

# ŞİRİNLER

Muhammed Taha METİN

## 1. ÖZET

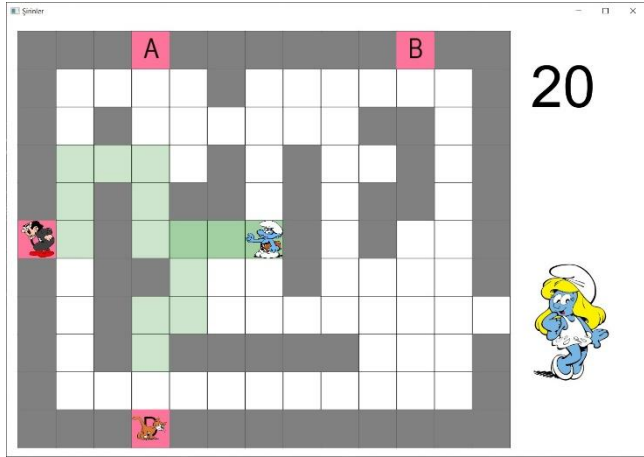
Proje, dijkstra algoritması ile en kısa yolu bulmayı ve kalıtım kullanarak nesne yönelimli programlamayı daha iyi kavramamızı amaçlar.

## 2. GİRİŞ

Oyun açıldığında karakter seçimi yapılır ve oyun ekranına geçer.



Karaktere göre her 1 adımda veya 2 adımda düşman karakterler en ekranda gösterilen en kısa yol üzerinden karaktere doğru gelir. Düşmanlar karaktere değerse çıktıkları kapıya doğru giderler ve kapıya ulaştıklarında tekrar karaktere doğru giderler. Oyunun amacı skoru 0'ın altına düşürmeden labirentin çıkışıındaki şirineye ulaşmaktır.



## 3. YÖNTEM

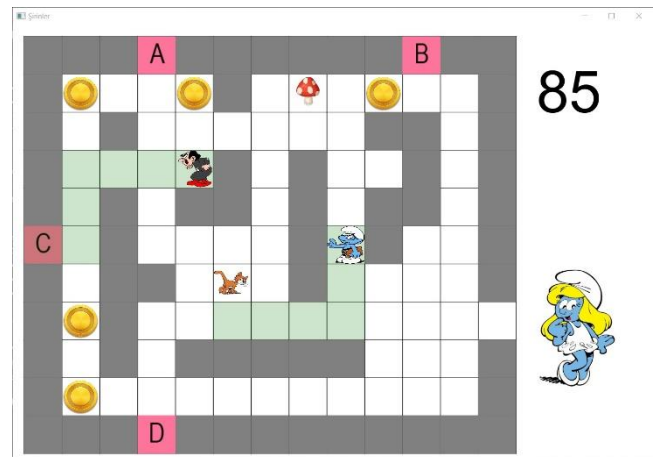
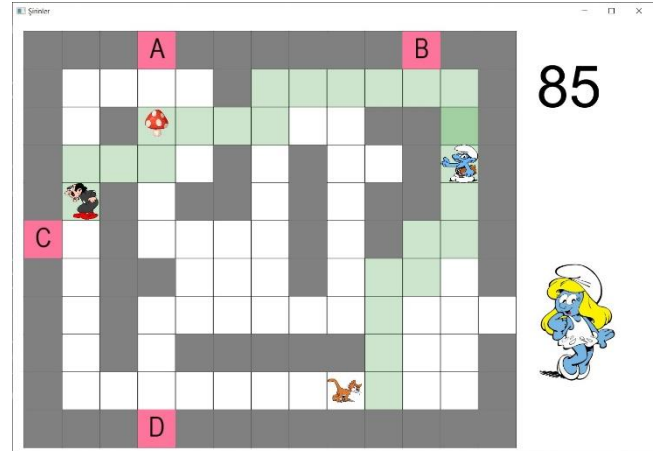
Program Java dilinde JavaFx kütüphanesinden yararlanılarak yapılmıştır.

İlk önce verilen metin dosyasından bilgilerin okunup uygun biçimde depolanması ayarlanmıştır. Sonra JavaFx kütüphanesinin yardımı ile labirent, düşmanlar ve oyuncu ekrana çizilmiştir. Sonrasında oyuncunun denetimli bir şekilde (labirentin dışına çıkmayacak şekilde) hareket etmesi ayarlanmıştır. Ardından Dijkstra algoritması hakkında araştırma yapılmıştır ve Dijkstra algoritmasının ağırlıksız graflar için kullanılan versiyonu olan düzenlenmiş bir BFS algoritması kullanılarak düşmanların en kısa yolu bulması sağlanmıştır. Sonrasında düşmanların en kısa yolda ilerlemesi ve o yolun ekrana çizilmesi işlemleri yapılmıştır. Ardından düşman ve nesnelerin skor üzerindeki etkileri ayarlanmıştır. Sonrasında mantar ve altınların zamana bağlı oluşumu ve kaybolması sağlanmıştır.

En son olarak sınıf arkadaşlarımdan birkaçı ile programın sınırları zorlanarak hata olup olmadığı kontrol edilmiştir.

## 4. DENEYSEL SONUÇLAR

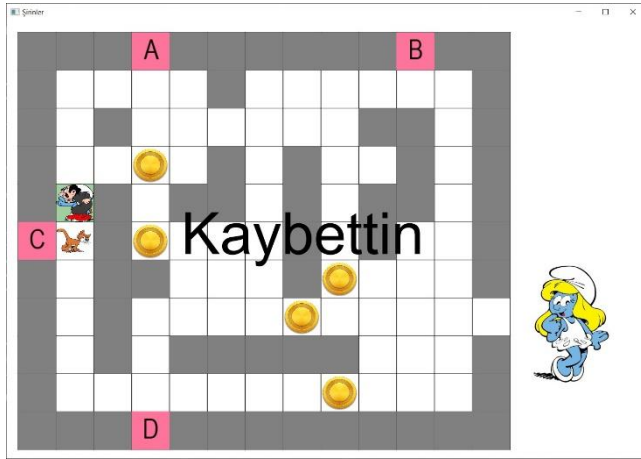
Oyun oynanırken alınan bazı ekran görüntüleri.



Kazanıldığında:



Kaybedildiğinde:



## 5. KABA KOD (PSEUDO CODE)

### Class Main

Void Start:

- Dosyayı okur.
- Labirentin çizgilerini çizer.
- Duvarları ve kapıları çizer.
- Şirineyi çizer.
- Dosyadan alınan bilgi ile düşmanları oluşturur.
- Dosyadan alınan bilgi ile noktaları ve komşularını tutan bir Lokasyon listesi oluşturur.
- Aşağı, yukarı ,sağ ve sol ok tuşlarına basıldığında gidilmek istenilen yerde duvar yoksa gidilmesini sağlar. Düşmanın türüne göre her 2 ya da 1 basışta hareket etmesi sağlanır.
- Pencerenin başlığı ayarlanır.
- İlk açılışta seçilen karakter oluşturulur ve düşmanın en kısa yolu çizilir.

Void PuanYazisiniOlustur:

- Puan yazısının yerini ve boyutunu ayarlar.
- Puan yazısının değerini oyuncudan alıp yazdırır.

Void AltinOlustur:

- 7-9 saniye aralığında bekleyip üst üste gelmeyecek şekilde altın oluşturur.
- Oluşturduğu altını objeler listesine ekler.
- AltinYokEt() metodunu çağırır.

Void AltinYokEt:

- 5 saniye bekleyip altınları yok eder.
- AltinOlustur() metodunu çağırır.

void MantarOlustur:

- 13-17 saniye aralığında bekleyip diğer objelerle üst üste gelmeyecek şekilde mantar oluşturur.
- MantariYokEt() metodunu çağırır.

Void MantariYokEt:

- 7 saniye bekleyip mantarı yok eder.
- MantarOlustur() metodunu çağırır.

Void EtkiletisimVarmi:

- Düşmanların konumları ile oyuncunun konumunu karşılaştırarak üst üste olup olmadıklarını kontrol eder. Üst üste iseler skoru azaltır.
- Tüm objeler ile oyuncunun konumunu kıyaslayarak üst üste olup olmadıklarını kontrol eder. Üst üste iseler skoru artırır ve objeyi siler.
- Çıkış kapısı ile oyuncunun koordinatlarını karşılaştırır. Oyuncu çıkış kapısına vardığında oyunu kazandın yazısını çıkarır.

Void DusmaniHaraketEttir:

- YollariSil() metodunu çağırır.
- Her düşman için EnKisaYoluCiz() ve EnKisaYoldallerle() metodlarını çağırır.

Void OyuncuyuOlustur:

- Yapılan seçime uygun karakteri ekrana ekler.

Void EnKisaYoldallerle:

- Hesaplanan en kısa yolda; düşman azman ise 1 , gargamel ise 2 adım ilerler.

Void EnKisaYoluCiz:

- Düşmanın konumunu bulup source değişkenine atar
- Düşman başlangıç kapısında ise hedefeUlasti değişkenini false yapar.
- Hedefe ulaştı değişkeni true ise dest(destination, hedef) değişkenini giriş kapısı yapar.

- false ise hedefi oyuncu yapar.
- BFS() metodunu çağırır.
- Ekrana yolu çizer

Void YollarıSil:

- Ekrana çizilen yolları siler.

Void BFS:

- BFS algoritmasından yararlanarak en kısa yolu bulur.

Void XYToScreen:

- Verilen x ve y değerlerinin ekranda denk düştükleri yerin değerlerini geri döner.

Void MoveUP:

- Görüntüyü 1 birim yukarı taşır.

Void MoveDown:

- Görüntüyü 1 bitin aşağı taşır

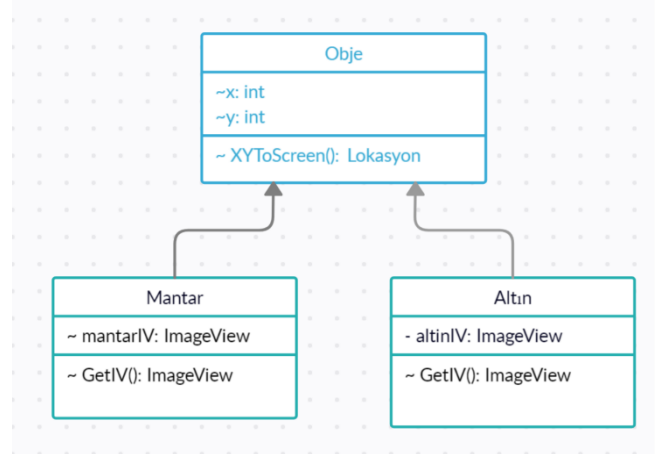
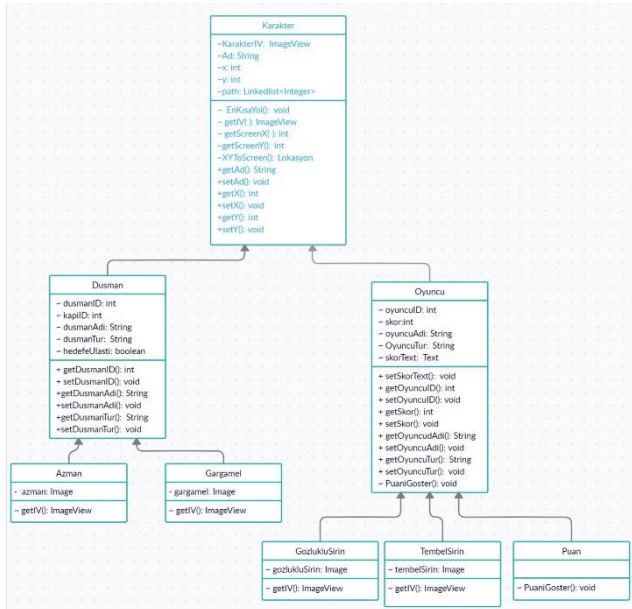
Void MoveRight:

- Görüntüyü bir birim sağa taşır.

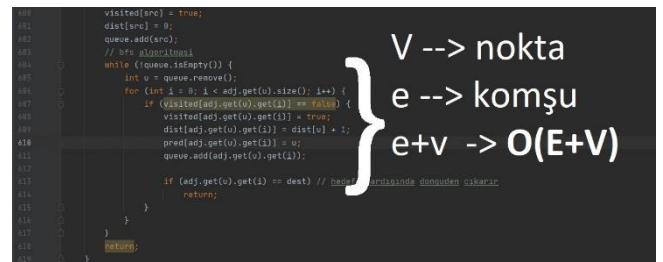
Void MoveLeft:

- Görüntüyü bir birim sola taşır.

## 6. UML SINIF DİAGRAMI



## 7. KARMAŞIKLIK



Zaman karmaşıklığı en büyük olan fonksiyon BFS() fonksiyonudur. Bu yüzden programın zaman karmaşıklığını bu fonksiyon belirler ve zaman karmaşıklığı  $O(E+V)$ 'dir.



Bellek karmaşıklığı en büyük olan kod start() fonksiyonunun içindeki matristeki gidilebilir her noktayı bir komşularını tutan bir lokasyon listesi haline getiren koddur. Bu yüzden programın bellek karmaşıklığını bu kod belirler ve bellek karmaşıklığı  $O(n)$ 'dir.

## 8. SONUÇ

Program verilen labirentte değişen düşman ve oyuncu konumları arasındaki en kısa yolu çizer. Program yapılırken kalıttan faydalanarak karakter sınıfları oluşturulmuştur ve kod tekrarı önlenmiştir. Nesne yönelimli programlama ve veri yapıları konularında pratik yapılarak daha iyi anlaşılması sağlanmıştır.

## 9. KAYNAKÇA

- [https://docs.oracle.com/javafx/2/ui\\_controls/button.htm](https://docs.oracle.com/javafx/2/ui_controls/button.htm)
- <https://docs.oracle.com/javafx/2/canvas/jfxpub-canvas.htm>
- <https://stackoverflow.com/questions/27203671/java-vafx-how-to-clear-the-canvas>
- <https://www.geeksforgeeks.org/javafx-button-with-examples/>
- <https://stackoverflow.com/questions/29463144/why-cant-i-use-the-same-image-twice>
- [https://www.tutorialspoint.com/javafx/javafx\\_2\\_d\\_shapes.htm](https://www.tutorialspoint.com/javafx/javafx_2_d_shapes.htm)
- [https://www.tutorialspoint.com/javafx/javafx\\_event\\_handling.htm](https://www.tutorialspoint.com/javafx/javafx_event_handling.htm)
- <https://stackoverflow.com/questions/13567019/creating-fxml-window-by-code-javafx>
- <https://stackoverflow.com/questions/55983186/how-can-i-make-one-image-over-another-in-javafx-and-change-it-again-in-an-action>
- <https://stackoverflow.com/questions/37200845/how-to-switch-scenes-in-javafx>
- [https://www.youtube.com/watch?v=PPAp\\_w\\_oXzw](https://www.youtube.com/watch?v=PPAp_w_oXzw)
- <https://www.youtube.com/watch?v=saLcT5UE-JM>
- <https://www.youtube.com/watch?v=ZtBMVxyD8A0>
- [https://www.youtube.com/watch?v=PDuciUID4\\_4](https://www.youtube.com/watch?v=PDuciUID4_4)
- <https://stackoverflow.com/questions/26454149/make-javafx-wait-and-continue-with-code>
- <https://stackoverflow.com/questions/16764549/timers-and-javafx>
- <https://stackoverflow.com/questions/54229373/why-thread-sleep-doesnt-work-accordingly-in-javafx>
- <https://stackoverflow.com/questions/42653597/java-vafx-draw-shapes-with-delay>
- <https://stackoverflow.com/questions/59732441/i-want-to-change-the-color-of-a-random-button-wait-1-sec-then-change-it-back-to>
- <https://stackoverflow.com/questions/40729967/drawing-shapes-on-javafx-canvas>
- <https://stackoverflow.com/questions/49881109/how-to-properly-execute-thread-sleep-in-javafx>
- [https://stackoverflow.com/questions/22047457/how-to-change-the-text-font size-in-javafx](https://stackoverflow.com/questions/22047457/how-to-change-the-text-font-size-in-javafx)
- <https://www.tutorialspoint.com/how-can-we-implement-a-timer-thread-in-java>
- <https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/>
- <https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/>
- <https://livebook.manning.com/book/javafx-in-action/chapter-5/13>
- <https://www.geeksforgeeks.org/shortest-path-unweighted-graph/>
- <https://www.baeldung.com/cs/graph-algorithms-bfs-dijkstra>