**Star Wars**

Arda Talu

Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü

170202037 [arda.starkiler@gmail.com](mailto:arda.starkiler@gmail.com)

**Özet**

Projenin amacı kullanıcıdan alınan hareket komutları ile labirentte bulunan karakteri yakalanmadan çıkışa ulaştırmaya çalışmaktır.

**1.Giriş**

Proje yazılırken Java dili kullanılmıştır. Programı kodlarken araç olarak Eclipse IDE ve Notepad++ kullanılmıştır.

**2.Algoritma**

Program başladığında kullanıcıdan karakter seçmesi istenir.

Sonrasında yön tuşları yardımıyla karakter hareket ettirilir ve çıkışa doğru gidilmeye çalışılır.

Bu esnada kötü karakterler kontrol ettiğimiz karakteri yakalamaya çalışır.

Kötü karakter yakalandığında iyi karakterin canı azalır.

Kötü karakterlerden DarthVader duvarlardan geçebilir.

Kötü karakterlerden KyloRen tek bir hamlede 2 kare ilerler.

Kötü karakterlerden Stormtrooper’ın özel bir yeteneği yoktur.

İyi karakterlerden MasterYoda 6 cana sahiptir.

İyi karakterlerden LukeSkywalker 3 cana sahiptir.

**3.Sözde Kod**

1. Başla
2. io.BufferedReader’ı import et
3. io.File’ı import et
4. io.FileNotFoundException’ı import et
5. io.FileReader’ı import et
6. io.IOException’ı import et
7. awt.Color’u import et
8. awt.Graphics’i import et
9. awt.event.KeyEvent’i import et
10. util.ArrayList’i import et
11. util.List’i import et
12. swing.JFrame’i import et
13. swing.SwingUtilities’i import et
14. ProLab21 ms = new ProLab21()
15. BufferedReader oku
16. String dosya = "Harita.txt"
17. Lokasyon loc = new Lokasyon()
18. Karakter cha = new Karakter()
19. Dene
20. oku = new BufferedReader(new FileReader(dosya));
21. String satir = oku.readLine()
22. int lineNumber = 0;
23. İken(satir != null)
24. Eğer(lineNumber > 2)
25. satir = oku.readLine()
26. lineNumber++
27. oku.close()
28. Yakala(IOException e)
29. Boolean YolAra(int[][] lab, int x, int y, List<Integer> yol)
30. Eğer(lab[x][y] == lab[9][13])
31. yol.add(x)
32. yol.add(y)
33. Döndür true
34. Eğer (lab[x][y] == 0)
35. lab[x][y] = 2
36. int dx = -1
37. int dy = 0
38. Eğer (YolAra(lab, x + dx, y + dy, yol))
39. yol.add(x)
40. yol.add(y)
41. Döndür true
42. dx = 1
43. dy = 0
44. Eğer (YolAra(lab, x + dx, y + dy, yol))
45. yol.add(x)
46. yol.add(y)
47. Döndür true
48. dx = 0
49. dy = -1
50. Eğer (YolAra(lab, x + dx, y + dy, yol))
51. yol.add(x)
52. yol.add(y)
53. Döndür true
54. dx = 0
55. dy = 1
56. Eğer (YolAra(lab, x + dx, y + dy, yol))
57. yol.add(x)
58. yol.add(y)
59. Döndür true
60. Döndür false

**4.Sonuç**

Kullanıcıya yön tuşları ile oynanabilen bir labirent kovalamaca oyunu yazılmıştır.

**5.Kaynakça**

**[1]**https://github.com/Stealthii/Pacman/blob/master/src/Maze.java

**[2]** https://github.com/leonardo-ono/JavaMazeSolverTest/blob/master/src/mazesolver/View.java

**[3]** https://github.com/leonardo-ono/JavaMazeSolverTest/blob/master/src/mazesolver/DepthFirst.java

**[4]**https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/javada-dosya-islemleri

**[5]** https://medium.com/gokhanyavas/javada-i-o-i̇şlemleri-abde319f0544

**[6]**https://www.youtube.com/watch?v=d8rU71OBkxU

**[7]** https://doors2020.wixsite.com/learncoding