

# PROGRAMLAMA LABORATUVARI-II

## PROJE-I

Sümeýra USTA  
Bilgisayar Mühendisliđi II. Öğretim  
Kocaeli Üniversitesi  
Kocaeli, Türkiye  
220202070@kocaeli.edu.tr

Tuba Nur AKSIN  
Bilgisayar Mühendisliđi II. Öğretim  
Kocaeli Üniversitesi  
Kocaeli, Türkiye  
220202078@kocaeli.edu.tr

**Özetçe**—Bu rapor, Kocaeli Üniversitesi Bilgisayar Mühendisliđi Bölümü Programlama Laboratuvarı II dersi kapsamında gerçekleştirilen "Otonom Hazine Avcısı" projesini tanıtmaktadır. Projede, nesneye yönelik programlama ve veri yapıları bilgisinin kullanılmasıyla, Unity platformunda C programlama dili kullanılarak bir karakterin rastgele oluşturulan bir harita üzerinde çeşitli hazineleri topladıđı bir oyun tasarlanmıştır.

**Anahtar Kelimeler**—otonom, veri yapıları, en kısa yol, oyun, hazine toplama, nesneye yönelik

### I. GİRİŞ

"Otonom Hazine Avcısı" projesi, programlamada nesneye yönelik yaklaşımların ve veri yapılarının kullanımını pekiştirmeyi amaçlamaktadır. Proje kapsamında, karakterin en kısa yoldan tüm hazine sandıklarının toplamasını sağlayacak bir algoritmanın tasarlanması ve uygulanması gerekmektedir.

### II. YÖNTEM

#### A. Haritanın Oluşturulması

Kullanıcıdan haritanın boyut bilgisi alınır ve buna göre harita oluşturulur. Her harita oluşturulduğunda yeni bir harita üretilir.

#### B. Harita İçeriğinde Bulunması Gerekenler

Harita üzerinde yol, engeller ve hazine sandıkları bulunur. Sabit ve hareketli engeller farklı özelliklere sahiptir.

#### C. Haritada Başlangıç ve Bitiş Noktalarının Belirlenmesi

Karakterin başlangıç noktası uygun karelerden rastgele belirlenir. Bitiş noktası ise toplanması gereken en son hazine sandığının konumu olarak belirlenir.

#### D. Karakterin Hareketi

Karakter, otomatik olarak en kısa yol üzerinde hareket eder. Harita üzerindeki sisli bölgeler karakterin henüz keşfetmediđi alanları temsil eder.

#### E. En Kısa Yol Algoritması

En kısa yol algoritması kullanılarak karakterin tüm hazine sandıklarını top

#### F. Uygulama Geliştirme Süreci

Proje geliştirme süreci adım adım ilerler, her adımda sınıfların kodlanması, test edilmesi ve arayüzün tasarlanması gibi işlemler Unity oyun geliştirme ortamında gerçekleştirilir.

### III. SONUÇLAR

Bu projenin amacı, otonom hazine avcısı karakterinin, rastgele oluşturulan bir harita üzerindeki hazine sandıklarını en kısa sürede toplayabilmesi için geliştirilen bir algoritmanın tasarlanması ve uygulanmasıydı. Projeyi gerçekleştirirken nesneye yönelik programlama ve veri yapıları bilgisi kullanıldı. Aşağıda projenin sonuçları ve elde edilen bulgular detaylı bir şekilde sunulmaktadır:

#### A. Harita Oluşturma Algoritması

Projede harita oluşturulması için etkili bir algoritma geliştirildi. Harita boyut bilgisine göre rastgele olarak yol, engeller ve hazine sandıkları yerleştirildi. Harita her uygulama başladığında yeniden üretilir ve her adımda rastgele bir harita üretildiđi doğrulandı. Oluşturulan haritaların sol tarafı kış, sağ tarafı yaz temasına uygun olarak tasarlandı ve görseller bu temalara göre deđiştirildi. Ayrıca, harita içeriğinde tüm hazine sandıklarının ulaşılabilir olduđu doğrulandı.

#### B. Engeller ve Hareketli Nesneler

Sabit ve hareketli engellerin yanı sıra hazine sandıkları da harita içeriğine eklendi. Sabit engeller arasında ağaclar, dađlar, kayalar ve duvarlar bulunurken, hareketli nesneler arasında kuşlar ve arılar yer aldı. Her harita oluşturulduğunda en az 20 adet sabit engel ve 3 adet hareketli nesne yerleştirildi. Hareketli nesnelerin takip ettiđi yol kırmızı renkle gösterildi.

#### C. Karakterin Hareketi ve Hazine Toplama

Karakterin başlangıç ve bitiş noktaları belirlendi ve karakter, en kısa yol algoritması kullanılarak hazine sandıklarını toplamak üzere hareket etti. Karakter 3x3 bir alanı görebildi ve otomatik olarak hareket etti. Karakter her adımda yeni keşfedilen kareleri sisli olarak gösterdi ve keşfedilen nesnelerin türlerini ekranda bildirdi.

#### D. En Kısa Yol Algoritması ve Sonuçlar

Projede en kısa yol algoritması geliştirilerek karakterin hazine sandıklarını en kısa sürede toplamayı sağlandı. En kısa yol, ızgara üzerinde yeşil renkle gösterildi ve karakterin toplam adım sayısı ekranda yazdırıldı. Sistem başarıyla çalıştırıldı

ve karakter her durumda tüm hazine sandıklarını toplamayı başardı.

#### E. Arayüz ve Görsellik

Projenin arayüzü net ve açık bir şekilde tasarlandı. Yeni harita oluşturulması ve oyunun başlatılması için kullanıcı dostu butonlar eklendi. Harita içeriğinin sisli ve net görünümüleri arasında geçiş sağlandı. Ayrıca, en kısa yolun görsel olarak gösterilmesi ve karakterin hareketinin izlenmesi için etkili bir arayüz oluşturuldu.

#### IV. DENEYSEL SONUÇLAR

Projemizin deneysel sonuçları, geliştirilen algoritmanın ve uygulamanın performansını ve etkinliğini değerlendirmek için gerçekleştirilen testlerin sonuçlarına dayanmaktadır. Aşağıda, oyunun geliştirilme ve çalışma anında elde edilen görsel çıktı ve şemalar detaylı bir şekilde sunulmaktadır:

##### A. Ekran Çıktısı ve Diyagramlar

#### KAYNAKLAR

- [1] <https://bilgisayarkavramlari.com/2009/03/02/a-yildiz-arama-algoritmasi-a-star-search-algorithm-a/>
- [2] <https://metehanugus.com/unity-sekmelerin-detayli-bir-inceleme/>
- [3] Kocaeli Üniversitesi, Bilgisayar Mühendisliği Bölümü, Veri Yapıları ve Algoritmaları Dersi Notları





