

KOCAELİ ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
BLM210 PROGRAMLAMA LABORATUVARI II
PROJE 1

OTONOM HAZİNE AVCISI

Proje İlan Tarihi: 20/02/2024

Proje Teslim Tarihi: 18/03/2024

Otonom Hazine Avcısı projesinde, otonom hareket eden bir karakterin, içerisinde çeşitli hazineler ve engeller bulunan bir harita üzerindeki hazineleri topladığı bir oyun tasarlanması beklenmektedir. Oyunda amaç, karakterin tüm hazineleri en kısa sürede toplamasını sağlayacak algoritmanın tasarlanmasıdır. Bunun için nesneye yönelik programlama ve veri yapıları bilgilerinin kullanılması beklenmektedir.

Amaç: Proje gerçekleştirimi ile öğrencilerin nesneye yönelik programlama ve veri yapıları bilgisinin pekiştirilmesi, uygulanması ve problem çözme becerisinin geliştirilmesi amaçlanmaktadır.

Programlama Dili: Projenin geliştirilmesinde C++, C#, Java dilleri kullanılabilir.

PROJE İSTERLERİ:

Bu projede sizden oyun karakterinin, rastgele oluşturulmuş bir harita üzerinde rastgele belirlenmiş bir başlangıç noktasından başlayarak, oluşturulmuş engellere takılmadan, en kısa sürede ve en kısa yoldan giderek ızgaradaki tüm hazine sandıklarını toplamasını sağlamanız beklenmektedir. Karakter oyun boyunca tüm ızgarayı değil, yalnızca gerekli yolları gezmelidir.

Oyunun Genel Adımları:

Haritanın Oluşturulması: Program ilk çalıştırıldığında kullanıcıdan haritanın boyut bilgisi istenmelidir. Çalışmalarınızı küçük boyutlu haritalarda yapabilirsiniz fakat sunum sırasında 1000x1000 boyutlu haritalar oluşturmanız istenecektir. Kullanıcıdan alınan harita boyut bilgisine göre harita oluşturulmalıdır. Harita, her uygulama başladığında yeniden üretilmelidir. Haritanın oluşturulması için gereken algoritmanın sizler tarafından geliştirilmesi beklenmektedir. Geliştirdiğiniz algoritma, her adımda rastgele bir harita üretildiğini doğrulanmalıdır. Üretilmesi ve doğrulanması için etkili algoritmaların geliştirilmesi istenmektedir. Mümkün olduğu kadar efektif çözümler geliştirilmeli, kaba kuvvet (brute force) uygulamalarından kaçınılmalıdır.

Harita oluşturulduğunda haritanın sol tarafı kış, sağ tarafı yaz temasında olmalıdır. Bir sonraki başlıkta anlatılacak sabit nesnelerin görselleri bu temalara göre değişmelidir. Temaya göre sabit nesnelerin üretilmesinde hiyerarşik nesne yapısı kullanılmalıdır.

Tüm hazine sandıklarının ulaşılabilir (engellerin ortasında kalmadığından) olduğundan emin olmanız gerekir.

Harita İçeriğinde Bulunması Gerekenler: Harita üzerinde yol (üzerinde ilerlenebilen kareler), engeller ve hazine sandıklarını içeren kareler bulunmalıdır.

Engeller, sabit ve hareketli olmak üzere iki çeşit olmalıdır. Sabit engeller; ağaç, dağ, kaya ve duvar gibi oyun boyunca hareket etmeyen nesnelerdir. Yaz ve Kış için ayrı sabit engeller tanımlanmalıdır. Hareketli engeller ise kuş ve arı gibi nesneler olup kuşlar 5 karelik, arılar ise 3 karelik alanda hareket edebilir. Bu hareket kuşlar için yukarı-aşağı, arılar için sağ-sol şeklinde olmalıdır. Hareketli nesnelerin her biri 2x2 birimlik alan kaplamalıdır.

Her harita oluştuğunda en az 20 adet sabit (her sabit nesneden en az 2 adet oluşturulması zorunludur), 3 adet hareketli nesnenin üretilmiş ve harita üzerinde uygun yerlere yerleştirilmiş olması beklenmektedir. Harita üzerinde hareketli engellerin takip ettiği yol kırmızı renkle gösterilmelidir.

Hazine sandıkları; altın sandık, gümüş sandık, zümrüt sandık ve bakır sandık olmak üzere dört çeşit olmalıdır. Hazine sandıkları toplanabilen nesnelerdir. Karakter bir sandığı topladığında ilgili sandık harita üzerinde kaybolmalıdır. Sandıkların toplanma sırasına göre (kendi aralarında öncelik sıralaması altın sandık, gümüş sandık, zümrüt sandık ve bakır sandık sıralanmalıdır) ekranın sağ üst köşesinde sandığın türüne göre bilgi verilmelidir (Örneğin; toplanan sandık altın sandık ise “Altın sandık toplandı! (8,16-8,18) konumunda bulundu” yazılmalıdır).

Eğer karakter aşağıdaki sıralama ile sandıkları buluyorsa bunların öncelik sıralamasına göre değiştirilmesi gerekir. Sıra ile tespit edilenler:

Altın sandık toplandı! (8,16-8,18) konumunda bulundu. Gümüş sandık toplandı! (8,16-8,18) konumunda bulundu. Altın sandık toplandı! (8,16-8,18) konumunda bulundu.

Öncelik sırasına göre :

Altın sandık toplandı! (8,16-8,18) konumunda bulundu. Altın sandık toplandı! (8,16-8,18) konumunda bulundu. Gümüş sandık toplandı! (8,16-8,18) konumunda bulundu.

Her harita oluşturulduğunda her sandık türünden en az 5 adet üretilmiş ve harita üzerinde uygun yerlere yerleştirilmiş olması beklenmektedir.

Haritada Başlangıç ve Bitiş Noktalarının Belirlenmesi: Karakterin başlangıç noktası harita üzerindeki uygun (engel ya da sandık içermeyen) karelerde rastgele olacak şekilde belirlenmelidir. Bitiş noktası, geliştirilen algoritmaya göre toplanan en son hazine sandığının konumu olmalıdır.

Karakterin Hareketi: Karakter 3x3 bir alanı (3 birim sağ, 3 birim sol, 3 birim yukarı ve 3 birim aşağıyı) görebilir. Oyun boyunca karakterin hareketi otomatik olmalıdır. Program başlatıldığında karakter, tüm sandıkları toplamak için hesaplanan en kısa yol üzerinde ilerlemelidir. Karakter başlangıçta tüm haritayı bilmemelidir. Program sonlandığında karakterin öğrenmediği kareler sisli (kapalı) olarak gösterilmeli, öğrenilen kareler ise açılarak ilgili karelerde bulunan nesneye göre (engel, sandık, vs.) belirtilmelidir. Öğrenilen karelerin içeriği, öğrenilme sırasına göre yukarıdan aşağıya doğru olacak şekilde ekranın sağ üst köşesinde yazdırılmalıdır (Örneğin; “Ağaç keşfedildi!”, “Gümüş sandık keşfedildi!” gibi).

En Kısa Yol Algoritması ve Beklenen Çıktılar: Tüm bu bilgiler doğrultusunda, karakterin rastgele belirlenen başlangıç noktasından başlayarak tüm hazine sandıklarını en kısa sürede toplayabileceği en kısa yol hesaplanmalıdır. En kısa yolun hesaplanmasında geliştirilecek algoritmayı sizin tasarlamamız beklenmektedir. Tasarladığınız algoritma sonucunda elde edilen en kısa yol, ızgara üzerinde yeşil renkle gösterilmelidir. En kısa yola göre gidilmesi gereken adım sayısı (kaç kare üzerinden geçildiği) bilgisi ekranda yazdırılmalıdır.



Şekil 1. Örnek ızgara ve engel görünümü

Sınıf Tanımları:

Verilen projede aşağıdaki tanımlara uygun olacak şekilde sınıfların oluşturulması ve kodlanması beklenmektedir.

Karakter:

Bu sınıfta bulunması gereken özellikler ve fonksiyonlar:

- ID, Ad bilgileri tutulmalıdır.
- Karakterlerin ilerlediği koordinatları tutacak Lokasyon değişkenleri olmalıdır.
- Constructor, Get, Set ve En KısaYol metotları yer almalıdır.

NOT: Karakter çapraz gidemez. Sadece sağ, sol, yukarı ya da aşağı yönde hareket sağlayabilir.

Lokasyon sınıfı

- x ve y koordinatlarını tutan iki farklı değişken tutulmalı. Constructor, Get ve Set metotları yer almalıdır.

Engel Sınıfı:

- Engel sınıfında sabit ve hareketli olmak üzere 2 farklı tipteki engellerin oluşturulması gerekmektedir.

Hareketsiz Engeller:

Ağaçlar:

- Çeşitli türleri ve boyutları olabilir.
- Yoğun ormanlık alanlar veya tek başına ağaçlar olarak bulunabilirler.
- Oyuncu üzerinden geçemez.
- 2x2, 3x3, 4x4 veya 5x5'lik boyutta olabilirler.

Kayalar:

- Farklı boyutlarda ve şekillerde olabilirler (Örneğin; küçük çakıl taşları, büyük kayalar).
- Oyuncu üzerinden geçemez.
- 2x2 veya 3x3'lük boyutta olabilirler.

Duvarlar:

- Bir duvar başka bir duvarla yan yana olmamalıdır.
- Oyuncu üzerinden geçemez.

- 10x1'lik boyutta olabilirler.

Dağlar:

- Birden fazlası bir araya gelerek dağ kümeleri oluşturabilir.
- Oyuncu üzerinden geçemez.
- 15x15'lik boyutta olabilirler.

Dinamik Engeller:

Kuşlar:

- Yalnızca yukarı-aşağı yönde hareket edebilir.
- Oluşturulduğu kareden itibaren 5 kare yukarı ve aşağı gidip gelebilir.
- Oyuncu üzerinden geçemez.
- 2x2 birim boyutunda olmalıdır.

Arılar:

- Yalnızca sağ-sol yönde hareket edebilir.
- Oluşturulduğu kareden itibaren 3 kare sağa ve sola gidip gelebilir.
- Oyuncu üzerinden geçemez.
- 2x2 birim boyutunda olmalıdır.

Uygulama Sınıfı:

- Uygulama içerisinde karakterin hedefe kaç adımda ulaştığı, hangi nesneleri elde ettiği gibi bilgilerin tutulması ve ekranda gösterilmesi sağlamalıdır.

NOT: Yukarıdaki sınıflar, işlevsel bakımdan genel olarak tanımlanmış olup her sınıf için kullanılacak özellik (property) ve metotların tanımlaması sizden beklenmektedir. Ayrıca, gerektiği durumlarda neden kullanıldığının açıklanması koşuluyla yukarıdaki sınıflardan farklı sınıf tanımlamaları yapılabilir.

Projede ağaç, kuyruk gibi veri yapılarının ve Encapsulation, Inheritance, Polymorphism, Abstraction yapılarından gerekli olanların kullanılması gerekmektedir. Projede her yapıyı kullanmamış olsanız bile, proje sunumu esnasında bu yapıların ne olduğunu bilip bilmediğinizi ölçecek sorular sorulacaktır.

Arayüz ve Görsellik:

- Arayüzdeki tüm görseller net ve açık olmalıdır. Oluşturulan yol, hareketli nesnelerin hareket güzergahı net bir şekilde görülmelidir.
- Arayüzde 2 buton yer almalıdır: “Yeni Harita Oluştur” ve “Başlat” Butonu.

- Proje “Yeni Harita Oluştur” butonu ile çalıştırıldığında rastgele üretilmiş harita tasarımı ekrana getirilmelidir. Bu butona basıldığında haritada sis gösterilmemeli, başlangıç noktası belirlenmiş olmalıdır.
- “Başlat” butonuna tıklandığında tüm harita sisle kapatılarak harita içeriğinin görünümü engellenmelidir. Oyun sonuca ilerletilerek karakterin sırasıyla gezdiği kareler ve bulduğu en kısa yol ekranda yeşil renkle çizdirilmelidir. Geliştirdiğiniz algoritmanın belirlediği en kısa yola göre gidilmesi gereken adım sayısı ve sırasıyla hangi nesnelerin keşfedildiği bilgileri ekranda yazdırılmalıdır.

NOT: Verilen projede arayüz tasarımından puan alacaksınız. Bu nedenle arayüze özen göstermeniz beklenmektedir. Arayüz tasarımı için hazır tasarımlar kullanılmamalıdır.

ÖDEV TESLİMİ

- Proje raporu IEEE formatında (önceki yıllarda verilen formatta) 4 sayfa uzunluğunda olmalıdır. Rapor; akış diyagramı veya yalancı kod içermeli, özet, giriş, yöntem, deneysel sonuçlar, sonuç ve kaynakça bölümünden oluşmalıdır. **Raporda UML sınıf diyagramının çizilmiş olması beklenmektedir.**
- Dersin takibi projenin teslimi dâhil edestek.kocaeli.edu.tr sistemi üzerinden yapılacaktır. edestek.kocaeli.edu.tr sitesinde belirtilen tarihten sonra teslim edilen projeler kabul edilmeyecektir.
- Proje ile ilgili sorular edestek.kocaeli.edu.tr sitesindeki forum üzerinden Arş. Gör. Kadir Kesimal veya Arş. Gör. Gamze Korkmaz Erdem’e sorulabilir. **Proje teslimine 2 gün kala sorulan hiçbir soruya cevap verilmeyecektir.**
- Sunum tarihleri daha sonra duyurulacaktır.
- Sunum sırasında;
 - Geliştirdiğiniz algoritmanın çeşitli kısımlarının ne amaçla yazıldığı ve geliştirme ortamı hakkında sorular sorulabilir.
 - Kullandığınız herhangi bir satır kodu açıklamamız istenebilir.
 - Nesneye yönelik programlama ve veri yapılarına dair kullandığınız yapıları açıklamamız istenebilir.

Projenin tanıtım toplantısı 23 Şubat Cuma günü saat 15:30’da bölüm duyurularında ve e-destekte duyurulacak sınıfta yüz yüze yapılacaktır.

Proje grupları en fazla 2 kişiden oluşmalıdır. Proje grup bilgileri e-destekte paylaşılabilecek link üzerinden en geç 1 Mart Cuma günü saat 17:00 ‘a kadar girilmelidir. Bu tarihten sonra gruplarda herhangi bir değişiklik yapılmayacaktır.