Programlama Labaratuvarı 1. Proje Ödevi

1. Enes Furkan SAĞLAM
Bilgisayar Mühendisliği 2.öğretim
Kocaeli Üniversitesi
Kocaeli, Türkiye
enesfurkansaglam71gmail.com

2. Yaşar Yiğit USTA
Bilgisayar Mühendisliği 2.öğretim
Kocaeli Üniversitesi
Kocaeli, Türkiye
yigitusta112gmail.com

***Özetçe* —Bu belge, Kocaeli Üniversitesi Bilgisayar Mühendisliği bölümü programlama laboratuarı 2 dersi 1. Proje ödevi için hazırlanmıştır.**

Anahtar Kelimeler —BFS Algoritması ,javaFX,kalıtım,arayüz, interfaces,class,implement,extends,matris

ÖZET

Bu projede, Java programlama dili kullanılarak bir matris üzerinde otonom hareket eden ve engellerden kaçınarak hazineleri toplayan bir karakter kodlanmıştır. Uygulamanın kullanıcı tarafından kontrol edilebilir olması sağlanmıştır.

GİRİŞ

Bu projede, Java programlama dili ile hareketli ve hareketsiz olmak üzere iki çeşit engel türü oluşturulmuştur. Bu engeller matrise yüksek maliyetle yerleştirilmiştir. Ardından hazineler oluşturulmuş ve bu hazineler düşük maliyetle matrise yerleştirilmiştir. Bu maliyetler kullanılarak karakterin en kısa yolu bulması sağlanmıştır.

YÖNTEM

Classlar:

Engel:

Bu sınıf, engellerin genel (ad, imagePath, konum, boyut) özelliklerini tanımlamak için oluşturulmuştur. Engellerin içine atıldığı bir ArrayList oluşturulmuştur ve bu sınıfa ek olarak Clonable interface'i implement edilmiştir.

HareketsizEngelYaz:

Bu sınıf, HareketsizEngel sınıfını extends etmiş ve özelliklerini miras almıştır. Bu sınıfta haritanın sol kısmında çıkan yaz konseptli engeller tanımlanmıştır ve kullanıcın istediği miktara göre klonlanmıştır. YazEngelOlustur metodu da burada tanımlanmıştır.

HareketliEngel:

Bu sınıf, Engel sınıfını extends edip onun özelliklerini almıştır. Üstüne HareketliEngel sınıfına has hareketYonu ve hareketBoyutu tanımlanmıştır. Bu sınıfta hareketli engeller oluşturulup ArrayList'e atılmıştır. hareketliEngelOlusur, hareketEttir ve Boyama metotları yazılmıştır. Clonable interface'i kullanılarak her nesnenin klonu yaratılmıştır.

HareketsizEngel:

Bu sınıf, Engel sınıfını extends etmiş ve onun özelliklerini miras almıştır.

Karakter:

Bu sınıf, bir JavaFX uygulamasında kullanılmak üzere bir karakterin özelliklerini ve davranışlarını tanımlar. Karakter sınıfı, karakterin ID'si, adı, resim yolu, boyutları ve başlangıç konumu gibi özelliklerini içerir. Ayrıca, karakterin keşfettiği hazineleri ve engelleri takip etmek için listeler de içerir. KarakterOlustur metodu, karakterin rastgele bir konumda oluşturulmasını ve JavaFX sahnesine eklenmesini sağlar. Karakterin başlangıç konumu, engellerle çakışmaması için kontrol edilir. karakterHareket metodu, karakterin belirlenen koordinatlara doğru hareket etmesini ve bu sırada çevresindeki hazine ve engelleri kontrol etmesini sağlar. Her adımda, karakterin konumu güncellenir ve hazine ya da engel keşfedilirse ekranda gösterilir. Sınıf ayrıca, karakterin özelliklerini ve konumunu değiştirmek için gerekli olan setter ve getter metotlarını da içerir.

HareketsizEngelKis:

Bu sınıf, HareketsizEngel sınıfını extends etmiş ve özelliklerini miras almıştır. Bu sınıfta haritanın sağ kısmında çıkan kış konseptli engeller tanımlanmıştır ve kullanıcın istediği miktara göre klonlanmıştır. KisEngelOlustur metodu da burada tanımlanmıştır.

Hazine:

Bu sınıf, Clonanable interface'ini implement edilmiştir. Bu sınıfta hazineler tanımlanmış ve kullanıcının istediği miktara göre klonlanmıştır. HazineOlustur, enYakinHazineBul metodu da burada tanımlanmıştır.

Kordinat:

Bu sınıf, karakterin konumunu belirlemek için kullanılan x ve y koordinatlarını içerir. Ayrıca, karakterin hareket ettiği koordinatları takip etmek için bir kordinatArrayListKarakter adında bir ArrayList de içerir. Kordinat sınıfının constructor'ı, verilen x ve y değerlerini kullanarak bir Kordinat nesnesi oluşturur ve bu nesneye verilen değerleri atar. Bu nesneler, karakterin hareket ettiği konumları temsil etmek için kullanılır. Bu sınıf, karakterin konumunu temsil etmek ve bu konumu takip etmek için kullanılan basit bir veri yapısıdır.

Lokasyon:

Bu sınıf, oyun haritasındaki farklı öğelerin (karakter, hazine, engel vb.) konumlarını belirlemek ve kontrol etmek için kullanılır. KORDINATLAR adında bir 2 boyutlu bir matris kullanarak haritanın durumunu tutar. KarakterKordinatYaz, HazineKordinatYaz, HareketliEngelKordinatYaz, HareketsizEngelYazKordinatYaz ve HareketsizEngelKisKordinatYaz metotları, farklı öğelerin (karakter, hazine, hareketli engel, hareketsiz engel vb.) konumlarını matrise yazmak için kullanılır. HaritaMatrisYazdir metodu, matristeki verileri konsola yazdırmak için kullanılır. Kontrol metodu, belirtilen koordinatlarda matrisin kontrol edilmesini sağlar. Eğer belirtilen koordinatlarda herhangi bir engel varsa (-1) döner, yoksa (1) döner. MatrisiBirle metodu, matrisi tüm elemanları 1 olan bir matrise dönüştürmek için kullanılır. Bu, matrisin tamamını kullanılabilir bir duruma getirmek için kullanılabilir. Bu sınıf, oyun haritasındaki farklı öğelerin konumlarını ve durumlarını yönetmek için kullanılır ve oyun mekaniği tarafından kullanılır.

Sis:

Bu sınıf, oyun haritasında sis efektini temsil etmek için kullanılır. Sis sınıfı, sis efektinin özelliklerini ve görünümünü tanımlar. sis nesnesi, sis efektinin resim yolu ve adını tutar. Her bir harita karesi için bir sis efekti oluşturulabilir. sisArrayListImageView liste, ekran üzerinde görüntülenen tüm sis efektlerini tutar. Sisle metodu, ekran üzerinde sis efektlerini oluşturmak için kullanılır. Oyun haritasının her bir karesi için bir sis efekti oluşturulur ve ekran üzerinde gösterilir. Sınıf, sis efektinin özelliklerini tutmak ve ekran üzerinde göstermek için kullanılır. Bu sayede oyunun atmosferini zenginleştirebilir ve oyunculara daha etkileyici bir deneyim sunabilir.

Uygulama:

Bu sınıf, verilen bir 2D matriste başlangıç ve hedef noktalar arasında en kısa yolun bulunmasını sağlayan genel bir yol bulma algoritması olan Breadth-First Search (BFS) algoritmasını içerir. enKisaYoluBul metodu, BFS algoritmasını kullanarak başlangıç ve hedef noktalar arasında en kısa yolu bulur. Bu metot, harita adında bir 2D matris, başlangıç ve hedef noktaları alır ve en kısa yolu temsil eden bir yol listesi döndürür. yolunuYenidenOlustur metodu, BFS algoritmasıyla bulunan en kısa yolu yeniden oluşturur. Bu metot, BFS sırasında tutulan bir ebeveyn haritasını kullanarak başlangıç ve hedef noktalar arasındaki en kısa yolu oluşturur. Bu sınıf, oyun haritasında karakterin hedefe en kısa yolu bulmasını sağlamak için kullanılabilir. Karakterin hareket edebileceği yollar ve engeller matriste belirtilirken, bu algoritma karakterin engelleri aşarak hedefe ulaşmasına yardımcı olabilir.

HelloApplication:

Bu sınıf, bir JavaFX uygulaması olarak tasarlanmış bir oyunun ana ekranlarını ve oyun mekaniklerini içerir. Oyunun amacı, karakterin belirlenen hedeflere (hazine noktalarına) ulaşarak en kısa yolu bulması ve engelleri aşmasıdır. sayfa1 metodu, oyuna başlamak için hoşgeldin ekranını oluşturur. Bu ekranda oyuncuya oyun hakkında bilgi verilir ve başlama butonu bulunur. sayfa2 metodu, harita boyutu ve oyun özellikleri gibi bilgilerin girildiği ekranı oluşturur. sayfa3 metodu, oyun ekranını oluşturur ve oyunun ana mantığı burada gerçekleşir. Karakter hazine noktalarını toplamaya çalışırken, engelleri aşması gerekmektedir. sayfa4 metodu, oyunun sonuç ekranını oluşturur. Bu ekranda toplanan hazineler, keşfedilen engeller ve karakterin geçtiği koordinatlar gibi bilgiler görüntülenir. Oyunun çalışma mantığı, karakterin hedefe (hazineye) en kısa yolu bulmak için BFS (Breadth-First Search) algoritmasını kullanmasıdır. Oyun sırasında karakterin hareketleri ve engellerin durumu düzenlenir ve ekrana çizilir.

SONUÇ

Bu projede, Java programlama dili kullanılarak bir matris üzerinde otonom hareket eden ve engellerden kaçınarak hazineleri toplayan bir karakter kodlanmıştır. Projede, hareketli ve hareketsiz engeller oluşturulmuş, hazineler eklenmiş ve karakterin en kısa yolu bulması için BFS algoritması kullanılmıştır. Kullanıcı tarafından kontrol edilebilen bir uygulama geliştirilmiştir.

Psuedo Code(Kaba Kod):

```
class Engel implements Cloneable:
    Sabitler:
        - ad: string
        - imagePath: string
        - engelX: int
        - engelY: int
        - engelBoy: int
        - engelGenislik: int
        - engelArrayList: ArrayList<Engel>

    Metodlar:
        - Engel(imagePath, ad, engelX, engelY, engelBoy, engelGenislik)
        - getEngelX(): int
        - setEngelX(engelX)
        - getEngelY(): int
        - setEngelY(engelY)
        - getImagePath(): string
        - setImagePath(imagePath)
        - getAd(): string
        - setAd(ad)
        - getEngelBoy(): int
        - setEngelBoy(engelBoy)
        - getEngelGenislik(): int
        - setEngelGenislik(engelGenislik)
        - clone(): Engel


class HareketliEngel extends Engel:
    Sabitler:
        - hareketYonu: string
        - hareketBoyutu: int
        - kus: Engel
        - ari: Engel
        - hareketliEngeller: Engel[]
        - hareketliEngelArrayList: List<HareketliEngel>
        - hareketliEngelImageViews: List<ImageView>

    Metodlar:
        - HareketliEngel(imagePath, ad, engelX, engelY, engelBoy,
engelGenislik, hareketYonu, hareketBoyutu)
        - hareketliEngelOlustur(lokasyon, root)
        - hareketEttir()
        - Boyama()
        - getHareketYonu(): string
        - setHareketYonu(hareketYonu)
        - getHareketBoyutu(): int
        - setHareketBoyutu(hareketBoyutu)



class HareketsizEngel extends Engel:
    Metodlar:
        - HareketsizEngel(imagePath, ad, engelX, engelY, engelBoy,
engelGenislik)


class HareketsizEngelKis extends HareketsizEngel:
    Sabitler:
        - static Engel buzdagi
        - static Engel kutupayisi
        - static Engel penguen
        - static Engel kardanadam
        - static Engel buz
        - static Engel[] kisEngelleri
        - static ArrayList<HareketsizEngelKis> hareketsizEngelKisArrayList
        - static List<ImageView> hareketsizEngelKisImageViews
    Metodlar:
        - HareketsizEngelKis(imagePath, ad, engelX, engelY, engelBoy,
engelGenislik)
        - static KisEngelOlustur(lokasyon, root)


    class HareketsizEngelYaz extends HareketsizEngel:
    Sabitler:
        - static Engel agac
        - static Engel dag
        - static Engel duvar
        - static Engel kaya
        - static Engel gunes
        - static Engel[] yazEngelleri
        - static ArrayList<HareketsizEngelYaz> hareketsizEngelYazArrayList
        - static List<ImageView> hareketsizEngelYazImageViews

    Metodlar:
        - HareketsizEngelYaz(imagePath, ad, engelX, engelY, engelBoy,
engelGenislik)
        - static YazEngelOlustur(lokasyon, root)


class Hazine implements Cloneable:
    Sabitler:
        - private String ad
        - private String imagePath
        - private int X
        - private int Y
        - private int Boy
        - private int Genislik
        - private double karaktereUzaklik
        - static Hazine altin
        - static Hazine bakir
        - static Hazine gumus
        - static Hazine zumrut
        - static Hazine[] hazineler
        - static ArrayList<Hazine> hazineArrayList
        - static ArrayList<Hazine> hazineArrayListYedek
        - static List<ImageView> hazineImageViews
        - static ArrayList<Hazine> hazineArrayListToplamaSirasi

    Metodlar:
        - Hazine(imagePath, ad, x, y, boy, genislik, karaktereUzaklik)
        - static HazineOlustur(lokasyon, root)
        - static Hazine enYakinHazineBul()
        - getters and setters for attributes ad, imagePath, X, Y, Boy, Genislik
```

class Karakter:
Sabitler:
- ID: int
- ad: string
- imagePath: string
- boy: int
- genislik: int
- ilkKonumX: int
- ilkKonumY: int
- imageViewKarakter: ImageView
- hazineKesfedilen: ArrayList<Hazine>
- engelKesfedilen: ArrayList<Engel>

Metodlar:
- Karakter(ID, ad, imagePath, ilkKonumX, ilkKonumY, genislik, boy)
- KarakterOlustur(lokasyon, root)
- karakterHareket()
- setID(ID)
- setAd(ad)
- setImagePath(imagePath)
- getBoy(): int
- getGenislik(): int
- getIlkKonumX(): int
- setIlkKonumX(ilkKonumX)
- getIlkKonumY(): int
- setIlkKonumY(ilkKonumY)

class Kordinat:
Sabitler:
- int x
- int y
- static ArrayList<Kordinat> kordinatArrayListKarakter

Metodlar:
- constructor Kordinat(x, y)

class Lokasyon:
Sabitler:
- int X
- int Y
- static int[][] KORDINATLAR

Metodlar:
- void KarakterKordinatYaz(int boy, int genislik, int solUstKordinatX, int solUstKordinatY)
- void HazineKordinatYaz(int boy, int genislik, int solUstKordinatX, int solUstKordinatY)
- void HareketliEngelKordinatYaz(int boy, int genislik, int solUstKordinatX, int solUstKordinatY, int hareketBoyutu, String hareketYonu)
- void HareketsizEngelYazKordinatYaz(int boy, int genislik, int solUstKordinatX, int solUstKordinatY)
- void HareketsizEngelKisKordinatYaz(int boy, int genislik, int solUstKordinatX, int solUstKordinatY)
- void HaritaMatrisYazdir()
- int Kontrol(int x1, int x2, int x3, int x4, int y1, int y2, int y3, int y4)
- void MatrisiBirle()
- Object clone()

class Sis:
Sabitler:
- String ad
- String imagePath
- int X
- int Y
- static Sis sis
- static ArrayList<ImageView> sisArrayListImageView

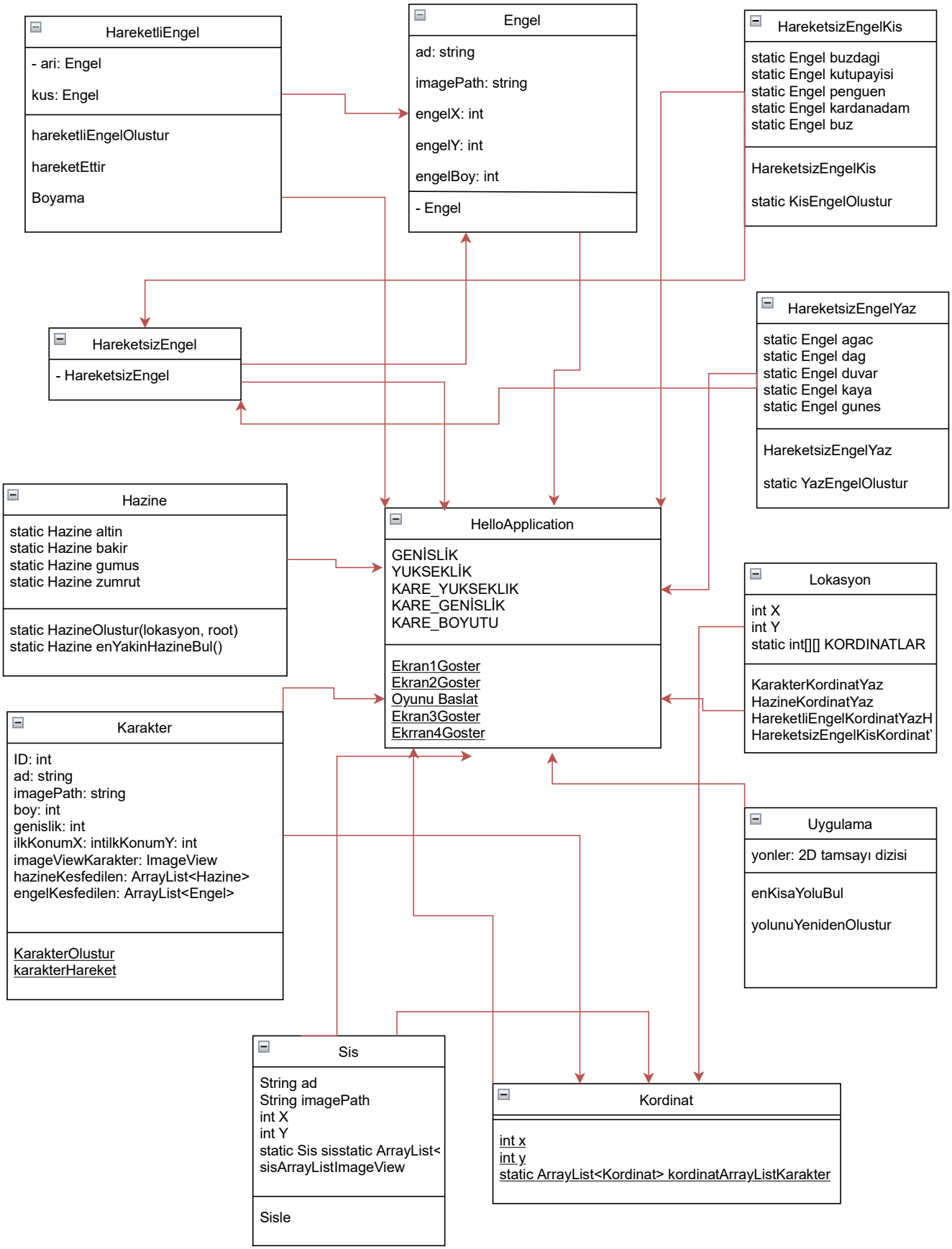
Metodlar:
- void Sisle(Group root)
- String getAd()
- void setAd(String ad)
- String getImagePath()
- void setImagePath(String imagePath)
- int getX()
- void setX(int x)
- int getY()
- void setY(int y)
- Engel clone()

class Uygulama:
Sabitler:
- yonler: 2D tamsayı dizisi

Metodlar:
- enKisaYoluBul(harita, baslangic, hedef): harita üzerinde başlangıç ve hedef noktalar arasındaki en kısa yolu bulan metot
- yolunuYenidenOlustur(parentMap, baslangic, hedef): en kısa yolun parentMap kullanılarak yeniden oluşturulmasını sağlayan metot

Kaynaklar

[1] https://youtu.be/W9F8fDQj7Ok?si=zfIfYmXTVJNi_l43
[2] https://youtu.be/KiCBXu4P-2Y?si=eVViIGN1rj1ahVMt
[3] https://youtu.be/D14YK-0MtcQ?si=itNifDwezZvYg6Zj
[4] https://youtu.be/H9WjCyPFOug?si=ejZWNV1cILlamkTR


HareketliEngel
- ari: Engel
kus: Engel
hareketliEngelOlustur
hareketEttir
Boyama
Engel
ad: string
imagePath: string
engelX: int
engelY: int
engelBoy: int
- Engel
HareketsizEngelKis
static Engel buzdagi
static Engel kutupayisi
static Engel penguen
static Engel kardanadam
static Engel buz
HareketsizEngelKis
static KisEngelOlustur
HareketsizEngel
- HareketsizEngel
HareketsizEngelYaz
static Engel agac
static Engel dag
static Engel duvar
static Engel kaya
static Engel gunes
HareketsizEngelYaz
static YazEngelOlustur
Hazine
static Hazine altin
static Hazine bakir
static Hazine gumus
static Hazine zumrut
static HazineOlustur(lokasyon, root)
static Hazine enYakinHazineBul()
HelloApplication
GENİSLİK
YUKSEKLİK
KARE_YUKSEKLIK
KARE_GENİSLİK
KARE_BOYUTU
Ekran1Goster
Ekran2Goster
Oyunu Baslat
Ekran3Goster
Ekrran4Goster
Lokasyon
int X
int Y
static int[][] KORDINATLAR
KarakterKordinatYaz
HazineKordinatYaz
HareketliEngelKordinatYazH
HareketsizEngelKisKordinat'
Karakter
ID: int
ad: string
imagePath: string
boy: int
genislik: int
ilkKonumX: intilkKonumY: int
imageViewKarakter: ImageView
hazineKesfedilen: ArrayList<Hazine>
engelKesfedilen: ArrayList<Engel>
KarakterOlustur
karakterHareket
Uygulama
yonler: 2D tamsayı dizisi
enKisaYoluBul
yolunuYenidenOlustur
Sis
String ad
String imagePath
int X
int Y
static Sis sisstatic ArrayList<
sisArrayListImageView
Sisle
Kordinat
int x
int y
static ArrayList<Kordinat> kordinatArrayListKarakter