

YAPAY ZEKA

A* - Best First Search

Rahmi Cemre Ünal 14011052

Canberk Güllüoğlu 18011602

Proje Bulguları:

1)Algoritma ilk başta 8 yöne hareket edebilecek şekilde kurgulandı. Daha sonra her piksel için 8 ayrı yön analizi, uzak başlangıç ve bitiş noktaları için hesaplama süresini çok uzattığı için bir pikselden gidilebilecek komşu sayısı sağ,sol,yukarı,aşağı olacak şekilde 4 olarak belirlendi.

2)Stackten eleman çekilirken bütün stacki sıralamak yerine, en küçük eleman bulunarak stackten çıkartıldı. Yine de minimum heap yapısı stack yapısına göre çok daha hızlı çalışıyor. Stack için minimum hesabı $O(n)$ eleman ekleme $O(1)$, minheap için minimum bulma $O(1)$, eleman ekleme $O(\log n)$.

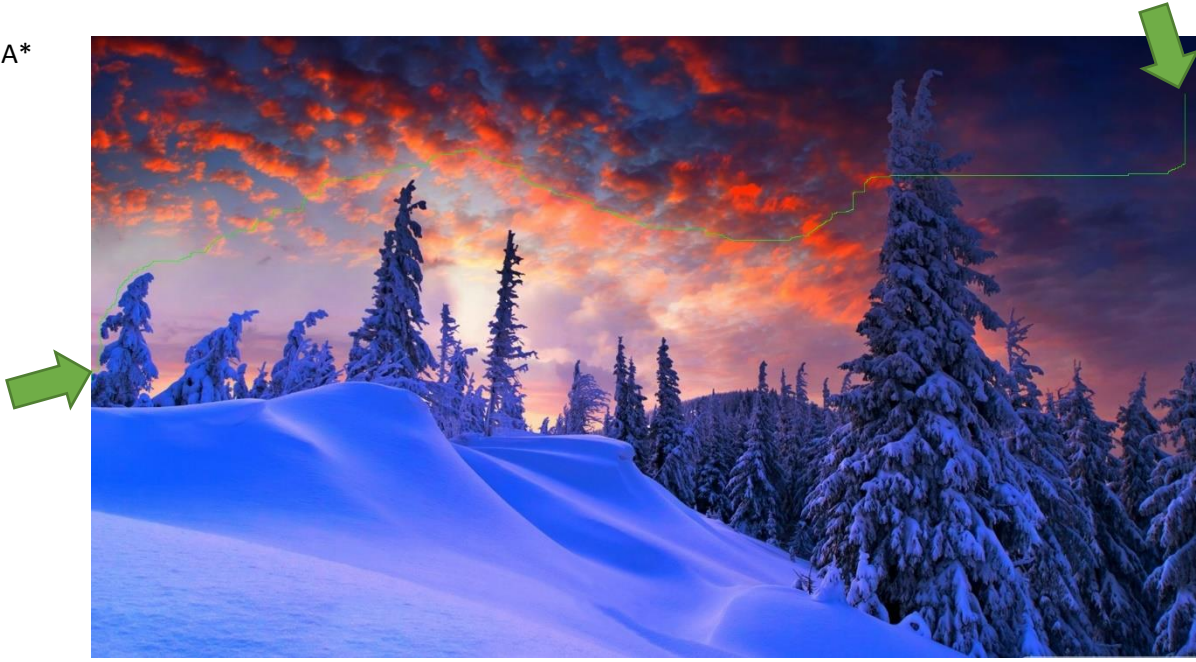
3) $h()$ hesaplanırken iki nokta arası uzaklık alındı [(1,1) noktası ile (1,10) arası mesafe 9 birim]. A^* için $g()$ hesabı, yeni ulaşılan noktaya hangi noktadan ulaşıldıysa o noktanın g maliyeti + 256 - yeni ulaşılan noktanın R değeri. (Bu sayede $R=255$ için yeni piksele geçme maliyeti 256-255 den 1 geliyor)

4) Bfs için maliyetin sadece h ile hesaplanması pixelin kırmızı değerine bakılmamasına sebep oldu. Bu yüzden bfs sadece iki nokta arasındaki uzaklığa bakarak en kısa yolu buluyor.

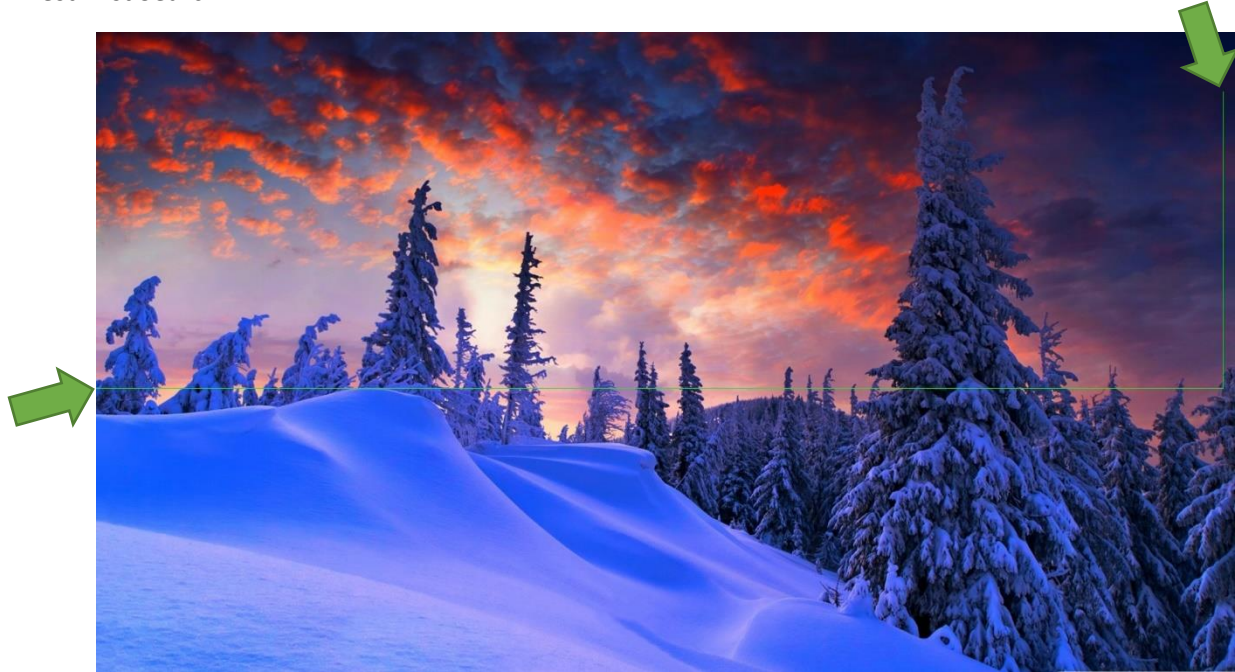
5) Yapılan denemeler sonucunda g hesaplanırken $\text{yeni_nokta.g} = \text{mevcut_nokta.g} + (256 - \text{yeni_nokta.r})/100$ olarak hesaplanmasına karar verildi. $(256 - \text{yeni_nokta.r})$ değerinin 100'e bölünmesinin sebebi, uzak başlangıç-bitiş noktaları için kırmızılık değeri çok baskın geliyordu. Bu şekilde çok derin bir analiz yapılması gerekiyordu. $g()$ fonksiyonu hesaplanırken bir piksel için maximum 256 birim maliyet çıkarılıyor. Oysa $h()$ fonksiyonu her piksel için 1 birim maliyet çıkarıyor. Kırmızılığın seçim önceliğini korurken mesafe hesabının da gözardı edilmemesi için $g/100$ yapılarak en fazla 2.56 birim maliyet üretilmesi sağlandı. Bu sayede algoritma hala kırmızıları seçerken çok uzak yoldan da hesap yapmak zorunda kalmıyor.

6) Algoritmada iki liste (closed list , open list) oldukça büyük olduğu için noktaların devamlı hangi listede olduğunu aramak programın çalışma hızını düşürüyordu. Bu nedenle resmin her noktasının sığabileceği bir matris oluşturup buradaki noktalara 3 durum verdik. Bu durumlar 0 -> daha önce bu noktaya ulaşılmadı, 1 -> nokta open listte , 2 -> nokta closed listte. Bu sayede noktanın gösterdiği değer 2 ise noktayı geçtik, 1 ise nokta open listte arandı ve yeni cost hesaplandı, 0 ise open liste eklendi.

A*



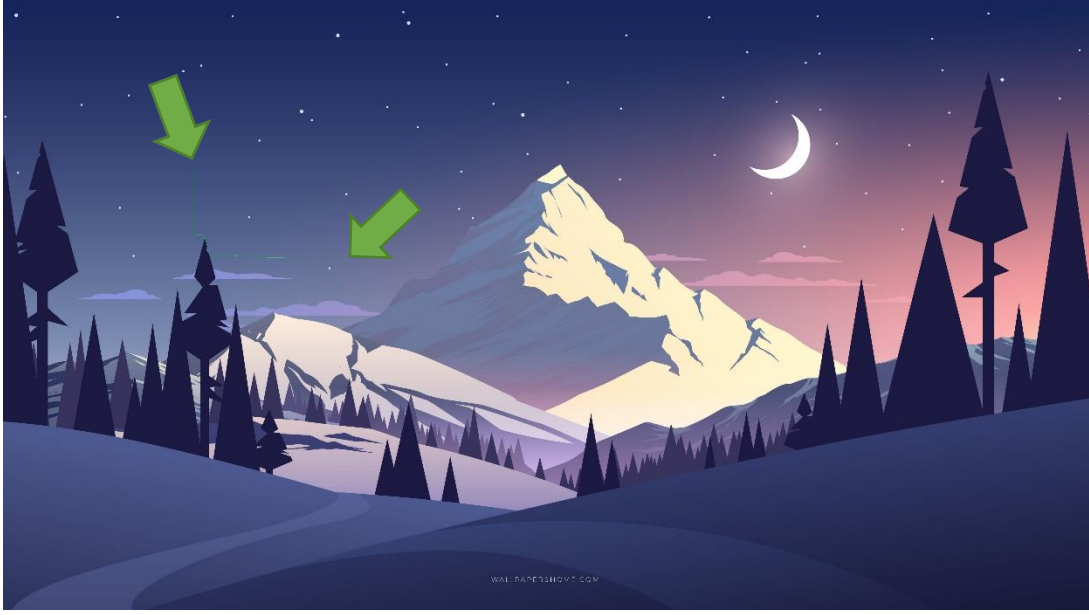
Best-first Search



600,0 - 100,1900	BFS(Stack)	BFS(Heap)	A*(Stack)	A*(Heap)
Süre(s)	0.48	0.07	1490	747

	BFS(Stack)	A*(Stack)
Max Elaman Sayısı	4800	5012
İşlem Sayısı	2401	13240

A*



Best-first Search



800,900 - 1230,1600	BFS(Stack)	BFS(Heap)	A*(Stack)	A*(Heap)
Süre(s)	0.13	0.055	355.8	224.7

	BFS(Stack)	A*(Stack)
Max Elaman Sayısı	2261	2875
İşlem Sayısı	1131	630807

A*



Best-first Search



800,900 - 600,1400	BFS(Stack)	BFS(Heap)	A*(Stack)	A*(Heap)
Süre(s)	0.055	0.03	46.42	24.77

	BFS(Stack)	A*(Stack)
Max Elaman Sayısı	1401	2067
İşlem Sayısı	701	114109

A*



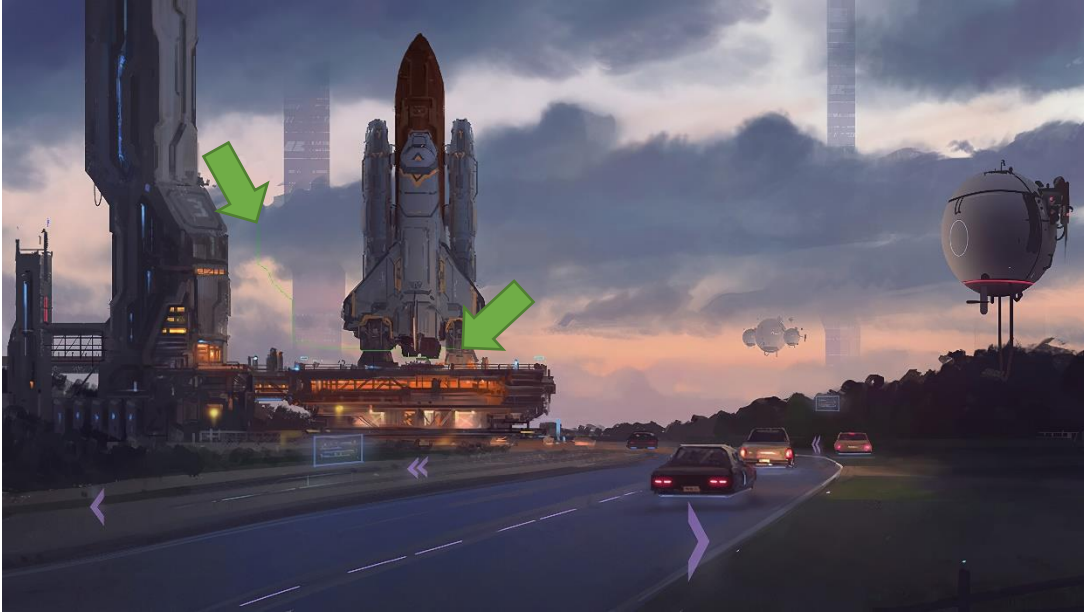
Best-first Search



800,900-600,1400	BFS(Stack)	BFS(Heap)	A*(Stack)	A*(Heap)
Süre(s)	0.055	0.032	38.05	21.23

	BFS(Stack)	A*(Stack)
Max Elaman Sayısı	1401	1540
İşlem Sayısı	701	108649

A*



Best-first Search



800,900 - 1230,1600	BFS(Stack)	BFS(Heap)	A*(Stack)	A*(Heap)
Süre(s)	0.13	0.057	56.69	19.21

	BFS(Stack)	A*(Stack)
Max Elaman Sayısı	2261	2450
İşlem Sayısı	1131	98221

A*



Best-first Search



250,200 -600,700	BFS(Stack)	BFS(Heap)	A*(Stack)	A*(Heap)
Süre(s)	0.088	0.04	103.81	53.98

	BFS(Stack)	A*(Stack)
Max Elaman Sayısı	1701	2609
İşlem Sayısı	851	228780

A*



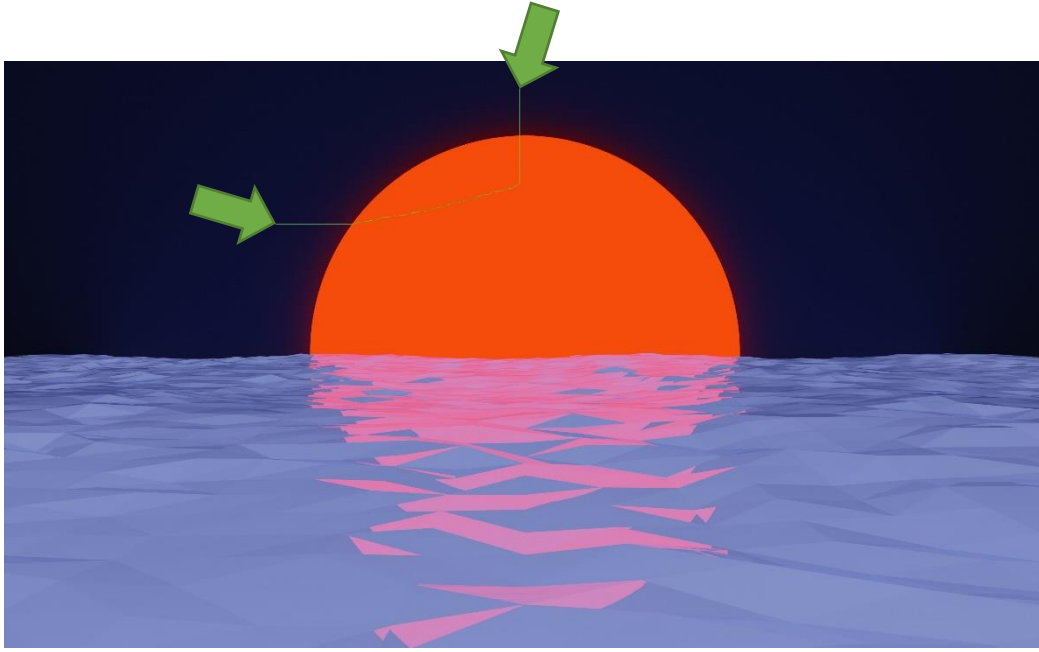
Best-first Search



2000,3800 – 1500,3231	BFS(Stack)	BFS(Heap)	A*(Stack)	A*(Heap)
Süre(s)	0.11	0.05	232.99	153.69

	BFS(Stack)	A*(Stack)
Max Elaman Sayısı	2139	2074
İşlem Sayısı	1070	601092

A*



Best-first Search



600,1000- 100,1900	BFS(Stack)	BFS(Heap)	A*(Stack)	A*(Heap)
Süre(s)	0.217	0.068	109.87	56.46

	BFS(Stack)	A*(Stack)
Max Elaman Sayısı	2801	2968
İşlem Sayısı	1401	142252

A*



Best-first Search



300,400 -900,600	BFS(Stack)	BFS(Heap)	A*(Stack)	A*(Heap)
Süre(s)	0.076	0.039	108	64

	BFS(Stack)	A*(Stack)
Max Elaman Sayısı	1601	2022
İşlem Sayısı	801	311865

A*



Best-first Search



600,0 - 100,1900	BFS(Stack)	BFS(Heap)	A*(Stack)	A*(Heap)
Süre(s)	0.6	0.04	98	42

	BFS(Stack)	A*(Stack)
Max Elaman Sayısı	4800	4838
İşlem Sayısı	2401	85475