

Programlama Laboratuvarı

Oğulcan Alınlı
Bilgisayar Mühendisliği 1.Öğretim
Kocaeli Üniversitesi
ogulcanalinli60@gmail.com

**Bu doküman kocaeli üniversitesi
bilgisayar mühendisliği bölümü
programlama laboratuvarı dersi ikinci
dönem 1.proje ödevi için hazırlanmıştır**

I. ÖZET

I. Giriş

Otonom Hazine Avcısı projesi, öğrencilerin nesneye yönelik programlama ve veri yapıları bilgilerini pratiğe dökerek pekiştirmelerini ve problem çözme becerilerini geliştirmelerini amaçlamaktadır. Bu proje kapsamında, bir oyun karakterinin rastgele oluşturulan bir harita üzerinde çeşitli engelleri aşarak hazine sandıklarını toplaması hedeflenmektedir.

Projenin temel amacı, öğrencilerin yazılım geliştirme sürecinde karşılaşılabilecekleri problemleri çözmek için etkili algoritmalar ve veri yapıları kullanarak programlama becerilerini artırmaktır. Aynı zamanda, projenin uygulanması sürecinde öğrencilerin analitik düşünme yeteneklerini ve yaratıcı çözüm odaklarını geliştirmeleri hedeflenmektedir.

Bu şekilde projenin genel amacı ve kapsamı daha ayrıntılı bir şekilde açıklanmış olacaktır.

II. GİRİŞ

Otonom Hazine Avcısı projesi, bir oyun karakterinin rastgele oluşturulan bir harita üzerinde çeşitli engelleri aşarak hazine sandıklarını toplamasını sağlayacak bir algoritmanın tasarlanmasını hedeflemektedir. Proje, belirli adımlar ve gereksinimler doğrultusunda uygulanacak ve öğrencilerin programlama becerilerini geliştirmek için önemli bir fırsat sunacaktır.

Projede Kullanılacak Programlama Dilleri
Projenin geliştirilmesinde C++, C# veya Java gibi yaygın olarak kullanılan programlama dillerinden biri tercih edilebilir. Bu dillerin seçilme nedeni, geniş kütüphane desteği, nesne tabanlı programlama yetenekleri ve performans avantajlarıdır. Her bir dilin kendine özgü avantajları ve dezavantajları bulunmaktadır, bu nedenle öğrencilerin tercih ettikleri dili kullanmaları teşvik edilmektedir.

Proje İstekleri

Haritanın Oluşturulması: Proje başladığında kullanıcıdan harita boyutu bilgisi alınacak ve bu bilgiye göre rastgele bir harita oluşturulacaktır. Harita oluşturulurken etkili algoritmalar kullanılmalı ve harita her başlatıldığında yeniden üretilmelidir.

Harita İçeriği: Harita üzerinde yol, engeller ve hazine sandıkları bulunacaktır. Engeller, sabit (ağaç, dağ, duvar, vb.) ve

hareketli (kuş, arı, vb.) olmak üzere iki farklı türde olacaktır. Hazine sandıkları, farklı türlerde (altın, gümüş, zümrüt, bakır) olacak ve toplandıklarında haritadan kaybolacaktır.

Karakterin Hareketi: Oyun karakteri, başlangıç noktasından başlayarak en kısa yol üzerinde ilerleyerek hazine sandıklarını toplayacaktır. Karakterin hareketi otomatik olacak ve en kısa yol algoritması kullanılarak hesaplanacaktır.

Sınıf Tanımları: Proje kapsamında, karakter, engel, lokasyon gibi sınıfların doğru bir şekilde tanımlanması ve kodlanması gerekmektedir. Her bir sınıfın özellikleri ve metotları belirlenmeli ve işlevsel bir yapı oluşturulmalıdır.

Arayüz ve Görsellik: Projenin arayüzü, net ve açık olmalıdır. Harita, hareketli nesnelerin güzergahı ve en kısa yolun görsel olarak net bir şekilde görüntülenmesi sağlanmalıdır. Ayrıca, arayüzde kullanılacak butonlar ve bilgilendirme alanları da dikkatle tasarlanmalıdır.

Veri Yapıları ve Algoritmalar
Projede kullanılacak veri yapıları arasında ağaçlar, kuyruklar gibi yapılar bulunabilir. Ayrıca, en kısa yol algoritması için genellikle graf tabanlı algoritmalar tercih edilir. Öğrencilerin, veri yapıları ve algoritmalar konusunda bilgi sahibi olmaları ve etkili çözümler üretmeleri beklenmektedir.

Bu bölümde belirtilen gereksinimler doğrultusunda proje detaylı bir şekilde planlanacak ve uygulanacaktır.

III. YÖNTEM

Haritanın Oluşturulması

Proje başladığında kullanıcıdan harita boyutu bilgisi alınacak ve bu bilgiye göre rastgele bir harita oluşturulacaktır. Harita oluşturulurken kullanılan algoritma, her seferinde farklı bir harita üretecek şekilde tasarlanmalıdır. Örneğin, bir dizi rastgele engel ve hazine sandığı oluşturulabilir, ancak her seferinde farklı konumlarda ve dağılımlarda olacak şekilde bu öğeler haritaya yerleştirilmelidir. Harita oluşturulduktan sonra, sol tarafı kıs temasıyla, sağ tarafı ise yaz temasıyla uyumlu olarak görselleştirilmelidir.

Harita İçeriği

Harita içeriğinde, yol, engeller ve hazine sandıkları bulunmalıdır. Engeller, sabit ve hareketli olmak üzere iki farklı türde olacaktır. Sabit engeller, ağaçlar, dağlar, duvarlar ve kayalar gibi oyun boyunca hareket etmeyen nesnelerdir. Her bir sabit engelin harita üzerinde doğru bir şekilde konumlandırılması gerekmektedir. Hareketli engeller ise kuşlar ve arılar gibi nesnelerdir. Bu engellerin hareketi belirli kurallara göre olacak ve uygun görsel efektlerle animasyonları sağlanacaktır.

Hazine sandıkları, farklı türlerde (altın, gümüş, zümrüt, bakır) olacak ve her türden en az 5 adet olmak üzere haritaya yerleştirilecektir. Karakter bir sandığı topladığında, ilgili sandık harita üzerinden kaybolacak ve ekranda toplanan sandığın türüne göre bilgilendirme yapılacaktır.

Karakterin Hareketi ve Yol Belirlenmesi
Oyun karakteri, belirlenen başlangıç noktasından başlayarak en kısa yol üzerinde ilerleyerek hazine sandıklarını toplayacaktır. Karakterin hareketi otomatik

olarak gerçekleşecektir, yani kullanıcı müdahalesi olmadan karakter belirlenen algoritma doğrultusunda hareket edecektir. En kısa yol algoritması, haritadaki engeller ve hazine sandıkları göz önünde bulundurularak hesaplanmalı ve karakterin bu yolu takip etmesi sağlanmalıdır.

Sınıf Tanımları ve Veri Yapıları
Proje kapsamında, karakter, engel, lokasyon gibi sınıfların doğru bir şekilde tanımlanması ve kodlanması gerekmektedir. Her bir sınıfın özellikleri ve metotları belirlenmeli ve işlevsel bir yapı oluşturulmalıdır. Veri yapıları olarak ağaçlar, kuyruklar, graf yapısı gibi yapılar kullanılabilir. Bu veri yapıları, algoritma ve mantık işlemlerinde kullanılarak projenin etkin bir şekilde yönetilmesini sağlayacaktır.

Arayüz ve Görsellik
Projenin arayüzü, kullanıcı dostu ve görsel açıdan çekici olmalıdır. Harita, engeller, hazine sandıkları ve karakterin hareketi net bir şekilde görüntülenmeli ve anlaşılabilir olmalıdır. Arayüzde yer alan butonlar ve bilgilendirme alanları, kullanıcıyı doğru şekilde yönlendirmeli ve oyunun ilerleyişi hakkında bilgi vermelidir. Ayrıca, hareketli nesnelerin animasyonları da görsel efektlerle desteklenmelidir.

IV. SONUC

Bu bölümde, projenin kodlama aşaması ve karakterin en kısa yol algoritmasının tasarlanması detaylı bir şekilde ele alınacaktır. Kodlama süreci, belirtilen sınıf yapıları ve veri yapıları kullanılarak gerçekleştirilecek ve algoritma, karakterin

hazine sandıklarını toplamak için izleyeceği en kısa yolu hesaplayacaktır.

Kodlama Süreci

Projenin kodlama sürecinde, öncelikle belirtilen sınıf tanımları doğru bir şekilde oluşturulmalıdır. Karakter sınıfı, engel sınıfları (sabit ve hareketli engeller için ayrı ayrı), lokasyon sınıfı gibi yapılar kodlanmalı ve her bir sınıfın özellikleri ve metotları detaylı olarak tasarlanmalıdır. Veri yapıları olarak ise ağaçlar, kuyruklar, graf yapısı gibi yapılar kullanılabilir.

Kodlama sürecinde, her bir sınıfın işlevselliği sağlanmalı ve proje gereksinimlerine uygun olarak özellikler, metotlar ve yapılar oluşturulmalıdır. Örneğin, karakterin hareketini kontrol eden metotlar, engellerin farklı türlerine göre davranışları, harita oluşturma algoritması, en kısa yol hesaplama algoritması gibi önemli fonksiyonlar sağlanmalıdır.

En Kısa Yol Algoritması

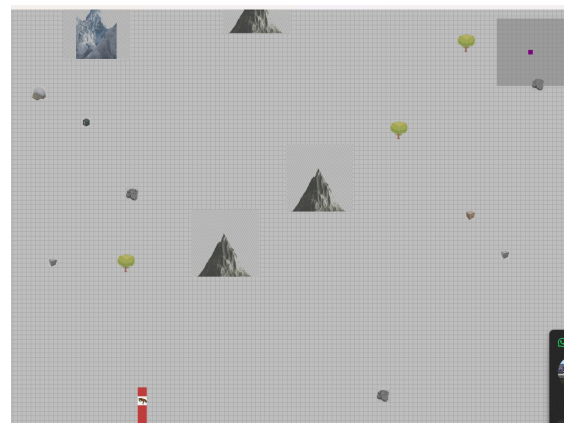
Projenin en önemli kısmı olan en kısa yol algoritması, karakterin hazine sandıklarını toplamak için izleyeceği optimum yolu belirlemelidir. Bu algoritma, harita üzerindeki engelleri, hazine sandıklarını ve karakterin mevcut konumunu göz önünde bulundurarak en kısa yolu hesaplamalıdır.

En kısa yol algoritması genellikle graf teorisi prensipleri kullanılarak tasarlanır. Örneğin, A* (A star) algoritması, Dijkstra algoritması gibi algoritmalar kullanılabilir. Bu algoritmalar, karakterin hazine sandıklarını toplamak için haritada hangi yolları izlemesi gerektiğini belirleyecek ve optimum çözümü sağlayacaktır.

V. DENEYSEL SONUÇLAR



<https://www.youtube.com/playlist?list=P L22TK-6CtDafPHI91N2Swcedg56L9Sk nO>



Agackış

Sınıf

→ HareketsizEngel

Alanlar

boyut

Metotlar

Agackış

Draw

Generate...

AgacYaz

Sınıf

→ HareketsizEngel

Alanlar

boyut

Metotlar

AgacYaz

Draw

Generate...

Arilar

Sınıf

→ HareketliEngel

Alanlar

asHareket...

boyut

sagaHare...

yukHarek...

Metotlar

Arilar

Draw

Drawari

DrawRed...

Generate...

Move

ChestGene...

Static Sınıf

Metotlar

Generate...

CopperChest

Sınıf

→ TreasureChest

Metotlar

CopperCh...

Draw

Daglarkış

Sınıf

→ HareketsizEngel

Alanlar

boyut

Metotlar

Daglarkış

Draw

Generate...

Daglaryaz

Sınıf

→ HareketsizEngel

Alanlar

boyut

Metotlar

Daglaryaz

Draw

Generate...

Duvarlar

Sınıf

→ HareketsizEngel

Özellikler

Image

Metotlar

Draw

Duvarlar

Generate...