

YAPAY ZEKA ÖDEV 1

Serkan YAMAÇ 15011908

Proje bağlantısı : <https://github.com/srknymc/PathFinder>

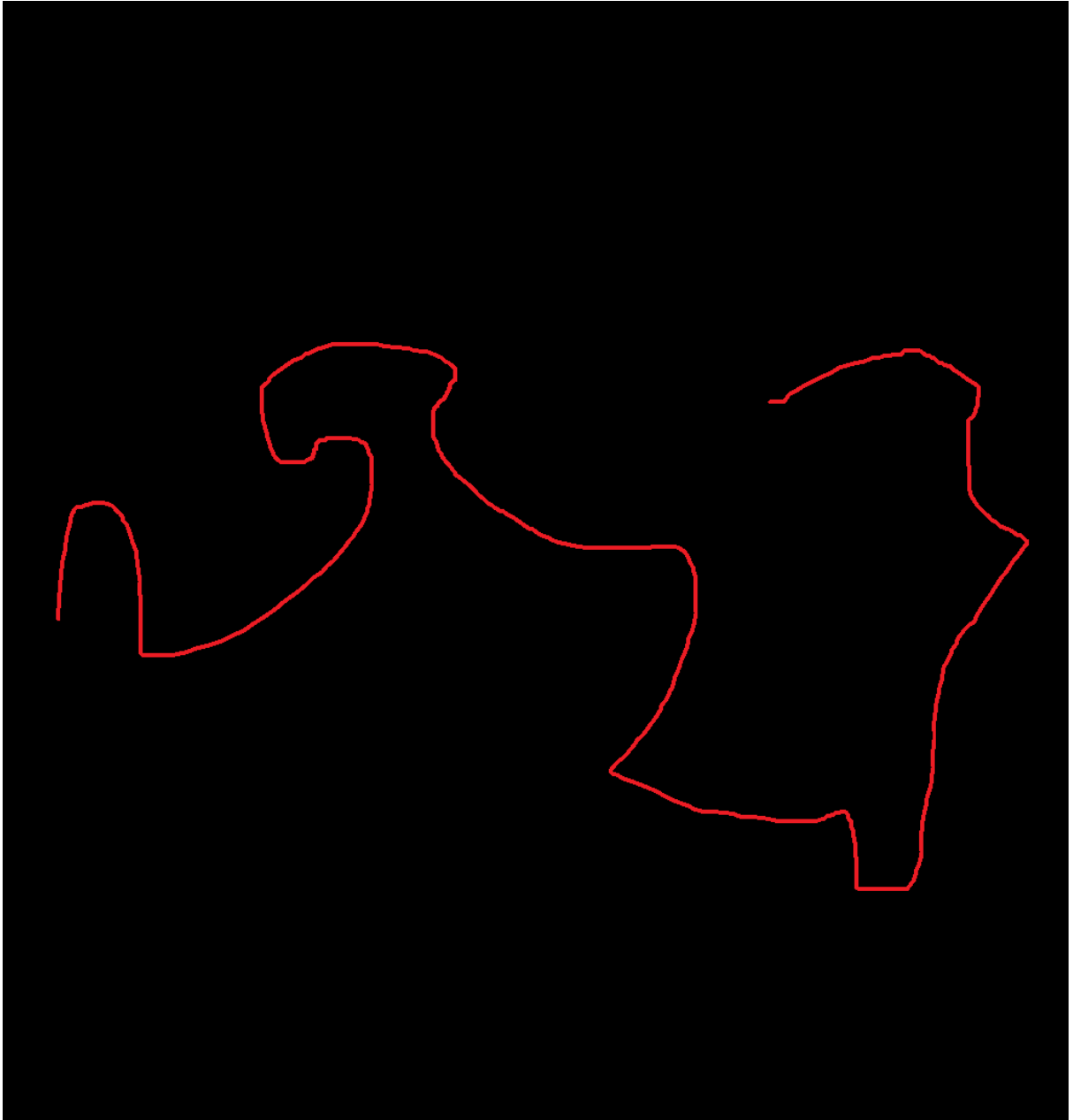
Bu ödevde, kullanıcının girdiği bir resmin, yine kullanıcının belirlediği 2 nokta arasındaki yol A* ve ilk en iyi arama algoritmaları ile bulunarak resim üzerine çizdirilmiştir. Bu algoritmalar geliştirilirken Heap Yapısı ve Normal Dizi veri yapıları üzerinde test edilmiştir. Heap üzerinde çalıştırıldığında aşağıda da görüldüğü üzere 4-5 kate kadar daha hızlı çalışarak fark yaratmıştır. Maliyet hesabında her iki algoritma için h fonksiyonu olarak Öklid bağıntısı kullanılmıştır. A* g fonksiyonu içinse ödev dökümantasyonunda verildiği üzere piksellerin kırmızı değerleri göz önünde bulundurulmuştur.

- Test Resmi İncelemesi

Algoritma	Süre	Stack Boyutu(MAX)	Listeden Çekilen Eleman Sayısı
BFS(Array)	0.023 sn	1858	958
BFS(Heap)	0.005 sn	1858	958
A*(Array)	6.01 sn	3463	460346
A*(Heap)	20.057 sn	3463	460346

Bu resim için tablo sonuçlarına bakacak olursak Best First Search(BFS) ve A* algoritmalarında stack boyutu ve pop edilen elemanların sayısında fark yoktur fakat sürede yaklaşık 4 kat fark oluşmuştur bu da Heap yapısının erişim hızını göstermektedir.

Görsel sonuçlar aşağıda ki gibidir. Sonuç Yeşil renk ile gösterilmiştir.

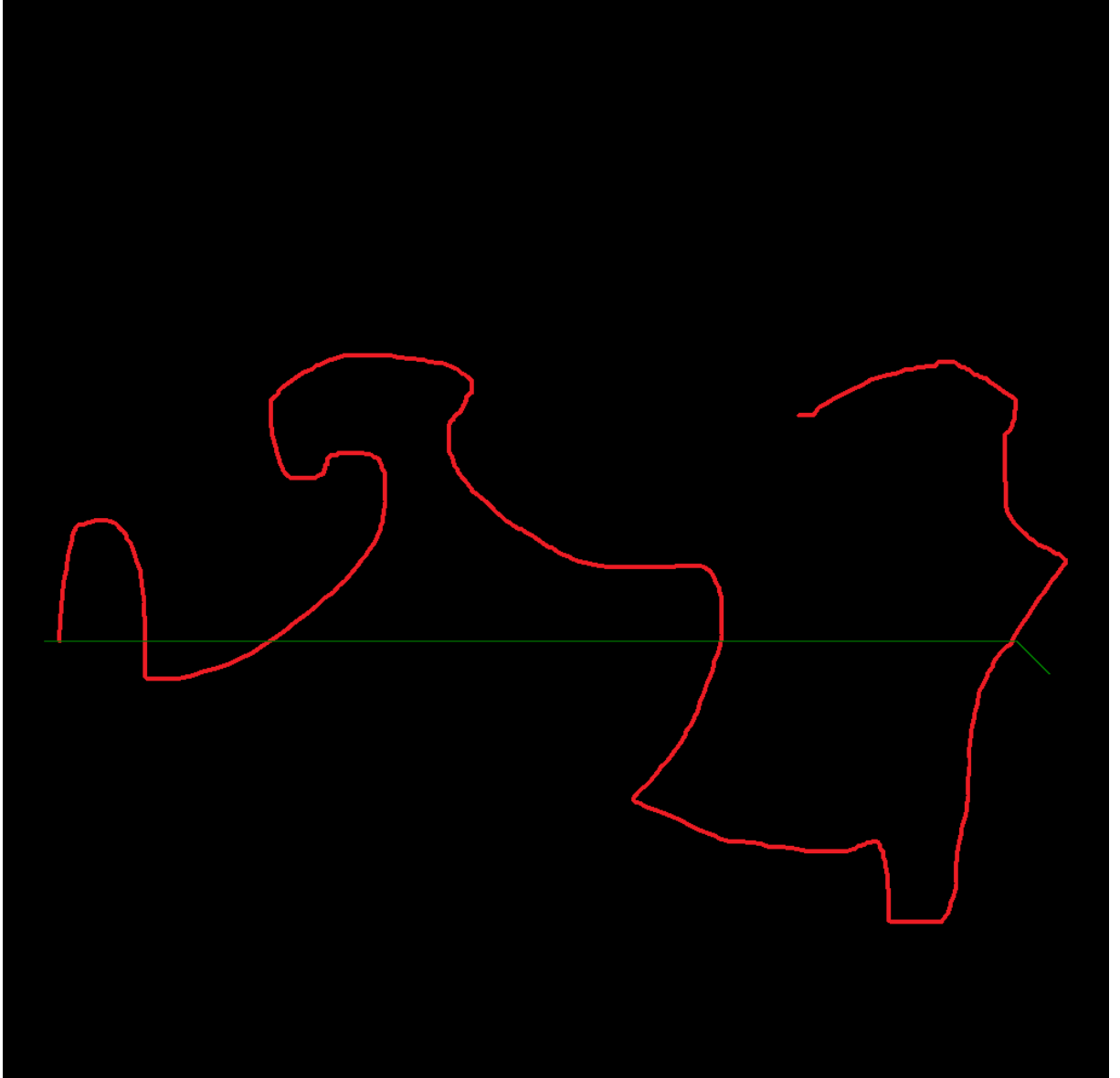


Orijinal Resim



A* Sonucu

Maliyet hesabında pikselin kırmızı değerin yüksek olduğu pikseller düşük maliyet kabul edildiği için algoritmamız kırmızı çizgiyi takip ederek sonuca yaklaşmıştır.



BFS Sonucu

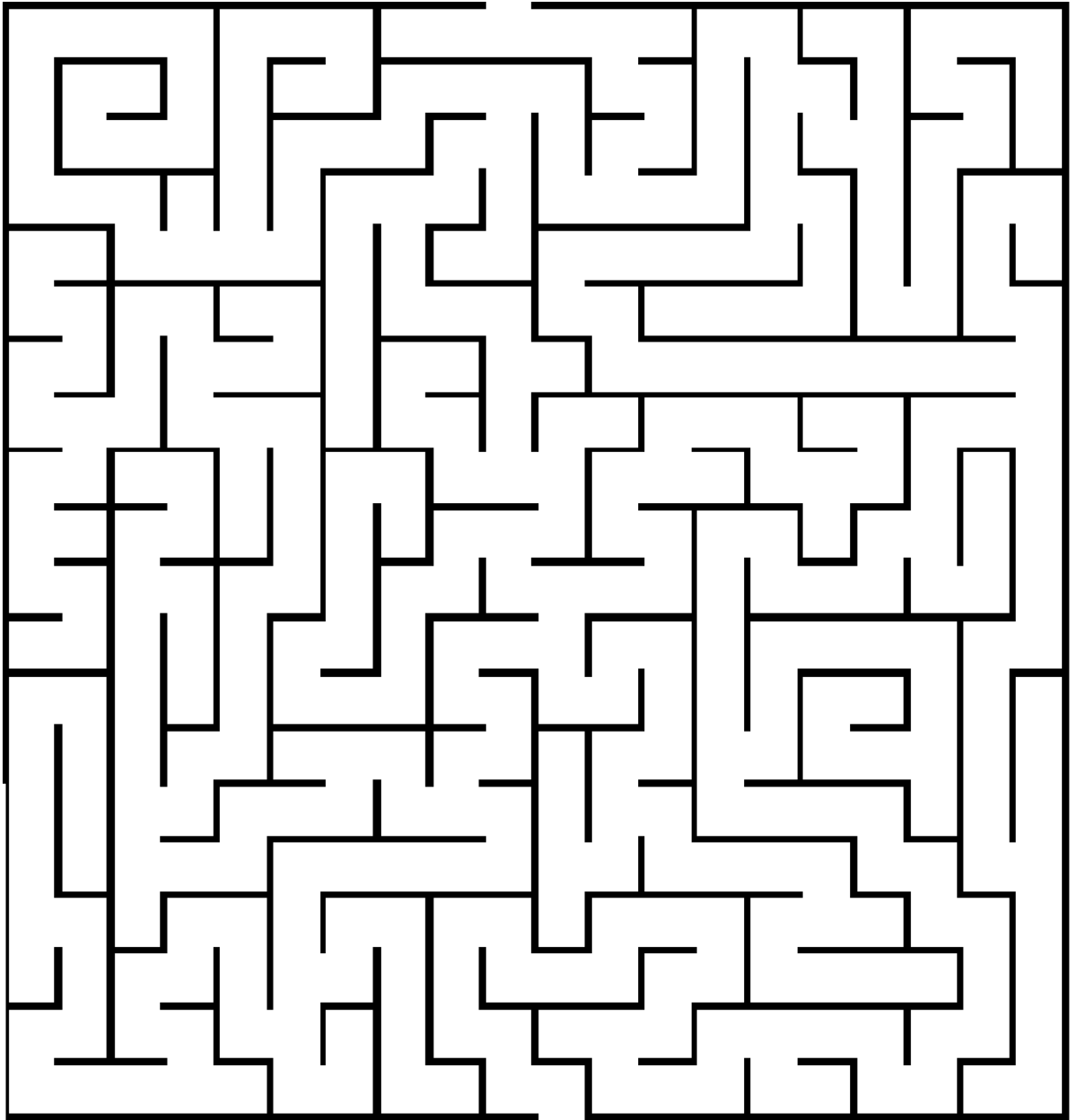
Maliyet hesabında Öklid bağıntısı kullanılarak algoritma çalıştırılmıştır.

- Labirent Resmi A* incelemesi

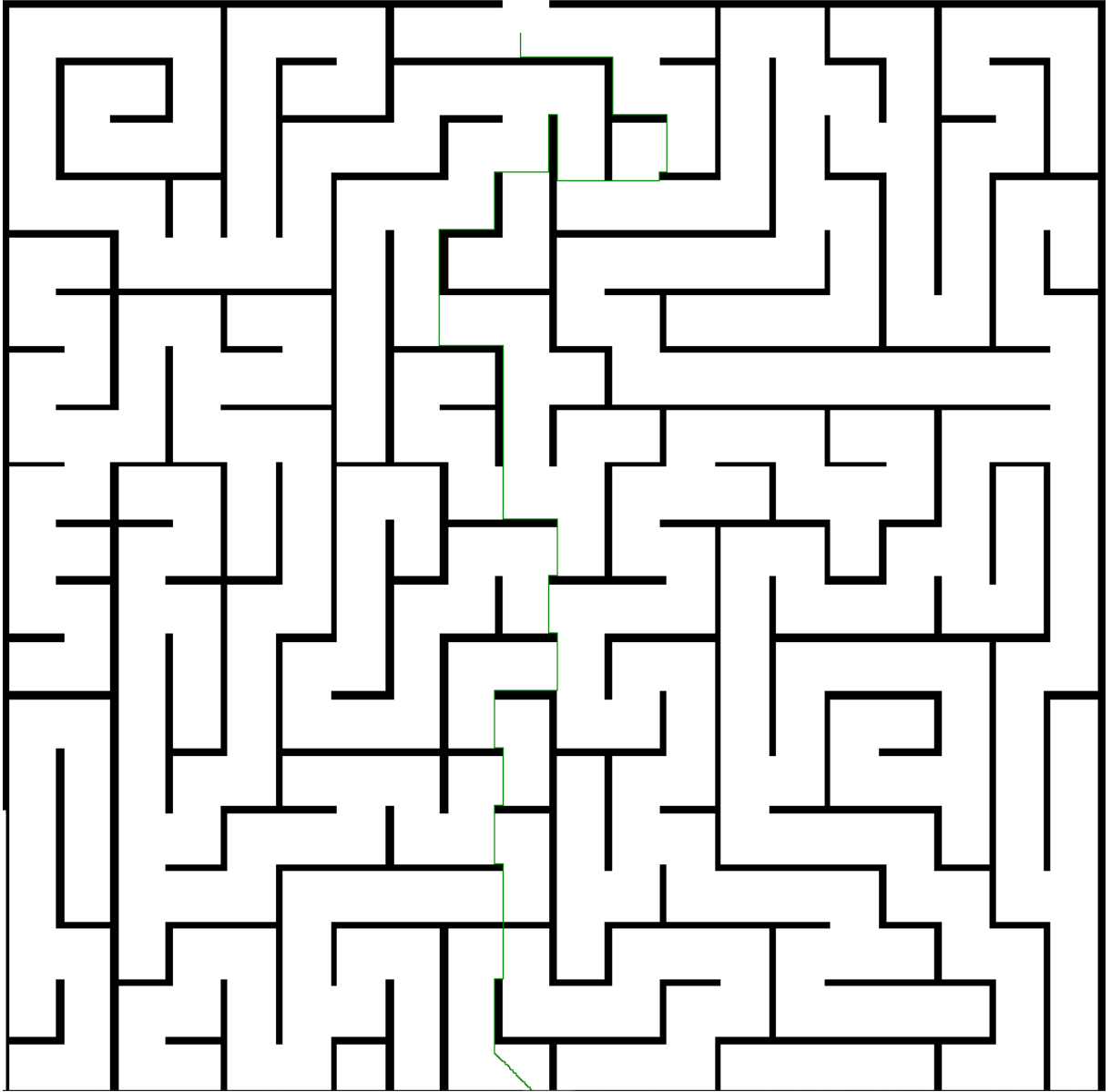
Algoritmalarımız duvarları siyah(rgb(0,0,0)) yolları beyaz(rgb(255,255,255)) olan bir labirent üzerinde test edilmiştir. A* sonuçları aşağıda ki gibidir. (BFS algoritmasında piksel renginin önemi olmadığı için bu teste dahil edilmemiştir.)

Algoritma	Süre	Stack Boyutu(MAX)	Listeden Çekilen Eleman Sayısı
A* Heap	36.354 sn	23391	1009521
A* Array	304.620 sn	23391	1009535

Kullanıcının girdiği resmin karmaşıklığı arttıkça Heap yapısın avantajı daha da ön plana çıkmaktadır. 9-10 kata kadar fark yaratmıştır.

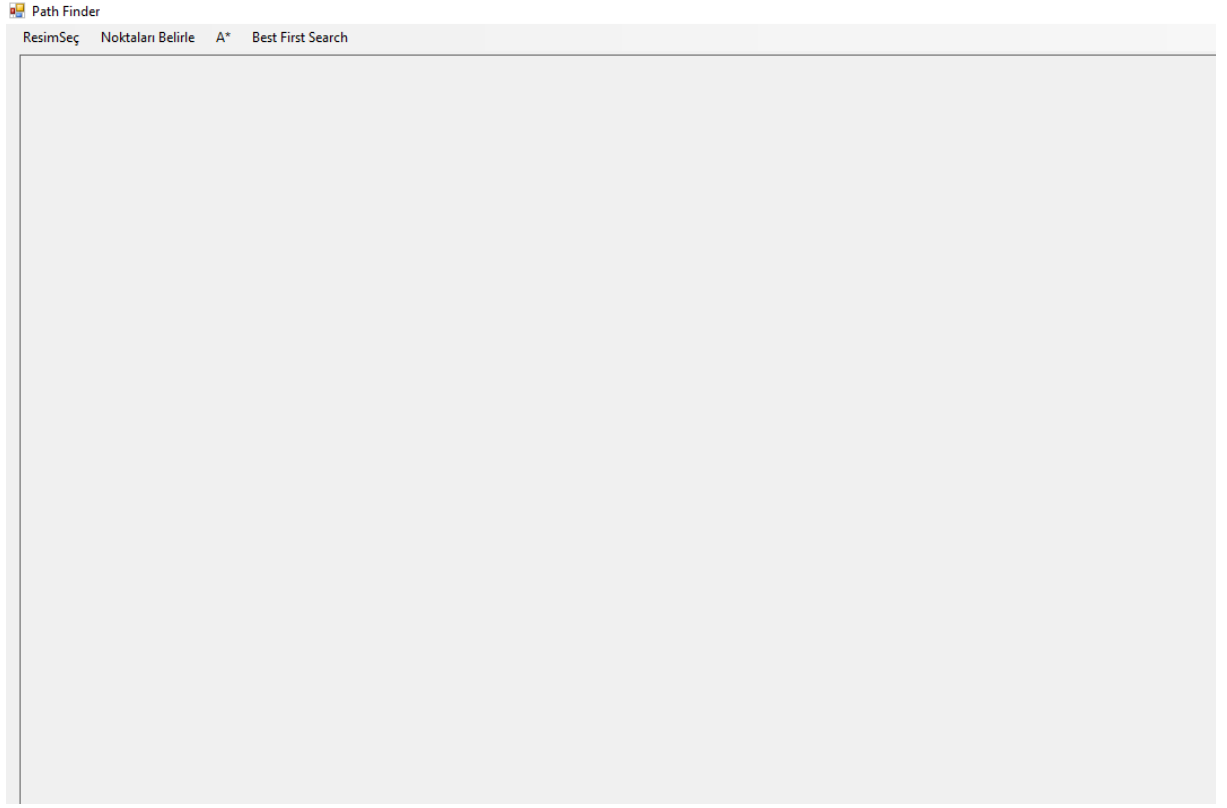


Orijinal Resim



A* sonucu

- Geliştirilen Uygulamanın Kullanımı



Program ana görünümü yukarıda ki gibidir. Öncelikle Resim Seç menüsünden resim seçilir ve daha sonra noktaları belirle menüsü altından başlangıç ve bitiş noktaları seçilir daha sonra çalıştırılmak istenen algoritma seçilir ve sonuç için yeni pencerenin açılması beklenir. Uygulamanın çalışabilir hali de github üzerinde bulunmaktadır.

<https://github.com/srknymc/PathFinder/tree/master/PathFinder/bin/Release>