**LABİRENT OYUNU**

*YAĞMUR ÇATAK , SİDAR İŞSEVER*

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

170202068@kocaeli.edu.tr , 170202099@kocaeli.edu.tr

* 1. **Problem Tanımı:**

Labirent üzerinde bir giriş noktası ve çıkış noktası belirleyerek, en uygun yolu bulmak temel amacımız. Labirentteki 1 rakamı yol olduğunu, 0 rakamı ise olmadığını bize bildirmektedir. Giriş noktasından başlayarak yol olan bütün yönleri kontrol ederek, çıkış noktasına ulaşmak amaçlanmaktadır.

* 1. **Yapılan Araştırmalar:**

Yol bulma algoritmasında 1 değerini bulduğumuzda, etrafını da kontrol etmeyi amaçladık. Fakat bir yoldan, birden fazla yola gitme seçeneği oluşabilmektedir. Bu noktada, dallanmanın olduğu konuma tekrardan erişebilmek için yığın yapısını araştırdık. Yığın yapısı için gerekli pop ve push fonksiyonlarını algoritmamıza uygun olarak geliştirdik.   
 Bağlı liste kullanarak matris oluşturabilmek için bağlı listede matris yapısı nasıl kullanılır sorusunu kendimize sorarak, araştırma yapıp kendi algoritmamızı oluşturmaya çalıştık.

**1.3Genel Yapı:**

Labirentin yollarını ve duvarlarını rastgele olarak yol olduğunu temsil eden 1 ve duvarı temsil eden 0 değerlerini atayarak, matris formda oluşturulan bağlı liste de tutulmaktadır. Kullanıcıdan alınan, başlangıç ve bitiş noktalarının ilk önce yol olup olmadığının kontrolünü yaparak; eğer seçilen yerler duvarı temsil eden 0 değerine eşit ise tekrardan başlangıç ve bitiş noktası bilgisi istenmesi sağlanmaktadır.

Başlangıç noktası belirlenen labirent de, ilk olarak bulunduğumuz noktanın etrafındaki tüm yönleri kontrol ederek eğer birden fazla yol var ise oluşturulan yığın yapasına bulunduğumuz noktanın konumu eklenerek tüm yolların kontrolü yapıldıktan sonra yol bulunamazsa kaldığımız yere tekrardan geri dönebilmesi amaçlanmıştır. Geliştirilen algoritma gereğince yönlerin kontrolü sırasıyla; sağ, sol, yukarı ve aşağı olarak yapılmaktadır. Eğer bu yollardan birden fazlasında gidilecek yol var ise, sırasıyla dallanmayı kontrol eder. Örneğin, bulunan düğümün etrafında birden fazla yol var ise, ilk önce sağ yöne ilerleyecek ve eğer o yoldan aynı metotları kullanarak bitiş noktasına ulaşabiliyorsa yolu bulacak. Fakat eğer sağ yöne gittiğinde bir yol bulunamadıysa, yığından son kalınan düğümü çekerek; gidilmeyen yönde aramaya yapmaktadır.

Yığından çekilen dallanmış düğümün, bir yönüne gittikten sonra yol bulamayıp; diğer yöne gidebilmek için gidilen yönü gidildi olarak işaretlemek için color değişkenini tuttuk ve gidilen düğümden artık yol bulunamadığı için gidilen yerlerdeki değerleri 0 yaparak, yol olmadığı belirlenmektedir.

* 1. **Referanslar:**

http://www.paylasimevi.com/2011/03/15/c-labirentten-cikis-oyunu.html

http://www.geeksforgeeks.org/sparse-matrix-representation

http://bilgisayarkavramlari.sadievrenseker.com

https://ckaynak.com

http://turgayozgur.com

http://www.baskent.edu.tr

https://mehmetsalihdeveci.net

Harvey M.Deitel ve Paul J. Deitel, C ve C++,Sistem Yayıncılık,2004.

Muhammet Yorulmaz ve Seher Yorulmaz,Programlamayı C ile öğreniyorum,Palme Yayıncılık,2016.