KOCAELİ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ



PROGRAMLAMA LABORATUVARI II 1.PROJE

-ŞİRİNLER-

200202098 M.Numan MERCAN

Özet

Bu rapor Programlama Laboratuvarı II Dersinin 1.Projesisini açıklamak ve sunumunu gerçekleştirmek amacıyla oluşturulmuştur.

Raporda projenin tanımı, isterleri, yapım aşaması, kullanılan araç ve yöntemler, kod parçacıkları vb. bulunmaktadır.Proje aşamasında yararlanılan kaynaklar raporun son bölümünde bulunmaktadır.

Giriş

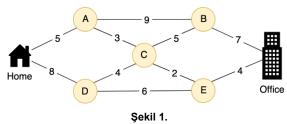
Projede istenen , bir labirentte düşmana yakalanmadan Şirine ' nin bulunduğu yere ulaşmak.

Tabi bu 1 e 1 atak hakkıyla gerçekleşen bir oyun bu yüzden oyuncu atak hakkını kullandıktan sonra düşmanın atak hakkını beklemek zorundadır.

Oyuncu oyuna belli bir puan (can) ile başlar, eğer düşman hamlesi oyuncunun olduğu bloğa erişirse oyuncudan belirlenen miktarda puan(can) eksiltilir ve düşman başladığı ilk bloğa geri döner, eğer oyuncu puanları bitmeden Şirine 'ye ulaşamazsa oyuncu oyunu kaybeder.

Burada sonradan açıklanacak bir şeyi hatırlatmakta fayda var,oyuncu ve düşman özellikleri karaktere göre değismektedir.

Yine ilerde açıklanacağı gibi proje de eksik kalan ve önemli bir kısmı içeren problem düsmanın oyuncuyu vakalamak icin algoritma kullanacağı Dijkstra. Bu algoritma en kısa yol algoritmalarından birisidir ve yol düğümlerle belirlenen çeşitli maliyetlerini hesaplayarak en az maliyetl yolu tercih eder.Boylece zorlu ve akıllı denebilecek bir rakip ile karşı karşıya kalmış oluruz.Şekil 1 de görüleceği gibi dijkstra algoritması bu tür yol seçimlerinde verimli kullanılabilir.



İlerleyiş-Yöntem

- Başlamadan Önce

Projeye ve kodlamaya iyice girmeden önce kullanılacak araç ve metodları , en önemlisi olarak arayüz oluşturmak için gerekli bazı tutorialları araştırdım.Kod ile nesneleri vs hareket ettirmeyi , hareketi klavye eventlerine bağlamayı vs gibi temel bilgileri öğrendim.

- Başlangıç

Projeye başlarken öncelikle problemleri anlamaya ve mantığı kavramaya çalıştım,tamamen başarılı olduğum söylenemesede belli bir mantık edindikten sonra kodlama aşamasına geçtim.

- İlerlevis

İlk olarak istenen sınıfları , içerecekleri değşkenleri , metodları ve sınıflar arası hiyerarşik yapıyı oluşturdum.

Absraction,İnheritance,Encapsulation vs gibi temel kavramları kullanarak oluşturduğum sınıfları son olarak arayüzümde kullanabilmeliydim.

2 farklı arayüz sınıfı açtım.Bunlardan ilki MyFrame adlı JFrame sınıfını extend ettiğim pencere yapım ve ikinci arayüz sınıfım olan Oyun adlı JPanel sınıfını extend etmis ovun haritasının oluşturulup oynanacağı Frame in üstüne insa edileceği bir Panel yapım. Tabii ki bu arayüz yapısına klavve eventlerini KeyListener sınıfını eklemek için de implement ettim.

Not: Burada classlarımı tek tek yazarak açıklamadım çünkü classları en güzel şekliyle açıklayan bir UML diyagramı oluşturdum.Raporun ilerleyen kısımlarında onu göreceksiniz.

Projede ilerlerken kaydettiğim bazı aşamaları da görsel olarak paylaşıyor olacağım.

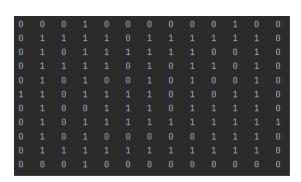
- Arayüzü Oluşturma

Haritayı oluşturmak için önce "harita.txt" dosyasını Java da okuttum.Ardından dosyanın içinde bulunan matris kısmını labirenti oluşturmak için okudum. İlk 2 satırdaki veri düşmanın kim olacağı ve labirentin hangi kapısında başlayacağını belirten veriydi.Bu veriyide ayrı bir kısımda düşmanın yerini belirtmek için okudum.Haritayı "Paint()" kütüphanesi ile çizdirdim.

Şekil 2 de "harita.txt" deki matris kısmı bulunmaktadır

0 - > Duvarlar

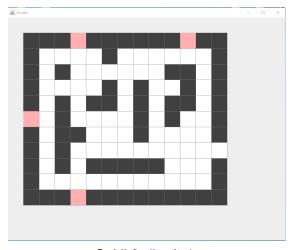
1 -> İlerlenebilecek Yollar



Şekil 2. (Matris)

Şekil 3 te dosyadan alıp oluşturulan haritamız bulunmaktadır.

Pembe ile belirtlimis bloklar düşmanların olası başlangıç base'i dir.



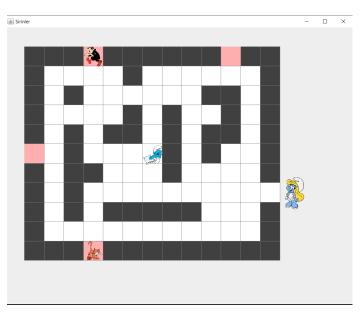
Şekil 3. (harita)

Ardından karakterleri sınıflara implement edip arayüzde yine Paint metodu ile çizdirdim.Karakterlerin belirli İmagelerini Bufferedlmage değişken türü ile her karakterin kendine ait sınıfında oluşturdum ve Oyun (JPanel) sınıfımda değişkenlerini oluşturarak resimlerini arayüzüme aktardım.

Karakter:Gargamel,Kapi:A Karakter:Azman,Kapi:D

Şekil 4. (Düşman ve base belirleme)

Şekil 5 te Artık dosyadan okunmuş bir haritanın son hali bulunuyor.

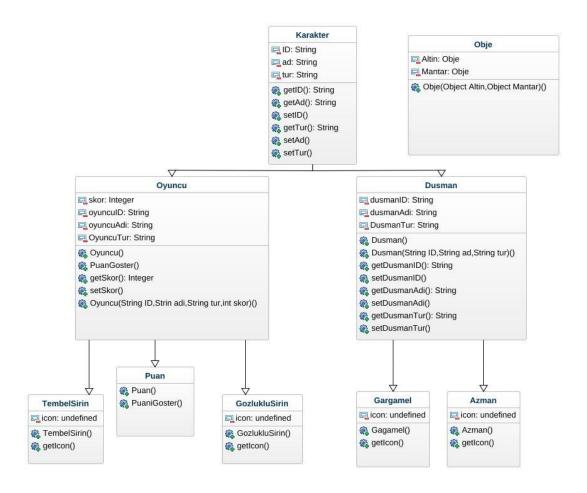


Şekil 5.(Haritanın tam hali)

- Sınıfların Oluşturulması

Sınıfları oluştururken UML diagramı çizimi işimizi oldukca açık ve kolay hale getiriyor.Fakat UML diagram çizimimi projeye başladığım aşamada çizmedim , yine de daha rahat görebilmek açısında ilerleyen vakitlerde diagramı uluşturdum ve yararlandım.

Şekil 6 da projenin UML diagramına yer verilmiştir.



Şekil 6.(UML diagram)

-Algoritma Oluşturma

Önce kaba kodumuzu ve bu kodun yardımıyla Akış Şemamızı oluşturduk.Bu aşama projenin ilerleyişi açısından çok büyük bir önem taşıyor , projemin tam olarak yetişmemesinin en büyük sebebi bu aşamayı da olması gerekenden geç yapmam.

Kaba kodları Şekil 7 ve 8 deki gibidir.

Şekil7.(ilk kısım)

```
//klavye hareketleri icin KeyListener implement edildi
//oyuncu sirinenin oldugu bloğa ulasana kadar ya da yanana kadar
if(basilanTus.kod=37)
if(basilanTus.kod=39)
if(basilanTus.kod=49)
düşmanın hareketleri için dijkstra algoritması

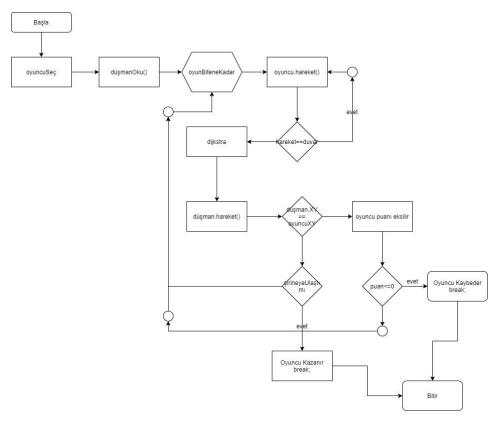
if(düşman.lokasyon=sirin.lokasyon)//sirinin ve duşmanın tipine göre puan eksilir
if(secilenşirin.lokasyon=sirine.lokasyon 8& puan > 0)
sout("Tebrikler oyunu kazandınız");
else if(secilensirin.lokasyon != şirine.lokasyon && puan >0)
//oyun devam etsin
else
sout("Maalesef kaybettiniz, lütfen bir daha deneyin");
```

Şekil8.(İkinci Kısım)

- Akış Şemaları

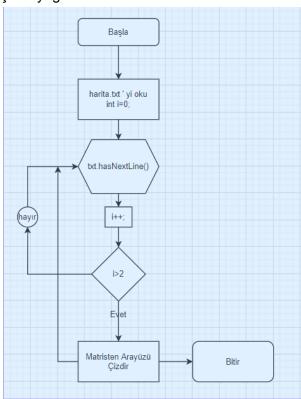
Proje için 2 adet akış şeması oluşturduk.Bunlardan biri Oyun için diğer şema ise arayüz için oluşturuldu.

Şekil 9 da Oyun için oluşturulan şema gözükmektedir.



Şekil9.(Oyun Akış Şeması)

Şekil 10 da Arayüz çizimi için oluşturulan şemayı göreceksiniz.



-Son Söz

Not : Proje de bazı eksiklikler bulunmaktadır bunlardan en önemlisi zaman yönetimi başarısızlığı dolayısıyla düşman yol algoritmasını ve haliyle hareket kodlarını oluşturamamam,bu yüzden projemin fazlasıyla eksikleri olduğunu bilerek fakat ona rağmen elimden geleni yaptığımı da belirterek projemi açıklamak istedim

Projenin belli aksaklık ve tecrübesizlikler dolayısıyla tam anlamıyla bitmediğini belirtmem gerek.Proje ye yeterli kavrama planlama yapılmadan oluşturma aşamasına girmiş olmam günler azaldıkça yanlış bir strateji izlediğimi kanıtladı. Projenin en büyük eksiği dijkstra algoritmasını entegre edememek.Bu eksik azımsanmayacak büyüklükte olduğunu kabul etmekte fayda var fakat geriye kalan isterlerin büyük çoğunluğunu proje tamamlamış olup sunuma çıkma cesaretini gösterecektir.

KAYNAKÇA

- https://www.youtube.com/watch?v=9XJicRt_Fal
- Sıfırdan Komple Java Geliştirici Kursu -> Mustafa Murat COŞKUN (Udemy) (Bazı unutulmuş kavramlar için)
- https://www.javatpoint.com/java-swing
- https://www.draw.io
- https://www.javatpoint.com/java-keylistener
- https://www.youtube.com/watch?v=dyrvXiMumXc&ab_channel=BroCod eBroCode -> Leo ONO
- https://github.com/irealva/maze-gui/blob/master/Maze.java
- https://stackoverflow.com/
- https://www.youtube.com/watch?v=q3yKyE19OR0&ab_channel=Suresh
 Srivastava -> for Dijkstra Algorithm

https://www.youtube.com/watch?v=uPXIsSd_wK4&Iist=WL&index=9&ab
 channel=coderodde

- https://github.com/yunusemregul
- https://www.geeksforgeeks.org/java-program-for-rat-in-a-maze-backtrac king-2/
- https://blog.emrahkahraman.com.tr/java-dosyadan-istenilen-satiri-okum a/
- https://medium.com/gokhanyavas/javada-i-o-i%CC%87%C5%9Flemleri-abde319f0544