LABİRENT ÇÖZEN ROBOT SİMÜLASYONU(UYGULAMASI)

Proje Ekibi: Simge Akosman,Kaan Çetiner,Merve Esenlik

Danışman: Doç.Dr.Murat Erşen Berberler

Amaç

Robot, istenilen görevleri yerine getirmek için tasarlanmış mekatronik sistemdir. Robotlar önceden programlanabileceği gibi otonom olarak çalışarak da istenilen görevleri yerine getirebilirler. Robotlar için çeşitli sınıflandırmalar vardır. Bunlar kullanıldığı yer, robotun sabit veya hareketli oluşu ya da çalıştığı ortam olabilir. Belirli bir ortamda hareket ederek istenilen görevleri yerine getirebilirler.

Yolların ya da geçitlerin çokluğu, karışıklığı yüzünden içinden kolay kolay çıkılamayan yerlere labirent denir. Bu çalışma, robot tarafından önceden bilinmeyen bir labirentin tanımlanması, ortam haritasının çıkartılması ve çözüme ulaşması için hazırlanmıştır.

Hedef

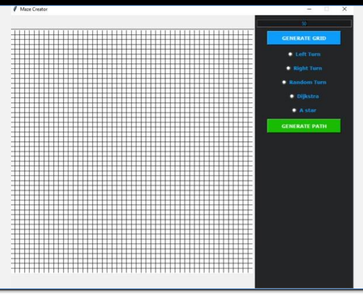
Bu projede, girilen labirentten çıkış için çözüm bulmaktır. MxM boyutundaki labirentleri DFS algoritmasını modifiye ederek oluşturduğumuz algoritmaları kullanarak çözüme ulaşmasından oluşacaktır.

Analiz

Robotun başlangıç noktasından bitiş noktasına ulaşması için DFS algoritması kullanılarak üç farklı algoritma modifikasyonu yapılmıştır.
Bunlar:

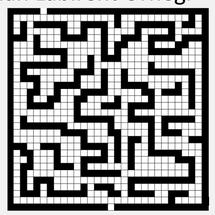
- DFS LEFT
- DFS RIGHT
- DFS RANDOM

Uygulama Arayüzü



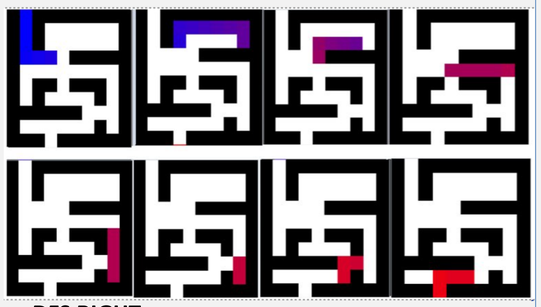
Uygulama

- Oluşturulan Labirent Örneği

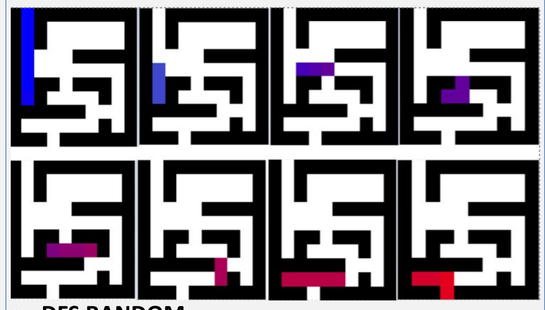


Aşağıdaki görsellerde DFS modifikasyonları 10x10 labirent üzerinde adım adım çalıştırılmıştır.

- **DFS LEFT**

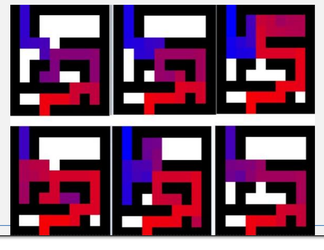


- **DFS RIGHT**



- **DFS RANDOM**

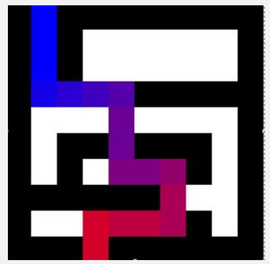
Aşağıdaki görsellerde 10x10 labirent üzerinde DFS Random için programın farklı çıktıları mevcuttur.



Uygulama

- **Dijkstra En Kısa Yol Algoritması**

Haritası bilinen labirentlerde en kısa yolu bulmak amacıyla kullanılan algoritmadır. Aşağıdaki görselde 10x10 labirent üzerinde en kısa yol gösterilmiştir.



Sonuç

DFS LEFT-DFS RIGHT-DFS RANDOM-DIJKSTRA-A*

Tip	Ortalama Hamle Sayıları			*EN KISA YOL*
	LEFT	*RIGHT*	*RANDOM*	*DIJKSTRA*
10x10	39,4	34,4	315,6	18,9
20x20	167	206,5	6184,6	60,4
30x30	234,8	213,8	42083,5	80,3
40x40	275	264	7122,3	90,5
50x50	441,2	465,4	6855,7	148,2
60x60	586	672,4	91483,7	186,3
70x70	669,3	755,8	386256	208,1
80x80	842	741,5	211618	262,1
90x90	1088,1	977,5	129916	307,2
100x100	1432,7	1151,7	314620	413,2

Değerlendirme

Labirentin başlangıç noktasından bitiş noktasına kadar rota çizebilmek amacıyla hazırlanan projemizde iki ana konu üzerinde çalışılmıştır. Bunlardan ilki ortamın tanımlanması, ikincisi ise rotanın bulunmasıdır. Gerçek robotun DFS algoritması ile arama yapması en iyi şekilde sonuç veren algoritmalardan birisidir. Tek robotlu yapılan çalışmalarda ortamın tanımlanması için iyi sonuçlar vermektedir. Dijkstra optimizasyonu ile DFS algoritması daha da verimli hale dönüştürülmüştür. Sol ve Sağ Duvar Takip algoritmaları, hedefe birden fazla yol ile gidilen, karmaşık ortamların analizinde kullanılması neredeyse imkansızdır. Rastgele üretilen bir labirentte DFS Sağ ve DFS Sol algoritmalarının aralarında bir üstünlük durumu yoktur. Oluşturulan labirentin durumuna göre Sol ağırlıklı ise DFS Sol , Sağ ağırlıklı ise DFS Sağ daha iyi sonuç elde eder. DFS Random algoritmasında ise hamlelerin rastgele olmasından dolayı hamle sayısı diğer algoritmalara göre oldukça fazladır.

Öneriler

Gelecek çalışmalarda ,DFS Random algoritmasının iyileştirilmesi, uygulamaya labirent oluşturma algoritmalarının eklenmesi, elimizde bulunan labirentlerin programa yüklenerek çözüme ulaşmasını sağlayabiliriz.

Kaynakça

- Fredman, Michael Lawrence; Tarjan, Robert E. (July 1987). "Fibonacci heaps and their uses in improved network optimization algorithms" (PDF). Journal of the Association for Computing Machinery. 34 (3): 596–615. CiteSeerX 10.1.1.309.8927. doi:10.1145/28869.28874.
- Fredman, Michael L.; Sedgewick, Robert; Sleator, Daniel D.; Tarjan, Robert E. (1986). "The pairing heap: a new form of self-adjusting heap" (PDF). Algorithmica. 1 (1–4): 111–129. doi:10.1007/BF01840439. S2CID 23664143.
- https://en.wikipedia.org/wiki/Taxicab_geometry
- https://en.wikipedia.org/wiki/Maze-solving_algorithm#:~:text=The%20maze%2Drouting%20algorithm%20is,for%20any%20grid%2Dbased%20maze.
- http://www.astrolog.org/labyrnth/algrithm.htm#solve
- http://weblog.jamisbuck.org/2011/2/7/maze-generation-algorithm-recap