

```

// ACTION SCRIPT
// Bu uygulama, as3coder.com / Burak AYDIN tarafından hazırlanmıştır

// İşlemler dizisiyle haberleşirken kullanmak üzere işlem sırası integer
değişkeni oluşturuyoruz
// İlk değer -1 olmalı çünkü diziler 0 değerinden başlar ve ilk işlem
varsayılan olarak kaydedilecek.
// Yani şuan kutunun durduğu yer bir işlem olarak kaydedildiğinde işlem
numarası 0 olacak.
// Böylece diziden konumlar 0 elemanından geri çağırılabilir.
var islemSirasi_int:int=-1;

// Kutu buton gibi davranсын
kare_mc.buttonMode=true;

// Taşıma Fonksiyonu oluşturuluyor
// Baskı uygulandığında taşımaya başla fonksiyonunu çalıştır
kare_mc.addEventListener("mouseDown",tasimayaBasla);
// Baskı sonlandırıldığında taşımayı bitir fonksiyonunu çalıştır
kare_mc.addEventListener("mouseUp",tasimayiBitir);
// Taşımaya başla fonksiyonu oluşturuluyor
function tasimayaBasla(e:Event) {
    // Kare film klibini taşıma başla
    MovieClip(kare_mc).startDrag();
}

// Taşımayı bitir fonksiyonu oluşturuluyor
function tasimayiBitir(e:Event) {
//if (kare_mc.hitTestPoint(mouseX,mouseY)) {
// Kare 'yi taşımayı bitir
kare_mc.stopDrag();
    // İşlem kaydedici fonksiyonuna değerler gönderiliyor
    islemiKaydet(kare_mc.x,kare_mc.y);
//}
}

// İşlem sırası için dizi oluşturuluyor
var islemSirasi_arr:Array = new Array();

// İşlem kaydedici fonksiyonu oluşturuluyor. x konumu ve y konum için
sayısal birer değer kabul edeceğiz.
function islemiKaydet(xKonum:Number,yKonum:Number) {
    // Normal bir şekilde işlem seyrediyorsa, taşıma yapılıyorsa, işlem
sırası toplam işlemde bir eksik olur.
    // Böylece işlem sonlandığında yeni işlem dizisine kaydedilir. Biz
burada bunu sorguluyoruz.
    // Eğer işlem sırası işlem toplamından bir eksik ise işlem sırasını
bir artır.
    // Eğer aksine, geri al ve ileri al butonları kullanılıyorsa işlem
sırasının işlem toplamından bir eksik olma kuralı bozulmuştur.
    // Böyle bir durumda o anki işlemde sonraki tüm işlemleri
siliyoruz. Aynı paket programlarda olduğu gibi.
    // Eğer işlem sırası, işlem toplamından bir eksik ise, yani
kullanıcı usul usul taşı bırak işlemi yapıyorsa
    if (islemSirasi_int == islemSirasi_arr.length -1) {

```

```

        // İşlem sırasını bir artır, böylece diziye yeni değerleri
        işlem sırası numarasıyla ekleyebileceğiz
        işlemSirasi_int++;
// Değilse, yani kullanıcı işlemi geri al, ileri al butonlarıyla
yapıyorsa
} else {
    // Bütün işlem sırası bozulmuştur. Öyleyse şu anki adımdan
    sonrasını diziden temizle. (Adobe Flash'ta olduğu gibi)
    işlemSirasi_arr.splice(islemSirasi_int,islemSirasi_arr.length
    - işlemSirasi_int)
}

// işlemSirasi_arr[5] = [12,43] gibi düşünebilirsiniz. Buradaki 5
sonraki işlem numarası. Dizideki 5 elemanı içinde iki eleman daha
var.
// xKonum ve yKonum. Bunlara yeni konum değerlerini yazdırıyoruz
işlemSirasi_arr[islemSirasi_int]=[xKonum,yKonum];
// Çalışma sırasında gerekli bilgileri output ekranımdan görmek
istedim.
trace("iş sır arr: ",islemSirasi_arr.length);
trace("iş sır int: ",islemSirasi_int);
}

// işlem sırası için ilk konum kaydediliyor. Yani kutunun x ve y konumu
ilk andaki konum için işlem sırasına yazdırılıyor.
işlemiKaydet(kare_mc.x,kare_mc.y);
// Geri alma fonksiyonu tanımlanıyor
function geriAl(e:Event=null) {
    // Eğer işlem sırası integer 0'dan büyükse halen history'de yaşayan
    bir işlem var demektir. Geri alırken bunu sorguluyoruz.
    if (islemSirasi_int>0) {
        // İşlem sırasını bir azalt, birazdan bir azaltılmış işlem sırası
        diziden numara olarak çağırılacak.
        işlemSirasi_int-=1;
        trace("İşlem Sırası: ",islemSirasi_int);
        // Değeri bir azaltılmış işlem sırası kaçsa, dizide kayıtlı
        işlemlerden dizi numarasına göre çağırılıyor.
        kare_mc.x=islemSirasi_arr[islemSirasi_int][0];
        kare_mc.y=islemSirasi_arr[islemSirasi_int][1];
        // Değilse, yani işlem sırası integer 0'a eşitse ve başka geri
        alınacak işlem kalmadıysa, bunu geliştiriciye bildir.
    } else {
        trace("Yapılacak İşlem Yok: Geri Al");
    }
}

// İleri alma fonksiyonu tanımlanıyor
function ileriAl(e:Event=null) {
    // işlem sırası, dizideki tüm işlemlerin toplamından küçükse, ileri
    alma işlemine izin veriyoruz
    if (islemSirasi_int<islemSirasi_arr.length-1) {
        // İşlem sırasını bir artır, böylece birazdan, bir artırılmış
        değeri dizideki yeriyile eşleştirebileceğiz.
        işlemSirasi_int++;
        trace("İşlem Sırası: ",islemSirasi_int);
        // Bir artırılmış işlem sırası integer'ı dizideki yeriyile
        eşleştiriyoruz. Kutunun konumunu ileri alıyoruz
    }
}

```

```
        kare_mc.x=islemSirasi_arr[islemSirasi_int][0];
        kare_mc.y=islemSirasi_arr[islemSirasi_int][1];
    } else {
        trace("Yapılacak İşlem Yok: İleri Al");
    }
}

// Tüm işleri sıfırlamak için yardımcı bir fonksiyon.
function sıfirla(e:Event=null) {
    islemSirasi_arr.splice(0,islemSirasi_arr.length);
    islemSirasi_int=-1;
    trace("Dizi Temizlendi, Eleman Sayısı: " , islemSirasi_arr.length);
    kare_mc.x=stage.stageWidth/2-kare_mc.width/2;
    kare_mc.y=stage.stageHeight/2-kare_mc.height/2;
    // İşlem ilk kayıt için diziye yazdırılıyor.
    islemiKaydet(kare_mc.x,kare_mc.y);
}

// Geri al butonuna geriAl fonksiyonu ekleniyor
geri_btn.addEventListener("click",geriAl);
// İleri al butonuna geriAl fonksiyonu ekleniyor
ileri_btn.addEventListener("click",ileriAl);
// Sıfırla butonuna geriAl fonksiyonu ekleniyor
sifirla_btn.addEventListener("click",sifirla);
```