

ALTIN TOPLAMA OYUNU PROJESİ

Hüseyin Yılmaz

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

ylmzhsyn98@gmail.com

Özet

Bu projede, $m \times n$ boyutlu bir tahta üzerinde farklı hedef seçme algoritmaları kullanan oyunculara sahip, başlangıç değerlerinin kullanıcı tarafından belirlenebildiği, anlık oyun durumunun bir arayüzle canlandırıldığı, hareket çıktısının her kullanıcı için ayrı ayrı dosyalara kaydedildiği, en sonunda da oyun özetinin gösterildiği, altın toplama yarışına dayanan bir oyun tasarlamamız istenmiştir.

Giriş

Çalışmada amaç, farklı arama algoritmalarının etkinliğini gözlemlemektir. Oyun, sayıları kullanıcı tarafından belirlenen $m \times n$ kareden oluşan bir oyun tahtasında gerçekleşmektedir. Program çalıştırıldığında ekrana ayar seçim ekranı gelmektedir. Oyundaki kare sayısı, oyuncuların başlangıç altını, altınların ve gizli altınların yüzdesi, hamle ve hedef belirleme başına altın, oyun esnasındaki her aksiyon arasında geçecek zaman ve tüm oyuncuların ayrı ayrı hamle ve hedef belirleme bedelleri varsayılan değerler girili şekilde gelmekte ve kullanıcı tarafından değiştirilebilmektedir. Değerler girildikten sonra “Başlat” butonuna tıklayarak oyun başlatılabilmektedir, eğer ki değerler girilirken bir yanlışlık yapılmışsa oyun başlamamakta ve kullanıcıdan bir uyarı yazısıyla yanlış değerleri düzeltmesi istenmektedir.

Oyun başladığında ekrana girilen $m \times n$ değerindeki oyun tahtası, oyuncuların anlık durumu ve altın miktarı ve oyunda gerçekleşen olayların yazılı çıktısı yansıtılmaktadır. Oyuncuların anlık bulundukları kare, her oyuncunun adı olan harfin o kare üzerine yazdırılması şeklinde gösterilmektedir. Altınlar ise kırmızı yazıyla, önceden gizli olup sonradan görünür hale gelen altınlar pembe yazıyla gösterilmektedir. Ayrıca oyuncuların hedefledikleri kareler yine isimlerindeki harfin o kareye küçük puntuyla yazılması şeklinde gösterilmiştir.

Oyundaki her eylem arasına takibi kolaylaştırmak için bir bekleme süresi konmuştur.

Oyuncuların hepsi oyuna tahtanın ayrı bir kenarından başlamakta ve farklı hedef belirleme algoritmaları kullanmaktadır. A Oyuncusu kendisine en yakın olan kareyi hedeflemektedir. Eğer ki tam aynı uzaklıkta birden fazla kare varsa bu kareler arasından daha fazla altın barındırana seçmektedir. B Oyuncusu ise en karlı olan kareyi seçmektedir. Kâr ise o karedeki altın miktarından, kareye gitmek için yapılacak hamle sayısı ile oyuncunun hamle bedelinin çarpımı çıkarılarak hesaplanmaktadır. C Oyuncusu, B Oyuncusu ile aynı hedef seçme yöntemini kullanmaktadır ancak farkı, hedef seçmeden önce kendisine en yakın iki gizli altını görünür hale getirebilmesidir. Bu sayede gizli altınları diğer oyunculardan önce hedefleyebilmekte ve bir avantaj elde etmektedir. D Oyuncusunun asıl amacı, diğer oyuncuların hedefledikleri karelere onlardan önce ulaşarak, onların altınlarını almak ve bu sayede öne geçmektir. Tek tek her oyuncunun hedeflediği kareyi incelemekte, eğer oyuncunun o anda bir hedefi yoksa hedefleme ihtimalleri yüksek olan kare üzerinden işlemini yapmaktadır. Bu hedef karelere kendisinin daha önce ulaşmış ulaşamama duruma bakmakta, daha önce ulaşabildikleri arasından en karlı olanı seçmektedir. Eğer hiçbir oyuncunun hedefine daha önce ulaşamıyorsa da tahtadaki tüm karelerden kendisi için en karlı olanı hedeflemektedir.

Bir oyuncunun bir altını alabilmesi için o altını hedeflemiş olması lazımdır, hedeflediği altına giderken altın olan bir kareden geçse bile o altını alamamaktadır. Oyuncular hedefledikleri altına giderlerken geçtikleri karelerde bulunan gizli altınları da tüm oyuncular için görünür hale getirmekte ama yine o altını hedeflemeden alamamaktadırlar. Eğer hedeflenen altını başka oyuncu alırsa yeni hedef seçilir.

Oyuncular adım başına değil, hamle başına ücret ödemektedirler. Örneğin hamle bedeli 5 altın ise o turda hedefine ulaşmak için 2 adım da atması gerekse, 3 adım da atması gerekse ödeyeceği ücret 5 altın olacaktır. Hamle sırası gelen oyuncuların hamle yapmaya veya hedef seçmeye altını yetmezse oyundan elenmektedir. Tüm oyuncular oyundan elendiğinde veya tahtada görünür bir altın kalmadığında oyun sonlandırılmakta, tüm oyuncuların hareketleri ayrı ayrı kendi adlarında bir dosyaya yazdırılmaktadır. “Bitir” butonuna tıklandığında ise oyun tahtası kaybolmakta ve yerini oyuncuların istatistiklerinin olduğu bir tablo almaktadır. Buradan oyuncuların oyun boyunca harcadığı altınlar, kazandığı altınlar, attığı adım sayısı ve oyun sonundaki altın miktarı incelenebilmektedir.

Yöntem

Projede Java dili tercih edildi. Arayüz için Swing kütüphanesi kullanıldı. Oyun başında başlangıç ekranı için frame'e bir JPanel objesi eklenmektedir. Bu class, JPanel classından kalıtım almaktadır. Seçim işlemleri tamamlandıktan sonra bu panel frame'den kaldırılmak ve yerine yine JPanel classını kalıtım alan GamePanel classının bir objesi yerleştirilmektedir.

Oyunun tüm bilgileri Board classında saklanmaktadır, bu class tüm kareleri oyuncuları tutmakta, oyundaki her türlü veriye erişebilmektedir. Tahtadaki tüm kareler ise Block classının birer objesi olarak tutulmaktadır. Board classı tüm bu kareler için birer tane obje yaratırken, o kareleri “blocks” adındaki bir diziye kaydetmektedir, bu sayede karelerin tüm bilgilerine indisi sayesinde ulaşmakta ve gerektiğinde düzenleme yapabilmektedir. Tüm karelerin satır ve sütun bilgileri Block sınıfında tutulduğu için satır ve sütun bilgileri girilerek dizideki indisleri kolaylıkla bulunabilmektedir.

Oyun tahtasının boyutu sabittir, karelerin boyutlarının tek tek hesaplanmasından kaçınmak için tahtanın bulunduğu panelin layoutu GridLayout olarak ayarlanmıştır, bu layoutta bir satır ve sütun sayısı girilmekte, içindeki diğer elementler o satır ve sütun sayısına göre otomatik olarak boyut almaktadır. Burada satır ve sütun sayısı tahta boyutlarının birer değer üstünde girilmiştir, çünkü

her satır ve sütunun başında numarası yazmaktadır. Bu numaralar da birer kare olarak yaratılmıştır ama yaratılırken blocks dizisine eklenmemiştir, bu sayede kareler üzerinde işlem yaparken o numara karelerinin hesaba katılması gerekmedi.

Bu arayüzdeki kareler de birer panel objesidir. Bu karelerde ise GridBagLayout kullanılmıştır, bu layout tipinin diğerinden farkı, yaratılırken satır ve sütun sayısı girilmemesi, elementler eklendikçe otomatik şekil almasıdır. Başta tüm bu karelerin içleri boştur.

Daha sonra ise oyuncular yaratılmaktadır. Oyuncuları tutan asıl sınıf Player sınıfıdır. Bu sınıfta oyuncuların genel özellikleri, en yakın kareyi hesaplama, en karlı kareyi hesaplama, hamle yapma gibi özellikler bulunmaktadır. Bu sınıftaki hedef seçmeyi sağlayan selectTarget fonksiyonu her oyuncu için farklı olacağından dolayı bu sınıf abstract olarak yazılmıştır. Daha sonra da her oyuncu için bu Player sınıftan kalıtım alan ayrı birer sınıf yazılmıştır. Bu sınıflarda oyuncuların adları belirlenmekte, tahtadaki yerleri belirlenmekte ve hedef seçme fonksiyonu belirlenmektedir. Diğer tüm özellikler Player sınıfında belirlendiği için başka bir işleme gerek kalmamaktadır.

Board sınıfında oyuncuların hepsi yaratıldıktan sonra oyun paneli ekrana yansıtılmaktadır. Oyun panelini tutan GamePanel sınıfında oyunun anlık durumuna göre tahrayı güncelleyen bir updateBlocks fonksiyonu bulunmaktadır. Bu fonksiyon çağırıldığında karelerin panelleri boşaltılmakta, oyuncuların bulunduğu karelere oyuncuların adı, oyuncuların hedefledikleri karelere de küçük puntoyla oyuncuların adı, altın bulunan karelere ise altın miktarı bir JLabel ile eklenmektedir. Oyundaki her değişiklikten sonra bu fonksiyon çağırılarak arayüz güncellenmektedir.

Oyundaki hareketler ise iki ayrı şekilde saklanmaktadır. Birincisi Player sınıfındaki statusTexts listesinde, ikincisi ise Board sınıfındaki statusTexts listesinde. Player sınıfındaki updateStatus fonksiyonu çağırıldığında belirtilen durum güncellemesi bu sınıftaki listeye eklenmekte, daha sonra da Board sınıfındaki updateStatus fonksiyonunu çağırıp oradaki listeye eklemektedir. Board sınıfındaki güncelleme listesi arayüzde adımların takip edilebilmesi için tutulmakta, Player

sınıfındaki liste ise oyun sonunda hareket dökümünün dosyalara ayrı ayrı yazdırılması için kaydedilmektedir.

Oyun esnasında hareketler arasında bekleme süreleri olduğu için arayüzün donmasını engellemek amacıyla oyunun döndüğü fonksiyon ayrı bir thread olarak çağırılmıştır. GameThread sınıfındaki bu fonksiyon, sonsuz bir döngüde ilerlemektedir. Bu döngüde sırasıyla her oyuncu için playRound fonksiyonu çağırılmaktadır. O fonksiyon öncelikle oyuncuların hedefleri olup olmadığını kontrol etmekte, hedefleri yoksa oyuncunun hedef seçme fonksiyonunu çağırılmaktadır. Hedef seçme işleminden önce oyuncunun altınının hedef seçme bedelinden düşük olup olmadığı kontrol ediliyor, eğer düşükse oyuncu inaktif hale getirilip seçim yapılmadan fonksiyondan çıkılıyor. Eğer altın yeterliyse hedef seçme işlemi oyuncuya göre gerçekleştiriliyor. A Oyuncusu için hedef seçilirken tüm karelerin o anda bulunan kareye mesafesi kıyaslanıp, mesafenin en kısa olduğu kare döndürülüyor. B oyuncusu için en karlı kare hesaplanırken ise doğrudan mesafeler alınmak yerine, hamle sayısı uzaklığı bulunuyor. Bir hamlede 3 adım atılabiliyorsa 10 adım uzaktaki karenin uzaklığı 4 hamle olarak bulunmaktadır. Daha sonra bu bulunan değer hamle başına altın bedeliyle çarpılmaktadır. Bu bulunan değer ise o karede bulunan altından çıkarılmakta ve o karenin karı bulunmaktadır. Tek tek her kare için bu işlem yapıldıktan sonra bu kar değeri en yüksek çıkan kare hedef olarak döndürülmektedir. C oyuncusu hedef seçerken ise B oyuncusu ile aynı şeyleri yapmakta ama daha öncesinde bir işlem daha yapmaktadır. Haritadaki gizli altın içeren kareleri tespit edip, bunların içinden kendisine en yakın olanındaki altını görünür yapmaktadır, bu işlemi iki defa tekrarlamaktadır. Ondan sonra da en karlı kareyi döndürmektedir. D oyuncusu ise öncelikle diğer oyuncuların hedeflerine bakmaktadır. Oyuncuların hedeflerini bir hashmap'e eklemektedir, eğer oyuncunun hedefi yoksa, o oyuncunun objesindeki en yakın kareyi bulma ve en karlı kareyi bulma fonksiyonunu çağırarak muhtemel hedeflerini belirlemektedir. Daha sonra bu karelere ulaşmak için gereken hamle sayısını o oyuncunun konumuyla kıyaslamaktadır. Eğer ki bu karelerden birine daha önce ulaşabiliyorsa, içlerinden en karlı olanı almaktadır. Bu işlem sonucunda başka bir

oyuncunun karesine daha önce ulaşamayacağı tespit edilmişse B oyuncusundaki gibi en karlı olan kare seçilmektedir.

Hedef seçme işlemi tamamlandıktan sonra ise oyuncuda hamle yapmak için yeterli altın bulunup bulunmadığına bakılmaktadır. Eğer yeterli altın varsa, bir döngü içinde hamledeki adım sayısı kadar adım atılmaktadır. Örneğin hedef kare şu anki karenin üstünderse bir satır yukarı çıkılmaktadır. Her adımdan sonra da hedefe ulaşıp ulaşılmadığına bakılmaktadır. Hedefe ulaşıldıysa altın alınıp döngü bozulmaktadır. Ayrıca her adımdan sonra o anda bulunan karede gizli altın olup olmadığına bakılmakta ve varsa bu altın görünür hale getirilmektedir. Sonrasında threaddeki fonksiyonda oyuncunun başarıyla hamle yapıp yapamadığı kontrol edilmektedir, tüm oyuncuların o elki hamlesi başarısız olmuşsa döngü bozulup oyun sonlandırılmaktadır.

Sonrasında da her oyuncunun durum güncelleme listesindeki stringler bir döngüyle dosyaya yazdırılmaktadır.

Sonuçlar

Projede belirtilen tüm isterler başarıyla tamamlanmıştır. İstendiği gibi oyundaki varsayılan olan verilen tüm değerler düzenlenebilmektedir. Oyunun takibi bir arayüz aracılığıyla anlık olarak takip edilebilmekte, sonrasında da dosyadan hamle özetleri okunabilmektedir.

İstendiği gibi tüm oyuncular için farklı hedef seçme algoritmaları yazılmış, oyun sonunda bu algoritmaları kullanarak bu oyuncuların topladıkları altın, attıkları altın, harcadıkları altın ve oyun sonunda ellerinde bulunan altın ekrana çıktı olarak yazdırılmıştır.

Ayrıca oyun takibini kolaylaştırmak için aralara konulan bekleme sürelerini de kullanıcının belirlemesine olanak sağlanmış ve anlaşılabilirlik artırılmıştır. Oyun defalarca kez çalıştırılıp farklı durumlarda runtime hatalarına rastlanmadığından emin olunmuştur. Bu aşamada yalnızca nadiren çok hızlı arayüz güncellemelerinde hata alınmış ama oyuna veya arayüze herhangi bir etkisi olmadığı saptanmıştır.

Çıktılar

Baslangic Altini:	200
Hamle Basi Adim:	3
Altin Yuzdesi:	20
Gizli Altin Yuzdesi:	10
Yatay Kare Sayisi:	20
Dikey Kare Sayisi:	20
Bekleme Suresi (ms):	1000

A Hamle Maliyeti:	5
A Hedef Maliyeti:	5
B Hamle Maliyeti:	5
B Hedef Maliyeti:	10

C Hamle Maliyeti:	5
C Hedef Maliyeti:	15
D Hamle Maliyeti:	5
D Hedef Maliyeti:	20

Baslat

Cikis

AltinToplama

Block

Board

GamePanel

GameThread

HomePanel

Main

Player

PlayerA

PlayerB

PlayerC

PlayerD

AltinToplamaOyunu.xml

oyuncuA.txt

oyuncuB.txt

oyuncuC.txt

oyuncuD.txt

Rapor.docx

temel Libraries

atches and Consoles

Main

D oyuncusu (i18, j1) karesindeki 20 altini aldı.
A oyuncusu için en yakındaki kare (i2, j5) karesine ilerledi.
A oyuncusu hedef belirleme için ödeme yaptı, yeni altını 200.
2 adım (1 hamle) uzaktaki, 0 kar sağlayacak (i2, j17) karesi secildi.
A oyuncusu hamle için ödeme yaptı, yeni altını 180.
A oyuncusu (i11, j4) karesinden (i2, j5) karesine ilerledi.
A oyuncusu (i12, j4) karesinden (i2, j5) karesine ilerledi.
A oyuncusu (i12, j5) karesindeki 5 altını aldı.
B oyuncusu için en karlı kare secilecek.
B oyuncusu hedef belirleme için ödeme yaptı, yeni altını 205.
2 adım (1 hamle) uzaktaki, 15 kar sağlayacak (i2, j17) karesi secildi.

	j1	j2	j3	j4	j5	j6	j7	j8	j9	j10	j11	j12	j13	j14	j15	j16	j17	j18	j19	j20
i1									5		15	5								
i2					A	5	20							10			*20			B
i3								15		20		20	15							
i4								5									20			
i5				20							20							10		
i6																	20			
i7							5				10	20								
i8						10			20						15	15	15			
i9												15								
i10							20													
i11		5				20		5		10	15			15				10	10	
i12						5	15		10											
i13								5												5
i14					5															
i15	10	5								5					15					15
i16				20							20		20					10		5
i17						15	15					5	15							
i18	D			20	10			15		5		15	10		20	15				C
i19		10			15		5							15						
i20				10					15											

Oyuncu: A

Altin: 200

Durum: Aktif

Oyuncu: B

Altin: 195

Durum: Aktif

Oyuncu: D

Altin: 195

Durum: Aktif

Oyuncu: C

Altin: 200

Durum: Aktif

Bitir

C oyuncusu hedef belirleme için ödeme yaptı, yeni altını 200.
2 adım (1 hamle) uzaktaki, 15 kar sağlayacak (i18, j20) karesi secildi.
C oyuncusu hamle için ödeme yaptı, yeni altını 180.
C oyuncusu (i20, j20) karesinden (i19, j20) karesine ilerledi.
C oyuncusu (i19, j20) karesinden (i18, j20) karesine ilerledi.
C oyuncusu (i18, j20) karesindeki 20 altını aldı.
D oyuncusu, diğer oyuncuların hedeflerini inceliyor...
A oyuncusunun (i2, j5) karesini sececegi tahmin edildi.
B oyuncusunun (i2, j17) karesini sececegi tahmin edildi.
C oyuncusunun (i15, j20) karesini sececegi tahmin edildi.
(i2, j17) karesine diğer oyunculardan önce ulaşamaz.
(i15, j20) karesine diğer oyunculardan önce ulaşamaz.
(i2, j5) karesine diğer oyunculardan önce ulaşamaz.
Diger oyuncularin hedefleri secilemedigi için en karlı kare secilecek...
D oyuncusu hedef belirleme için ödeme yaptı, yeni altını 200.
2 adım (1 hamle) uzaktaki, 15 kar sağlayacak (i18, j1) karesi secildi.
D oyuncusu hamle için ödeme yaptı, yeni altını 175.
D oyuncusu (i20, j1) karesinden (i19, j1) karesine ilerledi.
D oyuncusu (i19, j1) karesinden (i18, j1) karesine ilerledi.
D oyuncusu (i18, j1) karesindeki 20 altını aldı.
A oyuncusu için en yakındaki kare secilecek.
A oyuncusu hedef belirleme için ödeme yaptı, yeni altını 205.
2 adım (1 hamle) uzaktaki, 0 kar sağlayacak (i2, j5) karesi secildi.
A oyuncusu hamle için ödeme yaptı, yeni altını 195.
A oyuncusu (i1, j4) karesinden (i2, j4) karesine ilerledi.
A oyuncusu (i2, j4) karesinden (i2, j5) karesine ilerledi.
A oyuncusu (i2, j5) karesindeki 5 altını aldı.
B oyuncusu için en karlı kare secilecek.

Oyuncu A	Oyuncu B	Oyuncu C	Oyuncu D
-----	-----	-----	-----
Toplam Adim: 83	Toplam Adim: 111	Toplam Adim: 69	Toplam Adim: 50
Harcanan Altin: 355	Harcanan Altin: 395	Harcanan Altin: 440	Harcanan Altin: 385
Kasadaki Altin: 195	Kasadaki Altin: 25	Kasadaki Altin: 0	Kasadaki Altin: 5
Toplanan Altin: 350	Toplanan Altin: 220	Toplanan Altin: 240	Toplanan Altin: 190

B oyuncusu (i16, j3) karesinden (i17, j3) karesine ilerledi.
A oyuncusu için en yakındaki kare seçilecek.
A oyuncusu hedef belirleme için ödeme yaptı, yeni altını 195.
1 adım (1 hamle) uzaktaki, 5 kar sağlayacak (i17, j20) karesi secildi.
A oyuncusu hamle için ödeme yaptı, yeni altını 185.
A oyuncusu (i18, j20) karesinden (i17, j20) karesine ilerledi.
A oyuncusu (i17, j20) karesindeki 10 altını aldı.
B oyuncusunun hedefi silindi.
B oyuncusu için en karlı kare seçilecek.
B oyuncusunun secebileceği bir kare bulunamadı.
A oyuncusu için en yakındaki kare seçilecek.
A oyuncusunun secebileceği bir kare bulunamadı.
B oyuncusu için en karlı kare seçilecek.
B oyuncusunun secebileceği bir kare bulunamadı.
Oyun bitti...
A oyuncusunun hareket kayıtları dosyaya yazdırıldı.
B oyuncusunun hareket kayıtları dosyaya yazdırıldı.
C oyuncusunun hareket kayıtları dosyaya yazdırıldı.
D oyuncusunun hareket kayıtları dosyaya yazdırıldı.

oyuncuD.txt - Notepad
File Edit Format View Help

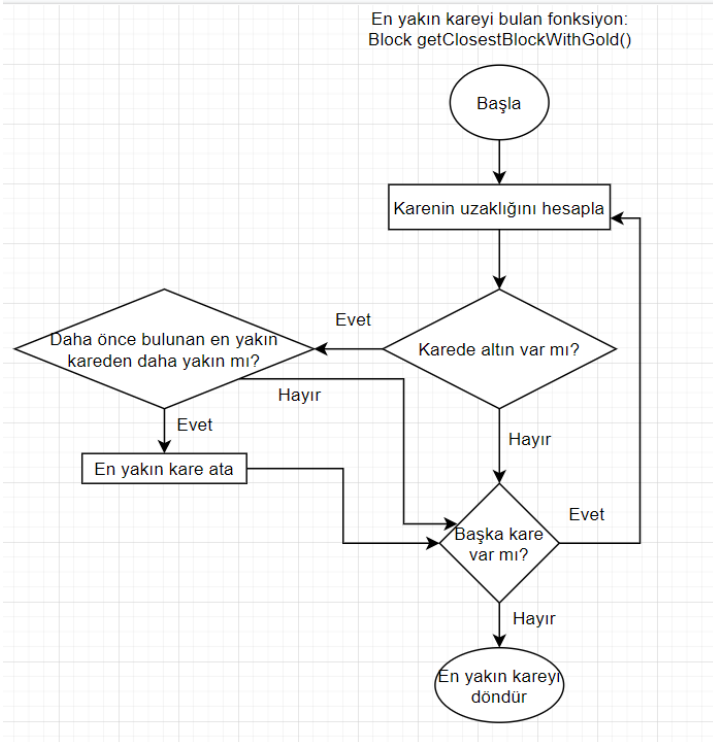
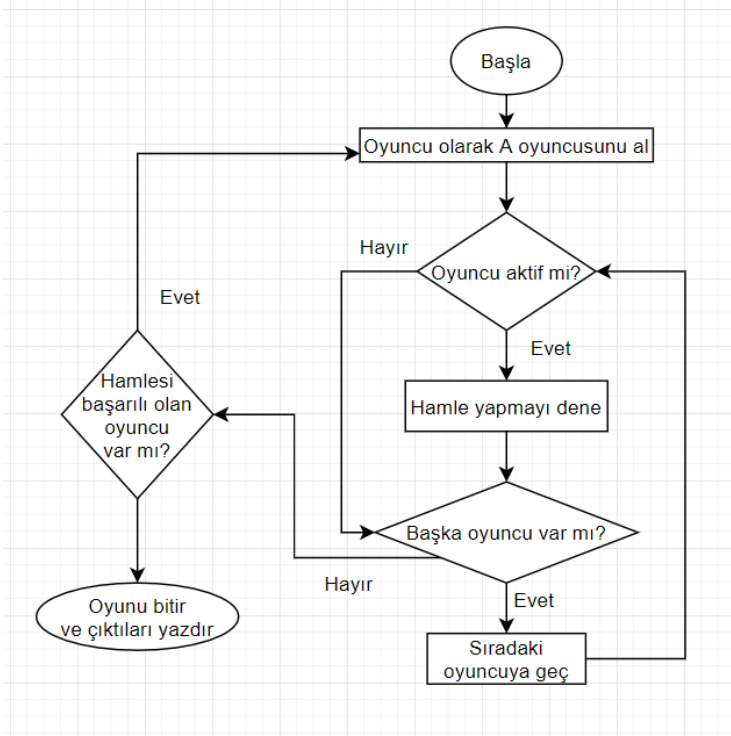
D oyuncusu, diğer oyuncuların hedeflerini inceliyor...
A oyuncusunun (i9, j13) karesini sececeği tahmin edildi.
B oyuncusunun (i3, j10) karesini sececeği tahmin edildi.
C oyuncusunun (i2, j8) karesini sececeği tahmin edildi.
(i2, j8) karesine diğer oyunculardan önce ulaşamaz.
(i3, j10) karesine diğer oyunculardan önce ulaşamaz.
(i9, j13) karesine diğer oyunculardan önce ulaşılabilir.

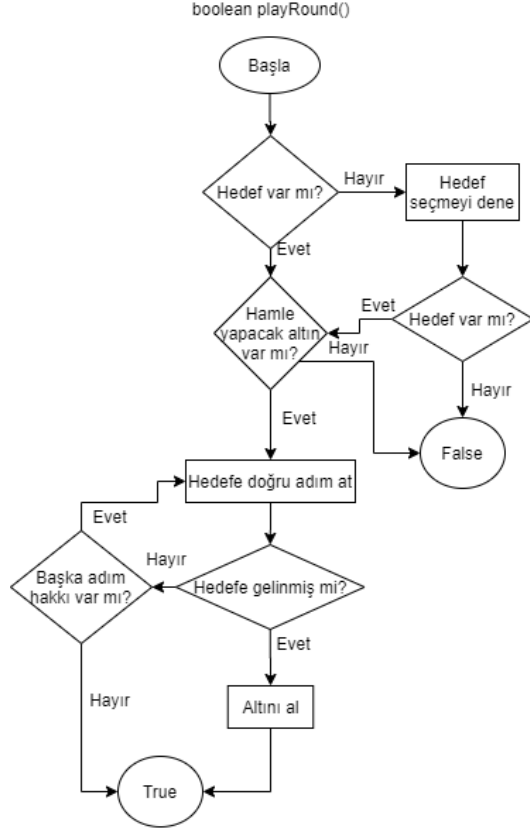
D oyuncusu hedef belirleme için ödeme yaptı, yeni altını 130.
3 adım (1 hamle) uzaktaki, 10 kar sağlayacak (i9, j13) karesi secildi.
D oyuncusu hamle için ödeme yaptı, yeni altını 105.
D oyuncusu (i12, j13) karesinden (i11, j13) karesine ilerledi.
D oyuncusu (i11, j13) karesinden (i10, j13) karesine ilerledi.
D oyuncusu (i10, j13) karesinden (i9, j13) karesine ilerledi.
D oyuncusu (i9, j13) karesindeki 15 altını aldı.

D oyuncusu, diğer oyuncuların hedeflerini inceliyor...
A oyuncusunun (i12, j15) karesini sececeği tahmin edildi.
B oyuncusunun (i6, j13) karesini sececeği tahmin edildi.
C oyuncusunun (i8, j8) karesini sececeği tahmin edildi.
(i8, j8) karesine diğer oyunculardan önce ulaşılabilir.
(i12, j15) karesine diğer oyunculardan önce ulaşamaz.
(i6, j13) karesine diğer oyunculardan önce ulaşılabilir.

D oyuncusu hedef belirleme için ödeme yaptı, yeni altını 120.
3 adım (1 hamle) uzaktaki, 5 kar sağlayacak (i6, j13) karesi secildi.
D oyuncusu hamle için ödeme yaptı, yeni altını 95.
D oyuncusu (i9, j13) karesinden (i8, j13) karesine ilerledi.
D oyuncusu (i8, j13) karesinden (i7, j13) karesine ilerledi.
D oyuncusu (i7, j13) karesinden (i6, j13) karesine ilerledi.
D oyuncusu (i6, j13) karesindeki 10 altını aldı.

Akış Şemaları:





Şema çizimleri draw.io üzerinden yapılmıştır.

Kaynakça:

<https://www.javatpoint.com/java-swing>

<https://docs.oracle.com/javase/7/docs/api/javax/swing/JPanel.html>

<https://docs.oracle.com/javase/tutorial/uiswing/layout/grid.html>

https://www.w3schools.com/java/java_threads.asp

https://www.tutorialspoint.com/java/java_multithreading.htm

<https://www.javatpoint.com/java-jscrollpane>

<https://docs.oracle.com/javase/7/docs/api/java/io/BufferedWriter.html>

<https://www.geeksforgeeks.org/iterating-arrays-java/>