylı bilgi verilecektir.

SEMBOLLERİ OLUŞTURMAK

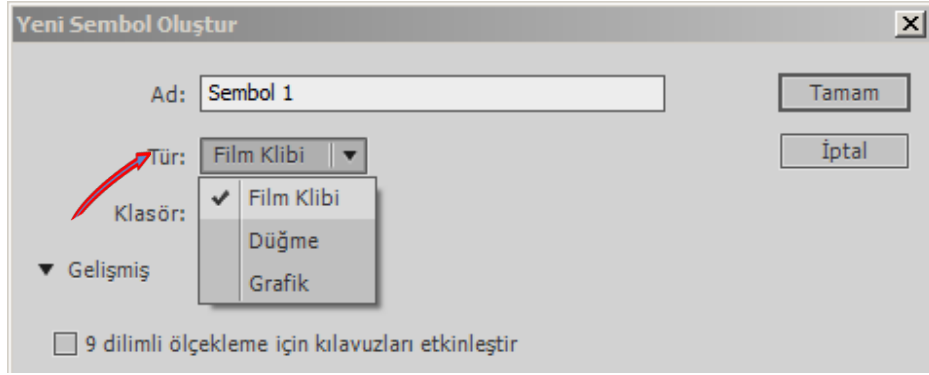
Adobe Flash'ın önemli özelliklerinden biri de çizilen grafikleri ve nesneleri sembole dönüştürebilmesidir. Flash'ta Grafik, Düğme, Film klipi olmak üzere üç tür sembol vardır. Sembolleri animasyonlar içinde kullanılarak, dosya boyutunu küçültebilir ve yükleme süresini kısaltabilirsiniz. Sembole dönüştürülen bir nesne Flash kütüphanesine eklenir. Sembol kullanılmak istendiğinde kütüphaneden çağrılırken bir yansıması (örneği) oluşturulur. Kütüphaneden ilgili sembol silindiğinde bütün kullanılan yansımaları da otomatik olarak silinir.

Sembol oluşturmak için iki yöntem vardır.

- **1.Yönetim:** Sahnenizde hiçbir şekil, yazı, grup yokken boş bir sembol oluşturabilirsiniz. Bunu yapmak için **Ekle (Insert)** menüsünden Şekil 4.1'de gösterilen **Yeni Sembol (Ctrl+F8) (New Symbol)** seçeneğine tıklayarak açılan Şekil 4.2'deki ekran üzerinden seçtiğiniz sembol türünde içi boş bir sembol oluşturabilirsiniz. Sembol oluşturulduktan sonra içeriğinin tasarımına geçilir.

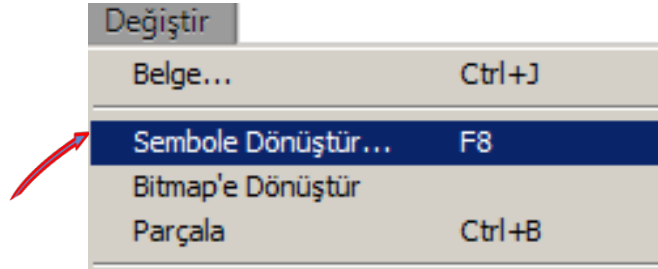


Şekil 4.1 Çalışma alanı içerisine yeni bir sembol ekleme işlemi



Şekil 4.2 Yeni bir sembol türü seçme işlemi

- **2.Yöntem:** Sahnenizde hazırladığınız şekil, yazı gruplarını önce seçerek *Değiştir (Modify)* menüsünden Şekil 4.3'te gösterilen *Sembole dönüştür (F8)* (*Convert to Symbol*) seçeneğini kullanarak, karşımıza çıkan Şekil 4.2'de gösterilen pencere üzerinden sembole bir isim verebilir ve sembolün tipini seçebilirsiniz. Aynı işlem sembol yapmak istediğinizde de nesnenin üzerinde iken farenin sağ tuşu ile de yapılabilir.



Şekil 4.3 Çalışma alanı içerisinde yer alan bir nesnin yeni bir sembole dönüştürülmesi

Semboller yukarıda anlatılan tekniklerle oluşturduktan sonra Şekil 4.5'te gösterildiği gibi Kütüphane Paneli'ne yerleşir. Kütüphane Paneli ekranda görüntülenmiyorsa *Pencere (Window)* Menüsünden Kütüphane seçeneğini kullanarak Şekil 4.4'teki gibi çalışma alanına eklenebilir.

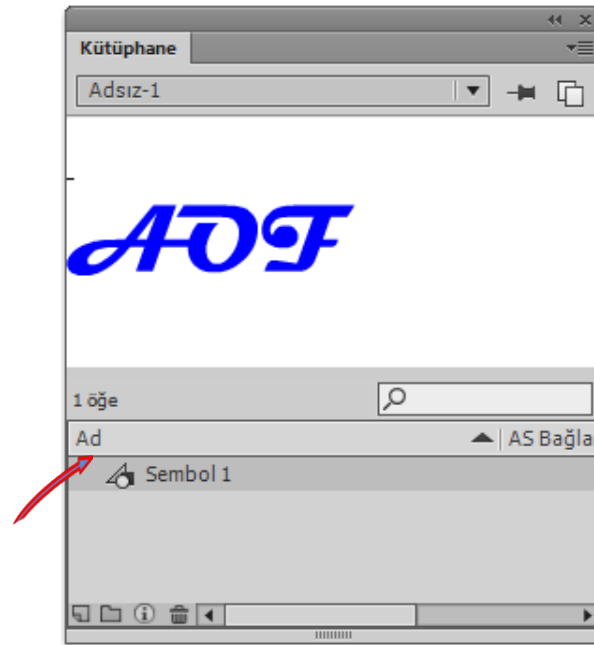
Kütüphane Paneli (Library Panel)

Çalışma alanı içerisinde oluşturduğunuz sembolleri Kütüphane Paneli'nden görebilir ve istediğiniz zaman kullanabilirsiniz. Kütüphane Paneli yardımıyla Sembolün ismini değiştirebilir, içine girerek kendi zaman çizgisinde düzenleme yapabilirsiniz.





Şekil 4.4 Çalışma alanı içerisinde Kütüphane Panelinin gösterilmesi



Şekil 4.5 Kütüphane panelinde bir grafik nesnesinin görüntülenmesi

SEMBOL ÇEŞİTLERİ ve ÖZELLİKLERİ

Adobe Flash programında kütüphane Paneli'nde saklanabilen üç çeşit sembol tanımlanabilir: Bunlar grafik, düğme ve film klibi sembolleridir. Her bir sembol türü farklı simgeler ile gösterilir ve Kütüphane Paneli'ndeki simgesine bakarak bu sembolün grafik mi? düğme mi? yoksa film klibi mi? olduğunu anlaşılabılır. Aşağıda bu üç sembol türü detaylı olarak anlatılacaktır.

Grafik (Graphic) Sembolleri

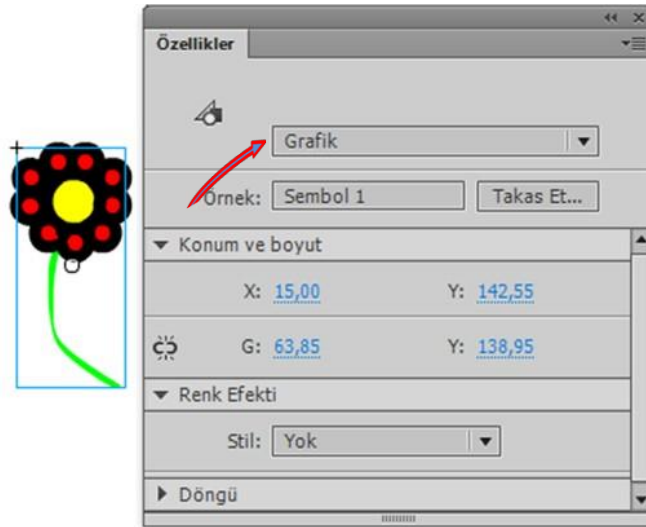
Grafik sembolleri kütüphane Paneli'nde () simgesiyle gösterilir. Genellikle resim, şekil gibi hareketsiz görüntülerden oluşturulur. Grafik sembolleri; yazılımsal olarak kontrol edilemeyen ve ana Zaman Çizelgesi'yle senkronize çalışan bir sembol türüdür. Çalışma alanına bir grafik sembolü oluşturmak için;

1. **Adım:** Şekil 4.6'daki nesneyi çalışma alanına çiziniz. Bunun için Genişlik: 30, Yükseklik: 30 olan bir daire etrafına, siyah 3 birim kalınlığında G: 15, Y: 15 dokuz adet küçük daireler çizerek yerleştiriniz ve içlerini Şekil 4.6'daki gibi boyayınız.



Şekil 4.6 Araç kutusundan daire ve diğer araçları kullanılarak çizilen çiçek görüntüsü.

2. **Adım:** Çizilen çiçek nesnesinin tamamını seçerek grafik sembolüne dönüştürünüz. Bunun için Seçim aracı seçiliyken farenin sol tuşuna basılı tutarak nesneyi içine alacak şekilde bir dikdörtgen çizersiniz. Böylece tüm görüntü seçilmiş olacaktır. Sonrasında sırasıyla *"Değiştir"* menüsünü kullanarak *"Sembole dönüştür"* komutu verilir, ardından açılan pencereden Grafik sembol türü seçilir ve sembol ismine çiçek adı verilerek işlem tamamlanır (Şekil 4.7).

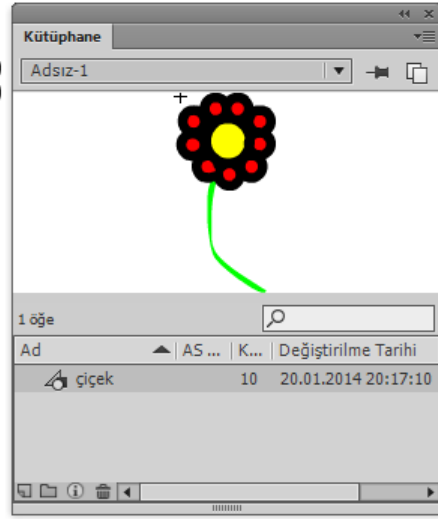
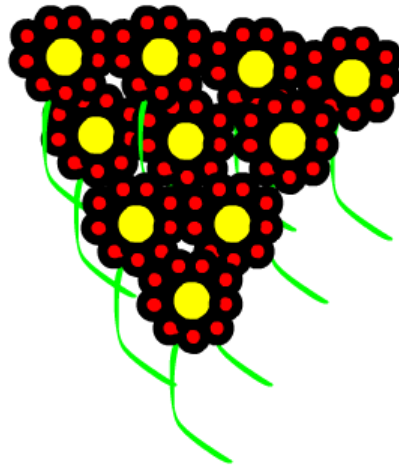


Şekil 4.7 Çiçek sembolünün özellikler

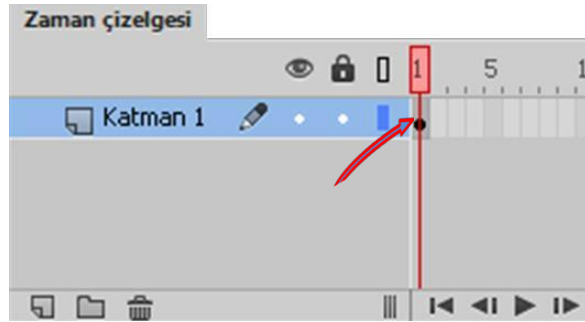
3. **Adım:** Çiçek sembolünü kütüphaneden seçerek, çalışma alanına birden fazla sembol yansımaları (örneği) taşıyarak Şekil 4.8 de gösterilen çiçek demeti oluşturabilirsiniz. Dikkat edilirse Şekil 4.9 da gösterildiği gibi oluşturulan grafik sembolünün zaman çizelgesinde sadece bir film karelik yer kaplar.



Grafik sembolünden tasarım ekranına ne kadar çok kopya taşınırsa taşınısın dosya boyutunda herhangi bir artışa neden olmaz.



Şekil 4.8 Çiçek sembolünün kütüphaneden çalışma alanına taşınması.



Şekil 4.9 Çiçek şeklinin grafik sembolü olarak zaman eksenindeki etkisi



Düğme sembolünün tasarımında değişiklik yapılacaksa, çalışma alanı içerisine Kütüphane Paneli'nde yer alan "Çoğalt" komutuyla oluşturulan kopyası taşınmalıdır. Aksi halde Sembolün kendisinde yapılan değişiklikler tüm kopya semboller üzerinde etkisini gösterir.

Düğme (Button) Sembolleri

Düğme (Button) sembolleri kütüphane Paneli'nde () simgesiyle gösterilir. Fare tıklatmaları, üzerine gelme durumları (fare ile bu sembolün üzerine gelindiğinde aldığı şekil), başka bunun gibi eylemlere karşılık veren (etkileşimli) düğmeler oluşturmak için kullanılan sembol türüdür.

Düğme sembolünün olaylara vereceği tepkiler şekilsel olarak (fare ile üzerine gelindiğinde oluşan görüntü) tanımlandıktan sonra bu düğme sembolüne eylemler atayabilirsiniz. Flash'ın kullanıcılara ve özellikle web tasarımcılarına sunduğu önemli özelliklerden biri de etkileşimli, üç boyut etkisine sahip düğmelerin hazırlanmasında kolaylık sağlamasıdır. Bu düğmeler sayesinde izleyiciler filmleri kontrol edebilir ve filmlere etkileşim katılabilirler.

Çalışma alanına bir düğme sembolü oluşturmak için;

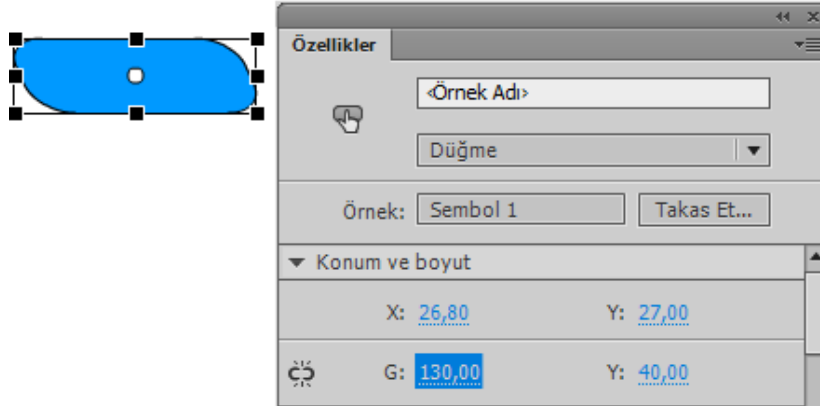
1. **Adım:** Şekil 4.10'daki nesneyi çalışma alanına çiziniz. Bunun için Genişlik: 130, Yükseklik: 40'lık dikdörtgen temel aracı ve Serbest

Dönüştürme aracını kullanabilirsiniz. Şeklin içerisini Şekil 4.10'daki gibi renklendiriniz.



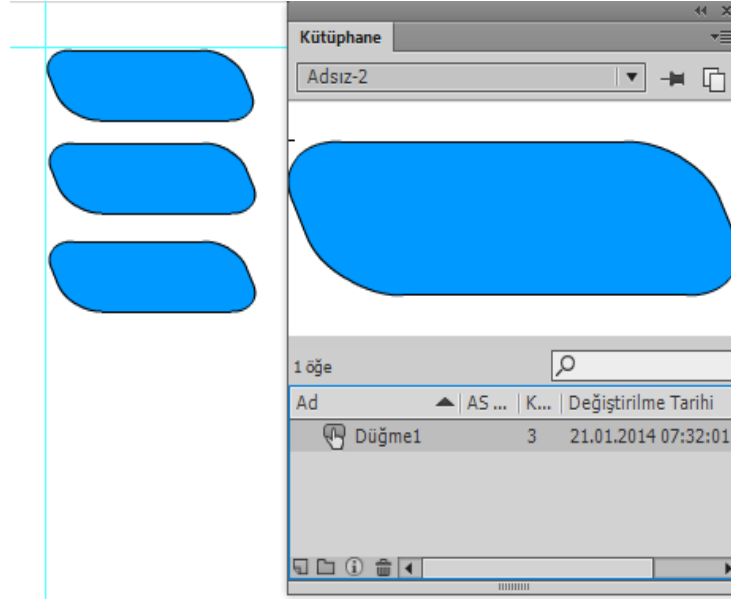
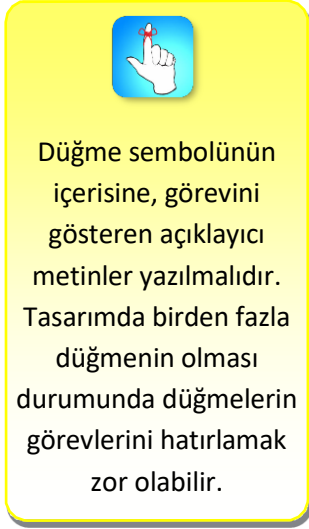
Şekil 4.10 Araç kutusundan dikdörtgen ve Serbest Dönüştürme araçları kullanılarak çizilen bir düğme şekli.

2. **Adım:** Çizilen bu nesnesinin tamamını seçerek Düğme sembolüne dönüştürünüz. Bunun için Seçim aracı seçiliyken farenin sol tuşuna basılı tutarak nesneyi içine alacak şekilde bir dikdörtgen çizersiniz. Böylece tüm görüntü seçilmiş olacaktır. Sonrasında sırasıyla “*Değiştir*” menüsünü kullanarak “*Sembol dönüştür*” komutu verilir, ardından açılan pencereden Düğme sembol türü seçilir ve sembol ismine Düğme1 adı verilerek işlem tamamlanır (Şekil 4.11).



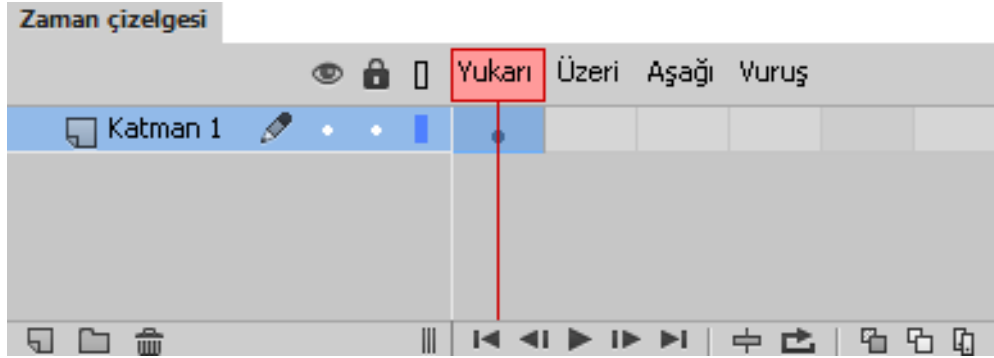
Şekil 4.11 Düğme1 sembolünün özellikleri.

3. **Adım:** Düğme1 sembolünü kütüphaneden seçerek, çalışma alanına birden fazla sembol yansıması taşıyarak Şekil 4.12'de gösterilen üç tane düğme oluşturunuz. Bu işlem sırasında çalışma alanına cetvel ve klavuz çizgileri ekleyerek düğmelerin konumunu ayarlayabilirsiniz.



Şekil 4.12 Düğme1 sembolünün kütüphaneden çalışma alanına taşınması.

Düğme sembolüne çift tıklayarak şekil 4.13'te gösterildiği gibi sembolün zaman eksenini açılabilir. Zaman panelinde dört karelik bir alan kapladığı görülür. Düğme sembolünün kendi zaman eksenini diğer sembollerden farklıdır. Diğer tip sembollerin zaman çizgileri Ana zaman çizgisinin aynısıdır. Ancak Düğme Sembolleri zaman ekseninde dört film karesinden oluşur.



Şekil 4.13 Düğme1 şeklinin Düğme sembolü olarak zaman eksenindeki etkisi

Bu karelerin anlamı aşağıdaki gibidir:

- **Yukarı:** Bu kareye çizilen şekil düğmesinin ilk görünen şeklidir.
- **Üzeri:** Fare imleci düğme üzerine geldiğinde görünecek şekil buraya çizilir.
- **Aşağı:** Fare imleci düğme üzerindeyken tıklandığında düğmenin alacağı şekil buraya çizilir.
- **Vuruş:** Düğmenin vuruş alanıdır. Buraya çizdiğiniz şekil önemli değildir. Ancak buraya Düğmenizin kapladığı alan kadar bir şekil çizmeye çalışınız.

Düğmenin etki alanı ve fare imleci düğmenin üzerine geldiğinde imlecin el

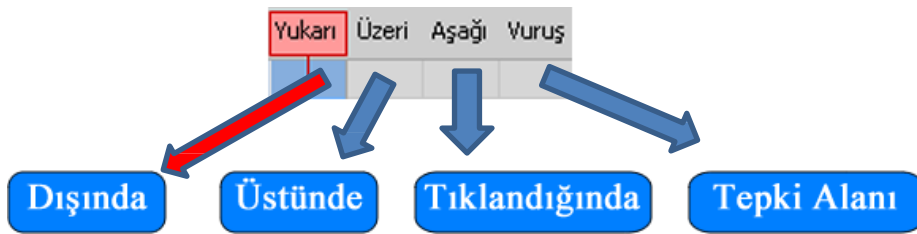


() şekli almasını sağlayacak yüzey alanı bu kare yardımıyla belirlenir.

Düğme sembolünün tepki alanının çok küçük seçilmesi düğmenin kullanılmasını zorlaştırır.

Şekil 4.13'te gösterilen zaman çizelgesinde sadece Yukarı durumu tanımlanmıştır. Diğer karelerin boş olduğu görülmektedir. Bunun anlamı Düğme çalışma alanında çizel şekildeki görülmekte iken fare ile üzerine gelip, tıklayınca hiçbir şeklin görüntülenemeyeceğidir. Bu boş film karelerine önceden tasarlanan ve kütüphane panelinde yer alan sembollerden herhangi birisi eklenebilir.

Bir sembol türünün üzerine fare ile tıklandığında Şekil 4.16'da gösterildiği gibi bir düzenleme penceresi açılır.



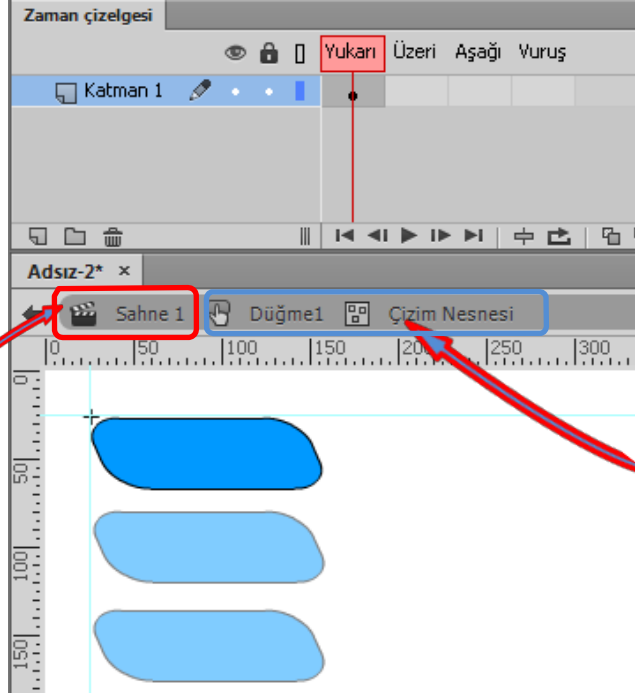
Şekil 4.14 Düğme1 tanımlanabilecek alanlarının gösterilmesi

Zaman çizelgesinde önceden "Yukarı" durumu tanımlandığı için Şekil 4.14'te içi dolu nokta ile gösterilmiştir. Boş görülen alanların her birine ise fare ile üzerine gelinip Sağ tuşa basarak ilgili film karelerine "Boş Anahtar Kare Ekle" komutu verilerek Şekil 4.15'de gösterilen, buton şekilleri ilgili yerlere aktarılarak düğmenin yeni görüntüsü tasarlanır. Şayet "Anahtar Kare Ekle" komutu kullanılırsa bir önceki film karesinden şekil bu seçili film karesine aktarılabilir. Sonrasın da ise şeklin renkleri değiştirerek Şekil 4.16'da gösterilen alandaki Sahne1 isim alanına tıklanarak sembol düzenlemesi tamamlanır. Sahneye kütüphaneden eklenen aynı ismi taşıyan bütün sembollerin kopyalarında yapılan tasarımı ekrana aynı yansıtır.



Şekil 4.15 Düğme1 'in kırmızı okla gösterilen durumu "Yukarı", sırasıyla "Üstünde", "Tıklandığında" aldığı tasarım şekilleri

Bulduğunuz Sahneyi gösterir. Sahneye geri dönmek için buraya tıklayabilirsiniz. Sembolün içine girmek için üzerinde fare ile çift tıklayabilirsiniz. Hangi sembol üzerinde işlem yapılıyorsa o sembol renkli görünürken diğerleri soluklaşır.



Oluşturulan sembolün içinde olduğunuzu gösterir. İsterseniz kütüphaneden yeni nesneleri bu alana ekleyebilirsiniz.

Şekil 4.16 Düğme1 sembolünün düzenlenmesi

Film klibi (Movie clip) Sembolleri

Film klibi sembolleri kütüphane panelinde () simgesiyle gösterilir. Etkileşimli ve en esnek sembol türüdür. Ses, animasyon, düğme gibi filmdeki tüm öğeler bu sembol içinde kullanılabilir. Film kliplerinin Ana Zaman Çizelgesi'nden bağımsız kendi çok kareli Zaman Çizelgeleri vardır. Ana filmin zaman çizgisi dursa bile onların kendi zaman çizgileri oynamaya devam eder. Animasyonlu düğmeler oluşturmak için film klibi örneklerini düğme sembolünün Zaman Çizelgesi'nin içine yerleştirilebilir.

Çalışma alanına bir Film Klibi(Movie clip) sembolü oluşturmak için;

1. **Adım:** Şekil 4.17'deki nesneyi çalışma alanına çiziniz. Bunun için Kalem, Fırça, Boya Kovası ve Metin araçlarını kullanabilirsiniz.



Şekil 4.17 Savaş gemisi görüntüsü

2. **Adım:** Çizilen bu nesnesinin tamamını seçerek Grafik sembolüne dönüştürünüz. "Savaş gemisi" ismi veriniz. Daha sonra Ekle menüsünden Yeni Sembol komutunu kullanarak Film Klibi oluşturunuz ve adını da "Gemi_filmi" olarak değiştiriniz. Zaman Panelinden üç

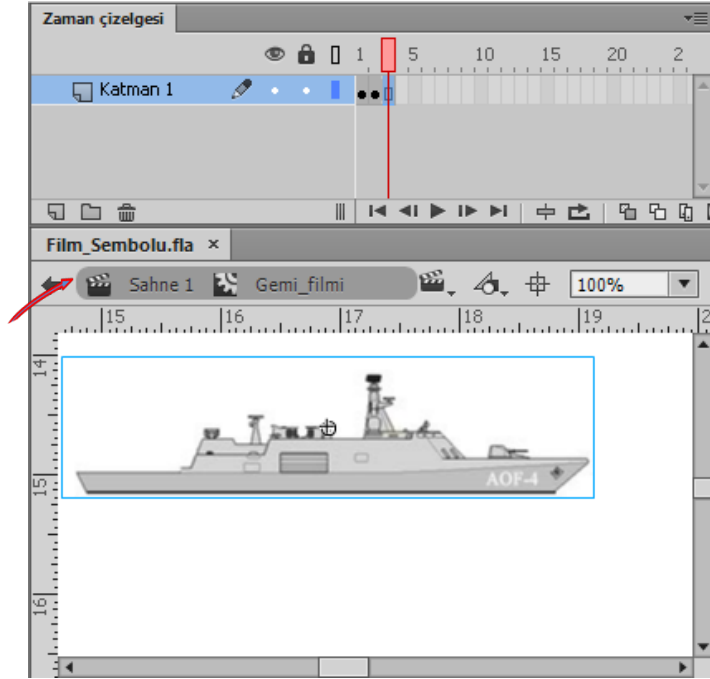
adet “Boş Anahtar Kare” ekleyerek bunların içlerine Şekil 4.18’de gösterilen şekillerden kırmızı okla işaretlenen Savaş gemisi resmi Şekil 1.19’da gösterilen 1 nolu film karesine diğerleri ise sırasıyla 2 ve 3 nolu boş film karelerine yerleştirilir.



Şekil 4.18 Gemi_Filmi isimli Film sembolüne eklenecek görüntüler



Film Klibi sembolünün, Düğme ve Grafik sembolünden daha fazla Anahtar Film Karesine ihtiyacı vardır. Anahtar Kare Sayısı artırıldığında animasyon göze daha akıcı gözükür.



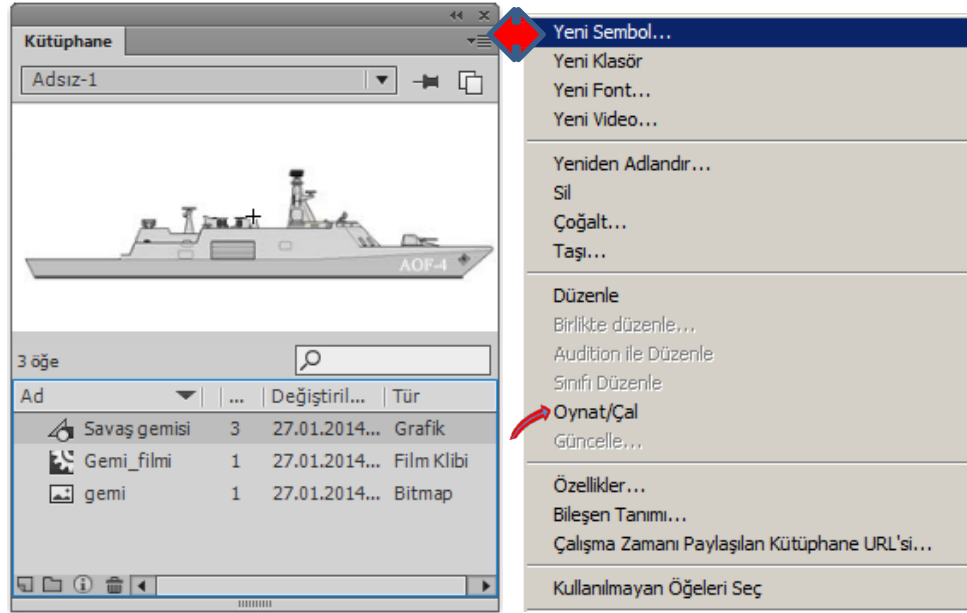
Şekil 4.19 Film Klibinin Zaman Çizelgesi ve çizim düzenleme ekranı

Şekil 4.19’da gösterilen ekranda Sahne1 isim alanına tıklanarak Film klibi tamamlanmış olur. Şekil 4.20’de tasarımda kullanılan bütün semboller görülmektedir. Kütüphane panelinde 3 adet Grafik sembolü ve 1 adetde Film Klibi sembolü kullanıldığı görülmektedir. Bu panelden semboller sürüklenerek çalışma alanına taşındığında Ana zaman ekseninde bir kare içerisinde yer almasına rağmen her sembolün kendi zaman eksenine sahiptir.

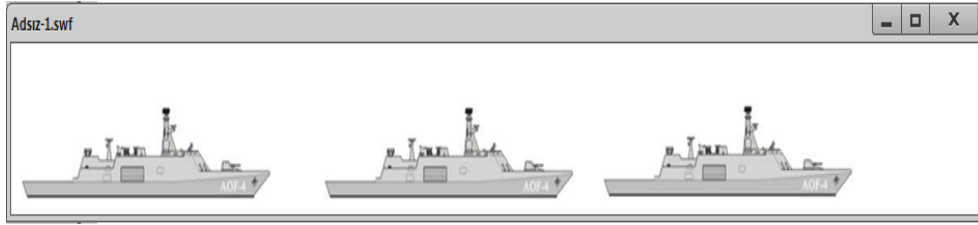
Bir Film klibinin nasıl çalıştığını test etmek için “*Kontrol Et*” menüsünden “Test Et” veya “Sahneyi Test Et” komutu verilmelidir. Bu komut sonrası açılan Test ekranında sahnede yer alan her semboller kendi zaman çizelgelerinde bulunan film karelerini ekranda teker teker gösterirler (Şekil 4.21). Gemi_Filmi isimli sembolün zaman dilimindeki film kareleri, tek tek işlenerek Test ekranında bir gemi hareket ediyor görüntüsü ortaya çıkarılır.



Bir Film Klibinin nasıl hareket ettiğini kütüphane içerisinden izlemek için “Oynat/Çal” komutu verilmelidir.



Şekil 4.20 Gemi_Film klibi ve diğer sembollerin Kütüphane Paneli'nde görüntülenmesi



Şekil 4.21 Test Ekranında tümü gösterilen 3 ayrı savaş gemisi



Özet

- Adobe Flash programıyla grafikler sembollere dönüştürülerek animasyon içerisinde kullanılabilir.
- Adobe Flash'ta Film klibi, Düğme ve Grafik olmak üzere üç tür sembol vardır.
- Semboller birden fazla kullanılabilen nesnelerdir. Sembolleri animasyonlar içinde kullanılarak, dosya boyutunu küçültebilir ve yükleme süresini kısıltabilirsiniz.
- Sembole dönüştürülen bir nesne Flash kütüphanesine eklenir. Sembol kullanılmak istendiğinde kütüphaneden çağrılırken bir örneği oluşturulur. Kütüphane Paneli'nde bir sembolün yanındaki simgeye bakarak bunun bir grafik mi düğme mi yoksa film klibi mi olduğunu anlayabilirsiniz.
- Genellikle film ve Düğme sembollerinin pek çok yerinde kullanılan hareketsiz görüntülerdir. Grafik sembolleri; Yazılımsal olarak kontrol edilemeyen ve Ana Zaman Çizelgesi'yle senkronize çalışan bir sembol türüdür.
- Düğme sembolleri, fare tıklatmaları, üzerine gelmeler ve başka eylemlere karşılık veren, etkileşimli düğmeler oluşturmak için kullanılan sembollerdir. Çeşitli düğme durumlarıyla ilişkili grafikleri tanımlayabilir ve sonra bir düğme örneğine eylemler atayabilirsiniz. Düğmeler dört film karesinden oluşur. Bu karelerin anlamı aşağıdaki gibidir: Yukarı: Bu kareye çizilen şekil düğmenin ilk görünen şeklidir. Üzeri: Fare imleci düğme üzerine geldiğinde görünecek şekil buraya çizilir. Aşağı: Fare imleci düğme üzerindeyken tıklandığında düğmenin alacağı şekil buraya çizilir. Vuruş: Düğmenin vuruş alanıdır. Buraya çizdiğiniz şekil önemli değildir. Ancak buraya butonunuzun kapladığı alan kadar bir şekil çizilmelidir. Şekilin sınırları içerisinde etkileşim gerçekleşir.
- Film klibi sembolleri, en etkileşimli, esnek sembol türüdür. Ses, animasyon, düğme gibi filmdeki tüm öğeleri içerebilen bağımsız, sembollerdir. Film kliplerinin Ana Zaman Çizelgesi'nden bağımsız kendi çok kareli zaman çizelgeleri vardır. Ana filmin zaman çizgisi dursa bile onların kendi zaman çizgileri oynamaya devam eder. Animasyonlu düğmeler oluşturmak için film klibi örnekleri bir düğme sembolünün Zaman Çizelgesi'nin içine yerleştirilir.

DEĞERLENDİRME SORULARI

1. Birden fazla kere kullanılabilen nesnelerdir. Animasyonlar içinde kullanılarak, dosya boyutunu küçültebilir ve yükleme süresini kısaltabilirsiniz. Seçeneklerden hangisinin yukarıdaki verilen özelliklere sahiptir?
 - a) Sembol
 - b) Dosya
 - c) Resim
 - d) Nesne
 - e) Katman
2. Düğme üzerine geldiğinde görünecek şekil seçeneklerden hangisinde verilmiştir?
 - a) Yukarı
 - b) Üzeri
 - c) Aşağı
 - d) Vuruş
 - e) Sola
3. Fare imleci düğme sembolü üzerindeyken tıklandığındaki şekil, seçeneklerden hangisinde verilmiştir?
 - a) Yukarı
 - b) Üzeri
 - c) Aşağı
 - d) Vuruş
 - e) Sola
4. Çizimler, grafik sembolüne nasıl dönüştürülür?
 - a) Değiştir > Sembole dönüştür >Grafik
 - b) Değiştir > Sembole dönüştür >Düğme
 - c) Değiştir > Sembole dönüştür >Film klibi
 - d) Değiştir > Dönüştür >Grafik
 - e) Değiştir > Dönüştür >Ölçek



5. () simgesi Kütüphane panelinde hangi sembolü gösterir?
 - a) Klasör
 - b) Bileşen
 - c) Film klibi
 - d) Düğme
 - e) Grafik

6. Adobe Flash içerisinde kullanılan bir düğmenin zaman çizelgesinde kaç özel durumu tanımlanır?
- a) İki
 - b) Üç
 - c) Dört
 - d) Beş
 - e) Altı
7. Kütüphane Paneli'nde yer alan bir sembolün, birden fazla kopyasının yeniden tasarlanması gerektiğinde hangi komut kullanılmalıdır?
- a) Taşı
 - b) Çoğalt
 - c) Sil
 - d) Yeni Klasör
 - e) Yeni Sembol



8. () simgesi Kütüphane Paneli'nde hangi sembolü gösterir?
- a) Klasör
 - b) Bileşen
 - c) Film klibi
 - d) Düğme
 - e) Grafik



9. Yanda verilen zaman çizelgesinde tasarımı tamamlanmamış bölüm hangi seçenekte verilmiştir?
- a) Yukarı
 - b) Üzeri
 - c) Aşağı
 - d) Vuruş
 - e) 5
10. Üzerine gelince kendiliğinden küçülüp büyüyen bir Animasyon gösteren düğme tasarlanacaktır. Düğmenin "Üzeri" durumu içerisinde hangi nesne eklenince bu durum gerçekleştirilebilir?
- a) Klasör
 - b) Bileşen
 - c) Film klibi
 - d) Düğme
 - e) Grafik

Cevap Anahtarı

1.a, 2.b, 3.c, 4.a, 5.c, 6.c, 7.b, 8.d, 9.c, 10.c

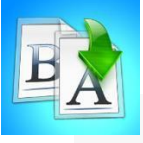
YARARLANILAN VE BAŞVURULABİLECEK DİĞER KAYNAKLAR

[1]Adobe Flash Pro CC, Yardımcı Esl kitabı. <http://helpx.adobe.com/tr/support>

[Erişim tarihi: 30 Ekim 2013].

[2]SCHULZE Pati.(2004) Macromedia Fireworks 8 Kaynağından Eğitim,
Medyasoft Yayınları.

ZAMAN ÇİZELGESİ



İÇİNDEKİLER

- Zaman Çizelgesi
- Anahtar Kareler
- Boş Anahtar Kareler
- Normal Kareler
- Ara Doldurma Kareleri
- Katmanlar
- Kontrol Et Menüsü



HEDEFLER

- Bu üniteyi çalıştıktan sonra;
- Zaman çizelgesi üzerindeki bölümleri tanıyabilecek,
- Film karelerini düzenleyebilecek,
- Katmanları ve Sahneleri organize edebilecek,
- Tasarımlardaki her çeşit animasyonu çalıştırabileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

ANİMASYON TASARIMI

Okt.
Mustafa AKBUĞA

ÜNİTE 5



Animasyonlar, film karelerinin bir araya gelmesiyle oluşur.

Giriş

Animasyon Tasarım Programı (Adobe Flash Pro CC) yardımıyla, proje içeriklerini ve animasyonları oluşturan her kareyi zaman çizelgesi üzerinden yönetebilirsiniz. Zaman çizelgesinde ses, resim, grafik, video veya ActionScript programlama kodları önce kare filmler içerisine aktarılır. Sonrasında bu nesneler üzerinde gösterim süresi, *en önde* veya *en sonda* göster gibi sıralamalar ayarlanır.

Bu ünite de bir animasyon tasarımında kullanılan Anahtar Kare, Katman, Sahne ve Kontrol Et menüsü gibi konularda hakkında bilgi verilecektir.

ZAMAN ÇİZELGESİ

Zaman ve hareket dilimlerini belirleyen kareler animasyonlarda kullanılan en küçük tasarım birimleridir. Filmlerde kullanılan karelerin sayısı ve onların oynatıldığı hız, filmin uzunluğunu belirler. Adobe Flash dosyalarındaki zamanlama gerektiren ses, resim, video gibi birçok şey zaman çizelgesi (Timeline) üzerine yerleştirilir. Dosya içeriğinin zaman çizelgesi üzerinde nasıl sıralanmış oldukları animasyonun nasıl görüntüleneceğini belirler. Zaman Çizelgesi üzerinde üç çeşit alan bulunur. Bunlar, Anahtar kareler, Kareler ve Boş film alanlarıdır. Anahtar kareler üzerinde, çalışma ekranında gösterilecek nesneler tasarlanırken; Kareler, Anahtar karelerin üzerindeki tasarımların etkisini devam ettirmek amacıyla kullanılır. Boş film alanları ise henüz kare yada Anahtar kare tanımlanmamış bölümlerdir. Boş film alanlarda film oynatılamaz, filmlerin var olabilmesi için en az bir tane Anahtar kareye ihtiyaç duyulur. Şekil 5.1’de görülen Zaman çizelgesinde kırmızı okla gösterilen 1 nolu alan içi boş Anahtar film karesini, 10 nolu alan içi dolu Anahtar karesini, şekilde 2 ile 9 numaralı alanlar arasında içi boş kareleri ve 11-24 numaralı alanlar arasında ise içi dolu film karelerini gösterilmektedir. Boş film alanları ise 25. kareden sonrasında görüntülenmektedir.

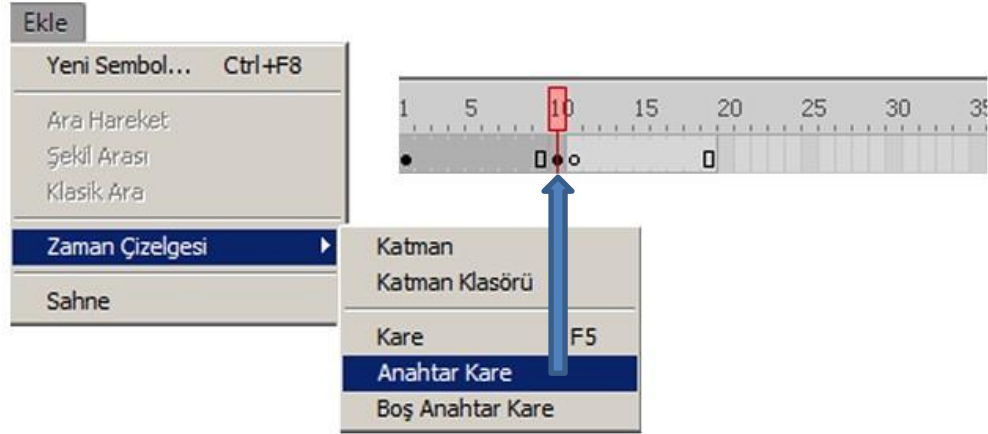


Şekil 5.1 Adobe Flash Professional programının zaman çizelgesinden bir bölüm

Zaman çizelgesindeki film kareleri aşağıdaki gibi gruplandırılabilir.

Anahtar Kareler (Keyframe) :

Animasyonunuzda bir değişikliğin olmasını veya bir eylemin ortaya çıkmasını istediğinizde Zaman çizelgesinde (Timeline) o noktaya bir *Anahtar kare (Keyframe)* eklemeniz gerekir. Bu ekleme işlemlerini *Ekle (Insert)* menüsünü kullanarak veya Zaman çizelgesinin üzerinde sağ tuşa tıklayarak yapabilirsiniz. Şekil 5.2’de 10 nolu film karesi bu şekilde eklenmiştir.



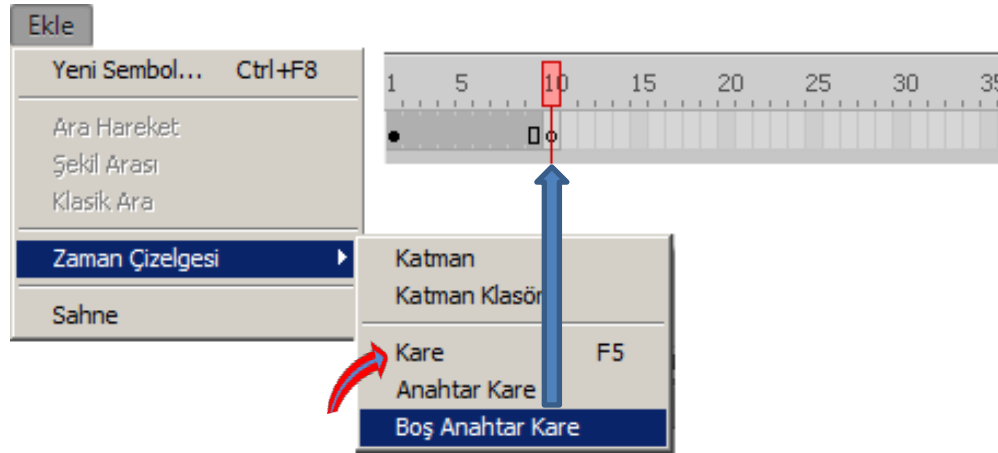
Şekil 5.2 Zaman çizelgesine Ekle menüsünden Anahtar kare ekleme işlemi



Eklenen kareleri silmek veya yeniden eklemek için zaman çizgisi üzerinde o karelerin üzerinde farenin sağ tuşunu kullanabiliriz.

Boş Anahtar Kareler (Blank Keyframe) :

Zaman çizelgesine eklenen boş anahtar kareler o katmanın içeriğinin artık görünür olmamasını sağlayan karelerdir. Boş anahtar kare ekleme işlemlerini *Ekle (Insert)* menüsünü kullanarak (Şekil 5.3) veya Zaman çizelgesi üzerinde sağ tuşa tıklanarak yapabilir.



Şekil 5.3 Zaman çizelgesine Ekle menüsünden boş Anahtar kare ekleme işlemi

Normal Kareler (Frame) :

Normal kareler, Anahtar bir kare ile başlayan hareketin devam süresini belirler. Bu kareleri *Ekle (Insert)* menüsünden veya Timeline üzerinde eklemek istediğimiz boş karenin üzerinde farenin sağ tuşuna basarak ekleyebiliriz.

Normal kareler her zaman *Anahtar kareyi* takip eder ve onunla aynı içeriğe sahiptir. İçeriğin değiştiği ve yeni bir hareketin başladığı yer tekrar bir *Anahtar kare* eklenen yerdir. Bu işlem Şekil 5.3'te kırmızı okla gösterilmektedir.

Ara Doldurma Kareleri (Tween Frames) :

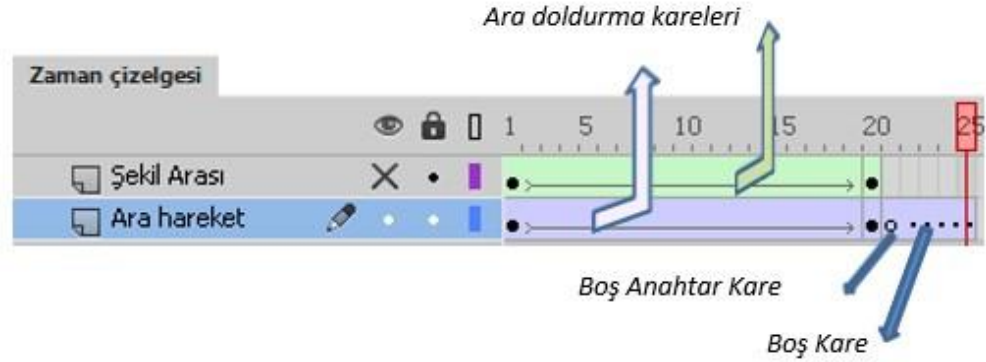
Zaman çizelgesi üzerinde bu tip kareler oluşturulurken başlangıç ve bitiş noktalarındaki Anahtar kareler tasarımcı tarafından tasarlanır ve arada kalan film



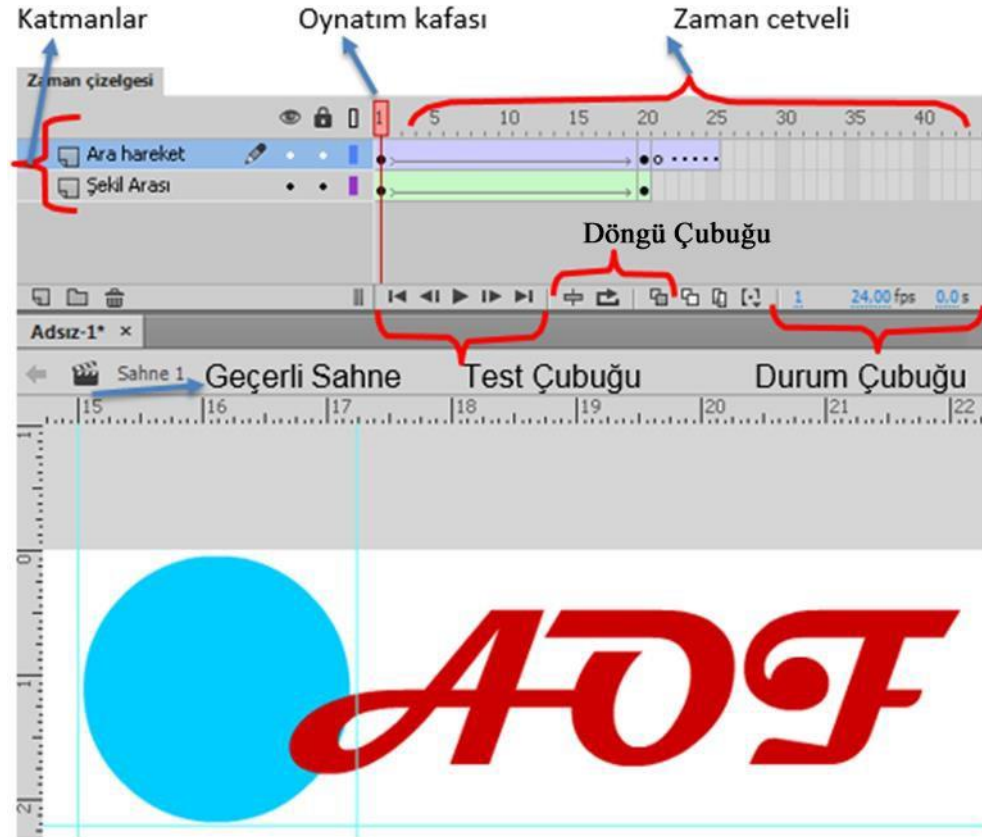
Her sembolün kendine özel zaman çizelgesi vardır.

kareleri ise Adobe Flash'a bırakılır. Aradaki karelerin değişimleri Flash programı tarafından hesaplanır ve bu hesapların sonuçlarına göre yine Anahtar kare olarak Zaman çizelgesine yerleştirilir.

Ara doldurma kareleri iki çeşittir; *Hareket (Motion)* ve *Şekil (Shape)*. Bu kareler Şekil 5.4 de farklı renklerde gösterilmektedir. Hareket animasyonunda nesnelerin bulundukları konum ve dönüş açıları kullanılarak animasyon oluşturulur. Şekil animasyonunda ise bir şekli (örneğin, Daire) başka bir şekle (örneğin, Dikdörtgen) dönüştürmek için kullanılır.



Şekil 5.4 Zaman çizelgesinde film karelerin gösterimi



Şekil 5.5 Çalışma alanındaki bölümler

Zaman Çizelgesi'nin Şekil 5.5'te gösterilen işaretli bölümler şunlardır;

-  **Durum Çubuğu**, animasyon hızıyla ilgili bilgi verir. Üç tür bilgi sunar.
-  **Geçerli Kare**, üzerinde çalışılan film karesini gösterir.
-  **Oynatım Hızı**, saniyede gösterilen kare sayısını ifade eder.
-  **Geçen Süre Göstergesi**, birinci kareden oynatma kafasının bulunduğu film karesine kadar aradaki karelerin ekranda kalma süresini gösterir.
-  **Döngü Çubuğu**, animasyon içerisinde bulunan karelerin gösterimi bittikten sonra kaç kez daha gösterileceğini belirler. Merkez kare, Döngü, Döngü kaplaması ve anahtarı, birden çok kare düzenle gibi seçenekleri vardır.
- **Test Çubuğu**, yapılan animasyon tasarımının test edilmesi amacıyla kullanılır. Toplamda beş adet kontrol tuşuna sahiptir. Bunlar kareler üzerinde İlk kareye git, Bir kare geri git, Oynat, Bir kare ileri git, Son kareye git gibi kontrollerin yapılmasını sağlar.
-  **Geçerli Sahne**, bir animasyon üzerinde zaman cetvelinde nasıl ki birden fazla film karesi varsa, aynı zamanda birden fazla sahnede eklenebilir. Ekranda şekilde gösterilen sahne, üzerinde çalışılmakta olan aktif sahneyi gösterir.
-  **Oynatım Kafası**, zaman cetvelinde Sahne üzerinde gösterilen kareyi işaretlemek için kullanılır.
- **Zaman Cetveli**, bir belgenin içeriğini zaman çizelgesi içerisinde, sayısal numaralandırma yaparak kareler halinde gösterir ve ActionScript komutlarıyla kontrol edilmesini kolaylaştırır.
-  **Katmanlar**, zaman çizelgesinin sol tarafındaki bir sütunda listelenir. Her katmandaki kareler katman adının sağında bulunan bir satırda görüntülenirler. Şimdi katmanlar konusu daha ayrıntılı inceleyelim;

KATMANLAR (LAYERS)

Katmanlar; semboller, Sesler, ActionScriptler ve diğer nesneler için ambalaj vazifesi görürler. Bunlar film oluşturan nesneleri özelliklerine göre birbirinden ayırmak için oluşturulur.

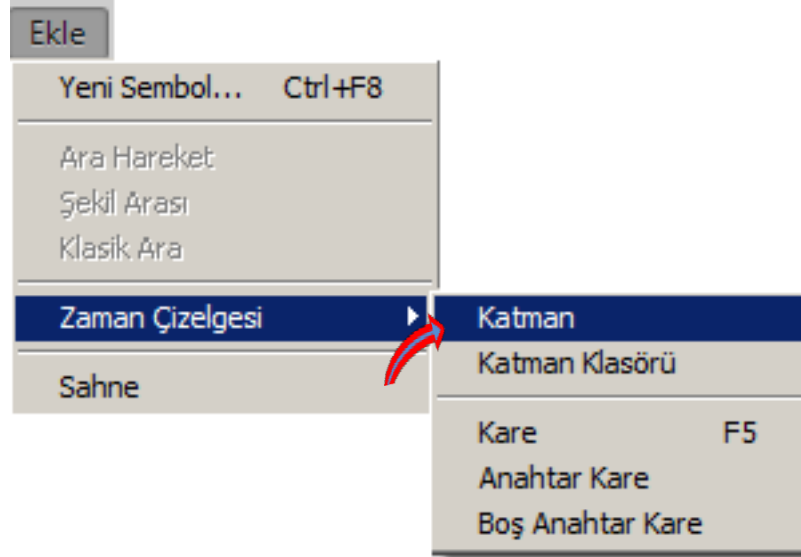


Animasyonlarda yer alan içerikler çoğaldıkça, yönetilmeleri de zorlaşır. Bunları organize etmek amacıyla katman klasörleri kullanılır.

Katman (LAYER) Ekleme:

Bir sahneye katman eklemenin iki yolu bulunmaktadır. Bunlar:

- **Ekle (Insert)** menüsündeki Zaman Çizelgesinden katman seçeneğine tıklayarak yeni bir katman ekleyebilirsiniz (Şekil 5.6). Bu işlemleri yaptığınızda yeni katmanı bulunduğumuz katmanın bir üzerine eklersiniz.



Şekil 5.6 Zaman çizelgesine katman eklemek

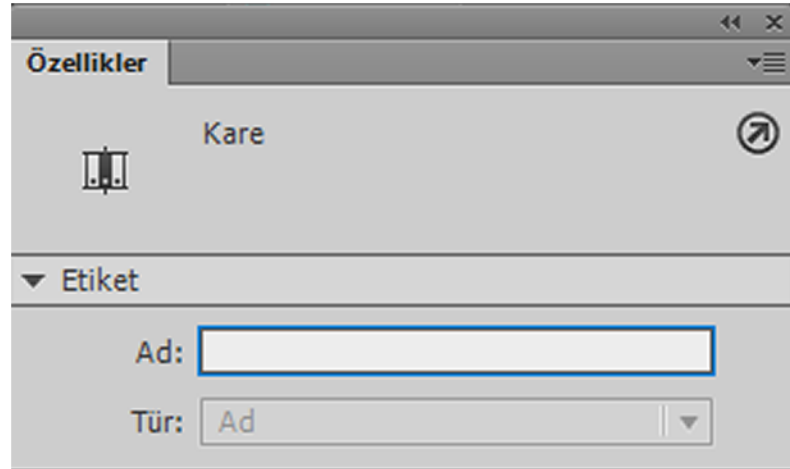
- **Zaman Çizelgesi üzerindeki Yeni Katman (Insert Layer)** seçeneğine tıklayarak bulunduğunuz katmanın üzerine yeni katman ekleyebilirsiniz. Katmanın ismini değiştirmek isterseniz, yapmanız gereken tek şey katmanın üzerine fare ile iki kez tıklamaktır (Şekil 5.7). Bu işlemi yaptığımızda "Katman2" olan ismi hazırladığınız katmanın içeriğine göre değiştirebilirsiniz. Aynı işlemi değişiklik yapmak istediğiniz katmanın üzerinde farenin sağ tuşuna tıklayarak çıkan menüden Özellikler (Properties) seçeneğini seçerek, katmanın özelliklerini ve adını değiştirebilirsiniz (Şekil 5.8).



Şekil 5.7 Zaman çizelgesi üzerinden katman eklemek



Zaman çizelgesi üzerindeki kalem sembolünün üzerinde çizgi varsa o katman düzenlenemez.



Şekil 5.8 Eklenen yeni katmanın isminin değiştirilmesi

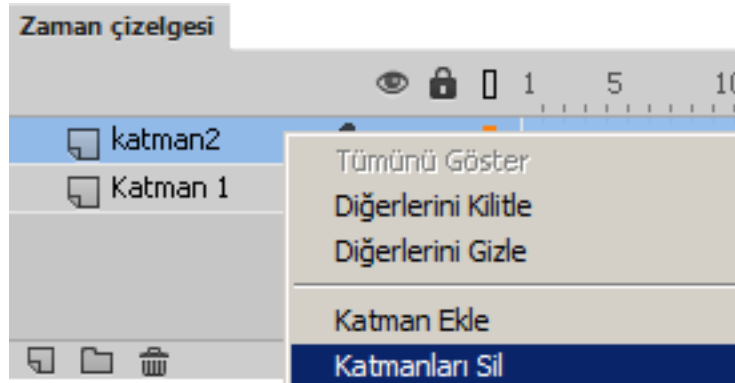


Çöp kutusu sembolü, seçili olan katmanı kolay yoldan silmek için kullanılır.

Katman Silmek:

Katmanları silmek için yapmamız gerekenler;

1. Silmek istediğimiz katmanın üzerinde farenin sağ tuşuna tıkladığınızda “Katmanları Sil” seçeneği çıkar (Şekil 5.9). Bu seçeneği seçerseniz katman silinir.



Şekil 5.9 Katmanı silmek

2. Diğer bir yol ise Zaman çizelgesi üzerinden katmanların olduğu bölümünde



bulunan çöp kovası () simgesidir. Herhangi bir katman üzerinde iken bu simgeye tıkladığınızda bulunduğunuz katman silinir.

Katman Modları

Katmanlar üç farklı duruma veya moda girebilir.

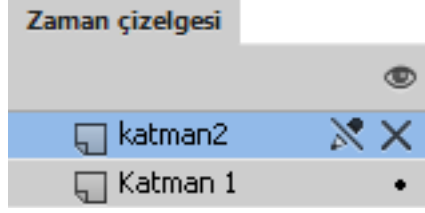
Gizlenmiş mod : () Sahne üzerinde çalışırken, bazı katmanların içeriğini gizlemek için kullanılır. Şekil 5.10’da gizlenmiş katman2 görülmektedir. Katmanın isim çubuğunda görülen nokta işareti (.), çarpı işaretine (x) dönüştüğünde katmanın gizlenmiş olduğunu belirtilir. Kalem simgesi üzerinde ise bir çizgi belirir. Bu işaret katman gizlendiği için üzerinde düzenleme

yapılamayacağını gösterir. Bu işaretin üzerine bir kez daha tıklandığında (.) şekline dönüşür ve böylece gizlenmiş moddan çıkılır.



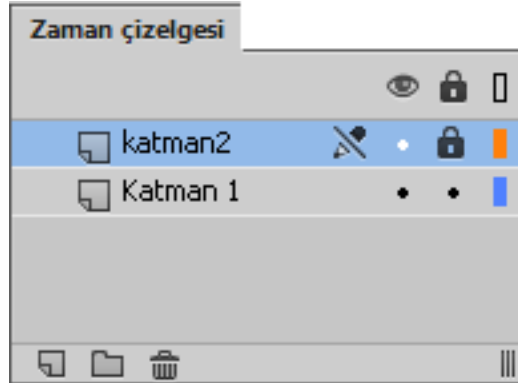
Kilitli mod, katman üzerindeki nesnelerin değiştirilmesini engeller ve bu şekilde ekranda göstermesine izin verir.

Gizlenmiş mod ise nesnelerin görüntülerini tamamen ekrandan kaldırır. Bu yüzden yeni eklenen katmanların oluşturulmasında, Kilitli mod, Gizlenmiş moddan daha kullanışlıdır.



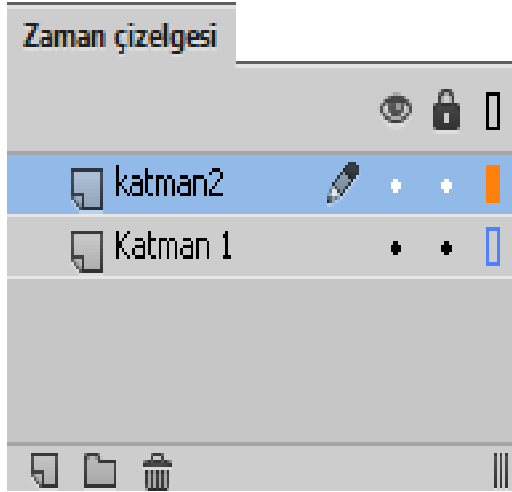
Şekil 5.10 Bir katmanı gizlemek

- **Kilitli mod:** () Bir katmanı bu simgeye tıklayarak kilitlediğinizde o katmanın içeriği görünür ancak üzerinde herhangi bir düzenleme yapılamaz. Bu modu, eğer bir katmandaki çalışmanızı bitirdiğinizde ve üzerinde değişiklik yapmak istemediğinizde emin olduğunuzda, diğer bir katmanla çalışırken yanlışlıkla değiştirmenizi engellemek için kullanabilirsiniz. Şekil 5.11’de kilitlenmiş katman2 görülmektedir. Dikkat edilirse gizlenmiş moddaki gibi Kalem simgesi üzerinde bir çizgi belirir. Bu katman kilitlendiği için üzerinde düzenleme yapılamaz. Bu işaretin üzerine bir kez daha tıklandığında (.) şekline dönüşür. Böylece kilitli moddan çıkılmış olunur.



Şekil 5.11 Bir katmanı kilitlemek

- **Dış Hat Modu :** () Katmanınız bu moddayken içeriği sadece dış hatlar hâlinde görünür. Şekil 5.12’de dış hat moduna alınmış katman2 görülmektedir. Bu mod öğelerin kenarlarını kolayca ayırt etmenizi sağlayarak şekilleri düzenlemeniz ve değiştirmeniz kolaylaştırır.

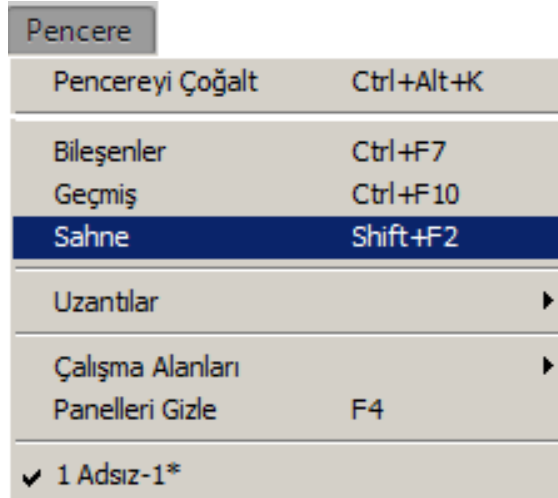


Şekil 5.12

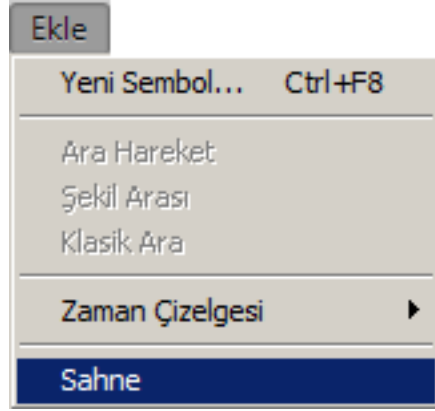
Sahne

Adobe Flash'taki Sahne, tıpkı stüdyoda film çekiminde kullanılan gerçek bir sahne gibi, izleyenlerin oynatılan filmi görmesini sağlayan alan gibidir. Sahne, ekranda görüntülenen metinleri, resimleri, sesleri ve videoları içerisine alır.

Bir tasarımda bulunan sahneler, Sahne panelinde listelendikleri sırayla oynatılır. Animasyon çalıştırıldığında, Oynatma kafası bir sahnenin son karesine ulaşır ve sonraki sahneye geçiş yapar. Tasarım SWF dosyası olarak yayınladığında, her sahnenin Zaman Çizelgesi birleşerek SWF dosyasında tek bir Zaman Çizelgesi şeklini alır. Eklenmiş sahneler içerisinde bir sahne seçimi yapmak için sahne paneli kullanılır. Sahne panelini ekrana taşımak içinse Şekil 5.13'de gösterildiği gibi Pencere menüsünden Sahne seçeneği işaretlenir. Şekil 5.15'de ise sahne paneli gösterilmektedir.



Şekil 5.13 Pencere menüsünden sahne panelinin ekrana getirilmesi

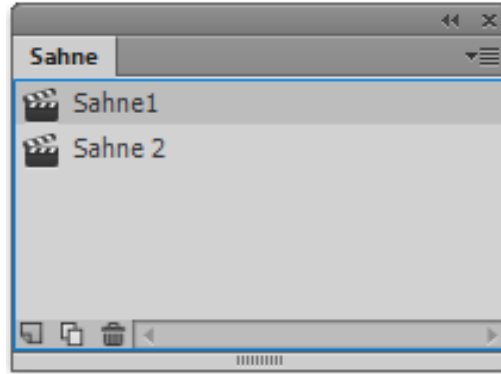


Şekil 5.14 Yeni bir sahne eklemek

Sahne Ekleme

Bir tasarıma sahne eklemenin iki yolu bulunmaktadır. Bunlar;

1. **Ekle (Insert)** Menüsünden **Sahne** seçeneğine tıklayarak yeni bir sahne ekleyebilirsiniz (Şekil 5.14). Bu işlem yapıldığında yeni sahne, mevcut sahnenin bulunduğu yerin hemen sonuna eklenmiş olarak görülür. Bu durum Şekil 5.15'te sahne panelinde gösterilmektedir.



Şekil 5.15 Sane panelinin görünümü

2. Diğer bir yol ise Sahne paneli üzerinden sol alt bölümde bulunan Yeni () simgesidir. Bu simgeye tıkladığınızda sahne paneli üzerine yeni bir sahne eklenmiş olur.

Sahne Silme

Sahne paneli üzerinde iken Çöp sepeti () simgesine tıkladığınızda seçili olan sahne silmiş olur.

Sahneyi Çoğaltma

Sahne paneli üzerinde iken Kopyala () simgesine tıkladığınızda bulunduğunuz Sahnenin aynısından bir kopya sahne daha oluşturur.

Sahnenin Sırasını Değiştirme

Sahne panelinde sahnenin adını sıralanmak istenen bölümden farklı bir konuma sürükleyerek yapılır.



Sahneleri, sahne paneli üzerinde yer alan simgeler yardımıyla yönetebilirsiniz.

KONTROL ET (CONTROL) MENÜSÜ

Kontrol Et menüsü filminizin nasıl çalışacağı, geliştirme ve test ortamında nasıl oynatılacağıyla ilgili çeşitli seçenekleri seçmenizi sağlar.

Kontrol Et	
Oynat/Çal	Enter
Geri Sar	Shift+,
Sona Git	Shift+.
Bir Kare İleri Git	,
Bir Kare Geri Git	.
Test Et	Ctrl+Enter
Filmi Test Et	▶
Sahneyi Test Et	Ctrl+Alt+Enter
Yayınlama Ön Belleğini Temizle	
Yayınlama Ön Belleğini Temizle ve Filmi Test Et	
✓ Döngüde Oynat/Çal	
Tüm Sahneleri Oynat/Çal	
✓ Basit Düğmeleri Etkinleştir	
Sesleri Kapat	Ctrl+Alt+M

Şekil 5.16 Kontrol Et Menüsü

Şekil 5.16’da gösterilen Kontrol Et Menüsü seçenekleri şunlardır;

- **Oynat/Çal:** Zaman cetveli üzerinde geçerli konumundan itibaren animasyonu oynatır.
- **Geri Sar:** Zaman çizelgesinde oynatma kafasını en baştaki kareye götürür.
- **Sona Git:** Zaman çizelgesinde oynatma kafasını en son kareye götürür.
- **Bir kare İleri Git:** Zaman cetvelinde mevcut konumuna göre bir kare ileri götürür.
- **Bir kare Geri Git:** Zaman cetvelinde mevcut konumuna göre bir kare geri götürür.
- **Test Et:** Geçerli filmin mevcut hâlini Flash’ın test ortamına aktararak swf olarak çalıştırır. **Filmi Test Et:** Geçerli filmin mevcut hâlini Flash’ın seçilen ortamına (Flash Player veya internet Explorer) aktararak swf olarak çalıştırır.
- **Sahneyi Test Et:** Seçilen sahnenin geçici bir versiyonunu Flash’ın test ortamına gönderir.
- **Yayınlama Ön Belleğini Temizle:** Önceki yayınlanan çalışmanın kaydedilmiş bölümlerini disk üzerinden siler.
- **Yayınlama Ön Belleğini Temizle ve Filmi Test Et:** Önceki test sonuçlarını silerek



“Geri Sar/Sona Git “ ve Oynatma Kafası, büyük animasyonların tasarlanmasında film kareleri üzerinde hareketle büyük kolaylık sağlar.

Filmin mevcut halinin oynatılmasını sağlar.

- **Döngüde Oynat/Çal:** Bu bölüm işaretlendiğinde Oynat/Çal veya Test Et gibi komutlarla başlatılan çalışma, sürekli kendini tekrar ederek çalışır. Bu bölüm işaretli değilse film kareleri bittiğinde test işlemi de biter. Aktif edildiğinde komutun yanında bir onay işareti görüntülenir. **Tüm Sahneleri Oynat/Çal:** Bu bölüm işaretlendiğinde Oynat/Çal veya Test Et gibi komutlarla başlatılan çalışma, ilk sahneden son sahneye kadar test işlemi devam eder. İşaret kaldırıldığında sadece ilgili sahne icra edilir ve test sonlandırılır.
- **Basit Düğmeleri Etkinleştir:** Basit Düğmeleri Etkinleştir seçildiğinde Düğmelerin etkinleştirildiğini belirtmek için komutun yanında bir onay işareti görüntülenir. Sahne Alanı üzerindeki düğme sembollerinin tepki vermesini sağlar.
- **Sesleri Kapat:** Test işlemi sırasında film kareleri üzerindeki ses dosyalarının dışarı ses vermesini engeller. Aktif edildiğinde komutun yanında bir onay işareti görüntülenir.



Özet

- Adobe Flash programı, Animasyon tasarımlarını zaman aralıklarına (karelere) bölerek kontrol eder. Zaman çizelgesinde, tasarımı düzenleyebilmek için Anahtar kare, Kare, Sahne ve katmanlar bölümlerini kullanılır.
- Anahtar kare, animasyon için bir nesnenin özelliklerinde değişiklik tanımlanabilen bir karedir. Bu kare, belgenizin bazı yönlerini denetlemek için ActionScript kodunda içerebilir.
- Zaman cetvelinde bazı kareler ara doldurmak için bulunur. Bu kareler ile yapılan animasyonda sadece iki anahtar karenin tasarımı yapılır. Aradaki karelerdeki değişimleri Flash programı kendisi hesaplar ve uygun şekilde içi hesaplanmış karelerle doldurur. Ara doldurma kareleri , hareket ve şekil olmak üzere iki çeşittir; Hareket ara doldurma karelerinde, nesnelerin bulundukları konum ve dönüş açıları kullanılarak animasyon oluşturulurken, Şekil koldurma karelerinde ise bir şekli başka bir şekle dönüştürülürken kullanılır.
- Zaman çizelgesinde değişiklik yapabileceğimiz Durum Çubuğu, Döngü Çubuğu, Test Çubuğu, Oynatma Kafası ve Katmanlar gibi alanlar bulunur. Durum Çubuğu, Animasyon hızıyla ilgili bilgi verir. Döngü Çubuğu, Animasyon içerisinde bulunan karelerin gösterimi bittikten sonra kaç kez daha gösterileceğini belirler. Test Çubuğu, Yapılan animasyon tasarımının test edilmesi amacıyla kullanılır. Geçerli Sahne, bir animasyon üzerinde zaman cetvelinde nasıl birden fazla film karesi varsa, aynı zamanda birden fazla sahnede bulunabilir. Ekranda görülen sahne, üzerinde çalışılmakta olan aktif sahneyi gösterir. Oynatım Kafası, zaman cetvelinde Sahne üzerinde gösterilen kareyi işaretlemek için kullanılır. Zaman cetveli, Bir belgenin içeriğini zaman içerisinde, sayısal numaralandırma yaparak kareler hâlinde gösterir ve animasyonun kontrol edilmesini sağlar. Katmanlar, semboller, sesler, diğer nesneler için bir ambalaj vazifesi görürler. Katmanlar üç farklı duruma sahiptir. Gizlenmiş mod sahnenin belirli bir kısmı üzerinde çalışırken, bazı katmanların içeriğini gizlemeye ihtiyaç duyulduğunda kullanılır. Kilitli mod, bir katmanı bu simgeye tıklayarak kilitlediğinizde içeriği görünür ama üzerinde düzenleme yapamazsınız. Dış Hat Modu, katmanınız bu moddayken içeriği sadece dış hatlar halinde görünür. Bu mod öğelerin kenarlarını kolayca ayırdetmenizi sağlar. Böylece şekilleri düzenlemeniz ve değiştirmeniz kolaylaşır.
- Sahneler, ekranda görüntülenen metinleri, resimleri, sesleri ve videoları içerisine alırlar. Bir tasarımda bulunan sahneler, Sahne panelinde listelendikleri sırayla oynatılır. Animasyon çalıştırıldığında, Oynatma kafası bir sahnenin son karesine ulaşır ve sonraki sahneye geçiş yapar.

DEĞERLENDİRME SORULARI

1. Aşağıdaki seçeneklerden hangisi kullanılarak “Zaman çizelgesine kare ekleme” işlemi yapılabilir?
 - a) Düzenle > Zaman Çizelgesi > Kareleri Kaldır
 - b) Ekle > Zaman Çizelgesi > Kare
 - c) Değiştir > Zaman Çizelgesi > Ana Kare
 - d) Ekle > Zaman Çizelgesi > Boş
 - e) Ekle > Zaman Çizelgesi > Anahtar Kare
2. Aşağıdaki seçeneklerden hangisi kullanılarak “Zaman çizelgesine katman ekleme” işlemini yapılabilir?
 - a) Ekle > Zaman Çizelgesi > Katman klasörü
 - b) Değiştir > Zaman Çizelgesi > Yeni sembol
 - c) Değiştir > Belge > Katman
 - d) Düzenle > Zaman Çizelgesi > Kareleri Kaldır
 - e) Ekle > Zaman Çizelgesi > Katman
3. Animasyonda bir değişikliğin olmasını veya bir eylemin ortaya çıkması istendiğinde zaman çizelgesinde o noktaya bir (.....) eklemek gerekir. Seçeneklerden hangisi nokta ile verilen boşluğu doldurmak için kullanılabilir?
 - a) Kare
 - b) Anahtar kare
 - c) Katman
 - d) Klasör
 - e) Ara doldurma karesi
4. Bir katmandaki çalışma bitirildiğinde üzerinde değişiklik yapmayı engellemek için seçeneklerde verilen hangi mod kullanılır?
 - a) Kilitle
 - b) Sakla
 - c) Göster
 - d) Grupla
 - e) Dış hat
5. Animasyon çalıştırıldığında, Oynatma Kafası bir sahnenin son karesine ulaşır ve sonraki (.....) geçiş yapar. Seçeneklerden hangisi verilen boşluğu doldurmak için kullanılabilir?
 - a) Kare
 - b) Sahne
 - c) Boş Anahtar kare
 - d) Anahtar kare
 - e) Katman

6. ) şeklinde verilen bölüm hangi seçenekte verilmiştir?
- a) Zaman cetveli
 - b) Sahne paneli
 - c) Özellikler paneli
 - d) Araç paneli
 - e) Kütüphane paneli
7. “Zaman cetveli üzerinde geçerli konumundan itibaren animasyonu oynatır.” Seçeneklerde verilen hangi komut bu işlevi gerçekleştirir?
- a) Oynat/Çal
 - b) Geri Sar
 - c) Bir kare İleri Git
 - d) Sona Git
 - e) Basit Düğmeleri Etkinleştir
8. Zaman çizelgesinin üzerinde animasyonları kontrol etmekte kullanılan birçok bölüm bulunur. Aşağıdaki seçeneklerden hangisi zaman çizelgesinde yer almaz?
- a) Döngü Çubuğu
 - b) Test Çubuğu
 - c) Oynatma Kafası
 - d) Durum Çubuğu
 - e) Yeni Sembol
9. ) Simgesi aşağıdakilerden hangisinin yapılmasını sağlar?
- a) Kopya sahne oluştur
 - b) Katman ekle
 - c) Anahtar kare ekle
 - d) Sahne ekle
 - e) Kareleri kopyala
10. Çalışma alanında bulunan bir çizim nesnesinin başka bir çizim nesnesine dönüştürmek amacıyla kullanılan ara doldurma karelerini aşağıdakilerden hangisi oluşturur?
- a) Hareket arası oluştur
 - b) Şekil arası oluştur
 - c) Anahtar kareyi temizle
 - d) Boş Anahtar kareye dönüştür
 - e) Anahtar karelere dönüştür

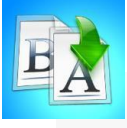
YARARLANILAN KAYNAKLAR

[1] Adobe Flash Pro CC, Yardımcı El kitabı. <http://helpx.adobe.com/tr/support>

[Erişim tarihi: 30 Ekim 2013].

[2]SCHULZE Pati.(2004) Macromedia Fireworks 8 Kaynağından Eğitim, Medyasoft Yayınları.

SES VE VIDEO İŞLEMLERİ



İÇİNDEKİLER

- Ses İşlemleri
- Adobe Flash İçerisine Seslerin Eklenmesi
- Kütüphane Panelinden Seslerin Yönetilmesi
- Video İşlemleri



HEDEFLER

- Bu üniteyi çalıştıktan sonra;
- Animasyon hazırlamak için kullanılan ses ve videoları seçebilecek,
- Video ve sesleri düzenleyebilecek,
- Tasarlanan animasyon içerisinde ses ve videoları kullanabileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

ANİMASYON TASARIMI

Okt.
Mustafa AKBUĞA

ÜNİTE 6



Adobe Fireworks programıyla animasyon tasarlamak mümkündür. Fakat bu animasyonlara Flash programındaki gibi ses dosyası eklemek mümkün değildir.



Yalnızca Flash Professional CC versiyonun'da sesleri içe aktarmak veya çalmak için QuickTime veya iTunes'un önceden yüklenmesine gerek yoktur.

Giriş

Adobe Flash animasyonlarının internette popüler olmasının en önemli sebeplerinden biri de sunmuş olduğu zengin ses ve video desteğidir.

İnternet ortamlarındaki tasarımların daha etkileşimli ve dikkat çekici olmasını sağlamak amacıyla onlara, ses, video ve Film Klipleri eklenir. Bu eklenen bileşenler çoğu zaman birtakım sorunları da beraberinde getirir. Kaliteli bileşenler internet kotasını gereksiz yere doldururken, bağlantı hızından dolayı tasarımın yavaş yüklenmesine ve cihazlarının kilitlenmesine neden olur. Kalitesiz görüntüler ve sesler ise kullanıcılar tarafından beğenilmez. Bu gibi sorunların üstesinden gelmek için ses ve video dosyalarının özelliklerini, yapısal ayarlarını ve Flash içerisine dâhil edilme biçimlerini iyi öğrenmek gerekir.

Bu amaç doğrultusunda bu ünite de ses ve video bileşenlerinin tasarıma nasıl dâhil edileceği, dosya biçimleri ve özellikleri hakkında bilgiler verilecektir.

SES İŞLEMLERİ

Adobe Flash içinde yapılandırılabilen iki türlü ses vardır. Bunlar olay sesleri ve akış sesleridir. *Olay sesi*, çalmaya başlamadan önce bilgisayara tamamen yüklenmiş olmalıdır. Bu ses animasyon içinde durduruluncaya kadar çalmaya devam eder. *Akış sesleri* ise kendisine ayrılan film karesi oynatılmaya başladığında çalmaya başlar ve kendisine ayrılan film karesinin sonuna gelindiğinde ise çalma işlemi ses dosyasının uzunluğuna bakılmadan sonlandırılır.

Adobe Flash aşağıda anlatılan ses dosya biçimlerini desteklemektedir.

Wave (.wav): İlk kullanılan ses dosyası biçimlerinden biridir. Genellikle, kalite bakımından ses CD'leriyle aynı olan sıkıştırılmamış ses dosyalarını saklamak için kullanılır. WAV biçimindeki bir dakikalık ses ortalama 10 MB hafıza gerektirir. CD'ler genellikle WAV biçiminde dijital ortama aktarıldıktan sonra bir ses dönüştürücüsü yardımıyla MP3'e dönüştürülebilir.

MPEG Layer-3 (.mp3): Dünyada kullanılan en yaygın ses biçimidir. MP3 biçiminde, dosya boyutu çeşitli kayıplı sıkıştırma yöntemleri kullanılarak küçültülür. Şu anda MP3 dosya boyutu açısından en iyi biçim olmamasına rağmen, ses kalitesi/boyut açısından rakipsizdir ve birçok cihaz tarafından desteklendiğinden dolayı çoklukla tercih edilen biçimdir.

FLAC (.flac): Yaygın olarak kullanılan kayıpsız bir biçimdir. Ses akışını değiştirmez ve bu biçimde kodlanan ses orijinalle aynıdır. Genellikle son teknoloji ses sistemleri tarafından kullanılır. Bu biçim normal cihazlar ve oynatıcılar tarafından sınırlı olarak desteklenmektedir. Bu tür cihazlarda uyum problemleri yaşanmaması için çalınmadan önce diğer biçimlere dönüştürülmesi önerilir. *Ogg Vorbis (.ogg, .oga)*: Çeşitli codeclarle ses kodlamasını destekleyen açık bir biçimdir. Sıkıştırma kalitesi MP3 ile kıyaslanabilir; fakat çeşitli ses oynatıcıları ve cihazlarının desteklemesi yönünden daha az yaygındır.

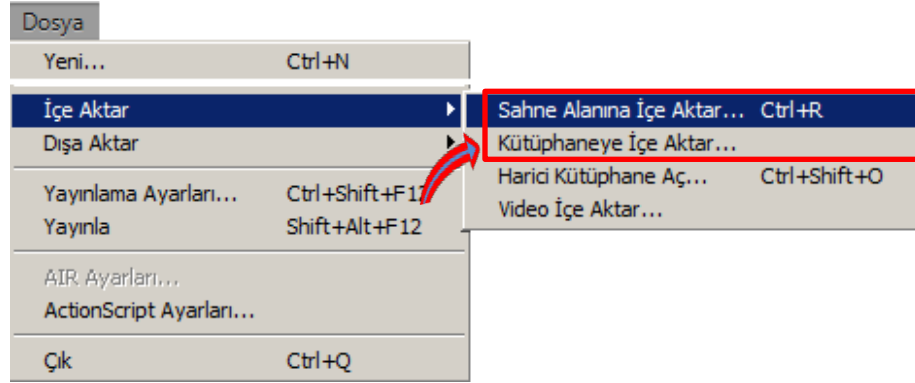
Adobe Flash içerisine bir ses dosyasını aktarma işlemi iki aşamada gerçekleşir. Birinci aşamada, Ses dosyası kütüphane paneline dâhil edilir. İkinci aşamada ise kütüphane panelindeki bu sesler bir film karesine aktarılır.

1. Kütüphane paneline seslerin eklenmesi

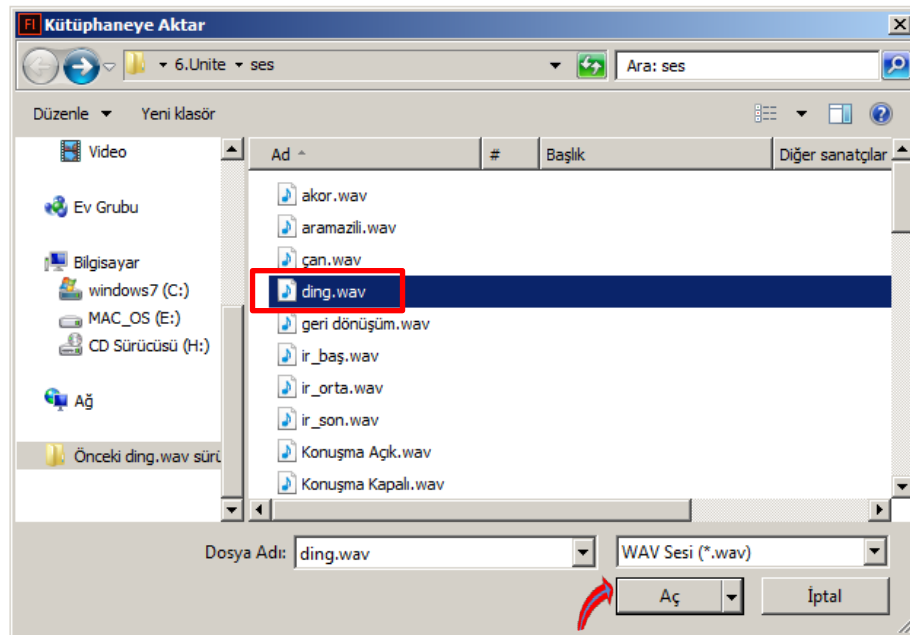
Bilgisayar diskinde yer alan bir ses dosyasını, kütüphane paneline eklemek için sırasıyla aşağıda adımlar takip edilir;

Adım 1: Şekil 6.1’de gösterilen Dosya> İçe Aktar Kütüphaneye İçe Aktar veya Sahneye İthal Et komut seçeneklerinden birini seçin (Resim dosyalarını aktarıırken sahne ve kütüphane olmak üzere iki ayrı yeri seçen bu komutlar ses dosyalarında ise sadece kütüphane paneli içerisine aktarma işlemi gerçekleştirir).

Adım2: Şekil 6.2’de gösterildiği gibi İçe Aktar iletişim kutusundan istenen ses dosyası bulun ve Aç butonunu tıklayın.



Şekil 6.1 Ses dosyalarının Adobe Flash programı içerisine alınma işlemi

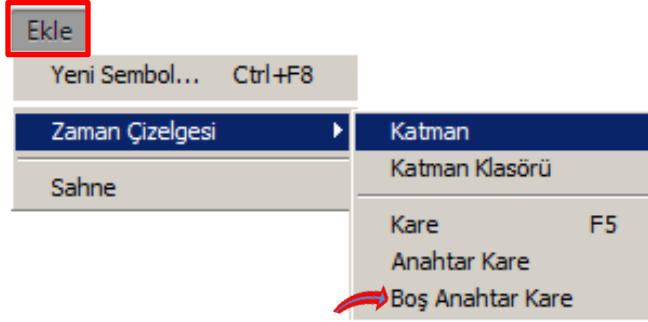


Şekil 6.2 Ses dosyalarının seçilmesi

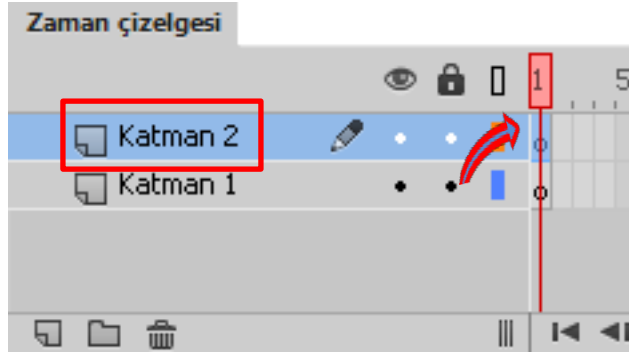
2. Kütüphane panelindeki seslerin film karesine eklenmesi

Kütüphane panelinden, Zaman çizgisine ses eklemek için sırasıyla aşağıda adımlar takip edilir;

Adım1: Şekil 6.3 'de gösterildiği gibi *“Ekle > Zaman Çizelgesi > Katman”* bölümlerini seçin. Bu işlem neticesinde zaman çizelgesine ses dosyasında kullanılmak üzere Yeni bir katman oluşturulur. Şekil 6.4'teki gibi oluşturulan katman üzerinde bir adet boş film karesi bulunmaktadır. Bu film karesini seçin.

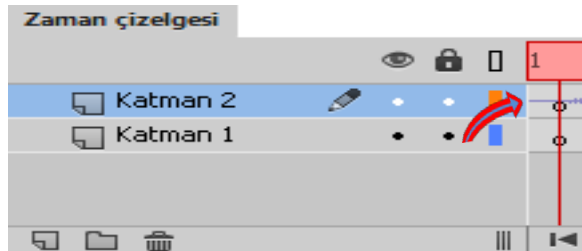


Şekil 6.3 Zaman çizelgesine katman ekleme



Şekil 6.4 Zaman çizelgesine eklenen katman ve boş film karesi

Adım 2: Yeni ses katmanını seçiliyken, sesi Kütüphane panelinden Sahne Alanı'nın üzerine sürükleyin. Bu sürükleme işlemi sonrasında ses seçili katman üzerindeki boş film karesine eklenir. Şekil 6.4'te gösterilen kare küçük olduğundan ses dosyası görülmeyebilir. Bu durumda zaman paneli özelliklerinden kare görüntüleme özelliğini *“Büyük”* görüntüleme moduna alarak Şekil 6.5'te kırmızı okla gösterilen ses dosyası görülebilir (Bu işlemi hatırlamak için 5. üniteadaki zaman çizelgesi özelliklerine bakınız).



Şekil 6.5 Katman 2 'ye eklenen ses dosyasının tek karedeki görünümü



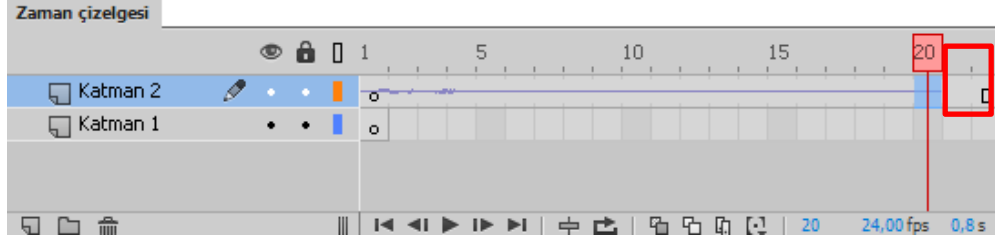
Tasarım içerisine eklenecek her ses dosyası için bir tane de katman eklemek seslerin kontrolünü kolaylaştırır.



Tasarım içerisinde
*.wav ses dosya
uzantısı yerine *.mp3
uzantılı dosyaları
kullanmak dosya
boyutunu azaltır.



Katman içindeki film
karelerine eklenen ses
dosyalarının her birine
özellikler panelindeki
efektlerden biri
uygulanabilir.

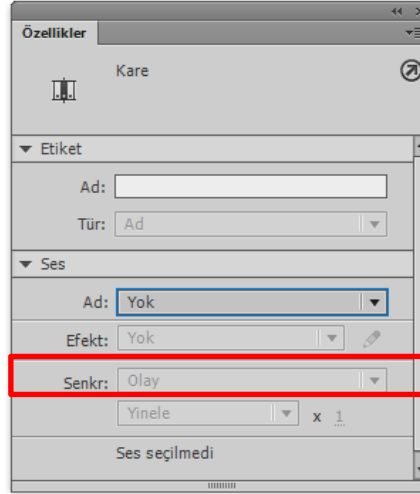


Şekil 6.6 Katman 2 'ye eklenen ses dosyasının süresi kadar kare eklenmiş hâli

Adım 3: Eklenen ses dosyasının tam olarak görüntülenmesi ve çalınması için Şekil 6.6'da gösterildiği gibi ses dosyasının çalma süresi kadar ilgili katmana kare eklemek gereklidir. Şekil 6.6'da gösterilen ses dosyası 0.8 sn süren ve 20 filmkarelik yer kaplamıştır. Zaman cetveline bakılırsa 21 ve 22 karelerde ses dosyasının etkisini yitirdiği görülür.

Adım 4: Katman üzerine eklenen ses dosyasının özelliklerini değiştirin.

"Pencere > Özellikler " seçeneğini seçin ve Özellik denetçisini genişletmek için sağ alt köşedeki oku tıklayın. Şekil 6.7'deki film karesinin özelliklerine bakılınca ses bölümünün Ad kısmında "Yok" görülür.

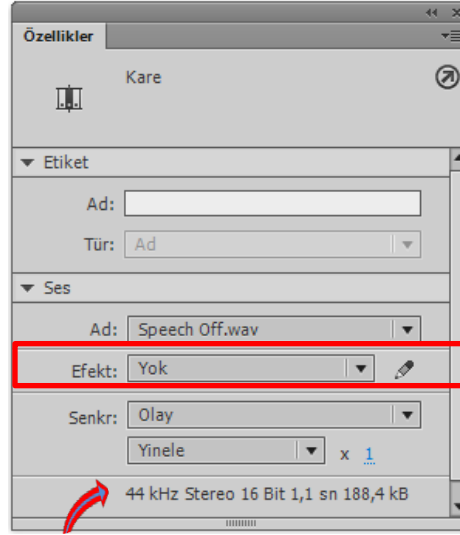


Şekil 6.7 Katman 1'deki film karesinin özellikleri

Bu film karesinin içerisinde ses dosyası olmadığını gösterir. İstenirse bu kısımdaki açılır menüden kütüphanedeki ses dosyalarından birisi seçilip, film içerisinde alınabilir. Etiket kısmında yer alan Ad bölümü ActionScript kodlarıyla sesin üzerinde kontrol komutlarını çalıştırmak için kullanılır.



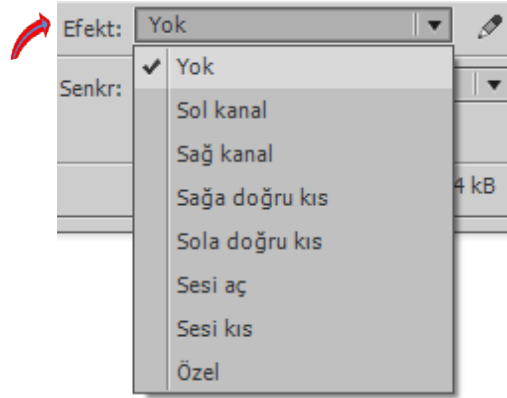
Ses dosyalarında kullanılan Efektler bilgisayar donanımı tarafından desteklenmelidir. Ses kartı olmayan veya sürücüsü (driver) yüklenmemiş bilgisayarlarda ses içeren animasyonlar çalışmaz.



Şekil 6.8 Katman 2'deki film karesinin özellikleri

Şekil 6.7'deki görüntünün ses bölümünün Ad kısmında atanan ses dosyası görülmektedir. Şimdi bu Ses bölümünde yer alan kısımları ayrıntılı inceleyelim;

- **Ad**, kütüphane panelinde yer alan ses dosyasının isim ve uzantısını gösterir.
- **Efekt**, katman üzerinde yer alan ses dosyasına efekt seçmek amacıyla kullanılır. Bu bölümde aşağıdaki Şekil 6.9'da gösterilen seçeneklerden biri ayarlanabilir.



Şekil 6.9 Özellikler panelinde yer alan efekt seçimi

- **Yok**: Ses dosyasına hiçbir efekt uygulamaz. Önceden uygulanmış efektleri kaldırmak için bu seçeneği seçin.
- **Sol Kanal**: Sesi sadece sol kanalda çalar. **Sağ Kanal**: Sesi sadece sağ kanalda çalar.
- **Sağa doğru kıs**: Sesi sol kanaldan diğerine kaydırır.
- **Sola doğru kıs**: Sesi sağ kanaldan diğerine kaydırır.



Ses dosyasını ana Zaman Çizelgesi'nde 1. kare'den başka bir kareye yerleştiriyorsanız, istenmeyen seslerin önlenmesi için Durdur seçeneğini mutlaka seçmeniz gerekir.

- **Sesi Aç:** Bir sesin ses düzeyini, sesin süresi boyunca giderek artırır.
- **Sesi Kıs:** Bir sesin ses düzeyini, sesin süresi boyunca giderek azaltır.
- **Özel: Zarfı** Düzenle'yi kullanarak ses için özel giriş ve çıkış noktaları oluşturmanızı sağlar.
- **Senkr:** Eklenen sesin zaman çizelgesindeki film kareleriyle olan ilişkisini düzenler. Senkr açılır menüsünden Şekil 6.10'da gösterilen senkronizasyon seçeneklerinden biri ayarlanabilir.



Şekil 6.10 Özellikler panelinde yer alan Senkr seçimi

- **Olay Sesi:** bir olayın oluşuyla birlikte çalmaya başlar. Olay sesinin yerleştirildiği ilk film karesi görüntülendiğinde ses dosyası da çalmaya başlar ve ses dosyasının çalma süresi bitene kadarda çalmaya devam eder.



Örnek

- Kullanıcının birkaç defa düğmeyi tıklaması ya da Oynatma kafasının sesin başlangıç anahtar karesinden birkaç defa geçmesiyle birbirine karışan olay sesleri çalmaya başlar.

- **Başlat,** Ses zaten çalınıyorsa sesin yeniden bir daha çalınmasını engeller.
- **Durdur,** Belirtilen sesi susturur.
- **Akış Sesi,** Özellikleri ayarlanan sesi, bir web sitesinde çalmak için olay seslerinden farklı olarak, SWF dosyası oynatılmayı bırakırsa akış sesleri durur. Ayrıca, bir akış sesi asla bulunduğu karelerin uzunluğundan daha uzun bir süre çalamaz.

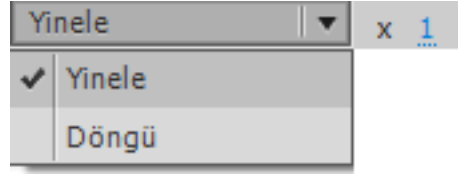


- , Sesin kaç döngü yapacağını belirler.

Şekil 6.11'de gösterilen Yinele ve Döngü olmak üzere iki seçeneği vardır. Tekrar sayısını belirtmek için Mavi renkli 1 yazan Alana sayısal bir değer girilmelidir. Bu sayısal değer kadar çalma işlemi tekrarlanır. Döngü seçeneği seçildiğinde sesin çalma işlemi sürekli kendini tekrarlar.



Sınırlı RAM'iniz varsa,
tasarımlarda ses
kliplerinizi kısa tutun.



Şekil 6.11 Tekrarla seçenekleri

- **44 kHz Stereo 16 Bit 0,8 sn 147,5 kB**, Animasyon içerisine dâhil edilmiş ses dosyasının teknik özelliklerini gösterir. Bu teknik özellikler aşağıdaki örnekte açıklanmıştır.



Örnek

- Yukarıda verilen teknik bilgiler, ilgili sesin 44kHz frekans sıklığında 16 bit sayı sistemine göre örneklenmiş, sağ ve sol hopperlör tarafından çalınabilen 0.8 sn uzunluğunda ve 147,5 kB'lık yer kapladığını belirtir.

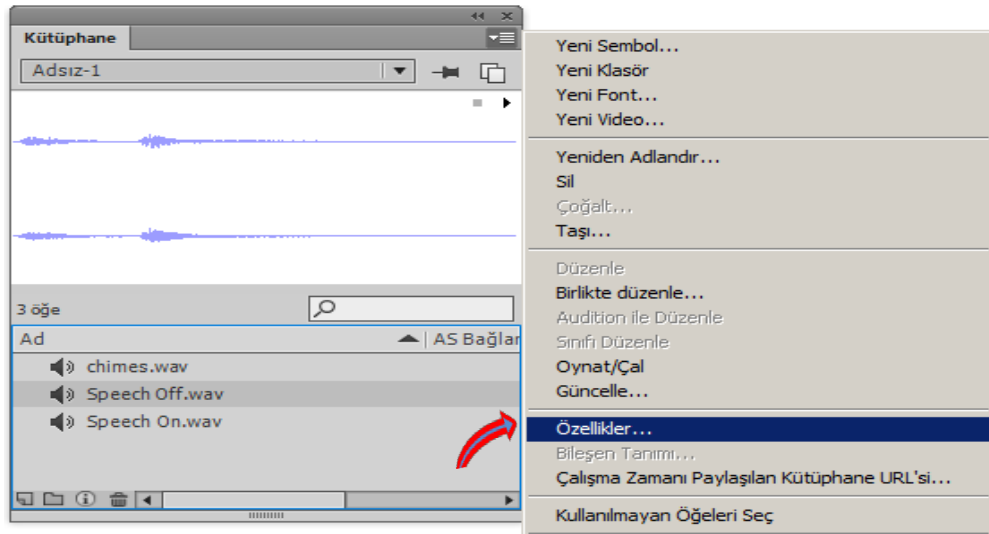


Ses dosyalarını Zaman çizelgesine atmadan önce kütüphane paneli üzerinden test edebilirsiniz.

Kütüphane Panelindeki Seslerin Yönetilmesi

Adobe Flash içerisinde kullandığınız sesler, tasarımın yayınlanması sırasında belli seçenekler doğrultusunda sıkıştırılırlar. Seslerin sıkıştırılması tasarım dosyasının daha hızlı çalışmasını ve az yer kaplamasını sağlar.

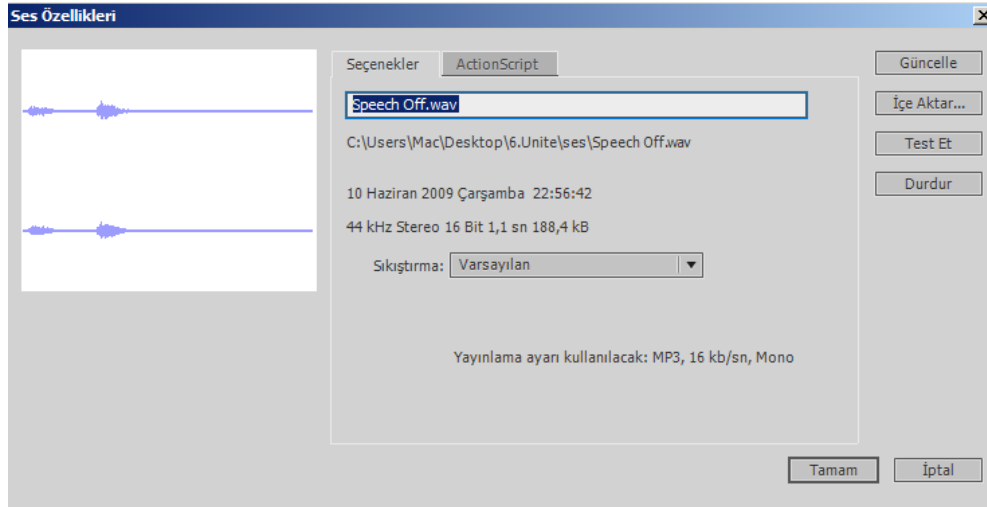
Adobe Flash içinde yer alan her bir ses için farklı sıkıştırma yöntemleri kullanılabileceği gibi tüm seslerin aynı sıkıştırma yöntemi ile de sıkıştırılması sağlanabilir. Bu ses düzenleme seçeneklerinin kullanılabilmesi Şekil 6.12'de kırmızı okla gösterilen menüden, özellikler seçeneğinin seçilmesiyle sağlanır. Bu bölümün açılmasıyla ekrana Şekil 6.13'te gösterilen düzenleme seçenekleri gelir.



Şekil 6.12 Kütüphane panelinden ses dosyasının özelliklerinin gösterilmesi



Ses dosyası olarak kullandığınız dosya sıkıştırılmış olsa bile animasyon dışarı aktarılırken yapılan ayarlara göre tekrar sıkıştırılabilir.



Şekil 6.13 Ses dosyasının yayınlanma esnasında kullanılacak sıkıştırma özellikleri
Sesin özellikleri panelinde Şekil 6.13'te gösterilen seçenekler yardımıyla ses dosyasının aşağıdaki özellikleri ayarlanabilir.

- **Güncelle:** Sesin disk üzerindeki kaynağından Flash'ın içerisine yeniden dâhil edilmesini sağlar. Eğer ses harici bir ses kaynağından alınmış veya kaynağıda değişikliğe uğramışsa bu seçenek kullanılmalıdır.
- **İçe Aktar:** Mevcut ses dosyasının yerine farklı bir ses dosyasının diskten yüklenmesini sağlar.
- **Test Et:** Ses dosyasının dinlenilmesini sağlar.
- **Durdur:** Ses dosyasının çalma işlemini durdurur.
- **Sıkıştırma:** Sesin sıkıştırma yöntemini belirler. Varsayılan ADPCM, MP3, Ham, konuşma sıkıştırma yöntemlerini destekler. Genellikle Mp3 seçeneğinin seçilir.

VIDEO İŞLEMLERİ

Flash videoları internet üzerinde gün geçtikçe yaygınlaşan uygulamalardan biridir. Yaygın kullanıma örnek olarak YouTube ve Dailymotion siteleri gösterilebilir. Bu siteler içerisinde izleyeceğimiz videoların büyük çoğunluğu Flash ile hazırlanan videolardır.

Tasarlanan animasyonlar içerisine video ekleyerek animasyonları daha ilgi çekici hâle getirebiliriz. Adobe Flash içerisine iki farklı şekilde video eklenebilir: kısmi denetimli ve tam denetimli. Kısmi denetimli video ekleme işlemi ile video dosyası, Adobe Flash dosyası içerisine dâhil edilmez sadece video üzerine çerçeve eklenir. Bu çerçeve ile harici video dosyasının oynatılması ve kontrol edilmesi (videoyu ileri/geri alınma, sesi açma/kapama ve görüntüyü tam ekran yapma vb.) sağlanabilir.

Bu yöntemle *.mpeg, *.flv, *.mp4, *.mov, *.3gp gibi dosya formatları Adobe Flash içerisine eklenebilir. Tam denetimli video ekleme işleminde sadece *.flv dosya formatına sahip video



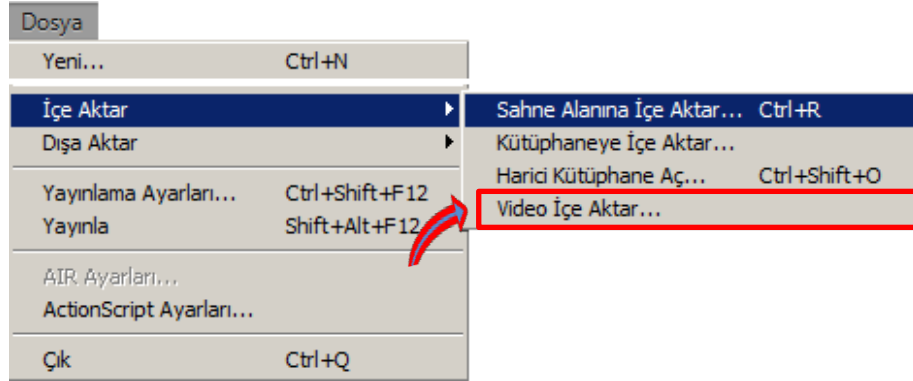
eklenebilir. Bu yöntem ile video dosyası film karelerine ayrılarak Adobe Flash içerisine dâhil edilir ve her bir film karesi üzerinde flash araçları ve kodları (ActionScript komutları) kullanılarak istenen değişiklikler yapılabilir. Uzun süreli video dosyalarının Adobe Flash içerisine eklenmesinde SWF dosyasının boyutu daha küçük olması için kısmi denetimli video ekleme yöntemi tercih edilir.

Bu yöntemler aşağıda detaylı bir şekilde anlatılmaktadır.

Kısmi denetimli video ekleme işlemi

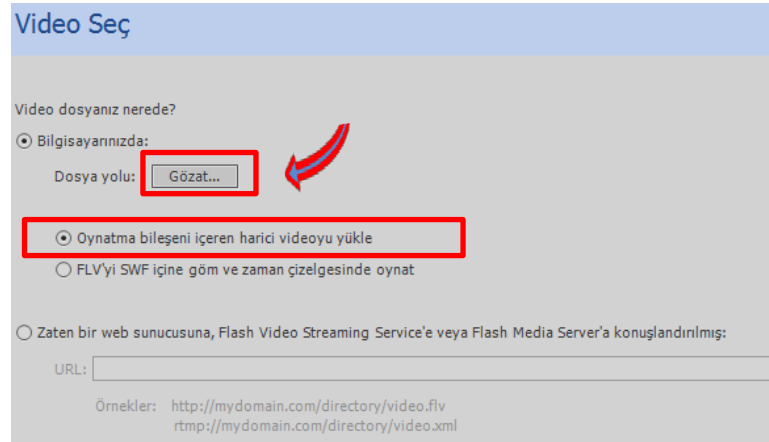
Adobe Flash içerisine video eklemek için

1.Adım: Dosya > İçe Aktar > Video İçeri Aktar komutu verilir. (Şekil 6.14).



Şekil 6.14 Video dosyalarının Adobe Flash programı içeresine alınması

2. Adım: Şekil 6.15'te gösterildiği gibi açılan Video Seç iletişim kutusundan Yüklencek video dosyasının nerede olduğunu belirtiniz. Şekil 6.14'te görüldüğü üzere videolar ya kendi bilgisayarımızda veya ağ üzerindeki başka bir bilgisayarda olabilir.



Şekil 6.15 İçeri aktarılacak video dosyasının seçilmesi

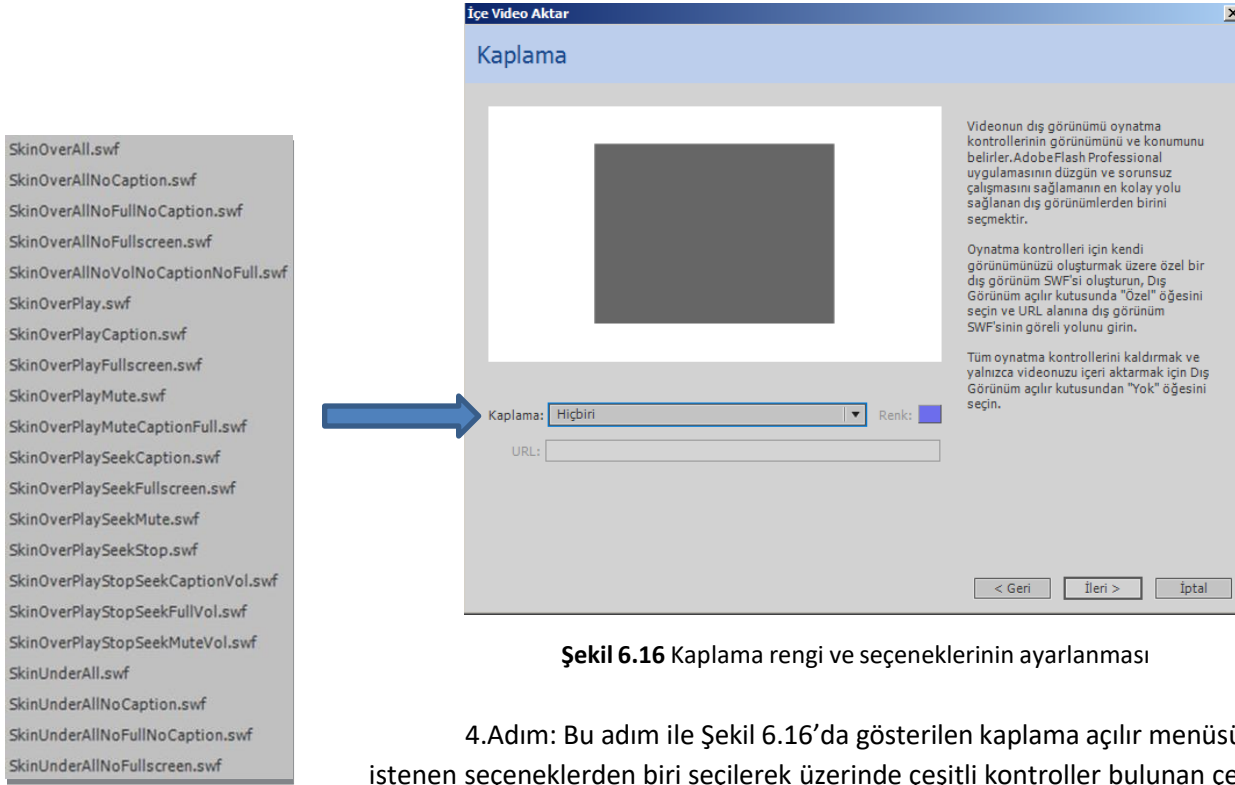
Şayet yüklenecek video dosyası kendi bilgisayarınızda ise Şekil 6.15'te gösterilen "**Bilgisayarınızda**" seçeneğini seçiniz.

Eğer video dosyaları ağ üzerinde başka bir bilgisayar üzerinde ise bu

durumda “*Zaten bir web sunucusuna konuşlanmış*” seçeneğini işaretleyiniz. Bu durumda ise Flash Media Server’a (Web üzerinde uzaktaki bir ana makinede kurulu hizmet programı) “rtmp”ya da “http” protokolleri ile bağlanılarak okunabilecek video dosyalarının adresini girmenizi gerekir.

Aşağıdaki adımlar bu adımda “Bilgisayarınızda” seçeneğini seçtiğiniz varsayımı ile devam edecektir.

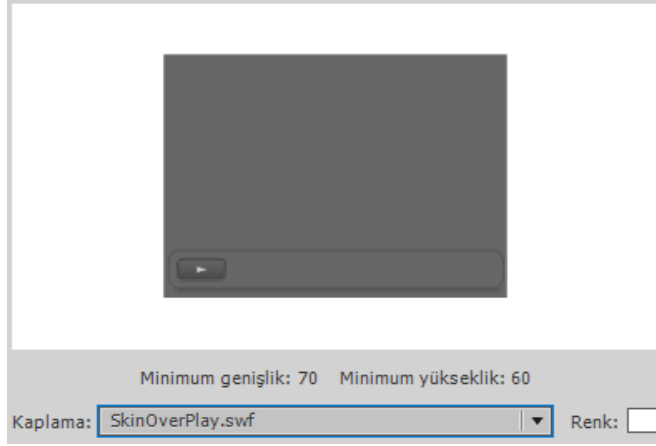
3.Adım: Şekil 6.14’te gösterilen “Oynatma bileşeni içeren harici videoyu yükle” seçeneğini seçiniz. Bu durumda Gözet seçeneği ile video dosyalarının konumunu gösteriniz ve İleri butonuna tıklayınız. Bu işlem sonrasında Şekil 6.16’daki görüntü ekrana gelecektir. Bu ekran ile Kaplama Videonun dış görünümü ve oynatma kontrollerinin görünümü belirlenir.



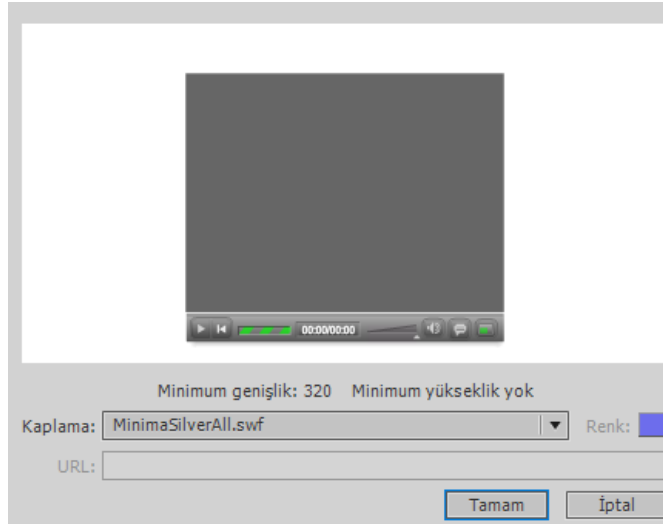
Şekil 6.16 Kaplama rengi ve seçeneklerinin ayarlanması

4.Adım: Bu adım ile Şekil 6.16’da gösterilen kaplama açılır menüsünden istenen seçeneklerden biri seçilerek üzerinde çeşitli kontroller bulunan çerçeve eklenir.

Örnek olarak kaplama seçeneklerinden “SkinOverPlay” seçildiğinde ve Tamam tıklandığında Şekil 6. 17’de gösterilen çerçeve eklenir. Bu çerçeve ile Video yalnızca oynatılıp durdurulabilir. Şayet kaplama seçeneklerinden “MinimaSilverAll.swf” seçilirse bu durumda Şekil 6.18’de gösterilen çerçeve eklenir. Bu çerçeve ile video dosyasının oynatılıp durdurulmasının yanında ses seviyesinin azaltılıp/artırılması da sağlanabilir.



Şekil 6.17 Kaplama olarak “SkinOverPlay. swf” çerçevesi ve özellikleri



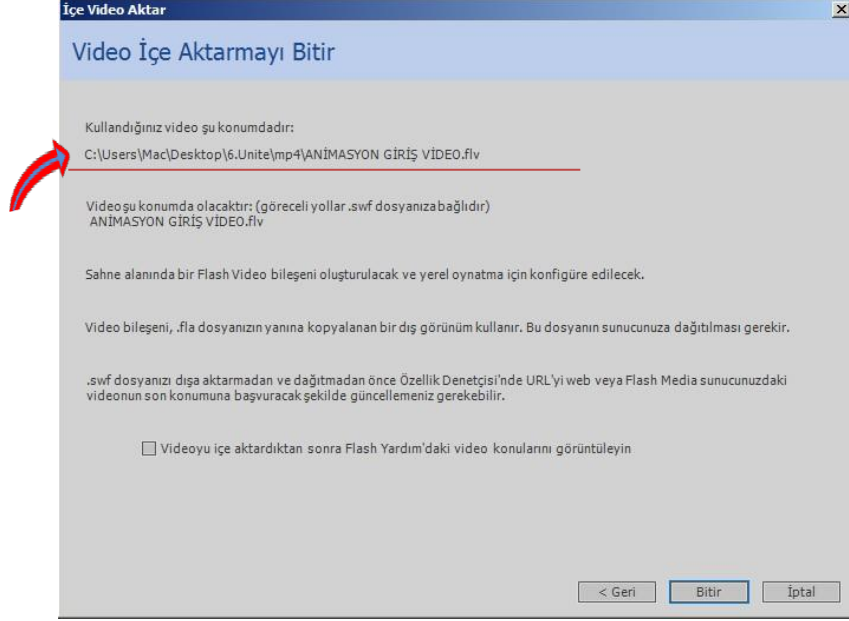
Şekil 6.18 Kaplama olarak “MinimaSilverAll. swf” çerçevesi ve özellikleri

Şekil 6.16’nın yanında gösterilen film çerçevelerinden, kaplama isimlerine bakarak istenen kontrolleri üzerinde bulunduran çerçevelerden biri seçilir ve ileri *komutu* verilir. Şekil 6.19’ile gösterilen iletişim penceresinden videonun kayıtlı olacağı yer not alınarak *Bitir komutu* verilir ve böylece video çerçevesinin sahneye ve aynı zamanda kütüphane paneline yerleşmesi sağlanır.

Bu kütüphane paneline eklenen nesneye FLVPlayback bileşeni adı verilir. Bu FLVPlayback bileşeni başka sahnelerde kullanılmak istenirse FLVPlayback bileşen parametreleri değiştirilmesi gerekir. Bu işlem aşağıda detaylı olarak anlatılmaktadır.



Tasarım içerisinde kullanılan video dosyalarının konumu değiştirilirse animasyon içerisindeki videoların kaynak bilgileri güncellenmelidir. Konumu değişen videolar çalışma içerisine gömülmemişse animasyonda oynatılamaz.



Şekil 6.19 Üzerine video kontrolleri eklenen dosyasının kayıt yolu

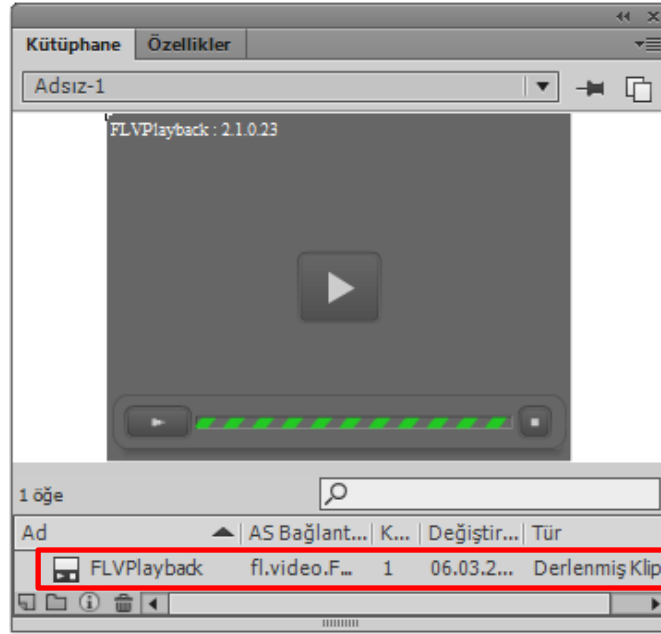
FLVPlayback bileşen parametrelerini değiştirme

Çalışma sahnesine yerleştirilen video dosyasıyla birlikte seçilen kaplama şekline göre kütüphane üzerine birde FLVPlayback bileşeni eklenir. Bu bileşen sayesinde tasarım içinde yer alan başka bir film karesi ya da sahne üzerinde yeni bir ayar yapmadan video dosyalarını kullanabiliriz.

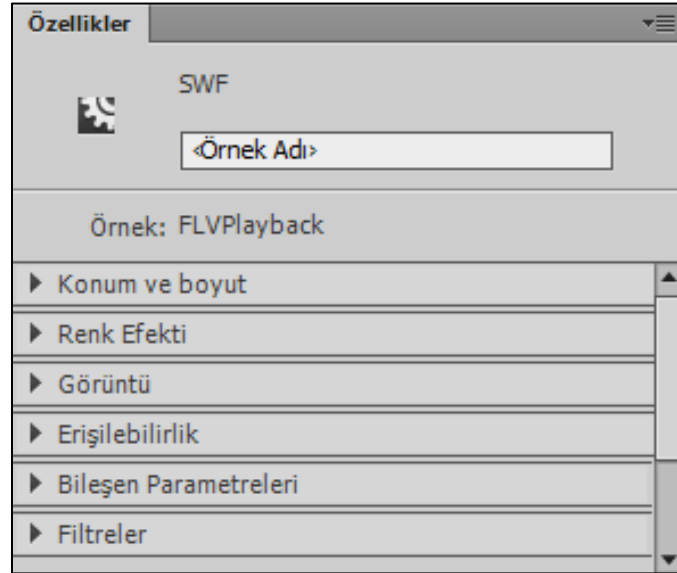
Kütüphane üzerinden eklenen FLVPlayback bileşeni başlangıçta sahne üzerine eklenen çerçeveli video dosyası gibi video gösteremez. Aşağıda verilen işlemler takip edildiğinde kütüphaneden eklenen FLVPlayback bileşeni de başlangıçta sahne üzerine eklenen çerçeveli video dosyası gibi video gösterebilir.

İşlem 1: Pencere menüsünden> Kütüphane paneli (Ctrl+L) açıldığında FLVPlayback bileşeni Şekil 6.20'deki gibi görülür. Kütüphane panelinde FLVPlayback bileşeni sürüklenerek çalışma sahnesi üzerine bırakılır.

İşlem 2: Sahne alanına eklenen FLVPlayback bileşenini seçilerek Şekil 6.21'de gösterildiği gibi Özellikler paneli açılır (Pencere > Özellikler).

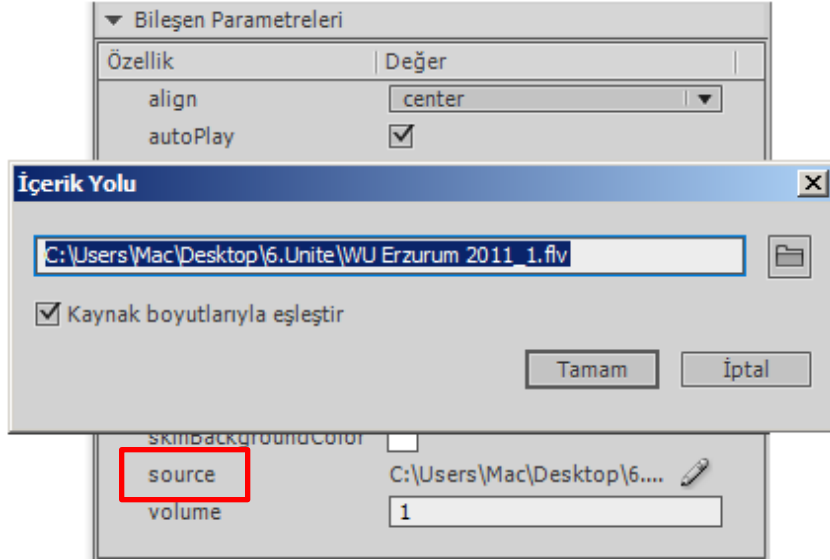


Şekil 6.20 Kütüphane paneline eklenen derlenmiş klip

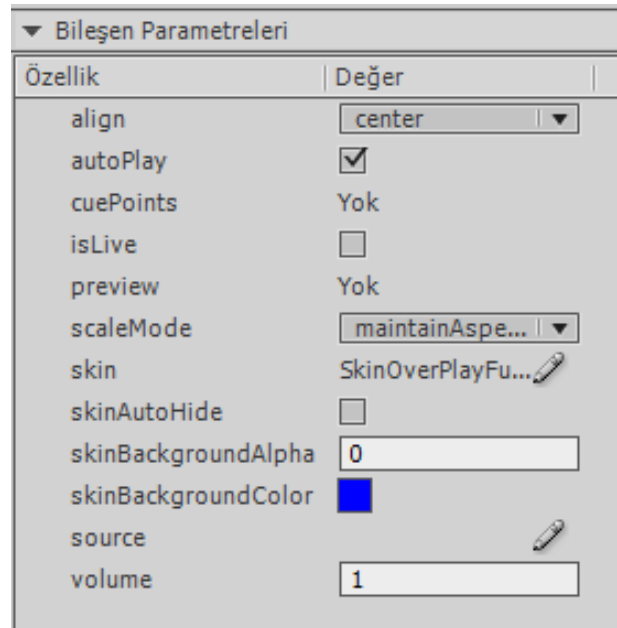


Şekil 6.21 Sahne üzerindeki FLVPlayback bileşenin özellikleri

İşlem 3: Şekil 6.21’de gösterilen Bileşen Parametreleri özelliklerini açarak “Source” parametresini Şekil 6.22’de gösterildiği gibi video dosyasının bulunduğu konuma ayarlayınız ve Tamamı’ tıklayın. Bu işlem sonrası FLVPlayback bileşeni artık video oynatabilir.



Şekil 6.22 Kütüphaneden sürüklenerek alınmış çerçevenin içerik yolunun güncellenmesi



Şekil 6.23 Çalışma alanı üzerindeki FLVPlayback bileşenin parametreleri

Şekil 6.23'te gösterilen FLVPlayback bileşenin "Source" parametresi dışında aşağıdaki parametreleride istenirse ayarlanabilir.

- **Align**, videonun yeniden boyutlandırılması durumunda ekrandaki konumunu kontrollere göre hizalar.
- **Autoplay**, FLV veya F4V dosyasının nasıl oynatılacağını belirleyen Boolean değeri. True ise, video yüklendiğinde hemen oynatılır. False ise, ilk kareyi yükler ve duraklar. Varsayılan olarak true değerindedir.
- **Cuepoints**, Yok seçeneği işaretlidir. Noktalar eklendiğinde işaret noktaları, videodaki belirli noktaları, grafik veya metinle senkronize etmenizi sağlar.



Tasarımda seçilen FLVPlayback bileşenin üzerindeki kontrollerin video gösterilirken gizlenmesi için Skin AutoHide özelliğini kullanılır.

- **IsLive**, true ise, videonun Flash Media Server'dan canlı olarak akışa alındığını belirten Boolean değeridir. Varsayılan olarak false değerindedir.
- **Preview**, Yok seçeneği işaretlidir. Noktalar eklendiğinde işaret noktaları, videodaki belirli noktaların videoda görüntülenmesini sağlar.
- **ScaleMode**, kaynak videonun en boy oranını seçimini sağlar. FLVPlayback bileşeninin kendisi yeniden boyutlandırılmaz.
- **Skin**, Kaplama Seç iletişim kutusunu açan ve bileşen için yeni bir kaplama seçmenize izin veren bir parametredir. **Skin AutoHide**, Kaplama üzerindeki video araçlarını video gösterime başladığında otomatik olarak gizler. Üzerine tıklandığında yeniden gösterir.
- **SkinBackgroundAlpha**, Kaplama üzerindeki video araçlarının rengini şeffaflaştırır. 0 ile 100 arasında bir değer alır. 0 -> değeri control çerçevesini şeffaf yaparken, 100 -> değeri ise çerçeve rengini video üzerinde tam olarak gösterir.
- **SkinBackgroundColor**, Kaplama üzerinde yer alan video araçlarının arka plan rengini ayarlar.
- **Source**, video dosyasının bilgisayarınızdaki adını ve konumunun seçilmesini sağlar.
- **Volume**, ses düzeyini ayarlayacağınız maksimum düzeyin yüzdesini temsil eden, 0'dan 100'e kadar bir sayıdır.



Büyük bir video dosyası gömerseniz, SWF dosyasının, bu dosyanın tamamını indirmesi ve oynatmaya başlaması uzun zaman alabilir.

Tam denetimli video ekleme işlemi

Bu yöntemle sadece *.FLV video dosyası, Adobe Flash içerisine yüklenebilir. Bu seçenek ile video dosyası, küçük film kareleri şeklinde paçalara ayrılır ve her bir parça tek tek görebileceğimiz şekilde Zaman Çizelgesi'ne yerleştirilir. Bu işlem sonunda oluşan SWF dosyasının boyutu k

ismi denetimli video ekleme işlemi ile elde edilen dosya boyutundan çok daha büyüktür.

Zaman Çizelgesi'nde her video karesi bir kare ile temsil edildiği için, video klip ile SWF dosyasının kare hızı aynı olmalıdır.

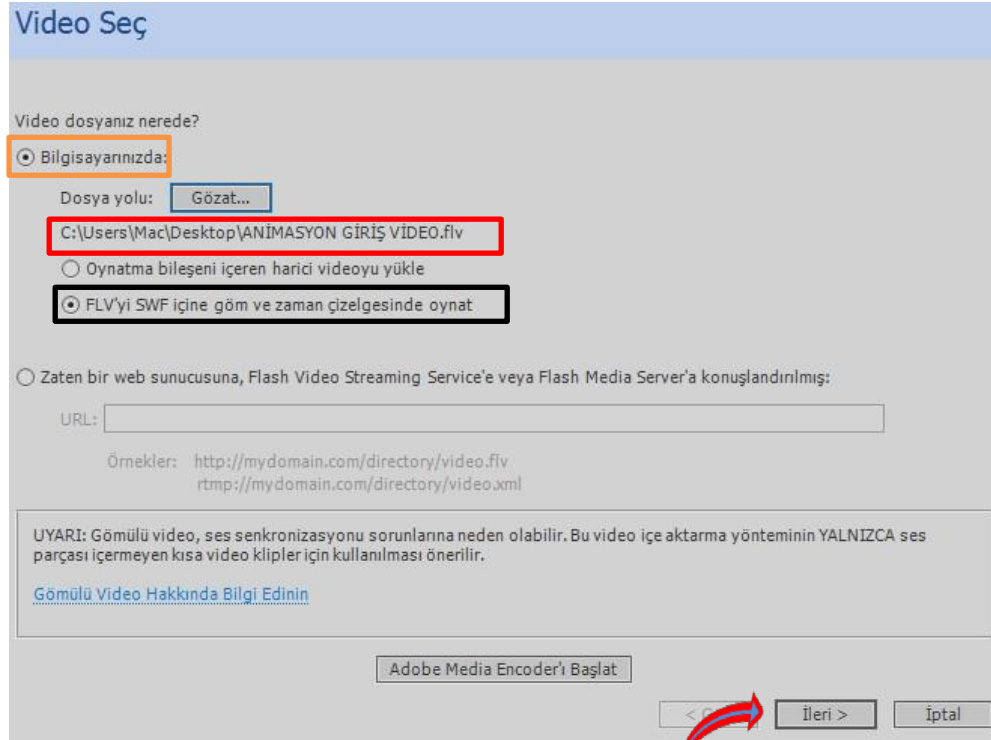
Bilgisayar diskinde yer alan FLV uzantılı bir video dosyası, aşağıdaki adımlar takip edilerek Adobe Flash içerisine tam denetimli olarak eklenebilir.

1. **Adım: Dosya (File) > İçe Aktar (Import) > Video İçeri Aktar (Import Video...)** komutu verilir. (Şekil 6.14).
2. **Adım:** Şayet yüklenecek *.flv uzantılı video dosyası kendi bilgisayarınızda ise Şekil 6.24'te turuncu dikdörtgen içerisinde gösterilen "**Bilgisayarınızda**" seçeneğini seçiniz.
3. **Adım:** Şekil 6.24'te siyah dikdörtgen içerisinde gösterilen "FLV'yi SWF

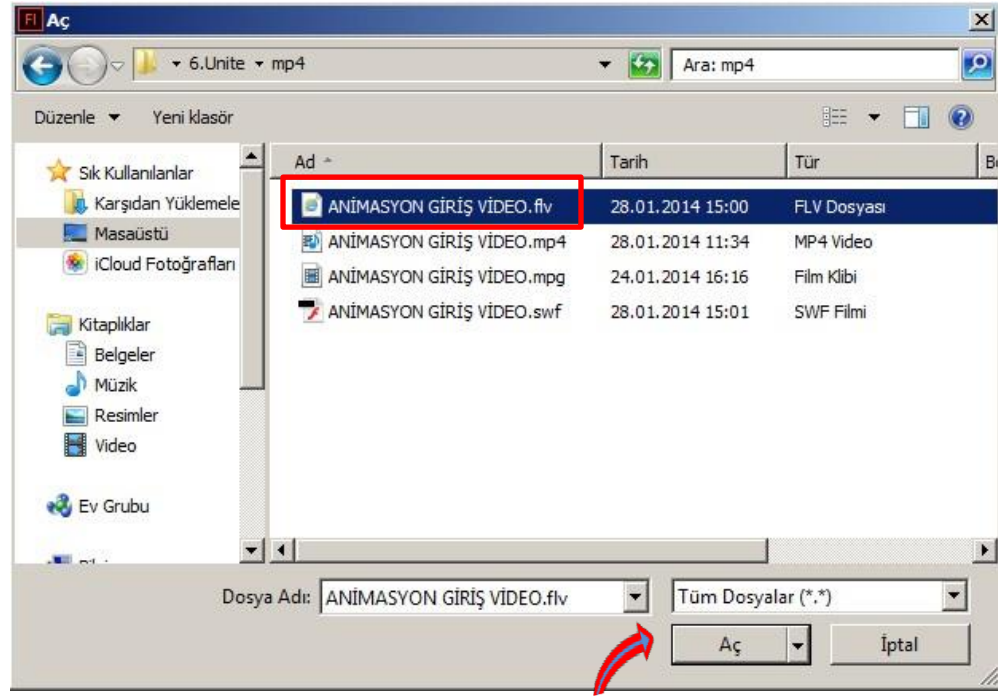
içine göm ve zaman çizelgesinde oynat “ seçeneğini seçiniz.

4. **Adım:** Şekil 6.24’te gösterilen “Gözet” bölümünü kullanarak Şekil 6.25’te Aç iletişim penceresi üzerinden bir *.FLV dosyası seçerek Aç butonuna basın böylece Şekil 6.24’te kırmızı dikdörtgen içerisinde gösterilen bir video dosya yolu elde edilecektir. Bu şekil üzerindeki kırmızı ok ile işaretli ileri butonunu tıklayın.

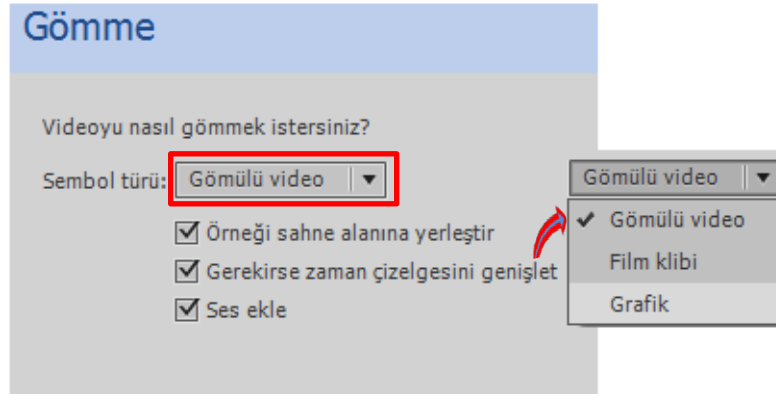
- **Adım:** Şekil 6.26’da gösterilen SWF dosyasında, videoyu birlikte gömeceğiniz sembol türünü seçin. **Gömülü Video**, Zaman Çizelgesi’nde doğrusal oynatma için kullanıyorsanız, videoyu Zaman Çizelgesi’ne içe aktarmak en uygun yöntemdir.
- **Film Klibi**, Videonun bir film klibinin içerisine yerleştirilmesini sağlar. Film Klibi içerisindeki videonun Zaman Çizelgesi, ana Zaman Çizelgesi’nden bağımsız olarak oynatılır. Bu sayede bir tasarım içerisinde birden fazla video control edilebilir.
- **Grafik**, bir *. FLV video klibi bir grafik sembolü olarak gömülebilir fakat ActionScript dili bu videoyla iletişim kurulmaz.



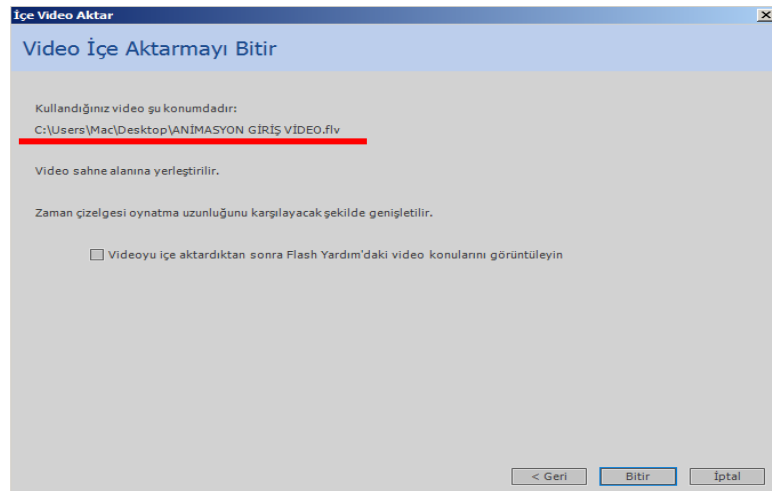
Şekil 6.24 Video içe aktar iletişim penceresinden tam denetimli video ekleme



Şekil 6.25 Çalışma alanı üzerindeki eklenecek *.Flv dosyasının seçimi



Şekil 6.26 Video dosyasının ekleneceği alanine seçimi



Şekil 6.27 Video dosyasının kayıt yolu

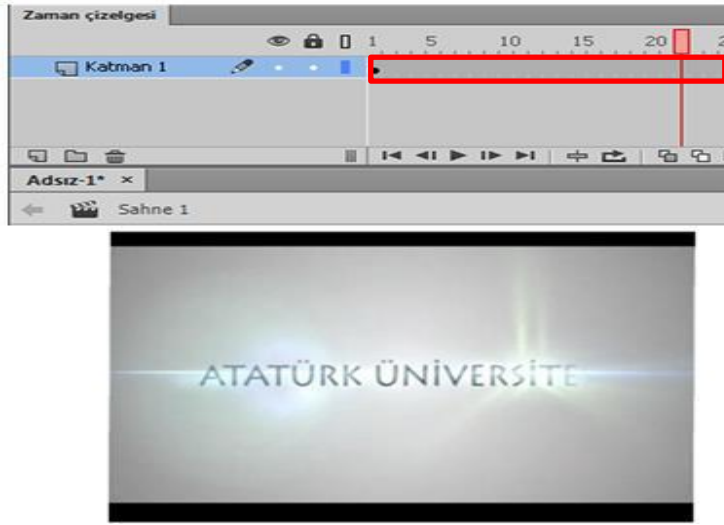
5. **Adım:** Şekil 6.27’ile gösterilen iletişim penceresinden videonun kayıt edileceği yer not alınarak **Bitir komutu** verilir ve böylece video çerçevesinin sahneye ve aynı zamanda kütüphane paneline yerleşmesi sağlanır.

Şekil 6.28’de gösterilen oynatma kafası zaman çizelgesi boyunca sürüklenerek (serbest kaydırma) gömülü videonun karelerini izlenebilir. Serbest kaydırma sırasında video ses parçasının oynatılmadığını unutmayın. Videoyu sesle önizlemek için Filmi Test Et komutu kullanılmalıdır.

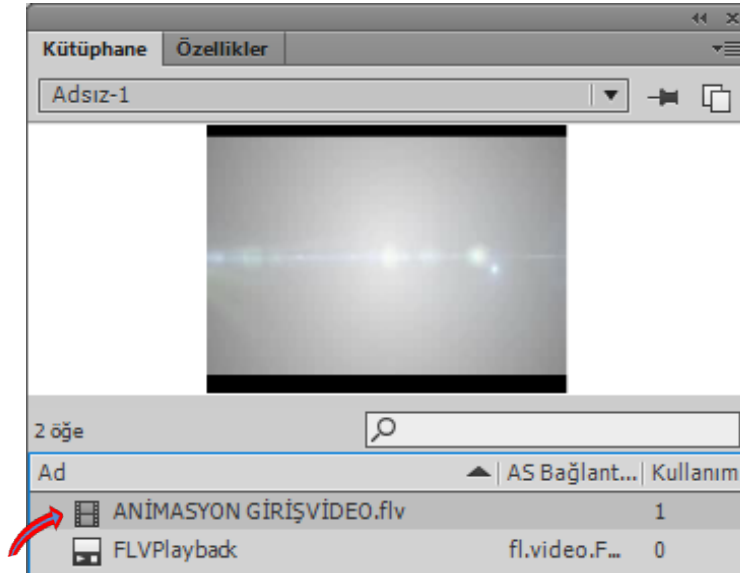
Şekil 6.29’da kütüphane paneline bu teknikle yerleşen video (*.flv) dosyası gösterilmektedir. Film klipi sembollerinde olduğu gibi bu video dosyası da Flash araçlarıyla yönetilebilir.



Video kare hızı ile Adobe Flash Zaman Çizelgesi kare hızı aynı olmalıdır. Flash Pro dosyanızın kare hızını, gömülü videonun kare hızıyla eşleştirecek şekilde ayarlayın.



Şekil 6.28 Zaman çizelgesi üzerine tam denetimle yerleştirilen video dosyası



Şekil 6.29 Kütüphane paneli üzerinde tam denetimle yerleştirilmiş video dosyası



Özet

- İnternet ortamlarındaki tasarımların daha etkileşimli ve dikkat çekici olmasını sağlamak amacıyla onlara ses, video ve film klipleri eklenir.
- Adobe Flash'ta sesleri kullanmanın yolu onları dışarıdan ithal etmektir. Bu işlem için en çok , Dosya(File) > İçe Aktar (Import)> Kütüphaneye İçeri Aktar seçeneği kullanılır.
- Adobe Flash içerisinde en çok "*.wav" ve "*.mp3" ses dosyaları kullanılır.
- Flash içinde yapılandırılabilen iki tür ses vardır. Bunlar olay sesleri ve akış sesleridir. Olay sesi, çalmaya başlamadan önce bilgisayara tamamen yüklenmiş olmalıdır. Bu ses animasyon içinde durduruluncaya kadar çalmaya devam eder. Akış sesleri, ilk birkaç kare için yeterli veri indirilir indirilmez çalmaya başlar; akış sesleri bir web sitesinde çalınmak üzere Zaman Çizelgesi ile senkronize çalışır.
- Katman üzerinde yer alan ses dosyasına Sesi Aç, Kapat, Sağ kanal ve Sol kanalda çal gibi efekt etkileri uygulanabilir.
- Animasyonlara video ekleyerek tasarımları daha ilgi çekilebilir hâle getirebilir. Flash'ın video formatı F4V ve FLV dir. FLV uzantılı video dosyaları zaman cetveli içerisine, gömülerek film kareleri hâline getirilebilir ve Çalışmanın bir parçasıymış gibi yönetebilir. Adobe Flash, MPEG türündeki harici video dosyalarını ise tasarıma dâhil edemez yalnızca belirli bir çerçeve içerisinde oynatılmasını sağlar.
- Adobe Flash *.flv, *.mp4, *.mov, *.3gp gibi birçok video dosyalarını çalışma içerisinde kabul eder. Bu işlem için; Dosya(File) > İçe Aktar (Import)> Video İçeri Aktar(Import Video..) seçeneği kullanılır.
- Video dosyalarının, boyutuna göre iki tür yükleme yöntemi kullanılır. Küçük boyutlu dosyalar bilgisayar üzerinde tutulurken yüksek çözünürlüklü boyutu büyük dosyalar Media Server (Ana bilgisayar) içerisinden aşamalı olarak yüklenir.
- Adobe Flash'a yüklenen videolar çalışılan bilgisayarda ve FLV türünden farklı ise; oynatma bileşeni içeren harici videoyu yükle seçeneği kullanılabilir,Bu seçenek seçildiğinde *.flv, *.mov, *.mp4 gibi video dosyalarına Adobe Flash dosyası içerisinden üzerine oynatma kontrolleri bulunan bir kaplama eklenir.
- FLV türündeki dosyalarda ise, FLV'yi SWF içine göm ve zaman çizelgesinde oynat seçeneği ile video dosyası gömdüğünüz zaman, video dosya verilerinin tümü Flash Pro dosyasına eklenir. Video, Zaman Çizelgesi kareleriyle temsil edilen video karelerini tek tek kontrol edilebilecek şekilde Zaman Çizelgesi'ne yerleştirilir. Bu çizelgeden diğer semboller gibi yönetilebilir.

DEĞERLENDİRME SORULARI

1. Aşağıdaki seçeneklerden hangisi ses için kullanılan dosya uzantısıdır?
 - a) *.flv
 - b) *.gif
 - c) *.wav
 - d) *.png
 - e) *.jpg
2. Aşağıdaki seçeneklerden hangisi video için kullanılan dosya uzantısıdır?
 - a) *.flv
 - b) *.gif
 - c) *.wav
 - d) *.png
 - e) *.jpg
3. Bir çalmaya başlamadan önce tamamen yüklenmiş olmalıdır ve bu ses açıkça durdurulana kadar çalmaya devam eder. Aşağıdaki seçeneklerden hangisi nokta ile verilen boşluğu doldurmak için kullanılabilir?
 - a) Olay sesi
 - b) Akış sesi
 - c) Film sesi
 - d) Grafik sesi
 - e) Katman sesi
4. Zaman Çizelgesi'ne ses ekledikten sonra eklenen ses için hangi efekt seçilemez?
 - a) Sesi aç
 - b) Sesi kıs
 - c) Özel
 - d) Yok
 - e) Soldur
5. Ses eklenen bir film karesinin özelliklerinde yer alan “Senkr” açılır menüsünde hangisi seçenek yer almaz?
 - a) Yeniden başlat
 - b) Olay
 - c) Başlat
 - d) Durdur
 - e) Akış

6. Kütüphane panelindeki bir ses dosyasının özelliklerinde hangi seçenek bulunmaz?

- a) Güncelle
- b) İçer aktar
- c) Test et
- d) Tekrarla
- e) Sıkıştırma

7.  şeklinde verilen zaman cetvelinde, ses dosyası hangi film karelerinde yer alır?

- a) 1-9
- b) 1-11
- c) 5-10
- d) 2-11
- e) 3-12

8. FLVPlayback üzerinde aşağıdaki kontroller hangisi bulunamaz?

- a) Oynat
- b) Durdur
- c) Duraklat
- d) İleri Sar
- e) Kapat

9. Video dosyası ekranda gösterilirken, video kontrol düğmelerinin, gösterim ekranında gizlenmesi için hangi özellik kullanılır?

- a) AutoPlay
- b) ScaleMode
- c) Skin
- d) Skin AutoHide,
- e) Source

10.Çalışma alanına eklenen bir videoyu sembollerle birlikte kullanabilmek için aşağıdaki seçenekten hangisi kullanılır?

- a) Video bileşenini içeren dosyayı zaman çizelgesinde oynat
- b) Oynatma bileşeni içeren harici videoyu yükle
- c) FLV'yi SWF içine göm ve zaman çizelgesinde oynat
- d) Media Server (Ana bilgisayar) içerisinden aşamalı olarak yüklenir.
- e) MPG4' dosyasını SWF içine göm ve zaman çizelgesinde oynat

Cevap Anahtarı

1.c, 2.a, 3.a, 4.e, 5.a, 6.d, 7.d, 8.e, 9.d, 10.c

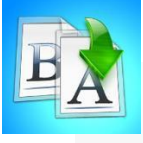
YARARLANILAN KAYNAKLAR

[1]Adobe Flash Pro CC, Yardımcı el kitabı. <http://helpx.adobe.com/tr/support>

[Erişim tarihi: 30 Ekim 2013].

[2]SCHULZE Pati.(2004) Macromedia Fireworks 8 Kaynağından Eğitim, Medyasoft Yayınları.

ANİMASYON



İÇİNDEKİLER

- Animasyon
- Kare Kare Animasyon
- Hareket Animasyonları
- Bir Yol Boyunca Hareket Eden Animasyonlar
- Hareket Ön Ayarlarını Kullanarak Yapılan Animasyonlar
- Şekil Animasyonları



HEDEFLER

- Bu üniteyi çalıştıktan sonra;
- Animasyon hazırlamak için kullanılan teknikleri öğrenebilecek,
- Kare kare animasyon yapabilecek,
- Çeşitli nesnelere hareket ve şekil animasyonları uygulayarak onları canlandırabileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

ANİMASYON TASARIMI

Okt.
Mustafa AKBUĞA

ÜNİTE 7

Giriş

Web sitelerinde kullanılan animasyonların hazırlanmasında genellikle Adobe flash programı kullanılır. Çünkü bu program dosya boyutunu küçük tutarak yüksek kalitede animasyonlar oluşturulmasını sağlar. Adobe Flash ile animasyon dosyalarını hazırlarken Film kareleri tek tek tasarlayabilir ya da hazır efektler ve kodlardan yararlanabilirsiniz. Bu ünite kare kare, hazır efektler, hareket ve şekil animasyon teknikleri üzerinde durulacaktır.



Animatörlük, uzmanlaşması uzun yıllar alan ve yetenek gerektiren bir meslektir.



Adobe Flash içerisinde oluşturulan bütün animasyonlar Film Klibi sembolü içinde oluşturulmalıdır.

ANİMASYON

Grafik ya da resim nesnelerinden oluşan film karelerinin art arda ve belirli bir hızla gösterilmesi sonucu oluşan hareketli görüntülere animasyon denir. Adobe Flash'ta animasyonu oluşturan temel öğeler zaman çizelgesi üzerinde yer alan ve çeşitli nesneleri içerisinde barındıran film karelerdir. Film kareleri içindeki nesneler araç panelindeki araçlarla çizilen çizimler veya fotoğraflardan oluşabilir.

Adobe Flash'ta animasyonlar kare kare, hazır efektler, hareket-şekil olmak üzere üç farklı yapım tekniği kullanılarak hazırlanır. Tekniklerin birincisinde, animasyonu oluşturan tüm karelerin tasarımı animatörler tarafından yapılır. İkincisinde, ilk film karesi tasarlanır ve geriye kalan animasyon kısımları, Adobe Flash hazır efektleri kullanarak tamamlanır. Üçüncü teknikte ise başlangıç ve bitiş noktalarındaki anahtar kareler tasarlanarak arada kalan film kareleri Adobe Flash'a bırakılır.

Adobe Flash'ın desteklediği animasyon yapım teknikleri şunlardır;

Kare Kare Animasyon Oluşturma

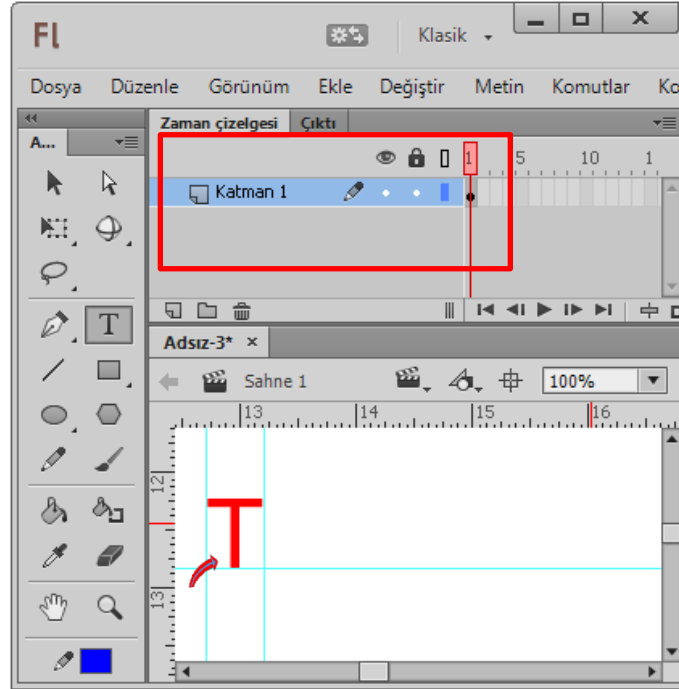
Kare kare animasyon oluşturulması uzun zaman alan bir animasyon yapım tekniğidir. Bu animasyonu oluşturan her bir film karesi tek tek oluşturulup, elle düzenlenir. Bu yöntemle, otomatik olarak oluşturulan film karelerine göre çok daha karmaşık etkiler üretmeniz mümkündür. Ancak kare kare animasyon yapmak tahmin edebileceğiniz gibi hem daha çok zamanınızı alır, hem de tasarım dosyasının boyutunu artırır.

Aşağıdaki uygulamada, kare kare animasyonun nasıl oluşturulduğu aşamalar hâlinde gösterilmiştir.

1.Adım: "Dosya>Yeni Oluştur>ActionScript 3.0" komutu verilerek Şekil 7.1'deki gibi Katman 1 ve film1 karesinden oluşan yeni bir çalışma sayfası oluşturun ve 1. film karesine "T" harfini yerleştirin. Şekil 7.1 'de verilen görüntünün oluşması için çalışmaya Cetvel ve Klavuz çizgilerini de ekleyiniz.

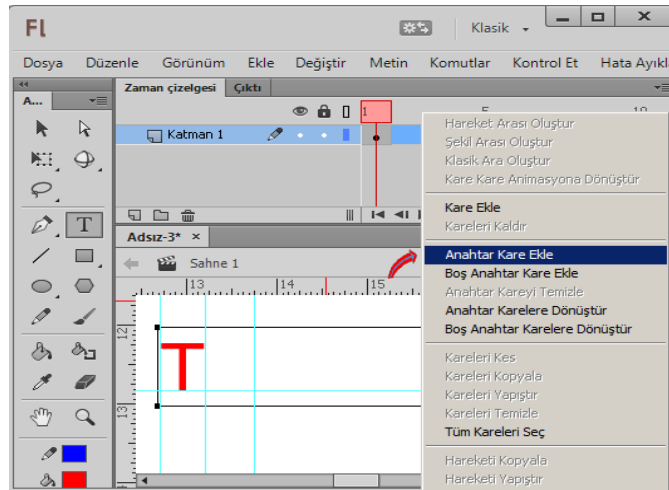


Adobe Flash ile tasarlanan bütün animasyon türleri istenirse zaman çizelgesi üzerinde “mouse sağ tuş>kare kare animasyona dönüştür” komutunun kullanılması ile kare kare animasyona dönüştürülebilir.



Şekil 7.1 Zaman çizelgesindeki ilk anahtar karenin tasarlanması

2. Adım: Zaman çizelgesi üzerinde Şekil 7.1’de gösterilen 1. anahtar film karesinin yanında yer alan boş film karesine Şekil 7.2’de gösterilen menüden 2. anahtar film karesini ekleyin.



Şekil 7.2 Zaman çizelgesine ikinci anahtar karenin eklenmesi

3.Adım: Zaman çizelgesi üzerinde 2. anahtar film karesi içerisinde Şekil 7.3’te gösterilen “A” harfini yerleştiriniz. Sonrasında “TASARIM” yazabilecek kadar her harf için ayrı ayrı Anahtar kare oluşturarak toplamda zaman cetveli üzerinde 7 Anahtar kare oluşturarak animasyonu tamamlayınız.

Her eklenen yeni anahtar kare başlangıçta, kendisinden önce gelen anahtar karenin içeriğini aynen yeni alana taşır, böylece animasyondaki karelerin kolayca değiştirilmesi sağlanır.Yapılan Şekil 7.4’teki tasarıma dikkat edilirse ana zaman

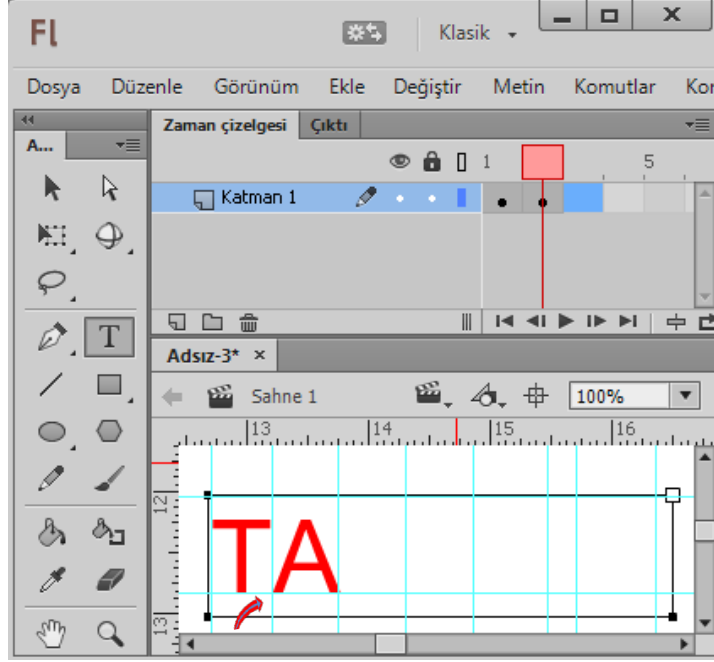
Animasyon

çizelgesi üzerindeki film kareleri kullanarak tasarım tamamlanmıştır. Bu şekilde tasarlanan bir animasyona başka sahnelerde kullanmaya ihtiyaç duyulduğunda yeniden tasarlanması gerekecektir. Yeniden aynı tasaramı yaparak zaman kaybetmemek için tasarlanan bir animasyona aşağıdaki işlemler uygulanarak gerektiğinde başka yerlerde de kullanılabilir. Bunun için çalışma sahnesine tasarımdan önce bir film klipi eklenir ve film klipinin içindeki zaman çizelgesinde tasarım gerçekleştirilir.

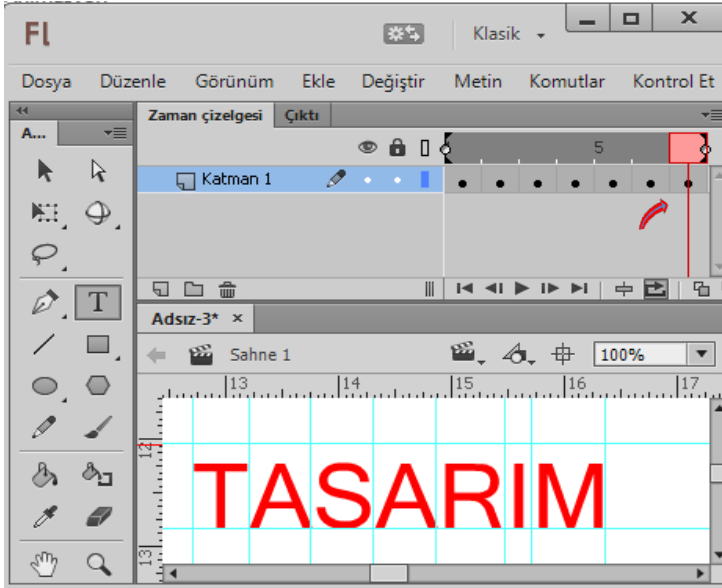
Film kliplerinde yapılan animasyonlar Kütüphane paneline kaydedildiği için gerektiğinde bu alandan alınan animasyonlar istenilen yerlerde (Başka sahne, Düğme ve Film kliplerinde) yeniden kullanılabilir.



Kare Kare animasyon esnasında film karelerinin yavaş yavaş kaybolmasını sağlarsanız animasyonlar daha iyi görünür. Bunu yapmak için, aynı içerikten iki anahtar kare oluşturarak ikincisinin renk efekti özelliğini kullanarak Alfa değerini değiştirebilirsiniz.



Şekil 7.3 Zaman çizelgesine eklenen ikinci anahtar kareye içerik eklenmesi



Şekil 7.4 Kare kare animasyon için gereken bütün Anahtar karelerin düzenlenmesi



Animasyonlardaki çizimleri elle yapıyorsanız, bunun için bir çizim tabletini kullanmayı deneyebilirsiniz.



Bir katmanda sadece bir tane hareket animasyonu oluşturabilirsiniz. Aynı anda birden fazla hareket animasyonu oluşturmak istiyorsanız, sembollerin her birini ayrı katmanlara yerleştiriniz.

3. Adım: Zaman çizelgesi üzerindeki oynatım kafası hareket ettirildiğinde animasyon hareket kazanır ya da Kontrol Et>Test Et komutu verilerek, Şekil 7.5'teki gibi bir sonuç ekranda görüntülenir.

Word belgesinde "TASARIM" şeklinde yazının tamamı görülmesine rağmen bilgisayar ortamında film kareleri, ekranda geliş zamanlamasına göre her bir harf zaman içerisinde tek tek görüntülenir.



Şekil 7.5 Zaman çizelgesindeki ilk anahtar karenin tasarlanması

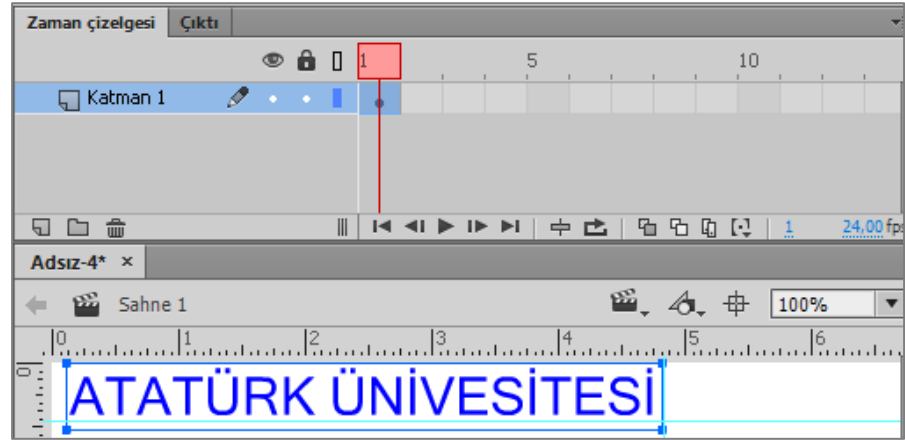
Hareket animasyonları oluşturma;

Hareket animasyon tekniği, sabit bir zaman aralığında bir nesnenin konumunu değiştirme, ölçeklendirme, şeffaflık ayarlama, tonlama ya da döndürme işlemleri yapmak için de kullanılmaktadır. Hareket animasyonları ara doldurma film kareleri kullanılarak yapılmaktadır. Hareket Animasyonu (Motion Tween) Adobe Flash'ta en çok kullanılan nesne hareketlendirme tekniğidir. Hareket sağlayan bu animasyon yapım şeklinde nesnenin ilk karedeki ve son karedeki durumunu iki anahtar kare ile tasarlanır. Flash bu iki kareyi kullanarak, izlediğinizde nesnenin ilk karedeki konumundan son karedeki konuma gidiyormuş algısını oluşturacak, gerekli ara kareleri kendisi hesaplayarak animasyonu oluşturur.

Bu teknikte animasyon oluşturmak için çalışma alanına eklenen şekil ve grafikler mutlaka sembol türlerinden birine dönüştürülmelidir. Nesne şekil hâlindeyken ara hareket yöntemleri uygulanamaz. Hareket animasyonları, kare kare animasyona göre daha az yer kaplar.

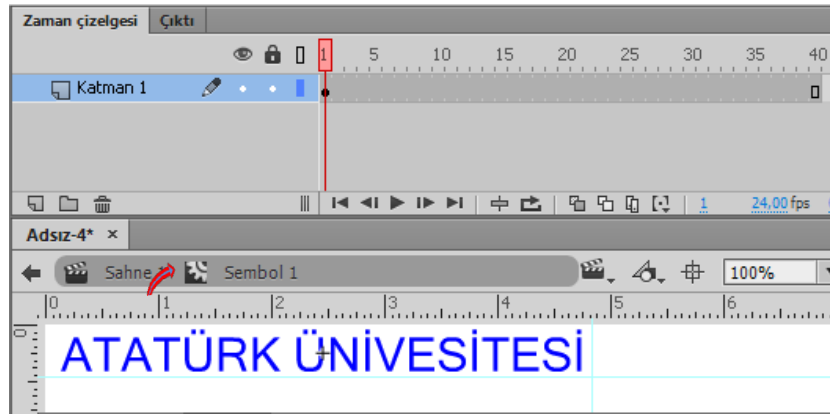
Aşağıdaki örnek uygulama ile web sayfalarında sıkça karşılaşılan kayan yazı (Son dakika haberleri, borsa verileri ve reklam vb.) animasyonun nasıl hazırlandığı adımlar hâlinde anlatılacaktır.

- 1. Adım:** Adobe Flash yazılımında "Dosya>Yeni Oluştur>ActionScript 3.0" komutu verilerek Katman 1 ve film1 karesinden oluşan yeni bir çalışma
- 2. Adım:** sayfası oluşturun ve oluşturulan 1. film karesine Şekil 7.6'da gösterildiği gibi Araç Çubuğu üzerindeki yazı aracını kullanarak "ATATÜRK ÜNİVERSİTESİ" yazan metni çalışma alanına yerleştirin.



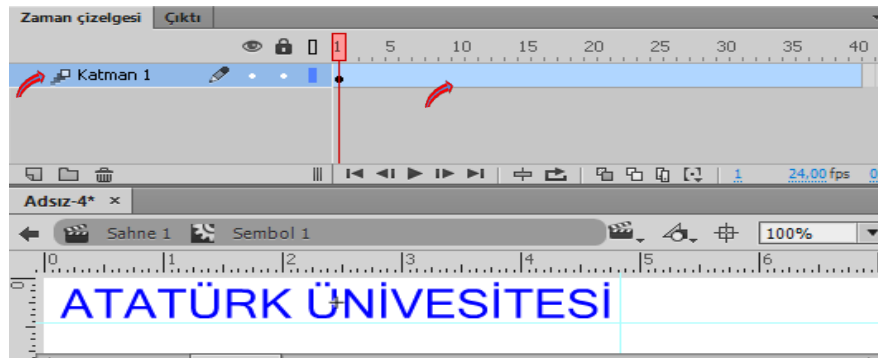
Şekil 7.6 Zaman çizelgesindeki ilk Anahtar karenin tasarlanması

3. **Adım:** Çalışma alanındaki “ATATÜRK ÜNİVERSİTESİ” yazan metni seçerek “Değiştir > Sembole Dönüştür” komutu yardımıyla metni film klipi sembolüne dönüştürün. Şekil 7.7’de gösterilen sembol1 içerisine girilerek sembolün zaman çizelgesi üzerinde kayan yazının uzunluğunu 40 film karesi olarak seçin. Bu nokta üzerine de normal kare ekleyin.



Şekil 7.7 Zaman çizelgesinde hareket animasyonu için yer ayrılması

4. **Adım:** Bu adımda yapmanız gereken yazıyı hareketli hâle getirmektir. Bu aşamada 1.Kare seçilerek “Ekle>Ara Hareket” komutu verilerek hareket oluşturulur. Şekil 7.8’de gösterilen zaman çizelgesinde kırmızı okla gösterilen katman simgesi ve zaman çizelgesindeki film karesinin renklerinden hareket etkisinin uygulandığını aralığı görebilirsiniz.

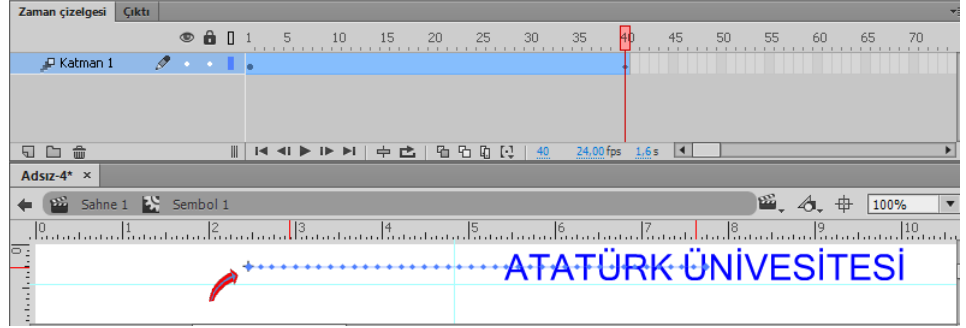


Şekil 7.8 Film karelerine ara hareket animasyon tekniğinin uygulanması



Zaman çizelgesinde mavi ile işaretlenen hareket animasyonunun gösterim süresini uzatmak isterseniz iki anahtar karenin arasına kare ekleyebilirsiniz

5.Adım: Bu adımda kayan yazının sahne içerisinde hangi noktalar arasında hareket edeceği ayarlanacaktır. Metnin sahne üzerinde Şekil 7.8’de gösterildiği nokta, hareketin başlangıç noktasıdır (1. Film karesi). Yazının hareket ettirilmesi hâlineson karede hangi noktaya kadar gideceğine karar vermesi gerekir. Bunun için öncelikle son kareyi 40. kare olarak seçin ve Şekil 7.9’da gösterildiği gibi kırmızı okla gösterilen noktadaki metni tutarak son gideceği yere kadar taşıyınız (metni taşımak için araç kutusundaki seçim aracını kullanabilirsiniz).



Şekil 7.9 Hareket animasyonu uygulanmış bir nesneye son konum bilgisinin verilmesi

6. Adım: Bu adımla kayan yazı animasyonu test edilecektir. Bu işlem için Kontrol Et>Test Et komutu verilirse Şekil 7.10’daki gibi bir sonuç ekranda görüntülenir. Test sonucunda Şekil 7.10’da kırmızı ok ile gösterilen noktadan başlayan ve ekranda sürekli sağa doğru kayan “ATATÜRK ÜNİVERSİTESİ “ metni görüntülenecektir.



Şekil 7.10 Animasyonun Flash Professional üzerinde test edilmesi

Hareket animasyonları tasarlanırken animasyonun hangi yönde hareket edeceği aşağıdaki yöntemle belirlenir.

- **Bir Yol Boyunca Hareket etmeyi sağlayan animasyon oluşturma**

Eğer isterseniz, bir sembolü sadece düz bir yol üzerinde değil de belirlenen bir rota üzerinde hareket etmesi bu yöntemle sağlanabilir. Bunu yapabilmek için, normal bir hareket animasyonu oluşturduğunuz katmanın üzerine “Klasik Hareket Kılavuzu Ekle” komutu yardımıyla klavuz katman oluşturup bu katman üzerine istediğiniz yolu belirleyerek animasyonun sizin çizdiğiniz bu yolu takip etmesini sağlayabilirsiniz.

Şimdi bu tekniği bir örnekte adım adım uygulayalım;

1. **Adım:** Öncelikle “Dosya>Yeni Oluştur>ActionScript 3.0” komutu verilerek Katman 1 ve film1 karesinden oluşan yeni bir çalışma sayfası oluşturun ve

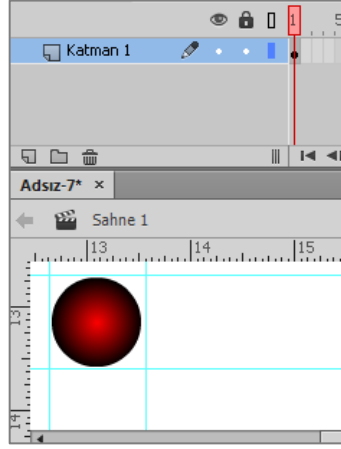


Bir film klipi sembolü içerisine birden fazla animasyon içeren başka film klipi sembolleri de kullanabilirsiniz.



Zaman çizelgesindeki bütün kareleri seçmek için mouse yardımıyla nesnenin bulunduğu katmandaki herhangi bir film karesine çift tıklamak yeterlidir.

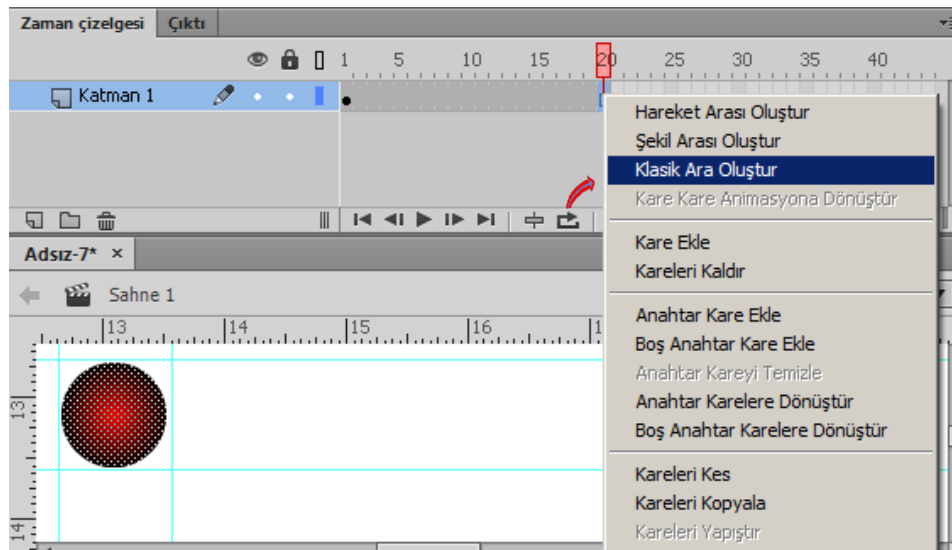
oluşturulan 1. film karesine Şekil 7.11’de gösterildiği gibi Araç Çubuğu üzerindeki Oval aracını kullanarak bir daire çizip, dolgu rengini şekildeki gibi seçerek çalışma alanına bir daire yerleştiriniz.



Şekil 7.11 Zaman çizelgesindeki ilk Anahtar karenin tasarlanması

2. Adım: Bu adımda kırmızı top şeklindeki daireye yatay hareket kazandırılacaktır. Bu işlem için öncelikle, 1. film karesinden 20. film karesine kadar dairenin yatay ekseninde, düz bir şekilde hareket etmesini sağlamak için son kareye (20. kareye) tıklayarak normal kare ekleyiniz.

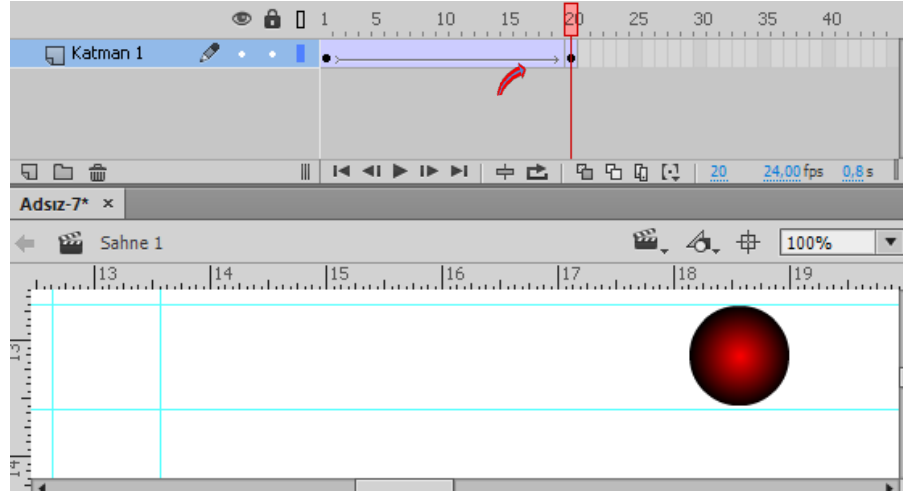
Sonraki aşamada 1. kare ile 20. kareye kadar olan bütün kareleri seçerek Şekil 7.12’de gösterildiği gibi “Ekle>Klasik Ara hareket” ya da “Farenin Sağ tuşu>Klasik Ara Oluştur” komutu ile hareket oluşturun. Bu işlem sonunda son kareye (20. kare) tıklayarak araçlardan seçme aracını farenin sol tuşu ile seçiniz ve Şekil 7.13’te gösterildiği gibi 1. Film karesinin bulunduğu noktada dairenin orta noktasından tutarak son gideceği yere kadar yatay olarak nesneyi taşıyınız. Yapılan animasyon artık 1. film karesinden 20. film karesine kadar yatay olarak hareket bir daire şeklinde olacaktır.



Şekil 7.12 ilk Anahtar kareye Klasik Ara Oluşturma tekniğinin uygulanması



Bir hareket animasyonu uygulanmış nesne ile sonradan çizilen klavuz yolun ilişkilendirilmesi amacıyla nesne mouse yardımıyla taşınarak klavuz yol üzerine bırakılır.

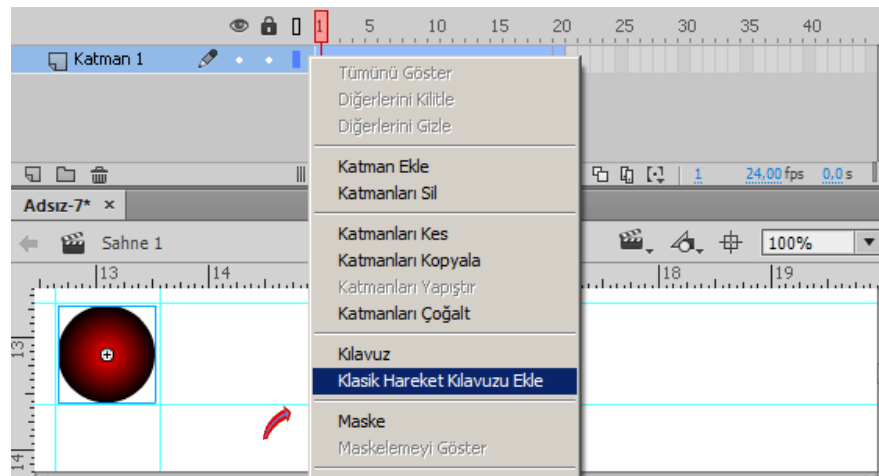


Şekil 7.13 Animasyonda bitiş noktasındaki Anahtar karenin tasarlanması

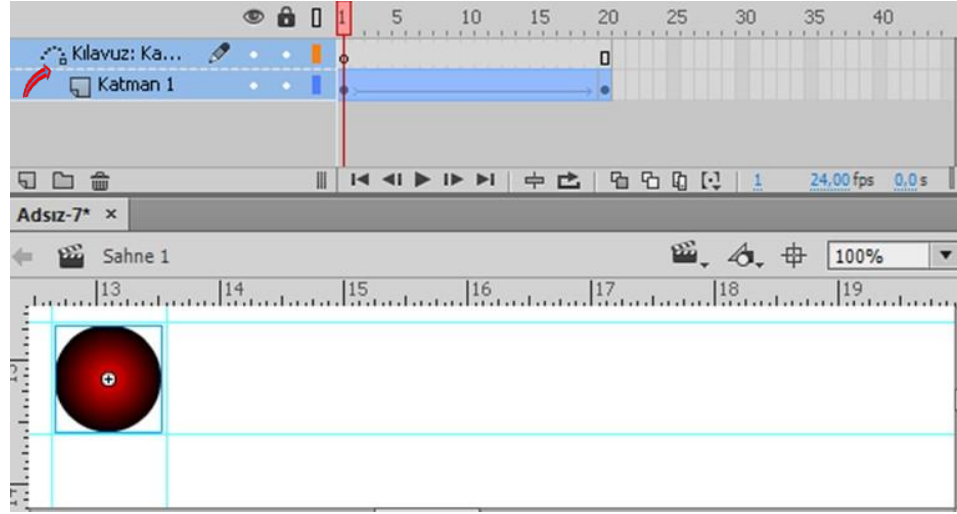
3. Adım: Bu adımda yatay hareket etmekte olan dairenin bir klavuz çizgisi takip etmesi sağlanacaktır. Bu işlem için dairenin içinde bulunduğu katman seçilerek Şekil 7.14'te gösterildiği gibi mouse ile bu katman üzerindeyken "Mouse sağ tuş > Klasik Hareket Kılavuzu Ekle" komutu verilir.

Şekil 7.15'de gösterilen klavuz katmana ait 1. film karesine Araç kutusundaki kalem aracıyla Şekil 7.16'da gösterilen yola benzer bir klavuz yolu çizin. Şekil 7.15'te gösterilen zaman çizelgesindeki 1. film karesinde yer alan dairenin orta noktasından tutarak, çizilen klavuz yolun başlangıç noktasıyla ilişkilendirin. Sonrasında zaman çizelgesindeki 20. film karesini (son film karesi) seçerek dairenin orta noktasını yolun son noktasıyla ilişkilendirin.

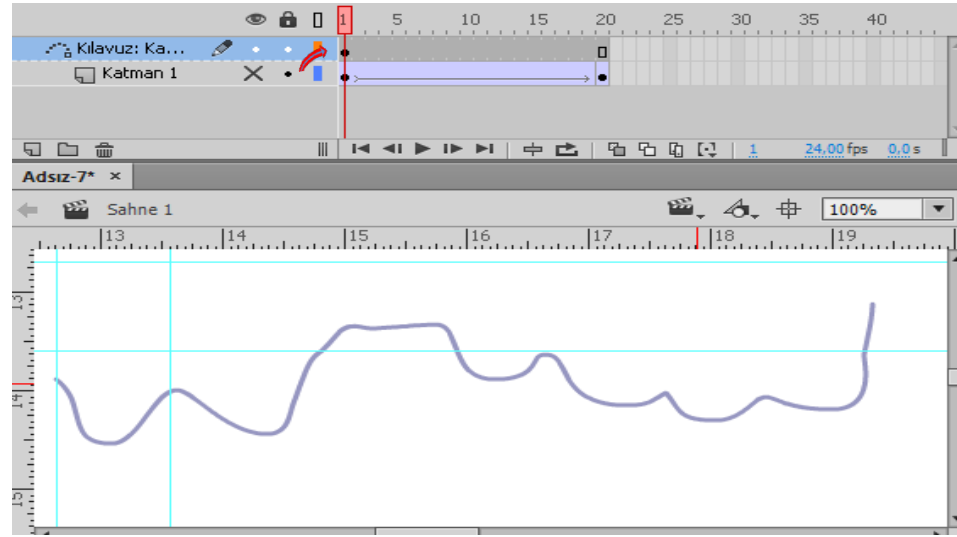
Yapılan animasyon test edildiğinde Şekil 7.17'de görüldüğü gibi dairenin klavuz yolu takip ettiği görülecektir. Test işlemi sırasında klavuz yol ekranda görülmez. Sadece dairenin takip ettiği yoldan farkedilebilir.



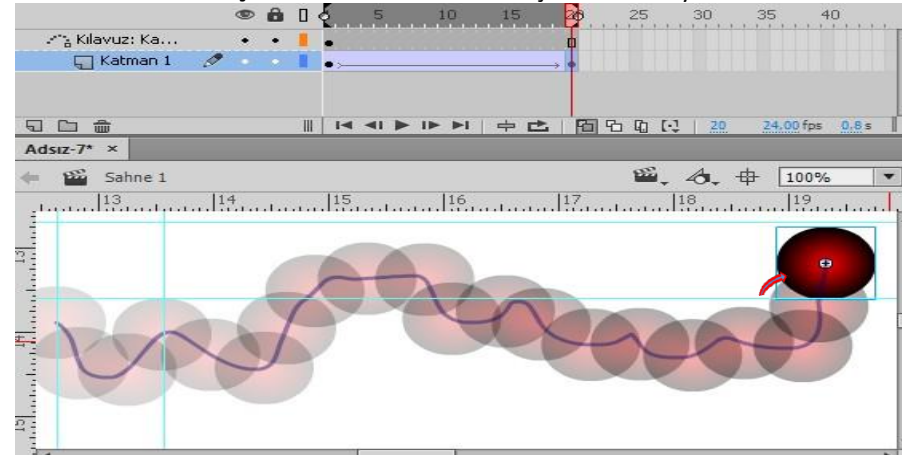
Şekil 7.14 Katman 1 için Klasik Hareket Kılavuzu Ekleme işlemi



Şekil 7.15 Eklene Klavuz katman ve zaman çizelgesinin görünümü



Şekil 7.16 Katman üzerinde çizilen klavuz yol



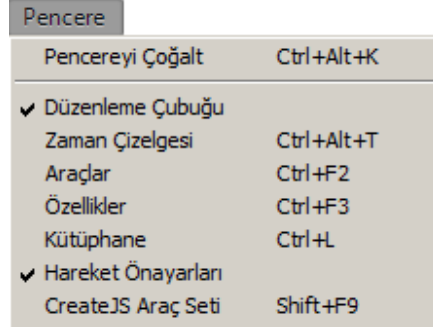
Şekil 7.17 Hareket amimasyonu uygulanmış nesnenin klavuz yol ile ilişkilendirilmesi



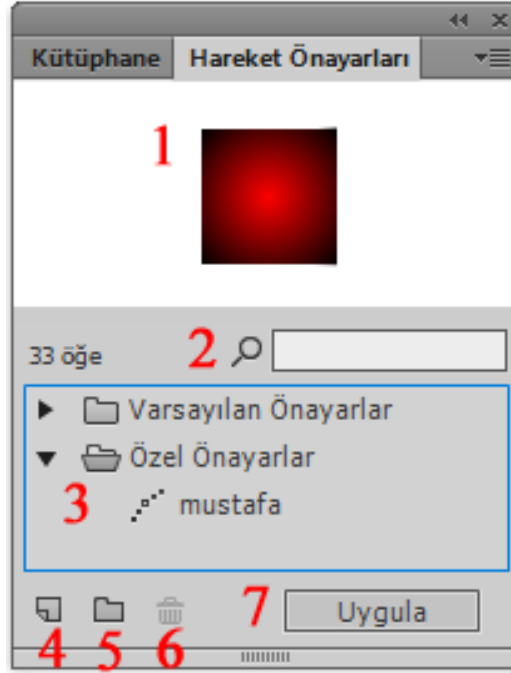
Hareket ön ayarlar panelinde yer alan 32 adet varsayılan ön ayarlar istenirse (değişikler yapıldıktan sonra) başka isimle tekrar kaydedilebilir.

Hareket Ön Ayarlarını Kullanılarak Animasyon Oluşturma

Animasyon tasarımı, film karelerini tek tek tasarlamak yerine kullanılan diğer bir yöntem ise hareket ön ayarları panelinde yer alan hazır efektleri kullanmaktır. Bu panel yardımıyla önceden nesnelere uygulanan tekniklerdeki animasyonlar kaydedilebileceği gibi içerisinde yer alan 32 tane varsayılan ön ayarlar kullanılarak da animasyon oluşturulabilir. Adobe Flash çalışma alanında bu panel bulunmuyorsa Şekil 7.18’de görülen pencere menüsünden, “Pencere>Hareket Ön ayarları” komutu verilerek bu panel açılabilir.



Şekil 7.18 Hareket ön ayar panelinin çalışma alanına eklenmesi



Şekil 7.19 Hareket ön ayar paneli

Şekil 7.19’da numaralandırılan hareket ön ayarlar panelinde alanlar şunlardır;
1. Canlı görüntüleme: Seçim yapılacak ön ayarlardan bir tanesi seçildiğinde bu alan yardımıyla oluşacak animasyonlar hiçbir komuta gerek kalmadan bu alanda canlandırılır.



Hareket ön ayarları panelinde bulunan varsayılan ön ayar klasöründeki kayıtları silmek mümkün değildir.



Varsayılan önayar klasöründeki kayıtlardan test ederek sizin için kullanışlı olanlarını farklı isimle kaydedin ve bir klasör içerisinde toplayın. Bu işlem animasyon tasarımında size büyük kolaylık sağlar.

2. Kayıtlı ayarları ara: Panel içerisinde tek seferde gösterilemeyen ön ayarların, yazılan isimlerine göre araması yapılarak ayar seç bölümüne getirilmesi sağlanır.

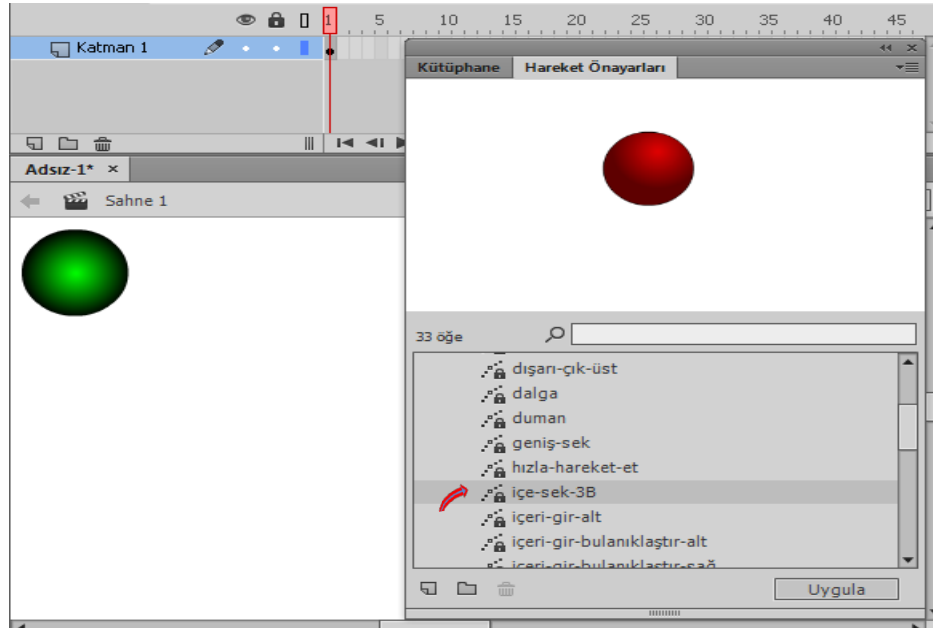
3. Varsayılan veya kaydedilmiş ayarları seç: Bu alan 32 tane kayıtlı ya da sonradan kullanıcı tarafından “Seçimi ön ayar olarak kaydet” seçeneği kullanılarak kaydedilmiş animasyon ön ayarlardan birinin seçilmesini sağlar.

4. Seçimi ön ayar olarak kaydet: Çalışma alanı içerisinde panellerin ve animasyon ekleme yöntemleriyle oluşturulan ayarların verilecek isme göre kaydedilmesini sağlar.

5. Yeni klasör oluştur: Birçok ön ayarın bulunduğu durumlarda oluşturulan etkilere göre gruplandırılma yapılabilmesi için gereken klasörleri oluşturmayı sağlar.

6. Ögeyi kaldır: Seçim alanında sonradan kaydedilen ön ayarların silinmesi sağlar. Hareket ön ayarlar Panelinde yer alan hazır efektler ile animasyon oluşturmak için: Çalışma alanına bir grafik nesnesi (daire, yıldız vb.) çizilir ve bu nesne film klipi sembolüne dönüştürülür. Sonrasında ise Şekil 7.20’de Varsayılan ön ayarlar içerisinde bulunan hazır efektlerden biri seçilerek oluşturulan sembole uygulanır. Yapılan bu işlemlerle grafik nesnesi animasyon nesnesine dönüştürülür. Şimdi bu tekniği bir örnekte adım adım uygulayalım;

- 1. Adım:** Bu adımla yeni bir çalışma sayfası oluşturulacaktır. Bu işlem için; “Dosya>Yeni Oluştur>ActionScript 3.0” komutu verilerek Katman 1 ve film1 karesinden oluşan yeni bir çalışma sayfası oluşturun ve oluşturulan 1. Fim karesine Şekil 7.20 ’de gösterildiği gibi Araç Çubuğu üzerindeki Oval aracını kullanarak bir daire çizin, dairenin dolgu rengini şekildeki gibi seçerek çalışma alanına bu daireyi yerleştiriniz.



Şekil 7.20 Çalışma alanında yer alan bir nesneye seçilen bir ön ayarın uygulanması



Nesne başına yalnızca bir ön ayar uygulayabilirsiniz. Aynı nesneye ikinci bir ön ayar uygularsanız, bu ikinci ön ayar birinci önayarın yerini alır.



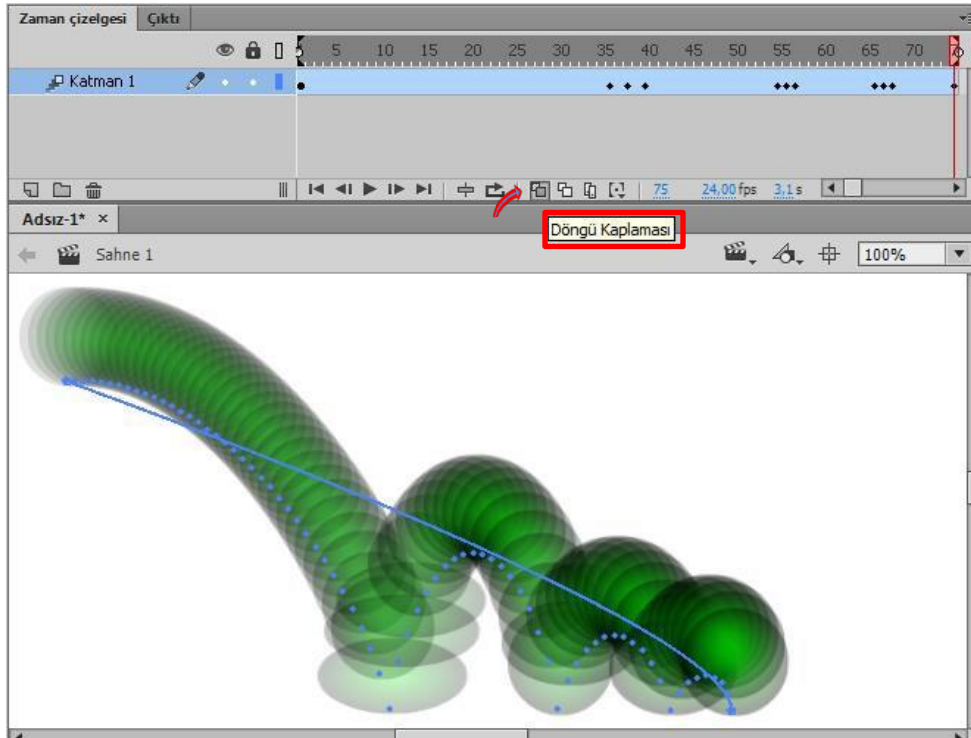
Döngü kaplaması seçeneği Zaman cetveli üzerindeki işaretli alanların aynı anda ekranda görüntülenmesi için gereklidir.



Bir şekil çizildikten sonra sembole çevrilirse ana zaman çizelgesindeki şekille beraber cetvel ve klavuz çizgilerde sembol içine taşınır.

2.Adım: Bu adımda yapmanız gereken yeşil top şeklindeki dairenin bir zemin üzerinde zıplamasını sağlamaktır. Bu işlem için; Şekil 7.20’de gösterilen hareket ön ayarları panelinde bulunan “İçe-Sek-3B” isimli varsayılan ayar seçilerek nesneye uygulanır.

Seçilen nesne önceden sebole dönüştürülmediyse “Ara için sembole dönüştür” uyarısı verilir. “Tamam” butonu, seçildiğinde nesne Film klipi sembolüne dönüştürülerek animasyon için gereken ayarlar otomatik olarak zaman çizelgesine aktarılır. Sahne üzerinde yer alan nesnenin, zaman çizelgesi üzerindeki hareketleri Şekil 7.21’de kırmızı okla gösterilen “Döngü Kaplaması” bölümü seçilerek izlenebilir.



Şekil 7.21 Bir daire nesnesine uygulanan ön ayar etkisinin döngü kaplama seçeneği yardımıyla görüntülenmesi

Şekil animasyonları oluşturma;

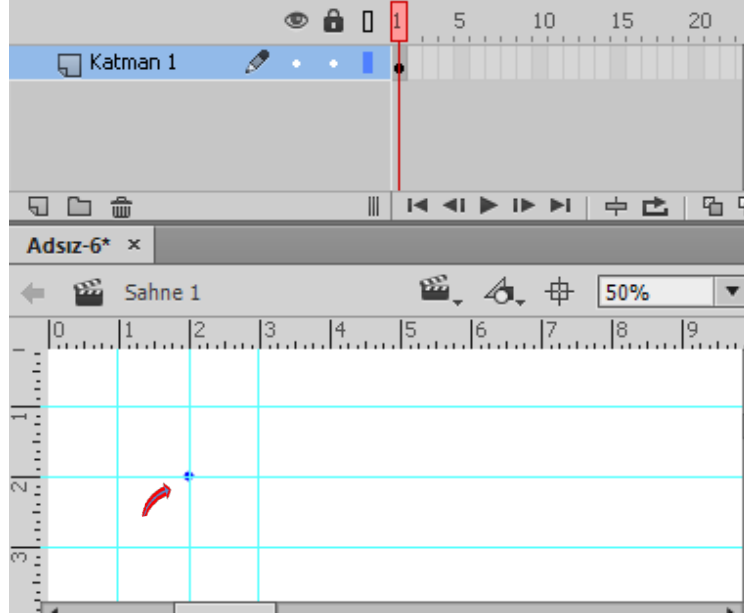
Şekil animasyonları döndürme, ölçekleme, eğme gibi dönüşüm işlemlerini kullanarak elde edemediğimiz değişimleri oluşturmak için kullanılan bir tekniktir. Bu animasyon oluşturma tekniğinde; bir nesnenin belirli bir süre içerisinde başka bir nesneye dönüşmesi sağlanır.

Bir sembole, gruba veya resme şekil animasyonu uygulamak istendiğinde bu nesnelere Parçala (Break Apart) komutu verilerek parçalamak gerekir. Bu işlem yapılmadığı takdirde şekil arası animasyon seçenekleri kullanılamaz. Aynı katman üzerinde birden fazla şekil animasyonu kullanılabilir. Şekil animasyonları, hareket animasyonları gibi kare kare animasyona göre daha az yer kaplar.

Aşağıda küçük bir noktanın bir daireye sonra dikdörtgen şekline nasıl

dönüştürüleceği adımlar hâlinde gösterilmiştir.

1. **Adım:** Bu adımda yeni çalışma sayfası oluşturulacaktır. Bu işlem için; “Dosya>Yeni Oluştur>ActionScript 3.0” komutu verilerek Katman 1 ve film1 karesinden oluşan yeni bir çalışma sayfası oluşturun ve oluşturulan katman içerisindeki 1. film karesine Şekil 7.22’de gösterildiği gibi Araç Çubuğu üzerindeki fırça aracını kullanarak çalışma alanına kırmızı okla gösterildiği gibi bir nokta yerleştiriniz.



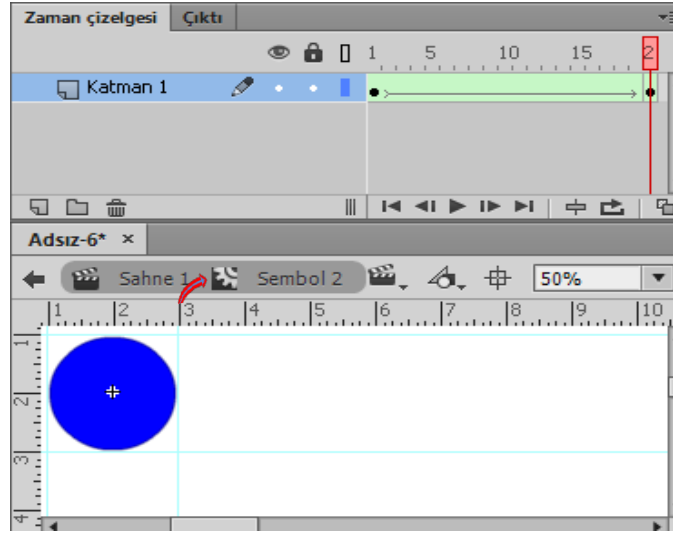
Şekil 7.22 Şekil animasyonu için zaman çizelgesinde ilk anahtar karenin tasarlanması

2. **Adım:** Bu adımda şekil animasyonu oluşturmak için film klibi sembolü oluşturulacaktır. Bu işlem için; çalışma alanında oluşturulan nokta seçilerek “Değiştir > Sembole Dönüştür” komutu ile film klibi sembolüne dönüştürünüz. Şekil 7.23’de gösterildiği gibi bu sembolün üzerinde mouse sol tuşuyla çift tıklayarak ya da kütüphane menüsünden düzenle komutuyla kırmızı okla gösterilen sembol düzenleme bölümüne girilir.

Sembol ait zaman çizelgesi üzerinde, animasyonun kaplaması gereken film kare sayısı düşünülerek en son kare seçilir ve bu kareye boş anahtar karesi eklenir. Şekil 7.22’de animasyonun 20 film karesi kapladığı farzedildiğinden boş anahtar kare 20. film karesine konulmuştur. Bu eklenen kareye şekil 7.23’teki gibi bir daire çizin.

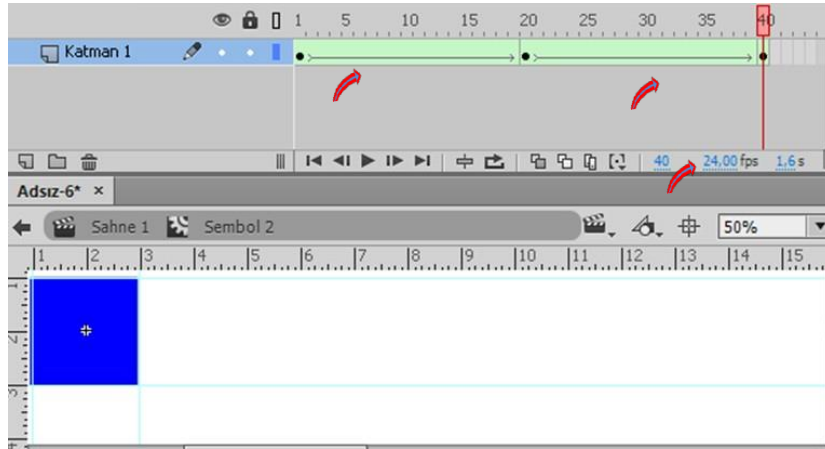


Animasyonun oynatıldığı hız, saniyedeki kare sayısı ile ölçülür (fps). Çok düşük bir kare hızı animasyonun durup devam ediyormuş gibi görünmesine neden olur; çok yüksek kare hızı ise animasyonun ayrıntılarını bulanıklaştırır.



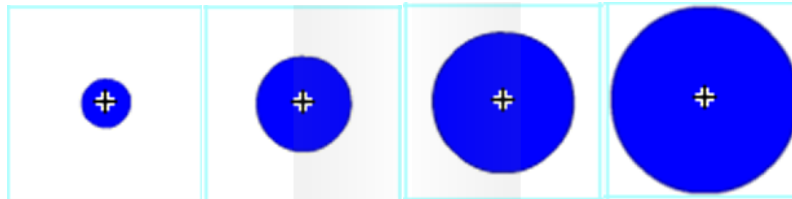
Şekil 7.23. Sembol 2 içerisinde şekil animasyonu tekniğinin uygulanması ve ikinci anahtar karenin tasarımı

3. Adım: Bu adımda aynı katman üzerine yeni bir şekil animasyonu oluşturulacaktır. Bu işlem için Şekil 7.24'te gösterilen zaman çizelgesindeki 40. kare üzerine gelip boş bir anahtar kare ekleyiniz. Eklenen bu kare içerisine bir dikdörtgen çizerek animasyonu tamamlayınız.



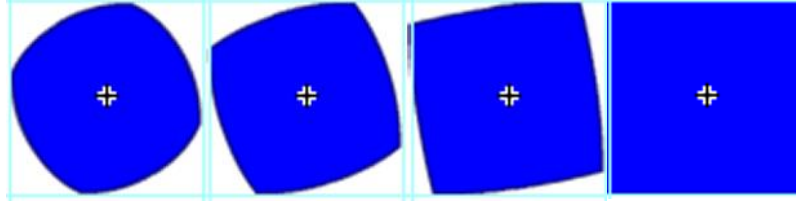
Şekil 7.24 Sembol 2 içerisinde farklı film karelerinde şekil animasyonu için son anahtar karenin tasarlanması

5.film karesi 10.film karesi 15. film karesi 20.karesi



Kare hızının 24 fps olması yeni Flash belgelerinde varsayılan ayardır ve genellikle web'de en iyi sonuçları verir.

25. film karesi 30. film karesi 35. film karesi 40. film karesi



Şekil 7.25 Sembol 2 ye ait isimleri verilen film karelerindeki Flash tarafından yapılandırılmış grafik nesneleri

Şekil animasyonu uygulanmış nesneler, bulunduğu yerde canlandırıldığı için zaman çizelgesinin “döngü kaplama” özelliği ile çalışma alanında gösterilememiştir. Bunun yerine animasyon uygulandığında Adobe Flash tarafından hesaplanarak oluşturulmuş film karelerindeki nesnelerin tümü olmasada isimleri yazan film kareleri Şekil 7.25’te gösterilmiştir.

Bu ünite de anlatılanların dışında ActionScript programlama dilinde script yazarak da animasyon oluşturabilirsiniz. ActionScript’in, çizgiler ve şekilleri script kullanarak oluşturulmasını sağlayan Drawing API adında bir bölümü vardır. Her ne kadar bu dersin kapsamı dışında olsa da, ActionScript kullanmayı tercih ediyorsanız bu konuyu incelemeniz faydalı olacaktır.



Özet

- Animasyon için tasarım yapan sanatçılara animatör denir. Animatörlerin görevi animasyon prensiplerini kullanarak birbiri arkasına gelen resimlerdeki karakterleri tasarlayarak canlı gibi görünmesini sağlamaktır.
- Grafik ya da resim nesnelerinden oluşan film karelerinin art arda ve belirli bir hızla gösterilmesi sonucu oluşan hareketli görüntülere animasyon denir.
- Web sitelerinde kullanılan animasyonların hazırlanmasında genellikle Adobe flash programı kullanılır, çünkü bu program dosya boyutunu küçük tutarak yüksek kalitede animasyonlar oluşturmasını sağlar.
- Adobe Flash kare kare, hareket ve şekil animasyonları olmak üç tür animasyon yapım tekniğini destekler.
- Kare kare animasyon oluşturulması uzun zaman alan bir animasyon yapım tekniğidir. Bu animasyonu oluşturan her bir film karesi tek tek oluşturulup, elle düzenlenir. Bu yüzden arası doldurulan animasyona göre daha fazla dosya boyutuna sahiptir.
- Hareket Animasyonları (Motion Tween) Adobe Flash'ta en çok kullanılan nesne hareketlendirme tekniğidir. Hareket sağlayan bu animasyon yapım şeklinde siz iki kare belirlersiniz. Nesnenin ilk karedeki ve son karedeki durumunu Adobe Flash kullanarak, bu iki kare arasındaki gereken kareleri kendi oluşturur ve izlediğinizde, o nesne ilk karedeki konumundan, son karedeki konuma gidiyormuş gibi ekranda izlenir.
- Eğer isterseniz, hareket ettireceğiniz sembolün veya grubun yolunu sadece düz yol olarak değilde istediğiniz gibi bir yol çizerek de oluşturabilirsiniz. Bunu yapabilmek için, normal bir hareket animasyonu oluşturduğunuz katmanın üzerine “Klasik Hareket Kılavuzu Ekle” komutu yardımıyla istediğiniz yolu belirleyerek animasyonun sizin çizdiğiniz yolu takip etmesini sağlayabilirsiniz.
- Adobe Flash içerisinde yer alan hareket ön ayarları panelini kullanarak kolayca animasyon tasarlanabilir. Bu panel yardımıyla önceden nesnelere uygulanan tekniklerdeki animasyonlar kaydedilebileceği gibi içerisinde yer alan 32 tane varsayılan ön ayarlar kullanılarak da animasyon oluşturulabilir.
- Şekil animasyonları döndürme, ölçekleme, eğme gibi dönüşüm işlemlerini kullanarak elde edemediğimiz değişimleri oluşturmak için kullanılan bir tekniktir. Bu animasyon oluşturma tekniğinde; bir nesnenin belirli bir süre içerisinde başka bir nesneye dönüşmesi sağlanır.
- Animasyonun oynatıldığı hız, saniyedeki kare sayısı ile ölçülür (fps). 24 fps'lik kare hızı, yeni Flash belgelerinde varsayılan ayardır ve genellikle web'de en iyi sonuçları verir.

DEĞERLENDİRME SORULARI

1. Aşağıdaki seçeneklerden hangisi bir sembolün başka bir sembol içerisinde kullanılmasını sağlamak için kullanılır?
 - a) Zaman Çizelgesi
 - b) Özellikler
 - c) Kütüphane
 - d) Hizala
 - e) Bilgi
2. Aşağıdaki seçeneklerden hangisi kullanılarak seçili bir animasyon karesi yanına normal kare ekleme” işlemi yapılır?
 - a) Düzenle > Zaman Çizelgesi > Kareleri Kaldır
 - b) Ekle > Zaman Çizelgesi > Kare
 - c) Değiştir > Zaman Çizelgesi > Ana Kare
 - d) Ekle > Zaman Çizelgesi > Boş
 - e) Ekle > Zaman Çizelgesi > Anahtar Kare
3. Aşağıdaki seçeneklerden hangisi Adobe Flash’ın desteklediği animasyon türü değildir?
 - a) Ara hareket
 - b) Klasik ara
 - c) Şekil arası
 - d) Kare kare
 - e) Süper hareket
4. Aşağıdaki seçeneklerden hangisi hareket kılavuzun tanımı olarak verilebilir?
 - a) Animasyonun sahne üzerindeki koordinatlarını verir.
 - b) Sahne üzerinde nesnelerin görüldüğü bölgedir.
 - c) Zaman çizelgesindeki son kareyi verir.
 - d) Nesnenin Sahne üzerinde izleyeceği yolu belirler.
 - e) Katmanlar içerisinde en üsteki katmanı gösterir.
5. Animasyon için tasarım yapan sanatçılara (.....) denir. Aşağıdaki seçeneklerden hangisi nokta ile verilen boşluğu doldurmak için kullanılabilir?
 - a) Grafiker
 - b) Animatör
 - c) Armatör
 - d) Programcı
 - e) Yönetmen

6. Hareket animasyon tekniğinde, sabit bir zaman aralığında nesnenin aşağıdaki seçeneklerde verilen hangi özelliğini değiştirmek için kullanılmaz?
- Konumu
 - Ölçeklendirmesi
 - Döndürülmesi
 - Alfa (Şeffaflık)
 - Şekli
7. Animasyonun oynatıldığı hız, saniyedeki kare sayısı ile ölçülür. Aşağıdaki seçeneklerde verilen hangi kare hızı, yeni Flash belgelerinde varsayılan ayardır ve genellikle web'de en iyi sonuçları verir?
- 16
 - 24
 - 4
 - 3
 - 43
8. Animasyonları her zaman film karelerini tek tek tasarlamak yerine Adobe Flash içerisinde yer alan (.....) paneli kullanılarak da kolayca tasarlanabilir. Nokta ile verilen boşluğu aşağıdaki seçeneklerden hangisi doğru olarak doldurur?
- Kütüphane
 - Bileşenler
 - Dönüştür
 - Özellikler
 - Hareket önayarıları
9. () Simgesi aşağıdaki seçeneklerde verilen hangi alanı temsil eder?
- Canlı görüntüleme
 - Kayıtlı ayarları ara
 - Varsayılan veya kaydedilmiş ayarları seç
 - Seçimi ön ayar olarak kaydet
 - Yeni klasör oluştur
10. Aşağıdaki seçeneklerde verilen animasyon tekniklerinden hangisi en fazla dosya boyutuna sebep olur?
- Klasik ara animasyonları
 - Şekil arası animasyonları
 - Kare kare animasyonlar
 - Hareket ön ayarlarıyla yapılan animasyonlar
 - Ara Hareket animasyonları

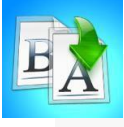
Cevap Anahtarı

1.c, 2.c, 3.e, 4.d, 5.b, 6.e, 7.b, 8.e, 9.d, 10.c

YARARLANILAN KAYNAKLAR

- [1] Adobe Flash Pro CC, Yardımcı el kitabı. <http://helpx.adobe.com/tr/support>
[Erişim tarihi: 30 Ekim 2013].
- [2] SCHULZE Pati.(2004) Macromedia Fireworks 8 Kaynağından Eğitim,
Medyasoft Yayınları.

OPTİMİZASYON



İÇİNDEKİLER

- Optimizasyon
- Genel Animasyon Optimizasyonu
- Element Ve Çizgi Optimizasyonu
- Yazı Karakterleri Optimizasyonu
- Renk Optimizasyonu



HEDEFLER

- Bu üniteyi çalıştıktan sonra;
- Projenin kalitesi için en iyi seçeneklerin neler olduğunu öğrenebilecek,
- En iyi kalitede oluşacak problemlerin neler olduklarını kavrayabilecek,
- Kaliteden fazla taviz vermeden, problemleri ortadan kaldırmayı öğrenebileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

ANİMASYON TASARIMI

Öğr.Gör.
Suavi SERTCAN

ÜNİTE 8

GİRİŞ

Adobe Flash belgelerinin erişilebilirliğini artırmak, diskte kapladığı alanı küçülmek ve çalışmasını hızlandırmak, kısacası mükemmeli yakalamak için yapılan yayınlama öncesi çalışmalara *optimizasyon* denir.

Optimizasyon, işlerin daha iyi yürütülmesi anlamına gelmektedir. Adobe Flash belgelerinde yapılacak optimize işlemleri 4 ana başlık altında toparlanabilir. Bunlar:

- Genel animasyon optimizasyonlar,
- Elementler ve çizgilerin optimizasyonu
- Renklerin optimizasyonu
- Yazı karakterlerinin optimizasyonu

Bu ünite, Adobe Flash uygulamalarını verimli kullanabilmek için bazı iyileştirme çalışmaları (optimizasyon) uygulayacak, bu optimizasyonların niçin yapılması gerektiği, nasıl yapılacaklarını ve optimizasyon sonrasında ne tür avantajlar elde edileceği anlatılacaktır.

OPTİMİZASYON

Adobe Flash projelerinin optimizasyonu projenin geliştirilmesi kadar önemlidir. İnternet üzerinden tasarımınıza ulaşan her kullanıcının bağlantı hızı birbirinden farklıdır. Bazıları yüksek band genişliğine sahipken bazıları da sınırlı band genişliğine sahip olabilir. Bunun için SWF dosyanızın mümkün olduğu ölçülerde küçük tutulması projenizi gezen kullanıcıların zaman kaybına sebep olmayacaktır. Ayrıca projenizi gezen kullanıcılar, ilgilenmedikleri kısımların indirilmesini beklemek zorunda bırakılmamalıdır. Bunun için SWF dosyalarının daha küçük parçalara bölünmesi de yine kullanıcıların avantajına olacaktır. Adobe Flash projenizde kullanmış olduğunuz varlıkları daha hafif bir hâle getirerek çıktı SWF dosyalarınızı daha küçük hâle getirebilirsiniz.

İyi planlanmış bir Adobe Flash projesi, optimizasyon işlemlerine ihtiyaç duymaz. Çünkü planlama aşamasında optimizasyon çalışmalarının neler olacağı bilindiği için bunlar yapım aşamasında projeye uygulanır.

Bilgilerin nasıl yüklendikleri incelendiğinde Web projelerinin birçok farklı yapıda olduklarını görürüz. İyi planlanmamış bir Web projesinde bilgilerin tamamı dev gibi bir FLA dosyasına konulmuştur. Sitede olması gereken Şekiller, sesler, vs. kullanıcılar bu siteye bağlandığında bu bilgilerin tamamı kullanıcının bilgisayarına indirilir. Bu da zaman kaybı demektir ve kullanıcıların bu sürede sıkılmalarına ve belki de siteye bağlanmaktan vazgeçmelerine sebep olabilir.

Web projelerinin tamamını bir tek FLA dosyasına yerleştirmek yerine, proje bilgilerinin büyük bir kısmını çalışma zamanında dinamik olarak yükleyen bir site oluşturmak kullanıcıların daha fazla tercih edecekleri bir tasarım olacaktır. Optimizasyon yapım şekilleri aşağıdaki gibi incelenebilir.

GENEL ANİMASYON OPTİMİZASYONLARI

Bitmap'ı ön belleğe alarak performansını arttırma

Çalışma zamanı bitmap ön belleğe alma, statik bir film klibi (örneğin, bir arka plan görüntüsü) veya düğme sembolünün çalışma zamanında bir Bitmap olarak ön belleğe alınacağını belirtmeniz, oynatma performansını iyileştirmenizi sağlar. Varsayılan olarak, Flash Player Sahnede her bir karedeki tüm vektör öğeleri yeniden çizer.



Web sitenize bir giriş bölümü (intro) koyarsanız, mutlaka girişi atla (skip intro) tuşu da koyun. Aksi takdirde kullanıcıları giriş bölümünü izlemeye zorlamış olursunuz.

Film klibini veya düğme sembolünü bitmap olarak ön belleğe alma, Flash Player'ın onları sürekli yeniden çizmesini engeller çünkü görüntü bitmap' tir ve sahnedeki konumu değişmez. Bu oynatma performansında önemli bir gelişme sağlar. Örneğin, karışık arka planlı bir animasyon oluşturduğunuzda, arka plandaki tüm öğeleri içeren bir film klibi oluşturun. Daha sonra Özellik Denetçisinde artalan film klibi için Bitmap olarak ön belleğe al seçeneğini seçin. Oynatma sırasında arka plan, geçerli ekran derinliğinde saklanan bir bitmap olarak çizilir. Flash Player sahnede Bitmap'ı hızlıca ve bir defaya mahsus çizer, bu da animasyonun daha hızlı ve daha pürüzsüz biçimde oynamasını sağlar.

Bitmap'ı ön belleğe alma olmadan yapılan animasyonlar çok yavaşça oynatılır. Bitmap ön belleğe alma, bir film klibi kullanıp onu otomatik olarak yerinde dondurmanızı sağlar. Bir bölge değişirse, vektör verisi bitmap ön belleğini günceller. Bu süreç, Flash Player'ın yapması gereken yeniden çizimlerin sayısını en aza indirir ve daha düzgün, daha hızlı oynatma performansı sunar.

Çalışma zamanı Bitmap ön belleğe almayı, animasyondaki her karede film klibinin içeriğinin değil, konumunun değiştiği statik, karmaşık film klipleri üzerinde kullanın. Çalışma zamanı bitmap ön belleğe almayı kullanmanın sonucunda oluşan oynatma veya çalışma zamanı performansı iyileştirmesi, sadece karmaşık içerikli film kliplerinde fark edilebilir. Basit film klipleriyle çalışma zamanı Bitmap ön belleğe alma, performansı geliştirmez.

Adobe Flash programında bu işlemlerin yapılma sırası aşağıda verilmiştir.

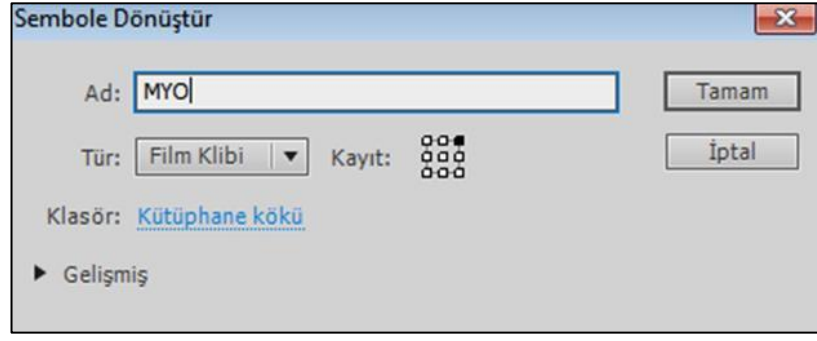
- Seçim aracı ile nesneyi seçiniz. Birden fazla nesneyi seçmek için Ctrl tuşunu kullanınız.
- Şekil 8.1'de görüldüğü gibi *Değiştir* isimli menüden *Sembolle dönüştür* komutunu veriniz. Açılan pencereden sembol ismi vererek tamam düğmesine basınız (Şekil.8.2).
- *Özellikler* penceresinden gerekiyorsa takas düğmesini kullanarak daha önce oluşturduğunuz sembolü seçiniz. Görüntü başlığı altında bulunan *oluştur* seçeneğinde bulunan *Bitmap olarak Ön Belleğe Al* seçeneğini onaylayınız (Şekil.8.3).



Flash programının birçok versiyonunda 'Bitmap Olarak Ön Belleğe Al' seçeneği Sembolle dönüştür penceresinin sağ alt kısmındaki kutucuğunun onaylanması ile yapılır.



Şekil 8.1 Seçili nesneyi sembole dönüştürme



Şekil 8.2 Sembol Dönüştür iletişim penceresi

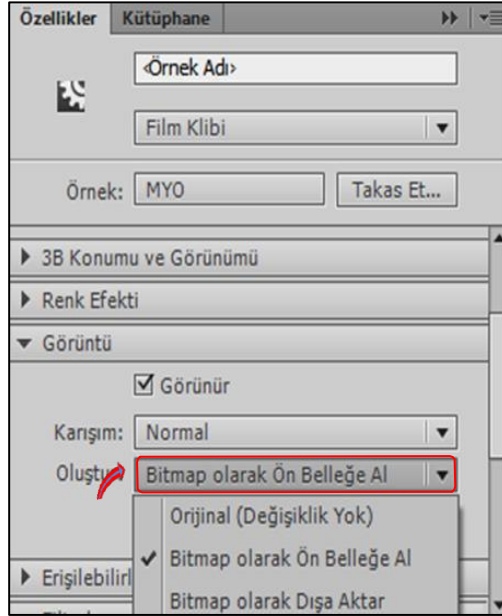
Çalışma Zamanı Bitmap Önbelleğe Alma Kullan seçeneğini sadece film klibi ve düğme sembolleri için kullanabilirsiniz. Çalışma Zamanı Bitmap'ı Önbelleğe Alma seçili olsa bile aşağıdaki koşullarda, bir film klibi bitmap olarak kullanılamaz.

- Bitmap çok büyükse (herhangi bir yönde 2880 pikselden daha büyükse.)
- Flash Player bitmap için bellek ayırmayı başaramazsa (yetersiz hafıza hatası üreterek).

Bu durumlarda, bunun yerine film klibi veya düğme sembolünü vektör verisi kullanarak çizer.



Animasyon içeren film kliplerinde ve hareketli film klipler içeren düğmelerde "Çalışma Zamanında Bitmap'ı Önbelleğe Al" özelliği aktif değildir.



Şekil 8.3 Bitmap Olarak Ön Belleğe Al seçeneği

ELEMENT VE ÇİZGİ OPTİMİZASYONU

Gereksiz tutturma noktalarını silmek

Kalem ile çizilmiş şekillerde bir yoldaki gereksiz tutturma noktalarını silmek, projenizin performansını en iyi duruma getirir ve oluşturulan SWF dosyasının boyutunu azaltır. Bu işlem Adobe Flash içerisinde iki yöntemle yapılır.

Birinci yöntemde;

Bir köşe noktasını silmek için noktayı Kalem aracıyla bir defa tıklatın. Seçili çizgi parçasından bir tutturma noktası silinebiliyorsa, Kalem aracının yanında bir eksi (-) işareti görünür. Çizgi parçası henüz seçili değilse, seçmek için Kalem aracıyla tıklatın ve sonra tutturma noktasını silin.

Düz bir noktayı silmek içinse noktayı Kalem aracıyla bir defa tıklatın. Seçili çizgi parçasından bir tutturma noktası silinebiliyorsa, Kalem aracının yanında bir eksi (-) işareti görünür. Çizgi dilimi henüz seçili değilse, seçmek için Kalem aracıyla

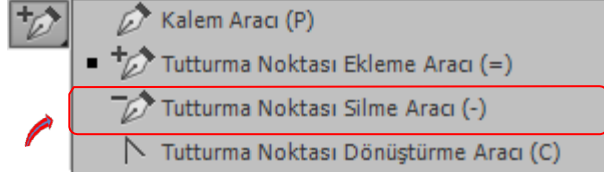
Optimizasyon

tıklatın ve sonra köşe noktasını silin. (Noktayı bir köşe noktasına dönüştürmek için bir defa, noktayı silmek için bir defa daha tıklatın.)

İkinci yöntem ise Şekil 8.4'te görüldüğü gibi kalem aracının altında gizli olan **Tutturma Noktası Silme Aracı (-)**'ni seçiniz. Gereksiz gördüğünüz tutturma noktalarına tek tek tıklamak suretiyle onları siliniz. Bunları yaparken çiziminizde arzu etmediğiniz değişiklikler olursa işleminizi geri alınız.

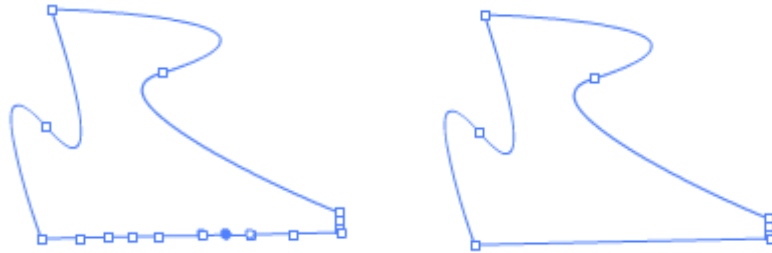


Düzgünleştir, düzleştir, gelişmiş düzgünleştir, gelişmiş düzleştir ve en iyileştirme komutlarına, **Değiştir** isimli menüden **Şekil** alt başlığı kısmından ulaşabilirsiniz.



Şekil 8.4 Tutturma noktası silme aracının seçimi

Şekil 8.5'te ise gereksiz tutma noktaları olan bir şekil ve gereksiz tutturma noktalarının silinmiş hali görülmektedir.



Şekil 8.5 Gereksiz tutturma noktalarının silinmesi

Eğrilere en iyileştirme uygulanması

Düzgünleştir, düzleştir, gelişmiş düzgünleştir, gelişmiş düzleştir ve en iyileştirme komutları eğri çizgileri ve dolgu ana hattını arıtarak, eğrileri düzgünleştirir, böylece bu öğeleri tanımlamak için kullanılan eğrilerin sayısını azaltır. En iyileştirme ayrıca, Flash Professional belgesinin (FLA dosyası) ve dışa aktarılan Flash Professional uygulamasının (SWF dosyası) boyutunu azaltır. Aynı öğelere en iyileştirme birçok defa uygulanabilir.



Şekil 8.6 Şekil düzleştirmenin öncesi ve sonrası

Öneriler:

- Öğeleri gruplayın.
- Animasyon sırasında değişen öğeleri değişmeyen öğelerden ayırmak için katmanları kullanın.
- Şekilleri tanımlamak için kullanılan ayrı çizgilerin sayısını en aza indirmek amacıyla **Değiştir > Şekil > En İyileştir**'i kullanın.
- Kesikli, noktalı veya düzensiz gibi özel çizgi türlerinin sayısını sınırlandırın. Düz çizgiler daha az bellek gerektirir. Kurşun Kalem aracıyla oluşturulan çizgiler,

Optimizasyon

fırça konturlarından daha az bellek gerektirir.

- Bir SWF dosyasında kullandığınız çizgilerin ve noktaların sayısını azaltın. Bir çizimdeki vektörlerin sayısını azaltmak için Eğrileri En İyileştir iletişim kutusunu (*Değiştir > Şekil > En İyileştir*) kullanın. Daha fazla en iyileştirme için Birden Fazla Geçiş Kullan seçeneğini seçin. Bir grafiği en iyileştirmek dosya boyutunu azaltır ancak çok fazla sıkıştırmak kalitesinden ödün verir. Ancak eğrileri en iyileştirme dosya boyutunuzu azaltır ve SWF dosyası performansını iyileştirir.

RENK OPTİMİZASYONU

Renkleri iyileştirme

JPEG biçimi, bir FLA dosyasını yüksek oranda sıkıştırılmış, 24-bit bitmap olarak yayınlanmasına imkân sağlar. Genelde, GIF biçimi çizgi çizimlerini dışa aktarmak için ve JPEG biçimi de fotoğraflar, degradeler veya gömülü bitmap'lar gibi sürekli tonlu görüntüler için daha iyidir.

Kaydırıcıyı sürükleyin veya JPEG dosya sıkıştırma miktarını kontrol etmek için bir değer girin. Görüntü kalitesi ne kadar düşükse, dosya boyutu da o kadar küçüktür. Boyut ve kalite arasındaki dengeyi belirlemek için farklı ayarlar denenmelidir.

Öneriler:

- Tek bir sembolün farklı renklerde birçok örneğini oluşturmak için Sembol Özellik denetçisindeki Renk menüsünü kullanın.
- Belgenin renk paletini tarayıcıya özgü bir paletle eşleştirmek için Renk panelini (*Pencere > Renk*) kullanın.
- Degradeleri tutumlu olarak kullanın. Bir alanı degrade rengiyle doldurmak, onu düz renkle doldurmaya göre 50 bayt kadar daha fazla dosya boyutu oluşmasına neden olur.
- Bitmap'leri fazla sıkıştırmadan en iyileştir seçeneğini kullanın. 72 dpi 'lik bir çözünürlük web için en uygundur. Bir bitmap görüntüsünü sıkıştırmak
- dosya boyutunu azaltır ancak çok fazla sıkıştırmak grafiğin kalitesinden ödün verir. Yayınlama Ayarları iletişim kutusundaki JPEG kalitesi ayarlarının görüntüyü fazla sıkıştırmadığını denetleyin. Bir görüntüyü bir vektör grafiği olarak temsil etmek çoğu durumda tercih edilebilirdir.
- Vektör görüntülerini kullanmak dosya boyutunu azaltır, çünkü görüntüler birçok piksel yerine hesaplamalardan yapılır. Kaliteyi koruyarak, görüntünüzdeki renklerin sayısını sınırlandırın.

YAZI KARAKTERLERİNİN İYİLEŞTİRİLMESİ

Kenar yumuşatma, ekran üzerindeki metnin köşelerini düzgünleştirmenizi sağlar. Kenar yumuşatma seçenekleri özellikle daha küçük font boyutlarını oluşturmada etkilidir. Kenar yumuşatma etkin olduğunda, geçerli seçimdeki tüm metin etkilenir. Kenar yumuşatma, tüm nokta boyutlarındaki metinlerde aynı şekilde işler. Flash Professional metin oluşturma araçları, Flash Professional (FLA) belgeleri ve yayınlanmış SWF dosyalarında temiz, yüksek kalitede metin oluşturma seçeneği sağlar. Okunabilirlik için Kenar Yumuşatma ayarı, özellikle küçük boyutlu fontlarda metni daha okunaklı hâle getirir. Özel kenar yumuşatma, ayrı metin alanlarında kullanılan fontların kalınlık ve keskinliklerini belirtmenizi sağlar. Yüksek kaliteli kenar yumuşatma, Flash Player 8 veya daha sonrasına her yayınladığınızda ve Okunabilirlik



Bitmap 'ları orijinal boyutlarından daha geniş bir şekilde ölçeklemekten kaçınin, çünkü bu, görüntünün kalitesini azaltır ve işlemciyi yoğun olarak kullanır.

Optimizasyon

İçin Kenar Yumuşatma ya da Özel Kenar Yumuşatma seçildiğinde otomatik olarak etkinleşir. Okunabilirlik İçin Kenar Yumuşatma, özellikle bir Flash Professional belgesinin ilk karesinde dört veya beş farklı karakter kümesi kullanıyorsanız, Flash Professional SWF dosyalarını

yüklediğinizde küçük bir gecikmeye neden olabilir. Yüksek kaliteli kenar yumuşatma Flash Player'ın bellek kullanımını da artırabilir. Örneğin dört veya beş font kullanılması, bellek kullanımını yaklaşık 4 MB kadar artırabilir.

Öneriler:

- Fontların ve font stillerinin sayısını sınırlandırın. Dosya boyutunu artırdıklarından, gömülü fontları tutumlu olarak kullanın.
- Fontları Göm seçenekleri için fontun tamamını dâhil etmek yerine sadece gereken karakterleri seçin.



Özet

- Flash projelerinin kalitelerinin artırılması amacıyla yapılan bütün iyileştirme çalışmalarına optimizasyon denir. İyileştirme çalışmaları bir yandan kaliteyi artırırken bir yandan da dosya boyutunun büyümesine, filmin yavaşlamasına, indirme süresinin artmasına neden olmaktadır. Burada dengelerin iyi korunması gerekmektedir. Temel amaç, problemleri büyük ölçüde ortadan kaldırırken kaliteden en az miktarda ödün vermek olmalıdır. Bunun için Flash programının sunduğu seçenekler içerisinde en uygun olanları seçilmeli ve uygulanmalıdır.
- Problemler ve çözümleri önceden bilinirse yapım aşamasında bazı önlemler uygulamaya konulabilir. Filmin test edilmesi sırasında görülen problemler daha sonra giderilebilir. Bunun için Flash projesini her yeni ilaveden sonra test ederek neticesini görmek yararlı olacaktır.
- Optimizasyonlar, genel olarak Flash projelerinin dosya boyutunu düşürmek ve bilgisayarda bulunan mikroişlemci ve RAM'i yoğun olarak kullanmamak için uygulanırlar. Çünkü bunlar bilgisayarın yavaşlamasına sebep olur.
- Bir nesnenin her karede yeniden çizilmesi mikroişlemciyi fazlasıyla meşgul edeceği için, bunun yerine nesnenin görünümünde bir değişiklik olursa mikroişlemcinin bunu düzenlemesi daha uygun bir seçim olacaktır.
- Kullanılan renk sayısının azaltılması, görünüm kalitesinde düşme yapmıyorsa bu seçenek kullanılmalı, kalitede önemli sayılabilecek bozulmalar oluyorsa kullanılmayan renkler paletlerden kaldırılmalıdır.
- Kalemle yapılan çizimler elden geçirilmeli ve mikroişlemci ve RAM kullanımı en aza indirecek önlemler alınmalıdır.
- Çok fazla yazı fontu kullanarak dosya boyutlarını artırmak, dosya boyutunun gereksiz yere büyümesine sebep olur.

DEĞERLENDİRME SORULARI

1. Aşağıdaki seçeneklerden hangisi bir animasyon projenin iyileştirilmesine edilmesine yardımcı olmaz?
 - a) Birden fazla çizim nesnesini gruplandırmak
 - b) Birbirinden farklı yazı fontları kullanmak
 - c) Mp3 ses dosyalarını tercih etmek
 - d) Mp4 video dosyalarını tercih etmek
 - e) Çizimleri sembole dönüştürmek
2. Adobe Flash belgelerinin diskte kapladığı alanı küçültmek, çalışmasını hızlandırmak, kısacası mükemmeli yakalamak için yapılan yayınlama öncesi çalışmalara ne denir?
 - a) Optimizasyon
 - b) Yayınlama
 - c) Gruplama
 - d) Çevirme
 - e) Birleştirme
3. Adobe Flash belgelerinin hızlı açılmasını sağlamak için seçeneklerde verilen hangi işlem yapılamaz?
 - a) Yazı karakterlerinin iyileştirmesi
 - b) Renklerin iyileştirilmesi
 - c) Çizgilerin iyileştirilmesi
 - d) Şekilleri, bitmap olarak önbelleğe alma
 - e) Çok düşük bir kare hızının seçilmesi
4. Seçeneklerde verilen komutlardan hangisi çizim nesnelerindeki eğri çizgilerini ve dolgu ana hattını aratarak, optimizasyon yapılmasına sebep olmaz?
 - a) Düzgünleştir
 - b) Düzleştir
 - c) Gelişmiş düzgünleştirme
 - d) Gelişmiş düzleştirme
 - e) Döndür ve Eğrilt
5. Aşağıda verilen menülerden hangisiyle şekil optimizasyonun da kullanılan bölümler yer alır?
 - a) Değiştir
 - b) Ekle
 - c) Kontrol Et
 - d) Pencere
 - e) Komutlar

6. Bir animasyon dosyasına dışarıdan alınan *.jpg uzantılı görüntülerin web için optimize edildiğinde hangi kalitede olması gerekir?
- 100
 - 72
 - 40
 - 10
 - 32
7. Animasyonlarda optimizasyonları yaparken seçeneklerden hangisinin yapılması gerekli değildir?
- Oynatma süresi 10 saniyeden fazla olan videolarda aşamalı olarak indirilen video klipler kullanılır.
 - Seslere efektleri 16 bit olarak içe aktarılır.
 - Ses dosyalarını içeri aktarırken *.mp3 formatı tercih edilir.
 - Bir kereden fazla görünen her öge için semboller kullanılır.
 - Animasyon oluştururken, mümkün olduğu sürece kare kare animasyon tekniği kullanılır.
8. Yazı karakterlerinin optimizasyonunda hangi seçenek kullanılmalıdır?
- Proje içerisine fontun tamamını dâhil etmek yerine sadece gereken karakterler dahil edilmelidir.
 - Metin ekleme işlemlerinde parçala komutu kullanılmalıdır.
 - Statik metin yerine dinamik metin türü seçilmelidir.
 - Metinler üzerinde özel filtreler kullanılmalıdır.
 - Projelerde birden fazla yazı fontu tercih edilmelidir.
9. Animasyon projeleri optimize edilirken "Bitmap olarak Ön Belleğe Al" özelliği hangi tip nesnelerde kullanılabilir?
- Film klibi
 - Grafik
 - Ses
 - Video
 - Resim
10. Animasyonlarda tasarımlarında iyileştirme yaparken uygulanan önerilerden hangisi yanlıştır?
- Bir SWF dosyasında _alpha düzeyini 0'a veya 1'e değiştirmek yerine, visible özelliğini kullanın.
 - Anahtar karelerde değişikliğin küçük bir alanda meydana gelmesini sağlayın.
 - Fontların ve font stillerinin sayısını sınırlandırın.
 - Web sitesinin tamamını bir tek FLA dosyasına yerleştirin.
 - Animasyon dizileri oluştururken, mümkün olduğu sürece arası doldurulmuş animasyonları kullanın.

Cevap Anahtarı

1.b, 2.a, 3.e,4.e,5.a,6.b,7.e,8.a,9.a,10.d

YARARLANILAN KAYNAKLAR

- [1] Adobe Flash Pro CC, Yardımcı el kitabı. <http://helpx.adobe.com/tr/support>
[Erişim tarihi: 30 Ekim 2013].
- [2] SCHULZE Pati.(2004) Macromedia Fireworks 8 Kaynağından Eğitim,
Medyasoft Yayınları.

YAYINLAMA AYARLARI



İÇİNDEKİLER

- Yayınlama Ayarları
 - Dosya Biçimleri
 - Flash Player Sürümü
 - JPEG Kalitesi
 - Ses Akışı ve Olayları
- Gelişmiş Ayarlar
 - Filmi Sıkıştır
 - Gizli katmanları Dahil Et
 - İç Aktarmaya Karşı Korum



HEDEFLER

- Bu üniteyi çalıştıktan sonra;
 - Yayınlama mantığını anlayabilecek,
 - Değişik formatlarda dosyalar yayınlatabilecek,
 - Yayınlama ayarlarını kavrayabileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

ANİMASYON TASARIMI

Öğr.Gör.
Suavi SERTCAN

ÜNİTE 9



Karakter grubu tanımlanırken, * karakteri ile tanımlanır. Yazımızda * karakteri dosya ismini ifade etmektedir.

GİRİŞ

Adobe Flash programında hazırlanmış ve saklanmış filmler *.fla veya *.xfl uzantılı olarak saklanır. *.xfl dosyasının boyutu, *.fla dosyasından daha büyüktür. Çünkü *.fla dosyaları oluşturulurken bir miktar sıkıştırma olmaktadır. *.fla veya *.xfl olarak saklanmış filmler tekrar izlenmek veya yeniden düzenlenmek istenirse mutlaka Adobe Flash programı içerisinde açılmalıdır. Eğer Adobe Flash programının kurulu olduğu bir bilgisayarda bir flash filmi hazırlayıp, bu filmi başka bir flash programının kurulu olmadığı bilgisayarda izlemek istersek bu mümkün olmayacaktır. Bu nedenle flash filmleri taslak veya proje hâlindeki çalışmalardır. Film en son hâlini aldıktan sonra yayınlanmalıdır.

Yayınlama sonrasında Adobe Flash programında hazırlanmış filmler, isteğe göre *.swf, *.swc, *.html, *.gif, *.jpg, *.png, *.exe, *.app ve *.mov uzantılı dosyalar hâline dönüştürülebilir. Bu uzantılı dosyaları (filmleri) izlemek için Adobe Flash programının bilgisayarda kurulu olması gerekmemektedir. Böylece Adobe Flash programının kurulu olmadığı bilgisayarlarda da bu dosyaları izlemek mümkün olmaktadır. Ayrıca yayınlanmış filmlerde kullanıcının tercihlerine göre bir miktar da sıkıştırma olmaktadır. Böylece yayınlanmış filmler sabit disklerde daha az yer kaplarlar.

Ancak yayınlanmış olan bir film üzerinde tekrar düzenlemeler yapılmak istenirse, Adobe Flash programı içerisinde *.fla veya *.xfl dosyası açılmalı, gerekli görülen düzenlemeler yapıldıktan sonra tekrar yayınlanmalıdır.

Bu ünite yayınlanmanın nasıl yapılacağı, yayınlama sırasında karşılaşılabilecek ayarların ve seçeneklerin filminizi nasıl etkileyeceği anlatılacaktır.

YAYINLAMA AYARLARI (PUBLISH SETTINGS)

Önce Adobe Flash programında filminizi hazırlayınız. Sonra Dosya isimli menüyü açınız. Burada yayınlama ile ilgili iki tane komut satırı görülecektir. Bu komut satırları Şekil.9.1 'de görülmektedir.

Sadece **Yayınla** komut satırı seçilirse (Klavye kısayolu Shift+Alt+F12) hiçbir iletişim kutusu, ayar penceresi vb. açılmaz ve varsayılan ayarlar kullanılarak yayınlama yapılır. Varsayılan ayarlarda sadece *.swf ve *.html dosyaları oluşur. Bu dosyaların kayıt edildiği klasör ise yine varsayılan ayarlarda **Belirtilen Belgeler** isimli klasördür.

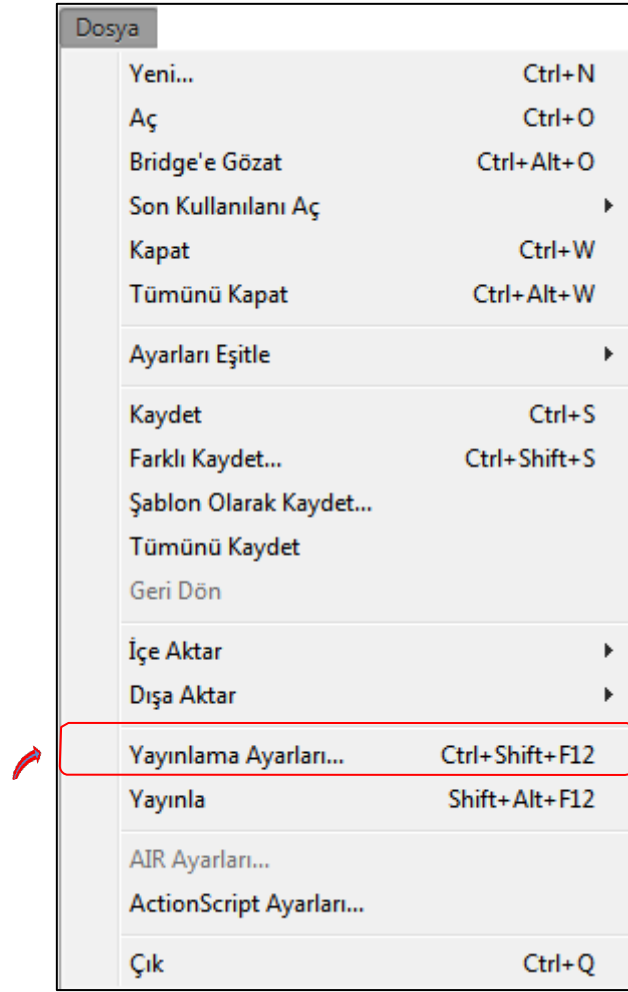
Yayınlama ayarlarını istediğinize göre yaptıktan sonra **Tamam** düğmesine basarsanız yaptığınız ayarlar saklanır. Sonra **Yayınlama Ayarları** penceresinden veya **Dosya** menüsünden **Yayınla** komutunu verdiğinizde sizin belirlediğiniz ayarlara göre yayınlama yapılacaktır.



*.fla uzantılı dosyalar sıkıştırılmış dosyalardır. *.xfl uzantılı dosyalarda sıkıştırma yoktur.



SWC uzantılı dosya oluşturma yeteneği Adobe Flash Professional CC ile beraber gelen bir özelliktir.

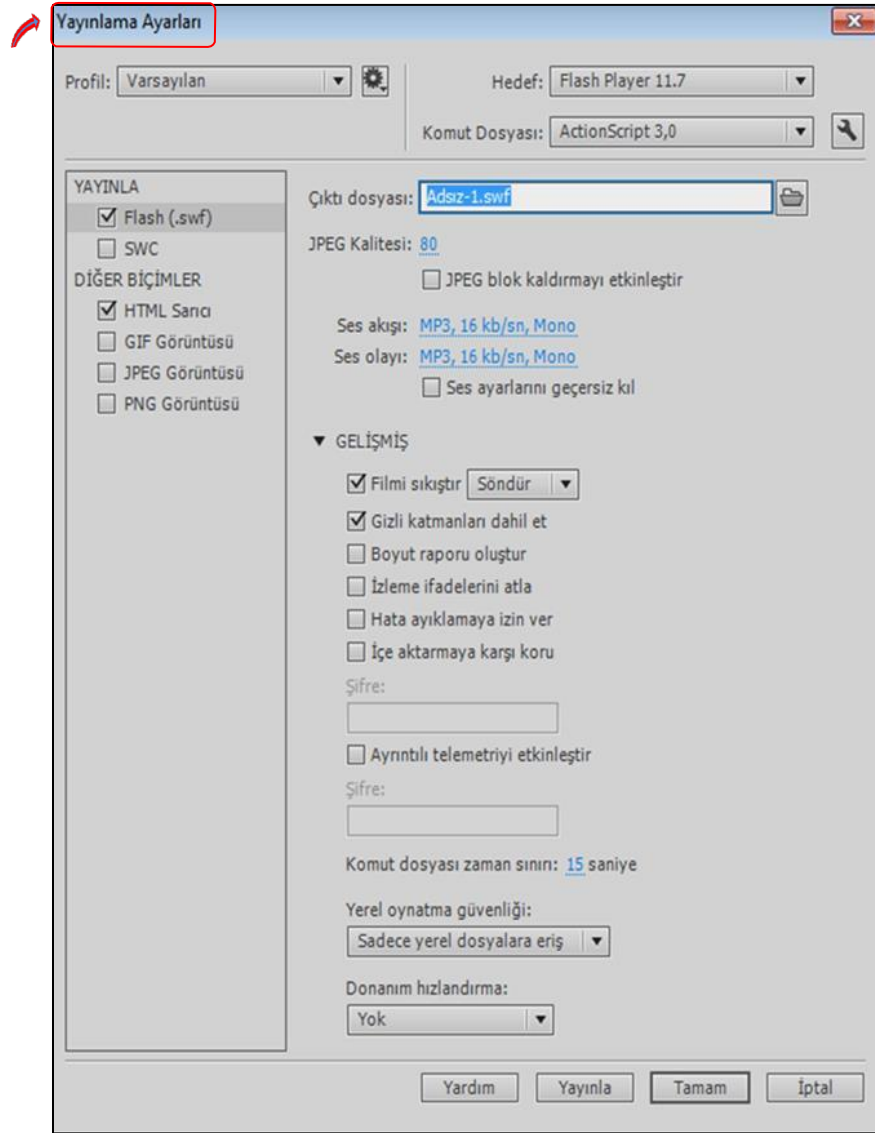


Şekil.9.1 Dosya menüsü

Yayınlama Ayarları... Komut satırı seçildiğinde (Klavye kısayolu Ctrl+Shift+F12) Şekil.9.2 'de görülen Yayınlama Ayarları penceresi açılır. Bu pencerede kullanıcının tercihinine göre birçok ayar ve seçenek bulunmaktadır. Şimdi bunların neler olduklarını sırasıyla inceleyelim.

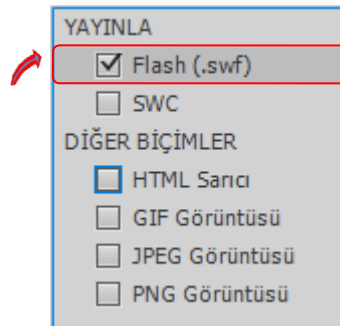
Profil olarak varsayılan ayarlar seçilidir. Varsayılan ayarlarda yalnızca *.swf ve *.html dosyaları istenilmektedir. Eğer istenirse diğer seçeneklerin önünde bulunan onay kutuları işaretlenerek *.swc, *.gif, *.jpeg ve *.png dosyaları da oluşturulabilir.

Bu uzantılar hakkında bir miktar bilgi sahibi olunması seçimler sırasında mutlaka yararlı olacaktır.



Şekil.9.2 Yayınlama Ayarları penceresi

SWF dosya biçimi:



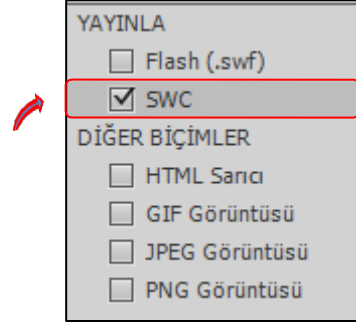
Şekil.9.3 SWF dosyası oluşturma ayarı

Şekil 9.3.te *.swf dosyasını oluşturmak için *Yayınlama Ayarları* penceresinde yapılması gereken ayar görülmektedir. *Adobe* firmasının Flash isimli internet eklentisinin dosyalarıdır. Bu türdeki dosyaları açmak için Shockwave Flash eklenti kullanılmaktadır. Bu tür dosyalarda genellikle basit uygulamalar ya da GIF tarzı şekiller bulunur. İnternette animasyon, video ve etkileşimli oyun sitelerinde oldukça fazla kullanılmaktadır. Çoğu internet tarayıcısı SWF formatını desteklemekle beraber, Flash Player tarafından oynatılabilir.



Eğer SWF dosyalarını FLA tipine dönüştürüp içeriğini değiştirmek isterseniz. Sothink SWF Decompiler programını kullanabilirsiniz.

SWC dosya biçimi:

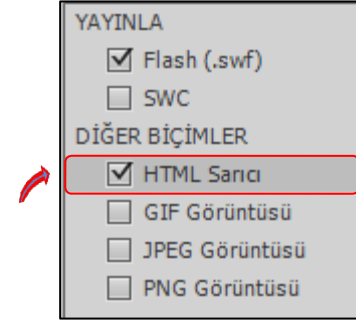


Şekil.9.4 SWC dosyası oluşturma ayarı

Şekil 9.4.te *.swc dosyasını oluşturmak için *Yayınlama Ayarları* penceresinde yapılması gereken ayar görülmektedir.

SWC dosyası, önceden derlenmiş Flash sembollerinden ve değiştirilemeyecek sembolleri ve kodu yeniden derlemenizi önleyen ActionScript kodundan oluşan bir pakettir. Örneğin; FLVPlayback ve FLVPlaybackCaptioning bileşenleri, SWC tabanlı bileşenlerdir.

HTML dosya biçimi:



Şekil.9.5 HTML dosyası oluşturma ayarı

HTML dosyası tek olarak oluşturulamaz. SWF dosyasını da oluşturmak gerekir.

Şekil 9.5.te *.html dosyasını oluşturmak için *Yayınlama Ayarları*

penceresinde yapılması gereken ayar görülmektedir.

Zengin Metin İşaret Dili (İngilizce *Hyper Text Markup Language*, ks. HTML) günümüzde İnternet üzerinden veri paylaşımı için kullanılan en yaygın metin tabanlı dildir. Dilin son sürümü HTML5'tir.

HTML, bir programlama dili olarak tanımlanamaz. Zira HTML kodlarıyla kendi başına çalışan bir program yazmak mümkün değildir. Ancak bu dili yorumlayabilen programlar aracılığıyla çalışabilen programlar yazılabilir.

Programlama dili denilememesinin nedeni tam olarak budur.

Temel gereği yazı, görüntü, video gibi değişik verileri ve bunları içeren sayfaları birbirine basitçe bağlamak, buna ek olarak söz konusu sayfaların web tarayıcısı yazılımları tarafından düzgün olarak görüntülenmesi için gerekli kuralları belirlemektir.

HTML, W3C tarafından standartlaştırılmaktadır. HTML, "etiket" ismi verilen çeşitli başlıklardan oluşur. Aşağıdaki örnek

bu etiketler kullanılarak yazılmıştır.

Web kavramı, CERN'de bir bilgisayar programcısı olan Tim Berners-Lee'nin **HTML**

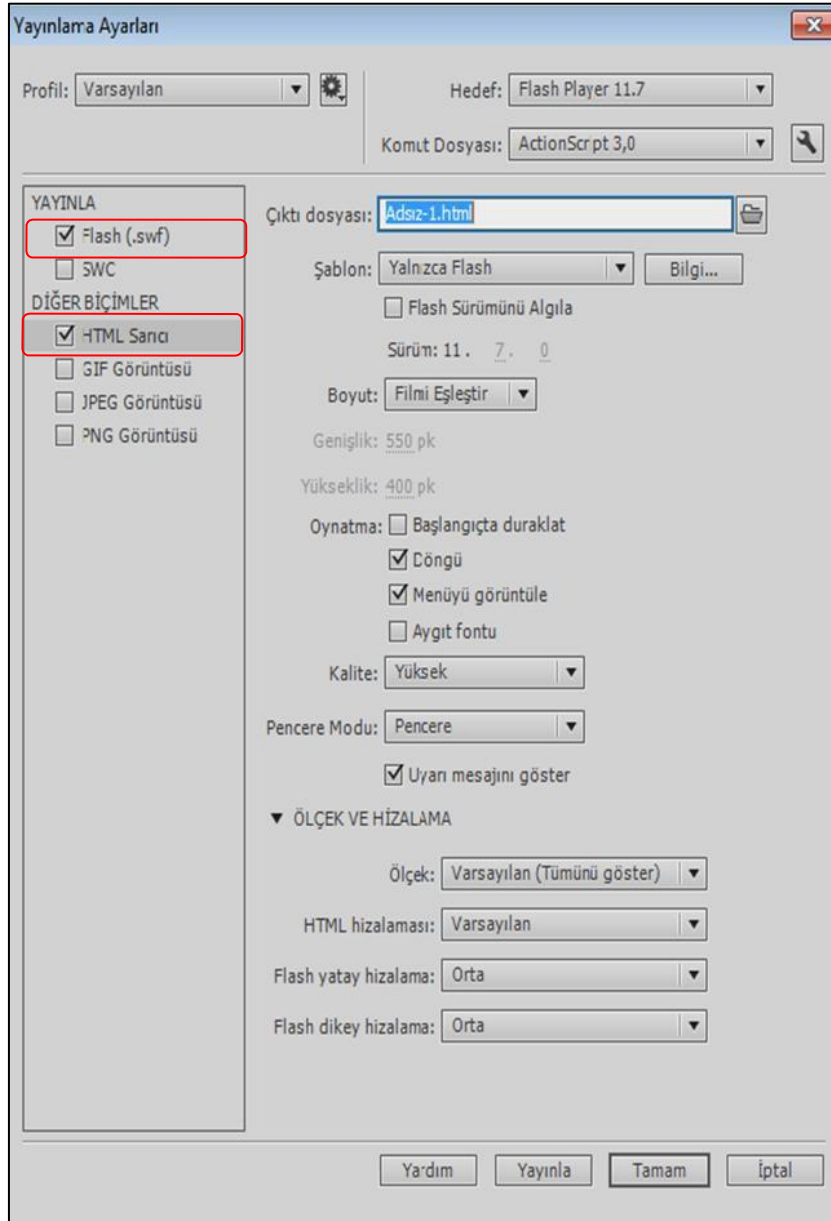
adlı bilgisayar metin dilini bulup geliştirmesiyle başlamıştır.

HTML ayarları:

Yayınlama Ayarları penceresi ilk açıldığında yayınlanacak dosyalar kısmında SWF ve HTML Sarıcı seçenekleri onaylı olarak açılır. Sağ tarafta ise bazı ayar seçenekleri mevcuttur ve bunların tamamı ilerleyen bölümlerde açıklanmıştır. Ancak HTML Sarıcı seçeneğinin onayı kaldırıldıktan sonra tekrar onaylanırsa sağ tarafta bulunan seçenekler HTML Sarıcı'ya göre değişim gösterir. Bu durum Şekil.9.6 'da gösterilmiştir.



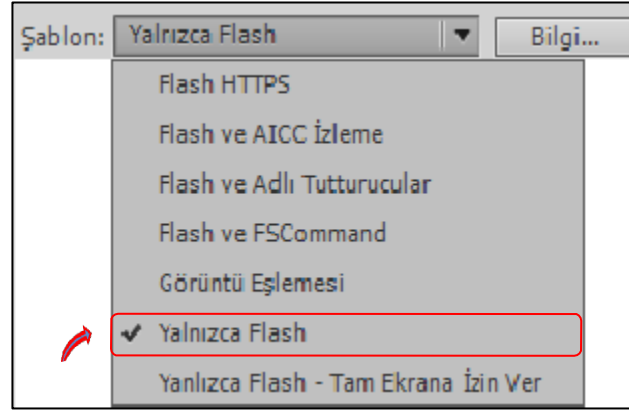
HTML ayarları, SWF dosyası için yapılan ayarları geçersiz hâle getirir.



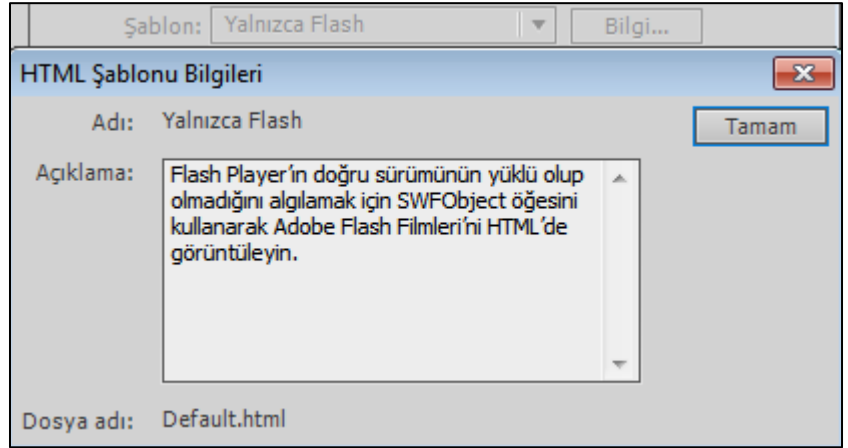
Şekil 9.6 HTML Sarıcı ayarları

Şablon

Varsayılan değer *Yalnızca Flash* seçeneğidir. Dışarı çıkartılan dosya içindeki kodların hangi biçimde verileceğini ayarlar. Menüde 7 tane seçenek vardır ve bunların gelişmiş olanları genellikle tarayıcı araştırma kodlarını içerir. Seçilen seçenek hakkında bilgi almak için *Bilgi* tuşuna basmanız yeterlidir. Şablon seçenekleri Şekil 9.7’de, şablon seçenekleri hakkında bilgi alınması ise Şekil 9.8’de gösterilmiştir.



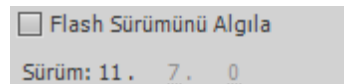
Şekil.9.7 Şablon seçenekleri



Şekil.9.8 Şablon seçenekleri hakkında bilgi almak.

Flash sürümünü algıla

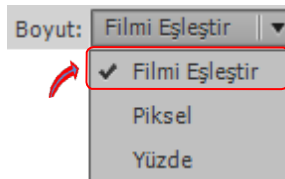
Bilgisayarda kurulu Adobe Flash Player sürümünün algılanması için kullanılır. Hemen altında ise bulunan Flash Player sürümü görülmektedir. *Flash Player Sürümü* bölümünde en yeni Flash Player sürümüne nasıl sahip olacağınız anlatılmıştır. Şekil 9.9'da Flash Sürümünü Algıla seçeneği görülmektedir.



Şekil.9.9 Flash Sürümünü Algıla.

Boyut

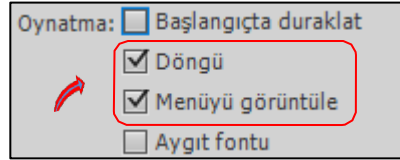
Dosyanın boyutunu ayarlar *Filmi Eşleştir* seçeneğinde hazırladığınız boyut korunur. Dosya boyutunu değiştirmek istiyorsanız *Piksel* veya *Yüzde* seçeneklerinden birini işaretleyip *Genişlik* ve *Yükseklik* kısımlarına gerekli değerleri yazmanız yeterlidir. Boyut seçenekleri Şekil 9.10'da gösterilmiştir.



Şekil.9.10 Boyut seçenekleri.

Oynatma

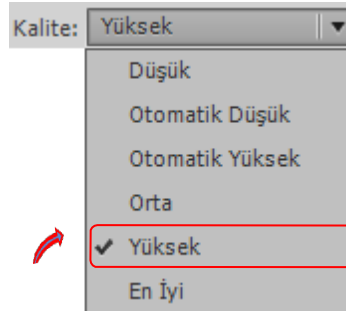
Özel istekler için kullanılır.. *Başlangıçta duraklat* animasyonun durmasını sağlar. Varsayılan değer değildir. *Döngü* işaretli ise animasyon bitince tekrar başa döner. Ama dosyayı hazırlarken bir stop komutu kullanmışsanız flash döngü seçeneğini dikkate almaz. *Menüyü görüntüle* seçeneği onaylı ise SWF dosyasına sağ tıklandığında kısa yol menüsü görünür. *Aygıt fontu* animasyondaki yazı karakterleri oynadığı bilgisayarda yoksa sistemde kayıtlı karakterlerden birinin kullanımını sağlar. Bu animasyonun boyutunu azaltacaktır. Windows için kullanılan bir seçenektir. Şekil 9.11’de oynatma seçenekleri gösterilmiştir.



Şekil.9.11 Oynatma seçenekleri.

Kalite

Filmin görüntü kalitesini belirler. Varsayılan değer *Yüksek*’tir. En iyi değer *En İyi*, en kötü değer ise *Düşük*’tür. *Otomatik Düşük*’te animasyon düşük kalitede başlar işlemci özellikleri yeterli ise *En İyi* kalitede devam eder. *Otomatik Yüksek*’te ise *En İyi* kalitede başlar işlemci yetersiz ise *Düşük* kalitede devam eder. Kalite seçenekleri Şekil.9.12’de gösterilmiştir.



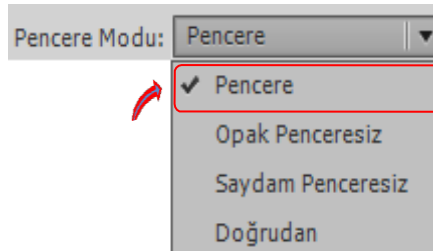
Şekil 9.12 Kalite seçenekleri.



Doğrudan modu; 3 boyutlu staj penceresi kullanır. Bu seçim yapılmadan önce mikroişlemcinin buna uygunluğu araştırılmalıdır.

Pencere modu

Filmin HTML sayfasında nasıl davranış göstereceğini belirler. Varsayılan *Pencere* seçeneği filmin arka planı ile önde oynayacağını belirler. *Opak Penceresiz* de arka planı ile fakat arkada oynar. Bu özellikle DHTML formatı ile beraber swf dosyalarının kullanıldığı durumda önemlidir. Bu seçenek işaretli olduğunda film arka planda oynayacağından DHTML kodları ile hazırlanan bir özellik, Flash filmi üzerinde görülebilir. *Saydam Penceresiz* seçeneğinde ise arka plan kaybolmuştur. Sadece nesneler HTML arka plan rengi üzerinde görülürler. Şekil 9.13’te pencere modu seçenekleri gösterilmiştir.



Şekil 9.13 Pencere modu seçenekleri.

Uyarı mesajını göster

HTML kodlarında herhangi bir uyumsuzluk varsa uyarı mesajı verilmesini sağlar. Şekil 9.14'te uyarı mesajını göster seçeneği gösterilmiştir.

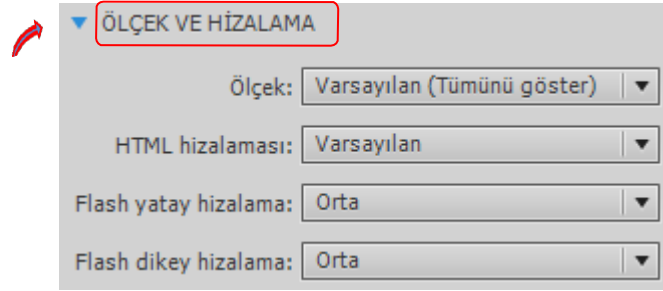
☒ Uyarı mesajını göster

Şekil 9.14 Uyarı mesajını göster seçeneği.

Ölçek ve hizalama

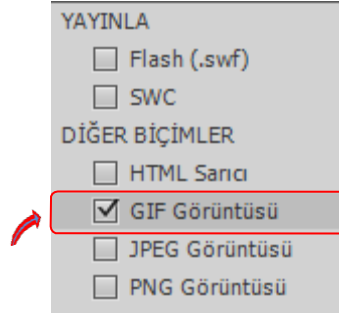
Ölçek: Dışarı çıkartılan animasyonun boyutlarını ayarlar. Varsayılan (Tümünü göster) seçeneğinde filminizin en ve boy oranı korunarak belirlenen çerçeveye filmin tamamı yerleştirilir. **Tam sığdır** seçeneğinde, en ve boy oranları korunmadan verilen çerçevenin tümüne filmi yerleştirir. **Kenarlık yok** seçeneğinde filmin en ve boy oranı değiştirilmeden çerçeve içine alınır ve fazla kalan kısımlar varsa sayfada gözükmezler.

Hizalama: Animasyon içindeki nesnelerin nereye yanaşık olacağını belirler. HTML hizalamasında varsayılan, sol, sağ, üst ve alt seçenekleri bulunur. Flash yatay ve dikey hizalamalarda sol, orta ve sağ seçenekleri bulunur. Şekil 9.15'te ölçek ve hizalama seçenekleri gösterilmiştir.



Şekil 9.15 Ölçek ve hizalama seçenekleri.

GIF dosya biçimi:



Şekil 9.16 GIF dosyası oluşturma ayarı

Şekil 9.16'da *.gif dosyasını oluşturmak için **yayınlama ayarları** penceresinde yapılması gereken ayar görülmektedir.

GIF, İngilizce **Grafik Değiştirme Biçimi** anlamına gelen **Graphics Interchange Format** kısaltmasıdır ve bir sayısal şekil saklama biçimidir. Kayıpsız sıkıştırma kullanır. 8-bit renge (yani 256 renge) kadar destek verir ve 1-bit'lik (yani tek renk için) saydamlık sunar. JPEG ile birlikte bilgisayar dünyasında kullanılan en yaygın şekil saklama biçimlerinden biridir. Genelde grafiklerin (az renk içermeleri dolayısıyla) saklanması için kullanılır.

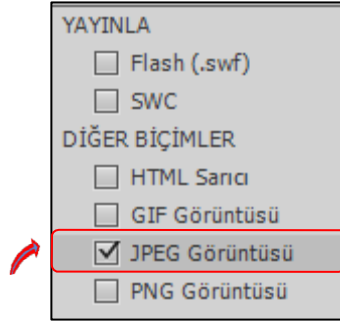
GIF, animasyonları küçük ve düşük çözünürlüklü film klipleri için kullanılabilir. Genel sınırlama dikkate alındığında 256 renkle GIF, genellikle dijital fotoğrafçılık için kullanılmaz. Dijital fotoğrafçılar görüntü dosyası formatları daha geniş bir dizi üretebilen TIFF, RAW veya JPEG gibi formatlar kullanır

Ölçek yok seçeneği Flash Player penceresi yeniden boyutlandırıldığında belgenin ölçeklendirilmesini engeller.

GIF, JPEG ve PNG türü seçimlerde, sağ tarafta bulunan ayarlar, seçilen dosya türüne göre değişir.

PNG ve JPEG'den farklı olarak, GIF formatı piksel tabanlı animasyonların üretilmesine olanak tanır.

JPEG dosya biçimi:



Şekil 9.17 JPEG dosyası oluşturma ayarı

Şekil 9.17 'de *.jpeg dosyasını oluşturmak için *yayınlama ayarları* penceresinde yapılması gereken ayar görülmektedir.

JPEG standardında görüntü saklayan dosya biçimi de çoğunluk tarafından JPEG olarak adlandırılır. Bu dosyalar genellikle *.jpg, *.jpe ya da *.jfif uzantılıdır, ancak çoğunlukla *.jpg uzantısı kullanılır. Ancak, JPEG standardı sadece görüntünün nasıl kodlanacağını tanımlar, görüntünün herhangi bir saklama ortamında depolanma biçimini belirtmez. JPEG olarak bildiğimiz dosya biçimi, Independent JPEG Group adlı başka bir grubun JFIF (JPEG File Interchange Format - JPEG Dosya Alışveriş Biçimi) adlı standardı tarafından tanımlanmıştır.

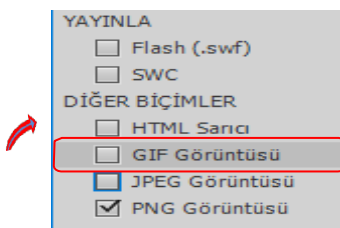
Bu dosya biçimi, WWW üzerinden görüntü iletmek ve fotografik görüntü saklamak için en popüler dosya biçimi olmuştur. JPEG / JFIF formatı, web için gerçekten de başarılı bir depolama ve veri transfer yapısına sahiptir. Çünkü bu işi yapan GIF formatı sadece 256 rengi desteklediğinden fotografik görüntülerde yetersiz kalmaktadır. PNG'nin görüntü kalitesi daha iyi olsa da web için yeterli boyut optimizasyonunu yapamadığı için büyük boyutlu şekiller ortaya çıkmaktadır. Bu da foto-grafik şekillerin web için kullanılacağı durumlarda da JPEG'i kullanışlı ve tercih edilir bir format hâline getirmektedir.

Genel dilde JPEG kısaltması, JPEG standardından çok JFIF dosya biçimine karşılık gelir. Ancak TIFF gibi başka dosya biçimleri de JPEG standardında görüntü saklayabilmektedir.

JPEG dijital kameralarda ve diğer foto-grafik görüntü yakalama cihazları tarafından kullanılan en yaygın görüntü biçimidir. JPEG / JFIF birlikte foto-grafik görüntülerin saklanması ve iletiminde World Wide Web üzerindeki en yaygın biçimdir. Diğer varyasyonlar seçkin değildirler, sadece JPEG olarak adlandırılır.

Dijital ortamda, JPEG dijital fotoğraf (Şekil) için kayıplı sıkıştırma yaygın olarak kullanılan bir yöntemdir. Depolama boyutu ve görüntü kalitesi arasında seçilebilir değişim sağlayan ayarlanabilir dereceli sıkıştırma değildir. JPEG genellikle görüntü kalitesinde az algılanabilir kaybı ile 1/10 oranında sıkıştırma sağlar.

PNG dosya biçimi:



Şekil.9.18 JPEG dosyası oluşturma ayarı



EXE, MOV ve APP uzantılı dosyaların yayınlanması Adobe Flash Prefessional CC versiyonunda kaldırılmıştır.

Şekil 9.18'de *.png dosyasını oluşturmak için *yayınlama ayarları* penceresinde yapılması gereken ayar görülmektedir.

Adobe Fireworks'un dosya uzantısıdır. PNG, "Taşınabilir Ağ Grafiği" anlamındaki (Portable Network Graphics) 'in kısaltmasıdır ve kayıpsız sıkıştırarak görüntü saklamak için kullanılan bir saklama biçimidir.

EXE dosya biçimi:

Executable, (Türkçesi çalıştırılabilir) uygulanabilir bir dosya, bilgisayar biliminde, içeriği bir bilgisayar ile bir program arasında yorumlanmak için ifade edilen bir dosyadır.

Sözlük anlamı çalıştırılabilir olmakla beraber bağımsız çalışabilen Windows Application dosya uzantısı (*.exe) uzatılmış hâlidir. DLL dosyaları executable bir uygulama olmaksızın kullanılamazken exe uzantılı uygulama dosyaları bağımsız çalışabilirler. Genellikle Microsoft Windows türü işletim sistemlerinde kullanılmalarına karşın Linux türevi işletim sistemlerinde de ek programlarla kullanılabilmektedirler. Fakat Linux türevi sistemlerde bu tür dosyalar, uzantılarıyla değil, uygulanabilir olma özelliği ile (x: executable) kendini belli eder.

MOV dosya biçimi:

< *QuickTime* > çeşitli biçimlerdeki sayısal görüntü, ses, yazı, animasyon, müzik, Şekil dosyalarını işleyen, Apple Computer tarafından geliştirilen bir çoklu ortam teknolojisidir.

Genel sanının aksine QuickTime sade bir oynatıcı olmaktan öte, bir çoklu ortam mimarisidir. Yeni içerik yaratma (planlanmış içeriğin elde edilmesi, kaydedilmesi), üretim (içeriğin dağıtılabilecek hale getirilmesi) ve dağıtım özelliklerine sahiptir. Bu süreçler kapsamında QuickTime'dan gerçek zamanlı yakalama, montaj, içe-dışa aktarma, sıkıştırma, son hazırlık, dağıtım, yayın ve ortam dosyalarını oynatma için yararlanır. Özetle profesyonel üretime yönelik bir teknolojidir, son kullanıcıların bildiği şekli QuickTime Player adıyla bilinen oynatıcıdır.

APP dosya biçimi:

Macintosh bilgisayarlar için projektör dosyaların uzantısıdır.

FLASH PLAYER SÜRÜMÜ

Adobe Flash programında hazırlanan filmler, yayınlama sonrasında genellikle Adobe Flash Player'da izleneceği için, kullandığınız Adobe Flash Player sürümü yayınlama sırasında oldukça önemlidir. *Yayınlama Ayarları* penceresinin, *Hedef* kısmında belirteceğiniz Adobe Flash Player sürümüne göre *komut dosyaları* kısmında uygun olan *Action Script* komutlar dosyası aktif olacaktır.



Örnek

- Adobe Flash Player 5 ve öncesi sürüm kullanılıyorsa, bununla sadece Action Script 1.0 komut seti kullanılabilir. Diğer komut setleri Adobe Flash Player'ın eski versiyonlarını desteklemezler.



Adobe Flash Player programının son sürümünün yüklemek için, Adobe Flash programı içerisinde Yardım menüsünü açın ve *En Yeni Flash Player'ı al* komutunu verin.

Adobe'nin önerisi Action Script 3.0 komut setinin kullanılması yönündedir. Bu komut seti ise Adobe Flash Player 10 ve sonrasını desteklemektedir.

Komut Dosyası açılır menüsünden Action Script sürümünü seçiniz. Action Script 2.0 veya 3.0'ı seçerseniz ve sınıflar oluşturduysanız, sınıf dosyalarına tercihler 'de ayarlı varsayılan izin yolundan farklılık gösteren göreceli sınıf yolunu ayarlamak için komut dosyası açılır menüsünün sağında bulunan ayar ikonuna tıklayın ve gerekli ayarları buradan yapın.

ÇIKTI DOSYASI:

Yayınlanacak dosyalarınızın ismini bu pencerede belirtebilirsiniz. Bunu yapmadığınız takdirde yayınlanacak dosyalarınızın ismi **Adsız-1** olacaktır. Daha sonra yayınlanacak dosyalarınızın ismi otomatik olarak **Adsız-2** olacaktır. Uzantıları ise **yayınla** kısmında seçim yaptığınız dosya türlerine göre oluşacaktır. Ayrıca karşısında bulunan klasör ikonuna tıklayarak kayıt yeri belirleyebilirsiniz.

JPEG KALİTESİ:

Bitmap sıkıştırmasını kontrol etmek için JPEG Kalitesi kaydırıcısını ayarlayın veya bir değer girin. Düşük görüntü kalitesi daha küçük dosyalar oluşturur; yüksek görüntü kalitesi daha büyük dosyalar oluşturur. Boyut ve kalite arasındaki dengelemeyi belirlemek için farklı ayarlar denemek gerekir. 100, en yüksek kalite ve en az sıkıştırmayı sağlar.

Yüksek oranda sıkıştırılmış JPEG görüntülerinin daha düzgün görünmesini sağlamak için, JPEG Blok Kaldırmayı Etkinleştir seçeneğini belirleyin. Bu seçenek, JPEG sıkıştırmasından kaynaklanan tipik görünüm yapaylıklarını azaltır. Bu seçenek belirlendiğinde bazı JPEG görüntüleri az da olsa ayrıntı kaybına uğrayabilir.

SES AKIŞI ve SES OLAYI:

SWF dosyasındaki tüm akış sesleri veya olay sesleri için örnek hızını ve sıkıştırmayı ayarlamak üzere, Ses Akışı veya Ses Olayı 'nın yanındaki değerlerin üzerine tıklayın ve seçenekleri gerektiği gibi seçin.

Not: Ses akışı, ilk birkaç kare için yeterli veri yüklenir yüklenmez çalınır; Zaman Çizelgesi ile senkronizedir. Bir olay sesi, tamamen yüklenmeden çalmaz ve dışarıdan durduruluncaya dek çalmaya devam eder.

Özellik denetçisinin Ses bölümündeki ayrı ayrı sesler için ayarları geçersiz kılmak üzere Ses Ayarlarını Geçersiz Kıl'ı seçin. Bir SWF dosyasının daha küçük, lo-fi sürümünü oluşturmak için bu seçeneği seçin.

Not: Ses Ayarlarını Geçersiz Kıl seçeneği belirlenmemişse, Flash Professional belgedeki tüm ses akışlarını tarar (içe aktarılmış videodaki sesler dâhil) ve tüm ses akışlarını kendilerine özgü en yüksek ayarda yayınlar. Bir veya daha fazla akış sesinin yüksek bir dışı aktarma ayarı varsa, bu işlem dosya boyutunu artırabilir.

GELİŞMİŞ AYARLAR

< Gelişmiş > yazısının üzerine tıklamak suretiyle gelişmiş ayar seçeneklerini görünür veya görünmez duruma getirebilirsiniz.

Filmi Sıkıştır

Filmi Sıkıştır (Varsayılan) Dosya boyutunu ve indirme zamanını azaltmak için SWF dosyasını sıkıştırır. Bu işlem bir dosya metin veya Action Script yoğun olduğunda en faydalı seçenektir. Sıkıştırılmış bir dosya sadece Flash Player 6 veya sonrasında oynatılır.

GİZLİ KATMANLARI DAHİL ET:

Gizli Katmanları *Dâhil Et* (Varsayılan) Flash belgesindeki tüm gizli katmanları dışa aktarır. Gizli Katmanları Dışa Aktarma 'nın onayını kaldırma, film kliplerinin içinde yuvalanmış katmanlar dâhil, gizli olarak işaretlenmiş tüm katmanların oluşturulan *.swf dosyanızla dışa aktarılmasını engeller. Bu sizin katmanları görünmez yaparak Flash belgelerinin farklı sürümlerini kolayca test etmenizi sağlar.

BOYUT RAPORU OLUŞTUR:

Boyut Raporu Oluştur son Flash Professional içeriğindeki veri miktarını dosyalara göre listeleyen bir rapor üretir.

İZLEME EYLEMLERİNİ ATLA:

İzleme Eylemlerini Atla Flash Professional uygulamasının geçerli SWF dosyasındaki *ActionScript trace* ifadelerinin yok sayılmasına neden olur. Bu seçeneği onayladığınızda, trace ifadelerinden gelen bilgiler çıktı panelinde görünmez.

HATA AYIKLAMAYA İZİN VER:

Hata Ayıklamaya İzin Ver Hata Ayıklayıcıyı etkinleştirir ve bir Flash Professional SWF dosyasında uzaktan hata ayıklanmasına izin verir.

İÇE AKTARMAYA KARŞI KORU:

Diğer kişilerin bir SWF dosyasını içe aktarmasını ve bunu bir FLA dosyasına geri dönüştürmesini engeller. Flash Professional, SWF dosyanızla şifre koruması kullanmanıza izin verir. İçe Aktarmaya Karşı Koru'yu seçeneğini onaylarsanız, şifre metin alanı aktif olacaktır buraya bir şifre girmeniz gerekir. Eğer bir şifre eklerseniz, diğer kullanıcılara SWF dosyasını içe aktarmadan veya dosyada hata ayıklamadan önce şifre sorulacaktır. Doğru şifreyi girmeleri durumunda bu işlemlere ulaşabileceklerdir. Şifreyi kaldırmak için şifre metin alanını temizlenmesi yeterlidir.

AYRINTILI TELEMETREYİ ETKİNLEŞTİR:

Yerel Oynatma Güvenliği açılır menüsünden kullanılacak Flash Professional güvenlik modelini seçiniz. Yayınlanmış SWF dosyanıza yerel veya ağ güvenlik erişimi verip vermeyeceğinizi belirtin. Sadece Yerel Dosyalara Eriş, yayınlanmış SWF dosyasının ağ üzerindeki değil, yerel sistemdeki dosyalar ve kaynaklarla etkileşime girmesine izin verir. Sadece Ağa Eriş, yayınlanmış SWF dosyasının yerel sistemdeki değil, ağdaki dosyalar ve kaynaklarla etkileşime girmesine izin verir.

KOMUT DOSYASI ZAMAN SINIRI:

Komut dosyalarının SWF dosyalarında çalışacağı maksimum süreyi ayarlamak üzere, bir Komut Dosyası Süre Sınırı değeri girin. Flash Player, bu sınırı aşan komut dosyası çalıştırmalarını iptal eder.

DONANIM HIZLANDIRMA:

SWF dosyasının donanım ivmesini kullanmasını sağlamak için, Donanım İvmesi menüsünden şu seçeneklerden birini belirleyin:

- **Düzey 1:** Doğrudan modu, tarayıcının çizim yapmasına izin vermek yerine Flash Player'ın doğrudan ekran üzerinde çizim yapmasını sağlayarak oynatma performansını artırır.
- **Düzey 2:** GPU modunda, Flash Player, video oynatımını ve katmanlar içindeki grafiklerin birleştirilmesini gerçekleştirmeye yönelik grafik kartının hesaplama



Donanım ivmesi, Filmi Test Et modunda kullanılmaz. Ayrıca bu modu destekleyen tarayıcı sürümlerini de araştırınız.

gücünü kullanır. Bu da, kullanıcının grafik donanımına bağlı olarak başka bir performans avantajı düzeyi sağlar. İzleyici kitlenizin ileri düzey grafik kartlarına sahip olduğunu umduğunuzda bu seçeneği kullanın.

Birden çok SWF dosyaları içeren web sayfalarında en iyi performansı elde etmek için, SWF dosyalarından yalnızca birine yönelik donanım ivmesini etkinleştiriniz.



Özet

•Adobe Flash programında hazırlanan filmlerin, hazırlandığı programdan bağımsız olarak, player (oyuncu) programlarda izlenebilmesi için yayınlanması gerekir. Yayınlama işlemi neticesinde seçimlere bağlı olarak bir miktar da sıkıştırılır. Böylece sabit disklerde kapladığı alan da azalmış olur.

•Yayınlama sırasında tercihlere bağlı olarak SWF, HTML, GIF, JPEG ve PNG uzantılı dosyalar oluşturulabilir. Adobe Flash programının farklı versiyonlarında MOV, EXE, APP uzantılı dosyalar da oluşturulabilmektedir.

•Bilgisayarda kurulu olması zorunlu olan Adobe Flash Player'ın sürümü de önemlidir. Çünkü hazırlanan filmlerin bu programda izleneceği varsayılarak, filmin yayınlanması sırasında, bilgisayardaki Adobe Flash Player özelliklerine göre yayınlama yapılmaktadır.

•Gelişmiş ayarlar kısmında; yayınlanacak filmde sıkıştırma istenip istenmediği, gizli katmanların filme dahil edilip edilmeyeceği, yayınlama sonrasında oluşan dosyaların boyutlarına ait bir raporun istenip istenmediği, izleme ifadelerinin atlanıp atlanmayacağı, uzaktan hata ayıklamaya müsaade edilip edilmeyeceği, diğer kullanıcıların içe aktarım yapmak istemeleri durumunda şifre sorulup sorulmayacağı, ayrıntılı telemetreyi etkinleştirip etkinleştirmeyeceği, yayınlama sırasında yerel veya ağ üzerindeki dosyalarla etkileşime gireceği, donanım hızlandırma seçenekleri ve komut dosyalarının en fazla okunma süresi gibi tercihler bulunmaktadır.

DEĞERLENDİRME SORULARI

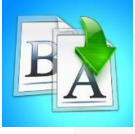
1. Adobe Flash programında yayınlama niçin yapılır?
 - a) Filme ses ilavesi yapmak için,
 - b) Filme Şekil ilavesi yapmak için,
 - c) Filme video ilavesi yapmak için,
 - d) İçerik aktarım yapmak için,
 - e) Filmi hazırlandığı programdan bağımsız izleyebilmek için.
2. Yayınlama sonrasında aşağıdakilerden hangi dosya oluşmaz?
 - a) *.mov
 - b) *.exe
 - c) *.dll
 - d) *.app
 - e) *.swf
3. Adobe Flash Player için hazırlanan dosyaların uzantısı aşağıdakilerden hangisidir?
 - a) EXE
 - b) FLA
 - c) XFL
 - d) SWF
 - e) MOV
4. Quick Time için hazırlanan dosyaların uzantısı aşağıdakilerden hangisidir?
 - a) EXE
 - b) FLA
 - c) XFL
 - d) MOV
 - e) APP
5. Yayınlama ayarları penceresinde bulunan *Filmi sıkıştır* seçeneği onaylandığında hangi uzantıya sahip dosya sıkıştırılır?
 - a) EXE
 - b) SWF
 - c) MOV
 - d) FLA
 - e) APP
6. Aşağıdaki seçeneklerden hangisi bilgisayarın donanım yapısı ile ilgilidir?
 - a) Filmi sıkıştır.
 - b) Boyut raporu oluştur.
 - c) Gizli katmanları dâhil et.
 - d) Hata ayıklamaya izin ver.
 - e) Donanım hızlandırma.

7. Yayınlama ayarları penceresinden hangi uzantılı dosyanın kalite ayarı yapılabilir?
- a) JPEG
 - b) SWF
 - c) XFL
 - d) EXE
 - e) MOV
8. Yayınlama ayarları penceresindeki ilgili alana şifre girilmesi, diğer kullanıcılarda hangi işlemleri engeller?
- a) Adobe Flash Prefessional CC programını açamazlar.
 - b) Adobe Flash Player programını açamazlar.
 - c) FLA uzantılı dosyayı açamazlar.
 - d) SWF uzantılı dosyayı açamazlar.
 - e) SWF uzantılı dosyayı açıp izleyebilirler, ancak içe aktarım yapamazlar.
9. Adobe Flash Player sürümü aşağıdakilerden hangisini doğrudan etkiler?
- a) Komut seti seçimini
 - b) Ses olaylarını
 - c) Hata ayıklamayı
 - d) Farklı formatlarda çıktı dosyası oluşmasını
 - e) Gizli katmanların filme dahil edilmesini
10. Adobe Flash Prefesional CC programında aşağıda verilen uzantılı dosyalardan hangisi yayınlanamaz?
- a) SWC
 - b) SWF
 - c) EXE
 - d) HTML
 - e) JPEG

YARARLANILAN KAYNAKLAR

- [1] Adobe Flash Pro CC, Yardımcı El kitabı. <http://helpx.adobe.com/tr/support>
[Erişim tarihi: 30 Ekim 2013].
- [2] SCHULZE Pati.(2004) Macromedia Fireworks 8 Kaynağından Eğitim,
Medyasoft Yayınları.

ACTIONSRIPT



İÇİNDEKİLER

- ActionScript
- Eylem Paneli
- Programlama Kuralları
- ActionScript ile Çalışmak
- Film Klipleriyle Çalışma
- Veri Türleri
- Değişkenler
- Sabitler
- Operatörler



HEDEFLER

- Bu üniteyi çalıştıktan sonra;
- Animasyolarda kod yazmak için Eylem panelinin bölümlerini öğrenebilecek,
- ActionScript kodlar için değişken tanımlayabilecek ve dilin kurallarını anlayabilecek,
- Kod yazarken uygun veri tiplerini seçerek işlemler yapabileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

ANİMASYON TASARIMI

Okt.
Mustafa AKBUĞA

ÜNİTE 10

Giriş

Günümüzde Web sitelerinde görmeye alıştığımız ses ya da hareket içeren görsel öğeler genellikle Adobe Flash ile tasarlanmaktadır. Çünkü bu programda diğer tasarım programlarda olmayan ActionScript programlama desteği vardır. ActionScript programlama diliyle, tasarım ortamı içerisinde araç ve menülerle yapılamayan kullanıcıdan aldığı yanıtlara göre hareket eden etkileşimli animasyonlar, internet siteleri ya da eğitim materyalleri kolayca yapılabilir.

Bu ünite de ActionScript programlama diline ait kavramlar, kodların yazıldığı Eylem paneli ve Oparatörler hakkında bilgiler verilecektir.

ACTIONSCRIPT

Adobe Flash yazılımının ilk versiyonlarında temel animasyon ve yönlendirme komutları ile ortaya çıkan bu dil, programın 5. sürümüyle birlikte tam olarak bir uygulama dili hâline gelmiş ve ActionScript adını almıştır.

Adobe Flash yazılımının yeni versiyonları çıktıkça ActionScript programlama dili de onunla birlikte yenilenmiş ve yeni eklenen bölümlere destek verebilecek hâle getirilmiştir. Günümüzde kullanılan en son sürüm ActionScript 3.0 versiyonudur. ActionScript dili ECMA script standardı (internette sunucu üzerinde çalışan programlar için geliştirilmiş dil standardı) üzerine inşaa edilmiştir.

ActionScript ile Değişken tanımlamaları yaparken veri türleri, hesaplamalar kısmında oparatörlerden, hareketlerin planlanmasında ise olaylar ve fonksiyonlardan faydalanılır.

ActionScript programlama diliyle zaman cetveli üzerinde yer alan film kareleri ve sahne üzerinde çizilmiş nesneler ayrıca sembollere çevrilerek klavye ve fare yardımı ile kontrol edilebilir. Adobe Flash yazılımı içerisinde tasarıma, ActionScript kod yazarken Eylemler paneli kullanılır.

Eylemler Paneli

Eylemler paneli, Adobe Flash programı içerisine Script yazmak için geliştirilmiş editördür. Bu eylem paneli sayesinde ActionScript kodlar nesneler ile ilişkilendirilir. Bu panel üzerinde kod yazarken kodla ilgili parametreler Eylem paneli tarafından programcıya gösterilerek kodlama işlemine yardımcı olur.

Eylem panelini açmak için:

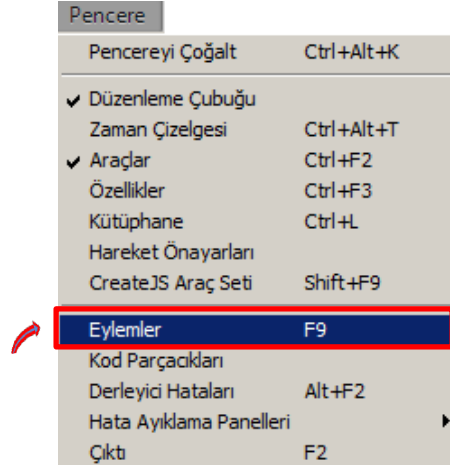
- Şekil 10.1’de gösterildiği gibi (Pencere > Eylemler) (F9) komutunu kullanabilirsiniz.



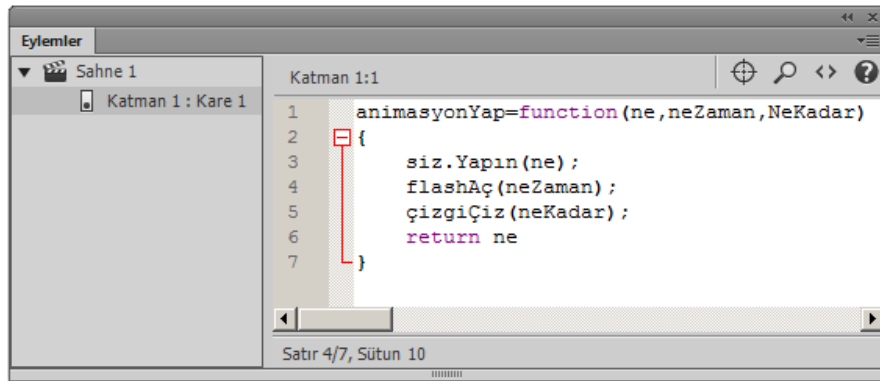
Adobe Flash tasarım ortamında ActionScript kodu yazarken, Eylemler paneli veya Komut Dosyası penceresi kullanılır.



Adobe Flash Pro CC versiyonuyla birlikte sadece ActionScript dilinin 3.0 versiyonunu kullanabiliriz. Önceki sürümlere verilen destek kaldırılmıştır.



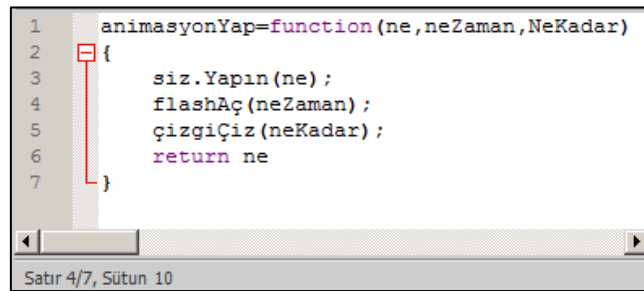
Şekil 10.1 Pencere menüsü



Şekil 10.2 Eylemler paneli

Şekil 10.2’de gösterilen Eylemler paneli 3 bölümden oluşmaktadır.

- **1- Script Ekleme** -> Düzenleme Alanı: Şekil 10.3’te gösterilen kodların yazıldığı ve daha önceden yazılmış kodların düzenlenebildiği bölümdür.

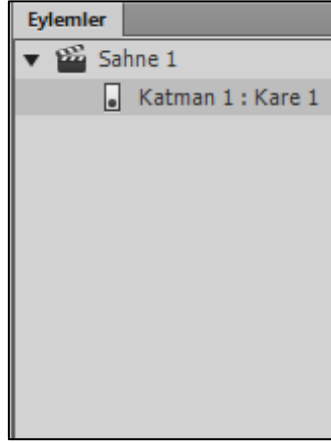


Şekil 10.3 Script ekleme- düzenleme alanı

- **2-Script gezinme alanı** -> Şekil 10.4’te gösterilen bölüm Karmaşık animasyon ve uygulamalarda kod yazılan yerler arasında hızlı geçiş yapmak için ActionScript 2.0 ile geliştirilmiş ve hâlen kullanılan özelliktir. Kod yazılan tüm nesneler bu bölümde listelenir.



Sahne üzerindeki her nesnenin Eylem paneli üzerinde bir bağlantı yolu vardır. Bu yolu belirlemenin en kolay seçeneği hedef yolu ekle simgesini kullanmaktır.



Şekil 10.4 Script gezinme alanı

- **3-Panel Araçları->** Eylemler panelinde kod yazarken kullanımı kolaylaştıran doğruluk denetimi yapan Şekil 10.5’te gösterilen araçlar bulunmaktadır.



Şekil 10.5 Panel araçları

Bu araçlar şunlardır;

() **Hedef Yolu Ekle (Sadece Eylemler paneli):** Komut dosyasındaki bir eylem için mutlak veya göreceli bir hedef yolu ayarlamanıza yardım eder.

() **Bul:** Komut Dosyanızın içindeki metni bulur ve değiştirir.

() **Kod Parçacıkları:** Örnek kod parçacıklarını görüntüleyen Kod Parçacıkları panelini açar.

() **Yardım:** Komut Dosyası bölmesinde seçili ActionScript ögesi için başvuru bilgisini görüntüler.

Programlama kuralları

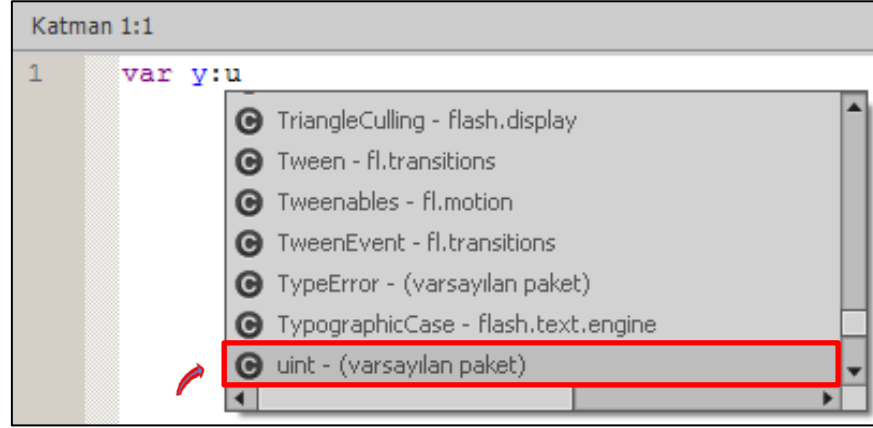
Kod ipuçları: Yazılan kodlar için ipucu kutusunu çalıştırır. Kod ile ilgili parametreler bu kutu içerisinde gösterilir.

Kod ipuçlarını göstermek için:

- Eylemler panelini açın.
var y, yazıp: konulunca Şekil 10.6’daki kod ipuçlarını içeren kutu görülecektir.
- Açılan pencereden uint veri türü seçilir.



Kod ipucu kutusunu açmak için (CTRL+Boşluk) tuşları, kapatmak için ise “Esc” tuşu kullanılabilir.



Şekil 10.6 Kod ipuçları

Nokta(.): Nesnelerin yol tanımlarını yapabilmek ya da sınıfların özellik ve metodlarına ulaşmak için kullanılan işarettir.

```
1 this.daire
```

Noktalı Virgül (;): Yazılan kodlarda ifadenin bittiğini göstermek amacıyla ifade sonuna konulur.

```
1 var universite:String="Atatürk Üniversitesi";
```

Satır sonundaki (;) ifadesi trace komutunun sonlandığını göstermektedir. Otomatik düzelt seçeneği çalıştırıldığında otomatik olarak ifadeyi yerleştirecektir. Noktalı virgül yazılmasa da program hata mesajı vermez, fakat okunuşu kolaylaştırmak ve düzeni sağlamak için yazılması gereklidir.

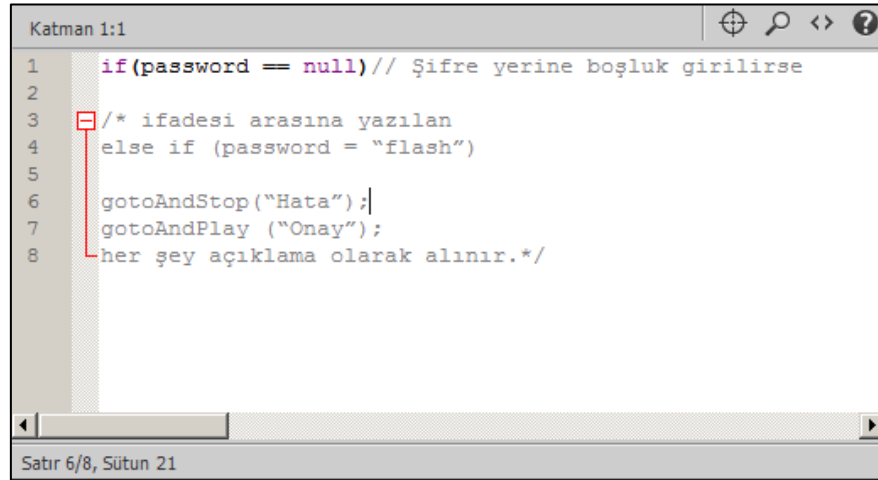
Açıklamalar: Kodlamanın yoğun ve uzun olduğu projelerde, kod parçalarının ne iş yaptığını hatırlamak ya da takım çalışması yaparken takım üyelerini kod blokları ve değişkenler hakkında bilgilendirmek için kullanılır. Kodların işleyişini etkilemez.

Açıklama ifadesi eklemek için:

- Eylemler paneli açılır ve (//) ifadesi yazılır. Buradan sonra yazılanlar tüm satırı açıklama satırı hâline getirir. Satır bittiğinde açıklamada bitmiş olur.
- /* */ ifadesi yazıldığında, arasına yazılan her şey açıklama olarak alınır (Şekil 10.7).



Eylem panelinde gösterilen satır numaraları program kodlarının başlangıç ve birtiş kısımlarında etkili değildir. Bir kod satırı ancak (;) ifadesiyle sonlandırılabilir.

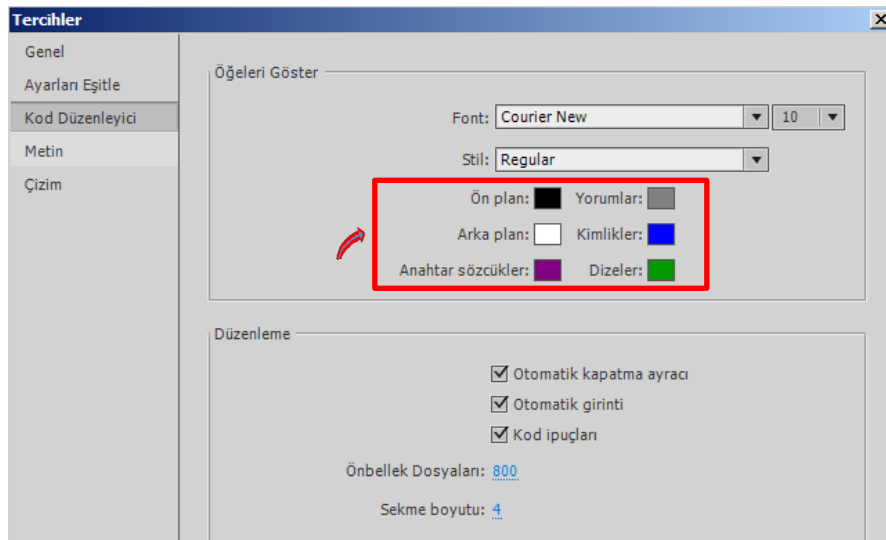


Şekil 10.7 Açıklama ifadeleri

Açıklama satırları, program yazarken bazı kodları iptal etmek ve yeniden yazmaktan kurtulmak için sıkça kullanılır. Açıklama satırı olarak eklenen satır, panel renklendirmesi değiştirilmedikçe gri ve silik olarak yazılacaktır (Şekil 10.7).

Panel renklerini değiştirmek için:

- Düzen > Tercihler ya da “Eylemler” panelinden köşedeki Tercihler simgesini tıklayın.
- Şekil 10.8’de gösterilen Tercihler iletişim penceresinden, ActionScript kategorisi seçildiğinde sağ taraftaki kod renklendirme bölümünden istediğinize uygun ayarı yapın.



Şekil 10.8. Kodlama renkleri

{ } parantezleri : {küme parantezi } arasında yazılan her kod birlikte çalışmak için bloklanır.

```

Eylemler:1
1  var aof:int=4;
2  while (aof>=0)
3  {
4      trace("Animasyon tasarımı");
5      a--;
6  }

```

Yukarıdaki döngüde küme parantezleri arasındaki kod bloğu her döngü basamağı için birlikte çalışacaktır.



Nesne tabanlı programlama dillerinin büyük bir bölümünde kod ifadeleri Camel standardı kullanılarak hazırlanır.

İsmlendirme standartları

Kod yazarken ya da animasyon programında örnek ismi verirken kullanılacak birçok isim vardır. Bu isimlerin belli bir standarta göre yazılması daha sonra ismin kullanılacağı yerlerde acaba “Nasıl yazmıştım?” derdinden kurtaracaktır. Belli standartlar olduğu gibi kullanıcı kendisi de bir standart oluşturabilir.

Camel standardı: Yazılan ismin ilk harfinin küçük, birden fazla kelimeden oluşan bir isimse diğer kelimelerin ilk harflerinin büyük olması ile yazılır. ActionScript 3.0 bu standardı kullanmaktadır. Bu standarda göre yazmak zorunlu değildir. Fakat kod yapısına alışmak için bu şekilde bir yazım şeklini benimsemek faydalıdır. “animasyonTasarımı” yazımı bu standart için bir örnektir.

Pascal standardı: Yazılan değişken isminin kelime ya da kelimelerin ilk harflerinin büyük olması ile yazılır. “AnimasyonTasarımı” yazımı bu standart için bir örnektir.

Underscore standardı: Yazılan kelimelerin arasına “_” işareti koyarak yazım yapılır. “animasyon_tasarımı” yazımı bu standart için bir örnektir.

ACTIONSCRIPT İLE ÇALIŞMAK

Film Klipleri ile Çalışma

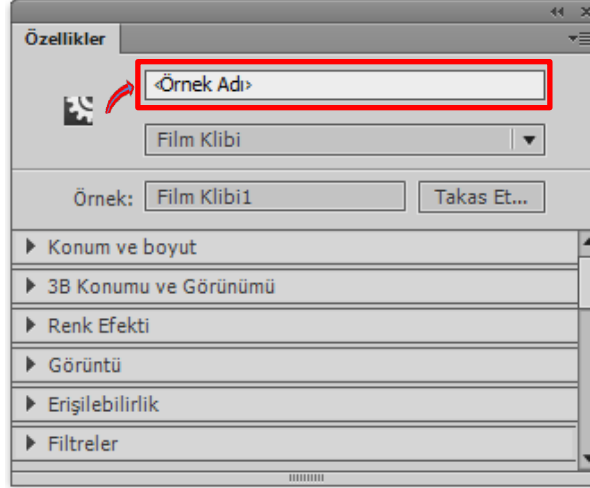
Film klipleri, butonların veya grafiklerin yaptığı her işi yapabilirler. Kendi başlarına bir animasyon çalışma dosyası gibidirler; yani her film klibinin kendi zaman çizgisi bulunmaktadır. Oluşturulan film klibi animasyon yazılımının kütüphanesine eklenir. Buradan istenildiği kadar çoğaltılıp sahneye eklenebilir. Film kliplerini ActionScript içerisinde kullanmak için sahnedeki yerini ve film klibinin ismini bilmek yeterlidir. Sahne içerisinde yer alan sembollerin bazıları kütüphane panelindeki bir sembolden kopya olarak üretilmiş olabilir. Bu durumda, Kod yazarken oluşabilecek karmaşayı önlemek için film kliplerinin kütüphane isimlerinin haricinde sahnedeki her nesneye bir örnek ismi (instance name) atamak gereklidir. Özellikler panelinden örnek ismi aşağıda gösterildiği şekilde oluşturulabilir.

Örnek ismi oluşturmak için:

- Örnek ismi ekleyeceğimiz nesnemizi sahneden seçin.
- Şekil 10.9’da gösterilen Özellikler panelindeki örnek adı bölümüne değişken isimlendirme kurallarına uygun olarak bir örnek ismi verin.

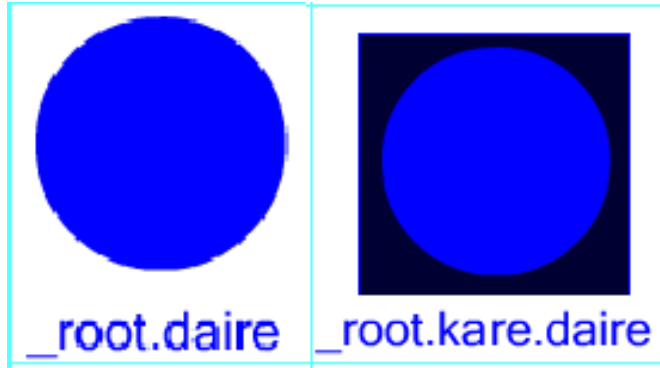


ActionScript ile kod yazarken Örnek adı bölümünde yazılan isimler, nesnelerin kod içindeki tanımlanmasında kullanılır.



Şekil 10.9 Örnek adı (instance name).

Film klibinin ismi dışında, sahnede bulunduğu yer de önemlidir. Örneğin, Şekil 10.10 ’da gösterilen grafikte yer adreslerine bakılırsa, ana sahne üzerinde bulunan bir film klibindeki daire isimli nesnesinin adresi “*_root.daire*” şeklinde ve başka bir sembolünün içinde yer alan daire isimli film klibinin adresi ise “*_root.kare.daire*” şeklinde tanımlanacaktır.



Şekil 10.10 Adres tanımlama

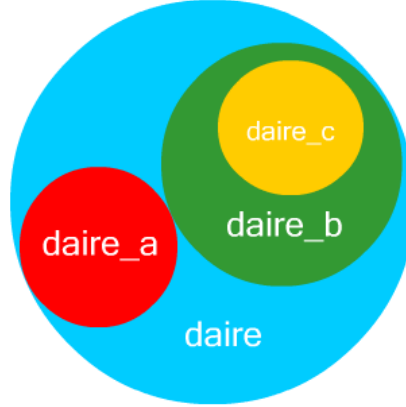
Adresleme kurallarını anlamak için aşağıdaki örneği uygulayınız.

- Yeni bir çalışma sahnesi açın.
- Sahneye daire çizip film klibi haline getirin örnek adını *daire* yapın.
- Şekil 10.11’de gösterilen daire klibine çift tıklayarak içerisine girin ve *daire_a*, *daire_b* örnek isimlerine sahip film kliplerini oluşturun.



Animasyon yazılımında sıkça kullanılan film klipleri, ActionScript içinde olmazsa olmaz bileşenlerdendir. Kodlar yazılırken film klipleri içerisinde yazılır

- *daire_b* film klipinin içerisine girip *daire_c* örnek ismine sahip yeni bir daire oluşturun.
- Sonuç olarak aşağıdaki şekli elde etmelisiniz.



Şekil 10.11 Nesne konumları

- Ana sahneye dönüp eylemler panelini açın ve aşağıdaki adımlar hâlinde verilen kodları sırayla deneyin.

1.Adım: “*daire*” isimli film klipini 45 derece döndürmek için aşağıda verilen komutları Eylem paneline yazınız.

```
Katman 1:1
1 this.daire.rotation=45;
```

CTRL+enter tuşlarına basarak tasarımı test ediniz. Test sonrası nesnelerin durumu Şekil 10.12 (a)’da görüldüğü gibi olacaktır.

2.Adım: “*daire_a*” isimli film klipini 45 derece döndürmek için aşağıda verilen komutları Eylem paneline yazınız.

```
Katman 1:1
1 this.daire.daire_a.rotation=45;
```

Yukarıdaki koda dikkat edilirse “*daire*” klipinin içinde yer alan “*daire_a*” sembolünün adresi yazılarak döndürme komutu kullanılmıştır. Tasarım, CTRL+enter tuşlarına basılarak test edildiğinde nesnelerden yalnızca “*daire_a*” nın 45 derece döndürüldüğü görülecektir (Şekil 10.12(b)).

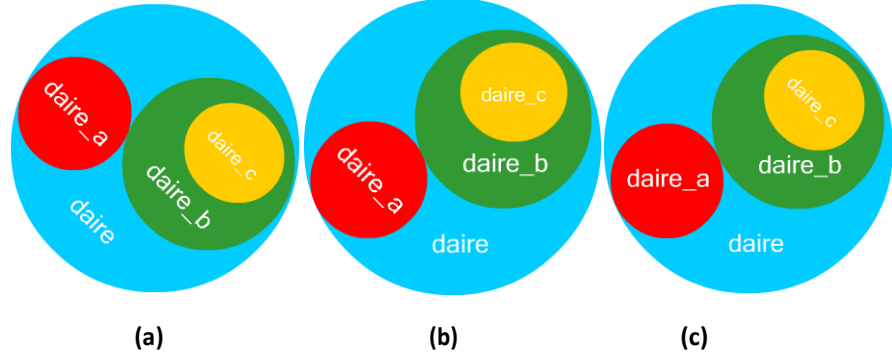
3.Adım: “*daire*” klipi içinde yer alan diğer “*daire_b*” nesnesinin dönderilmesi ise 2.Adımda verilen kodla neredeyse aynıdır. Sadece nesne ismi “*daire_a*” yazan yere “*daire_b*” yazmak yeterli olacaktır.

4.Adım: “*daire_c*” klipini 45 derece döndürmek için için aşağıda verilen komutları Eylem paneline yazınız.

Katman 1:1

```
1 this.daire.daire_b.daire_c.rotation=45;
```

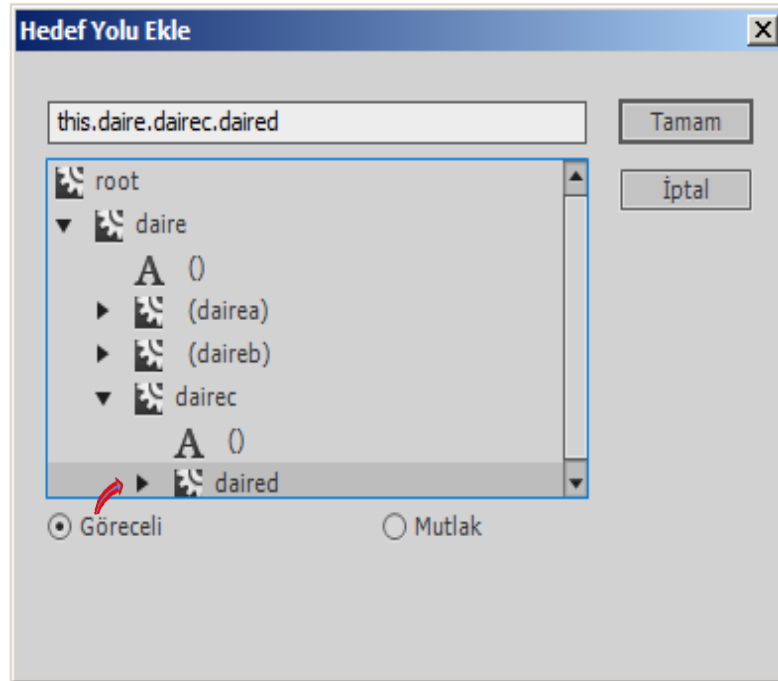
Yukarıda kodla verilen “this.daire.daire_b.daire_c” adresi ile *daire_c*’nin sahne üzerindeki konumu (daire nesnesi içindeki daire_b’nin içinde yer aldığı) işaret edilmektedir. Test sonrası nesnelerin durumu Şekil 10.12(c)’de görülmektedir.



Şekil 10.12 Nesnelerin sahne üzerindeki konumlarının değiştirilmesi

Nesnelerin konumlarını Eylem paneline yazmak size karmaşık geliyorsa, Hedef Yolu Ekle penceresi kullanılarak da nesnelerin adresini yazabilirsiniz. Bu işlem için;

Hedef yol ekle  simgesini kullanarak sahnedeki konuma göre nesneleri seçip, yolu Şekil 10.13’de gösterilen panel üzerinden kolay bir şekilde kod içerisine ekleyebilirsiniz.



Şekil 10.13 Hedef yol ekle.

Veri Türleri



Animasyon yazılımında en fazla uint ve string veri türleri kullanılır.

ActionScript 3.0 içerisinde kullanılan veri türleri basit düzeyli ve ileri düzeyli olmak üzere ikiye ayrılır. Uygulama içerisinde hangi tür verilerle çalışılacaksa ona uygun veri türünün seçilmesi gerekmektedir. Basit düzey veri türleri kodlamada sıkça kullanılmaktadır. Bunlar Pozitif tamsayılar, Tamsayılar, Tümsayılar, Alfanümerik, Mantıksal, Nesne veri türü olmak üzere 6 guruba ayrılır. Bu veri türleri aşağıda örnekler üzerinde açıklanmıştır. İleri düzey veri türleri ise XML, XMLList, Array, data, error, function, RegExp olmak üzere 7 kısımdır. Bunlardan bazıları ilerleyen ünitelerde açıklanacaktır. Karışık uygulamalarda belleği tasarruflu kullanmak ve uygulamanın performanslı çalışmasını sağlamak için veri türlerinden uygun olanı seçmek önemlidir.

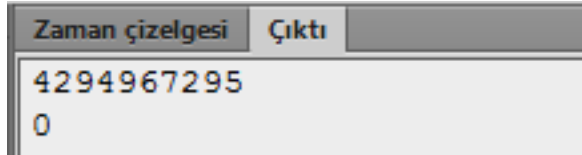
Pozitif tam sayılar (Uint): 0'dan büyük tam sayı değeri alabilen veri türüdür.

Veri türünün alabileceği maksimum ve minimum değerleri görmek için:

- Yeni bir çalışma sahnesi açınız.
- Pencere>Eylemler komutu yardımıyla, Eylem paneli açılarak aşağıdaki kodları ekleyiniz.

```
1 trace(uint.MAX_VALUE);
2 trace(uint.MIN_VALUE);
```

- CTRL+enter tuşlarına basıp çıktı ekranında yukarıda verilen değerleri test edebilirsiniz.



Tam sayılar (int): Pozitif veya negatif tam sayı değeri alabilen veri türüdür. Uintin aksine negatif değerler de alabilir.

Veri türünün alabileceği maksimum ve minimum değerleri görmek için:

- Yeni bir çalışma sahnesi açınız.
- Eylem paneli açılarak aşağıdaki kodlar ekleyiniz.

```
1 trace(int.MAX_VALUE);
2 trace(int.MIN_VALUE);
```

- CTRL+enter tuşlarına basıp çıktı ekranında yukarıda verilen değerleri test edebilirsiniz.



Mantıksal karşılaştırmaların sonuçları Boolean veri türünde saklanır.

Zaman çizelgesi	Çıktı
	2147483647 -2147483648

Tüm sayılar (Number): Gerçel (Reel) sayı değeri alabilen veri türüdür. Veri türünün alabileceği Maksimum ve minimum değerleri görmek için:

- Yeni bir çalışma sahnesi açınız.
- Eylem panelini açarak aşağıdaki kodları ekleyiniz.

```
1 trace(Number.MAX_VALUE);
2 trace(Number.MIN_VALUE);
```

- CTRL+enter tuşlarına basıp çıktı ekranında yukarıda verilen değerleri test edebilirsiniz.

Zaman çizelgesi	Çıktı
	1.79769313486231e+308 4.9406564584124654e-324

Yukarıdaki gösterilen veri aralıkları incelendiğinde; Number veri türü, çok büyük bir aralıkta olduğu ve her iki sayısal veri türünü (Unit ve İnt) kapsadığı görülür. Öyleyse neden diğer veri türlerine gerek var sorusu akla gelebilir. Bunun sebebi SWF haline getirilen animasyon dosyasının daha az yer kaplaması ve hızlı çalışması için değişken türünü, aktarılacak değere uygun seçmek daha akıllıca olacaktır. Oluşturulacak 300-500 kayıtlık bir adres defterinde veri türü önemli olmazken büyük veri grupları gerektiren, örneğin banka uygulamaları vb. uygulamalarda veri türü hayati öneme sahip olacaktır.

Alfanümerik veri türü (String): İçerisinde harf, kelime, sayı, özel karakterler kısacası klavyeden basılabilecek her tuşu barındırabilen veri türüdür. Yazılan her şeyi içerisine veri olarak alabilir. Sayısal değerleri veri olarak aldığı bu değerler ile matematiksel işlemler yapılamaz.

Mantıksal veri türü (Boolean): Doğru (true) ve yanlış (false) değerler alabilen veri türüdür. Genellikle karşılaştırmalarda kullanılır.

Yukarıda anlatılan veri türlerinin yanında Object ve Void veri türleride vardır ve bunlar sonraki ünitelerde anlatılacaktır.

Değişkenler

Değişkenler, verilerin program tarafından kullanımı için geçici olarak depolandığı bellek konumlarıdır. Değişkenler bir ada, türe ve değere sahiptirler.

Değişken tanımlama yapılmadan önce program içerisinde kullanılamaz,

Değişken tanımlama kuralları aşağıdaki şekilde ifade edilebilir.



Void veri tipi, Sadece fonksiyonlarda kullanılır.

- ActionScript 3.0 büyük küçük harf duyarlıdır. Yani *AOF, Aof, aOF, aoF* vb. yazılış durularını birbirinden farklı değişkenler olarak algılayacaktır.
- Sadece İngilizce karakterler kullanılır. *ı, İ, ş, Ş, ç, Ç, ü, Ü, ö, Ö, ğ, Ğ* karakterleri gibi Türkçe karakterler kullanılmaz. En çok karıştırılan küçük *ı* ve büyük *İ* karakterine dikkat ediniz. Özel karakterler (***, *?*, */*, vb.) değişken tanımlamakta kullanılmaz.
- Değişken isimleri rakamla başlayamaz fakat değişken içerisinde rakam kullanılabilir. “*1animasyon*” yanlış bir değişken ismi olacaktır. “*animasyon1*”, “*a1animasyon*”, “*ani1masyon*” gibi yazılışlar doğrudur.
- Değişken tanımlamada yapılan en büyük hatalardan biri değişken ismi olarak kısaltma kullanmaktır. Uzun kod satırlarından oluşan uygulamalarda daha sonra bu kısaltmaların ne olduğunu hatırlamak çok zor olabilir. Bu yüzden değişkenlere verilecek isimlerin mantıklı, daha sonra okunduğunda anlaşılacak isimler olması çok faydalı olacaktır. Örneğin: “*anmTas*” yerine “*animasyonTasari*” değişkeni ismini kullanmak, sonrası için daha kullanışlı olacaktır.
- Kelimeler arasında boşluk bırakılmamalıdır.

Değişken Tanımlama

ActionScript dilinde değişken tanımlama işlemi aşağıdaki gibi yapılmaktadır.

```
var Değişken_Adi: Veri Türü;
```

Yukarıdaki tanımlamada

Var, Değişken tanımlamak için *variable* kelimesinden kısaltılarak oluşturulan anahtar kelimedir.

Değişken_Adi, Değişkeni kod içerisinde kullanmak amacıyla verilmiş isim alanıdır.

Veri türü, Uygulama içerisinde hangi tür verilerle çalışılacaksa (int, uint, Number, string vb.) ona uygun veri türü bu alana yazılır.

Değişkene Değer Atama

ActionScript dilinde bir değişkene değer atama işlemi aşağıdaki gibi yapılmaktadır.

```
Değişken_Adi = Atanacak Değer;
```

Yukarıdaki tanımlamada

Atanacak Değer: Uygulama içinde değişkene atanacak değerler bu alana yazılır.

Yukarıda görüldüğü üzere Adobe Flash'ta bir değişkene değer atama işlemi (=)

işaretiyle yapılır. Burada gösterilen (=) işaretinin matematikteki gibi eşittir manası yoktur. Bu işaretin anlamı işaretin sağındaki değeri solundaki değişkene aktar şeklindedir.

Değişken Tanımlama ve Değer Atama

Şayet bir değişken tanımlanıp bu değişkene değer atanacak ise yukarıda anlatılan işlemler sırasıyla uygulanmalıdır. Bu uygulama iki yöntemle yapılabilir

1. Yöntem:

Bu yöntem kullanıldığında, önce değişken tanımlanır sonrasında ise değişkene değer ataması yapılır.

```
var Değişken_Adı: Veri Türü;
Değişken_Adı = Atanacak Değer;
```

2.Yöntem:

Bu yöntemde ise değişken tanımlı ve değer ataması aynı satırda yapılır

```
var Değişken_Adı: Veri Türü = Atanacak Değer;
```

Yukarıda gösterilen her iki yöntemi anlamak için aşağıdaki örnekleri inceleyin.

Örnek-1: Bir değişken tanımlayınız bu değişkeninin ismi *aSayısı*, türü *integer* olsun ve bu değişkene 10 değerini atayınız.

Bu işlem iki şekilde yapılabilir.

1.Yol: Aşağıdaki kod yardımıyla bu işlem gerçekleştirilebilir.

```
1 var aSayısı: int;
2 aSayısı = 10;
```

Yukarıdaki kodda birinci satırda ismi aSayısı olan değişken integer olarak oluşturulmuştur. İkinci satırda ise ismi Asayısı olan değişkene 10 değeri atanmıştır.

2.Yol: Yukarıda iki satırda gerçekleştirilen kodlama aşağıdaki işlemle tek satırda gerçekleştirilebilir.

```
1 var aSayısı: int = 10;
```

Örnek-2: Bir değişken tanımlayınız bu değişkeninin ismi *ornekSayi*, türü *Pozitif tam sayılar (Uint)* olsun ve bu değişkene 1 değerini atayınız.

```
1 var ornekSayi: uint = 1;
```

Örnek-3: Bir değişken tanımlayınız bu değişkeninin ismi *sayi*, türü *Tüm sayılar (Number)* olsun ve bu değişkene *123.456789* değerini atayınız.

ActionScript dilinde bir değişkene herhangi bir değer atamadan onu kullanmak yasaktır.

```
1 var sayi:Number=123.456789;
```

Örnek-4: Bir değişken tanımlayınız bu değişkeninin ismi *ornekAd* türü *AlfaNumerik (String)* olsun ve bu değişkene “*Adobe Flash Professional CC*” değerini atayınız.

```
1 var ornekAd:String="Adobe Flash Professional CC ";
```

String değerler “ ” işaretleri arasında yazıldığına dikkat ediniz.

Örnek-5: Bir değişken tanımlayınız bu değişkeninin ismi *medeniDurum*, türü *Mantıksal (Boolean)* olsun ve bu değişkene *true* değerini atayınız.

```
1 var medeniDurum:Boolean=true;
```

Sabitler

Program içerisinde değeri hiç değişmeyen yapılara sabit denir. Örneğin, pi sayısı matematiksel bir sabittir. Değeri asla değişmez. Ya da sizin uygulamanız için program boyunca sabit kalmasını isteyeceğiniz bir katsayı değeri de sabit olarak tanımlanabilir. Sabit tanımlamanın değişken tanımlamadan tek farkı, en başta yazılan var yerine “const” yazılmasıdır.

```
1 const pi: Number=3.14;
```

Bu ifadenin program içerisinde değeri değiştirilmek istenirse, animasyon yazılımında derleyici hatası oluşacaktır.

Trace Cümlecikleri

Kodlama yaparken değişkenlerin sonucunu ya da aldığı değerleri görmek için trace komutu kullanılır. Kullanımı oldukça basittir. Tek değişken yazdırılabileceği gibi birden fazla değişken ve string ifade de yazdırabilir. Örneğin;

```
1 var dersinAdı:String="Animasyon Tasarımı";
2 trace ("Dersin Adı:",dersinAdı);
```

Yukarıdaki yazım şekline dikkat edilirse trace içindeki her ifade birbirinden virgül işareti ile ayrılmış ve string ifadeler çift tırnak içerisinde yazılmıştır.

CTRL+enter tuşlarına birlikte basılarak çalıştırıldığında çıktı ekranında aşağıdaki sonuç görülür.

Zaman çizelgesi	Çıktı	Derleyici Hataları
	Dersin Adı: Animasyon Tasarımı	

Operatörler

Toplama operatörü (+): Bu operatör sayısal ifadeleri matematiksel olarak toplar. String ifadeleri yan yana yazdırır.

Aşağıdaki örnekte sayı türündeki iki değer matematiksel olarak toplanmıştır.

İstenilen anda istenilen değişkenlerin anlık değerleri trace komutu kullanarak çıktı ekranına yazdırılabilir.

```

1 var degisken_x:uint=1;
2 var degisken_y:uint=2;
3 trace(degisken_x+degisken_y);

```

CTRL+enter tuşlarına birlikte basılarak çıktı ekranında 3 sayısı görülür.

Zaman çizelgesi	Çıktı
	3

Aşağıdaki örnekte ise String türündeki iki ifade String olarak toplanmıştır. Bu yüzden çıktı ekranında değişken ifadeler yan yana yazılmıştır.

```

1 var degisken_x:String="1";
2 var degisken_y:String="2";
3 trace(degisken_x+degisken_y);

```

CTRL+enter tuşlarına birlikte basılarak çıktı ekranında 12 olarak görülür.

Zaman çizelgesi	Çıktı
	12

Fark operatörü (-): Sadece matematiksel ifadelerde kullanılır. Çıkarma işlemi yapar.

```

1 var degisken_x:uint=2;
2 var degisken_y:uint=1;
3 trace(degisken_x-degisken_y);

```

CTRL+enter tuşlarına birlikte basılarak çıktı ekranında 1 sayısı görülür.

Zaman çizelgesi	Çıktı
	1

Çarpma operatörü (*): Matematiksel çarpma işlemi yapılır.

```

1 var degisken_x:uint=3;
2 var degisken_y:uint=4;
3 trace(degisken_x*degisken_y);

```

CTRL+enter tuşlarına birlikte basılarak çıktı ekranında 12 sayısı görülür.

Zaman çizelgesi	Çıktı
	12

Bölme operatörü (/): Matematiksel bölme işlemi yapılır.

Değişkenlere veri tipleri atanırken bu değişkenlerin kullanıldığı işlem yapan operatörlerinde hesaba katılması gerekir. İki tam sayının bölünmesi sonrasında tam sayı olmayan sayılar ortaya çıkabilmektedir.

```

1  var degisken_x:uint=4;
2  var degisken_y:uint=2;
3  trace(degisken_x/degisken_y);

```

CTRL+enter tuşlarına birlikte basılarak çıktı ekranında 2 sayısı görülür.

Zaman çizelgesi	Çıktı
	2

Mod operatörü (%): Bölme işlemindeki kalanı bulmak için kullanılır.

```

1  var degisken_x:uint=9;
2  var degisken_y:uint=4;
3  trace(degisken_x%degisken_y);

```

CTRL+enter tuşlarına birlikte basılarak çıktı ekranında 1 sayısı görülür.

Zaman çizelgesi	Çıktı
	1



Döngü ve şartlı ifadelerde karşılaştırma operatörleri sıklıkla kullanılır.

Karşılaştırma Operatörleri (>, <, >=, <=, ==, !=): İki ifadenin aritmetik karşılaştırmasını yapar sonuç olarak boolean ifade elde edilir.

```

1  var degisken_x:uint=9;
2  var degisken_y:uint=4;
3  trace(degisken_x>degisken_y);

```

CTRL+enter tuşlarına birlikte basılarak çıktı ekranında “true” değeri görülür.

Zaman çizelgesi	Çıktı
	true

Atama operatörü (=): Sağ taraftaki değeri sol taraftaki değişkene atama işlemi yapar.

```

1  var degisken_x:uint=9;
2  degisken_x=999;
3  trace(degisken_x);

```

CTRL+enter tuşlarına birlikte basılarak çıktı ekranında 999 değeri görülür.

Zaman çizelgesi	Çıktı
	999

Arttırma operatörü (++): Sayının sonuna ya da başına eklenmesi durumunda sayının değerini 1 arttırır.

```
1 var degisken_x:uint=9;
2 degisken_x++;
3 trace(degisken_x);
```

Veya

```
1 var degisken_x:uint=9;
2 ++degisken_x;
3 trace(degisken_x);
```

CTRL+enter tuşlarına birlikte basılarak çıktı ekranında 10 değeri görülür.

Zaman çizelgesi	Çıktı
	10



Artırma ve azaltma operatörlerinin ön ek ya da son ek olması ActionScript dilinde değişkene aktarılabilecek sonucu etkilemez.

Azaltma operatörü (--): Sayının sonuna ya da başına eklenmesi durumunda sayının değerini 1 azaltır.

```
1 var degisken_x:uint=9;
2 --degisken_x;
3 trace(degisken_x);
```

Veya

```
1 var degisken_x:uint=9;
2 degisken_x--;
3 trace(degisken_x);
```

CTRL+enter tuşlarına birlikte basılarak çıktı ekranında 8 değeri görülür.

Zaman çizelgesi	Çıktı	Derleyici Hataları
	8	

Ekle ve ata operatörü(+=): toplam ve atama işlemi yapacaktır. Soldaki değişkenin değerini sağdaki değer kadar arttırıp değişkene yeni değeri aktarır.

“toplam+=i;” ifadesi, “toplam=toplam+i” ifadesinin kısa olarak yazılmasını sağlar.

```
1 var degisken_x:uint=9;
2 degisken_x+=11;
3 trace(degisken_x);
```

CTRL+enter tuşlarına birlikte basılarak çıktı ekranında 20 değeri görülür.



Ekle ve ata operatörünün kullanım şekli diğer operatörler için de geçerlidir (Çıkart ve ata, Çarp ve ata gibi).

Zaman çizelgesi	Çıktı	Derleyici Hataları
20		

Mantıksal VE operatörü (&&): Koşullardan hepsinin gerçekleşip gerçekleşmediğini kontrol eder. Yani şartlardan birisi gerçekleşmezse yanlış (false) değerini hepsi gerçekleşirse doğru (true) değerini döndürecektir

Örnek-1:

```
1 var a:int=9;
2 var b:int=3;
3 var c:Boolean=(a<10 && b>2) ;
4 trace(c) ;
```

Yukarıdaki örnekte a'nın 10'dan küçük ve b'nin 2'den büyük olması sebebiyle her iki ifadede de değerlendiren VE operatörü, c değişkenine "true" değerini atayacaktır. CTRL+enter tuşlarına birlikte basıldığında çıktı ekranında aşağıda gösterildiği şekilde görülür.

Çıktı
true

Örnek-2:

```
1 var a:int=9;
2 var b:int=3;
3 var c:Boolean=(a>10 && b>2) ;
4 trace(c) ;
```

Yukarıdaki örnekte b'nin 2 'den büyük olmasına rağmen a'nın 10 'dan büyük olmaması sebebiyle VE operatörünün değerlendirmesi sonucu, c değişkenine false değerini atayacaktır. Bu durum CTRL+enter tuşlarına birlikte basıldığında çıktı ekranında aşağıda gösterilmektedir.

Çıktı
false

Mantıksal VEYA operatörü (||): Koşullardan herhangi birisinin gerçekleşip gerçekleşmediğini araştırır. Şartlardan birisinin gerçekleşmesi doğru (true) değerini hiçbirisi gerçekleşmemiş ise yanlış (false) değeri geri döndürecektir.

Örnek-1:

```
1 var a:int=10;
2 var b:int=3;
3 var c:Boolean=(a<10 || b>2);
4 trace(c);
```

Yukarıdaki örnekte a'nın 10'dan küçük olmamasına rağmen b'nin 2'den büyük olması sebebiyle VEYA operatörü c değişkenine true değer atayacaktır. Bu durum CTRL+enter tuşlarına birlikte basıldığında çıktı ekranında aşağıda gösterildiği şekilde gösterilir.

Çıktı
true

Örnek-2:

```
1 var a:int=10;
2 var b:int=3;
3 var c:Boolean=(a>12 || b<2);
4 trace(c);
```

Yukarıdaki örnekte a'nın 12'den büyük olmaması false ve b'nin 2'den küçük olması sebebiyle false değerlerini VEYA operatörü değerlendirerek c değişkenine false değerini atayacaktır. Bu durum CTRL+enter tuşlarına birlikte basıldığında çıktı ekranında aşağıda gösterildiği şekilde gösterilir.

Çıktı
false



Özet

- Adobe Flash programının ilk versiyonlarında temel animasyon ve yönlendirme komutları olmak üzere 12 temel komut olarak ortaya çıkan bu dil, beşinci versiyonunda tam olarak bir uygulama diline dönüşmüş ve ActionScript ismini almıştır.
- Animasyon yazılımı geliştikçe ActionScript dili de gelişmiş, en son olarak ActionScript 3.0 versiyonu piyasaya çıkmıştır. Bu versiyonun, programlama özellikleriyle animasyon yazılımı üzerinde her türlü kontrolü sağlanmış, internet siteleri için uygulamalar, oyunlar ve buna benzer birçok uygulama için kullanıma hazır hâle getirilmiştir.
- ActionScript ile zaman çizgisini kontrol edebilir, yaptığınız animasyonlara etkileşim veya çeşitli şartlar ekleyebilir ve animasyonun istediğiniz şekilde oynamasını sağlayabilirsiniz. Sahneye düğmeler, yazı kutuları vb. form bileşenleri ekleyip kullanıcı ile etkileşime girilebilir.
- ActionScript kodlarıyla kod yazabilmek için Eylemler paneli kullanılır. Eylemler paneli 3 bölümden oluşmaktadır. 1 bölüm:Script Ekleme ve Düzenleme Alanı: Kodların yazıldığı ve daha önceden yazılmış kodların düzenlenebildiği bir bölümdür. 2 bölüm Script gezinme alanıdır. Bu bölüm Karmaşık animasyon ve uygulamalarda kod yazılan yerler arasında hızlı geçiş yapmak için kullanılır. 3 bölüm Panel Araçları bölümüdür. Eylemler panelinde kod yazarken kullanımı kolaylaştıran doğruluk denetimi yapan araçlar bulunmaktadır.
- ActionScript kodlar içinde; Sınıfları ayırırken Nokta (.), Satırları ayırırken Noktalı Virgül (;), Kod bloklarını ayırırken { } kullanılır.
- ActionScript dilinde kodları yazarken genelde Camel standardı kullanılır.
- Adobe Flash programında en fazla kullanılan film klipleri, ActionScript içinde olmazsa olmaz bileşenlerdir. Film kliplerini ActionScript içerisinde kullanmak için sahnedeki yerini ve film klibinin ismini bilmek yeterlidir.
- ActionScript 3.0 içerisinde kullanılan veri türleri basit düzeyli ve ileri düzeyli olmak üzere ikiye ayrılır. Uygulama içerisinde hangi tür verilerle çalışılacaksa ona uygun veri türünün seçilmesi gerekmektedir.
- ActionScript kodlamada kodlar ile işlem yapmak için değişken ve sabitleri kullanmak gereklidir. İşlemler Toplama operatörü, Fark operatörü, Çarpma operatörü, Bölme operatörü, Mod operatörü, Karşılaştırma operatörleri, Atama operatörü, Arttırma operatörü, Azaltma operatörü, Ekle ve ata operatörü, Mantıksal VE operatörü, Mantıksal VEYA operatörleri yardımıyla yapılır. Yapılan işlemlerin sonuçları görmek için Trace komutu kullanılır.

DEĞERLENDİRME SORULARI

1. Aşağıdakilerden hangisiyle Adobe Flash için program kodu yazılabilir?
 - a) Actionscript
 - b) VisualScript
 - c) JavaScript
 - d) VisualC
 - e) VirtualBox
2. Aşağıdaki seçeneklerden hangisi Flash için yazılan bir kodu sonlandırır?
 - a) ;
 - b) .
 - c) ()
 - d) :
 - e) { }
3. Aşağıdaki seçeneklerden hangisi Flash için yazılan bir kodu açıklama satırı haline getirir?
 - a) []
 - b) { }
 - c) .
 - d) ,
 - e) //
4. “Tüm şartların doğru olduğunu sınavan operatör” Seçeneklerden hangisinde doğru olarak verilmiştir?
 - a) <<
 - b) >>
 - c) ||
 - d) &&
 - e) !=
5. Değişken tanımla kuralları aşağıdaki seçeneklerden hangisinde hatalı ifade edilmiştir?
 - a) Değişken isimleri rakamla başlayamaz.
 - b) Kelimeler arasında boşluk bırakılmamalıdır.
 - c) Özel karakterler kullanılmaz.
 - d) ActionScript büyük küçük harf duyarlı değildir.
 - e) Verilecek isimlerin mantıklı, daha sonra okunduğunda anlaşılacak isimler olmalıdır.

6. Sayıları Aralığı (-2.147.483.648) ile (2.147.483.647) arasında olan tam sayı değerlerini alabilen veri türü seçeneklerden hangisinde verilmiştir?

- a) String
- b) Boolean
- c) uint
- d) int
- e) Number

7. Sayıları Aralığı (4.9406564584124654e-324) ile (1.79769313486231e+308) arasında olan tam sayı değerlerini alabilen veri türü seçeneklerden hangisinde verilmiştir?

- a) String
- b) Boolean
- c) uint
- d) int
- e) Number

```
1 var deðiřkenx:uint=3;  
2 var deðiřkeny:uint=4;  
3 trace(deðiřkeny*deðiřkenx);
```

8. Yukarıdaki işlemin sonucu seçeneklerden hangisinde doğru verilmiştir?

- a) 3
- b) 34
- c) 12
- d) 7
- e) 4

```
1 var deðiřkenx:uint=9;  
2 var deðiřkeny:uint=4;  
3 trace(deðiřkenx>deðiřkeny);
```

9. Yukarıdaki işlemin sonucu seçeneklerden hangisinde doğru verilmiştir?

- a) 9
- b) 2
- c) 1
- d) true
- e) false

```
1 var deęiřkenx:uint=9;  
2 deęiřkenx+=11;  
3 trace(deęiřkenx);
```

10. Yukarıdaki işlemin sonucu seçeneklerden hangisinde doğru verilmiştir?

- a) 9
- b) 11
- c) 2
- d) 20
- e) 119

Cevap Anahtarı

1.a, 2.a, 3.e, 4.d, 5.d, 6.d, 7.e, 8.c, 9.d, 10.d

YARARLANILAN KAYNAKLAR

[1] AKBUĞA, M. (2011). Grafik Animasyon- I Erzurum: Atatürk üniversitesi. ATAUZEM.

[2] Adobe Flash Pro CC, Yardımcı el kitabı. <http://helpx.adobe.com/tr/support>

[Erişim tarihi: 30 Ekim 2013].

[3] Adobe Flash, ActionScript Yardımcı el kitabı. [http://help.adobe.com / tr_TR/ActionScript/3.0_ProgrammingAS3/ flash_as3_programming.pdf](http://help.adobe.com/tr_TR/ActionScript/3.0_ProgrammingAS3/flash_as3_programming.pdf)

[Erişim tarihi: 01 Ocak 2014].

[4] MEGEP, Mesleki Eğitim ve Öğretim Sisteminin Güçlendirilmesi Projesi – Etkileşimli Animasyonlar Modülü, Ankara 2012.

[5] SCHULZE Pati.(2004) Macromedia Fireworks 8 Kaynağından Eğitim, Medyasoft Yayınları.

TEMEL PROGRAMLAMA YAPILARI-I



İÇİNDEKİLER

- Fonksiyonlar
- Fonksiyon Tanımlama
- Fonksiyonlara Veri gönderme/Alma
- Olaylar
- Fare Olayları
- Klavye Olayları
- ActionScript ile Animasyon



HEDEFLER

- Bu üniteyi çalıştıktan sonra;
- Animasyon için fonksiyon hazırlama tekniklerini öğrenebilecek,
- Klavye ve Fare olaylarını kullanarak animasyonları kontrol edebilecek,
- Çeşitli nesnelere ActionScript kod yazarak animasyon hazırlayabileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

ANİMASYON TASARIMI

Okt.
Mustafa AKBUĞA

ÜNİTE 11

Giriş

Web içeriklerinde sıklıkla karşılaşılan etkileşimli (klavye ve mouse hareketlerine tepki veren) görsel öğelerin oluşturulmasında ActionScript programlama dili kullanılır. ActionScript programlama dilinde etkileşimli tasarım yapmak için nesnelerin klavye tuşları ve mouse hareketleriyle nasıl kontrol edildiğini, kontrol sırasında hangi komut ve parametrelerin kullanıldığını bilmek gereklidir.

Bu ünite, şekil ya da hareket animasyonlarının ActionScript programlama dilinde Fonksiyon ve Olaylar'ın kullanılarak nasıl yapılabileceği anlatılacaktır.

FONKSİYONLAR

Animasyon için yazılan kodlar içerisinde bir komut veya komut grubunu tekrar tekrar kullanma gerekebilir. Bazen bu gibi durumlarda aynı kod satırlarını defalarca yazılmak gerekir. Bu durum birkaç yüz satırlık kodlamada çok büyük problem teşkil etmeyecektir. Fakat uygulama büyüdükçe hata kontrolünün yapılması da zorlaşacaktır.

Bu gibi durumlarda tasarımcıların yardımına fonksiyonlar yetişmektedir. Fonksiyonlar, tekrar eden kod bloklarının alınıp paketlenmesine yarar ve uygulama içerisinde bu paketin ismi kullanılarak istenilen yerde bu kodlar çalıştırılabilir. Bu sayede daha sonra programda yapılacak değişiklikler ve hata kontrolü daha hızlı bir şekilde yapılır.

Fonksiyon Tanımlama

ActionScript'te kullanılan temel fonksiyon tanımlama formatı aşağıda gösterilmektedir.

```
function fonksiyon_ismi (Girişdeğer_1, Girişdeğer_2,... Girişdeğer_n ):Çıkışdeğeri
{
// foksiyonun çalışması sonucu icra edilecek kodlar..
return çıkış_değişkeni;
}
```

Genel olarak tanımlanan bu fonksiyon yapısının Eylem panelindeki yazım şekli aşağıda gösterildiği gibidir.

```
1
2
3 function fonksiyon_ismi (Girişdeğer_1:int,Girişdeğer_2:String):uint
4 {
5     //Program içerisinde tekrarlı yapılması gereken işler..
6     return Çıkış_değişkeni
7 }
8
```

Bu gelen yapıda verilen değişkenlerin tanımları aşağıda verilmektedir.



Bir fonksiyon az ve öz iş yapmalıdır. Örneğin bir fonksiyonla hem dışarıdan veri giriş işlemi hem de bu verileri ekrana yazdırma işlemi birlikte yapılmamalıdır. Her iki işlem için de ayrı fonksiyonlar tanımlanması daha doğru olacaktır.

- **fonksiyon_ismi**: Fonksiyonların çağrılabilmesi için gerekli olan ismin yazıldığı alanıdır.
- **Girişdeğer_1, Girişdeğer_2... Girişdeğer_n**: Fonksiyonların iş yaparken kullandığı değişken isimleri ve bu değişkenlere ait veri türleri bu alanda tanımlanır.
- **Çıkışdeğeri**: Fonksiyonların yapacağı işten sonra ortaya çıkan sonuç değerinin fonksiyon dışına aktarılmasında kullanılacak veri türü bu alana yazılır. Eğer fonksiyon içerisinde ortaya çıkan değerler dışarı aktarılmıyorsa bu bölüme “**void**” anahtar sözcüğü yazılmalıdır.
- **return çıkış_değişkeni**: Fonksiyon çalıştığında, çalıştırılan kodlar yardımıyla hesaplanan değerler “**return**” anahtar sözcüğü ile **çıkış_değişkeni’ne** aktarılarak geri döndürülür. Çıkış_değişkeni ile belirtilen geri dönüş değerinin, Çıkışdeğeri türü ile uyumlu olması gerekir.

Fonksiyonu çağırıp, çalıştırmak için:

```
13 fonksiyon_ismi (Girişdeğer_1,Girişdeğer_2...,Girişdeğer_n) ;
```

Şeklinde yazılması yeterlidir.

Yukarıda anlatılan işlemleri şimdi de bir uygulama içerisinde görelim.

Uygulama 1: Bu uygulama örneğinde, Karenin alanını hesaplayan bir fonksiyon yazalım. Fonksiyon her çalıştırıldığında girilen kenar uzunluğu değerine göre karenin alanını hesaplayıp sonucu ekrana yazdırsın.

- Yeni bir belge oluşturunuz.
- Sahneye tıklayıp Pencere > Eylemler (F9) komutunu seçiniz.
- Aşağıda verilen kodları Eylem paneline yerleştiriniz.

```
1 function kareninAlanı(kenar:uint):void
2 {
3     //karenin alanı kenar*kenar şeklinde hesaplanır.
4
5     var alan:uint=kenar*kenar
6     trace("karenin alanı:",alan);
7 }
```

- Fonksiyon yukarıdaki kodlarla **kareninAlanı** ismiyle paketleyin.
- Şimdi bu fonksiyon ile kenar uzunluğu 10 olan bir karenin alanı hesaplanmak için aşağıdaki satırı eylem paneline ekleyiniz.

```
13 KareninAlanı(10) ;
```

- Kod çalıştırıldığında sonuç aşağıdaki gibi görülecektir:



Eylem panelinde bir nesneye yönelik hazırlanan kodlar, ilgili nesnelerin sahneden kaldırılması gereken durumlarda, eylem panelinden kaldırılması gereklidir.

Zaman çizelgesi	Çıktı	Derleyici Hataları
	karenin alanı: 100	

Yukarıda verilen fonksiyon tanımına dikkat edilirse *Çıkışdeğeri* olarak “void” sözcüğü kullanılmıştır. Bu sözcüğün kullanıldığı fonksiyonlar yaptığı işlemlerin sonuçlarını döndürmezler ve genellikle ekran çıktısı olarak gösterirler. Fonksiyon tanımında <void> sözcüğü yerine (int, uint vb.) veri türleri yazılacak olursa, bu tip durumlarda hesaplanan değerler <return> sözcüğü yardımıyla fonksiyon dışına kolayca döndürülebilir. Uygulama1’de fonksiyona tek bir değer (kenar uzunluğu) gönderilerek alan hesaplası yaptırılmıştır. Bazen fonksiyonların çalışması için içerisine birden fazla değer göndermek gerekebilir. Bu durum aşağıdaki uygulamada anlatılmaktadır.



"return" ifadesi, sağına yazılan değerlerden sadece birini döndürebilir. "return", birden fazla değeri döndüremez.

Uygulama 2: Bu uygulama ile dikdörtgenin alanını hesaplayan bir fonksiyon hazırlanacaktır. Bir dikdörtgenin alanı kısa kenar ve uzun kenarların çarpımı şeklinde hesaplanır. Bu şekilde alan hesaplayan bir fonksiyon için kısa kenar ve uzun kenar bilgilerinin fonksiyona gönderilmesi gerekmektedir.

Bir fonksiyona birden fazla parametre girilirken, parametreler virgül işareti kullanılarak birbirinden ayrılır.

- Aşağıdaki kodu eylem paneline yazarak fonksiyonu tanımlayınız.

```
1 function AlanHesapla(kisaKenar:uint,uzunKenar:uint):void
2 {
3     var alan:uint=(kisaKenar*uzunKenar);
4     trace("Dikdörtgenin alanı:",alan)
5 }
```

- Aşağıdaki kodu eylem paneline yazarak fonksiyonu çalıştırınız.

```
7 AlanHesapla(3,5);
```

- Animasyonu test ederek, ekran çıktısının aşağıdaki gibi olduğunu kontrol ediniz.

Zaman çizelgesi	Çıktı
	Dikdörtgenin alanı: 15

Yukarıda verilen fonksiyon tanımında kısaKenar ve uzunKenar olmak üzere iki ayrı parametre kullanılmış ve bu parametreler ile hesaplanan değer çıktı ekranında gösterilmiştir.

Hesaplanan değer fonksiyon dışına alınabilmesi için yukarıdaki alan uygulamasını yeniden aşağıdaki gibi düzenleyerek eylem paneline yazılması gereklidir.

```

1  function AlanHesapla(kisaKenar:uint,uzunKenar:uint):uint
2  {
3      return(kisaKenar*uzunKenar);
4  }

```

- Fonksiyonu çağırıp işlem yapması için aşağıdaki kodları eylem paneline ekleyiniz.

```

5  var alan:uint=(AlanHesapla(3,5));
6  trace("Dikdörtgenin alanı:",alan;

```

- Animasyonu test ederek, ekran çıktısının aşağıdaki gibi olduğunu kontrol ediniz.

Zaman çizelgesi	Çıktı
	Dikdörtgenin alanı: 15

Önceki uygulamaların aksine bu uygulamada fonksiyon tanımlanırken “*void*” anahtar sözcüğü yerine fonksiyonun ne tür veriyi geri döndüreceğini gösteren (*uint*) yazılmıştır. Uygulamanın test edilmesi sırasında: 5. satırdaki kod ile kısaKenar: 3 , uzunKenar :5 olarak AlanHesap fonksiyonuna gönderilmiş sonrasında ise hesaplanan değer “*alan*” değişkenine aktarılarak çıktı ekranında gösterilmiştir.

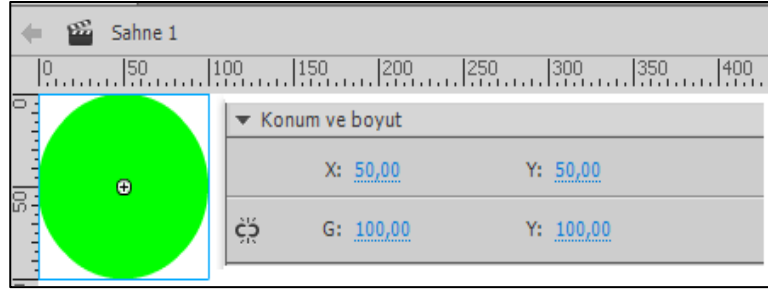
Uygulama 3: Önceki örnek uygulamalarda tanımlanan fonksiyonlar bir ya da daha fazla parametre alarak işlem yapmaktadır. Fakat bu örnek uygulamada tanımlanan fonksiyon ise parametre almayacaktır.

Bu uygulamada “*yesil_daire*” isimli Şekil 11.1’de gösterilen nesnenin ekrandaki konumunu “*x ekseninde 300px*” ve “*y ekseninde 300 px*” noktasına getirecek bir fonksiyonu yazmak için:

- Dosya>Yeni>ActionScript 3.0 komutu verilerek yeni bir belge oluşturunuz.
- Sahnemize 100*100 px büyüklüğünde Şekil 11.1’de özellikleri ve şekli gösterilen daire oluşturunuz ve X ekseninde: 50, Y ekseninde: 50 konumuna yerleştiriniz.
- Film klipi hâline getirip örnek adı olarak *yesil_daire* ismini veriniz.
- Sahneye tıklayıp Pencere > Eylemler (F9) komutunu seçiniz.
- Fonksiyon ismini *hareketEt* olarak belirleyin. İsterseniz değişken tanımlama kuralları çerçevesinde istediğiniz ismi verebilirsiniz.



Bir Flash dosyası swf olarak yayınlandıktan sonra içerisinde yer alan kodlar bir daha değiştirilemez.



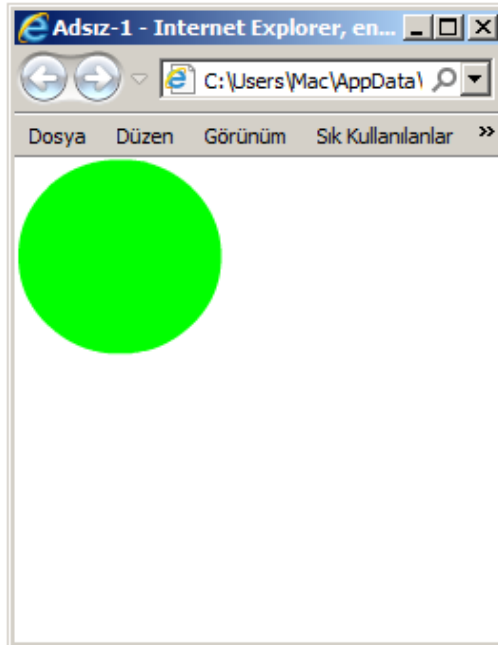
Şekil 11.1 Çizilen daire şekli ve Konum ve boyut özellikleri

```

1  function hareketEt():void
2  {
3      this.yesil_daire.x=300;
4      this.yesil_daire.y=300;
5  }
6

```

- Animasyonu test ediniz.
- Şekil 11.2’de görüldüğü gibi yesil_dairenin yeri değişmeyecektir.



Şekil 11.2 Çizilen daire şekli ve kodun test ekranı

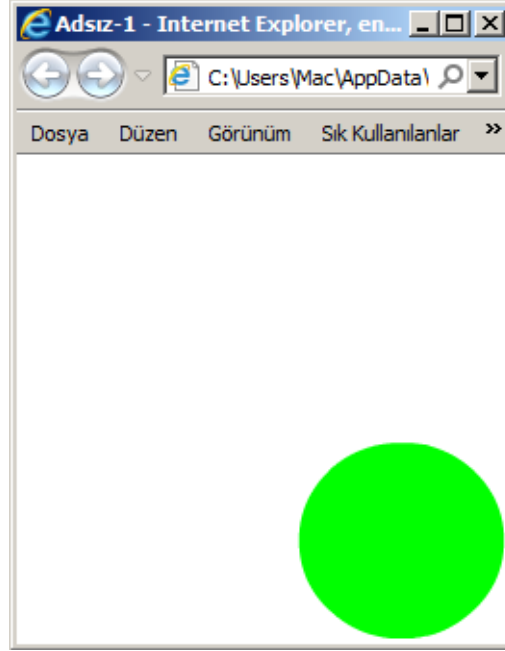
- Fonksiyon tanımlanmış, fakat tetiklenmediği için kodlar çalışmamıştır.
- Tetiklemek için fonksiyon çağırma komutunu eylem paneline aşağıda şekildeki gibi eklememiz gereklidir.

```

6  hareketEt();

```

Şekil 11.3’de görüldüğü gibi yesil_daire’nin yeri fonksiyon içerisindeki kodlara göre değişecektir



Şekil 11.3 Fonksiyon çalıştırma komutu eklendikten sonraki test ekranı

OLAYLAR (EVENT)

Olay, klavye, mouse gibi giriş aygıtlarından birisinin kullanılmasıyla ortaya çıkan bir tetikleyicidir. ActionScript 3.0 ile artık sahnede gerçekleşen her şey, bir olay olarak kabul edilmiştir. Örneğin; farenin herhangi bir yere tıklanması, klavyeden bir şeyler yazılması, sahnenin çalışması, zaman çizgisi üzerinde bir frame'in çalışması vb. şeyler olay olarak kabul edilmektedir. Olayların gerçekleşmesi yazılmış fonksiyonların tetiklenmesini sağlar.

ActionScript içerisinde kullanılanılabilen fare olayları, klavye olayları ve zamana bağlı olaylar olmak üzere üç tip olay vardır. Bu olaylar sistemin otomatik tetiklediği ve kullanıcının tetiklediği olmak üzere iki farklı şekilde harekete geçerler.

Otomatik tetiklenen olaylar, program içinde gerekli şartlar sağlandığında kendiliğinden gerçekleşir. Kullanıcının tetiklediği olaylarda ise mutlaka fare ya da klavye üzerinden bir takım davranışların (farenin bir yere tıklaması, bir tuşa basılması vb.) kullanıcı tarafından gerçekleştirilmesi beklenir.

Bir nesne için ayrı ayrı olaylar tanımlanabilir. Örneğin; internet üzerinden oyunlar oynanırken bazı durumlarda farenin tıklanması yetmeyebilir. Farenin belli bir bölgede dolaştırılması ya da klavyeden belli bir tuşa ya da tuşlara birlikte basılması gerekebilir. Tanımlanan bu tip olaylar olay dinleyicileri tarafından izlenir.

Olayın gerçekleşmesi durumunda dinleyici içerisinde tanımlanan fonksiyon çalıştırılır.

Fare Olayları

Bir Fare olayının oluşması, sahne üzerindeki nesnelerin mevcut durumlarında bir değişiklik olması anlamına gelir.

ActionScript 3.0'da kullanılan fare olayları aşağıda verilmiştir.

CLICK: Farenin bir nesneye sol tuş ile tıklanıp bırakılma olayıdır.

- `MouseEvent.CLICK` şeklinde yazılır.

DOUBLE_CLICK: Farenin bir nesneye çift tıklama olayıdır.

- `MouseEvent.DOUBLE_CLICK` şeklinde kullanılır

MOUSE_DOWN: Farenin sol tuşuna basma olayıdır. Click' ten farkı sadece basma ile çalışmasıdır.

- `MouseEvent.MOUSE_DOWN` şeklinde kullanılır.

MOUSE_UP: Farenin sol tuşunu bırakma olayıdır.

- `MouseEvent.MOUSE_UP` şeklinde kullanılır.

MOUSE_OVER: Farenin nesne üzerine gelme olayıdır.

- `MouseEvent.MOUSE_OVER` şeklinde kullanılır.

MOUSE_OUT: Farenin bir nesne üzerinden ayrılma olayıdır.

- `MouseEvent.MOUSE_OUT` şeklinde kullanılır.

MOUSE_MOVE: Farenin nesne üzerinde gezme olayıdır.

- `MouseEvent.MOUSE_MOVE` şeklinde kullanılır.

MOUSE_WHEEL: Farenin tekerleğinin döndürülme olayıdır.

- `MouseEvent.MOUSE_WHEEL` şeklinde kullanılır.

ROLL_OVER: Farenin sol tuşunun basılı tutularak nesne üzerine gelme olayıdır.

- `MouseEvent.ROLL_OVER` şeklinde kullanılır.

ROLL_OUT: Farenin sol tuşunun basılı tutularak nesne üzerinden ayrılma olayıdır.

- `MouseEvent.ROLL_OUT` şeklinde kullanılır.

Klavye Olayları

Klavyeden basılan tuşların hareketlerini algılamak için kullanılan olaylardır. Bir klavye tuşuna basma ve basılı tuşu bırakma olayı olmak üzere iki şekilde kullanılabilir.

Bunlar ;

KEY_DOWN: Klavyeden herhangi bir tuşa basma durumunda çalışır.

`KeyboardEvent.KEY_DOWN` şeklinde kullanılır.

KEY_UP: Klavye üzerindeki basılı olan bir tuşun bırakılması durumunda ortaya çıkar.

`KeyboardEvent.KEY_UP` şeklinde kullanılır.

Zamana Bağlı Çalışan Olaylar

Animasyon tasarımı içerisinde belirlenen zaman aralıklarında otomatik olarak çalışan olaylar da vardır. Bunlar içerisinde en çok kullanılanlar **ENTER_FRAME** ve **TIMER -TIMER_COMPLETE** olaylarıdır.

ENTER_FRAME: Bu olay, sahne çalışma hızına bağlı olarak zaman içinde otomatik olarak çalışan fonksiyonları tetikler. Sahne çalıştırıldığı anda tetiklenmeye başlar ve sahne hızına göre uygulama kapatılana kadar çalışmaya devam eder. Sahne çalışma hızı değiştirildiğinde tetiklenen fonksiyonun çalışması da değişir. Uygulama süresince tetiklenen fonksiyon sürekli çalışacağından işi bittiğinde tetikleyicinin etkisini durdurmak gerekir.

TIMER ve TIMER_COMPLETE: Bu olaylarda, sahne çalışmaya başladığı anda tetiklenme başlamaz. Olayın ne zaman başlayacağı ve durdurulacağı timer nesnesinin parametreleri ile belirlenir. Timer nesnesinin kullanımı incelemek için Eylem kodlarının yer aldığı 14. üniteye bakınız.

Olay Dinleyiciler (EventListener)

Olayları Eylem panelinde tanımlamak ve onları animasyon davranışlarında etkili hâle getirmek için Olay dinleyiciler kullanılır.

Olay dinleyicileri, sahneye eklemek ve tetikleyeceği fonksiyonu tanımlamak aşağıdaki şekilde yapılmaktadır.

```

this.Nesne_adı.addEventListener(Olay_çeşidi, fonksiyon_adı);

function fonksiyon_adı ( event: Olay_çeşidi ) :Veritürü
{
    // fonksiyonun çalışması sonucu icra edilecek kodlar.
    return çıkış_değişkeni;
}

```

Yukarıdaki gibi tanımlanan fonksiyonun Eylem panelindeki yazım şekli aşağıdaki gibidir.

```

1  this.Nesne_adı.addEventListener(Olay_çeşidi, fonksiyon_adı);
2  function fonksiyon_adı(event:Olay_çeşidi):Veritürü
3  {
4      // fonksiyonun çalışması sonucu icraa edilecek kodlar.
5      return çıkış_değişkeni;
6  }

```

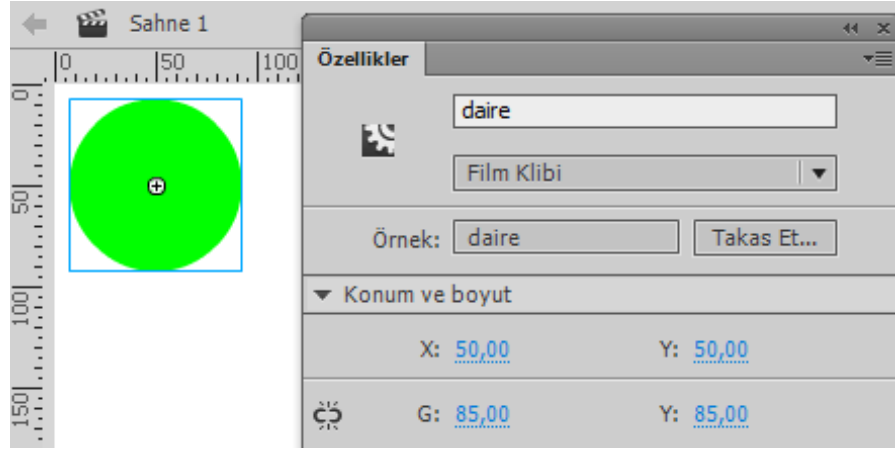
- **this.Nesne_Adı:** Olay dinleyicileri genelde bir nesneyle birlikte kullanılır. Burada *Nesne_adı*, Sahnedeki “Örnek adı”nı *this.Nesne_Adı* ise bu nesnenin sahnedeki hedef yolunu gösterir.
- **Olay_çeşidi:** Olay dinleyicinin ne tip bir olayın gerçekleşmesini bekleyeceğini gösterir.
- **fonksiyon_adı:** Olayın gerçekleşmesi durumunda çağrılacak fonksiyonun isminin yazıldığı alandır.

Aşağıdaki uygulamalarda Olay çeşitlerine göre ve Olay dinleyicilerin nasıl tanımlandığı görülmektedir.

Uygulama 4: Bu uygulamada sahne üzerine bir daire çizilerek, fare ile bu daire üzerine gelindiğinde “fare dairenin üzerinde” mesajı vermesi istenmektedir.

Bu işlem için;

- Sahneye Şekil 11.4’te gösterildiği gibi bir daire çizin ve daireyi film klibi hâline dönüştürün.
- Bu daire sembolü özelliklerini girin ve “Örnek Adı” bölümüne “daire” ismini verin.



Şekil 11.4 Çalışma alanına çizilmiş ve film klibine çevrilmiş bir daire ve özellikleri

- konumHesapla fonksiyonunu aşağıdaki gibi yazın.

```
2 function konumHesapla(event:MouseEvent):void
3 {
4     trace("fare şimdi dairenin üzerinde");
5 }
```

- Daire nesnesinin üzerine fareyle gelindiğinde *konumHesapla* fonksiyonun çalışması ve “fare şimdi dairenin üzerinde” çıktısını vermesi için aşağıdaki olay dinleyicisini ekleyin.

```
1 this.daire.addeventListener(MouseEvent.CLICK, konumHesapla);
```

Yukarıdaki verilen kod ile Olay dinleyicinin hangi olay karşısında tetikleneceğini göstermektedir. Farenin nesne üzerine gelmesi burada tanımlanan bir olaydır.

- Fare ile dairenin üstüne geldiğinde konumHesapla fonksiyonu çalışacaktır.
- Yukarıda kodları verilen *konumHesapla* fonksiyonu, fonksiyonlar konusunda öğrendiğiniz bilgilerden oluşmasına rağmen içerisine gönderilecek parametre önceki tanımlamalardan farklıdır. Burada fonksiyona mouse olayı olduğunu anlatmak için parametre olarak *event:MouseEvent* söz dizimi eklenmiştir.

Aşağıda, Eylem paneline yazılmış kodların tamamı gösterilmiştir. Bu kodların Eylem paneline doğru olarak yazıldığını kontrol ederek filmi klibini test ediniz. Şekil 11.5’te gösterildiği gibi fare daire üzerine her geldiğinde(aşağıdaki test de üç kez geldiği varsayılmaktadır) çıkış ekranına “fare şimdi dairenin üzerinde” şeklinde uyarı yazısı görülecektir.



Çalışma alanında çizilen nesnelerin kontrolü için yazılan kodlar grafiklerin sembole dönüştürüldükten sonra kullanılabilir.

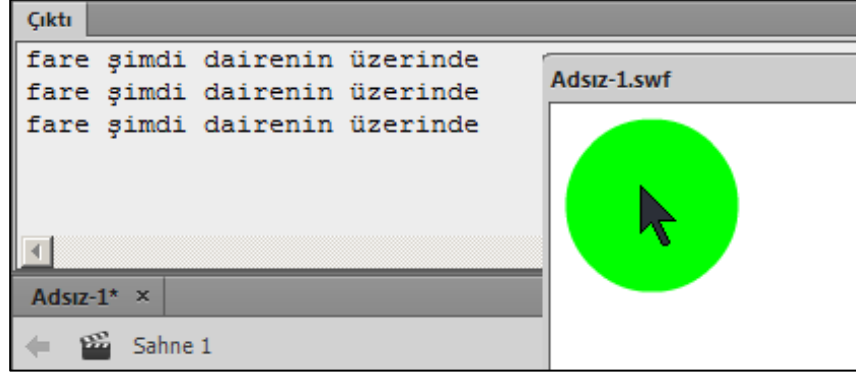


Çalışan olayları silmek için kullanılan *removeEventListener* komutu bir fonksiyon ya da bir karar yapısının içerisinde kullanılırsa daha iyi sonuç alınır.


```

1 import flash.events.MouseEvent;
2 this.daire.addEventListener(MouseEvent.CLICK, konumHesapla);
3 function konumHesapla(event:MouseEvent):void
4 {
5     trace("fare şimdi dairenin üzerinde");
6 }

```



Şekil 11.5 Olay yöneticisinin çalışması

Yukarıdaki kod incelendiğinde 1. satırda import ile başlayan aşağıdaki kod görülecektir. Bu kod Event sınıfının mouse metodlarını uygulama içerisine dâhil etmek için uygulamaya flash tarafından otomatik olarak eklenmiştir.

```
1 import flash.events.MouseEvent;
```

Uygulama içerisinde nesneye atanan bir olay çalıştıktan sonra görevinin sonlandırılması istenebilir. Örneğin, Eğer mouse bir kaç defa daire üzerine getirilirse Çıktı ekranında yukarıda Şekil 11.5'te gösterildiği gibi bir kaç defa aynı mesaj alt alta yazılacaktır. Verilen bu uyarıları kontrol etmek amacıyla *removeEventListener* komutunu kullanılabilir.

Yukarıda yaptığımız uygulamada olay dinleyicisini olay bir defa çalıştıktan sonra görevini sonlandırmak amacıyla :

- Aşağıda gösterilen komutunu eylem paneline ekleyiniz.

```
daire.removeEventListener(MouseEvent.CLICK, konumHesapla);
```

Eylem paneline yukarıdaki kod eklendikten sonra uygulamadaki ActionScript kodun en son hâli aşağıdaki şekilde görülecektir.

```

1 import flash.events.MouseEvent;
2 this.daire.addEventListener(MouseEvent.CLICK, konumHesapla);
3 function konumHesapla(event:MouseEvent):void
4 {
5     trace("fare şimdi dairenin üzerinde");
6     daire.removeEventListener(MouseEvent.CLICK, konumHesapla);
7 }

```

Bu durumda mouse dairenin üzerine bir kez geldinde "fare şimdi dairenin üzerinde" diye mesaj verir ve 6. Satırdaki *<daire.removeEventListener>* ile başlayan kod mouse olayının görevini sonlandırır ve mouse dairenin üzerine tekrar gelse bile mouse olayı artık izlenmediğinden çıktı ekranına hiçbir mesaj verilmeyecektir.



Animasyon kontrolünde en çok, CLICK ve DOUBLE_CLICK fare olayları kullanılmaktadır.

Uygulama 5: Bu uygulamada klavye üzerine basılan tuşun ASCII kod (Amerikan Standart Kodlama Sistemi) karşılığını ekranda gösterecek bir program yazılacaktır.

Aşağıda gösterilen ActionScript kodlar bir sahne üzerinde Eylem paneline yazılıp çalıştırıldığında: Sahne üzerinde sırasıyla klavyeden basılan “AOF” harflerine karşılık ascii kod karşılıkları Çıktı ekranına aşağıdaki şekilde yazılacaktır.

```
1 import flash.events.KeyboardEvent;
2 function klavye(event:KeyboardEvent):void
3 {
4     trace(event.keyCode);
5 }
6 stage.addEventListener(KeyboardEvent.KEY_DOWN,klavye);
```

Çıktı

```
65
79
70
```

Yukarıdaki çıktı ekranında 65 değeri “A”, 79 değeri “O”, 70 değeri “F” harfleri, alt alta ascii kod karşılıkları şeklinde gösterilmiştir.

Yukarıdaki verilen kodlar incelendiğinde Eylem panelinin ilk satırdaki;

```
import flash.events.KeyboardEvent;
```

kodunu siz eylem paneline yazmasanız da Adobe Flash tarafından bu kod 1. Satıra otomatik olarak eklenecektir. Bu kod, Event sınıfının klavye metodlarını uygulama içerisine dâhil etmekte kullanılır.

Yukarıdaki programın kodlarına dikkat edilirse klavye fonksiyonu 2 . satırda ve Olay dinleyici ise 6. satırda tanımlanmıştır. Yani Olay dinleyicisinden önce tanımlanmıştır. Bu fonksiyon istenirse bir önceki örnekte olduğu gibi olay dinleyiciden sonar da tanımlanabilir. Yazılış sırasının bir önemi yoktur. Çünkü fonksiyonlar isimleriyle çağrılmadıkça, içinde yazan kodlar çalıştırılmaz.

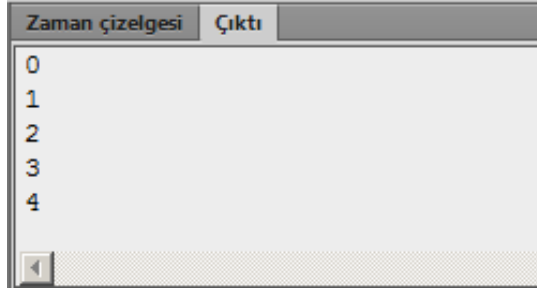
Uygulama 6: Bu örnek uygulamada birer birer kendiliğinden artan sayıların ekranda görüntülenmesi anlatılacaktır.

Aşağıdaki verilen kodlar eylem paneline yazılıp test edildiğinde; uygulama çalışmaya başlayacaktır. Olayları izleyici kod da uygulamayla beraber yürürlüğe girip başla fonksiyonunu tetikleyerek onu da çalıştırır. Çalışan fonksiyon içindeki sayıcı değişkeni, sahne hızına göre kendini birer birer arttırır. Şayet sahne hızı arttırılırsa sayıcı değişkenin değeri de aynı oranda artacaktır. Bu uygulamanın sonucu Şekil 11.6 ‘da gösterildiği gibidir.



Adobe Flash ile hazırlanan oyun kodlarının hazırlanmasında genelde klavye üzerindeki ok tuşları nesnelerin hareket kontrolünde kullanılır.

```
1 import flash.events.Event;
2
3 var sayıcı:uint=0;
4 stage.addEventListener(Event.ENTER_FRAME,basla);
5 function basla(event:Event):void
6 {
7     trace(sayıclı++);
8 }
```



Şekil 11.6 Say değişkeninin sahne çalıştığındaki artan değeri

ACTIONSCRIPT İLE ANİMASYON

Önceki ünitelerde anlatılan, zaman çizelgesi ve sahne üzerinde grafik nesneleri yardımıyla tasarlanan animasyonlar ActionScript kodları kullanılarak yapılabilir. Animasyon tasarımında kullanılan ActionScript komutları; nesnelerin şeklini değiştiren (şekil animasyonu yapan) ve onları hareketlendiren (hareket animasyonu yapan) komutlar olmak üzere iki gruba ayrılır.

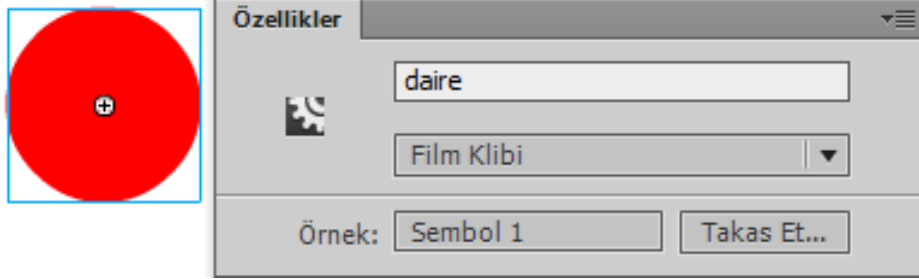
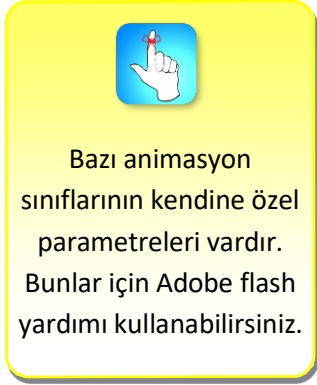
1. Şekil Animasyonu yapan komutlar

ActionScript kodları kullanarak uygulayabileceğimiz on adet şekil animasyonu mevcuttur. Bunlar;

- **1-Blinds:** Nesneye Panjur efekti verir.
- **2-Rotate:** Nesneyi döndürür.
- **3-Wipe:** Nesne üzerine silme efekti yapar.
- **4-Zoom:** Yakınlaştırma efekti uygular.
- **5-Fly:** Uçma efekti yapar.
- **6-Iris:** Açılıp kapanan iris efekti yapar.
- **7-Photo:** Görünür kaybolur efekti yapar.
- **8-Squeeze:** Sıkma efekti uygular.
- **9-PixelDissolve:** Piksel piksel yok etme ya da oluşturma efekti yapar.
- **10-Fade:** Nesneyi yok etme ve görünür yapma efekti uygular.

Uygulama 7: Bu uygulamada yukarıda isimleri verilen şekil animasyonları test edilecektir. Bu işlem için;

- Sahneye 85x 85 piksel büyüklüğünde bir daire çizersiniz.
- Dairenin rengini Şekil 11.7’de gösterildiği gibi kırmızı olarak belirleyiniz.



Şekil 11.7 Uygulama için hazırlanan kırmızı daire şekli

- Film klibi hâline getirip örnek ismini daire olarak belirleyiniz.
- Aşağıda gösterilen ActionScript kodlarını Eylem paneline yazınız.

Nesneye sırayla animasyon türlerini uygulayarak deneyiniz. Aşağıda 5. satırdaki “type:Zoom” yazan Alana “Type: Rotate” veya diğer şekil animasyonlarından birinin ismini yazarak, animasyon tipini değiştirebilirsiniz.

Aşağıda “Zoom” animasyonunda kullanılan type, direction, duration, easing parametreleri tüm animasyonlar için ortak parametrelerdir.

```

1  import fl.transitions.*;
2  import fl.transitions.easing.*;
3  TransitionManager.start(this.daire,
4  = {
5      type:Zoom,
6      direction:Transition.OUT,
7      duation:2,
8      easing:Elastic.easeInOut,
9      numStrips:2,
10     dimension:0
11 }
12 );
    
```

Yukarıdaki eylem panelinde kullanılan komutların ne anlama geldiği aşağıda sıra numaraları ile birlikte açıklanmıştır.

- 1-2 2: Adobe Flash içindeki şekil animasyon metodlarının uygulama içerisine dâhil edilmesinde kullanılır.
- 3: Bu satırda, Daire nesnesi için animasyon yöneticisi başlatıldı.
- 4: Bu satırdan sonra animasyon için parametreler verilmeye başlayacaktır.
- 5: Bu satırlarda, animasyon tipi verilir.

- 6: Bu satırda, animasyon yönü belirlenir.
- 7: Bu satırda, animasyonun gerçekleşmesi için gerekli zaman belirlenir.
- 8: Bu satırda, animasyon türü belirlenir.
- 9: Bu satırda, panjur sayısı belirlenir.
- 10: Bu satırda, 0 değeri yatay ve 1 değeri dikey panjur oluşturulur.
- 11: Bu satırda, parametre verme işlemi sonlandırılır.
- 12: Bu satırda, animasyon yöneticisi tanımı sonlandırılır.

2. Hareket Animasyonu yapan komutlar

Hareket animasyonları, ActionScript kodlarını kullanarak nesneleri hareket ettirme, döndürme, büyütme, küçültme ve görünürlük özelliğini değiştirme işlemleri yapmayı sağlar.

Kodlarla yapılan yedi çeşit hareket animasyonu vardır.

- **1-X:** Nesneyi X ekseninde hareket ettirmek için kullanılır.
- **2-Y:** Nesneyi Y ekseninde hareket ettirmek için kullanılır.
- **3-scaleX, Scale Y:** Nesnenin boyutlarını değiştirmek için kullanılır.
- **4-width, height:** Nesnenin genişlik ve yükseklik değerlerini değiştirmek için kullanılır.
- **5-rotation:** Nesneyi sağa sola döndürmek için kullanılır.
- **6-alpha:** Nesnenin saydamlık özelliği değiştirmek için kullanılır.
- **7-visible:** Nesnenin görünürlük değiştirmek için kullanılır.

Uygulama 8: Bu uygulamada Sahne üzerinde yer alan “Animasyon” metni ekranda ilk önce görülmeyecek, sonrasında ise yavaş yavaş renklenerek ekrana gelmesi sağlanacaktır. Bu uygulama için;

- Yeni bir animasyon belgesi açın ve metin aracını kullanarak “Animasyon” ifadesini yazın. (Uygulamada renk: Mavi boyut:70 seçilmiştir)
- Yazdığınız ifadeyi film klibine dönüştürün ve örnek isim olarak dersadı olarak belirleyin.
- ActionScript kodlarını aşağıda gösterildiği gibi Eylem paneline yazın.



ActionScript kodları kullanarak yapılan efektler nesneye uygun olarak seçildiğinde güzel sonuçlar ortaya çıkarır.



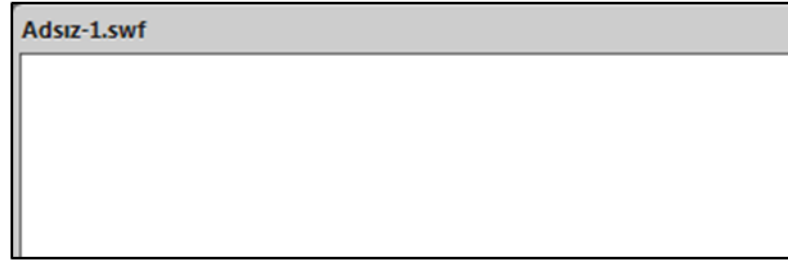
Sahne üzerindeki bir nesneye birden fazla animasyon türü uygulanabilir.

```
1  import fl.transitions.Tween;
2  import fl.transitions.easing.*;
3  var hareketAnimasyonu:Tween=new Tween
4  (dersadı,
5   "alpha",
6   Elastic.easeInOut,
7   0.001,
8   1,
9   2,
10  true);
```

Yukarıdaki eylem panelinde kullanılan komutların ne anlama geldiği aşağıda sıra numaraları ile birlikte açıklanmıştır.

- 1-2** **2:** Bu satırlarda, Adobe Flash içindeki hareket animasyon metodlarının uygulama içerisine dâhil edilmesinde kullanılır.
- 3:** Bu satırda, “hareketAnimasyonu” isimli yeni bir animasyon tanımlanmaya başlanır.
- 4:** Bu satırda, “dersadı” film klibi için animasyon parametreleri verilmeye başlandı.
- 5:** Bu satırlarda, animasyon tipi verilir.
- 6:** Bu satırda, animasyon türü belirlenir.
- 7:** Bu satırda, nesne için verilen şeffaflık komutunun alacağı başlangıç değeri verilir.
- 8:** Bu satırda, nesne için verilen şeffaflık komutunun alacağı son değer verilir
- 9:** Bu satırda, animasyonun tekrarlanma sayısı verilir.
- 10:** Bu satırda verilen “*true*” değeri tanımlanan animasyonu etkin hâle getirir. (animasyonun etkisini ekranda görebiliriz). Şayet bu “*true*” yazılan yere “*false*” değeri yazılsa idi, tanımlanan animasyon etkisiz hâle gelirdi (animasyonun etkisini ekranda göremezdik).

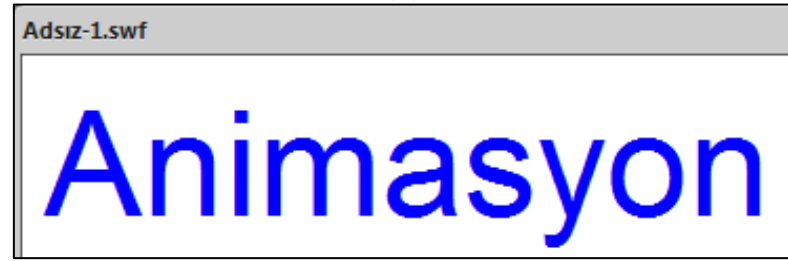
Yukarıdaki verilen kodlar eylem paneline yazılıp test edildiğinde; ilk önce Şekil 11.8 (a) ile verilen şekil ekrana gelecektir. Satır 7’de 0.001 ile verilen değer metnin ilk “alpha” değeri olduğundan yazı görülmez. Sonrasında “alpha” değeri yavaşça artırıldığından metin Şekil 11.8 (b) gibi silik görülür. En sonunda ise metin Şekil 11.8 (c) deki gibi tam renkli görülecektir. Çünkü “alpha” oranı en yüksek değer olan 1’e ulaşmıştır.



(a)



(b)



(c)

Şekil 11.8 Metne uygulanan hareket animasyonun test görüntüleri



Özet

- Animasyon tasarlarken hazırlanan kodların bir kısmı, tasarım içerisinde bir çok yerde kullanmak gerekebilir. Tekrar edilmesi gereken kod bloklarını alıp paketleme işlemine Fonksiyon denir. Fonksiyonlar tanımlanırken function anahtar sözcüğü kullanılır. Fonksiyonlar, bir program içerisinde olaylar veya fonksiyon isimleri kullanarak çağırılmadıkça çalışmazlar.
- Olaylar, klavye, mouse gibi giriş aygıtlarından birisinin kullanılmasıyla ortaya çıkan bir tetikleyicidir. ActionScript dilinde sahnede farenin herhangi bir yere tıklaması, klavyeden bir şeyler yazılması ve zaman çizgisi üzerinde bir frame'in çalışması gibi vb. şeyler olay olarak kabul edilmektedir.
- ActionScript içerisinde kullanılabilecek fare olayları, klavye olayları ve zamana bağlı olaylar olmak üzere üç tip olay vardır. Bu olaylar sistemin otomatik tetiklediği ve kullanıcının tetiklediği olmak üzere iki farklı şekilde harekete geçerler. Olayların gerçekleşmesi neticesinde yazılmış fonksiyonların tetiklenmesini sağlar.
- Her nesne için birden fazla olay tanımlanabilir. Tanımlanan bu olaylar olay dinleyicileri tarafından izlenir.
- Olaylar dinleyiciler, addEventListener anahtar sözcüğü kullanılarak tanımlanır. Nesne üzerindeki bir olay çalıştıktan sonra yenden birdaha çalışmasını engellemek için silmek gerekir. Olayları silmek için removeEventListener anahtar sözcüğü kullanılır.
- Kullanıcı tarafından tetiklenen Fare olayları, [CLICK, DOUBLE_CLICK, MOUSE_DOWN, MOUSE_UP, MOUSE_OUT, MOUSE_MOVE, MOUSE_OVER, MOUSE_WHEEL, ROLL_OVER, ROLL_OUT] olmak üzere 10 çeşittir. Fare olaylarında en çok CLICK (Tıklama) olayı kullanılır.
- Klavye olayları [KEY_DOWN ve KEY_UP] olmak üzere 2 çeşittir. Klavye olaylarından en çok KEY_DOWN (Tuşa basma) olayı kullanılır.
- Zamana bağlı tetiklenen olaylar ise ENTER_FRAME, TIMER ve TIMER_COMPLETE olmak üzere 3 çeşittir. Bunlardan en çok ENTER_FRAME (Film karesiyle başla) olayı kullanılır.
- Zaman çizgisi üzerinde yapılan animasyonların şekil ve hareket gerektiren bölümleri ActionScript kodlar yazarak da yapılabilir.
- ActionScript kodlarını kullanarak uygulayabileceğimiz, [Blinds, Rotate, Wipe, Zoom, Fly, Iris, Photo, Squeeze, PixelDissolv, Fade] 10-adet şekil animasyonu mevcuttur.
- Hareket animasyonları ise nesnelerin [x, y, width, height, scaleX, scaleY, rotation, alpha] özellikleri değiştirilerek yapılır.
- Bir nesneye hem şekil hem de hareket animasyonu birlikte uygulanabilir.

DEĞERLENDİRME SORULARI

1. Aşağıdaki seçeneklerden hangisi doğru fonksiyon tanımını gösterir?
 - a) `function void:isim(){};`
 - b) `function isim():void{};`
 - c) `fonksiyonisim();`
 - d) `function isim():`
 - e) `fonksiyonismi(){};`
2. Aşağıdaki seçeneklerden hangisi doğru olay tanımını gösterir?
 - a) `nesneadı. addEventListener(Olay, tetiklenmesi gereken fonksiyon );`
 - b) `addEventListener.nesneadı.(Olay, tetiklenmesi gereken fonksiyon );`
 - c) `addEventListener.nesneadı.( tetiklenmesi gereken fonksiyon, Olay);`
 - d) `removeListener.nesneadı.( tetiklenmesi gereken fonksiyon, Olay);`
 - e) `nesneadı. removeEventListener(tetiklenmesi gereken fonksiyon, Olay);`
3. Aşağıdaki seçeneklerden hangisi fare tetikleme olayı değildir?
 - a) `ROLL_OVER`
 - b) `DOUBLE_CLICK`
 - c) `KEY_DOWN`
 - d) `MOUSE_OVER`
 - e) `ROLL_OUT`
4. Seçeneklerden hangisi komut yazmak yerine ilgili panelden hazır kod seçimine imkân sağlar?
 - a) Kod parçacıkları
 - b) Kütüphane
 - c) Özellikler
 - d) Bileşenler
 - e) Geçmiş
5. Fonksiyonlarda parametreden sonra yazılan kelimesi, fonksiyonun geriye değer döndürmediği anlamına gelir
Cümlede boş bırakılan yere aşağıdakilerden hangisi getirilmelidir?
 - a) `int`
 - b) `trace`
 - c) `var`
 - d) `function`
 - e) `void`

6. ActionScript kodu yazılarak oluşturulan “Squeeze” animasyon tipi nesne üzerinde hangi etkiyi yapar?
 - a) Nesneye uęma efekti uygular
 - b) Nesneye sıkma efekti uygular
 - c) Nesneye panjur efekti verir
 - d) Nesneye yakınlaştıırma efekti verir
 - e) Nesneye dönme efekti verir
7. ActionScript kodu yazılarak oluşturulan “Fly” animasyon tipi nesne üzerinde hangi etkiye sahiptir?
 - a) Nesneyi gizler
 - b) Nesneyi döndürür
 - c) Nesneye sıkma efekti uygular
 - d) Nesneye panjur efekti verir
 - e) Nesneye uęma efekti uygular
8. Hareket animasyonları nesnelerin özelliklerini kullanarak yapılır. Aşağıdaki seçeneklerden hangisi nesne animasyonları için kullanılmaz?
 - a) X,Y
 - b) Width
 - c) Height
 - d) Alpha
 - e) Dissolve
9. Seçeneklerden hangisi nesneyi yok etme ve görünür yapma efekti uygular?
 - a) Zoom
 - b) Fade
 - c) Iris
 - d) Photo
 - e) PixelDissolve
10. Seçeneklerde hangisi ActionScript dili kullanılarak yapılan animasyonlar için kullanılan ortak parametrelerden biri değildir?
 - a) Type
 - b) Şekil arası animasyonları
 - c) Direction
 - d) Duration
 - e) Degress

Cevap Anahtarı

1.b, 2.a, 3.c, 4.a, 5.e, 6.b, 7.c, 8.e, 9.b, 10.e

YARARLANILAN KAYNAKLAR

AKBUĞA, M. (2011). Grafik Animasyon- I Erzurum: Atatürk üniversitesi. ATAUZEM.

Adobe Flash Pro CC, Yardımcı el kitabı. <http://helpx.adobe.com/tr/support>

[Erişim tarihi: 30 Ekim 2013].

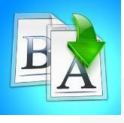
Adobe Flash, ActionScript Yardımcı el kitabı. [http://help.adobe.com / tr_TR/ActionScript/3.0_ProgrammingAS3/ flash_as3_programming.pdf](http://help.adobe.com/tr_TR/ActionScript/3.0_ProgrammingAS3/flash_as3_programming.pdf)

[Erişim tarihi: 01 Ocak 2014].

MEGEP, Mesleki Eğitim ve Öğretim Sisteminin Güçlendirilmesi Projesi – Etkileşimli Animasyonlar Modülü, Ankara 2012.

SCHULZE Pati.(2004) Macromedia Fireworks 8 Kaynağından Eğitim, Medyasoft Yayınları.

TEMEL PROGRAMLAMA YAPILARI-II



İÇİNDEKİLER

- Karar Yapıları
- "if" İfadesi
- "if else" İfadesi
- Döngü Yapıları
- For Döngüsü
- While Döngüsü
- Nesne Tabanlı Programlama
- Paket tanımlama



HEDEFLER

- Bu üniteyi çalıştıktan sonra;
- "if" ifadesini kullanarak karar yapıları oluşturabilecek,
- For ve While döngü yapılarını kullanabilecek,
- ActionScript dilinde Paket ve Sınıf tanımlayarak, Nesne tabanlı programlama yapabileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

ANİMASYON TASARIMI

Okt.
Mustafa AKBUĞA

ÜNİTE 12

Giriş

Önceki ünitelerde ActionScript diline ait veri türleri, Operatörler, Fonksiyonlar ve Olay çeşitlerini anlatmıştık. Bu ünite de anlatılacak konular ise iki ana bölümeden oluşmaktadır. Birinci bölümünde karar ve döngü yapıları anlatılacaktır. Karar yapılarında; program içerisinde yer alan koşullar denetlenir ve çıkan sonuçlara göre program, ilgili kod parçacıklarına yönlendirilir. Döngü yapılarında ise; program içerisinde döngüler kurularak, uygulamanın belirlenen bir işlemi otomatik olarak tekrar etmesi sağlanır.

Ünitenin ikinci bölümünde ise Nesne tabanlı programlama yöntemi anlatılacaktır. Nesne tabanlı programlama tekniğiyle bir animasyon projesi, birçok alt bölüme ayrılabilir ve birçok programcı aynı anda bu alt bölümlerde, çalışarak projeyi daha kısa sürede tamamlayabilir. Bu ünite de özellikle nesne tabanlı programlamada kullanılan Paket ve Sınıf tanımlama yöntemleri üzerinde durulacaktır.

KARAR YAPILARI

Program akışını verilen koşullara göre denetleyen ve çıkan sonuca göre programı yönlendiren ifadelere karar yapıları denir. ActionScript 3.0 programlama dilinde “if”, “if – else”, “if- else if ”, “switch-case” olmak üzere toplam dört tane karar yapısı (akış kontrol yapısı) vardır. Bu karar yapıları aşağıda detaylı olarak incelenecektir.

if ifadesi

Programlama içerisinde kullanılan en temel karar verme yapısıdır. “if” karar ifadesi, bir koşul sağlandığı zaman (koşul ifadesi doğru ise) gerçekleştirilmesi istenen program parçacıkları çalıştırılır, değilse program “if” ifadesini atlayarak normal akışına devam eder.

“if” ifadesinin genel yapısı aşağıdaki şekilde gösterilmektedir.

```
if (koşul)
{
    // Koşul doğruysa çalışacak kodlar buraya yazılır.
}
```

“if” ifadesinin içindeki koşul doğru ise program küme parantezleri arasındaki kodları çalıştırır. “if” komutuyla birlikte verilen koşul ifadesi, genellikle ilişkisel operatörler yardımıyla kullanılır. Önceki ünitelerde ayrıntılı olarak anlatılan bu operatörler Tablo 12.1’de kısaca gösterilmiştir.



“*if (Koşul)*” kontrol deyimi, bir koşulu test etmenizi, sağlar. Koşul yerine isenirse (true/flase) gibi boolean değerler de yazılabilir.

Tablo 12.1 Karar ifadelerinde kullanılan ilişkisel operatörler

(A==B)	Eşittir	A değeri B değerine eşit midir? (Eğer cevap evet ise koşul doğru (true) değerini alır. Aksi durumda koşul yanlış (false) değerini alır.)
(A!=B)	Eşit değildir	A değeri B değerine eşit değildir? (Eğer cevap evet ise koşul doğru (true) değerini alır. Aksi durumda koşul yanlış (false) değerini alır.)
(A<B)	Küçüktür	A değeri B değerinden küçük müdür? (Eğer cevap evet ise koşul doğru (true) değerini alır. Aksi durumda koşul yanlış (false) değerini alır.)
(A>B)	Büyüktür	A değeri B değerinden büyük müdür? (Eğer cevap evet ise koşul doğru (true) değerini alır. Aksi durumda koşul yanlış (false) değerini alır.)
(A<=B)	Küçüktür veya eşittir	A değeri B değerinden küçük veya eşit midir? (Eğer cevap evet ise koşul doğru (true) değerini alır. Aksi durumda koşul yanlış (false) değerini alır.)
(A>=B)	Büyüktür veya eşittir	A değeri B değerinden büyük veya eşit midir? (Eğer cevap evet ise koşul doğru (true) değerini alır. Aksi durumda koşul yanlış (false) değerini alır.)

Uygulama 1: Geçme notunun 50 olduğu animasyon tasarım dersinde bir öğrenci sınavlardan 65 not ortalaması almış olsun. Bu not ortalamasına karşılık geçti mesajını veren programı “*if ifadesi*” ile yazalım.

- Şekil 12.1’de gösterilen kodları eylem paneline yazınız.
- Eylem paneline yazılan kodları, “Ctrl+Enter” komutu ile test ediniz. Test sonucu oluşan çıktı ekranı Şekil 12.2’de gösterilmektedir.

```
Katman 1:1
1  var notOrtalama:uint=65;
2  var gecmeNotu:uint=50;
3  if(notOrtalama >=gecmeNotu)
4  {
5      trace("Geçti");
6  }
```

Şekil 12.1 if koşul ifadesi

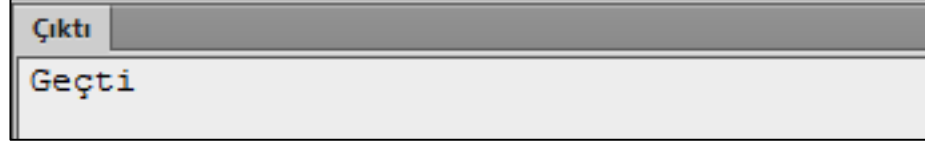
Bu uygulama için;

1. satırda : “notOrtalama” isminde “uint” veri türünde bir değişken tanımlanmış ve içerisine öğrencinin almış olduğu not ortalamasını gösteren 65 değeri atanmıştır.

2. satırda : “gecmeNotu” isminde bir kontrol değişkeni tanımlanmış ve içerisine geçme notu olan 50 değeri atanmıştır.

3. satırda : “if” ifadesiyle “notOrtalama” değerinin “gecmeNotu” değerinden büyük veya eşit olup olmadığını kontrol eden denetim yapısı oluşturulmuştur.

4-5-6. satırlarda: “*notOrtama*” değeri “*gecmeNotu*” değerinden büyük olduğu için koşul ifadesi doğru (true) değerini üretmiş ve çıktı ekranına “Geçti” mesajı yazılmıştır.



Şekil 12.2 Kod çıktı ekranı

Bu örnekte öğrencinin not ortalaması 50’den küçük olsaydı koşul sağlanmadığı için (Koşul →false değeri üretecektir) küme parantez içerisindeki komutlar çalıştırılmayacak ve ekrana “Geçti” gibi herhangi bir mesaj yazılmayacaktır.

if-else ifadesi

“ if ” ifadesinde koşul sağlandığı zaman küme parantezi içindeki kodlar çalıştırılır, fakat koşul sağlanmadığı durumda hangi işlemin yapılacağı tanımlanmaz. Şayet koşulun sağlanmadığı durumda başka bir işlem yapılması isteniyorsa “ if-else ” yapısı tercih edilebilir. “ if-else ” ifadesi koşulun doğru ya da yanlış olmasına göre her iki durumda kontrol edilerek gereken kod parçacıkları çalıştırılır.

“ if –else ” ifadesinin genel yapısı aşağıdaki şekilde gösterilmektedir.

```
if ( Kosul )
{
    // Koşul sağlanırsa çalışacak kodlar buraya yazılır.
}
else
{
    // Koşul sağlanmazsa çalışacak kodlar buraya yazılır.
}
```

Uygulama2: Bu örnek uygulamada uygulama1’den farklı olarak öğrencinin sınavlardan aldığı not ortalaması 50’ den küçük yazıldığı durumlarda öğrencinin kaldığını, 50’ den büyük yazıldığı durumlarda ise geçtiği mesajını veren bir programı “if... else” ifadesi ile yazalım.

- Yeni bir çalışma sahnesi açın. F9 fonksiyon tuşuna basarak Eylem panelini açın ve aşağıdaki kodları bu panele yazın. İlk olarak programa notOrtalamasını 25 olarak girelim.
- Eylem panlinde yazılan kodları, “Ctrl+Enter” komutu yardımıyla test ediniz. Test sonucu oluşan Çıktı ekranı Şekil 12.3’te gösterilmektedir.

```

Katman 1:1
1  var notOrtalama:uint=25;
2  var gecmeNotu:uint=50;
3  if(notOrtalama >=gecmeNotu)
4  {
5      trace("Geçti");
6  }
7  else
8  {
9      trace("Kaldı");
10 }

```

Bu uygulamada;

1. satırda :“notOrtalama” şeklinde “uint” veri türünde bir değişken tanımlanmış ve içerisine geçme notu olarak 25 değeri atanmıştır.

2. satırda :“gecmeNotu” isminde bir kontrol değişkeni tanımlanarak 50 değeri atanmıştır.

3.satırda : “if” ifadesiyle “notOrtalama” değerinin “gecmeNotu” ‘değerinden büyük veya eşit olup olmadığı kontrol eden denetim yapısı oluşturulur.

4-5-6. satırlarda: Koşul doğru olmadığı için alana yazılan küme parantezi arasındaki kodlar çalıştırılmayacaktır.

7-8-9-10. satırlarda: Koşul doğru olmadığı için 7. satırdaki “else” komut yapısı çalışarak 8-10.satırlar arasındaki kod parçacıklarını çalıştıracaktır. Bu sebeple Çıktı ekranında “Kaldı” mesajı verilecektir.

```

Çıktı
Kaldı

```

Şekil 12.3 Kod çıktı ekranı

Bu uygulama için Eylem paneline yazılan kodları tekrar açarak birinci satırdaki “notOrtalama”sı değerini 75 olarak değiştirin. Değiştirilen satır aşağıda gösterilmiştir.

```

1  var notortalama:uint=75;

```

Bu yeni durumda diğer satırlardaki kodlar aynı kalacaktır;
Bu yeni durumda program çalıştırıldığında; not otalaması geçme notundan

büyük olduğu için koşul doğru değerini alır ve if ifadesinin hemen altında bulunan küme parantezleri arasındaki satırları çalıştırılarak, çıktı ekranına “Geçti” mesajı verilir.



Bir Karar ifadelerinde koşulları her zaman parantez içerisinde kullanın.

if-else if- else ifadesi

“ if-else if- else ” kontrol ifadesi koşul sayısının birden fazla olduğu durumlarda kullanılır. Bu kontrol ifadesinin genel yapısı aşağıdaki şekilde gösterilmektedir.

```
if ( Kosul_1 )
{
    // Koşul sağlanırsa çalışacak kodlar buraya yazılır.
}
else if ( Kosul_2 )
{
    // Koşul sağlanmazsa çalışacak kodlar buraya yazılır.
}
.
.
.
else if ( Kosul_N)
{
    // Koşul sağlanmazsa çalışacak kodlar buraya yazılır.
}
else
{
    // Diğer tüm koşullar sağlanmazsa çalışacak kodlar buraya yazılır.
}
```

Uygulam3 : Bu örnek uygulamada kod içerisinde verilecek gün isminin (Pazartesi,Salı,Çarşamba.. vb.) haftanın kaçınıcı gününe denk geldiğini hesaplayan ve hesaplanan sonucu Çıktı ekranında gösteren bir uygulama yazalım.

- Yeni bir çalışma sahnesi açın. F9 fonksiyon tuşuna basarak Eylem panelini açın ve aşağıdaki kodları yazınız. Kodlar yazılırken ilk olarak *string* veri türünde tanımlanan “*gun*” değişkenine “Salı” değerini girin.
- Eylem panlinde yazılan kodları, “Ctrl+Enter” komutu yardımıyla test ediniz. Test sonucu oluşacak Çıktı ekranı Şekil 12.4’te gösterilmektedir.

```

1  var gun:String="Salı";
2  if (gun == "Pazartesi")
3  {
4      trace("Girilen değer haftanın birinci günüdür. ");
5  }
6  else if (gun == "Salı")
7  {
8      trace("Girilen değer haftanın ikinci günüdür. ");
9  }
10 else if (gun== "Çarşamba")
11 {
12     trace("Girilen değer haftanın üçüncü günüdür. ");
13 }
14 else
15 {
16     trace("Girilen değer programda tanımlı değil. ")
17 }

```

Bu uygulamada;

1. satır : “gun” isminde “string” veri türünde bir değişken tanımlanmış ve içerisine gün olarak “Salı” değeri atanmıştır.

2,6,10. satırlarda: “gun” değişkeninin Pazartesi, Salı ve Çarşamba günlerinden hangisine eşit olduğu sorgulanmaktadır.

14. satır : Bu satırda ise şayet “gun” değeri Pazartesi, Salı ve Çarşamba’dan farklı bir değer girilmiş ise programın yönlendirilecek satırı göstermektedir.

Bu programın başında “gun” değişkeni olarak Salı girildiğinden dolayı 6. satırdaki koşul doğru değerini verecek (2,10,14. satırlardaki koşullar doğru olmayacaktır.) ve program çıktısı Şekil 12.4’ teki gibi olacaktır.

Çıktı

```
Girilen değer haftanın ikinci günüdür.
```

Şekil 12.4. Kod çıktı ekranı

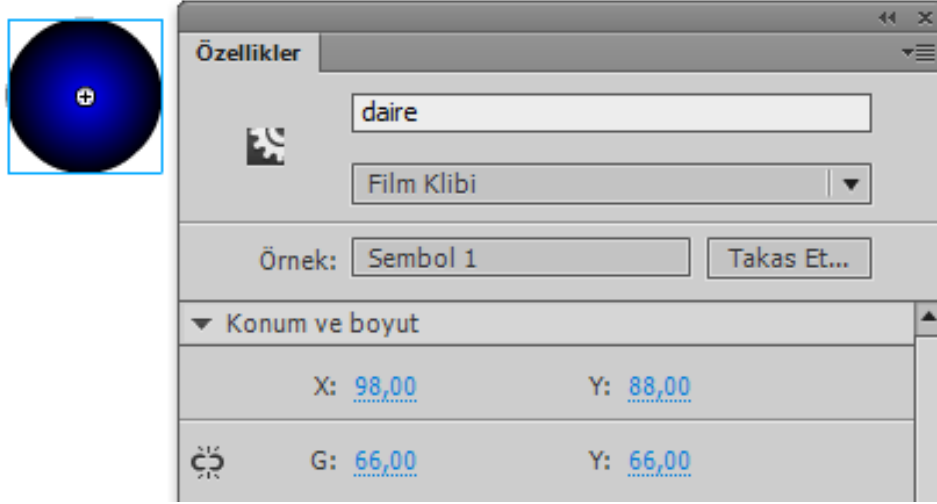
Bu uygulamadaki Eylem panelini tekrar açarak, sadece birinci satırdaki “gun” değişkenine aşağıda gösterildiği gibi “Cumartesi” değerini girin.

```
1  var gun:String="Cumartesi";
```

Yeni durumda “gun” değişkeni olarak Cumartesi girildiğinden dolayı 2,6,10 satırdaki koşullar yanlış değerini vereceğinden program otomatik olarak 14. satırın hemen altındaki küme parantezlerine yönlendirilerek buradaki satırlar çalıştırılacaktır. Bu durumda ekran çıktısına “Girilen değer programda tanımlı değil.” mesajını verilecektir. Bu bölüme kadar anlatılan uygulamalar kontrol yapılarının daha iyi anlaşılmasına yönelik temel uygulamalardır. Aşağıda karar ifadelerinin çizim nesneleri ile birlikte nasıl kullanılacağına yönelik bir tasarım uygulaması yapılacaktır.

Uygulama4:,. Bu uygulamada bir dairenin, farenin koordinatlarına göre nasıl hareket ettirilebileceği anlatılacaktır. Bu uygulama için;

- Yeni bir dosya açın. Sahne büyüklüğünü 800x600 px olarak ayarlayın.
- Çalışma alanı içerisine Şekil 12.5'te gösterilen bir daire şeklini çiziniz.
- Sahneye çizdiğiniz şekli film klibi hâline getirin ve örnek ismi olarak da daire ismini verin.



Şekil 12.5 Daire örnek ismi

- Sahneye ait Eylem paneline aşağıdaki kodları ekleyin.

```

1  stage.addEventListener(Event.ENTER_FRAME,kac);
2  function kac(event:Event):void
3  {
4      if (mouseX>100)
5      {daire.x-=3;}
6      if (mouseX<100)
7      {daire.x+=3;}
8      if (mouseY<100)
9      {daire.y+=3;}
10     if (mouseY>100)
11     {daire.y-=3;}
12 }

```

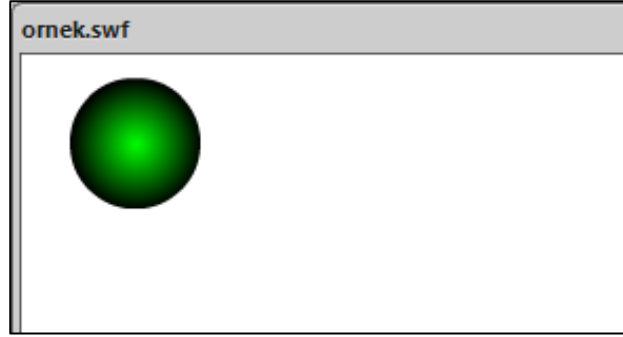
Bu uygulamada;

1. Satır: Sahneyi izleyen ve *kac* fonksiyonunu tetikleyen bir olay dinleyicisi eklenmiştir.

2-12. Satırlar: “*kac*” fonksiyonu tanımlanmıştır.

Adobe Flash içinde tanımlı olan fonksiyonlardan mouseX ve mouseY fonksiyonları (metotları) sahne üzerinde farenin yerini koordinat olarak tespit eder ve **if** kontrol

deyimi de farenin sahnenin sağ, sol, yukarı ya da aşağı taraflarda olup olmaması şartına bakar. Fonksiyon çalıştığında dört karar yapısı birlikte çalışır ve şartlara uyan **if** komutları altındaki kod parçacıklarını çalıştırır. Şekil 12.6'da gösterilen swf dosyasındaki ActionScript kodlar, daire.x ve daire.y değerleri yardımıyla dairenin sahnedeki koordinatlarını değiştirerek şartlara göre nesnenin fare hareketinin tersi yönünde hareket ettirilmesini sağlar ve bu sayede daire fareden kaçır ve fare ile daireyi tıklamak mümkün olmaz.



Şekil 12.6 swf dosyasının test ekranı

Switch- Case ifadesi

Bu ifade "**if - else if - else**" yapısında olduğu gibi birden fazla koşul için gerekli kontrolü yapar. Fakat şartın boolean(**true ya da false**) bir ifade döndürmesine bakmaz, değer kısmına yazılan ile kontrol ifadesinin aynı olup olmamasının kontrolünü sağlar. Kullanımı ve yapısı çok basittir. İstenilen sayı kadar kontrol ifadesi eklenebilir. Her **case** kontrol şartı bir **break** ifadesiyle sonlandırılmalıdır.

" **switch – case**" ifadesinin genel yapısı aşağıdaki şekilde gösterilmektedir.



Switch deyimi, bir Boolean değerinin koşulunu test etmek yerine, bir ifade olarak değerlendirilir ve hangi kod bloğunun çalıştırılacağını belirlemek için bu ifadeyi kullanır.

```
switch ( değer )
{
    case kontrol ifadesi_1:
        //değer ifadesi kontrol ifadesi_1' e eşitse çalışacak kodlar buraya yazılır.
        break;
    case kontrol ifadesi_2:
        // değer ifadesi kontrol ifadesi_2' e eşitse çalışacak kodlar buraya yazılır.
        break;
    .
    .
    case kontrol ifadesi_N:
        // değer ifadesi kontrol ifadesi_N' e eşitse çalışacak kodlar buraya yazılır.
        break;
    default:
        // değer hiçbir ifadeye eşit değilse çalışacak kodlar buraya yazılır.
```

Uygulama5: Uygulama3'te " if-else if- else " ifadesi ile yapılan işlemleri bu uygulamada "switch-case" ifadesi kullanılarak tekrar yapılacaktır.

- Yeni bir çalışma sahnesi açın. F9 fonksiyon tuşuna basarak Eylem panelini açın ve aşağıdaki kodları bu panele yazınız. Kodlar yazılırken ilk olarak *string* veri türünde tanımlanan *gun* değişkenine "Pazartesi" değerinin girin.
- Eylem panelindeki yazılan kodları, "Ctrl+Enter" komutu yardımıyla test ediniz. Test sonucu oluşan Çıktı ekranı Şekil 12.7'de gösterilmektedir.

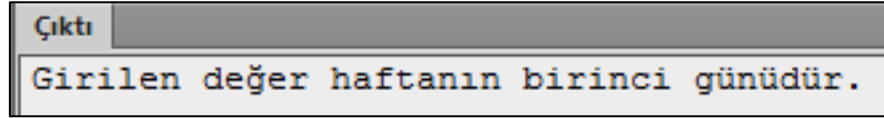
```
1  var gun:String="Pazartesi";
2  switch (gun)
3  {
4      case "Pazartesi":
5          trace("Girilen değer haftanın birinci günüdür. ");
6          break;
7      case "Salı":
8          trace("Girilen değer haftanın ikinci günüdür. ");
9          break;
10     case "Çarşamba":
11         trace("Girilen değer haftanın üçüncü günüdür. ");
12         break;
13     default:
14         trace("Girilen değer programda tanımlı değil. ")
15         break;
16 }
```

Bu uygulamada;

1. *satırda* : “*gun*” isminde “*string*” veri türünde bir değişken tanımlanmış ve içerisine gün olarak “*Pazartesi*” değeri atanmıştır.

2. *satırda* : “*switch*” ifadesiyle “*gun*” değerinin diğer “*case*” ifadeleriyle verilen kontrol ifadelerine “*Eşit*” olma koşulunu kontrol eden bir denetim yapısı oluşturmaktadır.

Gun değişkenine yüklenen Pazartesi değeri 4. Satırdaki “*case*” de birbirine eşit olduğundan dolayı 5. satırdaki kod hemen çalıştırılarak çıktı ekranına *Şekil 12.7’ de gösterilen* mesaj yazılacaktır. Program daha sonra 6. satırı çalıştıracak ve buradaki break komutu yardımıyla program “*switch*” ifadesinden çıkacaktır.



Şekil 12.7 Kod çıktı ekranı

Bu uygulamadaki Eylem panelini tekrar açarak, sadece birinci satırdaki “*gun*” değişkenine “*Pazar*” değerini girin. Yeni durumda “*gun*” değişkene girilen değer “*switch*” ifadesinde verilen günlere uymayacağından 13. Satırdaki *default:* ifadesi çalıştırılacak ve Çıktı ekranına “Girilen değer programda tanımlı değil.” mesajı verilip program sonlanacaktır.

DÖNGÜ YAPILARI

Eylem kodlarında, bir işlem grubunu çok sayıda tekrar etme imkânı sağlayan yapılara döngü denir. Aynı işlemi defalarca tekrar yazmak gerektiğinde, döngü kurulup uygulamanın otomatik olarak bu işlemi tekrar etmesini sağlayabiliriz. ActionScript 3.0 programlama dilinde en çok kullanılan “*for*”, “*while*”, “*do..while*” döngüleri aşağıda detaylı olarak anlatılmaktadır.

For döngüsü

ActionScript içerisinde *for* döngüsü kod bloklarını, belirli bir sayıda çalıştırmak için kullanılır.

“*for*” ifadesinin genel yapısı aşağıdaki şekilde gösterilmektedir.

```
for ( Başlangıç_değeri; Koşul; Artış miktarı)
{
    // Döngü içinde çalıştırılacak kodlar
}
```



For döngüsü, belirli bir değer aralığı için bir değişkeni sürekli yenilemeye olanak sağlar.

for döngüsünün kullanımı şu şekildedir;

Başlangıç_değeri: Kontrol değişkenine başlangıç değerinin atanmasını sağlar.

Koşul: Başlangıç değeri verilen kontrol değişkenin değeri, koşul ifadesiyle karşılaştırılır. Koşul sağlandığı sürece küme parantezi arasındaki kodlar çalıştırılır.

Artış miktarı: *for* döngüsü işleminin sonunda başlangıç değeri verilen değişkenin değerini artırır ya da azaltır.

Uygulama6: Bu uygulamada “Tasarım yapmayı seviyorum” şeklindeki bir ifadeyi *for* döngüsünü kullanarak dört kere çıktı ekranında yazdıran bir program hazırlayalım.

- Yeni bir çalışma sahnesi açın. F9 fonksiyon tuşuna basarak Eylem panelini açın ve aşağıdaki kodları yazın.

- Eylem panlindeki yazılan kodları, “Ctrl+Enter” komutu yardımıyla test ediniz. Test sonucu oluşan Çıktı ekranı Şekil 12. 8’de gösterilmektedir.



for(; ;) { } şeklindeki bir tanımlama sonsuz döngü oluşturulmasını sağlar. Bu şekildeki bir kullanımda küme parantez blokları arasında mutlaka kontrol deyimleri kullanılmalıdır.

```
1   for(var i:uint=0;i<4;i++)
2   {
3       trace("Tasarım yapmayı seviyorum");
4   }
```

Bu uygulama için;

1. satır: *for* ifadesinin içindeki **başlangıç_değeri:** Kontrol değişkeni olarak “i” , veri türünde “*uint*” ve başlangıç değerinde “0” olarak tanımlanmıştır.

Koşul: “Tasarımı seviyorum” ifadesinin dört kez tekrarlanması istendiği için *i* “Küçük”(<) 4 olarak oluşturulmuştur. **Artış Miktarı:** başlangıç_değeri sıfırdan başladığı için (*i++*) **kontrol** değişkeninin değeri birer birer artırılmıştır.

2-3-4. satırlarda: koşul dört kez sağlandığı için *trace* komutu dört kez çalıştırılacak ve Çıktı ekranına dört kez “Tasarım yapmayı seviyorum” şeklinde yazdırılacaktır.

```
Çıktı
Tasarım yapmayı seviyorum
Tasarım yapmayı seviyorum
Tasarım yapmayı seviyorum
Tasarım yapmayı seviyorum
```

Şekil 12.8 Kod çıktı ekranı

• 1. satırda tanımlanan *for* ifadesinde başlangıç_değeri: “*var i:uint=1*” olarak verilseydi koşul: (*i < 5*) olarak değiştirilmesi gerekirdi. Ayrıca başlangıç_değeri: “*var i:uint=4*” olarak verilirse koşul: (*i > 5*) olarak değiştirilip, Artış miktarıda: (*i--*) ifadesiyle kontrol değişkeninin değeri birer azaltılırdı.

While döngüsü

Döngünün çalışma sayısını bilemediğimiz durumlarda bu döngü ifadesini kullanırız. if ve for döngüsünün birleşimi gibidir. Parantez içerisindeki şart doğru olduğu sürece döngü çalışır.

“*while*” ifadesinin genel yapısı aşağıdaki şekilde gösterilmektedir.

```
while ( Koşul )
{
    // Döngü içinde çalıştırılacak kodlar
}
```



While döngüsü, koşul true olduğu sürece yinelenen if deyimine benzer.

Uygulama7: Bu uygulamada “Açıköğretim Fakültesi” şeklindeki bir ifadeyi *while* döngüsünü kullanarak üç kere çıktı ekranında görüntülenmesini sağlayalım.

- Yeni bir çalışma sahnesi açın. F9 fonksiyon tuşuna basarak Eylem panelini açın ve aşağıdaki kodları yazınız.
- Eylem panlindeki yazılan kodları, “Ctrl+Enter” komutu yardımıyla test ediniz. Test sonucu oluşan Çıktı ekranı Şekil 12.9’da gösterilmektedir.

```
1  var i:uint=0
2  while(i<4)
3  {
4      trace("Açıköğretim Fakültesi")
5      i++;
6  }
```

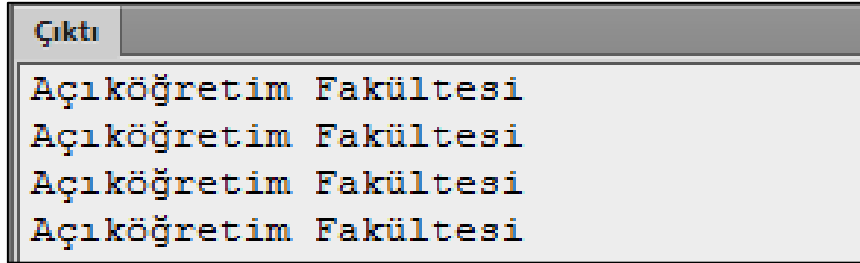
Bu uygulama için;

1. satır: *while* ifadesi için kontrol değişkeni olarak “*i*” , veri türünde “*uint*” ve başlangıç değeri de “*0*” olarak tanımlanmıştır.

2. satır: “Açıköğretim Fakültesi” ifadesinin dört kez tekrarlanması istendiğinden bir koşul ifadesi (*i < 4*) olarak oluşturulmuştur.

5. satır: kontrol değişkeninin değeri sıfırdan başladığı için *i*, değişkeninin değeri birer birer atılmıştır.

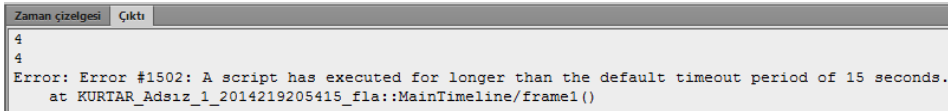
2-3-4-6. satırlarda: döngü koşulu dört kez sağlandığı için trace komutu dört kez çalıştırılacak ve Çıktı ekranına dört kez “Açıköğretim Fakültesi” yazdırılacaktır.



```
Çıktı
Açıköğretim Fakültesi
Açıköğretim Fakültesi
Açıköğretim Fakültesi
Açıköğretim Fakültesi
```

Şekil 12.9 Kod çıktı ekranı

- Yukarıdaki örnek uygulamada while ifadesinin genel kullanımında gösterilmeyen 1. satırda başlangıç_değeri ve 5. satırda ise artış miktarı tanımlanmıştır. Yukarıdaki kodda başlangıç_değeri verilir artış miktarı verilmeseydi şartın değeri (0 < 4) (*true*) şeklinde olacağından şart sağlandığı için program sonsuz bir döngüye girecekti ve döngüden çıkması için gereken kod bulunmadığı içinde döngü sonsuz defa çalışmak isteyecektir. Böyle durumlarda Adobe Flash programı 15 saniye bekler ve Şekil 12.10’da gösterilen Çıktı ekranda hata mesajı görüntüleyerek döngüyü ve programı sonlandırır.



```
Zaman çizelgesi Çıktı
4
4
Error: Error #1502: A script has executed for longer than the default timeout period of 15 seconds.
at KURTAR_Adsız_1_2014219205415 fla::MainTimeline/frame1()
```

Şekil 12.10 Sonsuz döngünün ortaya çıkardığı hata mesajı

do..while döngüsü

do-while yapısı döngüye devam etmek için gereken şartı döngünün en sonunda kontrol eder. Kontrol sonucuna göre ya döngüye devam edilir ya da döngüden çıkar ve böylece şart sağlanmasa bile döngü en az bir kez çalışmış olur.

“*do-while*” ifadesinin genel yapısı aşağıdaki şekilde gösterilmektedir.

```
do
{
// Döngü içinde çalıştırılacak kodlar
}
while ( Koşul )
```

Uygulama8: Bu uygulamada “AOF” şeklindeki bir ifadeyi *do-while* döngüsü kullanarak çıktı ekranında gösterilmesini sağlayalım.

- Yeni bir çalışma sahnesi açın. F9 fonksiyon tuşuna basarak, açılan Eylem paneline aşağıdaki kodları yazın.

- Eylem panlindeki yazılan kodları, “Ctrl+Enter” komutu yardımıyla test ediniz. Test sonucu oluşan Çıktı ekranı Şekil 12.11’de gösterilmektedir.

```

1  var i:int = 12;
2  do
3  {
4      trace("AOF");
5      i++;
6  } while (i < 12);
7

```

Bu işlem için;

1. satır : “i” adında , “int” veri türünde ve başlangıç değerinde “12” olan bir kontrol değişkeni tanımlanmıştır.

2. satır: *do* ifadesi ile döngüye başlanır.

3-4. satırlar: *trace* komutuyla çıktı ekranına bir kez “AOF” yazdırılır.

5. satır: “i” kontrol değişkeninin değeri bir artırılır.

6. satır: *while* ile(*i<12*) koşulu kontrol edilir. 13 değeri 12’den **Küçük** olmadığından şart sağlanamaz ve programdan çıkılır. Böylece şart sağlanmasa bile döngü en az bir kez çalışmış olur ve çıktı ekranına bir kez AOF yazısı yazılır.



Şekil 12.11 Kod çıktı ekranı

NESNE TABANLI PROGRAMLAMA

Adobe Flash, ActionScript 3.0 programlama diliyle beraber bir çok web uygulaması için ihtiyaç duyulan animasyon tasarım programı hâline gelmiştir. Uygulama geliştiren animatörler için flash içerisinde iki tür yazılım geliştirme ortamı bulunmaktadır.

Bunlardan birincisi yordam tabanlı diğeri ise nesne tabanlı programlama ortamlarıdır. Yordam tabanlı programlamada kodlar direkt olarak sahneye yazılıp çalıştırılmaktadır. Önceki ünitelerde ve bu ünitenin başında anlatılan uygulamalar yordam yöntemiyle yazılmıştır. Nesne tabanlı programlama yönteminde ise kodlar



ActionScript dili
klasörleri nokta
işaretini kullanarak
birbirinden ayırır.

öncelikle sınıf ve paket dosyalarına yazıldıktan sonra eylem panelinden çağrılıp çalıştırılırlar. Nesne Tabanlı programlama ortamının, yordam tabanlı programlama ortamına göre birçok avantajı vardır. Bunlardan bir kaçışunlardır;

- Geliştirilmekte olan bir uygulama, nesne tabanlı yaklaşım sayesinde modüllere ayrılabilir.
- Modül yapısı sayesinde, tasarımın grafik bölümünü animatörler hazırlarken kod bölümünü, programcılar farklı zamanlarda ve farklı ortamlarda geliştirebilirler. Daha sonra bu modüller birleştirilerek tasarım tamamlanır.
- Ayrıca yazılan her bir modül arşivlenerek, ileride başka uygulamalar içinde rahatlıkla kullanılabilir.

Nesne tabanlı programlamada, öncelikle paket (package) ve sınıf (class) tanımlaması yapılamalıdır. Aşağıda paket ve sınıf tanımlanma yöntemleri anlatılmaktadır.

Paket (Package)

ActionScript dilinde nesne tabanlı programlama ortamında yazılan kodlar genelde paketlerden oluşmaktadır. Paketlerin içinde sınıflar ve arayüzler bulunur. Uygulama içerisinde yer alan paketlerin dışında, istenirse yeni paketler de *“import”* anahtar sözcüğü kullanılarak uygulama içerisine dâhil edilebilir. Bir paket içinde tanımlanan sınıflar ve arayüzler, uygulama için gerekli olan yerlerde çağrılarak kullanılabilir.

Paket tanımlama yöntemi aşağıda gösterilmektedir.

```
package paket_ismi  
{  
    // Eylem kodları buraya yazılır.  
}
```

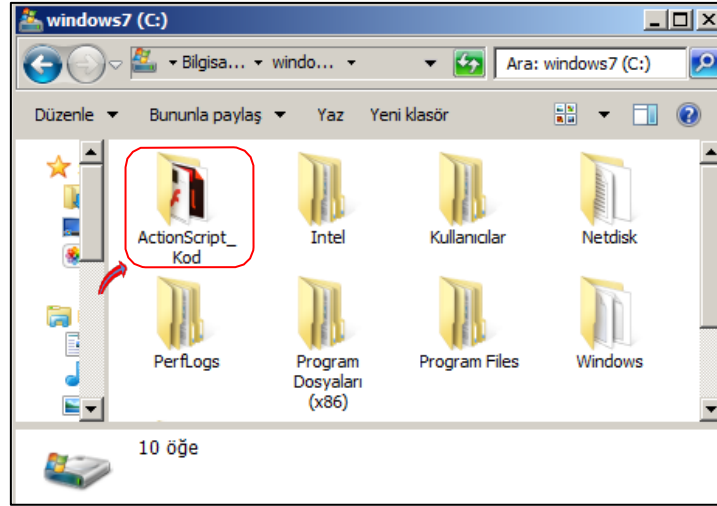


Paket içine yer alan
sınıflar, arayüzler ve
import tanımları için bir
sayı sınırlaması yoktur.

Yukarıda gösterilen tanımda, *“package”* deyiminden sonra *“paket_ismi”* verilmesi gereklidir. Paket_ismi ve oluşturulan *.as dosyasının bulunduğu klasör ismi, aynı olmalıdır.

Adobe Flash içerisinde paket tanımlama aşağıdaki gibi yapılır;

- Bilgisayarınızın c:\ sürücü üzerinde ActionScript_Kod isimli bir klasör oluşturun (Şekil 12.12).



Şekil 12.12 ActionScript_Kod isimli klasörün sistemdeki görünümü

- ActionScript_Kod isimli klasörün içine Şekil 12.13 'te gösterildiği gibi Kod isimli bir başka klasör daha oluşturun.

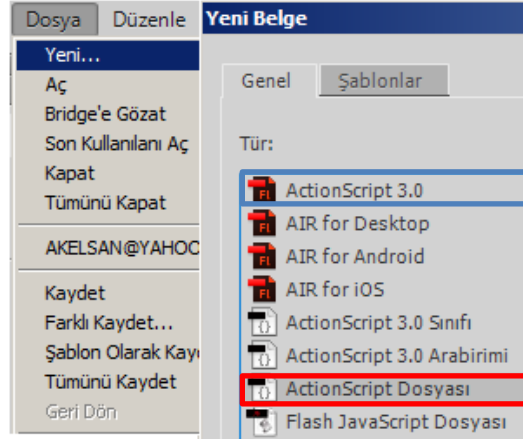


Şekil 12.13 Proje klasörünün içindeki Kod isimli klasör

- Adobe Flash programında Şekil 12.14'te mavi çerçeve ile gösterilen bölümlerden *Yeni Oluştur>ActionScript 3.0* ile yeni bir çalışma sayfası oluşturarak ActionScript_Kod klasörünün içerisine *tasarim fla* ismiyle kaydedin.
- Şekil 12.14'te kırmızı çerçeve ile gösterilen *Dosya > Yeni > ActionScript Dosyası* bölümünü seçerek, ActionScript dosyasını *paket.as* şeklinde kod klasörü içerisine kaydedin.

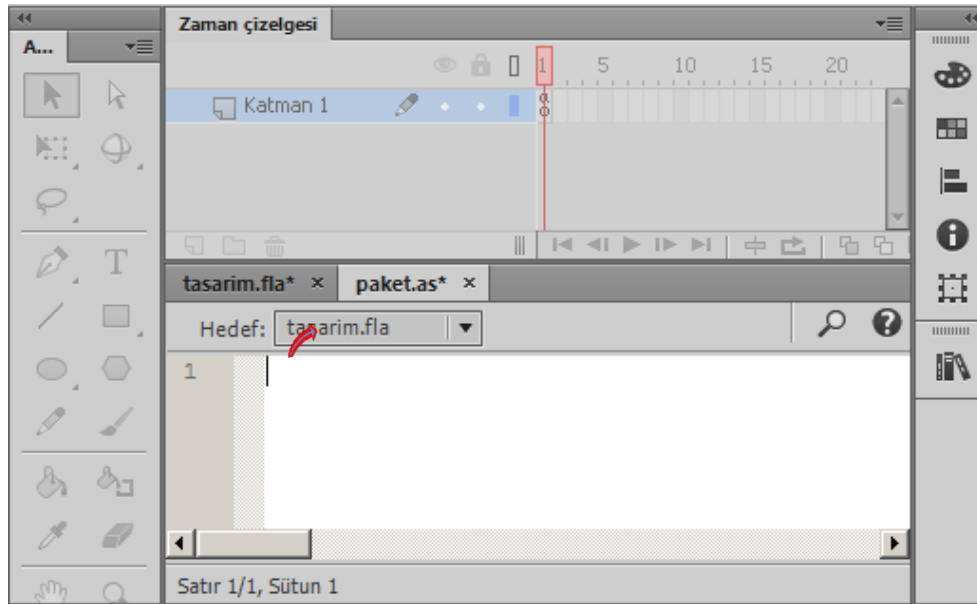


Konuda anlatılan (kod) isimli klasörün içinde (ornek) isimli bir klasör daha olsaydı. Adres yazarken “kod.ornek” şeklinde yazılması gerekirdi.



Şekil 12.14 Yeni dosyası açma penceresi

- Adobe Flash içerisinde yukarıdaki adımlar ile oluşturulan “*tasarim fla*” dosyası, “*paket as*” komut dosyasının hedef kısmında Şekil 12.15’de *kırmızı okla işaretlenen* konumda görülmesi gerekir.

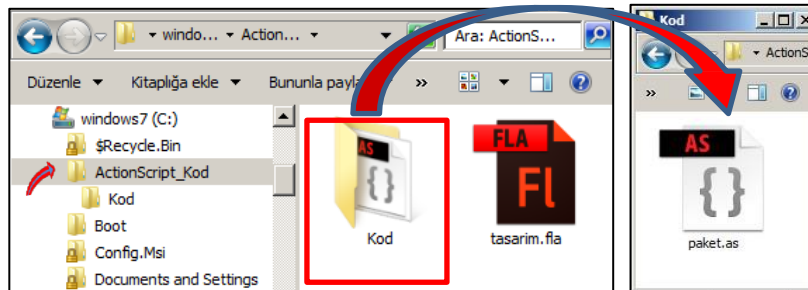


Şekil 12.5 Çalışma dosyalarının program içerisindeki görünümü

- Kod klasörünün içinde yer alan “*Paket.as*” dosyasının bilgisayar içindeki yerleşimi Şekil 12.16’da gösterilmiştir.

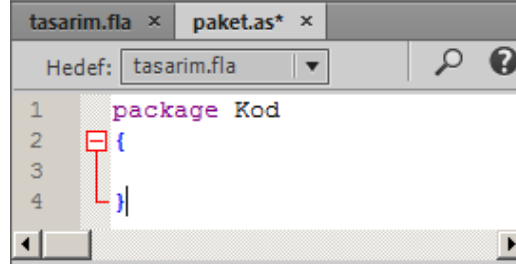


Bir paket içerisine birden fazla sınıf tanımlı yapılabilir.



Şekil 12.16 C:\ sürücüsü içindeki yer alan ActionScript klasörü ve Kod klasörünün içeriği

- Paket oluşturma işlemi, sınıfların klasördeki yerini belirleyen adres tanımlarıdır. Kaydedilen **.as* uzantılı dosyanın **.fla* dosyasına göre nerede olduğunu gösterir. Şekil 12.17’de gösterilen kod, paket.as dosyası içerisine yazılmalıdır.



Şekil 12.17 Kod isminde paket tanımlaması

- Yukarıdaki Şekil 12.17’de gösterilen kodun (Eylem paneli yerine) paket.as dosyasının içerisine yazılmasıyla paket tanımlama işlemi tamamlanmış olur.



Bir sınıftaki değişkenler o sınıfın niteliklerini, metotlar ise o sınıfın davranışlarını belirler.

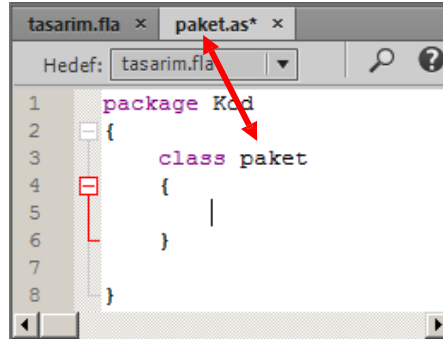
Sınıf (Class)

Nesne yönelimli programlama dillerinde bir nesnelerin özelliklerini, davranışlarını ve başlangıç durumlarını parametre olarak alarak, bu nesneye ve özelliklerine sonradan erişme imkânı sağlayan yapılara sınıf denir. ActionScript’ te kullanılan sınıf tanımlama formatı aşağıda gösterilmiştir.

```
class sınıf_ismi
{
    // Yapıcı fonksiyonlar ve metotlar buraya yazılır.
}
```

Yukarıda verilen tanımda “*class*” deyiminden sonra “*sınıf_ismi*” yazılmalıdır.

Şekil 12.18’de gösterildiği gibi sınıf ismi ve oluşturulan **.as* dosyasının ismi, aynı olmak zorundadır. Aksi hâlde sahnelerde bu tanımlara erişirken yetki sınırları problemi oluşturur. Bu durum özellikler bölümünde ayrıntılı anlatılmıştır.



Şekil 12.18 paket class’ının yazılması



Fonksiyonlara başka paket ve sınıflardan erişim sağlamak amacıyla genellikle public türünde bir yetkilendirme yapılır.

Adobe Flash içerisinde bir sınıf tanımlamak için:

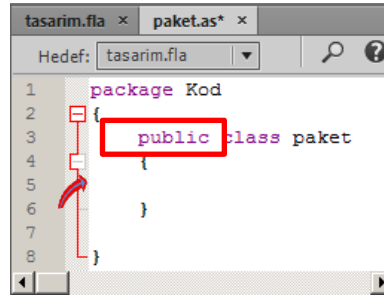
Bir sınıf yapısı temelde şu üç bölümden oluşur:

- Özellikler (Properties)
- Yapılandırıcı Fonksiyon (Constructor Method)
- Metotlar (Methods)

Özellikler (Properties): Bir sınıf tanımlı içinde, sınıf elemanlarına erişimi belirleyen niteliklerdir. ActionScript 3.0'da sınıflara ve sınıf elemanlarına erişimi belirleyen dört önemli özellik vardır. Bunlar tablo 12.2'de gösterilen "*internal*", "*private*", "*protected*", "*public*" özellikleridir. Adobe Flash içerisinde genelde "*private*" ve "*public*" nitelikleri yaygın kullanılmaktadır.

Bir sınıf "*private*" ise; o metoda sadece o sınıf içinden erişilebilir. Eğer bir sınıf "*public*" ise; o metoda istenilen herhangi bir yerden erişilebilir.

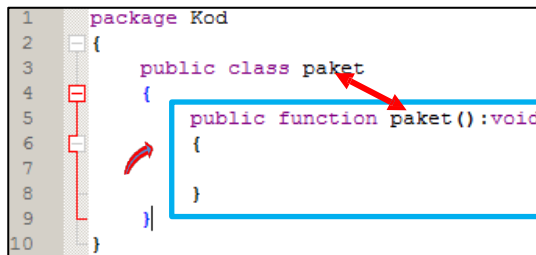
Bir sınıf tanımında hiçbir yetkilendirme yapılmamışsa o sınıf "*internal*" özelliğine sahip olur. Adobe Flash içerisinde bir sınıfı yetkilendirme işlemi Şekil 12.19'da gösterilmiştir.



Şekil 12.19 Public tanımlamasının yapılması

Yapılandırıcı Fonksiyon (Constructor Method): Bir sınıf oluşturulurken sınıf içinde otomatik olarak oluşturulan özel bir fonksiyondur. Yapılandırıcı fonksiyon, sınıf çalıştığı anda devreye girer. Yapılandırıcı fonksiyon ismi, sınıf ismi ile aynı olmak zorundadır. Bu fonksiyonun nitelik özelliği ise; "*public*" olmalıdır.

Adobe Flash içerisinde tanımlanan bir yapılandırıcı fonksiyon Şekil 12.20 'de mavi çerçeve içinde gösterilmiştir.



Şekil 12.20. Yapılandırıcı fonksiyon hazırlanması

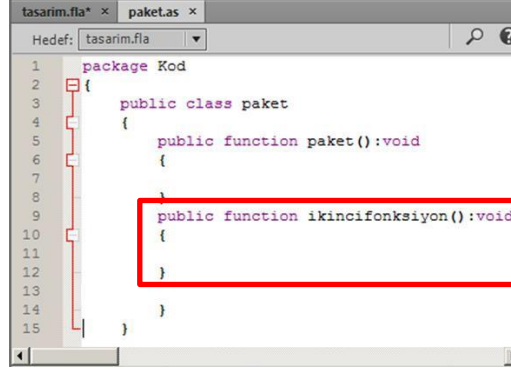
Metotlar (Methods): Bir sınıf yapısında var olan bölümlerden bir diğeri metotlardır. Metotlar, belirli davranışları yerine getirmek için yapıcı fonksiyondan



Sınıflar içinde mutlaka bir yapılandırıcı fonksiyon bulunmalıdır.

sonra tanımlanan ikincil fonksiyonlardır. Metotları tanımlarken kullanılan anahtar sözcük; fonksiyonlarda olduğu gibi *"function"* anahtar sözcüğüdür.

Adobe Flash içerisinde tanımlanan bir metod Şekil 12.21’de kırmızı çerçeve içinde gösterilmiştir.



Şekil 12.21 Kod dosyasının hazırlanması

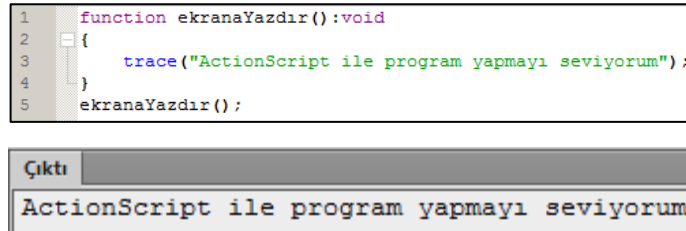
Sınıflar ve Paketlerin kullanılması

Paket ve Sınıflar tanımlandıktan sonra bir uygulama içerisinde kolayca çağrılarak kullanılabilir. Bu çağırma işlemi aşağıdaki uygulama ile gösterelim.

Uygulama 9: Çalıştırıldığında Çıktı ekranına *"ActionScript ile program yapmayı seviyorum"* mesajı veren programı yordam tabanlı ve nesne tabanlı olmak her iki yöntemle yazalım

1.Yöntem (Yordam tabanlı programlama):

- Yeni bir çalışma sahnesi açın. F9 fonksiyon tuşuna basarak açılan Eylem paneline aşağıdaki kodları yazınız.
- Eylem panelindeki yazılan kodları, "Ctrl+Enter" komutu yardımıyla test ediniz. Test sonucu oluşan Çıktı ekranı Şekil 12.22’de gösterilmektedir.



Şekil 12.22 Kod çıktı ekranı

2.Yöntem (Nesne tabanlı programlama):

Bu yöntemde Şekil 12.21’de gösterilen önceden tanımlanmış olduğumuz paket, sınıf ve fonksiyonlar bu uygulamada çağrılarak kullanılacaktır.

- Şekil 12.21’de gösterilen, içerisinde sadece tanımların yer aldığı *"paket.as"* dosyasını açın. Bu dosyanın içerisinde yer alan 10 ve 12 satırları aşağıdaki kırmızı ve mavi çerçeveler içinde Şekil 12.23’te gösterilen kodlarla yer değiştiriniz.



Oluşturulan bir sınıf örneğinin, sınıf içindeki özellik veya metotlarına ulaşabilmek için (.) nokta operatörü kullanılır.

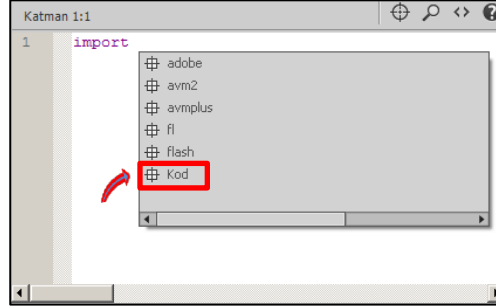
```

1 package Kod
2 {
3     public class paket
4     {
5         public function paket():void
6         {
7             //yapılandırıcı fonksiyon
8         }
9
10        public function metod():void
11        {
12            trace("ActionScript ile program yapmayı seviyorum");
13        }
14    }

```

Şekil 12.23 Yeniden yazılan Kod paketinin içeriği

- Ünitenin başında “*tasarım fla*” ismiyle kaydettiğiniz ve üzerinde henüz hiçbir tasarımın yer almadığı flash dosyasını açın. Bu dosya üzerindeyken Eylemler panelini F9 tuşu ile açın. Adobe Flash programı içerisinde birçok sınıf bulunmaktadır. Bunların bazıları otomatik olarak çalışma sahnesine dâhil olur, kendi yazdığınız sınıfların ise uygulama geliştirme aşamasında import yöntemi ile uygulamaya sonradan dâhil edilmesi gerekir.
- **Import** komutunu yazıp **boşluk** bırakıldığında ActionScript dosyası içindeki paket ismi yani Kod otomatik olarak çıkacaktır. Şekil 12.24’te gösterilen bölümden Kod isimli sınıf seçildikten sonra karşınıza Kod klasörünün içindeki paket gelecektir.



Şekil 12.24 Sınıf seçme penceresi

```

1 import Kod.

```

paket - Kod

- Yukarıda kırmızı okla gösterilen Panel, programcıya önceden tanımlanmış sınıf ve paketleri göstererek bunlar içerisinde kolayca seçim yapmasını sağlar. Aşağıda verilen birinci satırdaki kodla paket çağırılma işlemi tamamlanmış olur.

```

1 import Kod.paket;

```



Sınıfları çalışılan sahneye dâhil etmek için import komutu kullanılır.

Sınıflar örnek oluşturularak kullanılabilir. Örnek oluşturma işlemi, hazır olan sınıfın bir kopyasının alınıp çalışma sahnesinde kullanılmasıdır. Orijinal sınıf aynen

kalır. Tüm işlemler örnek üzerinden yapılır. Örnek çıkartılan nesne orijinali ile aynı özellik ve metotlara sahiptir.

Sınıflardan örnek sınıf oluşturma işlemi için;

“ **var kopyaSınıf: birinci= new birinci();** ” şeklinde bir Kod yazılır. Bu Kod yardımıyla *birinci* isimli sınıftan *kopyasınıf*, isimli bir başka sınıf örneği oluşturulmuştur.

- Aşağıda 2. satırda verilen kodu eylem paneline ekleyiniz. Bu Kod ile önceden tanımladığımız *paket* isimli sınıftan kopyaSınıf adında sınıf örneği çıkarılmaktadır. Bu Kod ile oluşturulan kopyaSınıf nesnesi artık paket sınıfının tüm özelliklerine sahip durumdadır.

```
1 import Kod.paket;
2 var kopyaSınıf:paket=new paket();
```

- Şekil 12.23’ te Kod içeriği gösterilen paket sınıfına ait metodları kullananmaya yarayan aşağıdaki kodları panele ekleyiniz.

```
1 import Kod.paket;
2 var kopyaSınıf:paket=new paket();
3 trace(kopyaSınıf.metod());
```

- Yukarıda son şekli gösterilen “*tasarim fla*” dosyasını “**Ctrl+Enter**” komutu yardımıyla test ediniz. Test sonucu oluşan Çıktı ekranı Şekil 12.25’te gösterilmektedir.

Çıktı
ActionScript ile program yapmayı seviyorum

Şekil 12.25 Nesne tabanlı olarak yazılan uygulamanın çıktı ekranı

Yukarıdaki uygulama yordam ve nesne tabanlı her iki programda çalıştırıldığında aynı ekran çıktısı elde edilir. Program yapılarına bakıldığında yordam tabanlı tekniğin, nesne tabanlı programlama tekniğine göre daha sade ve kullanışlı olduğu kanısına varılabilir. Bu kanı birkaç satırdan oluşan uygulamalar için doğru olarak gözükebilir binlerce satırdan oluşan büyük projeler için bu kanı doğru değildir. Yordam için yazılan kodlar, genelde sadece yazıldığı animasyon tasarımında geçerlidir. Nesne tabanlı yapı için yazılan paket, sınıf ve fonksiyonlar istenen her tasarım projelerinde yeniden çağrılarak çalıştırılabilir.



Özet

- if karar ifadesi, bir koşul sağlandığı zaman yapılacak işlemleri kapsar. Kontrol edilecek koşul ifadesinin sonucu doğru (true) ise gerçekleştirilmesi istenen program parçacıkları çalıştırılır.
- if..else koşul deyimi, bir koşulun doğru yada yanlış olmasına göre her iki durum içinde kontrol yaparak gereken kod parçacıklarını çalıştırır.
- if..else if deyimi, Birden fazla şartın kontrol edilmesi gereken yerlerde kullanılır. Hangi şart true değeri üretirse o kod bloğu arasındaki kodlar çalıştırılır.
- switch .. case deyimi, bir Boolean değerinin koşulunu test etmek yerine, bir ifade olarak değerlendirilir ve hangi kod bloğunun çalıştırılacağını belirlemek için bu ifadeyi kullanır. Her case.. bloğu içerisinde mutlaka break komutu kullanılmak zorundadır. Tanımlanamayan değer aralıklarında default tanımlaması kullanılır.
- for döngüsü, belirli bir değer aralığı için bir değişkenin değerinin sürekli yenilenmesinde kullanılır.
- while döngü deyimi, döngünün çalışma sayısını bilmediğimiz durumlarda kullanılır. if ve for döngüsünün birleşimi gibidir. Parantez içerisindeki şart doğru olduğu sürece döngü çalışacaktır.
- do –while döngüsünün diğerlerinden farkı önce döngü işlemleri yapılır daha sonra şart en son adımda kontrol edilir
- Adobe Flash ActionScript 3.0 dilini kullanarak uygulama geliştiren animatörler için iki tür yazılım geliştirme tekniği bulunmaktadır. Bunlardan birincisi yordam tabanlı yani kodları eylem paneli vasıtasıyla sahneye yazıp çalıştırma, ikincisi ise nesne tabanlı programlama yapmaktır. Nesne tabanlı yazılım geliştirme basit olarak uygulama için gerekli fonksiyonlar ve kodların dışarıdaki bir kod sayfasında yazıp daha sonra bunları yazılıma dâhil edilip kullanmak olarak düşünülebilir.
- Nesne tabanlı programlamayla geliştirilen animasyon tasarımları içerisinde bir grafik nesnelerinin yer aldığı *.fla dosyası bir de içerisinde ActionScript kodların bulunduğu *.as dosyası bulunmaktadır.
- ActionScript dosyası (*.as) içerisinde (package) komutuyla açılan ({}) bloğa (class) komutuyla istenildiği kadar sınıf tanımlanabilir. Bu sınıfların çalıştırılmasıyla (function) komutuyla tanımlanan yapılandırıcı (tamamlayıcı) fonksiyonlarda otomatik olarak çalışmaktadırlar.
- Tasarım dosyası(*.fla) içerisinde eylem panelinde paket, class isimleri kullanılarak bu kodlar çalışma içerisinde dâhil edilir. Bu işlem sırasında Sınıf isimleriyle, ActionScript dosyası ve fonksiyon isimleri ile sınıf isimleri aynı olmak zorundadır.

DEĞERLENDİRME SORULARI

1. Aşağıdakilerden hangisi for için sonsuz döngüyü ifade eder?
 - a) for (; ;){} - b) for (; i<0;){} - c) for (i=0; ;){} - d) for (int i=0 : :){} - e) for (, ,){} -
2. Aşağıdakilerden hangisi istenirse, if – else yerine kullanılabilir?
 - a) for
 - b) while
 - c) if
 - d) trace
 - e) switch- case
3. Switch ..Case yapısı içerisinde kullanılan “break” komutu ne işe yarar?
 - a) Mevcut dönüğünün sonlandırılmasını sağlar
 - b) Komutları devam ettirir
 - c) Değişkenin değerini ekrana yazdırır
 - d) Programa hata verir
 - e) Kontrol şartlarını denetler
4. Bir işlemi istediğimiz sayıda tekrar ettirmek için kullanılan kodlara ne denir?
 - a) Sınıf
 - b) Paket
 - c) Tanımlayıcı
 - d) Karar
 - e) Döngü

```
for (var i:int=10; i>=1; i--)
{
  trace(i);
}
```

5. Yukarıda verilen kod içerisindeki for ifadesi çalıştırıldığında çıktı ekranında aşağıdaki seçeneklerden hangisi görüntülenir.
- a) 1,2,3,4,5,6,7,8,9 sayilari alt alta görüntülenir.
 - b) 2, 4, 6, 8,10 sayilari yan yana görüntülenir.
 - c) 10, 8, 6, 4, 2 sayilari alt alta görüntülenir.
 - d) 10,9,8,7,6,5,4,3,2,1 sayilari yan yana görüntülenir.
 - e) 10,9,8,7,6,5,4,3,2,1 sayilari alt alta görüntülenir.

6. Eğer string tipindeki “karar” değişkeninin içeriği “Adobe Flash” ise “Animasyon Tasarımı” mesajını görüntüleyen komut satırı aşağıdakilerden hangisidir?

- a) if (karar = Adobe Flash){ trace(“Animasyon Tasarımı”);}
- b) if (karar == “Adobe Flash”) { trace(“Animasyon Tasarımı”);}
- c) if (karar = “Adobe ”); trace(“Flash Animasyon Tasarımı”);
- d) if (karar= “Flash”) { trace(“Animasyon Tasarımı”);}
- e) if(karar=“Animasyon Tasarimi”){ trace (“Adobe Flash”);}

7. Aşağıdakilerden hangisi tanımlayıcı fonksiyon özelliği taşımaz?

- a) Fonksiyon ismi, sınıf ismi ile aynı olmak zorundadır.
- b) Fonksiyon sınıf çalıştığı anda devreye girer.
- c) Fonksiyonu oluşturulmazsa otomatik olarak boş bir tane tanımlanır.
- d) Fonksiyonlar public olarak yetkilendirilir.
- e) Fonksiyonlar kaynak dosya ile aynı yere kayıt edilmek zorundadır.

```
var a:int=0
while(a<5)
{
    trace(a);
    ++a;
}
```

8. Yukarıda verilen kod içerisindeki while ifadesi çalıştırıldığında çıktı ekranında aşağıdaki seçeneklerden hangisi görüntülenir.

- a) 0,1,2,3,4,5 sayilari alt alta görüntülenir.
- b) 1,2,3,4,5 sayilari yan yana görüntülenir.
- c) 0,1,2,3,4 sayilari alt alta görüntülenir.
- d) 1,2,3,4 sayilari yan yana görüntülenir.
- e) 5,4,3,2,1 sayilari alt alta görüntülenir.

9. Aşağıdakilerden hangisi bir sınıftan kopya sınıf türetilmek için kullanılır?

- a) package
- b) var
- c) import
- d) new
- e) class

10. Aşağıdakilerden hangisi ActionScript dosya uzantısı olarak kullanılır?

- a) *.cs
- b) *.as
- c) *.fla
- d) *.swf
- e) *.flv

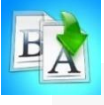
Cevap Anahtarı

1.a, 2.e, 3.a, 4.e, 5.e, 6.b, 7.a,8.c, 9.d, 10.b.

YARARLANILAN KAYNAKLAR

- [1] AKBUĞA, M. (2011). Grafik Animasyon- I Erzurum: Atatürk üniversitesi. ATAUZEM.
- [2]Adobe Flash Pro CC, Yardımcı el kitabı. <http://helpx.adobe.com/tr/support>
[Erişim tarihi: 30 Ekim 2013].
- [3]Adobe Flash, ActionScript Yardımcı el kitabı. [http://help.adobe.com /
tr_TR/ActionScript/3.0_ProgrammingAS3/ flash_as3_programming.pdf](http://help.adobe.com/tr_TR/ActionScript/3.0_ProgrammingAS3/flash_as3_programming.pdf)
[Erişim tarihi: 01 Ocak 2014].
- [4]MEGEP, Mesleki Eğitim ve Öğretim Sisteminin Güçlendirilmesi Projesi –
Etkileşimli Animasyonlar Modülü, Ankara 2012.
- [5]SCHULZE Pati.(2004) Macromedia Fireworks 8 Kaynağından Eğitim, Medyasoft
Yayınları.

DİNAMİK VERİLER



İÇİNDEKİLER

- Flash Animasyonlarda Metinlerle Çalışmak
- Flash Animasyonlarda Dış Kaynaktan Veri Çekmek
- Web Servisler
- Flash Animasyonlarda Web Servis Kullanımı
- Flash Animasyonlarda XML ile Çalışmak
- Flash Animasyonlarda TXT ile Çalışmak



HEDEFLER

- Bu üniteyi çalıştıktan sonra;
- Metin alanlarının farklarını görebilecek,
- Dinamik verileri animasyonlarda kullanabilecek,
- Dış kaynak dosyaları ile bilgi alışverişi yapabilecek,
- Haricî komut dosyaları ile iletişim kurabileceksiniz.



Atatürk Üniversitesi

Açıköğretim Fakültesi

ANİMASYON TASARIMI

**Öğr. Gör
ArifDAŞ**

ÜNİTE 13

GİRİŞ

Günlük hayatta sürekli değişiklik gösteren yapılar vardır. Hava durumu ve döviz kurları gibi yapıları web sayfalarına eklediğimizde bunların her gün ve bazen her an güncellenmesinin gerekebilir. Bu gibi bilgilerin sitenin kaynak kodlarından sürekli düzenlenmesi hem ciddi bir iş yüküne hem de zaman kaybına sebep olduğu düşünülürse; bu güncellemelerin yapılması için sistematik bir yapı kullanılmalıdır.

Bu tarz yapılar dinamik veriler adı verilmektedir. Dinamik veriler bireysel değişikliklere imkân verdiği gibi, verilerinizi depolayarak dosyanız veya siteniz çalışır durumdayken bilgilerin güncellenmesini de sağlar.

Dinamik veriler 3 grupta incelenebilir. Bunlar:

- Kullanıcı destekli bilgiler
- Dinamik metin alanları
- Kontrol edilemeyen veriler (dış kaynaklar) olabilir.

Kullanıcı destekli veriler, animasyonunuz çalışırken kullanıcı tarafından girilen verileri temsil eder. Dinamik metin alanları ise animasyon çalışırken kullanıcı tarafından girilen veya başka bir dış kaynaktan alınan verilerin görüntülenmesinde kullanılır. Kontrol edilemeyen veriler ise başka bir kaynaktan alınan ve değiştirilemeyen bilgileri tutar. Örneğin sistem saati veya başka bir siteden alınan anlık döviz kuru gibi bilgiler kontrol edilemeyen veriler grubuna girer.

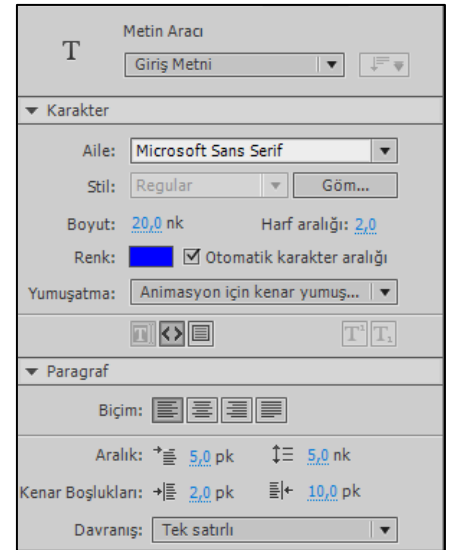
FLASH ANİMASYONLARDA METİNLERLE ÇALIŞMAK

Flash animasyonlarda en çok kullanılan nesnelerden biri de metinlerdir. Animasyona bir metin nesnesi eklemek için Araçlar (Tools)

panelinde bulunan  sembolüne tıklanır. Bu araç seçildiğinde özellikler (properties) panelinden araç ile ilgili düzenlemelerin yapılabileceği bilgilere ulaşılabilir. Panel genel düzenleme, karakter ve paragraf olmak üzere şekil 13.1’de gösterildiği gibi üç bölümden oluşur.

Yazı aracı seçildikten sonra sahneye bir yazı alanı çizilir. Bu alana istediğimiz yazı eklenir.

Daha sonra özellikler panelinde bulunan karakter ve paragraf bölümlerinden yazının stili,



Şekil 13.1 Metin alanı özellik penceresi

fontu, rengi, sayfa içi yerleşimi gibi özellikler düzenlenir. Genel özellikler panelinde ise yazı alanının türü seçilir.

Yazı alanı üç farklı türde olabilir. Metin türleri şekil 13.2’de gösterildiği gibi aşağıda listelenmiştir.

- Statik Metin (Static Text)
- Dinamik Metin (Dynamic Text)
- Giriş Metni (Input Text)

Yazı alanlarındaki bu farklılıklar kullanım amaçları ve işlevsellikleri açısından ortaya çıkmaktadır.

Statik metin: Diğer adıyla durağan metin türündeki yazı alanlarının içerikleri istendiği şekilde düzenlenebilir. Yalnız bu tür yazı alanları

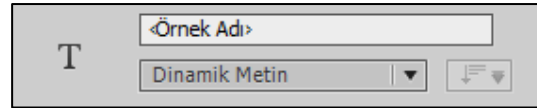


Şekil 13.2 Metin türleri

animasyonun kaynak dosyasında kullanılarak düzenlenir ve yapılan her güncelleştirmede animasyonun çalışır hâli yeniden oluşturulmalıdır. Çünkü animasyon çalışır hâle dönüştürülürken yazı alanındaki ifade kalıcı olarak animasyona eklenmiş olur ve değiştirilemez. Bu nedenle animasyon içerisinde sabit olan yazılar için kullanılır.

Giriş metni türündeki yazı alanı animasyon aracılığı ile kullanıcılardan bilgi alınması amacıyla kullanılır. İletişim formları gibi yapılarda kullanılır ve kullanıcının bilgi girişi yapmasına imkân sunar.

Dinamik metin türündeki yazı alanı ise belirli bir değişken veya dış kaynağa bağlanılabilir. Böylece animasyon çalışır hâle getirildiğinde bile dinamik metin



Şekil 13.3 Metin türü seçimi ve metin alanı isimlendirme

ifadeleri istenildiğinde değiştirilebilir. Daha çok döviz kurları, oyunlardaki kullanıcı bilgileri gibi sürekli değişen yapılarda kullanılır.

Bir dinamik metin alanını kaynak kodlar yardımıyla yönetmek için, dinamik metin alanına “örnek ad” verilir. Bu örnek ad değişken isimlendirmesi ile aynı şartlarda olmalıdır. Örnek ad özelliği metin alanı seçildikten sonra özellikler penceresinde genel özellikler grubundan düzenlenebilir. Şekil 13.3’te örnek ad özelliğinin tanımlanacağı kısım gösterilmiştir.



Dinamik metinlerin kaynak kodda kullanımını sağlayan özelliğine **örnek ad(instance name)** denir.



Örnek

- tasarladığımızda her gün hatta belirli saatlerde sistemin
- güncellenmesi gerekir. Eğer bu bilgiler sabit metin şeklinde girilmiş olsaydı her güncelleme işleminde animasyonun kaynak dosyası(.fla) üzerinde değişiklik yapılması ve animasyonun tekrar derlenmesi gerekir.
- Dinamik yapılarda ise bu durum daha farklıdır. Sistem bilgileri dış bir kaynaktan alınabilir. Böylece sistemi kaynaktan değiştirmek gerekmez. Dış kaynak değiştiğinde sistemdeki
- bilgi de değişir.

Flash Animasyonlara Dış Kaynaktan Veri Çekmek

Flash animasyonlarda dinamik metin alanları kullanılarak dış kaynaktaki veriler kolaylıkla animasyonun içerisinde aktarılabilir. Böylece animasyon içerisindeki bilgilerin sürekli güncel olması sağlanabilir.

Bir animasyon *içerisine* .txt ve .xml dosyalarından veri alınabileceği gibi web sayfalarından da veri alınabilir. Ayrıca flash tabanlı oyunlarda da kullanıcı bilgisi ve verilerinin işlenmesi yine dinamik yazı alanları ile sağlanmaktadır.

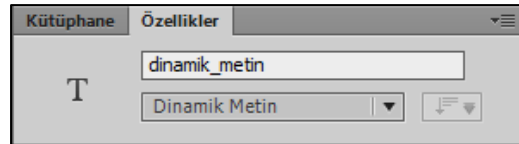
Basit bir örnek ile dinamik bir yazı *alanına* .txt dosyasından veri nasıl alınır inceleyelim.

1. Boş bir flash animasyon açınız ve içerisine bir yazı alanı ekleyiniz. (Şekil 13.4'teki gibi)

Dinamik metin alanı

Şekil 13.4 Metin alanının belirlenmesi

2. Yazı alanının türünü dinamik olarak seçiniz.
3. Yazı alanını seçtikten sonra Özellikler Penceresinden **<örnek ad>** özelliğine **dinamik_metin** yazınız (Şekil 13.5).



Şekil 13.5 Metin alanını isimlendirme

4. İlk frame'i seçiniz ve F9 tuşuna basarak eylemler panelini açınız.
5. Eylemler paneline aşağıdaki kodları yazınız ve dosyaya bir isim vererek masaüstüne kaydediniz.

Flash animasyonlarda TXT ve XML gibi dış kaynaklardan veri almak mümkündür.

```
1. var yazıYükleyici:URLLoader = new URLLoader();
2. yazıYükleyici.addEventListener(Event.COMPLETE,
   açılınca_fonksiyonu);

3. function açılınca_fonksiyonu (e:Event):void
{

4.   dinamik_metin.text = e.target.data;
5.   addChild(dinamik_metin);
}

6. yazıYükleyici.load(new URLRequest("metin.txt"));
```

Kaynak kodların açıklaması;

1. Text dosyasındaki metnin içerisindeki verileri içerisinde tutacak değişken tanımlanır. Bu kısımda **yazıYükleyici** değişkene verdiğimiz isimdir.
2. Değişkenin içerisindeki bilgiyi tetikleyecek fonksiyon çağırılır.
3. Fonksiyon tanımlanır.
4. Fonksiyon çağırıldığında **dinamik_metin** nesnesinin içerisine değişkendeki değer çekilir.
5. **dinamik_metin** aktifleştirilir.
6. Değişken içerisine verinin alınacağı kaynak belirlenir. Kaynak adresi URLRequest kısmına yazılır.

Metin dosyasının hazırlanması;

6. Bilgisayarımızda bulunan Not Defteri programını açıp içerisine istediğimizi yazıyı ekleyelim.
7. Not defterini **metin.txt** ismi ile masaüstüne kaydedelim.
8. Bütün işlemler bittikten sonra flash dosyamızı CTRL + Enter tuş kombinasyonu ile çalıştıralım.
9. Böylece dinamik metin alanımıza **metin.txt** dosyasındaki ifade eklenecektir (Şekil 13.6).



Şekil 13.6 Uygulama çalışmadan önceki ve sonraki ekran görüntüleri

Web servisler

Bazı sistemler güvenlik sebebiyle dışarıya kapalıdır. Fakat bu sistemlerden bilgi istenmesi durumunda da isteklere cevap verirler. Bunun için dinamik bir yapı olan web servisler kullanılmaktadır. Web servisler aslında birer web sayfası olarak düşünülebilir. Bu yapılar sunucu bilgisayarda bulunan bilgileri güvenlik sorgulamalarını yaparak gerekli istemci bilgisayarlar ile paylaşır. Veri çıktısı olarak genellikle XML formatı kullanılır. XML formatı bütün platformlar tarafından desteklendiği için platform bağımsızlığı söz konusudur.

Bir web servis tek başına bir anlam ifade etmez. Bu sistemin anlamlı bir şekilde çalışması için sistemden bilgi isteyen istemci sistemlerinde olması lazımdır. İstemciler internet ortamından servislere bağlanabileceği gibi yerel ağ üzerinden de çalışabilirler.

Web servisler çalışma stratejisi olarak istemci bilgisayarda özel bir portu iletişime açar ve bu port üzerinden sunucu bilgisayar sorgu gönderir. Sorguyu karşılayan sunucunun cevabı da tekrar ilgili port üzerinden gönderilir. Sonrasında ilgili port yeniden iletişime kapatılır ve böylece iletişim tamamlanmış olur.

Web servisi mimarisi katmanları şunlardır:

- **SOAP (Simple Object Access Protocol-Basit Nesne Erişim Protokolü):** İnternet üzerinde web servislerini çalıştırmak için kullanılan protokoldür.
- **WSDL (Web Services Description Language) :** Web servislerini tanımlama dilidir.
- **UDDI (Universal Description, Discovery and Integration) :** Web servislerinin indekslenip arşivlendiği kayıt servisi
- **RPC (Remote Procedure Call)** modelini kullanan, istemci/sunucu mantığına dayalı bir protokoldür. Dağıtık uygulamalarda ve Remote messaging için yeni bir teknoloji olmasına karşılık, alt yapısı sayesinde web üzerinde kullanmak için en uygun protokol olması nedeniyle Web Servislerinin adı SOAP'la birlikte anılmaktadır.

SOAP'ı benzeri protokollerden ayıran en belirgin ve üstün özelliği yapısının(daha doğru bir ifade ile mesaj formatının) **XML** üzerine kurulu olmasıdır.



Web servislerde veri çıktısı olarak genellikle XML kullanılır ve XML ile veri alışverişi daha kolaydır.



SOAP protokolünün mesaj formatı XML üzerine kuruludur.

- **Types:(Tip)** mesajlarda kullanılacak veri tiplerini belirtir.
- **Message: (Mesaj)** İletişimde kullanılacak mesajları tanımlar.
- **PortType :(Port tipi)** Web servisinin içerdiği işlemleri (methods) ve ilgili mesajları tanımlar.
- **Binding :(Ciltleme)** İşlem ve mesajlarda kullanılacak veri formatlarını tanımlar.
- **Port:** Binding ve web adresinden oluşan servis noktasını tanımlar. Web adresi servisin çalıştırılacağı URL'dir.
- **Service:(Servis)** Kullanılan port'lar kümesidir.

Web servisler yardımıyla yalnızca bilgi alınır. Sistem üzerinde değişiklik yapılamaz.



Örnek

- Bunu bir örnek ile açıklarsak;
- Merkez Bankası gün içerisinde döviz kurlarını belirleyen ve kendi web sayfasından yayınlayan kurumdur. Döviz kuru ile çalışan sistemler bu bilgiye merkez Bankası'nın onlara açtığı web servis sayesinde güncel olarak istedikleri an ulaşır ve kullanırlar.



Örnek

- Online mesajlaşma imkânı sunan bir web sunucusuna kayıt olduğunuzu ve bu sistemin programını bilgisayarınıza kurduğunuzu düşünün. Bu sistemin skype olduğunu düşünürsek, programı bilgisayardan açtığınızda program kullanıcı adı ve şifreyi girmenizi ister. Bütün bilgileri girip girişi onayladığımızda arka planda aşağıdaki işlemler gerçekleşir.
- Kullanıcı adı ve şifre girilir
- Sunucuya gönderilir
- Sunucu kullanıcı adı ve şifreyi veri tabanındaki
- bilgi ile karşılaştırır.
- Bilgiler doğru ise program sisteme bağlanır
- Arkadaş listesi ve diğer bilgiler alınır
- Bilgiler yanlış ise sistem erişimi engellenir
- Hata mesajı verilir

Flash Animasyonlarda Web servis Kullanımı

Flash animasyonlar da gerektiğinde birer istemci program olarak tasarlanabilir. Bu durumda bazı güncel bilgiler veya kullanıcı bilgileri gibi işlemlerde web servislerden faydalanılabilir.

Peki, bir flash animasyonda web servis bilgisi nasıl kullanılır?

Web servislerde bilgi XML olarak dışa aktarılabilir demiştik. XML dosyaları içerisinde farklı türlerden pek çok bilgi içerebilir ve çok basit bir şekilde oluşturulabilirler. Bu durumda flash animasyonlarda XML dosyaları kullanılarak bir web servisten bilgi alınması mümkündür.

Flash Animasyonlarda XML ile Çalışmak

XML yapılandırılmış bilgilerin, farklı bilgisayarlar ve kullanıcılar tarafından kullanılmasını ve düzenlenmesini kolaylaştıracak standartları belirleyen bir dildir. XML, eXtensible Markup Language'in kısaltmasıdır. XML bir ağaç yapısı ve HTML'e benzer etiketleri kullanır.

XML dosyaları düz metin dosyaları gibidir. Yalnız düz metinden farkları içerdiği bilgilerin belirli bir hiyerarşiye göre yapılandırılmış olmasıdır. İçerdiği bilgiler ve etiketler tamamen bizim isteğimize göre oluşturulur. Html gibi belirgin etiketler kullanmaz. Yalnız etiket oluşturmanın bazı kuralları vardır.

- XML dosyalarında içerik hazırlama her zaman temel bir etiket ile başlar ve yine temel etiketin kapatılması ile biter.
- Başlangıç etiketi *<etiket_adi>* şeklinde tanımlanır.
- Bitiş etiketi ise *</etiket_adi>* şeklinde tanımlanır.

Bu iki etiket arasına yeni etiketler tanımlanabileceği gibi yalnız bir etiket ile de XML dosyası tanımlanabilir.

AS3 dilinde XML dosyaları ile çalışırken bilinmesi gereken sınıflar şunlardır:

- a) **XML**: Animasyon içerisine XML dosyasından veri alırken bilgileri içerisinde tutacak olan bir değişken tanımlanır. Bu değişkenin türünü temsil eder.
- b) **XMLList**: XML dosyaları çok değerli yani çok etiketli olabilir. Bu durumda birden fazla etiket içeren XML dosyalarının verilerini tutabilecek yapı oluşturulur. Bu tür yapıları oluşturmak için ise ilgili sınıf kullanılır. Aşağıdaki XML örneği XMLList için uygun bir yapıdır.



XML dosyaları HTML benzeri etiket yapısı kullanırlar.



XML'in etiket yapılarını listelemek için XMLList sınıfı kullanılır.

Örnek

- <Plakalar>
- <sehir adi="Erzurum" plaka="25"> </sehir>
- <sehir adi="İstanbul" plaka="34"> </sehir>
- <sehir adi="Konya" plaka="42"> </sehir>
- </Plakalar>
- erişimi engellenir
- Hata mesajı verilir

Web servis tarafından oluşturulan veya bizzat oluşturduğumuz bir **.XML** dosyası ile iletişim nasıl sağlanır?

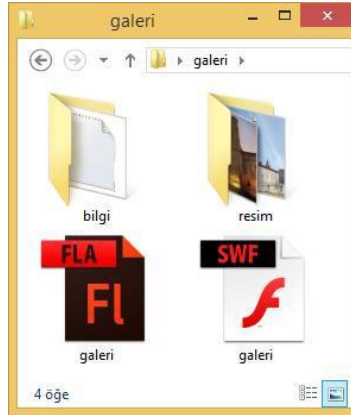
Bunu anlamak için aşağıdaki fotoğraf galerisi uygulamasını adım adım inceleyelim.

Amaç

Uygulamamızın amacı dış kaynakta bulunan resimlerin ve bu resimlere ait bilgilerin uygulamamız içerisine XML dosyasında bulunan kodlar yardımıyla aktarımını sağlamak. Uygulamada ekrana 3 adet resim gelecek ve bu resimlerin üzerine tıklandığında yanda bulunan açıklama alanına resimle ilgili açıklama yazısı eklenecektir

Uygulamaya başlamadan önce resimlerimizi ve bu resimlerle ilgili açıklamaları bulalım.

Sonrasında;



Şekil 13.7 Galerî klasörü ve iç Dizinler

- Bilgisayarımızda **Galeri** isimli bir klasör oluşturalım.
- Bu klasör içerisinde XML dosyamızı içinde barındıracak bir **bilgi** klasörü oluşturalım.
- Daha sonra galeri klasörü içerisine resimlerimiz için bir **resim** klasörü oluşturalım.
- En son boş bir AS3 flash uygulaması açalım ve **galeri** adıyla galeri klasörümüz içerisine kaydedelim.

XML Dosyasının hazırlanması:

- Dreamweaver gibi bir web editöründen boş bir xml dosyası açınız. Eğer web editör programınız yoksa Not defterini de kullanabilirsiniz.
- Açtığınız belgeyi *bilgi.xml* ismiyle galeri klasörü içerisinde bulunan bilgi klasörüne kaydediniz.
- Sonrasında *bilgi.xml* dosyasının içeriğini düzenlemeye başlayalım.
 - İlk etiketimiz <resimler> olsun. <resimler> etiketi bizim XML dosyamızın kök dizinini oluşturacaktır. Bu XML dosyası resimleri çekmemizi sağlayacağı için etiketimize bu ismi verdik,
 - <resimler> etiketinin alt dizinlerini oluşturmak için <resim> adında bir etiket açalım ve bu etiketin niteliklerini tanımlayalım,

```
<resim adres= "resmin_bulundugu_adres"
      aciklama= "resim_ile_ilgili_aciklama"></resim>
```

Yukarıdaki “adres” ve “aciklama” terimleri “resim” etiketinin niteliklerini oluşturuyor ve bu nitelikleri uygulamada kullanacağız. Adres resmin bulunduğu yerin adresini verirken, aciklama o resimle alakalı bilgiyi tutuyor.

- Her resim için yukarıdaki etiketten bir tane oluşturalım,
- Son olarak bütün etiketlerin bittiğini belirtmek için </resimler> etiketi kullanılır ve XML dosyasına bilgi girişi kapatılmış olur.
- Dosyayı kaydediniz ve kapatınız.

AS3 flash dosyasının hazırlanması:

- Galeri isimli flash animasyon dosyamızı açalım ve ilk katmanın adını eylemler olarak değiştirelim,
- Eylemler katmanının ilk karesini seçelim ve F9 tuşunu kullanarak eylemler panelini açalım,
- Aşağıdaki kodları yazalım:
 - İlk olarak animasyon çalıştığı anda XML dosyasının verilerinin animasyon içerisine yüklenmesi gerekir. Bilgileri yüklemek için bir XML yükleyici nesnesi tanımlayalım. Bu işlemi yapması için URLLoader nesnesini kullanacağız.

```
var xml_yukleyici:URLLoader= new URLLoader();
```

- Yüklenecek olan XML dosyasının kaynağı belirlenir. Bu işlem için URLRequest nesnesi kullanılır. Bilgi klasöründeki bilgi.xml dosyası animasyona yüklenecektir.



Action Script 3'te butonların olayları kodlarla düzenlenir.

Bunun için olay dinleyici (addlistener) kullanılır.

```
xml_yukleyici.load( new URLRequest("bilgi/bilgi.xml"));
```

- Yükleme işlemi tamamlanmadan yapılacak işlemler hatalı olacağı için işlemi kontrol etmek lazımdır. Yükleme işleminin bitip bitmediğini kontrol etmek için xml_yukleyici nesnemize bir olay dinleyici (*eventlistener*) ekleyeceğiz. Olayların içinden de *COMPLETE* (Tamamlandı) olayını seçelim. İşlem bittiğinde ise *"yüklendi"* isimli bir fonksiyonu çalıştırmasını isteyelim.

```
xml_yukleyici.addEventListener(Event.COMPLETE,yuklendi);
```

- *"yüklendi"* isimli fonksiyonun tanımlanması gerekir. Yalnız fonksiyonu tanımlarken bazı değişkenlere ihtiyacımız olacak. Bu nedenle önce değişkenleri tanımlamak gereklidir. Tanımlanacak değişkenler ve bu değişkenlerin türleri aşağıdaki gibidir.

```
▪ var xmldata:XML;
```

[Bu değişken XML dosyasının verilerini tutacak]

```
▪ var altdizin:XMLList;
```

[Bu değişken XML dosyasındaki altdizinleri tutacak]

```
▪ var fotoyukleyici:Loader;
```

[Bu değişken fotoğrafları içerisinde tutacak]

```
▪ var aciklama:TextField;
```

[Bu değişken fotoğraf açıklamalarını tutacak]

Fonksiyonun tanımlaması:

```
function yuklendi(e:Event)
{
    xmldata= new XML(xml_yukleyici.data); /*değişkene XML verileri aktarılıyor.*/
    altdizin= new XMLList(xmldata.children());/*değişkene XML altdizinleri
    aktarılıyor.*/
    for (var i:Number = 0; i<altdizin.length(); i++) /*Altdizinlerde tek tek işlem yapmak
    için döngü açılır.*/
    {
        fotoyukleyici= new Loader(); /*Fotoğraf yükleyicimiz tanımlandı.*/
        fotoyukleyici.load(new URLRequest(altdizin[i].@adres)); /*Alt dizinlerden tek tek
        adres bilgisi alınır.*/
        addChild(fotoyukleyici); /* Fotoğraf yükleyici sahnemize eklenir. */
        fotoyukleyici.x=20; /*Fotoğraf yükleyicinin x koordinatı belirlenir. */
        fotoyukleyici.y=i*(100+10); /*Fotoğraf yükleyicinin y koordinatı belirlenir. Alt alta
        gelmesi için değişken kullanılmıştır. */
        fotoyukleyici.name= String(i); /*Fotoğraflara tıklandığında işlem yapmak için her
        fotoğrafa isim verilir ve bu isimler için döngü değişkeni string olarak atanır.*/
        fotoyukleyici.addEventListener(MouseEvent.CLICK,aciklama_yaz); /* Fotoğraf
        yükleyiciye olay dinleyici eklenir ve farenin tıklama olayı aciklama_yaz
        fonksiyonuna bağlanır. */
    }
}
```

- Resimlerin açıklamaları için sahneye bir açıklama ekranı eklenmelidir. Bu işlem için aşağıdaki kodlar kullanılacaktır.

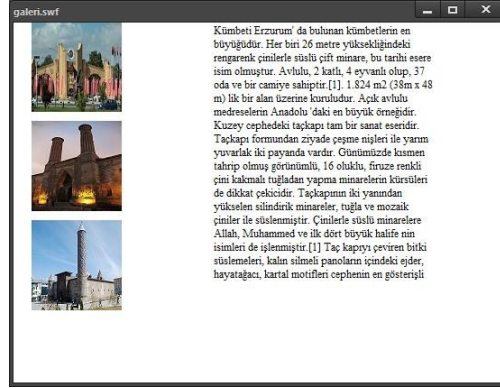
```
aciklama=new TextField();
/* açıklama için yeni bir yazı alanı tanımlanır.*/
aciklama.x=220;
aciklama.y=0;
aciklama.width=250;
aciklama.height=300;
aciklama.multiline=true;
/* açıklama birden fazla satıra sahip ise bu özellik aktif edilir.*/
aciklama.wordWrap=true;
/* açıklamada alt satıra geçiş için kullanılır.*/
aciklama.selectable=false;
/* açıklama yazısı seçilemesin. */
addChild(aciklama);

/* açıklama alanı sahneye eklendi.*/
```

- Son olarak aciklama_yaz fonksiyonu tanımlanmalıdır.

```
function aciklama_yaz(e:MouseEvent)
{
    aciklama.text=String(altdizin[e.target.name].@aciklama);
}
```

- Fonksiyonu da tanımladıktan sonra CTRL+Enter tuş kombinasyonu ile animasyonu çalıştırdığımızda ekran çıktısı şekil 13.8'deki gibi olacaktır.



Şekil 13.8 Fotoğraf Galerisi Ekran Çıktısı

Flash Animasyonlarda TXT Dosyasına Veri Gönderme

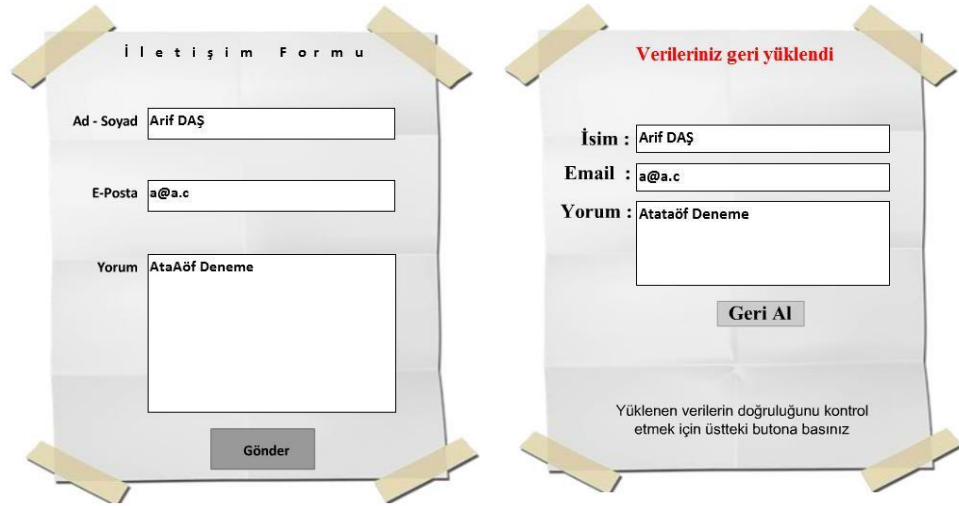
Flash animasyonlarda dış kaynaklardan bilgi alınabileceği gibi bilgi gönderiminin yapılması da mümkündür. Dış bir kaynağa veri göndermek için XML dosyaları ve TXT dosyaları kullanılabilir.

TXT dosyasına veri nasıl gönderilir ve bu veri üzerinde işlemler nasıl yapılır? Bunu daha iyi anlamak için bir örnek uygulama geliştirelim.

Uygulamada bir web sayfası için küçük bir iletişim paneli tasarlayalım. HTML dökümanın içerisine eklenen flash animasyon kullanıcının girdiği bilgileri küçük bir eklenti sayesinde sunucu bilgisayara kaydedecektir. Bu işlem sırasında kullanılacak eklenti, sunucu bilgisayarda *yukleme.txt* isimli bir belge oluşturacaktır. Daha sonra kullanıcının girdiği bilgiler bu belge üzerine kaydedilecektir. Çalışmanın server üzerinde işlem yapılabilmesi için bir PHP dosyası kullanılacaktır. Fakat aynı işlem ASP dosyasıyla da yapılabilir.



Flash animasyonlarda
TXT veya XML ile
sunucu destekli veri
iletişimi için PHP veya
ASP dosyalarından
destek alınır.



Şekil 13.9 İletişim Formu Ekran Görüntüleri

İletişim formu 2 kareden oluşmaktadır. İlk karede sırasıyla *isim*, *eposta* ve *yorum* örnek adlarına sahip 3 adet giriş metin kutusu ve *gonder* örnek adına sahip bir buton yer almaktadır. İkinci karede ise yine sırasıyla, işlem sonucunu göstermesi için *sonuc*, gönderilen bilgileri geri almak için de *isim_txt*, *eposta_txt* ve *yorum_txt* örnek adlarına sahip 4 adet dinamik metin kutusu bulunmaktadır. *gerial* örnek adına sahip buton ise verilerin geri çekilmesi işlemlerinde kullanılacaktır. Şekil 13.9'da birinci ve ikinci karelerdeki metin kutuları ve butonlar gösterilmektedir.

Nesnelerin bulunduğu katmanın haricinde birde eylem kodlarını barındıracak bir katman açalım ve bu katmanın adını *eylemler* olarak belirleyelim. Animasyondaki katmanlar şekil 13.10'da gösterilmektedir.

Eylemler katmanının ilk karesine verilerin gönderimi için gerekli kodlar yazılır. İkinci karesine ise doğrulama ve bilgileri geri çekmek için ilgili kodlar yazılır.



Şekil 13.10 İletişim Formu Katmanları

Eylemler katmanının ilk karesindeki kodlar aşağıdaki gibidir:

```
1. gonder.addEventListener(MouseEvent.CLICK,iletisim)
2. function iletisim(ilet:Event):void
   {
     a. if(isim.text!="" && yorum.text !="" && eposta.text!="")
       {
         i. var verilerim:URLVariables= new URLVariables();
         ii. verilerim.isim = isim.text;
             verilerim.yorum = yorum.text;
             verilerim.eposta = eposta.text;
         iii. var gonderici:URLRequest=new URLRequest("kaydet.php");
         iv. gonderici.method=URLRequestMethod.POST;
             gonderici.data=verilerim;
         v. var yukleyici:URLLoader= new URLLoader(gonderici);
         vi. yukleyici.addEventListener(Event.COMPLETE,tamamlandi);
         vii. yukleyici.dataFormat=URLLoaderDataFormat.VARIABLES;
         viii. yukleyici.load(gonderici);
       }
   }
3. function tamamlandi(olay:Event):void
   {
     a. if(olay.target.data.yazdir=="tamam")
       {
         i. gotoAndStop(2)
         ii. sonuc.text = "Verileriniz kaydedildi"
       }
     b. else
       {
         i. gotoAndStop(2);
         ii. sonuc.text = "Veri kaydında hata oluştu"
       }
   }
4. }
5. stop()
```

Yukarıdaki kodlarda;

1. Gönder butonuna tıklandığında *iletisim* isimli fonksiyonun çalıştırılmasını sağlıyor.
2. İletisim isimli bir fonksiyon tanımlanır. Fonksiyonda *ilet* isimli bir parametre tanımlanır ve fonksiyon sonuçları bu değer ile bildirir.
 - a. Giriş metin kutularının içlerinin boş olup olmadığı kontrol edilir ve boş değilse aşağıdaki eylemler gerçekleştirilir.
 - i. *verilerim* isimli değişkenler sınıfı tanımlanır.
 - ii. *verilerim* değişkenlerine sırasıyla değerler atanır.
 - iii. *gonderici* isimli bir *URLRequest* sınıfı tanımlanır.
 - iv. *gonderici* sınıfının yöntemi *method* deyimi ile *POST* yani gönderme olarak belirlenir.
 - v. *yukleyici* isimli *URLLoader* sınıfı tanımlanır. *URLLoader* sınıfı bir URL veya bir belgeden metin, ikili veri veya URL olarak kodlanmış verileri indirir.
 - vi. *yukleyici* sınıfının olayları dinlenir ve yükleme işlemi bittiğinde *tamamlandi* fonksiyonunun çalışması sağlanır.
 - vii. *yukleyici* sınıfının veri türü (*DataFormat*) değişken olarak belirlenir.
 - viii. *yukleyici*'nin verileri alacağı kaynak belirlenir.
3. *Tamamlandi* isimli bir fonksiyon tanımlanır ve parametresi olay olarak belirtilir.
 - a. Olay parametresinin yazdır verisi *tamam* değerini dönderirse
 - i. Animasyonun ikinci kareye gidip durması sağlanır.
 - ii. Sonuç isimli dinamik metin kutusuna "*Verileriniz kaydedildi*" deyimi yazdırılır.
 - b. Şart sağlanmıyorsa
 - i. Animasyonun ikinci kareye gidip durması sağlanır.
 - ii. Sonuç isimli dinamik metin kutusuna "*Veri kaydında hata oluştu*" deyimi yazdırılır.
4. Fonksiyonlar kapatılır.
5. Animasyon ikinci kareye gitmesini diye *STOP()* yazılır.

Eylemler katmanının ilk karesindeki kodlar aşağıdaki gibidir:

```
stop();  
1. gerial.addEventListener(MouseEvent.CLICK,bilgial)  
2. function bilgial(istek:Event):void  
   {  
     a. var veri:URLLoader= new URLLoader();  
     b. veri.dataFormat=URLLoaderDataFormat.VARIABLES;  
     c. veri.load( new URLRequest("yukleme.txt"));  
     d. veri.addEventListener(Event.COMPLETE,yukle);  
3. function yukle(e:Event):void  
   {  
     a. sonuc.text = "Verileriniz geri yüklendi";  
     b. isim_txt.text=e.target.data.isim;  
        eposta_txt.text=e.target.data.eposta;  
        yorum_txt.text=e.target.data.yorum;  
   }  
4. }
```

1. *gerial* isimli butona tıklandığında *bilgial* isimli fonksiyonun çalışması sağlanır.
2. *bilgial* isimli fonksiyon tanımlanır.
 - a. *veri* isimli *URLLoader* sınıfı tanımlanır.
 - b. *veri* değişkeninin formatı belirlenir.
 - c. *veri* içerisine URLRequest sınıfı kullanılarak bilgi çekilir.
 - d. Değişken dinlenir ve yükleme tamamlandığında *yukle* isimli fonksiyonun çalışması sağlanır.
3. *Yukle* isimli fonksiyon tanımlanır ve parametre *e* olarak tanımlanır.
 - a. *Sonuc* isimli dinamik metin aracına “Verileriniz geri yüklendi” deyimini yazdırılır.
 - b. *İsim, eposta* ve yorum bilgileri tek tek çekilir.
e parametresinin içerisindeki *bilgi.target.data* deyimini ile anılır ve sonrasında kendi etiketlerimizi kullanarak istendik bilgiye erişilir.
4. Fonksiyon bitirilir.

Yukarıdaki örneğin çalışması için gerekli olan PHP ve ASP kodları aşağıdaki gibidir. PHP veya ASP dosyalarından hangisi ile iletişim kurulacaksa eylemler

katmanının ilk karesindeki kodlarda URLRequest sınıfının adresi belirlenmelidir. PHP için “*kaydet.php*”, ASP için “*kaydet.asp*” dosya adresleri kullanılmalıdır.

Kaydet.php dosyası kodları

```
<?php
$ isim = $_POST['isim'];
$eposta= $_POST[' eposta '];
$yorum = $_POST['yorum'];
$ kaydet = "isim=" . $ isim . "&eposta =" . $eposta . "&yorum=" . $ yorum;
$fp = fopen("yukleme.txt", "w");
if(fwrite($fp, $ kaydet)) echo "yazdir=tamam";
else echo "yazdir=hata";
fclose($fp);
?>
```

Kaydet.asp dosyası kodları

```
<%
Dim objFSO
Dim objTextStream
Dim PathFile
PathFile = Server.MapPath("yukleme.txt")
set objFSO = createobject("Scripting.FileSystemObject")
Set objTextStream = objFSO.OpenTextFile(PathFile, 2, True)
objTextStream.Write "isim=" & Request("isim") & "&yorum=" &
Request("yorum") & "&eposta =" & Request("eposta ")
Response.Write "yazdir=tamam"
objTextStream.Close
Set objTextStream = Nothing
Set objFSO = Nothing
%>
```

Flash Animasyonlarda Haricî Komut Dosyalarıyla İletişim Kurma

Flash animasyonlarda haricî veri dosyaları yüklemenin yanı sıra, sunucu taraflı komut dosyasına değişkenler gönderip sunucunun yanıtını işlemek için de URLVariables sınıfını kullanabilirsiniz.



Örnek

- Bir oyun veya web sayfasında kullanıcı girişi yapmak için kullanıcı adı ve şifre bilgilerini sunucuya gönderip sunucudan bilgilerin doğruluğu ile ilgili cevap almak için kullanılabilir.



Flash animasyonda iletişim yöntemleri URLRequest sınıfının method parametresi ile belirlenir.

Aşağıdaki kod parçası, *isim* adında yeni bir değişken oluşturan *degiskenler* adında bir *URLVariables* nesnesi oluşturur. Daha sonra, değişkenlerin gönderileceği sunucu tarafı komut dosyasının adresini belirten bir *URLRequest* nesnesi oluşturulur. Ardından, değişkenleri bir *HTTP POST* isteği olarak göndermek için, *URLRequest* nesnesinin *method* özelliğini ayarlarsınız. *URLVariables* nesnesini URL isteğine eklemek için, *URLRequest* nesnesinin *data* özelliğini daha önce oluşturulan *URLVariables* nesnesine ayarlarsınız. Son olarak, *URLLoader* örneği oluşturulur ve *URLLoader.load()* yöntemi çağrılır, böylece istek başlatılır.

```
var degiskenler:URLVariables = new URLVariables("isim=arif");
var gonder:URLRequest = new URLRequest();
gonder.url = "http://www.[web syafasının adresi].com/webservis.cfm";
gonder.method = URLRequestMethod.POST;
gonder.data = degiskenler;
var yukleyici:URLLoader = new ULLoader();
yukleyici.dataFormat = ULLoaderDataFormat.VARIABLES;
yukleyici.addEventListener(Event.COMPLETE, completeHandler);
try
{
    yukleyici.load(gonder);
}
catch (Hata:Error)
{
    trace("Unable to load URL");
}

function completeHandler(event:Event):void
{
    trace(event.target.data.HosgeldinMesaji);
}
```

Yukarıdaki örnekte kullanılan Adobe ColdFusion® webservis.cfm belgesinin kodları aşağıdaki gibidir:

```
<cfif NOT IsDefined("Form.isim") OR Len(Trim(Form.isim)) EQ 0>
    <cfset Form.isim = "Stranger" />
</cfif>
<cfoutput>HosgeldinMesaji=#UrlEncodedFormat("Merhaba, " &
Form.isim)#
</cfoutput>
```



Özet

- Günlük hayatta sürekli değişiklik gösteren yapılar vardır.
- Hava durumu ve döviz kurları gibi yapıları web sayfalarına eklediğimizde bunların her gün ve bazen her an güncellenmesi gerekebilir.Flash animasyonlarda dinamik metin alanları kullanarak dış kaynaktaki veriler kolaylıkla animasyonumuz içerisinde aktarılabilir. Böylece animasyon içerisindeki bilgilerin sürekli güncel olması sağlanabilir.
- Dinamik veriler 3 grupta incelenebilir. Bunlar:
- Kullanıcı destekli bilgiler
- Dinamik metin alanları
- Kontrol edilemeyen veriler (dış kaynaklar) olabilir.
- Flash animasyonlarda dış kaynaklı verilerde genellikle XML veya TXT dosyaları kullanılır.
- XML dosyaları içerisinde farklı etiketler kullanarak farklı türlerde bilgi barındırabilir. Flash animasyonlar XML dosyalarını yönetmek için XML adında bir sınıfa sahiptir. Bu sınıf sayesinde XML dosyaları ile çalışmak daha kolaydır. Bu nedenle flash animasyonlarda Web servis bağlantıları yapılırken en çok tercih edilen bilgi çıktısı olarak XML kullanılır. SOAP protokolü ile yapılan bağlantılarda XML çıktısı almak mümkündür.

DEĞERLENDİRME SORULARI

- I. Statik Metin (Static Text)
 - II. Dinamik Metin (Dynamic Text)
 - III. Giriş Metni (Input Text)
1. Yukarıdakileri hangisi ya da hangileri yazı alanı türlerindendir?
 - a) Yalnız I
 - b) Yalnız II
 - c) I ve II
 - d) II ve III
 - e) I, II, III
 2. Dinamik nesnelerin kodlamada kontrol edilebilmesi için kullanılan özellikleri aşağıdakilerden hangisidir?
 - a) Örnek ad
 - b) Erişilebilirlik
 - c) Metin türü
 - d) Davranış
 - e) Filtreler
 3. Flash animasyonlarda içeriği yalnızca kaynak kodlarca değiştirilen yazı alanı aşağıdakilerden hangisidir?
 - a) Dinamik Metin
 - b) Giriş Metni
 - c) Çıkış Metni
 - d) Statik Metin
 - e) Düz Metin
 4. Aşağıdakilerden hangisi bir URL veya bir belgeden metin, ikili veri veya URL olarak kodlanmış verileri indirmek için kullanılan sınıftır?
 - a) URLRequest
 - b) URLLoader
 - c) Load
 - d) URLVariables
 - e) Eventlistener
 5. Aşağıdakilerden hangisi bir nesnenin olaylarını kontrol etmek amacıyla kullanılan sınıftır?
 - a) Eventlistener
 - b) URLListener
 - c) URLRequest
 - d) DataFormat
 - e) Post

6. Kullanılan portları listeleyen Web servisi temel elemanı aşağıdakilerden hangisidir?
- a) Port
 - b) PortType
 - c) Service
 - d) Message
 - e) Binding
7. Dinamik metinlerin kaynak kodlarca kullanılmasını sağlayan özellik aşağıdakilerden hangisidir?
- a) Örnek ad
 - b) Paragraf
 - c) Karakter
 - d) Filtre
 - e) Erişilebilirlik
8. Aşağıdakilerden hangisi dış kaynaktan alınan verileri değişkenlerin içine atmak için kullanılan sınıftır?
- a) URLRequest
 - b) URLLoader
 - c) Load
 - d) URLVariables
 - e) Eventlistener
9. URLRequest sınıfının yöntemlerinden(method) hangisi bilgi çekmek için kullanılır?
- a) Get
 - b) Post
 - c) Delete
 - d) Put
 - e) Head
10. URLRequest sınıfının yöntemlerinden(method) hangisi bilgi göndermek için kullanılır?
- a) Get
 - b) Post
 - c) Delete
 - d) Put
 - e) Head

Cevap Anahtarı

1.e, 2.a, 3.d,4.b,5.a,6.c,7.a,8.d,9.b,10.a

YARARLANILAN KAYNAKLAR

- [1] Adobe Flash Pro CC, Yardımcı el kitabı. <http://helpx.adobe.com/tr/support>
[Erişim tarihi: 01 MART 2014].
http://www.flashdersleri.com/html_dosyalar/veri_yukleme.html [Erişim tarihi: 10 MART 2014].
- [2] Actionscript 3.0 Kullanarak TXT Dosyaları İle Çalışmak.
<http://www.turkcescript.com/> [Erişim tarihi: 05 MART 2014].

EK

Xampp Local Server Kurulumu

Xampp yazılımı bilgisayarınıza yerel bir sunucu kurmanızı sağlayan yazılımdır. Yani bilgisayarınızda sunucu tabanlı yazılımların çalışması için gerekli olan sistemleri bilgisayarınıza kuran programdır.

Bu program sayesinde bilgisayarınız php tabanlı sistemleri rahatlıkla çalıştırabilir. Kitap içerisinde anlatılmış olan iletişim formu uygulamasını test edebilmeniz için bu yazılımı bilgisayarınıza kurmanız gerekmektedir.

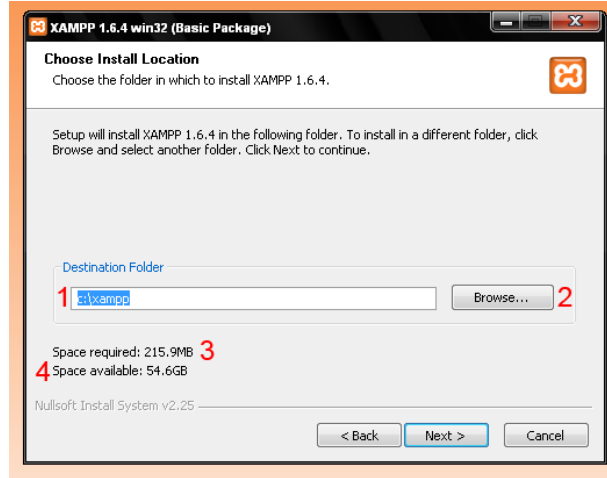
Peki, bu yazılım nasıl kurulur ve kullanılır?

- Öncelikle programı aşağıdaki adresten indiriniz.
<http://www.apachefriends.org/en/xampp.html>
- İndirdiğiniz dosyayı çalıştırdığınızda aşağıdaki ekran açılacaktır. Bu ekranda programın dilini seçiniz ve OK tuşuna basınız.

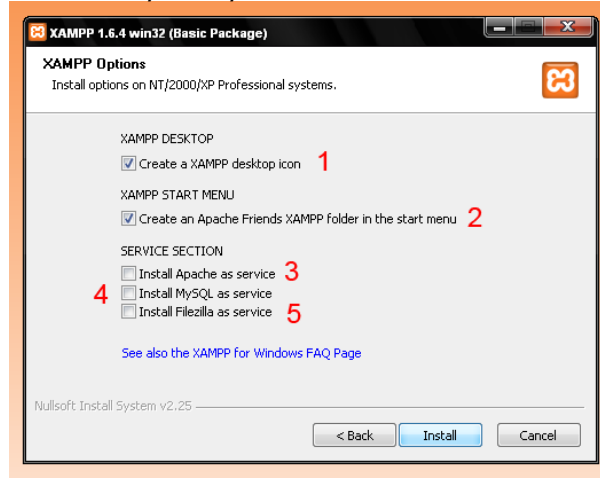


- Dili seçtikten sonra aşağıdaki pencere açılır. NEXT'e tıklayınız.



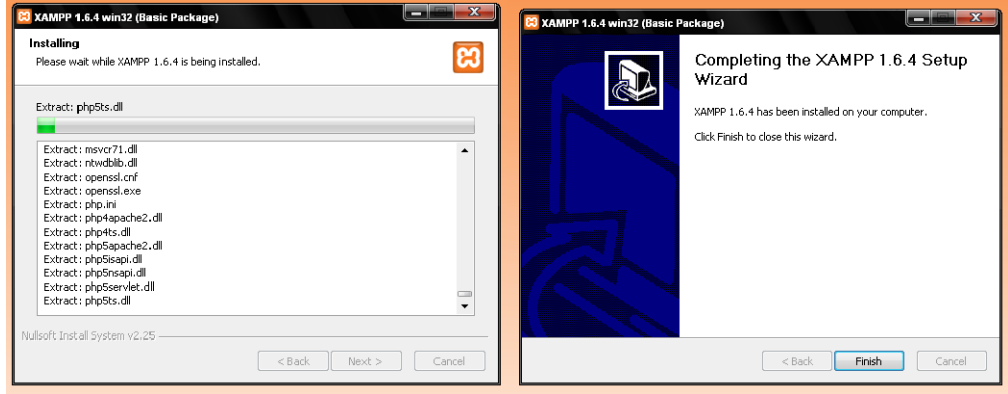


1. Kurulumun yapılacağı adres dizini.
2. Kurulum dizinini istersek değiştirebileceğimiz buton. (Varsayılan alan her zaman daha iyi bir seçimdir)
3. Kurulum 215.9 Mb'lık bir yer kaplayacaktır.
4. Harddiskte kullanılabılır 54,6 Gb alan olduğunu belirtiyor. Sonrasında Next diyoruz yine..

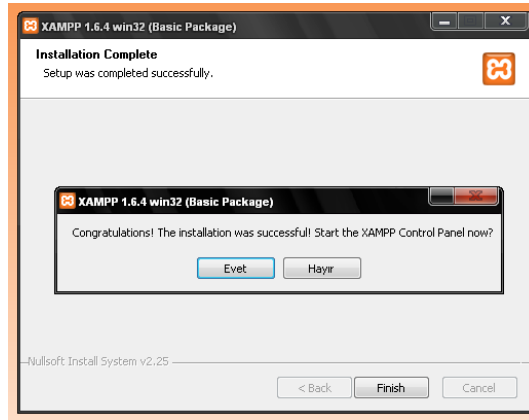


- 1) Xampp programımızın simgesini masaüstüne kısayol icon oluştursun.
- 2) Apache ve Xampp klasör kısayollarını başlat menüsüne oluştursun.
- 3) Apache Servisini kur (İşaretliyoruz)
- 4) Mysql Servisini kur (İşaretliyoruz)
- 5) Filezilla programını kur (İsterseniz işaretleyin. Filezilla ftp bağlantı programıdır. Localde çalışmak isteyenler kurmayabilir.)

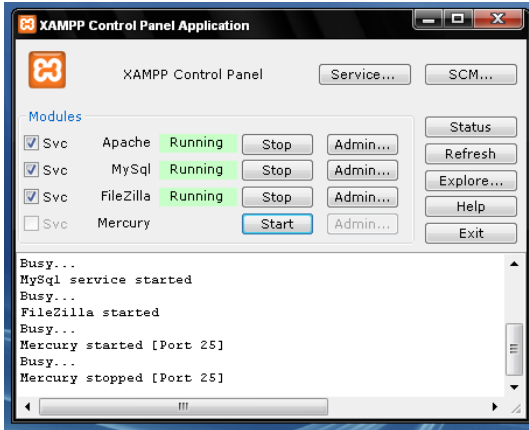
Install (Kurulum) a tıklıyoruz. Kurulum devam ediyor.



Kurulum bitti ve Finish butonuna tıklıyoruz.



Kurulumun sorunsuz bittiğini ve Xampp Kontrol Panelini açmak istediğini yazıyor. Evet diyerek panele bakıyoruz.



Panelde Apache – Mysql ve Filezilla'nın sorunsuz çalıştığını yazıyor. Start ve Stop butonları ile istediğimiz servisi başlatıp durdurabiliriz. Artık PHP'yi rahatlıkla kodlayabilir ve bilgisayarımızda çalıştırabiliriz. Hazırladığınız Flash ve PHP dosyalarını **C:\xampp\htdocs** yoluna kaydedin. Çalıştırmak için de tarayıcınızın adres çubuğunda

- <http://localhost/dosyaadi.php>
- <http://127.0.0.1/dosyaadi.php> yollarından birini kullanacaksınız.

EYLEM KODLARI



İÇİNDEKİLER

- Eylem Kodları
- Dış Kaynaklar ile İletişim
- Dış Kaynaklarla İletişim Esnasında Kullanılan Sınıflar
- Kod Parçacıkları



HEDEFLER

- Bu üniteyi çalıştıktan sonra;
- Eylem kodlarını tanımlayabilecek,
- Dış kaynaklar ile iletişim kurabilecek,
- Gerekli olan sınıfları tanımlayabilecek,
- Kod parçacıklarının ne işe yaradığını algılayıp kullanabileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

ANİMASYON TASARIMI

**Öğr. Gör
ArifDAŞ**

ÜNİTE 14

GİRİŞ

Dinamik yapılar web teknolojileri arasında bilgi alışverişi gerektiği zaman kullanılır. Animasyonlar tasarlanırken dış kaynaklar ve web sayfaları ile iletişim kurmak için bazı sınıflardan faydalanılır. Bu sınıflar genelde dış kaynaklar ile iletişim kurmak, bilgileri indirmek ve indirilen bilgiler üzerinde işlem yapmak amacıyla tercih edilir. Bu ünite de iletişim esnasında kullanılan sınıfların görevleri ve kullanım şekilleri anlatılmaktadır.

Ayrıca bu ünite de flash animasyonlarda sıkça karşılaşılan kod parçacıkları panelinde yer alan kodlarda anlatılacaktır. Kod parçacıkları animasyon içerisine eklenen nesnelerin eylemlerini düzenler, zaman çizelgesinde gezinmenize imkân sunar, sahneye eklenen ses ve video nesnelerini kontrol eder ve taşınabilir aygıtlar için gereken eylemlerin düzenlenmesine imkân sağlar.

EYLEM KODLARI

Adobe Flash, animasyonda sahneye eklenen nesnenin sahnedeki konumu, rengi ve büyüklüğü action script kodları yardımıyla kontrol edilebilir. Kontrol edilen bu özelliklerle beraber nesneye dokunulduğunda veya fare ile tıklandığında yapılacak eylemlerde belirlenebilir.

Nesneler türlerine göre farklı eylemlere sahiptirler. Örneğin buton nesnesinin tıklanma eylemi varken, ses nesnesinin oynama eyleminden söz edilebilir. Yani her nesnenin türüne göre kendine ait bir eylemi bulunmaktadır. Bu eylemler action script dilinin eski sürümlerinde nesne seçili iken o nesneye ait eylem penceresi açılarak tanımlanmaktaydı. Fakat Actionsript 3.0 sürümünde eylemlerin hepsi katmanların karelerinde kodlanmaktadır. Bunun için de nesnelerin *örnek ad* özellikleri kullanılmaktadır.

Nesnelerin eylemlerini kontrol etmek için *olay dinleyici* adı verilen bir sınıfı kullandığı önceki ünitelerde anlatılmıştı. Bu ünite de kısaca hatırlatmak gerekirse; Bu sınıf sayesinde nesnelerle iletişim kurulur ve eylemlerin sonucunda yapılacak işlemler tanımlanır.

Bir olay dinleyici tanımlamak için *addEventListener()* sınıfı kullanılır.

OLAY DİNLEYİCİ TANIMLAMA

Olay dinleyicilerin animasyona eklenen bir nesnenin olaylarını tanımlamak için kullanıldığından bahsetmiştik. Olaylar gerçekleştiğinde çalıştırılacak kod deyimleri bu dinleyiciler sayesinde kontrol edilmektedir.

Actionsript 3.0'da olayları takip etmek ve kontrol etmek için *addEventListener()* sınıfı kullanılır. Peki, bu sınıf nasıl tanımlanır?



Olay dinleyicilere

Olay işleyiciler de
denilmektedir.

Dinleyici işlevlerini kontrol etmek için iki parametre kullanılır. Dinleyiciyi tanımlarken *type* ve *listener* parametrelerinin kullanımı zorunludur. Type parametresi dinleyicinin olay türünü belirlemek için kullanılır. Olay gerçekleştiğinde çalışacak kodlar ise listener parametresi ile tanımlanır. Listener parametresinde bir sınıf veya bir fonksiyonu çağırabilirsiniz. Bir dinleyici aşağıdaki gibi tanımlanır.

```
var dinleyici_ismi addEventListener(type(tür),listener(çalıştırılacak kodlar))  
var dinleyici_ismi addEventListener(MouseEvent.CLICK,islevler)
```

DIŞ KAYNAKLAR İLE İLETİŞİM

Flash animasyonlar da dış kaynaklar ile iletişim kurulabilir fakat bu işlemi yapmak için bazı sınıflara ihtiyaç duyulur. *Sınıf*; ortak metot ve özellikleri tanımlayan script'lerden oluşan bir gruptur. Yani belirli bir işlem serisini yapmak için kullanılan kodların belirli bir isim ile adlandırılması ve ihtiyaç olduğunda her defasından kod satırlarını yazmaktansa bu isim ile çağırılmasını sağlayan yapıdır.

Her sınıfın içerisinde *method* adında bazı yapılar vardır. Method'lar sınıf içerisindeki değişken yapıların kontrolleri için kullanılırlar.

Dış Kaynaklar ile iletişimde kullanılan Sınıflar

Actionscript 3.0, harici kaynaklar ile iletişim için belirli sınıflar kullanılırlar. Bu sınıflar dış kaynaklardaki, metin dosyaları gibi sabit bilgilere ulaşabilirler. Bunun yanı sıra veritabanındaki bilgiler ile bir web sayfasındaki komut satırlarında bu sınıflar yardımıyla flash animasyona dâhil edilmesi mümkündür. Ayrıca bu sınıflar veri alma işleminin yanı sıra veri gönderme işlemini de yapabilmektedirler. Peki, bu sınıflar nelerdir.

- **URLRequest SINIFI**

Dış kaynak ile iletişim kurmak için kullanılan kaynağın adresini animasyona bildirmek gerekir. Animasyon dış kaynağa erişim için gerekli olan adres bilgilerine *URLRequest* sınıfını kullanarak ulaşır.

Kısaca URLRequest sınıfı dış kaynağın adresini belirlemek ve bu kaynak ile ne tür bir iletişim kurulacağını belirlemek için kullanılır. İletişim türü olarak dış kaynağa veri gönderilmesi ve veri alınması olarak düşünülebilir. *Bu ünitenin en sonunda sınıflara ait özelliklerinin daha iyi anlaşılması amacıyla bir iletişim formu hazırlanmıştır. Bu iletişim formunun tasarım ve kodlarını inceleyerek uygulayınız.*

Kullanımı aşağıdaki gibidir;

```
var kaynak:URLRequest=new URLRequest("URL/Dosya_adresi")
```

URLRequest sınıfının özellikleri ve bu özelliklerin açıklamaları aşağıdaki gibidir.	
contentType	İçerik türünü seçmek için kullanılır. Herhangi bir standart seçilmezse varsayılan standart uygulanır. (application/x-www-form-urlencoded)
data	URL isteği ile aktarılabacak olan verileri içeren bir nesne.
digest	Adobe® Flash® Player ön belleğine saklanacak (veya buradan alınacak) imzalanmış Adobe platformu bileşenini benzersiz şekilde tanımlayan bir dize.
method	İletişim yönünü belirleyecek olan özelliktir. "GET" veya "POST" olabilir. "GET" veri alımı, "POST" ise gönderimi için kullanılır.
requestHeaders	HTTP istemine eklenecek HTTP istem başlıkları. Flash Player'da ve uygulama güvenliği sanal alanı dışında çalışan AIR içeriğinde bazı başlıkların ayarlanmasına yönelik izinlerin sınırlandırıldığı unutulmamalıdır.
url	İletişim kurulacak olan dış kaynağın adresini alır. URL veya direkt dosya adresi olabilir.

• URLLoader SINIFI

Animasyonun bir dış kaynaktan metin, ikili veri veya URL olarak kodlanmış değişkenler olarak veri indirmesi için kullanılan sınıf **URLLoader** sınıfıdır. Yani bu sınıf sayesinde dinamik bir animasyonun dış kaynaktaki bilgileri kullanmak üzere indirmesi mümkündür. Kullanımı aşağıdaki gibidir;

```
var yukleyici_ismi:URLLoader= new URLLoader(bilgi_indirilecek_dosya);
```

URLLoader, verileri indirip kullanmamızın yanı sıra indirme işlemlerinin ilerleme durumu hakkında da bildirimler verir. Bu bildirimler **bytesLoaded** ve **bytesTotal** özellikleri ve gönderilen olaylar ile izlenebilir.

URLLoader sınıfının özellikleri ve bu özelliklerin açıklamaları aşağıdaki gibidir.	
bytesLoaded	URLLoader sınıfının yükleme esnasında o ana dek indirdiği bilginin bayt değerini verir.
bytesTotal	İndirmeye çalışılan verinin toplam boyutunu verir.
data	Yükleme işlemi sonucunda alınan verilerdir. Bu veriler dataFormat özelliğinin türüne göre değişir.
dataFormat	İndirilen verinin türünü belirlemek için kullanılır. Veri türleri; - Metin (TEXT) - Ham ikili (BINARY) - Değişken (VARIABLES) olarak belirlenebilir.

URLLoader sınıfının yöntemleri ve bu yöntemlerin açıklamaları aşağıdaki gibidir.	
Close()	İlerlemekte olan yükleme işlemini kapatmak için kullanılır.
Load()	Belirlenen adresteki dosya ile bilgi alışverişi yapılmasını sağlar.

URLoader sınıfı sayesinde, *complete*, *httpStatus*, *ioError*, *open*, *progress* ve *securityError* gibi olayları dinleyebilirsiniz. Bu olaylar sayesinde veri indirme işlemlerinin durumunu ve güvenlik bilgilerini kontrol edebilirsiniz.

URLoader sınıfının olayları ve bu olayların açıklamaları aşağıdaki gibidir.

complete	URLoader sınıfının indirme işleminin tamamlandığını belirten olaydır.
httpResponseStatus	Load() yöntemi ile yapılan bir iletişim HTTP protokolü üzerinden verilere erişmeye çalışıldığında karşı taraftan cevap gelirse kullanılan olaydır.
httpStatus	Load() yöntemi ile yapılan bir iletişim HTTP protokolü üzerinden verilere erişmeye çalışırsa kullanılan olaydır.
ioError	İndirme işlemini sonlandıran kritik bir hata varsa; bu hatanın sebepleri veren olaydır.
open	Sırayla yapılan bir indirme işlemi söz konusu ise, sıradaki indirme işleminin başlangıç bilgisini veren olaydır.
progress	İndirme işleminin durumunu bildiren olaydır.
securityError	Animasyona bilinmeyen bir dış kaynaktan veri indirilmeye çalışıldığında devreye giren olaydır.

• **URLStream SINIFI**

İndirilen verilere düşük düzeyde bir erişim sağlanacaksa ve verilerin tamamı inmeden kullanılmaya başlanacaksa veriler dış kaynaktan *URLStream* sınıfı kullanılarak çekilebilir.

Youtube gibi sitelerde yeni çıkan sistemler videonun tamamı indirilmeden yayınlanmasını sağlar. Bu tarz işlemler için ilgili sınıf kullanılır. URLoader gibi metin, ham ikili ve değişken olarak veriler alabilir. ,

URLStream ögesindeki okuma işlemleri sürekli bir cevap gönderir. Başka bir deyişle, okumadan önce yeterli miktarda verinin olup olmadığını belirlemek için *bytesAvailable* özelliğini kullanılabilir, eğer yeterli miktarda veri yoksa *EOFError* hata kontrol deyiimi kullanılabilir.

URLoader ile URLStream'in olayları aynıdır. Fakat özellikleri ve yöntemlerinde farklılıklar vardır.

URLoader sınıfının özellikleri ve bu özelliklerin açıklamaları aşağıdaki gibidir.

bytesAvailable	Okunmak için bekleyen hâlihazırdaki verinin bayt değerini gönderir.
Connected	URLStream sınıfı ile oluşturulan nesnenin o anda herhangi bir dış kaynağa bağlı olup olmadığını gösterir.



Bilgilerin yalnızca indirilen kısımları ile çalışmak mümkündür. Bu işlemi URLStream sınıfı yürütür.

Endian	Veri için bayt sırasını belirtir.
objectEncoding	Bir nesneyi okurken veya yazarken kullanılan eylem mesaj formatının sürümünü kontrol eder.

- **URLVariables SINIFI**

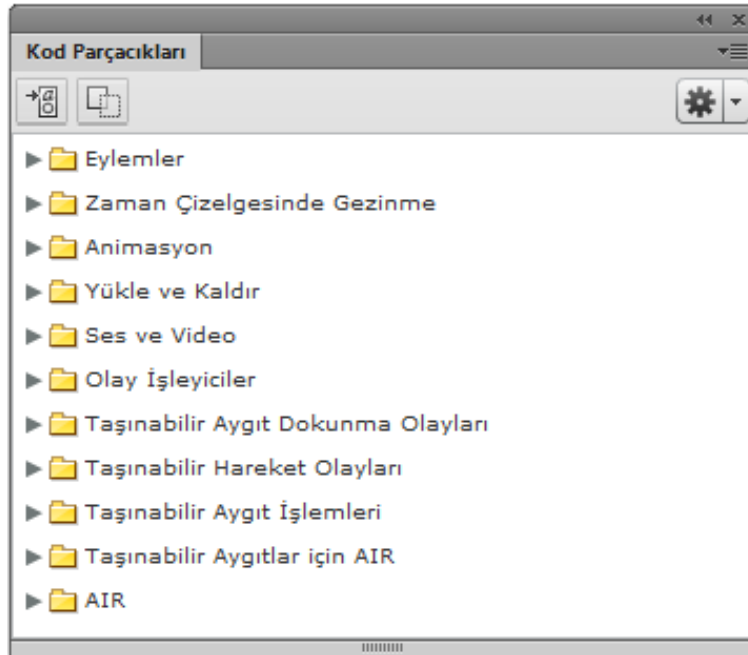
Animasyonda birden fazla veri türünde bilgiyi sunucudan almak veya göndermek için değişkenler kullanılır. Bu verileri tutan değişkenleri tanımlamak için **URLVariables** sınıfı kullanılır. URLLoader sınıfı ile alınan veriler birden fazla türde ise değişkenlere atanmak zorundadır. Bu durumda **URLLoader'in dataFormat** özelliği **VARIABLES** ise alınan veriler URLVariables ile değişkenlere alınır ve kullanılır. Kullanımı aşağıdaki gibidir;

```
var degiskenler:URLVariables=new URLVariables(yukleyici)
```

KOD PARÇACIKLARI

Adobe Flash CC ile birlikte gelen yeni özelliklerden biri de sıkça kullanılan eylemlerin hazırlanması için hazırlanan bir kod parçacıkları panelidir. Normalde hazırlanması için uzun kod karmaşası gerektiren eylem kodları artık nesne seçildikten sonra birkaç seçim işlemi ile kendiliğinden oluşturulmaktadır.

Pencere > Kod Parçacıkları komutları verilerek kod parçacıkları paneline ulaşılabilir (Şekil 14.1).



Şekil 14. 1 Kod Parçacıkları Paneli

Hazır kod parçacıkları nasıl kullanılır adım adım göstermek için;

- Boş bir flash animasyon açınız ve sahneye bir nesne ekleyiniz.
- Pencere > Kod Parçacıkları adımlarını izleyip paneli açınız.
- Eylemler grubundan ilgili eylemi seçiniz ve çift tıklayınız.
- Eylemler panelinde ilgili kodlar otomatik olarak oluşturulacaktır.

Sahneye eklenen nesne herhangi bir sembole dönüştürülmezse; Flash seçilen nesneyi otomatik olarak ilgili sembol türüne dönüştürüp örnek ad tanımlamasını yapar.

Panel içerisindeki gruplara tek tek göz atalım.

EYLEMLER GRUBU

Bir nesnenin özelliklerini yönetmek için kullanılan kod parçacıklarını içeren gruptur. Bu gruptaki kodlardan söz ederken bir nesneye bağlı olayların oluşturulup kontrol edileceğini unutmamak gerekir.

Gruptaki kod parçacıklarını amaçları, özellikleri ve kullanım şekilleri açısından inceleyelim.

- **Web Sayfasına Gitmek İçin Tıklat**

Bu kod parçacığı animasyon içerisinde bir nesneye tıklandığında; bir web sayfasının açılması isteniyorsa kullanılır. Eylemi tetikleyecek olan nesne oluşturulur ve nesne seçili iken kod parçacıkları panelinden ilgili eyleme çift tıklanır.

Çift tıkladıktan sonra eylemler paneli açıldığında göreceksiniz ki panelde nesnenin eylemi yerine getirmesi için gereken kodlama flash tarafından oluşturulmuştur.

Burada nesnenize siz bir örnek ad (instance name) tanımlamamışsanız eğer flash sırasıyla nesneye örnek ad tanımlamasını kendisi yapacaktır. Bildiğiniz üzere örnek ad tanımlaması olmadan nesnelerin özelliklerini kontrol etmek mümkün değildir.

```
/* Talimatlar:  
1. "http://www.adobe.com" adresini yönlendirme yapmak istediğiniz URL adresiyle  
değiştirin.  
Tırnak işaretlerini koruyun ("").  
*/
```

- **Özel Fare İmleci**

Bu kod parçacığı animasyondaki fare imlecinin yine animasyon içerisine eklenen bir nesne ile değiştirilmesini sağlar. Bu nesne seçildikten sonra ilgili kod parçacığı çift tıklanır ve eylemler oluşturulur.



Kod parçacıkları, seçilen nesnenin örnek adı yoksa örnek ad tanımlamasını da otomatik olarak yapar.



Belirtilen nesneye tıklattığınızda URL yeni bir tarayıcı penceresinde yüklenir.



Animasyon içerisinde varsayılan fare imlecinin yerine sahneye yüklenen bir sembol *özel fare imleci* komutu ile tanımlanabilir.

- **Sürükle Bırak**

Bu kod parçacığı animasyon içerisindeki bir nesnenin animasyon çalıştırıldığında istenen bir yere taşınmasına imkân tanıyan kodları oluşturur.

- **Film Klibi Oynat**

Durdurulmuş olan animasyonu yeniden başlatmak için kullanılır.

```
/* Talimatlar:
```

```
1. Bu kodu, o anda durdurulmuş olan film klipleri için kullanın.
```

```
*/
```

- **Film Klibi Durdur**

Sahne alanı üzerindeki belirtilen film klibini durdurur.

```
/* Talimatlar:
```

```
1. Bu kodu, o anda oynatılmakta olan film klipleri için kullanın.
```

```
*/
```

- **Bir Nesneyi Gizlemek için Tıklat**

Uygulandığı nesnenin animasyon çalıştırıldığında gizlenmesini sağlar.

```
/* Talimatlar:
```

```
1. Bu kodu, o anda görünür olan nesneler için kullanın.
```

```
*/
```

- **Nesneyi Göster**

Animasyonda gizlenmiş olan bir nesnenin animasyon çalıştırıldığında tekrar görünür olmasını sağlar.

```
/* Talimatlar:
```

```
1. Bu kodu, o anda gizli olan nesneleri göstermek için kullanın.
```

```
*/
```

- **Bir Nesneyi Konumlandırmak için Tıklat**

Animasyonda belirtilen nesneyi istenilen x ve y koordinatlarına taşır.

```
/* Talimatlar:
```

```
1. 200 değerini x koordinatıyla değiştirin  
   nesneyi konumlandırmak istediğiniz yer.
```

```
2. 100 değerini nesneyi konumlandırmak istediğiniz y koordinatıyla değiştirin.
```

```
*/
```

- **Bir TextField Ögesini Görüntülemek için Tıklat**

Belirtilen nesne tıklandığında istenilen x ve y koordinatında bir yazı alanı (TextField) ögesi oluşturulur ve görüntülenir. Bu yazı alanı içerisinde de gösterilmek istenen mesaj belirlenir.


```
/* Talimatlar:
1. 200 değerini TextField ögesini konumlandırmak istediğiniz x koordinatıyla değiştirin.
2. 100 değerini TextField ögesini konumlandırmak istediğiniz y koordinatıyla değiştirin.
3. "Lorem ipsum dolor sit amet" dize değerini, görünen TextField ögesinde görüntülenmesini istediğiniz metinle değiştirin. Tırnak işaretlerini koruyun.
*/
```

- **Rastgele bir Sayı Üret**

Bazen 0 ile belirlenen bir sayı arasında rastgele sayılar üretmek istenir. Bu durumda bu kod parçacığı kullanılabilir. Her defasında belirlenen aralıkta yeni bir sayı üretir.

```
/* Talimatlar:
1. Maksimum rastgele değeri değiştirmek için, bu parçacığın son satırındaki 100 sayısını kullanmak istediğiniz sayı olarak değiştirin.
2. Bu kod, rastgele sayıyı Çıktı paneline çıkarır.
*/

function rastgele_sayi_uret(limit:Number):Number
{
    var rastgele_sayi:Number = Math.floor(Math.random()*(limit+1));
    return rastgele_sayi;
}
trace(rastgele_sayi_uret(100));
```



Tıklatılan herhangi bir nesneyi öne getir, sahne alanı üzerindeki tüm nesneleri tıklatılabilir hâle getirir.

- **Tıklatılan Herhangi Bir Nesneyi Öne Getir**

Sahneye eklenen nesneler eklenme sıralarına göre birbirlerinin üzerine gelirler. Bu koda parçacığı ile sahne alanı üzerindeki herhangi bir sembol tıklatıldığında diğer tüm sembollerin önüne geçer.

```
/* Bu kod, sahne alanı üzerindeki tüm sembol örneklerini CLICK olayını dinlemelerini sağlayarak tıklatılabilir hale getirir. */
for (var sembol_listesi:int = 0;
    sembol_listesi < this.numChildren;
    sembol_listesi++)
{
    this.getChildAt(sembol_listesi).addEventListener(MouseEvent.CLICK,
    sembolu_onegetir);
}
/* Bu kod parçası ise, tıklatılan nesneyi görüntüleme listesinin önüne getiren işlevdir.
*/
function sembolu_onegetir(event:MouseEvent):void
{
    this.addChild(event.currentTarget as DisplayObject);
}
```

- **Basit Zamanlayıcı**

Bu kod parçacığı değiştirilmeden çalıştırılırsa 30 saniye geçene kadar **çıkıtı** panelinde bir geri sayım zamanlayıcısı görüntüler.

Basit bir geri sayaç yapmak istediğinizde size tam manasıyla karşılık verebilecek bir kod parçasıdır.

```
/* Talimatlar:
1. Zamanlayıcıdaki saniye sayısını değiştirmek için aşağıdaki ilk satırda bulunan 30
değerini istediğiniz saniye sayısı olarak değiştirin.
2. İlk satırda bulunan 1000 değeri ise zamanlayıcının hızını ayarlar ve 1000 değeri
tam olarak 1 saniyeye denk gelir. Zamanlayıcı dakika olarak çalıştırmak için 1000
yerine 60000 yazmak gerekir.
*/
var Sayac:Timer = new Timer(1000, 30);
Sayac.addEventListener(TimerEvent.TIMER, Zaman_Ayarlayici);
Sayac.start();

var gecen_zaman:Number = 1;

function Zaman_Ayarlayici(event:TimerEvent):void
{
    trace("Geçen saniye: " + gecen_zaman);
    gecen_zaman ++;
}
```

- *Geri Sayım Zamanlayıcısı*

Basit zamanlayıcıya benzer fakat tam tersi yönde çalışır. Saniye türünde belirlenen bir zaman diliminden geri sayım için kullanılır.

```
/* Talimatlar:
1. Zamanlayıcının uzunluğunu değiştirmek için kodlamadaki ilk satırda bulunan 10
değerini istediğiniz saniye sayısı olarak değiştirin.
*/
```

ZAMAN ÇİZELGESİNDE GEZİNME GRUBU

Bu grupta yer alan kod parçacıkları sayesinde katmanlardaki kareler arasında gezinmek ve işlemler yapmak mümkündür. Bu kodlarla istenilen kareden yine istenilen bir başka kareye geçilebilir ve animasyona buradan devam edilebileceği gibi animasyonun ilgili karede durdurulması sağlanabilir.

- *Bu Karede Dur*

Flash animasyonda zaman çizelgesi bu kodu eklediğiniz karede durur/duraklatılır. Ayrıca, film klipi olarak eklenen nesnelerin de zaman çizelgesini durdurmak/duraklatmak için kullanılabilir.

- *Kareye Gitmek ve Durdurmak için Tıklatın*

Belirtilen düğmeye (nesneye) tıklandığında animasyon zaman çizelgesinde belirtilen kareye gider ve animasyonun akışını durdurur.

Ana zaman çizelgesi veya film klibi zaman çizelgeleri üzerinde kullanılabilir.

Talimatlar:

1. Nesneye tıklatıldığında, animasyonun gitmesini istediğiniz kare numarasını kodda bulunan *gotoAndStop(5)* deyiminde bulunan 5 sayısının yerine girin.

- *Kareye Gitmek ve Oynatmak için Tıklatın*

Belirtilen düğmeye (nesneye) tıklandığında animasyon zaman çizelgesinde belirtilen kareye gider ve animasyonun akışı ilgili kareden çalışmaya devam eder.

Ana zaman çizelgesi veya film klibi zaman çizelgeleri üzerinde kullanılabilir.

Talimatlar:

1. Nesneye tıklatıldığında, animasyonun gitmesini istediğiniz kare numarasını kodda bulunan *gotoAndPlay(5)* deyiminde bulunan 5 sayısının yerine girin.

- *Sonraki Kareye Gitmek ve Durdurmak için Tıklat*

Belirtilen düğmeye (nesneye) tıklandığında animasyon zaman çizelgesinde bir sonraki kareye gider ve animasyonun akışını durdurur.

- *Önceki Kareye Gitmek ve Durdurmak için Tıklat*

Belirtilen düğmeye (nesneye) tıklandığında animasyon zaman çizelgesinde bir önceki kareye gider ve animasyonun akışını durdurur.

- *Sonraki Sahneye Gitmek ve Oynatmak için Tıklat*

Flash animasyonlar farklı sahnelerden oluşabilir. Belirtilen düğmeye(nesneye) tıklandığında animasyon zaman çizelgesinde bir sonraki sahneye gider ve ve animasyonun akışı ilgili sahneden çalışmaya devam eder.

- *Önceki Sahneye Gitmek ve Oynatmak için Tıklat*

Belirtilen düğmeye(nesneye) tıklandığında animasyon zaman çizelgesinde bir önceki sahneye gider ve animasyonun akışı ilgili sahneden çalışmaya devam eder.

- *Sahneye Gitmek ve Oynatmak İçin Tıklat*

Belirtilen düğmeye(nesneye) tıklandığında animasyon zaman çizelgesinde belirtilen sahneye ve kareye gider. Animasyonun akışı ilgili sahneden ve kardan çalışmaya devam eder.

Talimatlar:

gotoAndPlay(1, "Sahne 3") deyiminde;

1. "Sahne 3" adını, oynatmak istediğiniz sahnenin adıyla değiştirin.

2. 1 değeri yerine belirtilen sahnede filmin oynatılmasını istediğiniz kare numarasını girin.

ANİMASYON GRUBU

Bu grupta bulunan kod parçacıkları sayesinde sahnede bulunan nesnelerin koordinatlarını klavye tuşları ile değiştirebilirsiniz. Ayrıca nesnelerin yönlendirmeleri ile ilgili düzenlemeler yapmanızda mümkündür.

Kodlar, nesnelerin hareketlerinin yanı sıra görünürlükleri ile ilgili düzenlemelere de imkân verir.

- **Klavye Oklarıyla Hareket Ettir**

Belirtilen nesnenin klavyedeki yön tuşları ile hareket ettirilmesini sağlar.

1. Hareket miktarını artırmak veya azaltmak için kodlamadaki 5 değerlerinin yerine nesnenin tuşlara her basıldığında hareket etmesini istediğiniz piksel sayısını girin.

- **Yatay Hareket Ettir**

Sahnedeki bir nesnenin, yatay konumunu yani x özelliğini belirtilen sayıda piksel kadar azaltarak veya artırarak sola veya sağa hareket ettirilmesini sağlar.

Talimatlar:

1. Bu kod, nesneyi varsayılan olarak sağa taşır.
2. Nesneyi sola taşımak için koddaki 100 sayısını negatif bir değerle değiştirin.
3. Nesnenin hareket ettiği uzaklığı değiştirmek için koddaki 100 sayısını nesnenin hareket etmesini istediğiniz piksel sayısı ile değiştirin.

- **Dikey Hareket Ettir**

Sahnedeki bir nesnenin, dikey konumunu yani y özelliğini belirtilen sayıda piksel kadar azaltarak veya artırarak yukarı veya aşağıya hareket ettirilmesini sağlar.

Talimatlar:

1. Bu kod, nesneyi varsayılan olarak aşağı taşır.
2. Nesneyi yukarı taşımak için koddaki 100 sayısını negatif bir değerle değiştirin.
3. Nesnenin hareket ettiği uzaklığı değiştirmek için koddaki 100 sayısını nesnenin hareket etmesini istediğiniz piksel sayısı ile değiştirin.

- **Bir Kez Döndür**

Nesnenin konum özelliğini belirtilen derece kadar döndürmek için kullanılır.

Talimatlar:

1. Bu kod, nesneyi varsayılan olarak saat yönünde döndürür.
2. Nesneyi saat yönünün tersine döndürmek için koddaki 45 sayısını negatif bir değerle değiştirin.
3. Nesnenin döndüğü miktarı değiştirmek için koddaki 45 değerinin yerine istediğiniz dönüş derecelerini yazın.

- **Sürekli Döndür**

Sahnedeki bir nesneyi ENTER_FRAME olayındaki dönüş özelliğini güncelleyerek sürekli olarak döndürür.

Talimatlar:

1. Varsayılan olarak dönüş hareketi saat yönündedir.
2. Dönüş yönünü saat yönünün aksine çevirmek için koddaki 10 sayısını negatif bir değerle değiştirin.
3. Nesnenin döndüğü hızı değiştirmek için koddaki 10 değeri yerine nesnenin her karede dönmesini istediğiniz derece sayısını girin. Daha yüksek sayılar daha hızlı dönüşe yol açar.
4. Daha hızlı dönüş ise animasyonun tamamının hızını etkiler.

- **Yatay Hareketlendir**

Sahnedeki bir nesnenin ENTER_FRAME olayındaki yatay konumunu yani x özelliğini azaltarak veya artırarak nesnenin sahne alanı içerisinde sola veya sağa doğru sürekli hareket etmesi sağlanır.

Talimatlar:

1. Animasyonun varsayılan hareket yönü sağa doğrudur.
2. Animasyonun yönünü sola değiştirmek için koddaki 10 sayısını negatif bir değerle değiştirin.
3. Nesnenin hareket ettiği hızı değiştirmek için koddaki 10 sayısı yerine nesnenin her karede hareket etmesini istediğiniz piksel sayısını girin.

- **Dikey Hareketlendir**

Sahnedeki bir nesnenin ENTER_FRAME olayındaki dikey konumunu yani y özelliğini azaltarak veya artırarak nesnenin sahne alanı içerisinde aşağı veya yukarıya doğru sürekli hareket etmesi sağlanır.

Talimatlar:

1. Animasyonun varsayılan hareket yönü aşağı doğrudur.
2. Animasyonun yönünü sola değiştirmek için koddaki 10 sayısını negatif bir değerle değiştirin.
3. Nesnenin hareket ettiği hızı değiştirmek için koddaki 10 sayısı yerine nesnenin her karede hareket etmesini istediğiniz piksel sayısını girin.

- **Film Klibinde Yavaşça Görün**

Sahnedeki bir nesnenin ENTER_FRAME olayındaki alpha özelliğini artırarak nesnenin sahne alanı içerisinde yavaşça görünmesini sağlar.

Talimatlar:

1. Nesnenin çoğalarak girdiği hızı değiştirmek için koddaki `movieClip_2.alpha += 0.01;` değerini değiştirin (sayı 0'dan büyük ve 1'e eşit veya daha küçük olmalıdır). Yüksek değerler ögenin daha hızlı bir şekilde çoğalarak girmesini sağlar.

- **Film Klibinde Yavaşça Kaybol**

Sahnedeki bir nesnenin ENTER_FRAME olayındaki alpha özelliğini azaltarak nesnenin sahne alanı içerisinde yavaşça kaybolmasını sağlar.

Talimatlar:

1. Nesnenin azalarak çıktığı hızı değiştirmek için koddaki `movieClip_2.alpha += 0.01;` değerini değiştirin (sayı 0'dan büyük ve 1'e eşit veya daha küçük olmalıdır). Yüksek değerler ögenin daha hızlı bir şekilde azalarak çıkmasını sağlar.

YÜKLE VE KALDIR GRUBU

Bazen hazırlanan animasyonlara dış kaynaktan veya kütüphaneden veriler eklemek gerekebilir. Bu grupta bulunan eylem parçacıkları kütüphanede veya dış kaynaktan bulunan verilerin animasyon sahnesine alınmasını sağlar. Bunun yanı sıra sahnede bulunan semboller veya verileri sahneden kaldırmak içinde yine bu grupta bulunan kod parçacıkları kullanılabilir.

- **SWF veya Görüntüyü Bir URL'den Yükleme/Kaldırma için Tıklatın**

Kod parçacığının uygulandığı butonu tıklattığınızda belirtilen SWF veya görüntü sahnede görüntülenir. Aynı butonu ikinci kez tıklattığınızda SWF veya görüntü kaldırılır.

/* Talimatlar:

1. Aşağıdaki `"http://www.helpexamples.com/flash/images/image1.jpg"` adresini eklemek istediğiniz SWF'nin veya görüntünün URL adresiyle değiştirin. Tırnak işaretlerini koruyun ("").

2. Çağırılan SWF'nin bulunduğu etki alanından ayrı olan internet etki alanlarından gelen dosyalar özel bir yapılandırma olmadan yüklenemez.

*/

```

button_1.addEventListener(MouseEvent.CLICK, fl_ClickToLoadUnloadSWF);

import fl.display.ProLoader;
var fl_ProLoader:ProLoader;

/* Bu değişken SWF'yi yüklemek veya kaldırmak isteyip istemediğinizi takip eder.*/
var fl_ToLoad:Boolean = true;

function fl_ClickToLoadUnloadSWF(event:MouseEvent):void
{
    if(fl_ToLoad)
    {
        fl_ProLoader = new ProLoader();
        fl_ProLoader.load(new
        URLRequest("http://www.helpexamples.com/flash/images/image1.jpg"));
        addChild(fl_ProLoader);
    }
    else
    {
        fl_ProLoader.unload();
        removeChild(fl_ProLoader);
        fl_ProLoader = null;
    }
    // SWF'yi yükleme veya kaldırma isteğinize göre geçiş yapın
    fl_ToLoad = !fl_ToLoad;
}

```

- **Kütüphane'den Görüntü Yükleme İçin Tıklat**

Sembol örneğini tıklattığınızda Kütüphane'den belirtilen görüntü görüntülenir.

Kütüphane'den bir görüntü yüklemek için bu görüntünün ActionScript için Dışa Aktar özelliği açık olmalıdır ve geçerli bir sınıf adı olmalıdır.

```

/* Talimatlar:
1. Kütüphanedeki herhangi bir bitmap sembolünü sağ tıklatın ve "Özellikler..."
ögesini seçin.
2. "Bitmap Özellikleri" iletişim kutusunu genişletmek için "Gelişmiş" iletişim
kutusunu tıklatın.
3. "ActionScript için dışa aktar:Q seçeneğini etkinleştirin.
4. Sınıf metin alanına 'MyImage' yazın.
5. Tamam'ı iki kez Tıklat. */
button_1.addEventListener(MouseEvent.CLICK, fl_ClickToLoadImageFromLibrary);
function fl_ClickToLoadImageFromLibrary(event:MouseEvent):void
{

```

```
// Kütüphaneden farklı bir görüntü eklemek istiyorsanız
// Yukarıdaki adım 4'teki ve aşağıdaki koddaki Sınıf metin alanına farklı bir ad girin.
var libImage:MyImage = new MyImage();

var holder:Bitmap = new Bitmap(libImage);
addChild(holder);
}
```

- **Kütüphane'den Sahne Alanına Örnek Ekle**

Belirtilen MovieClip veya Button Kütüphane sembolünün bir örneğini sahne alanına ekler. Sahne alanına bir Bitmap Kütüphanesi sembolü eklemek istendiğinde bu kod parçasığını kullanın.

```
/* Talimatlar:
1. Kütüphanedeki herhangi bir sembolü sağ tıklatın ve "Özellikler..." ögesini seçin.
2. "Sembol Özellikleri" iletişim kutusunu genişletmek için "Gelişmiş" ögesini Tıklat.
3. "ActionScript için dışa aktar:Q seçeneğini etkinleştirin.
4. Sınıf metin alanına 'LibrarySymbol' yazın.
5. Tamam'ı iki kez Tıklat.
*/

// Kütüphaneden farklı bir sembol eklemek istiyorsanız
// Yukarıdaki adım 4'teki ve aşağıdaki koddaki Sınıf metin alanına farklı bir ad girin.
var fl_MyInstance:LibrarySymbol = new LibrarySymbol();
addChild(fl_MyInstance);
```

- **Örneği Sahne Alanından Kaldır**

Sahne alanına eklenen bir sembolü sahne alanından kaldırmak için kullanılır.

```
removeChild(button_1);
```

- **Haricî Metin Yükle**

Haricî bir metin dosyasındaki verileri animasyona çeker ve verileri **çıkı panelinde** görüntüler.

```
/* Talimatlar:
1. "http://www.helpexamples.com/flash/text/loremipsum.txt" adresini, yüklemek istediğiniz metin dosyasının URL adresiyle değiştirin.
Adres sıradan bir dosya adresi veya bir "http://" bağlantısı olabilir.
Adres tırnak işaretlerinin ("" ) içine yerleştirilmelidir.
*/

var fl_TextLoader_2:URLLoader = new URLLoader();
var fl_TextURLRequest_2:URLRequest = new
URLRequest("http://www.helpexamples.com/flash/text/loremipsum.txt");
```



```

fl_TextLoader_2.addEventListener(Event.COMPLETE, fl_CompleteHandler_2);

function fl_CompleteHandler_2(event:Event):void
{
    var textData:String = new String(fl_TextLoader_2.data);
    trace(textData);
}

fl_TextLoader_2.load(fl_TextURLRequest_2);

```

SES VE VİDEO GRUBU

Bu grupta bulunan kod parçacıklarıyla kendi müzik çalar programınızı yapmanız mümkündür. Bu grup ses ve videoları yönetmeniz için gerekli olan kodlara sahiptir. İlgili karde bulunan bütün sesleri yönetebilirsiniz.

- *Sesi Oynatmak/Durdurmak için Tıklatın*

Oluşturulan butonu tıklattığınızda belirtilen ses oynatılır. Butonu ikinci kez tıklattığınızda ise ses durur.

```

/* Talimatlar:
1. Aşağıdaki "http://www.helpexamples.com/flash/sound/song1.mp3" adresini
ses dosyasının istediğiniz URL adresiyle değiştirin. Tırnak işaretlerini koruyun ("").
*/
button_1.addEventListener(MouseEvent.CLICK, sesi_cal_durdur);

var ses_kanali:SoundChannel;

//Bu değişken sesi oynatmak veya durdurmak isteyip istemediğinizi takip eder.
var cal:Boolean = true;

function sesi_cal_durdur(evt:MouseEvent):void
{
    if(cal)
    {
        var s:Sound = new Sound(new
URLRequest("http://www.helpexamples.com/flash/sound/song1.mp3"));
        ses_kanali = s.play();
    }
    else
    {
        ses_kanali.stop();
    }
}

```

```

}
cal = !cal;
}

```

- *Tüm Sesleri Durdurmak için Tıklat*

Sembol örneğini tıklattığınızda çalınmakta olan tüm sesler durdurulur.

```
sembolunuz.addEventListener(MouseEvent.CLICK, tum_sesleri_kapat);

function tum_sesleri_kapat(event:MouseEvent):void
{
    SoundMixer.stopAll();
}
```

- *İşaret Noktasında Olayı*

Bir video oynatılırken video oynatıcıda bir *kilit noktası (cuePoint)* noktası oluşturulur ve bu nokta etkinleşince video oynatıcıya bilgi gönderir. Bu bilgi ile videonun belirlenen noktasında gerek yeni bir video gerekse bir resmin görüntülenmesi sağlanabilir. Popüler video paylaşım sitelerinde bazı videolarda ekranda birden bir reklam oluşur veya bir link ortaya çıkar. İşte bu reklam veya linkleri tetiklemek için *cuePoint* kullanılabilir.

```
/* Belirtilen video örneğinde bir kilit noktası oluşturulur ve videonun o noktadan her
geçişinde aşağıda tanımlanan kilit_Noktası işlevi yürütülür.
Talimatlar:
1. Aşağıda "// Özel kodunuzu başlatın" yazan satırdan sonra gelen boş satıra
çalıştırmak istediğiniz kod/kodları ekleyin.
Oynatılmakta olan bir videoda işaret noktalarına erişildiğinde kod yürütülür.
*/
import fl.video.MetadataEvent;
movieClip_4.addEventListener(MetadataEvent.CUE_POINT, kilit_noktasi);
function kilit_noktasi(event:MetadataEvent):void
{
    // Özel kodunuzu başlatın
    // Bu parçacık kodu, kilit noktasının adını Çıktı panelinde görüntüler

    // Özel kodunuzu sonlandırın
}
```

- *Video Oynatmak için Tıklat*

Sahneye eklenen bir video(FLVPlayback) bileşeninin yine sahnede oluşturulan bir buton yardımıyla başlatılması için kullanılır.



FLVPlayback bileşen
örneğinin özelliklerine
bir video kaynak
dosyası atadığınızdan
emin olun.

```
/* Aşağıdaki video_instance_name ögesini, videoyu oynatmak istediğiniz
FLVPlayback bileşeninin örnek adıyla değiştirin.
Sahne alanı üzerindeki FLVPlayback video bileşeninin belirtilen örneği oynatılır. */
buton.addEventListener(MouseEvent.CLICK, video_oynat);

function video_oynat(event:MouseEvent):void
{
    // video_instance_name ifadesini video bileşeninizin örnek adıyla değiştirin
    video_ornek_ad.play();
}
```

- *Videoyu Duraklatmak için Tıklat*

Sahneye eklenen bir video(FLVPlayback) bileşeninin yine sahnede oluşturulan bir buton yardımıyla duraklatılması için kullanılır.

```
/* Talimatlar:
1. Aşağıdaki video_instance_name ögesini, duraklatmak istediğiniz FLVPlayback
bileşeninin örnek adıyla değiştirin.
*/
button_1.addEventListener(MouseEvent.CLICK, video_duraklat);

function video_duraklat(event:MouseEvent):void
{
    // video_instance_name ifadesini video bileşeninizin örnek adıyla değiştirin
    video_ornek_ad.pause();
}
```

- *Videoyu Geri Sarmak için Tıklatın*

Sahneye eklenen bir video(FLVPlayback) bileşeninin yine sahnede oluşturulan bir buton yardımıyla başa sarılması için kullanılır.

```
Talimatlar:
1. Aşağıdaki video_instance_name ögesini, geri sarmak istediğiniz FLVPlayback
bileşeninin örnek adıyla değiştirin.
*/

button_1.addEventListener(MouseEvent.CLICK, yeniden_baslat);
```

```
function yeniden_baslat(event:MouseEvent):void
{
    // video_instance_name ifadesini video bileşeninizin örnek adıyla değiştirin
    video_instance_name.seek(0);
}
```

- *Video Kaynağını Ayarlamak İçin Tıklat*

Sahneye eklenen bir video(FLVPlayback) bileşeninin yine sahnede oluşturulan bir buton yardımıyla kaynağını değiştirmek için kullanılır. Bu sayede video oynatıcıda yeni bir video oynatılır.

```
/* Talimatlar:
1. Aşağıdaki video_instance_name ifadesini video bileşeninizin örnek adıyla
değiştirin
2. Aşağıdaki "http://www.helpexamples.com/flash/video/water.flv" adresini,
oynatmak istediğiniz yeni video dosyasının adresiyle değiştirin. Tırnak işaretlerini
koruyun (""").
*/

button_1.addEventListener(MouseEvent.CLICK, kaynak_sec);

function kaynak_sec(event:MouseEvent):void
{
    video_instance_name.source =
"http://www.helpexamples.com/flash/video/water.flv";
}
```

- *İşaret Noktasına Arama Yapmak İçin Tıklat*

Sahneye eklenen bir video(FLVPlayback) bileşeninin yine sahnede oluşturulan bir buton yardımıyla istenilen bir kilit noktadan başlatılması için kullanılır. Bir nevi video içerisinde oluşturulan kilit noktalara geçiş linki verilir.

```
/* Talimatlar:
1. Aşağıdaki video bileşeninizin örnek adıyla değiştirin
2. Aşağıdaki "Cue Point 1" ifadesini geçiş yapılacak kilit noktasının adıyla değiştirin.
Tırnak işaretlerini koruyun (""").
*/

button_1.addEventListener(MouseEvent.CLICK, fl_ClickToSeekToCuePoint_2);

function fl_ClickToSeekToCuePoint_2(event:MouseEvent):void
{
    // video_instance_name ifadesini video bileşeninin örnek adıyla değiştirin.
```

```
    // "Kilit Noktası 1" ifadesini aranacak kilit noktasının adıyla değiştirin.
    var cuePointInstance:Object = video_instance_name.findCuePoint("Kilit Noktası
1");
    video_instance_name.seek(cuePointInstance.time);
}
```

- **NetStream Video Oluştur**

Sahne üzerinde herhangi bir video oynatıcı (FLVPlayback) bileşenini kullanmadan sahne alanı üzerinde video görüntülemek için kullanılır.



Net Stream Sınıfı çalışmak için video oynatıcısına ihtiyaç duymaz.

/* Talimatlar:

1. Adobe Flash Media Server 2 gibi bir akış sunucusu üzerindeki bir video dosyasına bağlanıyorsanız aşağıdaki 'null' değerini video dosyasının URL adresiyle değiştirin. URL adresini tırnak işareti ("") içine alın.
2. Yerel olan veya akış sunucusunu kullanmayan bir video dosyasına bağlanıyorsanız, 'null' değerini aşağıda olduğu gibi bırakın.
3. "http://www.helpexamples.com/flash/video/water.flv" adresini, oynatmak istediğiniz videonun URL'siyle değiştirin. Tırnak işaretlerini koruyun ("").

*/

```
var fl_NC:NetConnection = new NetConnection();
fl_NC.connect(null);
// bir bağlantı başlatır; Flash Media Server kullanılmıyorsa null kullanılır
var fl_NS:NetStream = new NetStream(fl_NC);
fl_NS.client = {};
var fl_Vid:Video = new Video();
fl_Vid.attachNetStream(fl_NS);
addChild(fl_Vid);
fl_NS.play("http://www.helpexamples.com/flash/video/water.flv");
```

OLAY İŞLEYİCİLER GRUBU

Bu grupta bulunan kod parçacıkları butonların olaylarını tanımlamak için kullanılır. Bir butona tıklandığında, fare butonun üzerine geldiğinde veya belirlenen bir tuşa basıldığında yapılmasını istediğimiz eylemleri tanımlamak için kullanılır.

- **Fare Tıklatma Olayı**

Belirtilen sembol üzerine fare ile tıklatıldığında çalışmasını istediğiniz kodlar için bu kod parçacığı kullanılır.

/* Talimatlar:

1. Aşağıda "// Özel kodunuzu başlatın" yazan satırdan sonra gelen yeni bir satıra özel kodunuzu ekleyin.
- Sembol örneği tıklatıldığında kod yürütülür.

*/

```
Belirlenen_sembol.addEventListener(MouseEvent.CLICK, tiklayınca_calis);
```

```
function tiklayınca_calis(event:MouseEvent):void
{
    // Özel kodunuzu bu alana yazınız.
    // Bu örnek kod, Çıktı panelinde "Fare Tıklatıldı" sözcüklerini görüntüler.
    trace("Fare tıklatıldı");
}
```

- **Fareyi Üzerine Getirme Olayı**

Farenin belirtilen sembol üzerine getirildiğinde çalışmasını istediğiniz kodlar için bu kod parçacığı kullanılır.

```
/* Talimatlar:
1. Aşağıda "// Özel kodunuzu başlatın" yazan satırdan sonra gelen yeni bir satıra
özel kodunuzu ekleyin.
Fare sembol örneğinin üzerine getirildiğinde kod yürütülür.
*/
movieClip_1.addEventListener(MouseEvent.CLICK, fare_ustune_gelince);
function fare_ustune_gelince(event:MouseEvent):void
{
    // Özel kodunuzu başlatın
    // Bu örnek kod, Çıktı panelinde "Fare üzerine getirildi" sözcüklerini görüntüler.
    trace("Fare üzerine getirildi");
    // Özel kodunuzu sonlandırın
}
```

- **Fareyi Üzerinden Çekme Olayı**

Farenin belirtilen sembol üzerinden çekildiği an çalışmasını istediğiniz kodlar için bu kod parçacığı kullanılır.

Yukarıdaki kodlar ile kullanımı aynıdır. Yalnız **MOUSE_OVER** deyimini yerine **MOUSE_OUT** deyimini kullanılmalıdır.

- **Tuşa Basıldı Olayı**

Klavyeden herhangi bir tuşa basıldığında aşağıda tanımlanan

fl_KeyboardDownHandler fonksiyonu çalıştırılır. Çalışması istenilen kodlar bu fonksiyon içerisine yazılır.

```
/* Talimatlar:
1. Aşağıda "// Özel kodunuzu yazın" yazan satırdan sonra gelen yeni bir satıra özel
kodunuzu ekleyin. Herhangi bir tuşa basıldığında kod yürütülür.
*/
stage.addEventListener(MouseEvent.CLICK, fl_KeyboardDownHandler);

function fl_KeyboardDownHandler(event:MouseEvent):void
{
    // Özel kodunuzu yazın
}
```

- **Kareye Girme Olayı**

Animasyon, zaman çizelgesinde istenilen bir kareye gittiğinde aşağıda tanımlanan **fl_EnterFrameHandler** fonksiyonu yürütülür.

```

Talimatlar:
1. Aşağıda "// Özel kodunuzu yazın" yazan satırdan sonra gelen yeni bir satıra özel kodunuzu ekleyin.
*/
addEventListener(Event.ENTER_FRAME, fl_EnterFrameHandler);

function fl_EnterFrameHandler(event:Event):void
{
    // Özel kodunuzu yazın
}

```

TAŞINABİLİR AYGIT DOKUNMA OLAYLARI

Günümüzde teknolojinin hızla ilerlemesi bilgiye erişim kaynaklarını da etkiledi. Pek çok bilgisayar kullanıcısı artık taşınabilir aygıtları tercih etmeye başladı. Böylece animasyonları da hazırlarken bu cihazlara yönelik eylemlerin yapılmasına ihtiyacı oluştu.

Bu grupta bulunan kod parçacıkları sayesinde bir mobil cihazın dokunma, taşıma gibi eylemlerini kontrol edebilir ve bunlara işlemler tanımlayabilirsiniz.

- *Hafif Dokunma Olayı*

Sahne de bulunan sembole hafif bir şekilde dokunulduğunda yapılmasını istediğimiz eylemleri tanımlamak için kullanılabilir.

```

/* Talimatlar:
1. Aşağıda "// Özel kodunuzu başlatın" yazan satırdan sonra gelen yeni bir satıra özel kodunuzu ekleyin.
*/
Multitouch.inputMode=MultitouchInputMode.TOUCH_POINT;
movieClip_1.addEventListener(TouchEvent.TOUCH_TAP, fl_TapHandler);
function fl_TapHandler(event:TouchEvent):void
{
    // Özel kodunuzu başlatın
}

```

- *Dokun ve Sürükle Olayı*

Sahne de bulunan bir sembolün, tutularak ve sürüklenerek taşınmasını sağlar.

```

Multitouch.inputMode=MultitouchInputMode.TOUCH_POINT;
movieClip_1.addEventListener(TouchEvent.TOUCH_BEGIN, fl_TouchBeginHandler);

```

```

movieClip_1.addEventListener(TouchEvent.TOUCH_END, fl_TouchEndHandler);
var fl_DragBounds:Rectangle = new Rectangle(0, 0, stage.stageWidth,
stage.stageHeight);
function fl_TouchBeginHandler(event:TouchEvent):void
{
    event.target.startTouchDrag(event.touchPointID, false, fl_DragBounds);
}
function fl_TouchEndHandler(event:TouchEvent):void
{
    event.target.stopTouchDrag(event.touchPointID);
}

```

- **Uzun Basma**

Sahne de bulunan bir sembole bir saniye boyunca bastığınızda özel kodunuzu içeren bir fonksiyon yürütülür. Bu eylemi, o sembole yönelik bir sağ tuş menüsünü görüntülemek için kullanabilirsiniz.

```

/* Talimatlar:
1. Aşağıda "// Özel kodunuzu başlatın" yazan satırlardan sonra gelen yeni bir satıra
özel kodunuzu ekleyin.
*/
var fl_PressTimer:Timer = new Timer(1000);
fl_PressTimer.addEventListener(TimerEvent.TIMER, fl_PressTimerHandler);
function fl_PressTimerHandler(event:TimerEvent):void
{
    // Özel kodunuzu başlatın
}
Multitouch.inputMode=MultitouchInputMode.TOUCH_POINT;
movieClip_1.addEventListener(TouchEvent.TOUCH_BEGIN, fl_PressBeginHandler);
movieClip_1.addEventListener(TouchEvent.TOUCH_END, fl_PressEndHandler);
movieClip_1.addEventListener(TouchEvent.TOUCH_OUT, fl_PressEndHandler);
movieClip_1.addEventListener(TouchEvent.TOUCH_ROLL_OUT,
fl_PressEndHandler);
function fl_PressBeginHandler(event:TouchEvent):void
{
    fl_PressTimer.start();
}function fl_PressEndHandler(event:TouchEvent):void
{
    // Özel kodunuzu başlatır.
}

```

TAŞINABİLİR HAREKET OLAYLARI

Taşınabilir cihazlar için ekrana dokunma türleri ve türlere göre çalıştırılacak fonksiyonlar tanımlamada kullanılır.

- *İki Parmakla Hafif Dokunma Olayı*

Sahne alanına iki parmağınızla hafifçe dokunduğunuzda kendi özel kodunuzu ekleyebileceğiniz bir fonksiyon yürütülür.

```
/* Talimatlar:
1. Aşağıda "// Özel kodunuzu başlatın" yazan satırdan sonra gelen yeni bir satıra özel kodunuzu ekleyin.
Sahneye iki parmağınızla hafifçe dokunduğunuzda kod yürütülür.
*/
Multitouch.inputMode=MultitouchInputMode.GESTURE;
stage.addEventListener(GestureEvent.GESTURE_TWO_FINGER_TAP,
fl_TwoFingerTapHandler);
function fl_TwoFingerTapHandler(event:GestureEvent):void
{
    // Özel kodunuzu başlatın
}
```

- *Yakınlaştırmak İçin Çimdikle Olayı*

Sahne alanının çimdiklenmesi, seçili nesneyi yakınlaştırır.

```
/* Talimatlar:
1. Sembol, uygulamanızın arka planı olmalıdır ve sahne alanını kaplayacak şekilde boyutlandırılmalıdır.
*/
Multitouch.inputMode=MultitouchInputMode.GESTURE;
stage.addEventListener(TransformGestureEvent.GESTURE_ZOOM,
fl_ZoomHandler);
function fl_ZoomHandler(event:TransformGestureEvent):void
{
    movieClip_1.scaleX *= event.scaleX;
    movieClip_1.scaleY *= event.scaleY;
}
```



Sembolün kayıt ve dönüştürme noktaları merkezde olmalıdır.

- *Yatay Kaydırma Olayı*

Sahne alanında seçili nesneyi yatay kaydırır. Yatay kaydırılacak nesne genellikle görüntülenebilir alandan daha büyüktür.

```
/* 1. Sahne alanındaki tüm nesneleri yatay kaydırmak için nesneleri tek bir kap film klibine yerleştirin ve bu parçacığı o film klibine uygulayın.
*/
Multitouch.inputMode=MultitouchInputMode.GESTURE;
movieClip_1.addEventListener(TransformGestureEvent.GESTURE_PAN,
fl_PanHandler_2);
```

```
function fl_PanHandler_2(event:TransformGestureEvent):void
{
    event.currentTarget.x += event.offsetX;
    event.currentTarget.y += event.offsetY;
}
```



Sembolün dönüştürme
ve kayıt noktaları
merkezde olmalıdır.

- **Döndürme Olayı**

Genellikle iki parmak saat yönünde veya saat yönünün aksine döndürülerek seçili nesneyi döndürür.

```
Multitouch.inputMode=MultitouchInputMode.GESTURE;
movieClip_1.addEventListener(TransformGestureEvent.GESTURE_ROTATE,
fl_RotateHandler);
function fl_RotateHandler(event:TransformGestureEvent):void
{
    event.target.rotation += event.rotation;
}
```

- **Kaydırma Olayı**

Sahne alanını kaydirdığınızda özel kodunuzu içeren bir işlev yürütülür. Bir Metin Alanı'ndaki metni kaydırmak veya uygulamanızdaki durumları değiştirmek için kullanılabilir.

TAŞINABİLİR AYGIT İŞLEMLERİ

- **Sonraki/Önceki Kareye Gitmek ve Durdurmak için Kaydırın**

Sahne alanını kaydirdığınızda animasyon sonraki veya önceki kareye gider ve o karede durdurulur.

- **Sonraki/Önceki Kareye Gitmek ve Oynatmak için Kaydırın**

Sahne alanını kaydirdığınızda animasyon sonraki veya önceki kareye gider ve o kareden çalışmaya devam eder.

- **İvmeölçer ile kaydır**

Nesnenin, taşınabilir aygıtın eğilmesiyle hareket ettirilmesini sağlar.

```
/* Hareket miktarını artırmak veya azaltmak için aşağıdaki 30 değerinin yerine
sembolün taşınabilir aygıt eğildiğinde hareket etmesini istediğiniz piksel sayısını girin.
30 sayısının aşağıdaki kodda iki kez görüldüğünü unutmayın. */
var fl_Accelerometer:Accelerometer = new Accelerometer();
fl_Accelerometer.addEventListener(AccelerometerEvent.UPDATE,
fl_AccelerometerUpdateHandler);
function fl_AccelerometerUpdateHandler(event:AccelerometerEvent):void
{
    movieClip_1.x -= event.accelerationX*30;
    movieClip_1.y += event.accelerationY*30;
}
```

- **Devre Dışı Bırakma/Etkinleştirme Olayı**

Uygulama taşınabilir aygıtta pasif durumdayken ENTER_FRAME ve TIMER olayları gibi fazla kaynak kullanan işlemleri askıya alarak işlemci ve pil ömrünü korur.

- **Menü Tuşuna Basıldı Olayı**

Aygıtın Menü tuşuna basıldığında istenilen eylemlerin yapılmasını sağlar. Seçenekler menüsünü görüntülemek için kullanışlıdır.

TAŞINABİLİR AYGITLAR İÇİN AIR

Air teknolojisi sayesinde taşınabilir cihazınızın dış dünya ile bağlantısını kurabilirsiniz.



Coğrafi konum kullanımı için Dosya -> AIR for Android Ayarları'nda "ACCESS_FINE_LOCATION" ve "ACCESS_COARSE_LOCATION" izinlerini etkinleştirin.

- **Coğrafi Konumu Göster**

Taşınabilir aygıt üzerinde mevcut konumun eylem ve boylam bilgilerini almak için kullanılır. Mevcut kod parçacığı bilgileri bir metin alanında gösterir.

- **Aygıt Döndürüldüğünde Nesnelerin Yönünü Yeniden Ayarla**

Aygıt döndürüldüğünde nesnenin/nesnelerin yönünü yeniden ayarlar. Belirli nesneler uygulamanın geri kalanından farklı olarak yönlendirilebilir. Bu özellik, bir nesnenin geçerli dönüşünü korumasını veya uygulamadan bağımsız şekilde dönmesini sağlamak için kullanılabilir. Bu özelliğin kullanımı için *Dosya -> AIR for Android Ayarları ögesinde "Otomatik yönlendirme"* seçeneğini aktifleştirilmelidir.



Özet

- Adobe Flash CC ile ister masaüstü isterse taşınabilir aygıtlara yönelik animasyonlar geliştirmek mümkündür. Animasyonlar her ne kadar görsel çalışma olarak düşünülse de tasarım kadar arka planda çalışacak olan kodlamalarda önemlidir.
- Bazen kod yazmak animasyonu tasarlamaktan daha zor ve sıkıcı olmaktadır. Bu ünite de bir animasyonda en çok kullanılan kod parçacıklarına ve dış kaynaklardan nasıl veril çekildiğine yer verilmiştir.
- Kod parçacıkları özel olarak tanımlanmış olup sadece belirli düzeltmeler ile tasarımcıyı derin kod karmaşasından kurtarmayı amaçlamış yapılardır. bu yapılarda gerekli olan düzenlemeler kitabımızda açıklanmıştır fakat bunun yanı sıra benzer açıklamalara programın kod panelinde de rastlayabilirsiniz.
- Animasyonlar her zaman mevcut bilgileri kullanmazlar. Bazen dış kaynaklar ile bilgi alış verişi yapmak gerekir. Bu durumda action script 3.0 dili için geçerli olan sınıfları kullanmak gerekir.
- URL Request sınıfı ile iletişim kurulacak adres belirlenirken, URLRequest ile adreste bulunan kaynağın verileri animasyona indirilir.
- İndirilen bilgiler her zaman tek tip olmayabilir. Bazen bu bilgilerin gruplanması gerekir. Bu durumda da farklı bilgi türleri için değişkenler tanımlanı ve bu işlem içinde URLVariables sınıfından faydalanılır.

DEĞERLENDİRME SORULARI

- I. Type parametresi dinleyicinin olay türünü belirlemek için kullanılır.
 - II. Olay gerçekleştiğinde çalışacak kodlar ise listener parametresi ile tanımlanır.
 - III. Listener parametresinde bir sınıf veya bir fonksiyonu çağırabilirsiniz.
1. Yukarıdaki özelliklerden hangisi ya da hangileri *addEventListener* sınıfı için doğrudur?
 - a) Yalnız I
 - b) Yalnız II
 - c) I ve II
 - d) II ve III
 - e) I, II ve III
 2. Aşağıdakilerden hangisi *URLRequest* sınıfının iletişim türünü belirlemek için kullanılan özelliktir?
 - a) *contentType*
 - b) *data*
 - c) *digest*
 - d) *method*
 - e) *requestHeaders*
 3. Aşağıdakilerden hangisi *URLLoader* sınıfının indirme durumunu bildiren olaydır?
 - a) *complete*
 - b) *httpResponseStatus*
 - c) *httpStatus*
 - d) *open*
 - e) *progress*
 4. Aşağıdakilerden hangisi bir verinin sadece indirilen kısmının kullanılmasına imkân veren sınıftır?
 - a) *URLStream*
 - b) *URLLoader*
 - c) *URLVariables*
 - d) *URLRequest*
 - e) *AddEventListener*
 5. Aşağıdakilerden hangisi eylemler grubundaki kod parçacıkları için yanlıştır?
 - a) Fare imleci istendik bir nesne ile değiştirilebilir.
 - b) Nesneler istendiği zaman gizlenebilir.
 - c) Gizli olan nesneleri tekrar görüntülemek mümkün değildir.
 - d) Nesnelerin sahnedeki sıralaması değiştirilebilir.
 - e) Rastgele bir sayı üretilebilir.

6. Aşağıdakilerden hangisi bir videoda belirli saniyelere görev tanımlamak için kullanılan deyimdir?
- a) CuePoint
 - b) MouseEvent
 - c) MetaDataEvent
 - d) SoundChannel
 - e) SoundMixer
7. *KeyboardEvent* olayı aşağıdakilerden hangisine olay tanımlamak için kullanılır?
- a) Taşınabilir Aygıtlar
 - b) Klavye Tuşları
 - c) Fare
 - d) Hoparlörler
 - e) Dokunmatik Ekranlar
8. Aşağıdaki deyimlerden hangisi taşınabilir aygıtların olaylarını tanımlamak için kullanılır?
- a) MouseEvent
 - b) KeyboardEvent
 - c) MultitouchInputMode
 - d) TouchEvent
 - e) TimerEvent
9. Taşınabilir Aygıtlarda dokunma eyleminin parmak sayısını belirleyen deyim aşağıdakilerden hangisidir?
- a) GestureEvent
 - b) MouseEvent
 - c) KeyboardEvent
 - d) MultitouchInputMode
 - e) TouchEvent

- I. İvmeölçer ile kaydır kod parçacığı sahnedeki nesneyi taşınabilir aygıtın eğim yönüne göre hareket ettirir.
 - II. NetStream sınıfı sahneye video eklerken video oynatıcısını kendisi oluşturur.
 - III. Sahneye eklenen bir video oynatıcının kaynağı AS3 dilinde “source” kodu ile değiştirilir.
10. Yukarıdaki bilgilerden hangisi ya da hangileri animasyonlar için doğrudur?
- a) Yalnız I
 - b) Yalnız II
 - c) I ve II
 - d) II ve III

YARARLANILAN KAYNAKLAR

Adobe Flash Pro CC, Yardımcı el kitabı. <http://helpx.adobe.com/tr/support>

[Erişim tarihi: 01 MART 2014].

http://www.flashdersleri.com/html_dosyalar/veri_yukleme.html

[Erişim tarihi: 10 MART 2014].

<http://www.turkcescript.com/>

[Erişim tarihi: 05 MART 2014].

EK - İLETİŞİM FORMU

Ünitenin başında anlatılan sınıfların daha iyi anlaşılması amacıyla bir iletişim formu oluşturalım. Bu form sayesinde sitenizi ziyaret eden kullanıcıların size e-mail göndermesi sağlanabilir.

Bu iletişim esnasında kullanıcıdan ad, soyad, mail adresi ve göndermek istediği mesaj bilgileri alınır. Bu bilgilerin kullanıcıdan alınması için sahneye 4 adet giriş metni(input text) eklenir. İletişimin durum bilgisini kullanıcıya vermek için sahneye 1 adet dinamik metin eklenir. İşlemleri yapmak için 2 adet buton eklenir. Daha sonra nesnelerin örnek adları şekil 14,2'deki gibi düzenlenir.

Şekil 14.2 İletişim formu ekran görüntüsü ve nesnelerin örnek adları

Form ekranı tasarlandığına göre artık gerekli olan kodlamalar yapılabilir. Bu kodlamaların yapılması için yeni bir katman oluşturalım ve bu katmanın adını action olarak tanımlayalım. Action katmanının ilk karesine aşağıdaki kodları yazalım. Kodlamaya ilk başladığımızda her olasılığa karşılık metin nesnelerinin içeriğini boşaltalım. Daha sonra tab tuşu kullanılarak metin kutuları arasındaki geçişlerin sıralamasını ayarlayalım.

```
isim_txt.text="";
mail_txt.text="";
konu_txt.text="";           // Sahnedeki metin alanlarının içeriği silinir.
mesaj_txt.text="";
uyari_txt.text="";
isim_txt.tabIndex=1;
mail_txt.tabIndex=2;
konu_txt.tabIndex=3;       // Bu kodlarla tab tuşu sıralaması yapılır.
mesaj_txt.tabIndex=4;
```

Sahneye eklenen butonların etiketlerini düzenleyelim.

```
temizle_btn.label="Temizle";  
gonder_btn.label="Gönder";
```

Animasyon başlatıldığında fare imlecinin ilk metin alanı olan isim_txt nesnesinde olması istenir. Bu durumda imleci ilgili metin kutusuna almak için aşağıdaki kod kullanılır.

```
stage.focus=isim_txt;
```

Metin kutularına girilecek olan bilgilerin karakter sayısının sınırlandırılması mümkündür. Bunun için *maxchars* ifadesi kullanılır.

```
isim_txt.maxChars=30;  
mail_txt.maxChars=40;  
konu_txt.maxChars=100;  
mesaj_txt.maxChars=500;
```

Mesajlarda Türkçe karakter sorununu gidermek için yukarıdaki kod kullanılabilir.

```
System.useCodePage=true;
```

Artık butonlarımızın olaylarını tanımlamaya başlayabiliriz. Temizle butonuna tıklandığı zaman metin kutularının içeriğinin silinmesini isteriz. Bunu yapmak için aşağıdaki kodları kullanabiliriz.

```
temizle_btn.addEventListener(MouseEvent.CLICK, formuTemizle);  
  
function formuTemizle(e:MouseEvent) {  
    isim_txt.text="";  
    mail_txt.text="";  
    konu_txt.text="";  
    mesaj_txt.text="";  
    uyarı_txt.text="";  
    stage.focus=isim_txt;  
}
```

Şimdi de eksik kalan ve en önemli kodlama olan gönder butonunun kodlarını yazalım. Yalnız bilgileri direkt göndermeye çalışmadan önce bilgilerin doğruluğunun kontrol etmemiz lazım. Doğrulamaktan kastımız eksik bir bilgi var mı onu kontrol etmektir. Mail adresinin de belirli bir formatta yazılmış olması lazım.

Bilgiler eksiksiz ve mail adresi uygun bir formatta ise artık bilgilerin php dosyasına gönderilmesi için herhangi bir eksik kalmamış demektir.

Öncelikle gönder butonunun olaylarını dinleyelim ve metin kutularının içeriklerini kontrol edelim sonrasında ise mail adresini sorgulayalım.



Türkçe karakter kodu eylem panelinde en üst satıra yazılmalıdır.

```

gonder_btn.addEventListener(MouseEvent.CLICK, mesajGonder);

function mesajGonder(e:MouseEvent) {
    if (isim_txt.text==" " || mail_txt.text==" " || konu_txt.text==" " || mesaj_txt.text==" ")
    {
        uyari_txt.text="Lütfen tüm alanları eksiksiz doldurun.";
    } else if (mail_txt.text.indexOf("@")==-1 || mail_txt.text.indexOf(".")==-1) {
        uyari_txt.text="Mail adresi hatalı. Lütfen kontrol edin.";
    } else {
        var adres:URLRequest=new URLRequest("gonder.php");
        var formBilgileri:URLVariables=new URLVariables();
        var phpLoader:URLLoader=new URLLoader();

        adres.method=URLRequestMethod.POST;

        formBilgileri.isim=isim_txt.text;

        formBilgileri.mail=mail_txt.text;

        formBilgileri.konu=konu_txt.text;

        formBilgileri.mesaj=mesaj_txt.text;

        adres.data=formBilgileri;

        uyari_txt.text="Mesajınız gönderiliyor.";

        phpLoader.load(adres);

        phpLoader.addEventListener(Event.COMPLETE, mesajGonderildi);
    }
}

```

Yukarıdaki kodlarda mail adresinin kontrolü esnasında *indexOf* deyimi kullanılmıştır. Bir mail adresi standart olarak "@" sembolü ve alan adını uzantısı için "." işaretine ihtiyaç duyar. Kullanıcının girdiği mail adresinde bu karakterlerin olup olmadığı *indexOf* deyimi ile tesbit edilir. *indexOf* deyimi, metin kutusunda aradığımız karakterler mevcut ise 0, mevcut değilse -1 değerlerini geriye dönderir.

Bütün bilgiler kontrol edildikten sonra artık bilgilerin gönderilmesi gerekir. Öncelikle birden fazla değer olduğu için değerleri taşıyacak bir *URLVariables* sınıfı tanımlanır. Bu sınıfa her bir veri bir özellik olarak atanır.

Elde edilen verilerin iletişim kuracağı dosya belirlenmelidir. Bunun için iletişim sınıfı olan *URLRequest* kullanılır. *URLRequest* sınıfının metodlarından *POST* metodu seçilmiştir. Çünkü iletişim veri gönderimi şeklinde olacaktır. *URLRequest* sınıfının *data* özelliğine oluşturulan *URLVariables* sınıfı değer olarak atanır.

Artık veriler ve iletişim kurulacak dosya belirlendiğine göre bilgilerin belirli bir düzene gönderilmesi gerekir. Bu işlemi yapması için *URLLoader* sınıfı kullanılabilir.

Bütün bu kodlamalardan sonra bilgiler artık *gonder.php* dosyasına iletildi. Yalnız iletişim ile ilgili durum raporu almak gerekir. Bunun için yukarıdaki kodlarda kullanılmış olan *mesajGonderildi* fonksiyonunu tanımlamak gerekir.

```
function mesajGonderildi(e:Event)
{
    if(e.target.data=="mesajGonderildi")
    {
        uyarı_txt.text="Mesajınız Gönderildi";
    }
    else
    {
        uyarı_txt.text="Gönderme sırasında bir hata oluştu. Lütfen tekrar deneyin.";
    }
}
```

Flash kısmı bittiğine göre artık php dosyasını hazırlamak lazım. Not defterini açalım ve aşağıdaki kodları not defterine yapıştıralım. Sonrasında ise aşağıdaki kodlarda *"mail adresinizi buraya yazın."* kısmına kendi mail adresinizi ekleyin ve *gonder.php* adıyla animasyonun bulunduğu klasöre kaydedin.

```
<?php
$to = "Mail adresinizi buraya yazın.";
$subject = $_POST["konu"];
$headers = "From: " . $_POST["isim"] . "\r\n";
$headers .= "<".$_POST["mail"].">";
$headers .= "Reply-To: " . $_POST["mail"] . "\r\n";
$message = $_POST["mesaj"];
if( mail($to, $subject, $message, $headers) ){
    echo "mesajGonderildi";
}else if (!mail($to, $subject, $message, $headers)){
    echo "hata";
}
?>
```