

Kocaeli Üniversitesi
Bilgisayar Mühendisliği Bölümü
Yazılım Labaratuvarı 1

Ahmet Suna

210202071

1.Özet

Belirli kurallara göre hareket eden bir robotun önündeki engelleri aşarak istenen hedefe ulaşmasını sağlayan bir oyun tasarlanması beklenmektedir. Oyunda iki adet problemin çözülmesi gerekmektedir. Problemlerin çözümü için nesneye yönelik programlama ve veri yapıları bilgilerinin kullanılması beklenmektedir.

Amaç: Proje gerçekleştirimi ile öğrencilerin nesneye yönelik programlama ve veri yapıları bilgisinin pekiştirilmesi ve problem çözme becerisinin gelişimi amaçlamaktadır.

Programlama Dili: Proje C++, C#, Java veya Python dili kullanılarak gerçekleştirilecektir. Projede aşağıda tanımlanan üç probleme çözüm bulmanız beklenmektedir.

2.Giriş

Proje de iki farklı problem ile karşılaşılmaktadır. Problem 1 deki amaç:

Bu problemde sizden robotu ızgara (grid) üzerinde verilen hedefe engellere takılmadan en kısa sürede ve en kısa yoldan ulaştırmanız beklenmektedir. Robotu tüm ızgarayı değil, yalnızca gerekli yolları gezerek hedefe ulaşmasını sağlamalısınız.

Adım 1: Gerekli boyutlarda karesel bir ızgara alanı oluşturmanız gerekmektedir.

Adım 2: Izgara üzerine engeller ve duvarlar yerleştirilmelidir. Izgara boyutu, engel sayısı ve engellerin konum bilgileri içeriği matris biçimindeki bir text dosyasından alınacaktır. Bu text dosyasına önceden verilecek bir url adresinden uygulama çalıştırıldığında otomatik olarak erişilerek dosyadaki tasarıma göre ızgara ve engel yapısı oluşturulacaktır.

Adım 3: Robotun başlangıç ve hedef noktaları ızgara üzerindeki uygun (engel veya duvar içermeyen) karelere rastgele belirlenmelidir. Robot başlangıçta tüm ızgara dünyasını bilmemelidir, sadece bir adım sonraki kareleri görebilmelidir.

Adım 4: Tüm bu bilgiler doğrultusunda, robotun hedefe en kısa sürede ulaşabileceği en kısa yol, adım adım ızgara üzerinde gösterilmelidir.

Problem 2 deki amaç:

Bu problemde sizden robotu labirentteki çıkış noktasına ulaştırmanız beklenmektedir.

Adım 1: Kullanıcı tarafından istenilen boyutlarda bir ızgara oluşturmanız gerekmektedir. **Adım 2:** Izgara üzerine 1 nolu tipte engeller yerleştirilerek labirent oluşturulmalıdır. Labirent içerisinde mutlaka çıkışa ulaşamayan yollar bulunmalıdır.

Adım 3: Labirentin giriş ve çıkış noktaları dörtgen ızgaranın herhangi çapraz 2 köşesi

olarak belirlenmelidir. Robot başlangıçta labirenti bilmemelidir.

Adım 4: Tüm bu bilgiler doğrultusunda, robotun çıkışa ulaşmak için izlediği yol adım adım ızgara üzerinde gösterilmelidir. Her adımda robotun daha önce geçtiği yollar üzerinde iz bırakması gerekmektedir.

3. Yöntem

Projenin detaylı analizi yapıldı.

Frontend python tkinter ile yapıldı.

Kullanıcıdan veri alınacak yerler oluşturuldu.

Program başında kullanıcıdan alınan satır ve sütun bilgisi ile problem 2 için harita oluşturma işlemleri başlatıldı.

Problem1 için gereken tek şey butonlara basılması ve program doğrudan sizi labirente yönlendirmektedir.

Block classı türetilmiş ve definitely, changeable, value, cantreturn=0,blocksize=1,blur=0,blockcount=0,visited=0 özellikleri tanımlanmış ve gerekli yerlerde kullanılması için hazır bekletilmiştir.

Main class oluşturulmuş ve 2 tane metod tanımlanmış:

Labirent1 methodu açılan ilk arayüzde labirent 1 butonuna tıklandığı zaman programı labirent sınıfına gönderir ve oradan ızgara sınıfına giderek labirenti oluşturur.

Labirent2 methodu kullanıcının girdiği bilgileri labirent sınıfına gönderir ve oradan ızgara sınıfına giderek labirenti oluşturur.

Labirent clası oluşturuldu bu clasın amacı ızgara clası ile robot clası arasındaki bağlantıyı kurmak için oluşturulmuştur. 2 tane methodu bulunmaktadır:

Problem1maze methodu problem1 için ızgara fonksiyonunu çalıştırıp labirentin oluşturulmasını sağlar aynı zamanda butonlar ile robot fonksiyonundan gerekli methodların harekete geçirilmesini sağlar. Bu methodun 2 tane fonksiyonu bulunur:

Mazechange1 ve mazechange2 fonksiyonları problem 1 için harita değiştirme fonksiyonlarıdır.

Problem2maze methodu problem2 için ızgara fonksiyonunu çalıştırıp labirentin oluşturulmasını sağlar aynı zamanda butonlar ile robot fonksiyonundan gerekli methodların harekete geçirilmesini sağlar. Bu methodun 2 tane fonksiyonu bulunur:

Startbutton fonksiyonu başla butonuna basılırsa direkt robotun çıkışı bulmasını sağlayan robot clasındaki fonksiyonu çağırır.

Onebyone fonksiyonu adımla butonuna basılırsa robotun her basımda 1 adım ilerlemesini sağlayan robot clasındaki ilgili fonksiyonu çağırır.

Izgara classı oluşturuldu.

Bu classda 6 tane method bulunmaktadır:

Squareblock methodu problem 2 için kenar çerçevesini oluşturur.

Allblock methodu problem 2 için matrixin içerisini block nesneleri ile doldurmaktadır.

Createstartfinish methodu labirent 2 için başlangıç ve bithiş noktalarını ayarlar.

Createroad methodu labirent2 için yolların oluşturulduğu methottur burda yolların oluşturulmasında random fonksiyonu kullanılmıştır.

Allblockformaze1 methodu problem 1 için matrixin içerisini block nesneleri ile gerekli özellikler tanımlanarak eklenmesinisayar.

Createroadformaze1 methodu url den alınan matrixin değerlerine göre yolun oluşturulmasını sağlar.

Robot classı türetilmiş ve matrix, labellist, changeablelist=0 özellikleri tanımlanmış bu class robotun haritayı çözmesini içeren methodlar bulunmaktadır. Bu classta 11 method ve 1 fonksiyon bulunmaktadır:

Robotstart methodu problem 2 için robotun başlangıç noktasına yerleştirilmesi sağlar ve robotroadfound methodunu çağırır.

Robotroadfound methodu labirenti sağ ve aşağı öncelikli olmak üzere sırasıyla dolaşır çıkışı bulunca programı beklemeye alır.

Robotstartstep methodu problem 2 için başlangıç noktasını oluşturur. Problem2 de adım adım ilerlemeyi görmek için oluşturulmuş olan robotstepsroadfound methoduna yönlendirir.

Robotstepsroadfound methodu labirenti sağ ve aşağı öncelikli olmak üzere sırasıyla dolaşır bu fonksiyonu bitirmek için direkt başla butonuna basılarak çıkışa hızlıca gidilebilir. Çıkışı bulunca programı beklemeye alır.

Startformaze1 methodu problem1 için robotun başlangıç ve bitiş noktalarının bulunmasını sağlar.

Createstartfinishformaze1 methodu başlangıç ve bitiş noktalarını bulur.

Robotroadfoundformaze1 methodu robotun başlangıç karesinden başlayarak haritayı random bir şekilde dolaşarak bitiş noktasına ulaşması hedeflenmiştir.

Robotroadfoundformaze1closer methodu en yakın noktayı gösteren methoddur.

Closerdotfinder methodu problem 2 için yanlış girilen yollardan sonra en yakın gidebileceği farklı yöne sahip kareye ilerler.

Returnnewdot methodu problem 1 için çıkılmaz olan noktalarda random daha önce gittiği bir kareye ilerlemesini sağlar.

Robotstes fonksiyonu labirent 2 için adım adım ilerleme fonksiyonudur her seferinde butona basılarak ilerleme sağlanır.

Uygulama classı oluşturuldu ve start newwindow parametrelerini almaktadır bu classın 3 tane methodu bulunmaktadır:

Surestart methodu gerekli yerlerde programın süre başlangıcı oluşturulur.

Surefinish methodu program bitiminde bitiş süresini yeni bir pencerede yazdırır.

Matrixwritter methodu toplam kaç kareden geçildiği bilgisini yeni pencerede yazdırır.

Uml diagramı :



4. Deneyisel Sonuçlar

Kullanıcıdan gerekli bilgiler alındı.

Oyun haritaları başarı ile oluşturuldu.

Haritalarda adım adım ve hızlı çözme işlemleri yapıldı.

Oyun çıkış bulunana kadar devam ediyor.

Kaç blockdan geçildiği hesaplandı.

Tüm bilgiler ekranda gösterildi.

5. Katkılar

Python ile nesne sınıfları öğrenildi.

pythona nesneye yönelik çalışma prensipleri geliştirildi.

Sınıf kavramları ve onların kullanımı geliştirildi.

Constructor kullanımı araştırılmış ve öğrenilmiştir.

Polymorphisim kavramı araştırılmış ve projede kullanılmıştır bu sayede ağırtaş özelkağıt ustamakas sınıfları türetilmiş.

Global terim tanımları öğrenilmiş ve kullanılmış.

Tkinter öğrenilmiş ve kullanılmıştır.

Bütün proje python ile baştan aşağı kodlanmış olup herhangi bir kaynakdan direkt olarak etkilenilmemiştir.

6.Yalancı Kod

If labirent1 butona tıklanırsa
problem1maze git

Urllerdeki bilgileri al

Onları bir matriste tut

Bu matrisi allblockformaze1 gönder

Matrisin içini block nesnesi ile doldur.

If button1 basılırsa robot çıkışı bulması için robotrouadfound git ve random şekilde ziyaret edilmemiş kareleri ziyaret et

If çıkışı bulursan prograbmı durdur ve ekranda kaç kareden geçildiğini ve süreyi yazdır

Else devam et

If buton2 basılırsa adım adım robot çıkışı bulması için robotstepsrouadfound git ve random şekilde ziyaret edilmemiş kareleri ziyaret et

If çıkışı bulursan prograbmı durdur ve ekranda kaç kareden geçildiğini ve süreyi yazdır

Else devam et

If buton3 basılırsa bütün bulutlamayı kaldır

If buton4 basılırsa başlangıç bitiş arası en kısa yol bulunur.

If buton basılırsa harita diğer url e geçer ve yeniden ekran açılır.

If labirent2 butona tıklanırsa
problem2maze git

Kullanıcıdan alınan satır ve sütun bilgileri ile matris oluştur

Bu matrisi allblock gönder

Matrisin içini block nesnesi ile doldur.

If button1 tıklanırsa robot direkt labirentin çıkışını bulsun ve geçen süreyi ve toplam geçilen kare sayısını ekrana yazdır.

If buton2 tıklanırsa robot her tıklamada 1 kare ilerlesin

If buton2 tıkladıktan sonra buton1 tıklanırsa direkt çıkışa git.

7.Kaynakça

<https://github.com/OrWestSide/python-scripts/blob/master/maze.py>

<https://www.geeksforgeeks.org/python-matrix/>

https://www.w3schools.com/python/python_variables_global.asp

<https://stackoverflow.com/questions/35848046/tkinter-label-object-is-not-callable>