



Ankara Üniversitesi
Nallıhan Meslek Yüksekokulu

Actionscripts 3.0

NBP108 - GRAFİK ANİMASYON II
Öğr.Gör. Salih ERDURUCAN

ACTIONSCRIPT 3.0'A GİRİŞ:

Değişken Nedir?

Değişken basit tanımıyla bizim belirlediğimiz isimlerle oluşturulmuş bilgisayar hafızasında yer kaplayan alanlardır. Bir değişken bir isim ve veri tipinden oluşmaktadır.

Değişken Tanımlamak

Flash ta number, string ,boolean vs. veri tiplerinden birini kullanarak değişkenler şu şekilde oluşturulur;

```
var x:Number=5;
```

Yukarıda görüldüğü gibi ismi x olan Number veri tipinde bir değişken tanımladık. Buradaki önemli noktalar, değişken tanımlarken “var” ifadesi sonra oluşturacağımız değişkenin ismi, veri tipi ve son olarak noktalı virgül.

Number veri tipindeki x değişkenimizin değeri 5 tir bu değeri şu şekilde de verebilirdik:

```
var x:Number;
```

```
x=5;
```

Number, Int Ve Uint Veri Tipleri

Bilgisayarımızın hafızasında rakam saklamak istediğinizde Number, int veya uint veri tiplerinden bir tanesini kullanabiliriz. Hepsi rakam saklıyor ise niye üç tane? Üç tane olmasının sebebi alacağı değerlere göre değişiklik göstermeleridir.

Number veri tipi; aklımıza gelebilecek tüm rakamları pozitif, negatif veya ondalıklı sayıları hafızada tutabilen veri tipidir.

Int veri tipi; Number veri tipinden farklı olarak sadece pozitif ve negatif tamsayıları hafızada tutabilen veri tipidir.

Uint veri tipi; sadece pozitif tamsayıları hafızada tutabilen veri tipidir.

Örneğin;

```
var number:Number= -1.3; // Number: Pozitif, negatif ve ondalıklı sayılar
```

```
var number2:int= -1; // Int: Pozitif ve negatif tamsayılar
```

```
var number3:uint= 1; // Uint: Pozitif tamsayılar
```

Şöyle bir soru akla gelebilir: int yerine Number veri tipini kullanamaz mıydık?

Kullanabilirdik ama gereksiz yer kaplayan bir işlem yapmış olurduk. Bu olayı şu örneğe benzetebiliriz; elimizde 1 lt su var biz bunu 1 lt kapasitesi olan int veri tipi yerine 2 lt kapasitesi olan Number veri tipine koyuyoruz. Sonuç olarak su Number veri tipine sığar ama 1 lt yer boş kalır gereksiz bir işlem yapmış oluruz.

Boolean Veri Tipi:

Merhaba arkadaşlar bu yazımda Flash'ta kullanılan veri tiplerinden bahsedeceğim.

Boolean veri tipi daha çok mantıksal işlemler için kullanılan veri tipidir. İçerisinde (**Evet**, **Hayır**), (**Doğru**, **Yanlış**), (**1**, **0**) gibi mantıksal karşılaştırmayı yapabileceğimiz veriyi saklar.

Boolean veri tipinin alabileceği iki adet değer vardır. Bu değerler dışında herhangi bir değer alamaz. Bu değerler "**True**" ve "**False**"dir.

Geliştirdiğimiz uygulamanın içinde karşılaştırma gibi işlemler yaparsak veri tipini **Boolean** yapmamız doğru bir karar olacaktır.

Örneğin;

```
var veritipi:Boolean=true; // Bu şekilde yazmamız programımızda doğru, evet ve rakamsal olarak 1 anlamına gelir.
```

`var veritipi:Boolean=false;` // Bu şekilde yazarsak programımızda yanlış, hayır ve 0(sıfır) anlamına gelir. Kısaca true yazdığımızda aldığı değerlerin zıddını false de alır.

String Veri Tipi

String veri tipi uygulama içerisinde düz metinleri barındıran veri tipidir. String veri tipinde bir değişken oluştururken metni çift tırnak içerisinde yazmaya dikkat etmemiz gerekir.

```
var metin:String=""; //boş
```

```
var metin:String="düzmetin";
```

```
trace(metin) //output panelinde çıkacak sonuç=düzmetin
```

TRACE() KOMUTU

trace komutu uygulama geliştirirken sıkça kullanacağımız, kullanışı oldukça basit olan komutlardan birisidir. trace komutu, yaptığımız uygulamanın swf'sini çalıştırdığımızda çalışmayacaktır.

Bu komutun amacı; uygulama geliştirirken programlama sürecinde bize veri kontrolü yapmamız için yardımcı olmaktır. Aldığı verileri Flash içerisinde yer alan **Output** panelinde satır satır yazdırır. Örneğin;

```
trace("Merhaba"); // çıkacak sonuç = Merhaba
```

“ “ işareti içinde yazılan değeri ActionScript 3.0 metin olarak görüyor.

```
var deneme:String="Merhaba-2";
```

```
trace(deneme); // çıkacak sonuç = Merhaba-2
```

trace içine **deneme** yazdık ama sonuç **Merhaba-2** çıktı. Çünkü üstte biz **Merhaba-2** metnini **deneme** isimli değişkene atadık ve burada “ ” işareti kullanmadık.

```
trace(deneme,"metin",deneme); // üstte yazsaydık, çıkacak sonuç = Merhaba-2 metin
```

Merhaba-2 olurdu. Farklı renklerde yazdım ki ayırt edilsinler birbirlerinden

ActionScript Olayları

ActionScript 3.0 ile artık sahnede gerçekleşen her şey, bir olay olarak kabul edilmiştir. Örneğin; farenin herhangi bir yere tıklaması, klavyeden bir şeyler yazılması, sahnenin çalışması, zaman çizgisi üzerinde bir frame'in çalışması gibi... Her şey artık olay olarak kabul edilmektedir. Olayların gerçekleşmesi yazılmış fonksiyonların tetiklenmesini sağlar. Her nesne için ayrı olaylar tanımlanabilir. Tanımlanan bu olaylar olay dinleyicileri ile izlenmektedir. Uygulamayı istendiği gibi geliştirmek için olayların çok iyi bilinmesi gerekir. ActionScript içerisinde kullanılacak birçok olay bulunmaktadır. ActionScript dilinin gücü de buradan gelmektedir. Bunlar fare olayları, klavye olayları, zamana bağlı olaylar olarak ayrılabilir. Burada temel olaylar işlenecektir. Bu olaylar sistemin otomatik tetiklediği ve kullanıcının tetiklediği olaylar olarak ikiye ayrılmaktadır.

“Neden onlarca olay dinleyicisi var?” sorusu akla gelebilir. İnternet üzerinden oyunlar oynanırken bazı durumlarda farenin tıklanması yetmeyecektir. Sürüklenmesi veya belli bir bölgede dolaştırılması ya da klavyeden belli bir tuşa/tuşlara basılması istenebilir. İşte bu yüzden gereken tüm durumlar için olay yöneticisi oluşturulabilir.

EventListener

Olay, dinleyicilerini sahneye eklemek ve tetikleyeceği fonksiyonu tanımlamak şu şekilde yapılır: Öncelikle olay dinleyicisini nesneye eklemek gerekir.

Kare örnek ismine sahip bir nesneye olay dinleyicisi eklemek için:

- Sahneye bir kare çiziniz ve kareyi film klibi haline dönüştürünüz.
- “Örnek Adı” (instance name) bölümüne *kare* ismini verin.
- Kare nesnesinin üzerine fareyle gelindiğinde fare fonksiyonu çalışması ve fare karenin üzerinde çıktısını vermesi isteniyor. Buna göre olay dinleyicisini ekleyiniz.

```
1 kare.addEventListener();  
2
```

Burada nesnemizi hangi olaya karşı tetikleceği belinmektedir. Farenin nesne üzerine gelmesi buradaki olaydır.

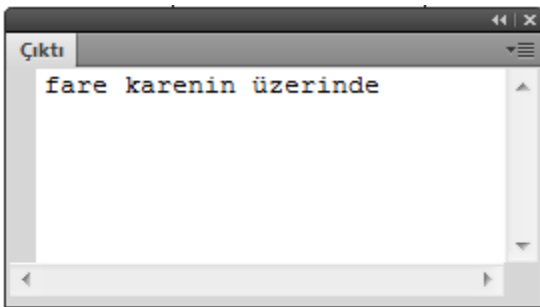
```
1 kare.addEventListener(MouseEvent.CLICK, fare);  
2
```

Fare karenin üstüne geldiğinde fare fonksiyonumuz çalışacaktır.

□ Fare fonksiyonunu yazın. Fonksiyona mouse olayı olduğunu anlatmak için parametre olarak *event:MouseEvent* söz dizimini kullanınız.

```
1 kare.addEventListener(MouseEvent.CLICK, fare);  
2 function fare(event:MouseEvent):void  
3 {  
4     trace("fare karenin üzerinde");  
5 }
```

Filmi test ediniz. Fare kare üzerine geldiğinde çıkış ekranında “fare karenin üzerinde” uyarısı görülmelidir.

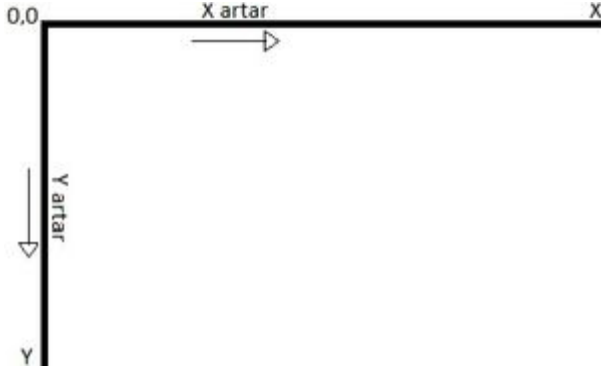


Uygulama içerisinde nesneye atanan olay iptal edilmek istenebilir. Örneğin, bir oyunda şart gerçekleşmişse nesne üzerindeki olayı silmek ya da değiştirmek gerekebilir. Bu gibi durumlarda *removeEventListener* komutunu kullanılarak nesne üzerindeki olaylar silinebilir. Yukarıda yaptığımız örnekteki olay dinleyicisini silmek için:

□ *kare.removeEventListener(MouseEvent.CLICK, fare);* komutunu kullanınız. Bu komutu bir fonksiyon ya da bir karar yapısının içerisinde kullanabilirsiniz.

Flash ActionScript 3.0 İle klavye kontrolü

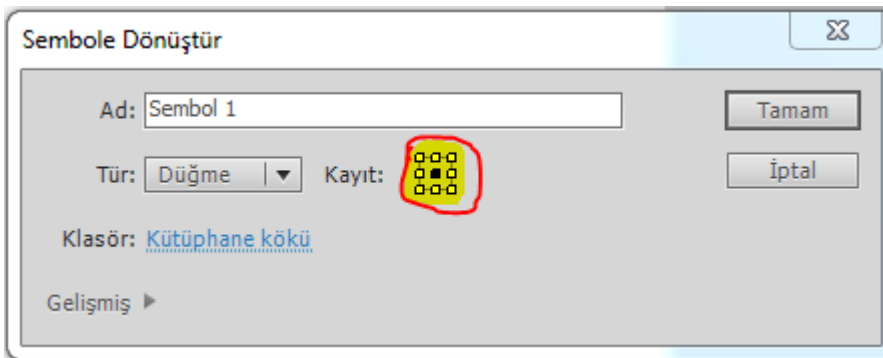
Bu uygulamamızda flash sahnesinde bulunan bir nesneyi klavye kullanarak hareket ettireceğiz. Bunu için nesnelerin koordinatları hakkında biraz bilgi vereceğim. Hepimiz matematikten az çok biliriz koordinat sistemini. Aşağıdaki bir görselle anlatacağım.



Sahnenin sol üst köşesi koordinat sisteminde (0,0) noktasıdır. (x,y) koordinat sistemine göre burada bulunan bir nesne $x=0$ ve $y=0$ koordinat değerindedir. Nesne sağa gittikçe x değeri, aşağı gittikçe ise y değeri artar. Sola gittikçe x değeri azalırken yukarı gittikçe de y değeri azalır. Bunu göz önüne alarak nesnelerin x ve y koordinatlarını değiştirdiğimizde nesne hareket eder. Bu bilgilerden sonra şimdi tasarıma geçelim.

Flash CS5 programını açarak actionscript 3 dosyası oluşturun. Sahneye çizim araç çubuğunu kullanarak bir şekil çizin. Ben sahneye bir daire çizdim.

Çizdiğiniz bu şekli film klibine çevireceğiz. Bunun için **F8** tuşuna basın ve türü **Film Klibi** seçin.



Burada dikkat edeceğimiz bir nokta var o da yukarıda işaretlediğim kızım. Kayıt kısmında 9 adet nokta verilmiş. Bunlardan hangisini seçerseniz nesnenin XY koordinatları o noktaya göre belirlenir. Biz bu uygulamamızda koordinat merkezi olarak ortayı seçtik. İsterseniz farklı noktayı da seçebilirsiniz. Bu aşamada çokta bir önemi yok aslında. İleride yapacağımız

uygulamalarda buna dikkat ederek işlerimizi yapacağız. Neyse geri derse dönelim. Tamamı tıkladıktan sonra nesnemiz artık film klibine dönüştü.

Şimdi nesnemizi seçiyoruz ve özellikler kısmında bulunan <Örnek Adı> kısmına nesnenin adını yazın. Nesneye ben TOP adını verdim. Artık kodlarını yazalım. Sahnede boş bir alanı tıklayın ve F9'a basın ve kodlama penceresini açın ve aşağıdaki kodları yazın.



```
1 stage.addEventListener(KeyboardEvent.KEY_DOWN, hareket);
2 //Bu satırla klavye dinleme dinleme işlemi başlatılır
3 //ve klavyeden bir olay gelirse hareket fonksiyonu
4 //çalıştırılır.
5 function hareket(event:KeyboardEvent) {
6     //eğer basılı tutulan tuş sol yön tuşuysa nesneyi sola
7     //doğru hareket ettiriyoruz.
8     if (event.keyCode===Keyboard.LEFT) {
9         TOP.x -= 5; // top nesnesinin x koordinatını 5
10        //eksiltir.
11    }
12    //eğer basılı tutulan tuş sağ yön tuşuysa nesneyi sağa
13    //doğru hareket ettiriyoruz.
14    if (event.keyCode===Keyboard.RIGHT) {
15        TOP.x += 5; // top nesnesinin x koordinatını
16        //5 artırır.
17    }
18    //eğer basılı tutulan tuş yukarı yön tuşuysa nesneyi
19    //yukarı doğru hareket ettiriyoruz.
20    if (event.keyCode===Keyboard.UP) {
21        TOP.y -= 5; // top nesnesinin y koordinatını 5
22        //eksiltir.
23    }
24    //eğer basılı tutulan tuş aşağı yön tuşuysa nesneyi
25    //aşağı doğru hareket ettiriyoruz.
26    if (event.keyCode===Keyboard.DOWN) {
```

```
27     TOP.y += 5; //top nesnesinin y koordinatını 5
28         //arttırır.
29     }
30
31 // TOP.y += 5; yerine TOP.x = TOP.x +5;şeklinde de
32     // kullanılabilir.
33 }
```

Bu uygulama ile çizdiğimiz bir şekli yön tuşlarıyla yukarı, aşağı, sağa ve sola hareket ettireceğiz.

Uygulamaya geçmeden önce **klavye komutlarını tanıyalım:**

- **KeyboardEvent** : Klavye olaylarını tanımlayan sınıf.
- **keyCode** : Klavyeden basılan tuş bilgisini aldığımız komut.
- **KeyboardEvent.KEY_DOWN** : Klavyeden herhangi bir tuşa basılma anıdır.
- **KeyboardEvent.KEY_UP** : Klavyeden basılan herhangi bir tuşun bırakılma anıdır.
- **Keyboard.UP** : Klavyedeki yukarı yön tuşudur. Diğer tuşlar da aynı mantıkla kullanılır.
- **Keyboard.DOWN** : Klavyedeki aşağı yön tuşudur.
- **Keyboard.RIGHT** : Klavyedeki sağ yön tuşudur.
- **Keyboard.LEFT** : Klavyedeki sol yön tuşudur.
- **Keyboard.A** : Klavyedeki A tuşudur. Diğer tuşlar da aynı mantıkla kullanılır.

Flash ActionScript 3.0 İle Mouse kontrolü

1. CLICK

Farenin bir nesneye sol tuş ile tıklanıp bırakılma olayıdır.

MouseEvent.CLICK şeklinde yazılır.

2. DOUBLE_CLICK

Farenin bir nesneye çift tıklama olayıdır.

MouseEvent.DOUBLE_CLICK şeklinde kullanılır

3. MOUSE_DOWN

Farenin sol tuşuna basma olayıdır. Click den farkı sadece basma ile çalmasıdır.

MouseEvent.MOUSE_DOWN şeklinde kullanılır.

4. MOUSE_UP

Farenin sol tuşunu bırakma olayıdır.

MouseEvent.MOUSE_UP şeklinde kullanılır.

5. MOUSE_OUT

Farenin bir nesne üzerinden ayrılma olayıdır.

MouseEvent.MOUSE_OUT şeklinde kullanılır. 39

6. MOUSE_MOVE

Farenin nesne üzerinde gezme olayıdır.

MouseEvent.MOUSE_MOVE şeklinde kullanılır.

7. MOUSE_OVER

Farenin nesne üzerine gelme olayıdır.

MouseEvent.MOUSE_OVER şeklinde kullanılır.

8. MOUSE_WHEEL

Farenin tekerleğinin döndürülme olayıdır.

MouseEvent.MOUSE_WHEEL şeklinde kullanılır.

9. ROLL_OVER

Farenin sol tuşunun basılı tutularak nesne üzerine gelme olayıdır.

MouseEvent.ROLL_OVER şeklinde kullanılır.

10. ROLL_OUT

Farenin sol tuşunun basılı tutularak nesne üzerinden ayrılma olayıdır.

MouseEvent.ROLL_OUT şeklinde kullanılır.

Örnek bir animasyon hazırlayalım. Sonra, **yeni bir katman** ekleyelim. Katmanımızın adı **butonlar** olsun ve bu katmana **iki adet buton** ekleyelim. İster Hazır buton ekleyin ister butonu kendiniz yapın. Ben dikdörtgen içine başlat ve durdur yazarak iki buton oluşturdum. Butonları düğme türünde sembole dönüştürelim. Sembole dönüştürdükten sonra başlat yazılı butonun örnek adına **baslat**, durdur yazılı butonun örnek adına da **durdur** yazalım ki butonları tanımlayalım. Butonlar katmanı da 50 karede görünecek unutmayalım.

Sonra **butonlar** katmanının **1.karesine** tıklayalım ve eylemler penceresini açalım. Aşağıdaki kodları yazalım:

```
1 import flash.events.MouseEvent;
2 function basla(Event:MouseEvent):void
3 {
4     play();
5 }
6 baslat.addEventListener(MouseEvent.CLICK,basla);
7 function dur(Event:MouseEvent):void
8 {
```

```
9   stop();
10  }
11  durdur.addEventListener(MouseEvent.CLICK,dur);
```

Kodları açıklayacak olursak;

.satır: Fare Olayları sınıfını animasyonumuza ekliyoruz.

2-3-4-5.satırlar: **basla** adında fonksiyon oluşturuyoruz. Basla fonksiyonu çağrıldığında animasyon duruyorsa **play** ile animasyonu başlatıyoruz.

6.satır: **baslat** butonuna, farenin tıklama olayını ekliyoruz ve fare ile tıklanıldığında **basla** fonksiyonunu çalıştırıyoruz.

Diğer satırlarda yukarıyla aynı mantık **dur** fonksiyonu oluşturup **stop** ile animasyonu durduruyoruz.

Gelelim diğer mouse olaylarını kullanmaya: [Kaynak: www.mustafakarsli.com](http://www.mustafakarsli.com)

İlk olarak bir adet daire çizdim ve film klibi şeklinde sembole dönüştürüp örnek adını top yaptım.

```
1   import flash.events.MouseEvent;
2   function ustunde(Event:MouseEvent):void
3   {
4       trace("Fare topun üstünde");
5   }
6   top.addEventListener(MouseEvent.MOUSE_OVER,ustunde);
```

Yukarıdaki örnek kod fare, top nesnesinin üzerine gelince çıktı penceresinde "**Fare topun üstünde**" yazıyor. Satırları açıklayacak olursak:

Yukarıdaki örnekte **kırmızı** ile yazılan kısımları kendimiz veriyoruz herhangi bir ifade yazılabilir. Diğer kısımlar ise standart kullanımdır.

1.satırda: Fare olayı sınıfı kütüphaneden animasyonumuza aktarılıyor. Kodların çalışması için yazmalıyız.

2.satırda: ustunde adında **fonksiyon** tanımlıyoruz. Fonksiyonumuza değer olarak fare olayı gönderilecek (**Event:MouseEvent**). 3 ile 5.satırdaki parantezler fonksiyonun başlangıç ve bitişini belirtiyor.

4.satırda: Çıktı ekranına mesaj verdiriyoruz. Bu satırda fonksiyon içinde yapılacak işlemleri yazıyoruz.

6.satırda: topa olay ekliyoruz. Farenin üzerine gelmesi olayını ekliyoruz yani fare topun üstüne gelince **ustunde** fonksiyonunu çalıştıracak.

Burada mantık nesneye 6.satırdaki gibi fare olayı ekleyip tanımladığımız bir fonksiyonu çağırıp çalıştırmak. Bu şekilde farenin diğer olaylarını da kullanabiliriz. Aşağıda birkaç olayla ilgili örnek kod ve swf mevcut inceleyebilirsiniz.

Yukarıdakine ek olarak: mesajları trace ile çıktı penceresinde değilde **sonucyaz** adında bir **dinamik metinde** yazdıralım: Bunun için **dinamik metin** ekleyelim ve örnek adını **sonucyaz** yapalım. Sonra aşağıdaki kodları yazalım:

```
import flash.events.MouseEvent;

function ustunde(Event:MouseEvent):void
{
    sonucyaz.text="Fare Topun Üstünde";
}
top.addEventListener(MouseEvent.MOUSE_OVER,ustunde);

function disinda(Event:MouseEvent):void
{
    sonucyaz.text="Fare Topun Dışında";
}
top.addEventListener(MouseEvent.MOUSE_OUT,disinda);

function tikla(Event:MouseEvent):void
{
    sonucyaz.text="Topa Tıklandı";
}
top.addEventListener(MouseEvent.CLICK,tikla);
```

Kaynak

1- Animasyon Temelleri - MEGEP Eğitim Modülleri Erişim Tarihi : 20.02.2018
Url : http://www.megep.meb.gov.tr/mte_program_modul/moduller_pdf/Animasyon%20Temelleri.pdf