

# KOCAELİ ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ PROGRAMLAMA

## LABARATUVARI 1.PROJE

190202040

Muhammed Koç

190202064

Osman Şafak

### 1-)ÖZET:

Bu projede hocalarımızın isteği üzerine java üzerinden 2 boyutlu bir oyun yaptık. Bize sunulan dillerden javayı seçmemizin sebebi, nesneye yönelik programlama konusunda sağladığı kolaylıklar oldu. Projede swing arayüzünü kullandık. Projede 2 ayrı JFrame ve 3 ayrı Jpanel kullandık. Oyunun amacı txt den seçilmiş olan belirli düşman veya düşmanların en kısa yol algoritması olan dijkstrayı kullanarak şirinlerin her hareketinden sonra onları yakalamak. Şirinin amacı ise kaybetmemek için rastgele yerlerde oluşan altın ve mantarları toplayarak şirineye ulaşmak. Oyun ise oyuncu şirineye ulaşmasıyla veya puanı 0 veya 0 in altına geldiğinde sonlanır ve oyuncu puanı ekrana yazdırılır.

### 2-)GİRİŞ:

Projede 16 farklı sınıf , her sınıfta constructor metotlarıyla getter-setter metotları bulunmakta.Kısaca sınıflara değinelim:

#### A-)Karakter Sınıfı:

Bu sınıfın Oyuncu ve Düşman olmak üzere 2 alt sınıfı mevcut. Hem oyuncunun hem de düşmanların sahip olması gereken isim,ID,oyuncunun yerini

belirten x ve y değişkenleri gibi özellikler burda mevcut.

#### B-)Oyuncu Sınıfı:

Kullanıcının yöneteceği karakterin özelliklerinin bulunduğu sınıf. Karakter sınıfından aldığı değişkenlere ek olarak, hız ve skor adlı 2 değişkene sahip.Gözlüklü ve Tembel olmak üzere 3 alt sınıfı bulunmakta.Bu alt sınıflardan ikisi farklı niteliklere sahip avatar. Diğer sınıf ise Puan isimli sınıf. Karakter sınıfına ek olarak bir de puanıGoster metodu bulunmakta.Bu metot sayesinde oyuncunun sahip olduğu skor sürekli güncellenip kullanıcıya gösterilmekte.

#### C-)Gözlüklü Sınıfı:

Kullanıcının seçebileceği 2 avatardan biri. Bu karakter tembel şirinden daha hızlı olmasına rağmen engellere takılma ihtimali de yüksek.Aynı zamanda yerde bulunan yıldız- ları veya mantarı toplaması daha zor.

#### D-)Tembel Sınıfı:

Bir diğer avatar olan tembel şirin yavaş olduğu için düşmanlara kolaylıkla yakalana- bilmekte. Diğer yandan engellere takılmadan , bolca yıldız ve mantar toplayarak hızla skor kazanabilmekte.

#### **E-)Obje Sınıfı:**

Skor kazandıran objeleri temsil eden bu sınıfın , Altın ve Mantar olmak üzere 2 alt sınıfı mevcut. Objelerin skora ne kadar katkıda bulunacağını belirlemek için değer, ne kadar haritada bulunabileceklerini anlamak içinse süre isminde 2 değişkeni bulunmakta.

#### **F-)Altın Sınıfı:**

Oyuncuya 5 puan kazandıran altın nesnelerinin üretildiği bu sınıf , Obje sınıfından aldığı değişkenlerin yanında , haritadaki konumunu belirtmek için x ve y değişkenlerine de sahip.

#### **G-)Mantar Sınıfı:**

Oyuncuya 15 puan kazandıran mantar nesneleri bu sınıftan üretiliyor. Skora katkısı fazla olduğundan daha uzun sürede ve daha az sayıda haritada belirtmekte. Haritadaki konumunun tutulması için x ve y değişkenlerine sahip.

#### **H-)Düşman Sınıfı:**

Kullanıcıyı oyun boyunca kovalayacak karakterlerin ana sınıfı. Azman ve Gargamel isminde 2 alt sınıfı mevcut. Karakter sınıfından aldığı değişkenlerin yanı sıra hasar ismindeki değişkene de sahip. Bu değişken sayesinde düşman , oyuncuyu yakalarsa ne kadar hasar verebileceğini anlamış oluyoruz.

#### **I-)Gargamel Sınıfı:**

Oyunda karşılaşılabileceğimiz 2 düşmandan biri. Gargamel , azmandan daha hızlı olmakla birlikte size çarptığında oyuncu 15 puan kaybeder.

#### **J-)Azman Sınıfı:**

Bir diğer düşman olan azman , gargamelden hem daha yavaştır, hem de daha az hasar verir.Size çarparsa 5 puan kaybedersiniz.

#### **K-)Lokasyon Sınıfı:**

Adından da anlaşılacağı üzere , düşmanların ve ya oyuncunun konumunu tutabilecek bir sınıf.

#### **L-)Puan Sınıfı:**

Oyuncunun skorunu tutmak için kullanılan sınıf.

#### **M-)Node Sınıfı:**

Dijkstra algoritmasında kullanılacak düğüm nesnelerini oluşturmak için kullanılacak bir sınıf.İçinde isim, önceki düğümü gösteren öncekiNode , konumu belirten x ve y değiş- kenleri , komşu düğümleri gösteren komşular değişkeni ve mesafeyi gösteren yol isimli değişken bulunmakta.

#### **N-)Kısayol Sınıfı:**

Düşmanın oyuncuya ulaşması için gidebileceği en kısa yolu bulmasını sağlayan sınıftır. Bunu yaparken Dijkstra algoritmasını kullanır. Değişken olarak son ve geçici isminde 2 node , komşutut , komşutut2 ve dolu isminde 3 node listesi bulundurur.

#### **O-) Panel Sınıfı:**

Programın arayüzünün ayarlandığı sınıftır. Ayrıca haritanın sürekli çizdirilmesi , nes- nelerin harita üzerindeki yerlerinin sürekli güncellenmesi , kullanıcının

girdisine göre ava- tarın hareket etmesi burdan ayarlanır.

### **P-)Test Sınıfı:**

Projedeki tüm sınıflardan nesnelerin üretildiği , main metodunun bulunduğu sınıftır.

### **3-)YÖNTEM:**

Projede toplamda 16 farklı sınıf kullandık. Bunların genel özelliklerine giriş kısmında yer vermiştik.2 sınıfı dijkstra algoritmasını gerçekleştirebilmek için oluşturduk.Bu sınıflar Node ve Kısayol isimlerine sahip.

İki temel sınıfımız mevcut. Bunlar Test ve Panel sınıfları. Test sınıfında nesneler oluş- turulurken , Panel sınıfında oyunun arayüzü ayarlandı. Txt dosyasından okuduğumuz düşman ve harita bilgileri öncelikle test sınıfında belirlendi. Aynı zamanda kullanıcı tarafından seçilen karakterin bilgileri de tutuldu. Tüm bu bilgiler Panel sınıfını oluşturulurken , constructor metoduna aktarıldı. Bu bilgiler sayesinde düşmanın ve oyuncunun konumları , haritadaki duvar ve yerlerin konumları , altın ve mantar objelerinin konumları , sürekli güncellenip paint ve repaint metotları yardımıyla çizdirildi. Kullanıcıdan klavye üzerinden sürekli bir input girişi beklendi ve bu girişler yön tuşları olduğunda oyuncunun hareket etmesi ayarlandı.

Düşman karakteri de oyuncu her hareket ettiğinde , öncelikle dijkstra algoritması yardımıyla en kısa yolu hesaplayacak , sonra da sahip olduğu hıza göre o yolda ilerleyecek şekilde tasarlandı. Oyuncunun kazanması veya kaybetmesi durumunda sahip olduğu skorun ekranda gösterilmesi ayarlandı. Bu sayede oyunun bittiği anlaşılır oldu.

### **Zaman Karmaşıklığı**

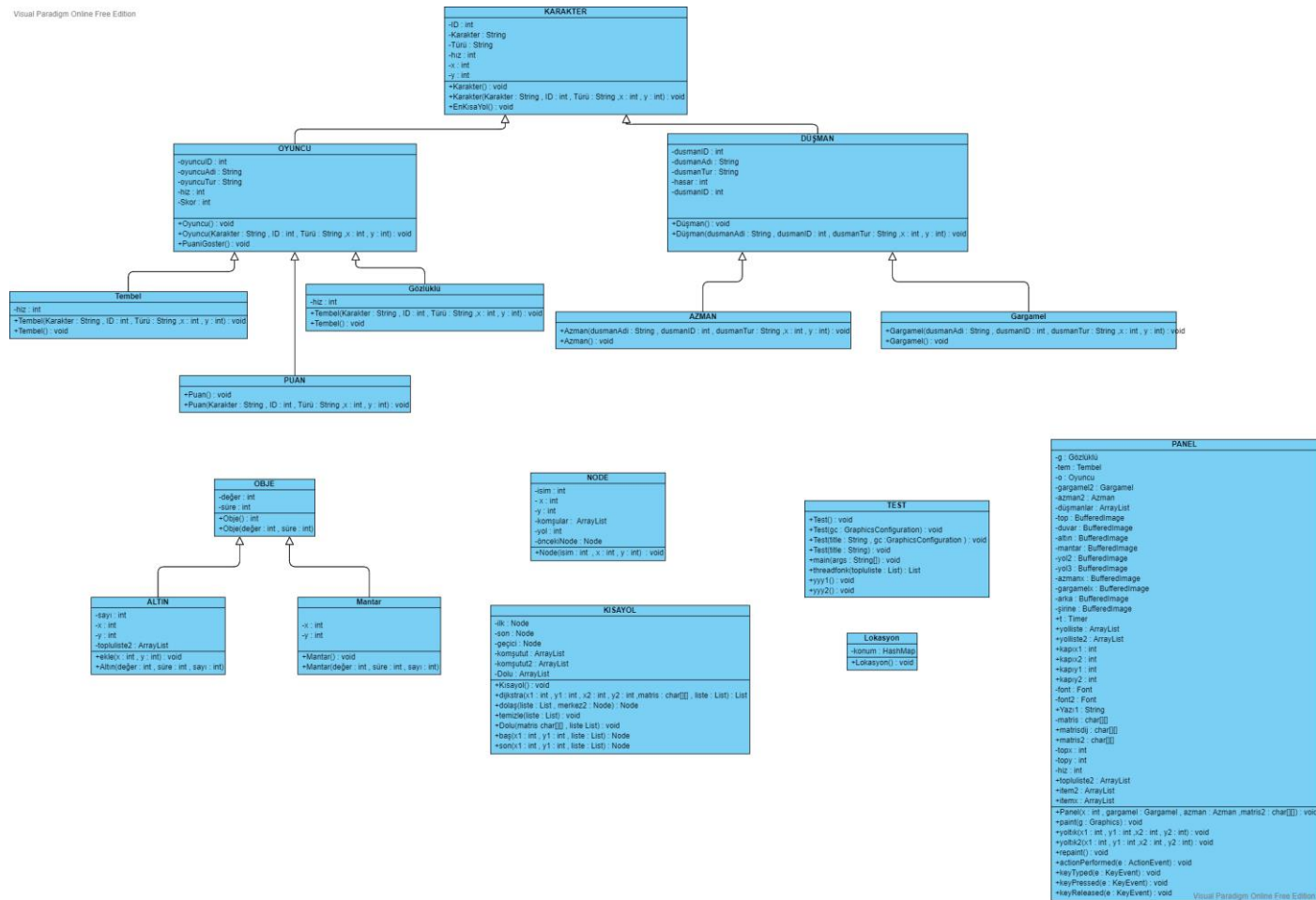
Zaman karmaşıklığı eğer harita sabit alınmazsa çünkü raporda sabit harita üzerinde çalışılacağı özellikle belirtilmişti haritanın matrisini okumasından dolayı  $O(n^2)$  ve dijkstra algoritmasındaki komşulara ulaşmak için listeleri birbirine aktarırken ve listeleri birbirine aktarma işleminde devamlılığını sağlamak için extra bir döngüye alırken  $O(n^2)$  olarak hesaplanmıştır. Projede her ne kadar birden fazla ikili döngü olsada bunlar birbirlerini etkilemediği için  $O(n^2)$  yi geçememiştir.

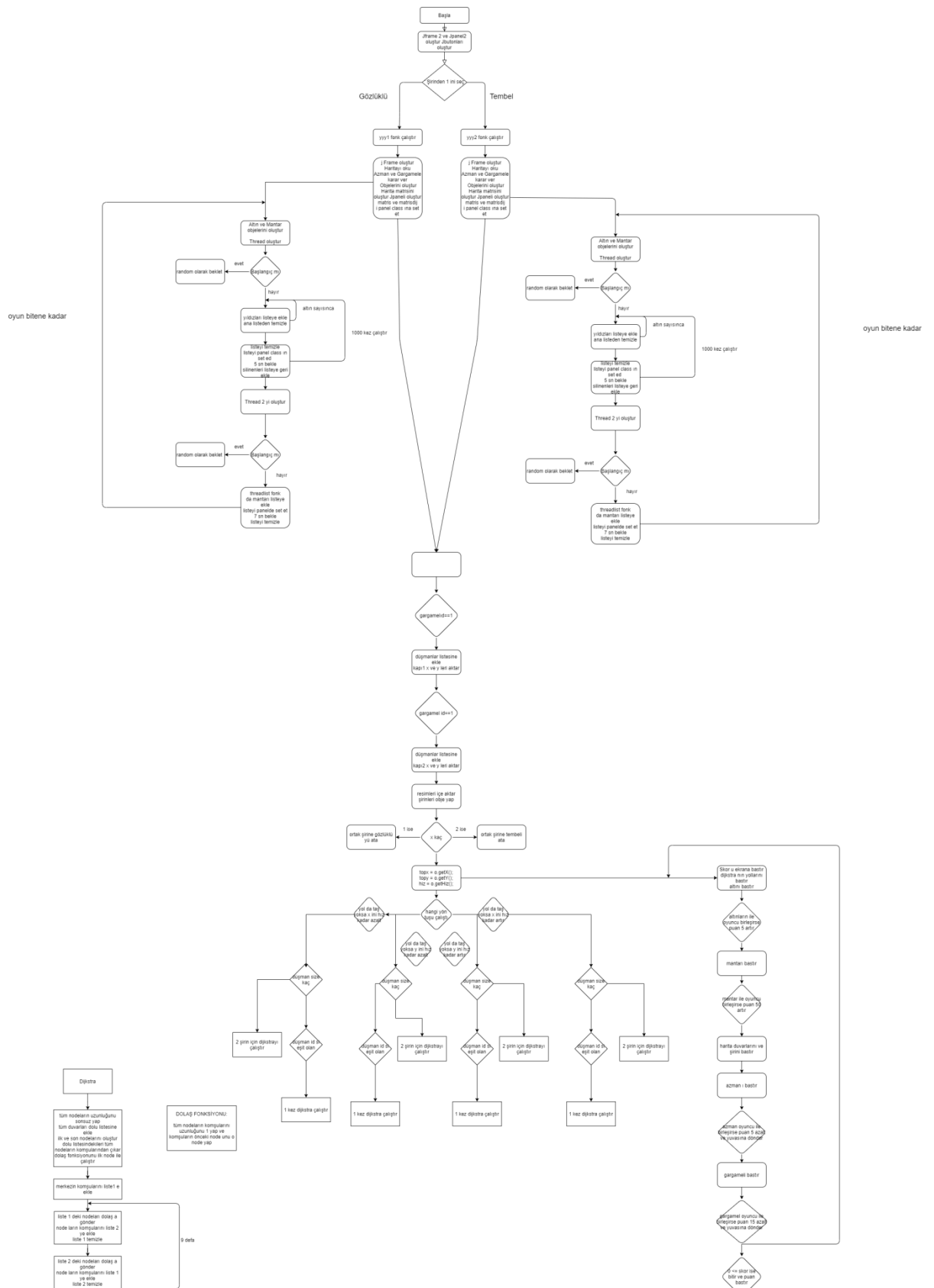
### **Bellek Karmaşıklığı**

Kullanılan algoritma (Dijkstra) daki bellek karmaşıklığı her bir Node classı için  $32n$  dir.

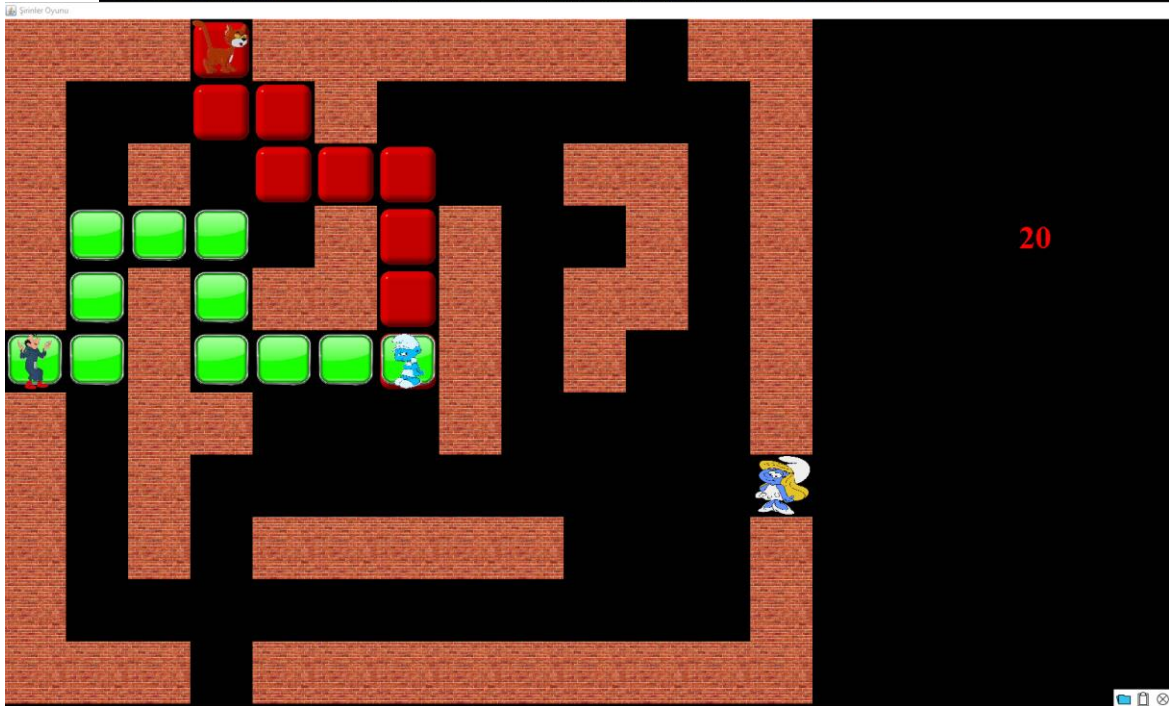
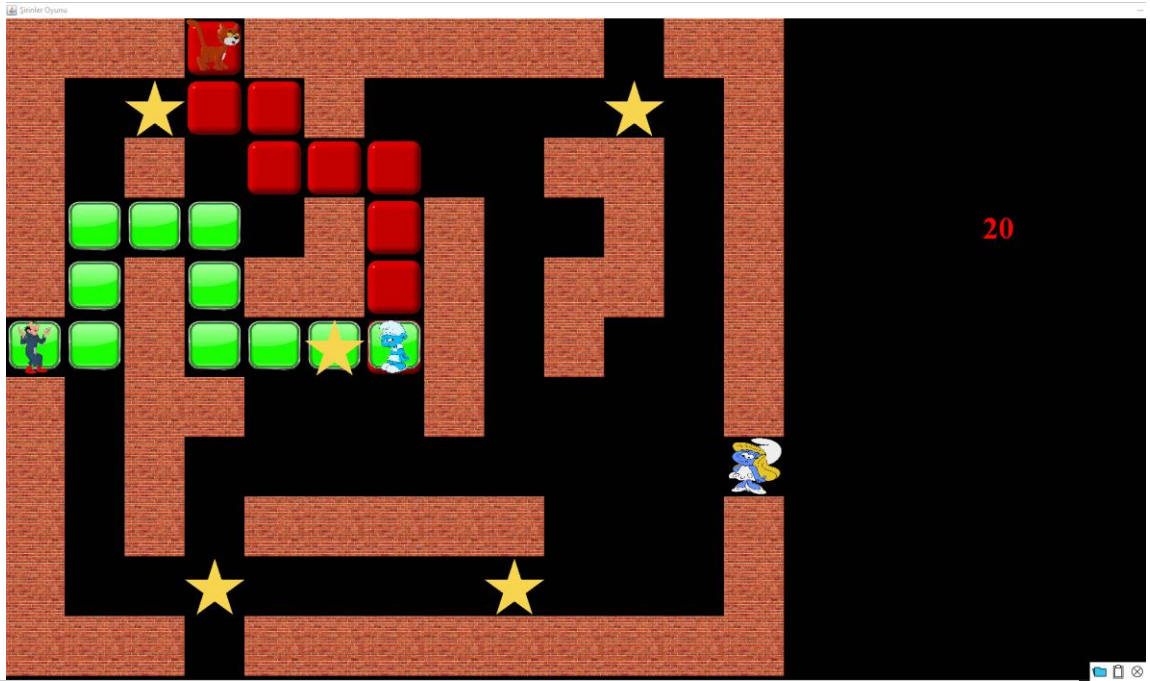
En kısa yol class ında ise  $(32n)*3+3n+12$  dir.

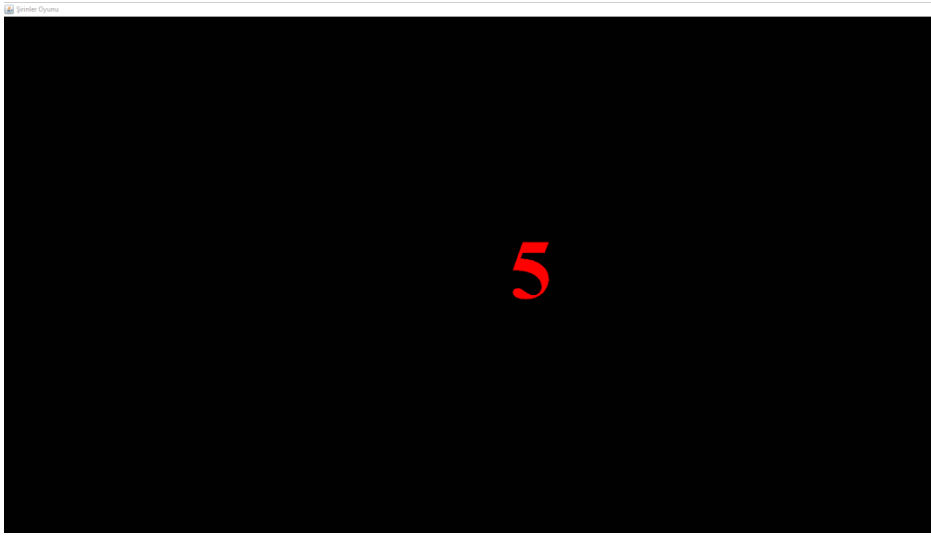
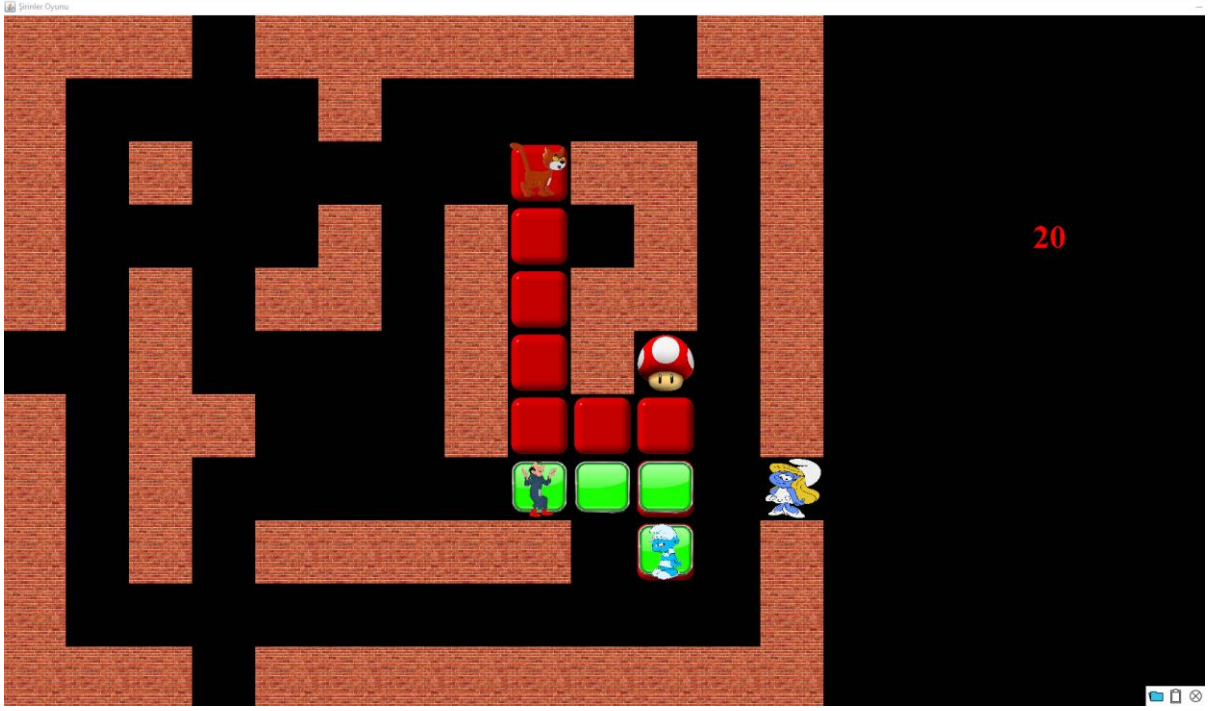
## Visual Paradigm Online Free Edition





#### 4-)DENEYSEL SONUÇLAR:





##### 5-)KARŞILAŞILAN PROBLEMLER:

Uzaktan eğitim vasıtasıyla grup halinde yapılan bir proje olunca haliyle grup üyeleri arasında iletişim bozukluğu yaşansada sonradan zoom gibi programlar vasıtasıyla iletişim sağlanmış olup ekip halinde çalışma yeteneği geliştirildi.

Panel class ın objesi oluşturulduktan sonra ona veri gönderildiğinde senkronizasyon sorunu olduğu ve verinin okunmadığı keşfedilince class ın constracter i kullanıldı.

## 5-)SONUÇ:

Java ile 2d oyun yapma konusunda temel bilgiler edinildi. Ayrıca arayüz tasarımı bilgileri tazelandı ve geliştirildi. Dijkstra algoritmasının mantığı hatırlandı ve kodlaması öğrenildi. Java dosya işlemleri bilgileri de tekrarlandı.

## 6-)KAYNAKÇA:

- <https://youtu.be/q3yKyE19OR0>

- <https://youtu.be/cLvFU1ilwwg>

<https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-in-java-using-priorityqueue/>

<https://www.baeldung.com/java-dijkstra>

<https://www.softwaretestinghelp.com/dijkstras-algorithm-in-java/>

<https://stackabuse.com/graphs-in-java-dijkstras-algorithm/#second-example>

<https://ssaurel.medium.com/calculate-shortest-paths-in-java-by-implementing-dijkstras-algorithm-5c1db06b6541>

<https://www.techiedelight.com/lee-algorithm-shortest-path-in-a-maze/>

[Java Swing Tutorial - javatpoint](#)

<https://www.youtube.com/watch?v=XTIYJU8uz0&t=2s>

<https://www.youtube.com/watch?v=Yw0p7tTFAoc>