

Maske, Eklem Kullanımı ve ActionScript'e Giriş

Maske Kullanımı

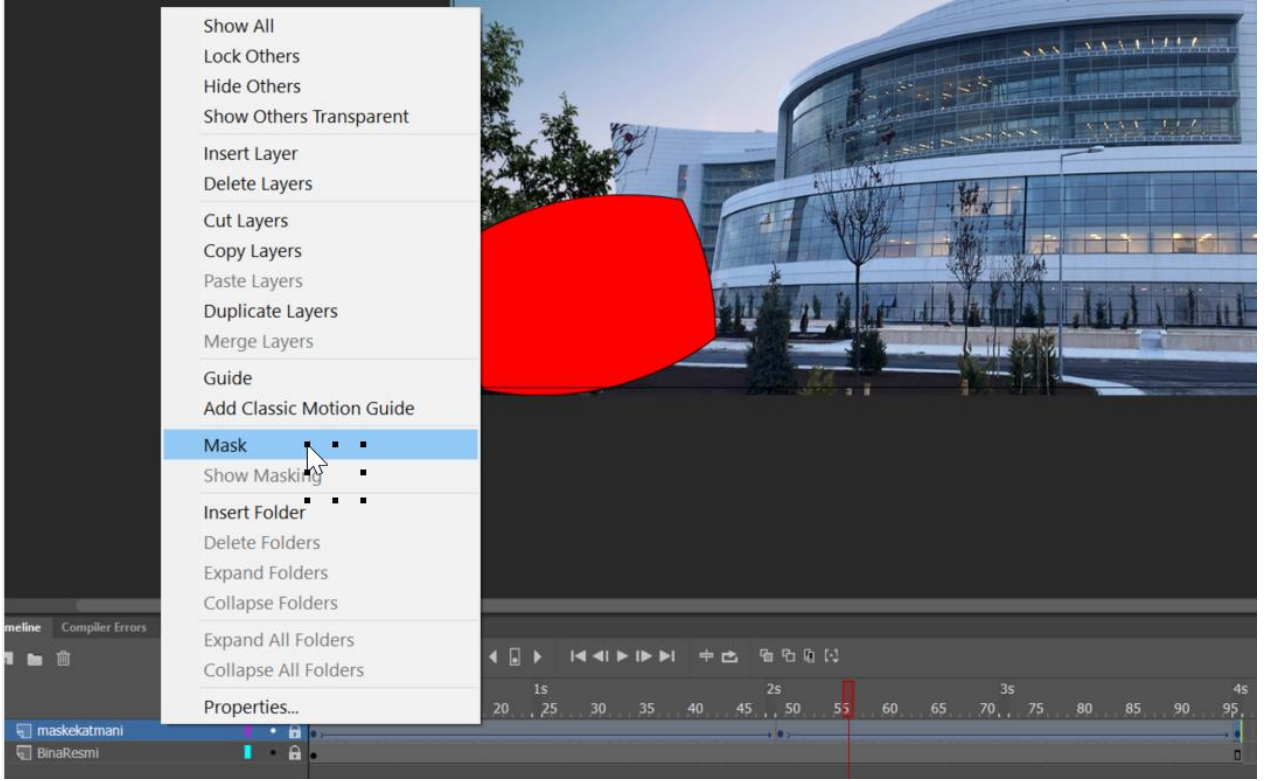
Maske katmanı altta bulunan katmanı belirlenen alan kadar göstermek için kullanılan bir özelliktir. Spot ışığı efektleri ve geçişleri için altta bulunan katmanların görüldüğü bir delik oluşturmak üzere bir maske katmanı kullanın. Maske ögesi; doldurulmuş bir şekil, bir tür nesnesi, bir grafik sembolü örneği veya bir film klibi olabilir. Karmaşık efektler oluşturmak için birden çok katmanı tek bir maske katmanı altında gruplayın.

Dinamik efektler oluşturmak için bir maske katmanını hareketlendirin. Maske olarak kullanılan doldurulmuş şekil için şekil arası kullanın; tür nesnesi, grafik örneği veya film klibi için ara hareket kullanın. Bir film klibi örneğini maske olarak kullanırken, maskeyi bir yol boyunca hareketlendirin.

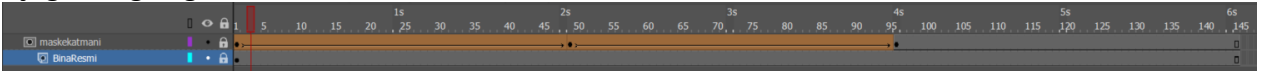
Maske katmanı oluşturmak için bir maske ögesini maske olarak kullanmak üzere katman üzerine yerleştirin. Bir dolgu veya kontura sahip olmak yerine, maske ögesi, bağlı katmanların maske ögesinin altındaki alanını ortaya çıkaran bir pencere gibi davranır. Maske katmanının geri kalanı, maske ögesinin arasından görünenler dışındaki her şeyi saklar. Bir maske katmanı sadece bir maske ögesi içerebilir. Bir maske katmanı bir düğmenin içinde olamaz ve bir maskeyi başka bir maskeye uygulayamazsınız.

Örnek: Yüklümiş `ornek3.zip` içindeki `swf`'ye benzer bir örnek yapacağız.

1. Öncelikle yeni bir `animate` dosyası oluşturun. Bu dosya içine mevcut katmana arka plan olacak şekilde daha önce de kullandığımız AFAD binası resmini ekleyebilirsiniz.
2. Animasyonumuz süresini uzatmak için 4. Saniyeye yani 96. Kareye kadar frame (kare) ekleyelim (kısayolu F5).
3. Bu katmanın üstüne yeni bir katman ekleyip ismini `maskekatmani` yapalım.
4. Bu katmana sahnenin merkezine bir daire şekli (dolgusu olduğu sürece rengi önemli değil) ekleyelim ve ardından 49. Kareye bir anahtar kare (keyframe) ekleyelim.
5. Merkeze konumlandırdığımız şekli sol alt köşeye çekelim ve iki anahtar kare arasına sağ tıklayıp `Create Shape Tween` (şekil arası oluştur) seçeneğine tıklayalım.
6. Şimdi aynı şekli büyütüp dikdörtgene çevirmek için son kare olan 96. Kareye gidip F7 tuşuna basarak boş bir anahtar kare ekleyelim ve bu kareye sahne büyüklüğünde bir dikdörtgen çizelim.
7. Bir önceki hareket için yaptığımız gibi bu kez 50. ve 96. Kareler arasına tıklayıp şekil arası oluştur seçeneğine tıklayalım. Kaymanı maskeye dönüştürmeden önce `Ctrl+Enter` ile animasyonu test edebilirsiniz.
8. Bu görüntünün maskeye dönüşmesi için tek yapmamız gereken üstteki katman olan `maskekatmani` na sağ tıklayıp maske seçeneğine tıklamak.



9. Bu ilemin ardından bina resmi oluşturduğumuz maske katmanındaki içerik kadar görünecektir. Eğer binanın resminin daha uzun süre görünmesini istiyorsanız iki katmanın sonuna 2 sn (48 kare) uzunluğunda kareler ekleyebilirsiniz. Zaman çizelgeniz işlem sonunda aşağıdaki gibi görünmelidir.



Zaman Çizelgesi Ayarları

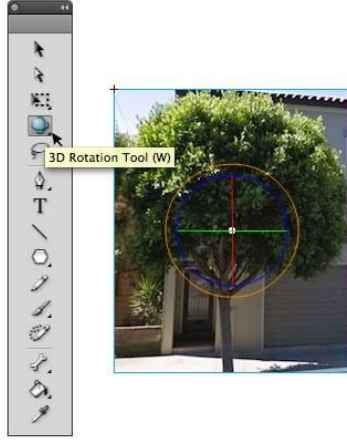
- Zaman Çizelgesini görsel olarak değiştirmek
- Döngü Kaplaması (Soğan kabuğu) ayarının kullanımı ve amacı
- Döngü ile zaman çizelgesinde test etme

Sahne ile Çalışma

- Yeni bir Sahnekleme
- Sahneler arası geçiş

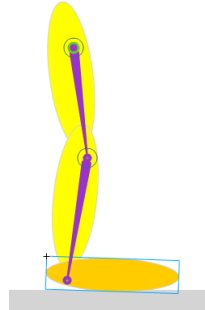
3D Çevirme Aracı

3D çevirme aracı 2 boyutlu nesnelere 3 boyutluymuş gibi x ve y koordinatları dışında z koordinatına göre de hareket ettirip animasyon oluşturmaya yarar.



Ekleme Aracı

- Eklemlerinkine benzer iki film klibinin birbirine göre ve bağlı hareketini oluşturmaya yarar.
- Örnek Ekleme için ayrı eklenmiş olan ayrı örneği kullanacağız.



ActionScript'e Giriş

addEventListener

Event listener'lar (olay dinleyiciler-tetikleyiciler), Flash'ın belirli olayları başlatması için kullandığı kodlardır. Event listener eklemek iki adımlık bir süreçtir. Öncelikle olaya cevap veren bir fonksiyon veya class method'u yaratılmalıdır. Sonra addEventListener() method'u kullanarak olay akışında gerçekleştirilmek istenenlerin yer aldığı fonksiyona yönlendiren bir tetikleyici eklenmelidir.

Örnek: Sahneye bir düğme ekleyin ve örnek adını "mybutton" olarak değiştirin.

```
1 mybutton.addEventListener(MouseEvent.CLICK, buttondown);
2 function buttondown(e:MouseEvent):void
3 {
4   trace ("Clicked"); // Output Clicked
5 }
```

Mouse EventListener tipleri-örnekleri

Temel MouseEvent'leri
CLICK : String = "click"
DOUBLE_CLICK : String = "doubleClick"
MIDDLE_MOUSE_UP : String = "middleMouseUp"
MOUSE_MOVE : String = "mouseMove"
MOUSE_OUT : String = "mouseOut"
MOUSE_OVER : String = "mouseOver"
MOUSE_UP : String = "mouseUp"
RELEASE_OUTSIDE : String = "releaseOutside"
RIGHT_CLICK : String = "rightClick"
RIGHT_MOUSE_DOWN : String = "rightMouseDown"
RIGHT_MOUSE_UP : String = "rightMouseUp"
ROLL_OUT : String = "rollOut"
ROLL_OVER : String = "rollOver"
MOUSE_DOWN : String = "mouseDown"

stop - nextFrame - prevFrame

stop(): Yazıldığı karede animasyonu durdurur.

Örnek: Yeni bir hareket arası ile oluşturmuş bir animasyon yaratın. Animasyonun son karesine stop(); yazıp animasyonu test edin.

nextFrame(): Animasyonu bir sonraki kareye ilerletir.

prevFrame(): Animasyonu bir önceki kareye taşır.

Örnek: Aynı animasyonu bu kez yeni bir katmana kendi oluşturduğunuz düğmeyi ekleyerek ve ilk kareye aşağıdaki kodları yazınız.

```
1 stop();
2 mybutton.addEventListener(MouseEvent.CLICK, buttondown);
3
4 function buttondown(e:MouseEvent):void
5 { nextFrame(); }
6
```

Örnek 2: Aynı animasyonun ikinci karesine aşağıdaki kodu yazınız.

```
1 stop();
2 mybutton.addEventListener(MouseEvent.CLICK, buttondown);
3
4 function buttondown(e:MouseEvent):void
5 { prevFrame(); }
6
```

gotoAndPlay() ve gotoAndStop()

gotoAndPlay(frame:Object , scene:String): Animasyonun belirlenen kare ve sahneye gönderip animasyonu oynatır. Eğer sahne belirtilmemişse mevcut sahnenin belirtilen karesine ilerler.

Sadece anasahne içine yazılan kodda Scene parametresi çalışacaktır. Sembol içinde bu parametre kullanılamaz.

Örnek: Yeni bir animasyon oluşturup ilk karesine bir düğme ekleyip örnek adını “ilkdugme” olarak belirleyin ve aşağıdaki kodu yazın.

```

stop();
ilkdugme.addEventListener(MouseEvent.CLICK, gotoplayornegi);
function gotoplayornegi (e:MouseEvent):void
{
gotoAndPlay(2);
}

```

Örnek: Aynı örneğe sceneTwo isimli (shift+F2) ile sahne penceresini açıp isim değişikliği yapabilirsiniz) ikinci bir sahne ekleyin ve bu sahnede yeni bir animasyon oluşturun. Birinci sahnenin ilk karesine yeni bir düğme ekleyerek örnek adını mybutton olarak belirleyip aşağıdaki kodları ekleyiniz.

```

1  stop ();
2
3  mybutton.addEventListener(MouseEvent.CLICK, gotoframe);
4
5  function gotoframe (e:MouseEvent):void
6  {
7  gotoAndPlay (1, "sceneTwo");
8  }

```

gotoAndStop(frame:Object, scene:String): gotoAndPlay() den farklı olarak animasyon gidilen karede oynamak yerine duracaktır.

Örnek: Yukarıdaki örneği aşağıdaki gibi değiştirip aradaki farka dikkat ediniz.

```

1  stop ();
2
3  mybutton.addEventListener(MouseEvent.CLICK, gotoframe);
4
5  function gotoframe (e:MouseEvent):void
6  {
7  gotoAndStop (1, "sceneTwo");
8  }

```