

T.C.
SAKARYA UYGULAMALI BİLİMLER ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
Bilgisayar Mühendisliği Bölümü



2024 – 2025 GÜZ DÖNEMİ

VERİ YAPILARI VE ALGORİTMA RAPORU

HAZIRLAYAN:
24010903131 - CEREN ÖZKAN
24010903128 - YUSUF POLAT

Labirent (Maze) Oyunu: Backtraking Uygulaması

Projenin Amacı ve Tanıtımı:

Bu proje, bir oyuncunun labirentte doğru yolu bulmaya çalıştığı bir simülasyonu gerçekleştirmektedir. Kullanıcı, labirentte belirtilen başlangıç noktasından başlayarak, bitiş noktasına ulaşmak için çeşitli yönlerde hareket eder. Labirent, engellerle dolu olup, oyuncunun bu engelleri aşabilmesi için geçiş yapabileceği boş alanları bulması gerekmektedir. Herhangi bir yanlış adım veya engelle karşılaşması durumunda, yedek hareketler (geri alma) yapılabilir.

Proje Adımları ve Yapılanlar:

Bu proje, C++ dilinde yazılmıştır ve ana bileşenler şu şekildedir:

1) Konum Yapısı:

- Oyuncunun harita üzerindeki mevcut konumu ve yönünü tutan bir Konum yapısı tanımlandı.
- Bu yapıda x ve y koordinatları ile oyuncunun hareket yönünü belirlemek için bir Yon enum yapısı kullanıldı.
- Konum sınıfında, oyuncunun hareketlerini sağ, sol, yukarı ve aşağı yönlerde yapmak için çeşitli metodlar tanımlandı.

2) Labirent Yapısı:

- Labirent sınıfı, oyun haritasını ve oyuncunun konumunu tutmak için kullanıldı.
- Labirentin haritası, bir 2D karakter dizisi ile temsil edilmiştir. Engeller # karakteri ile, boş alanlar ise ' ' (boşluk) ile ifade edilmiştir. Geçilen yollar ise '-' karakteri ile işaretlenmiştir.
- Labirent sınıfı, haritanın üzerinde hareket etmeyi ve yeni bir adım atmaya sağlayan fonksiyonlar içerir. Bu fonksiyonlar, oyuncunun geçiş yapabileceği yerleri kontrol eder ve labirentin doğru şekilde güncellenmesini sağlar.

3) Stack (Yığın) Kullanımı:

- Oyuncunun hareketlerini geri alabilmek için bir Stack (yığın) veri yapısı kullanıldı. Yığın, her hareketi kaydederek oyuncunun geri gitmesini sağlar.
- Yığın, Konum türünde bir eleman tutar ve oyuncunun her adımda gittiği konumu saklar. Bu sayede oyuncu, bir yanlış adım attığında geri gidip doğru yolu bulabilir.

4) Harita Okuma ve Yazdırma:

- Labirent sınıfında Harita.txt dosyasından harita verileri okunarak harita dizisine aktarılır. Bu dosyada engeller, boş alanlar ve başlangıç/bitiş noktaları belirtilmiştir.
- Labirent::yaz() fonksiyonu, haritayı ekrana yazdırır ve oyuncunun bulunduğu konum ile yönü doğru şekilde gösterir.

5) Hareket Algoritması ve Kontroller:

- Oyuncunun her hareketi, geçilecek alanın engel olup olmadığına bakılarak yapılır. Eğer geçiş yapılabilir bir alan ise, oyuncu bu alana geçer ve bu alan '-' ile işaretlenir.
- Ayrıca, geçilen alanda engel varsa (duvar veya daha önce geçilmiş bir yol), oyuncu ilerleyemez.
- Oyuncunun çıktığı yönler, her adımda yığın veri yapısına kaydedilerek geri alma (undo) işlemleri yapılabilir.

Kaynaklar:

<https://github.com/mfadak/DataStructures/tree/main/Week7/Labirent>