SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

NESNEYE DAYALI PROGRAMLAMA DERSİ PROJE ÖDEVİ

PROJE KONUSU: TETRİS OYUNU

GRUP ÜYELERİ:

G111210037- ABDURRAHMAN PARLATICI

(abdurrahman.parlatici@ogr.sakary.edu.tr)

G121210001- SEDA NARLI

(seda.narli@ogr.sakarya.edu.tr)

Proje Tanıtımı

Tetris oyunu karelerden oluşan çeşitli şekillerin hareketlerinin kontrol altında tutulduğu bir oyundur. Bu şekillerin oluşabilmesi için karelere ihtiyaç duyuldu. Karelerin ise koordinatlarını belirleyebilmek için nokta sınıfı tasarlandı. Tüm şekiller ise oyun sınıfı tarafından kontrol edildi.

Senaryo

Nokta sınıfı x ve y değişkenleri yardımıyla koordinat belirlemek için tasarlandı.

Kare sınıfında kareleri temsil eden resimler ve dört adet nokta değişkenleri tutularak karelerin istenildiği şekilde ekrana yansıtılması sağlandı.

Şekil sınıfı gerçekte bir nesneyi temsil etmediği için abstract olarak tanımlandı. Bu sınıfta şekillerin hareket etmeleri için gereken dönüş(abstarct),aşağı in, sağa git ve sola git fonksiyonları yazıldı. Hareketlerin mümkün olup olmadığı şekil sınıfında yazılan kontrol fonksiyonları ile belirlendi.

T, çubuk, Z ve ters Z, L ve ters L, T ve kare şekilleri için ayrı ayrı sınıflar yazıldı. Temel sınıf olarak belirlenmiş şekil sınıfının aracılığıyla aşağı in, sağa git ve sola git fonksiyonları tekrar yazılmaksızın kullanıldı. Bunlara ek olarak her fonksiyonun kendi dönüş fonksiyonları yazıldı.

Oyun sınıfına yeni şekil oluşturma ve oyunu sonlandırma işlevleri kazandırıldı. Yeni şekil oluşturma rastgele olacak şekilde gerçeklendi. Eğer yeni şekil oluşturulamıyorsa oyunun sonlandırılmasına karar verildi.

Yapılar

Programın ana yapıları aşağı inme ve dönme fonksiyonlarıdır.

Aşağı inme fonksiyonu:

```
public bool asagiIn(bool [,] oyunAlani)
{
    for (int i = 0; i < 4; i++)//sınır kontrolü
    {
        if (kareler[i].Noktalar[2].Y / 20 + 1 > 28)
            return false;
    }
    if (kontrolAsagiIn(oyunAlani) == false)//karelerin altı dolu mu
    {
        return false;
    }
    for (int i = 0; i < 4; i++)//karelerin yeni konumunu ayarlama
    {
            kareler[i].konumlandır(kareler[i].Noktalar[0].X ,
            kareler[i].Noktalar[0].Y+20);
    }
    return true;
}
```

Bu fonksiyonda öncelikle ekranın sınırlarını aşıp aşmadığı kontrolü yapılmaktadır. Formda kullanılan ekran boyutları 400x560 olduğundan ve karelerimiz yirmişer birimden oluştuğundan kontroller ekran boyutunun 20'ye bölünmesinden elde edilen sayılarla yapılmaktadır. Eğer herhangi bir sınır aşımı var ise hareket yapılmaksızın false değeri döndürülür.

Sınır aşımı kontrolünden negatif bir sonuç alınmadığı takdirde yazılan kontrol fonksiyonu yardımıyla ekranda gidilecek noktanın dolu olup olmadığı kontrolü yapılır. Ondan da negatif bir sonuç alınmaz ise tüm karelerin konumları bir kare boyutu kadar aşağı indirilerek şeklin aşağıya doğru hareketi sağlanmış olur. Hareket gerçekleştirildiği için fonksiyonun sonunda true değeri geri döndürülür.

Çubuk Şeklindeki Sola Dönme Fonksiyonu:

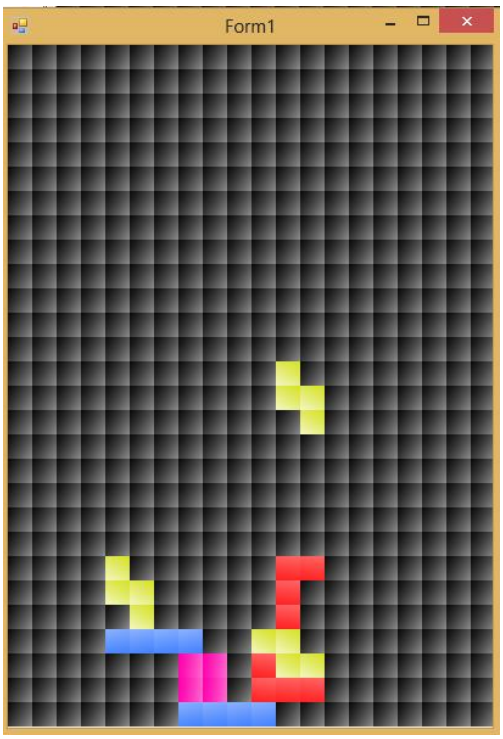
```
private void solaDon(bool [,]oyunAlani)
{
    if (kontrolSolaDon(oyunAlani) == false)
        return;
    durum = 1;
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            kareler[i].Noktalar[j].X -= 20*(3-i);
            kareler[i].Noktalar[j].Y += 20 * (3-i);
            kareler[i].setXY();
        }
    }


}
```

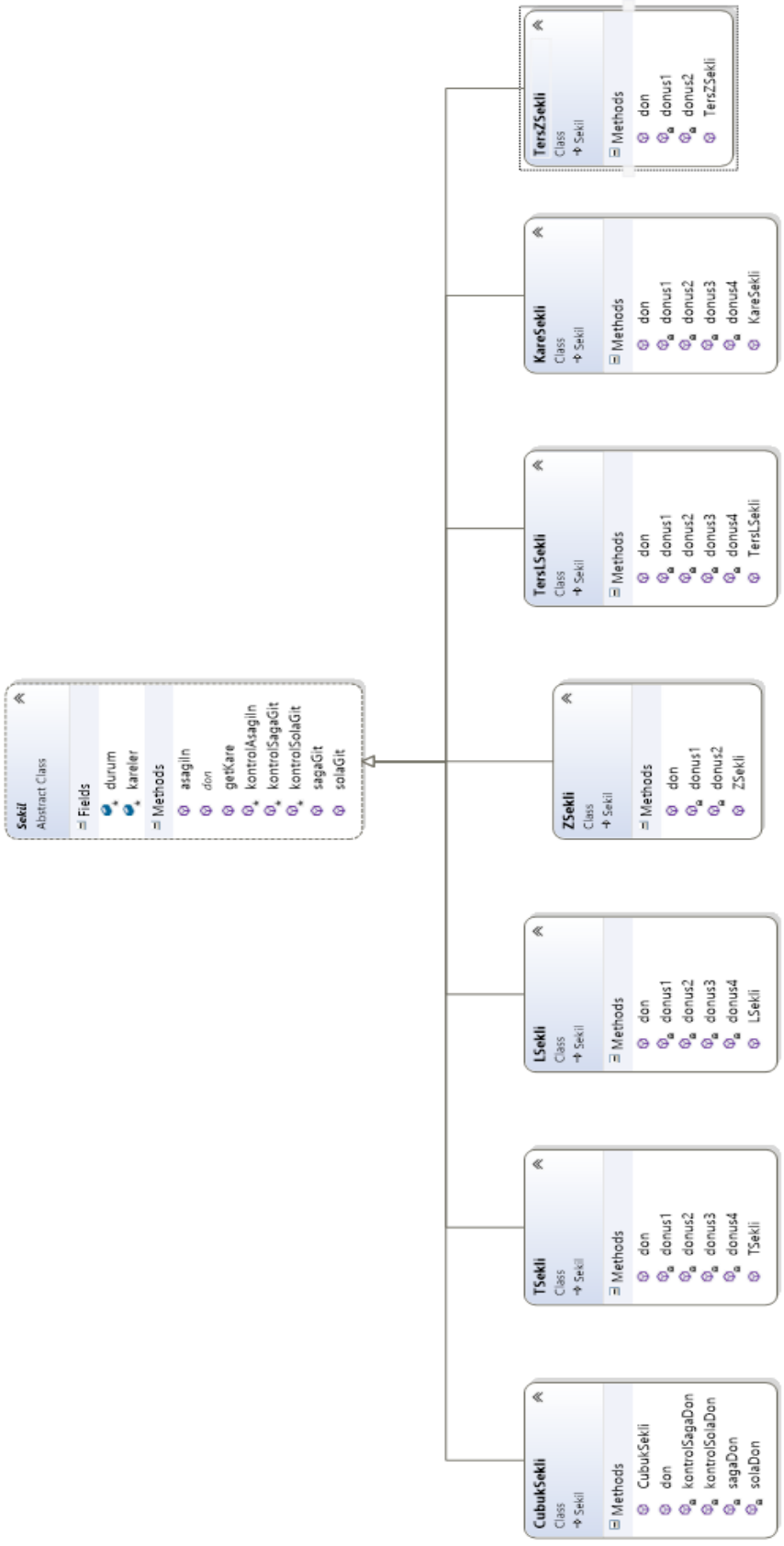
Bu fonksiyon örnek olarak yazılmıştır. Diğer şekillerdeki dönme fonksiyonları da buradaki mantık üzerine kurulmuştur.

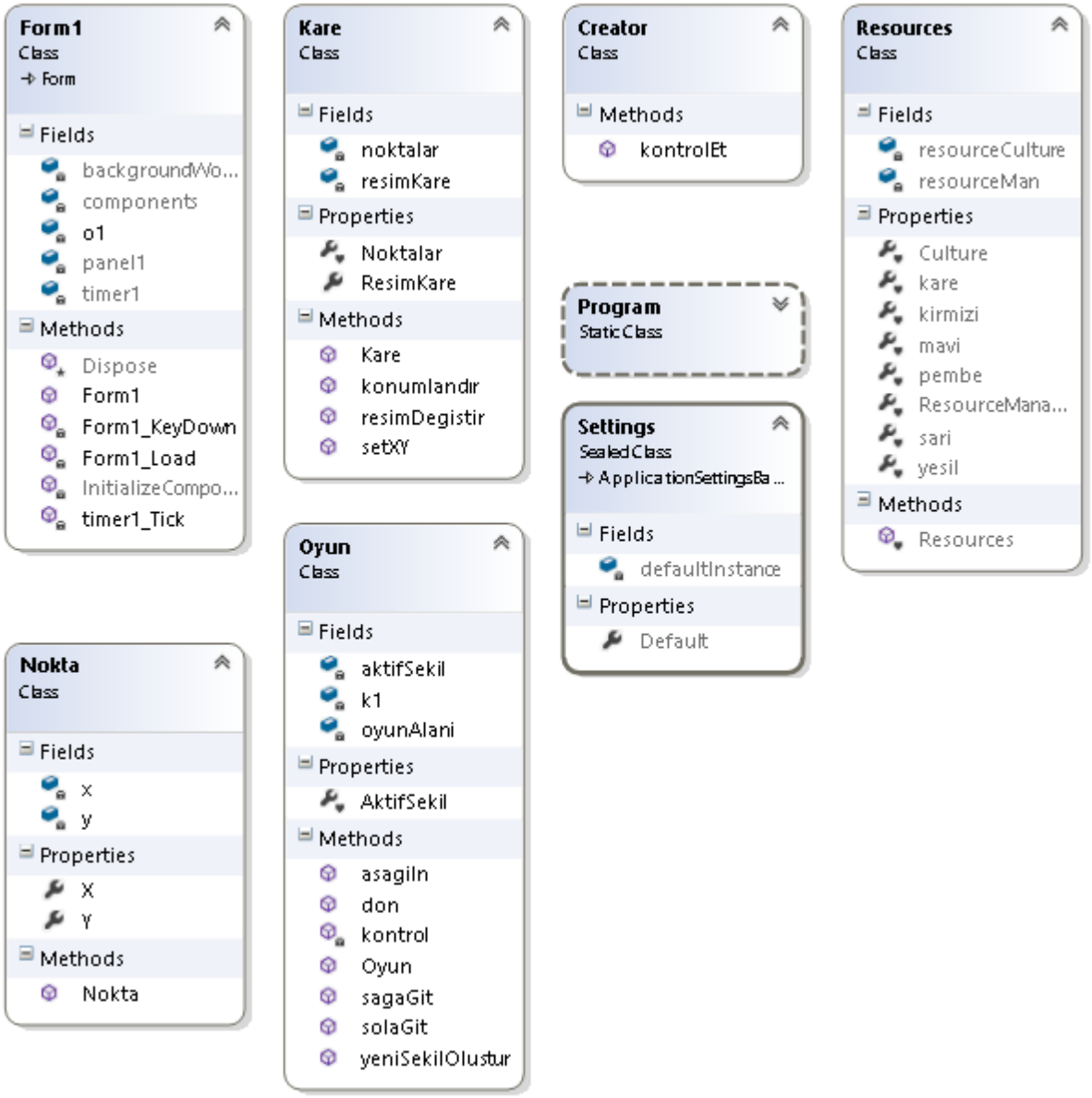
Burada öncelikle yazılan kontrol sayesinde şeklin sola dönüp dönemediği kararı verilmektedir. Eğer negatif bir sonuç çıkmaz ise durum değiştirilip karelerin hareketi sağlanmaktadır. Hareketler gerçekleştirilirken yazılan iç içe döngüdeki matematiksel hesaplamalar çubuğun sola dönmesi için özel olarak tasarlanmış olup diğer şekillerin de kendine ait hesaplamaları mevcuttur.

Ekran Görüntüsü



UML


Sekil
Abstract Class
Fields
durum
kareler
Methods
asagiIn
don
getKare
kontrolAsagiIn
kontrolSagaGit
kontrolSolaGit
sagaGit
solaGit
CubukSekli
Class
Sekil
Methods
CubukSekli
don
kontrolSagaDon
kontrolSolaDon
sagaDon
solaDon
TSekli
Class
Sekil
Methods
don
donus1
donus2
donus3
donus4
TSekli
LSekli
Class
Sekil
Methods
don
donus1
donus2
donus3
donus4
LSekli
ZSekli
Class
Sekil
Methods
don
donus1
donus2
ZSekli
TersLSekli
Class
Sekil
Methods
don
donus1
donus2
donus3
donus4
TersLSekli
KareSekli
Class
Sekil
Methods
don
donus1
donus2
donus3
donus4
KareSekli
TersZSekli
Class
Sekil
Methods
don
donus1
donus2
TersZSekli


Form1
Class
→ Form
Fields
backgroundWo...
components
o1
panel1
timer1
Methods
Dispose
Form1
Form1_KeyDown
Form1_Load
InitializeCompo...
timer1_Tick
Kare
Class
Fields
noktalar
resimKare
Properties
Noktalar
ResimKare
Methods
Kare
konumlandır
resimDegistir
setXY
Creator
Class
Methods
kontrolEt
Resources
Class
Fields
resourceCulture
resourceMan
Properties
Culture
kare
kirmizi
mavi
pembe
ResourceMana...
sari
yesil
Methods
Resources
Program
Static Class
Settings
Sealed Class
→ ApplicationSettingsBa ...
Fields
defaultInstance
Properties
Default
Nokta
Class
Fields
x
y
Properties
X
Y
Methods
Nokta
Oyun
Class
Fields
aktifSekil
k1
oyunAlani
Properties
AktifSekil
Methods
asagiIn
don
kontrol
Oyun
sagaGit
solaGit
yeniSekilOlustur