Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü

Programa Laboratuvarı II

Otonom Hazine Avcısı

Oğuzhan Çelik-190202105

Can Güneri-190202094

Projenin Özeti

Programlama Laboratuvarı II Projesi olarak bizden "Otonom Hazine Avcısı" adındaki bir algoritma geliştirmemize yönelik oyun ve arayüz tasarımı içeren bir uygulama yapmamız beklenmektedir.

Otonom Hazine Avcısı projesi, Visual Studio kullanılarak geliştirilmiş bir Windows Form uygulamasıdır. Bu proje, bir karakterin otomatik olarak hazine toplamasını ve engellerden kaçınmasını sağlayan bir oyun olarak tasarlanmıştır. Projenin temel amacı, kullanıcıların karakterin hareketlerini gözlemleyerek hazine toplama ve engellerden kaçınma yeteneklerini görmelerini sağlamaktır.

Proje kapsamında, harita oluşturma, karakter hareketleri, engel tespiti ve hazine toplama gibi temel işlevler yer almaktadır. Oyunun başlangıcında kullanıcılar, harita boyutlarını ve karakterin başlangıç konumunu belirleyebilirler. Daha sonra karakter otomatik olarak hazineyi bulmak için haritayı tarar ve engelleri aşmaya çalışır. Projede kullanılan teknolojiler arasında C#, Windows Form uygulama geliştirme, grafik arayüz tasarımı ve algoritma kullanımı yer almaktadır.

GİRİŞ

C# programlama dili; açık kodlu, nesneye yönelik,zeminden bağımsız, yüksek verimli, çok işlevli, yüksekseviye, derlenen bir dildir.

Visual Studio; Microsoft tarafından geliştirilen bir C# geliştirme ortamıdır (IDE) ve ücretsiz olarak dağıtılmaktadır.Özellikle kullanıcı arayüzü tasarımında sağladığı kolaylıklardan dolayı tercih edilmektedir.

Otonom Hazine Avcısı projesi, bilgisayar bilimleri ve yapay zeka alanlarında önemli bir konuyu ele alır: otonom karar alma ve hareket etme yetenekleri. Bu proje, bir karakterin belirli bir haritada otomatik olarak hazine aramasını ve engellerden kaçınmasını sağlayan bir oyunu simüle eder. Günümüzde otonom sistemlerin ve yapay zeka uygulamalarının gelişimi, benzer projelerin daha da önem kazanmasını sağlamıştır.

Otonom Hazine Avcısı, kullanıcıların temel olarak oyun içindeki karakterin otomatik karar verme yeteneklerini gözlemlemelerini sağlar. Kullanıcılar, harita boyutlarını ve karakterin başlangıç konumunu belirleyerek oyunu başlatır. Karakter daha sonra haritayı tarar, hazineyi bulmaya çalışırken engellerden kaçınır ve belirlenen hedefe ulaşmaya çalışır. Bu süreçte karakterin algoritmalı karar verme yetenekleri ve engellerle etkileşimi incelenir.

1. TEMEL BİLGİLER

Projede Form1.cs , Dag.cs , Kaya.cs , Duvar.cs , Agac.cs, Uygulama.cs , Program.cs, Karakter.cs , Lokasyon.cs, DinamikEngel.cs, gibi classlar bulunmaktadır.Ayrıca form sınıfı, DinamikEngelResim ve HareketsizEngelResim klasörleri yer almaktadır.

2. YÖNTEM

Otonom Hazine Avcısı projesinde, geliştirme sürecinde çeşitli metotlar kullanılmıştır. Bu metotlar, projenin farklı aşamalarında işlevlerini yerine getirmek üzere tasarlanmıştır.

1. İlk olarak, GirisEkraniGoster() metodu, oyunun başlangıç ekranını göstermek için kullanılır. Bu metodun çağrılması, Form1 sınıfının Form1_Load() olayına eklenen kodlar aracılığıyla gerçekleşir. Form1_Load() olayı, Windows Forms uygulamalarında form yüklendiğinde çalıştırılan bir olaydır. Dolayısıyla, projenin başlatılmasıyla birlikte bu metot da otomatik olarak çalışır ve kullanıcıya harita boyutunu belirlemesi için bir form açar

2. HaritaBoyutuFormuAc() metodu da benzer şekilde çağrılır ve kullanıcıya harita boyutunu seçme imkanı sunar. Bu metot, kullanıcının harita genişliği ve yüksekliğini belirlemesi için bir form oluşturur ve bu formun butonlarına tanımlanan Click olaylarıyla kullanıcının seçimlerini işler.

3. HaritayiOlustur() metodu, harita boyutları belirlendikten ve kullanıcı seçimlerini yaptıktan sonra çağrılır. Bu metot, belirlenen harita boyutlarına göre oyun haritasını oluşturur ve üzerine ızgaraları çizer. Ayrıca, hareketsiz engelleri (ağaçlar, duvarlar, granitler, dağlar) bu metot aracılığıyla haritaya yerleştirir.

4. boyutFormunuGuncelle() metodu, harita boyutu formunun güncellenmesi ve ekranda ortalanması için kullanılır. HaritaBoyutuFormuAc() metodunda seçilen harita boyutlarına göre formun boyutu güncellenir ve haritanın ekran üzerinde ortalanmasını sağlar.

5. ResmiIzgaraUzerineYerlestir() metodu, resimleri ızgara üzerine yerleştirmek için kullanılır. Bu metot, hareketsiz engellerin, karakterin ve diğer nesnelerin görüntülerini harita üzerinde konumlandırır. Örneğin, ağaç, duvar, dağ gibi statik engelleri ve karakter resmini bu metot aracılığıyla haritaya yerleştirir.

6. EngelleriYerlestir() metodu, haritaya çeşitli engellerin yerleştirilmesi için kullanılır. Rastgele olarak ağaçlar, duvarlar, granitler, dağlar ve dinamik engeller (kuşlar ve arılar) bu metot aracılığıyla haritaya yerleştirilir. Bu metot, rastgele sayı üreterek belirlenen bir ihtimale göre farklı engelleri yerleştirir ve çakışma kontrolü yaparak engellerin birbirleriyle çakışmasını önler.

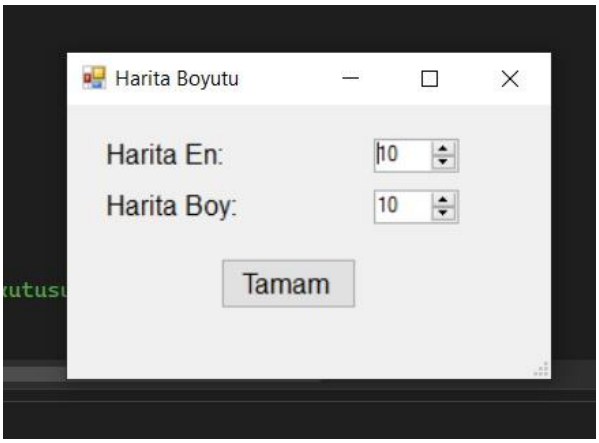
Bu metotlar, Otonom Hazine Avcısı oyununun geliştirme sürecinde temel işlevleri yerine getirir. Projenin çalışma mantığı, bu metotların doğru sıralama ve kullanımıyla sağlanır ve kullanıcıya etkileşimli bir oyun deneyimi sunulur.

Form1.cs dosyası, C# programlama dilinde yazılmış ve Visual Studio geliştirme ortamında geliştirilen bir Windows Form uygulamasıdır. Bu dosya, proje başlangıcında en önemli dosyalardan biridir çünkü uygulamanın temel mantığını ve kullanıcıyla etkileşimi sağlayan metotları içerir.

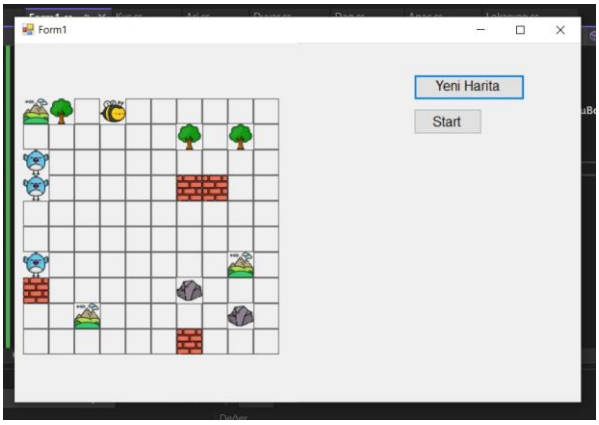
Form1.cs Dosyasının Açılması ve Başlangıç Metotları:
İlk olarak, Visual Studio'da Form1.cs dosyasını açarak projeyi başlatıyoruz. Bu dosya, Form1 sınıfını içerir ve programın başlangıcında çalışacak olan metotları barındırır. Bunlar arasında Form1() ve Form1_Load() metotları bulunur. Form1() metodu sınıfın yapıcı metodu olarak, Form1_Load() ise form yüklendiğinde çalışacak olan metottur.

Giriş Ekranını Gösterme:
Form1_Load() metodu içinde, GirisEkraniGoster() metodu çağrılır. Bu metot, kullanıcıya başlangıç ekranını gösterir ve harita boyutlarını belirlemesi için kullanıcıyı yönlendirir.

Harita Boyutlarını Belirleme:
Kullanıcı GirisEkraniGoster() ekranında harita boyutlarını seçtikten sonra HaritaBoyutuFormuAc() metodu çağrılır. Bu metot, bir form açarak kullanıcıya harita genişlik ve yükseklik değerlerini seçme imkanı verir.



Harita Oluşturma ve Engelleri Yerleştirme:
HaritaBoyutuFormuAc() metodu tarafından belirlenen boyutlara göre HaritayiOlustur() metodu çağrılır. Bu metot, haritanın ızgaralarını çizer ve engelleri rastgele olarak yerleştirir. EngelleriYerlestir() metodu da burada kullanılır ve statik veya dinamik engellerin haritaya yerleştirilmesini sağlar.



Formun Boyutunu Güncelleme ve Görsel İyileştirmeler:
Harita oluşturulduktan ve engeller yerleştirildikten sonra boyutFormunuGuncelle() metodu çağrılır. Bu metot, formun boyutunu harita boyutlarına uygun olarak günceller ve görsel olarak daha kullanıcı dostu bir deneyim sağlar.

Bu süreç, Form1.cs dosyasında yer alan metotların projenin başlangıcında nasıl kullanıldığını gösterir. C# programlama dili, Visual Studio geliştirme ortamı ve Windows Form uygulaması, Otonom Hazine Avcısı projesi için uygun seçimlerdir çünkü güçlü bir dil ve geliştirme aracı ile kullanıcı dostu bir arayüz oluşturmak mümkündür.

3. SONUÇ

Projenin geliştirme sürecinde birçok önemli adım atılmış ve başarıyla tamamlanmıştır. Otonom Hazine Avcısı projesi, Windows Forms kullanarak geliştirilmiş olup, temel amacı bir oyun deneyimi sunmaktır. Projede kullanılan çeşitli metotlar ve sınıflar sayesinde, oyunun başlangıcından bitişine kadar akıcı bir deneyim sağlanmıştır.

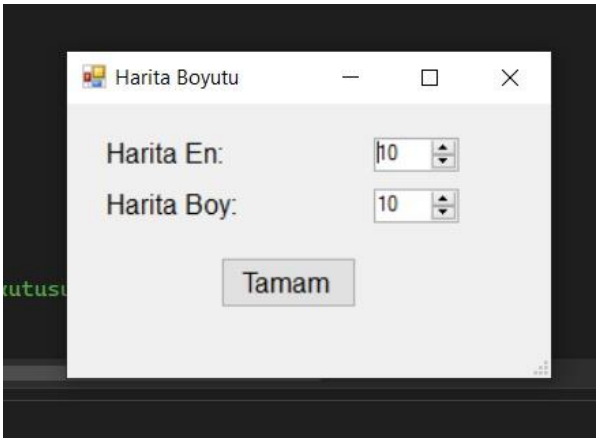
Geliştirme sürecinde karşılaşılan zorluklar, çözüm önerileriyle aşılmış ve projenin istenilen sonuca ulaşması sağlanmıştır. Özellikle harita oluşturma, engelleri yerleştirme ve karakterin hareketlerini yönetme gibi temel özelliklerin başarılı bir şekilde işlevsel hale getirilmesi, projenin kalitesini artırmıştır.

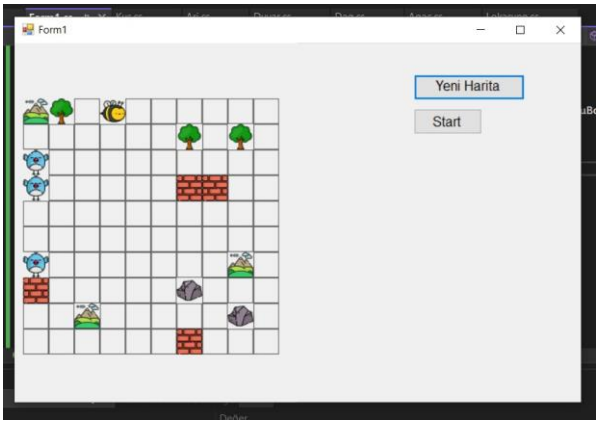
Projenin sonuçları incelendiğinde, kullanıcıların etkileşimli bir oyun deneyimi yaşadıkları görülmektedir. Harita boyutu seçimi, engellerin rastgele yerleştirilmesi, karakterin hareketleri gibi özellikler, projenin başarılı bir şekilde tamamlanmasına katkı sağlamıştır.

Geliştirme sürecinde elde edilen tecrübeler ve projenin sonuçları, gelecekteki benzer projeler için önemli bir referans teşkil etmektedir. Projenin başarıyla tamamlanması, ekip çalışmasının ve programlama becerilerinin olumlu sonuçlarını ortaya koymuştur.

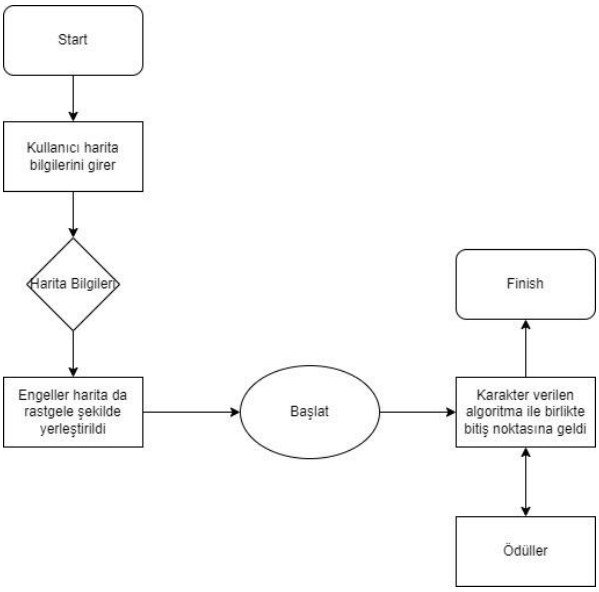
Sonuç olarak, Otonom Hazine Avcısı projesi, kullanıcıların keyifli bir oyun deneyimi yaşamalarını sağlayan başarılı bir uygulamadır. Geliştirme sürecindeki tüm adımların titizlikle planlanması ve uygulanması, projenin istenilen kalitede tamamlanmasını sağlamıştır.

4. *DENEYSEL SONUÇLAR*





5. *AKIŞ ŞEMASI*



6. *KAYNAKÇA*

https://www.youtube.com/watch?v=MqFc9YZBDHM

https://www.youtube.com/watch?v=-qkhhmX4nLU

https://stackoverflow.com

https://www.youtube.com/watch?v=Y37-gB83HKE

https://codeahoy.com/index.html

https://www.youtube.com/watch?v=lBvOX0lWd6E&list=PLvW0netwgVDRqA7VtY-A_jpu6D-6t3V2P

https://www.flaticon.com/search