Robot Sınıflandırılması

\*Ceren Genç, \*Berkehan Öztürk

\*Bilgisayar Mühendisliği Bölümü, Kocaeli Üniversitesi

Kocaeli, Türkiye

[cerengenc17@gmail.com](mailto:cerengenc17@gmail.com)

[berkehanozturk98@gmail.com](mailto:berkehanozturk98@gmail.com)

**Anahtar Kelimeler – “Gezgin”, “Manipülatör”, “Hibrit”, ”Engel”, ”Izgara”**

Özet – Programımız özellikleri kullanıcı tarafından girilen robotların sınıflandırıp, seçilen robota ait problem çözümünü gerçekleştirmektedir.

1.GİRİŞ

Programımız kullanıcıdan kaç adet robot girileceğini istemesi ile başlamaktadır. Girilecek robotun sınıfı, genel robot özellikleri ve robotun ait olduğu sınıfın özellikleri kullanıcı tarafından istenir. Birden fazla gezgin robot girişi yapıldığında gerekli hız karşılaştırmaları gerçekleştirilir. Manipülatör robot girişlerinde taşınılacak yükün kullanıcı tarafından alınmış olan yük kapasitesini geçip geçmediği kontrol edilir. Birden fazla manipülatör girişinde gerekli yük kontrolleri gerçekleştirilir. Tüm robotların bilgi girişleri bittikten sonra hangi robotla işlem yapılacağı kullanıcıya sorulur. İşlem yapılacak robotun hangi problem çözümüne ait olduğu kontrol edilip problem çözümüne başlanır. Gezgin robot türleri için kaç adet engel girileceği ve engellerin koordinatları kullanıcıdan istenir. Bu robotlar için hareket süresi içinde engelle karşılaşma durumları ve ızgaradan çıkma durumları kontrol edilir. Engellerle karşılaştığı durumda seçilmiş robot türünün engelleri geçebilme durumu kontrol edilip geçebiliyor ise engel geçme süresi hesaplanır. Engeller geçilemiyorsa, robot engelden önceki son hareketinde kalır ve toplam hareket süresi kullanıcıya gösterilip program sonlandırılır. Izgaradan çıkma durumlarında, robotun ızgaradan çıkmadan önceki son konumu, toplam hareket süresi hesaplanıp kullanıcıya gösterilir ve program sonlandırılır. Toplam hareket süresi hesaplanırken x=v\*t formülü kullanılır. Robot engelden geçebiliyor ise engelden geçme süreside hareket süresine eklenir. İşlem yapılacak robot manipülatör robot ise, engel girişi istenmez. Kullanıcıdan robotun konumu istenir. Yükün hareket ettirilmesi kollar yardımıyla gerçekleştirilir. Hareket mesafesinin kol uzunluğunu geçme ve ızgaradan çıkma durumu kontrol edilir. Izgaradan çıkıldıysa, yükün ızgaradan çıkmadan önceki son konumu ve hareket süresi kullanıcıya gösterilip program sonlandırılır. Kullanıcı tarafından girilen hareket mesafesi robotun kol uzunluğunu geçtiyse, yük taşınılabilecek son mesafeye taşınır ve hareket süresi kullanıcıya gösterilip program sonlandırılır. İşlem yapılacak robot Hibrit robot ise kaç adet engel girileceği, engellerin koordinatları ve robotun ilk yeri kullanıcıdan istenir. Hibrit robotun hareketli kısmı için robotun sabitleneceği yer kullanıcıdan istenir. Hareketli kısmın hareket esnasında engelle karşılaşma ve ızgaradan çıkma durumları kontrol edilir. Engelle karşılaşılmış ise engel geçme süresi hesaplanır. Manipülatör robot olarak devam edeceği koordinata gelindikten sonra, kullanıcı tarafından girilen mesafelerin kol uzunluğunu geçme durumu kontrol edilir. Izgaradan çıkma durumlarında son konum ve hareket süresi kullanıcıya gösterilerek program sonlandırılır. Hareket süresi hesaplanırken hibrit robotun gezgin robot gibi davrandığı sürede geçen zaman, varsa engel geçme süresi, manipülatör robot olarak davranılan sürede geçen zaman ve sabitlenme süresi toplanır.

2.TEMEL BİLGİLER

Programımız Java dilinde yazılmıştır. Geliştirme ortamı olarak “NetBeans 8.2” kullanılmıştır. Windows 10 ve Windows 7 işletim sistemlerinde sorunsuz olarak çalıştırılmıştır.

3.MİMARİ

Gerekli kalıtımları sağlayabilmek için arayüz ve sınıf yapıları kullanılmıştır. Robot arayüz olarak tanımlanmıştır. gezginRobotlar sınıf olarak tanımlanıp Robot arayüzünden implement edilmiştir. tekerlekliRobot, paletliRobot, spiderRobot yapıları sınıf olarak tanımlanmış olup gezginRobotlar sınıfından extends edilmiştir. gezmeyenRobotlar arayüz olarak tanımlanmış ve Robot arayüzünden extends edilmiştir. seriRobot ve paralelRobot sınıf olarak tanımlanıp gezmeyenRobotlar’dan implement edilmiştir. hibritRobot sınıf olarak tanımlanıp gezginRobotlar’dan extends, gezmeyenRobotlar’dan implement edilmiştir. Grafik ekranın kullanılabilmesi için sekil sınıfı tanımlanıp JPanel sınıfından extends edilmiştir. Main fonksiyonundan sekil sınıfı çağırılıp gerekli tüm işlemler sekil sınıfı içerisinde yapılmıştır.

4.KULLANILAN FONKSİYONLAR

*void problem1():* İşlem yapılacak robot gezginRobotlar sınıfına ait ise çalışır. Toplam hareket süresi hesaplanıp kullanıcıya gösterilir.

*void problem2():* İşlem yapılacak robot gezgmeyenRobotlar sınıfına ait ise çalışır. Toplam hareket süresi hesaplanıp kullanıcıya gösterilir.

*void problem3():* İşlem yapılacak robot hibritRobot sınıfına ait ise çalışır. Toplam hareket süresi hesaplanıp kullanıcıya gösterilir.

*dobule engelGecme():*İmzası farklı olan iki tanımlaması vardır. İşlem yapılacak robot türüne uygun olan çağırılmaktadır. Tanımlanmış robotun engel geçme süresi hesaplanır.

5.SONUÇ

İşlem yapılacak robota ait problem çözümleri gerçekleştirilmiştir. gezginRobotlar için hız kontrolü aynı anda girilmiş en fazla 3 robot için yapılmaktadır. gezmeyenRobotların yük taşıma sıralamaları için gerekli olan kontrol aynı anda girilmiş 2 robot için yapılmaktadır.

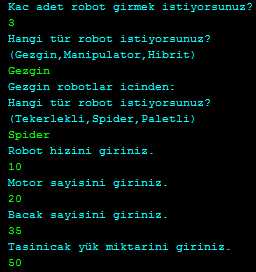
6. KABA KOD

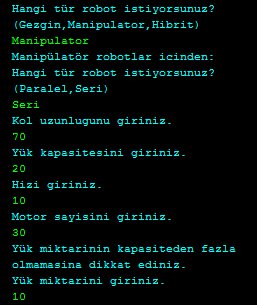
1. Başla
2. sekil sınıfını çağır
3. Kaç adet robot girmek istiyorsunuz?
4. Hangi tür robot girmek istiyorsunuz?
5. if(isim==gezgin)
6. GezginRobotlar içinden hangi tür?
7. Robotun hızı, motor sayısı, tekerlek/palet/bacak sayısı, taşınacak yük miktarı?
8. if(isim==manipülatör)
9. GezmeyenRobotlar içinden hangi tür?
10. Robotun hızı, motor sayısı, kol uzunluğu, yük kapasitesi, taşınacak yük miktarı?
11. İf(isim==hibrit)
12. Hareketli kısmı için hangi tür?
13. Robotun hızı, tekerlek/palet/bacak sayısı, robot geçiş süresi?
14. Manipülatör kısmı için hangi tür?
15. Robotun hızı, motor sayısı, kol uzunluğu, yük kapasitesi, taşınacak yük miktarı?
16. Hangi sıradaki robot kullanılacak?
17. if(robot==gezginRobotlar)
18. Kaç adet engel girilecek?
19. Engellerin koordinatları?
20. Robotu nereye koymak istiyorsunuz?
21. While(hareket==1)
22. Robot hareket etsin mi?
23. İf(hareketMiktarı>420 && hareketMiktarı<20)
24. Izgara dışına çıkıldı.
25. Engelle karşılaşıldıysa
26. Robot geçebiliyor mu?
27. Geçebiliyorsa engelSüresi()
28. Geçemiyorsa robot engelle karşılaştı.
29. Problem1() fonksiyonu çağır.
30. if(Robot==gezmeyenRobotlar)
31. Robotu nereye koymak istiyorsunuz?
32. While(hareket==1)
33. Robot hareket etsin mi?
34. İf(hareketMiktarı>420 && hareketMiktarı<20)
35. Izgara dışına çıkıldı.
36. if(hareketMiktarı>robotKol)
37. Robotun kolu oraya gidemez.
38. Problem2() fonksiyonunu çağır.
39. İf(Robot==hibritRobot)
40. Kaç adet engel girilecek?
41. Engellerin koordinatları?
42. While(hareket==1)
43. Robot nereden başlasın?
44. Hareketli kısım hareket etsin mi?
45. Robot sabitlendi.
46. Kol hareket etsin mi?
47. Problem3() çağır.
48. Son.

6.KAYNAKÇA

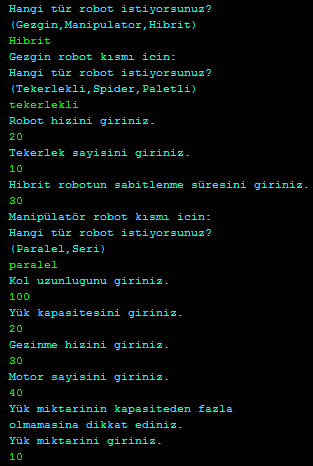
<http://www.yazilimmutfagi.com/index.php/2009/08/20/java-grafik-kutuphanesini-kullanarak-kare-ve-daire-cizmek/>

<https://creately.com/app/?tempID=h165rwt81&login_type=demo#>

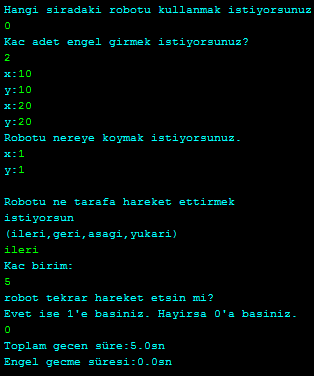
7.EKRAN ÇIKTILARI

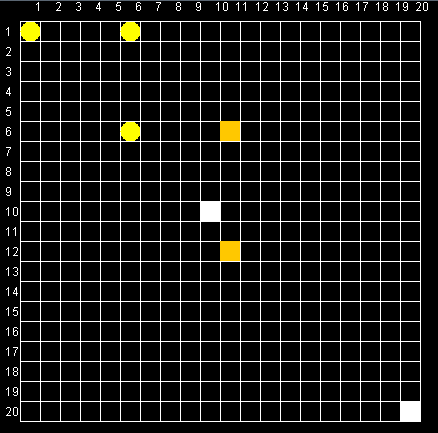
Şekil 1-Robot tanımlaması

Şekil 2-Robot tanımlaması



Şekil 3-Robot tanımlaması



 Şekil 4-Engel ve hareket istenmesi

Şekil 5-Izgara ekranı

8.UML DİYAGRAMI

