**ROBOT PROJESİ**

*Gözde ÖRGÜ - Büşra ERKAN*

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[gozde.orgu@gmail.com](mailto:gozde.orgu@gmail.com) [busraerkan39@gmail.com](mailto:busraerkan39@gmail.com)

**Özet**

*Robot Projesi verilen üç farklı problemden robot tipine göre seçilen problemi Gezgin, Manipülatör ve Hibrit olarak üç gruba ayrılan farklı türdeki robotların hareket etme, yük taşıma özelliklerini kullanarak çözer.*

**1.Giriş**

Robot Projesi; Robot, GezginRobot, ManipülatorRobot, Hibrit, Tekerlekli, Paletli, Spider, Seri, Paralel sınıflarından ve Proje sınıfından oluşur.

Her robotun ortak özelliği motor sayısı, yük miktarı ve robot tipidir. Robotlar 20x20 boyutlarındaki ızgara üzerinde konumlandırılırlar.

Gezgin robotlarda gezinme hızı vardır. En hızlısı tekerlekli, en yavaşı ise spider gezgin robottur. Tekerlekli robotta tekerlek sayısı, paletli robotta palet sayısı ve spider robotta da bacak sayısı kullanıcı tarafından belirlenir. Tekerlekli ve paletli gezgin robotlar kullanıcının isteğine göre konumları belirlenen engellerden geçebilirler ancak spider gezgin robot engellerden geçemez. Gezgin robotlar için engelden geçme süresi tekerlekli gezgin robotta motor sayısının 0.5 ile çarpılmasıyla, paletli gezgin robotta ise motor sayısının 3 ile çarpılmasıyla bulunur.

Manipülatör robotlarda kol uzunluğu, yük kapasitesi ve taşıma hızı vardır. Bu robotların konumları sabittir ve yükleri kolları yardımıyla taşırlar. Kol uzunluğunu aşan durumlarda yükleri taşıyamazlar. Seri robotlar çok yavaştır ve ağır yük kaldıramazlar, paralel robotlar ise çok hızlıdır ve ağır yük kaldırabilirler.

Hibrit robotlar hem gezgin robotların hem de manipülatör robotların özelliklerini taşır. Bu robotlar gezinebilme ve kol ile bir yükü taşıyabilme özelliklerine sahiptir. Hibrit robotlar engellerden de geçebilir.

**2.Temel Bilgiler**

Program Java dilinde geliştirilmiş olup tümleşik geliştirme ortamı olarak NetBeans IDE kullanılmıştır.

**3.Tasarım**

Robot Projesi’nin programlanma aşamaları aşağıda belirtilen başlıklar altında açıklanmıştır.

**3.1 Hiyerarşik Yapı**

Robot Projesi’nde bulunan oluşturulabilir, düzenlenebilir temel dokuz sınıf bulunmaktadır. Dokuz temel sınıfın birbirleri ile olan ortak özelliklerini belirleyen iki sınıf vardır. Tüm sınıfların ortak özelliklerini içeren temel ata “Robot” sınıfı bulunmaktadır.

Üzerlerinde işlem yapılabilen “Tekerlekli, Paletli, Spider, Seri ve Paralel” sınıfları, bu sınıfların ortak özelliklerini oluşturan “GezginRobot ve ManipülatorRobot”, GezginRobot’un özelliklerini taşıyan Hibrit ve tüm sınıfların atası olan “Robot” sınıfı olmak üzere hiyerarşik bir yapı kullanılmıştır.

**3.2 Algoritma**

Robot Projesi dokuz sınıftan oluşur. Temel ata sınıf Robot sınıfıdır. Robot sınıfında her robotta bulunması gereken motor sayısı, yük miktarı ve robot tipi özellikleri tanımlanır. Sınıflar oluşturulurken Encapsulation, Inheritance, Abstraction yapıları kullanılmıştır.

public void paint() içinde türlerine göre robotların ilk ve son konumları, varsa engel(ler) ızgara üzerinde gösterilir. Robotların hareket süreleri burada hesaplanır.

Program çalıştırıldığında kullanıcıdan tanımlanacak robot sayısını istenir. Kullanıcıdan robot sayısı kadar robot tipi girilmesi istenir. Bütün robotların minimum hızları, Seri ve Paralel robotların minimum yük kapasiteleri belirlenir. Eğer robot tipi “gezgin” ise gezgin robot türü seçilmesi istenir. Seçilen tür “tekerlekli” ise motor sayısı, yük miktarı, tekerlek sayısı, hızı istenir. Hız kontrolü yapılır. Tekerlekli robotun nesnesi oluşturulur. “paletli” ise motor sayısı, yük miktarı, palet sayısı, hızı istenir. Hız kontrolü yapılır. Paletli robotun nesnesi oluşturulur. “spider” ise motor sayısı, yük miktarı, bacak sayısı, hızı istenir. Hız kontrolü yapılır. Spider robotun nesnesi oluşturulur. Eğer robot tipi “manipülatör” ise manipülatör robot türü seçilmesi istenir. “seri” ise motor sayısı, yük miktarı, yük taşıma kapasitesi, taşıma hızı, kol uzunluğu istenir. Hız ve yük kapasitesi kontrolü yapılır. Seri robotun nesnesi oluşturulur. “paralel” ise motor sayısı, yük miktarı, yük taşıma kapasitesi, taşıma hızı, kol uzunluğu istenir. Hız ve yük kapasitesi kontrolü yapılır. Paralel robotun nesnesi oluşturulur. Eğer robot tipi “hibrit” ise robot türünün “tekerlekliSeri, paletliSeri, spiderSeri, tekerlekliParalel, paletliParalel, spiderParalel” olmasına göre yukarıdaki bilgiler kullanıcıdan alınır.

Kullanıcı hangi robotu hareket ettirmek istiyorsa o robotun sırasını girer. Robot “tekerlekli, paletli, spider” türlerinden biri ise kullanıcıya engel yerleştirmek isteyip istemediği sorulur. Eğer engel varsa engel sayısı, engel konumu, robotun başlangıç noktası ve yön bilgisi kullanıcıdan alınır. Gerekli bilgiler yapıcıya gönderilir. Robot “seri, paralel” türlerinden biri ise robotun başlangıç noktası ve yön bilgisi kullanıcıdan alınır. Gerekli bilgiler yapıcıya gönderilir. Robot “tekerlekliSeri, paletliSeri, spiderSeri, tekerlekliParalel, paletliParalel, spiderParalel” türlerinden biri ise kullanıcıya engel yerleştirmek isteyip istemediği sorulur. Eğer engel varsa engel sayısı, engel konumu, robotların başlangıç noktaları ve yön bilgileri kullanıcıdan alınır. Gerekli bilgiler yapıcıya gönderilir.

**3.3 Yöntemler**

*public void paint(Graphics g)*

robotTipi hangi robot türünü gösteriyorsa konum bilgisine göre robot ve (varsa) engel ızgaraya çizilir. Robotun hareketi gösterilir ve son konum çizilir. Geçen süre hesaplanır.

*public int getX()*

Robotun ilk konumunun x koordinatını döndürür.

*public void setX(int x)*

x değişkenine değer atanır.

*public int getY()*

Robotun ilk konumunun y koordinatını döndürür.

*public void setY(int y)*

y değişkenine değer atanır.

*public int getEngelSayisi()*

Kullanıcıdan alınan engel sayısını döndürür.

*public void setEngelSayisi(int engelSayisi)*

engelSayisi değişkenine değer atanır.

*public int[] getEngelKonumuX()*

Engel konumunun x koordinatını döndürür.

*public void set EngelKonumuX(int[] engelKonumuX)*

engelKonumuX değişkenine değer atanır.

*public int[] getEngelKonumuY()*

Engel konumunun y koordinatını döndürür.

*public void setEngelKonumuY(int[] engelKonumuY)*

engelKonumuY değişkenine değer atanır.

*public ArrayList <Integer> getList()*

ArrayList’teki engel konum bilgilerinden x koordinatını döndürür.

*public void setList(ArrayList <Integer> list)*

ArrayList’e atama yapar.

*public ArrayList <Integer> getList1()*

ArrayList’e eklenen engel konum bilgilerinden y koordinatını döndürür.

*public void setList1(ArrayList <Integer> list1)*

ArrayList’e atama yapar.

*public String getRobotTipi()*

Kullanıcıdan alınan robot tipini döndürür.

*public void setRobotTipi(String robotTipi)*

robotTipi değişkenine atama yapar.

**3.4 Kullanılan Yapıcılar**

*public Proje(int x,int y,String robotTipi,ArrayList <String> yonBilgisi,ManipulatorRobot r1)*

Manipülatör robot tipi için oluşturulan yapıcıdır. Izgara çizimi gerçekleştirilir. Alınan robot konum bilgisi, robot tipi, yon bilgisi yapıcı içindeki değişkenlere atanır.

*public Proje(int x,int y,int engelSayisi,int[] engelKonumuX,int[] engelKonumuY,String robotTipi,ArrayList <String> yonBilgisi,GezginRobot r2)*

Gezgin robot tipi için oluşturulan yapıcıdır. Izgara çizimi gerçekleştirilir. Alınan robot konum bilgisi, robot tipi, yon bilgisi yapıcı içindeki değişkenlere atanır. Engel konum bilgileri ArrayList’e eklenir.

*public Proje (int x,int y,int engelSayisi,int[] engelKonumuX,int[] engelKonumuY,String robotTipi,ArrayList <String> yonBilgisi1,ArrayList <String> yonBilgisi2,Hibrit r3)*

Hibrit robot tipi için oluşturulan yapıcıdır. Izgara çizimi gerçekleştirilir. Alınan robot konum bilgisi, robot tipi, yon bilgisi yapıcı içindeki değişkenlere atanır. Engel konum bilgileri ArrayList’e eklenir.

**3.5 Sınıfların Özellikleri**

*Robot(Abstract Class)*

Tüm sınıfların atasıdır. Ortak Get ve Set metotları burada tanımlanmıştır.

*GezginRobot(Abstract Class)*

Robot sınıfından extends edilmiştir. Robot’tan miras alınan fonksiyon ve değişkenler bu sınıfın yapıcısında kullanılır.

*Tekerlekli(Class)*

GezginRobot sınıfından extends edilir.Gezgin robotlardan miras alınan fonksiyon ve değişkenler bu sınıfın yapıcısında kullanılır.

*Paletli(Class)*

GezginRobot sınıfından extends edilir.Gezgin robotlardan miras alınan fonksiyon ve değişkenler bu sınıfın yapıcısında kullanılır.

*Spider(Class)*

GezginRobot sınıfından extends edilir.Gezgin robotlardan miras alınan fonksiyon ve değişkenler bu sınıfın yapıcısında kullanılır.

*ManipulatorRobot(Abstract Class)*

Robot sınıfından extends edilmiştir. Robot’tan miras alınan fonksiyon ve değişkenler bu sınıfın yapıcısında kullanılır.

*Seri(Class)*

ManipulatorRobot sınıfından extends edilir.Manipülatör robotlardan miras alınan fonksiyon ve değişkenler bu sınıfın yapıcısında kullanılır.

*Paralel(Class)*

ManipulatorRobot sınıfından extends edilir.Manipülatör robotlardan miras alınan fonksiyon ve değişkenler bu sınıfın yapıcısında kullanılır.

*Hibrit(Class)*

GezginRobot sınıfından extends edilmiştir. Gezgin robotlardan miras alınan fonksiyon ve değişkenler bu sınıfın yapıcısında kullanılır.

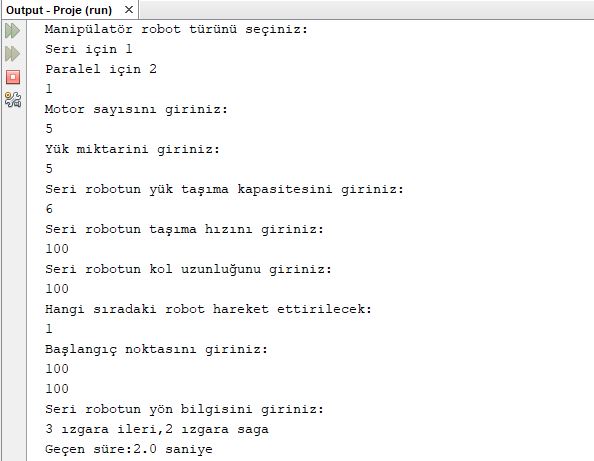
**3.6 Akış Diyagramı**

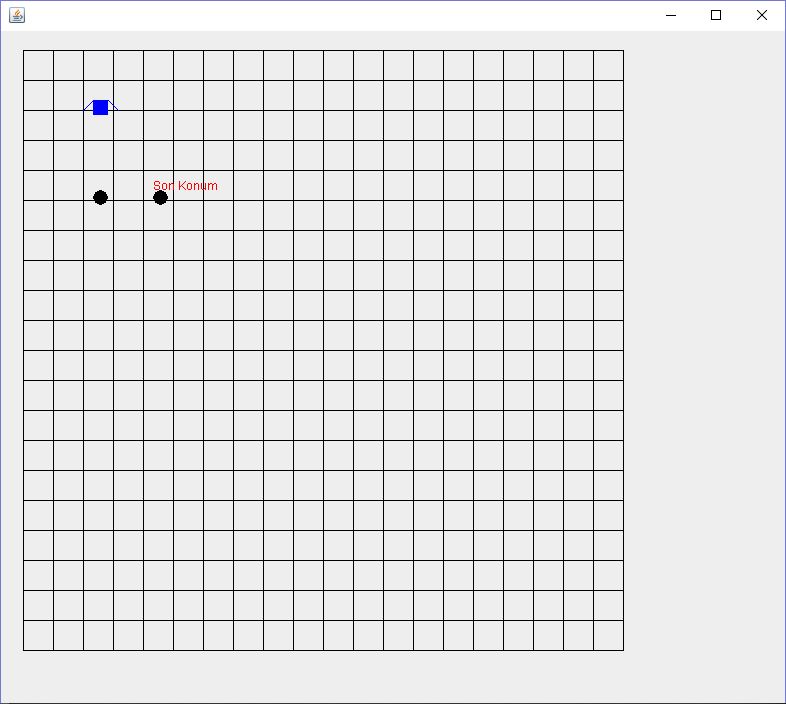


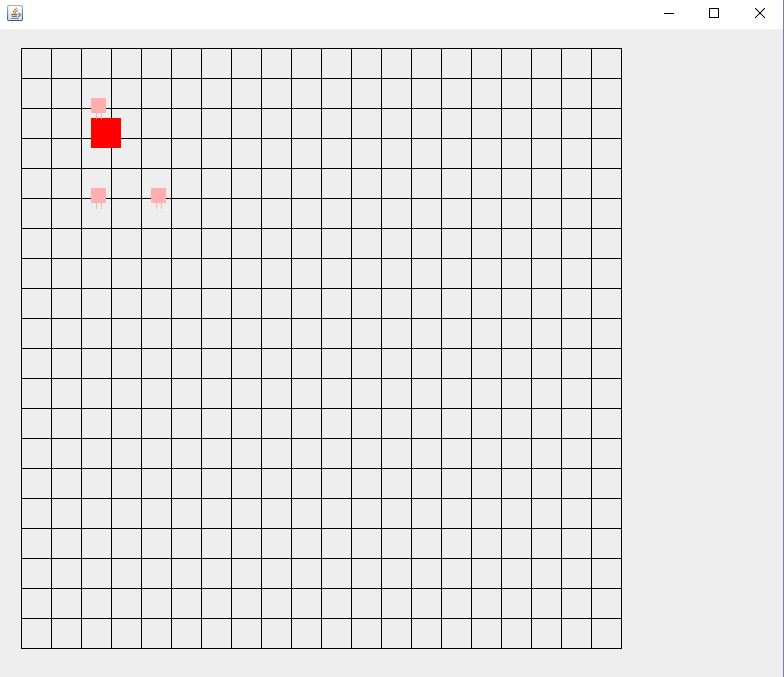
**4. UML Diyagramı**



**5. Deneysel Sonuçlar**

****

****

****

**6.Kaynakça**

<https://www.oguvenir.com/2013/06/java-gui-14grafikler.html>

<https://stackoverflow.com/questions/24104313/how-do-i-make-a-delay-in-java>