

ALTIN TOPLAMA OYUNU PROJESİ

Feyza DEMİREL – Yasin Ömer KARA

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

190201110 – 180201077

feyzahae@gmail.com y.omerkara1136@gmail.com

ÖZET

Bizden C, C# ya da Java programlama dillerinden biri kullanılarak bir Altın Toplama Oyunu geliştirmemiz istendi. Oyun varsayılan 20x20 kareden oluşacak ve A B C D oyuncularından oynanacaktı. Her oyuncu oyuna tahtanın bir köşesinden başlayacaktı. Tahtada varsayılan sayıda altın olacak ve altınlar tahtaya rastgele dağıtılacaktı. Aynı şekilde tahtada varsayılan sayıda gizli altın olacak ve bu altınlar kullanıcılar tarafından üstünden geçilmediği sürece görülmeyecekti. Oyun tahtadaki altın bitene kadar ya da oyuncuların altını bitene kadar devam edecekti. Oyuncuların her turda varsayılan 3 adım atma hakları vardı. Oyuncular adım atmadan önce hedef belirlemelilerdi. Adım atma ve hedef belirlemenin her oyuncu için varsayılan farklı maliyetleri vardı. Ve her oyuncunun

hedef belirleme algoritması birbirinden farklıydı.

A Oyuncusu: Kendisine en yakın uzaklıktaki altını hedefler.

B Oyuncusu: Altının uzaklığı ve altın miktarını dikkate alarak en karlı olan altını hedefler.

C Oyuncusu: Her hedef belirleme öncesinde kendisine en yakın belli sayıdaki gizlenmiş altınların olduğu kutuları açar. Bu açılan altınlarla birlikte tüm altınlar içerisinde B oyuncusu gibi en karlı olanı hedefler.

D Oyuncusu: Diğer oyuncuların yapacağı hamleleri önceden sezme yeteneği bulunur. Diğer oyuncuların hedeflediği altınlara onlardan önce erişemiyorsa bu altınları hariç tutar ve hedef olarak diğer altın kareler içerisinde en karlı olanı seçer.

C gizli altınları açtığında bu altınlar tüm oyuncular için görülebilir olmalıdır. Varsayılan olarak belirten

tüm değerler oyun başlamadan değiştirilebilir olmalıdır. Oyun dinamik bir yapıya sahip olmalıydı. Oyun sonunda oyuncuların attığı adımların her bir oyuncu için dosyaya yazdırılması istendi. Ve oyun sonunda kasada kalan altın sayısı, her bir oyuncunun toplam altın sayısı, harcanan altın miktarı ve toplam altın miktarının özet bir skor tablosu olarak verilmesi istendi.

1.GİRİŞ

Öncelikle dinamik bir yapı şeklinde kullanıcıdan alınan satır ve sütun değerlerine göre karelerden oluşan bir tahta hazırlandı. Daha sonra tahtanın her bir köşesine A B C D oyuncularını yerleştirdik. Oyuncular yerleştirildikten sonra tahtaya random bir şekilde altınları ve gizli yerleştirdik. Öncelikle her bir oyuncu için bir class oluşturduk ve ilk olarak A oyuncusunun algoritmasını oluşturduk. Kendisine en yakın altını hesaplamasını sağladık. Daha sonra B oyuncusunun algoritmasını oluşturduk. Uzaklık ve altın miktarını dikkate alarak en karlı altını hesaplamasını sağladık. C oyuncusu için ise var sayılan kadar gizli altını açtıktan sonra B oyuncusu gibi en karlı altını hesaplamasını sağladık. Daha sonra D oyuncusunun

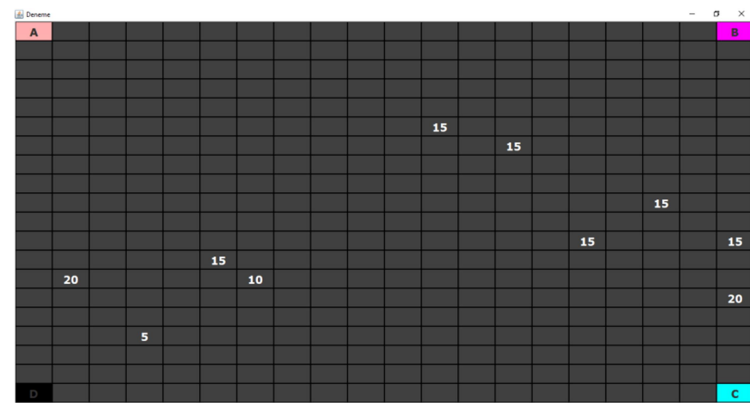
algoritmasını yazdık. Öncelikle diğer oyuncuların hedeflerine kendisi daha önce ulaşabiliyor mu kontrol etti eğer ulaşamıyorsa en karlı altını hedefledi.

Tüm oyuncuların algoritmaları yazıldıktan sonra oyuncuları sırasıyla tahta üzerinde varsayılan adım miktarı kadar hareket ettirdik. Ve oyun tahtadaki altın miktarı bitene kadar ya da oyuncular iflas edene kadar sürdü.

2.YÖNTEM

Java programlama dili kullanılarak Netbeans Ide üzerinde oyun yazıldı. Kullanıcının girdiği değerlere göre dinamik bir arayüz kodlaması oluşturuldu.(Şekil A.1)

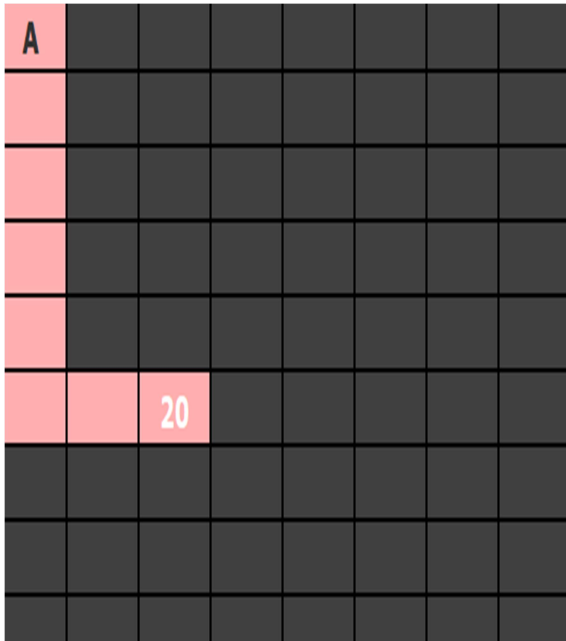
Şekil A.1:



Her oyuncu için belirtilen özellikler göz önüne alınarak her bir oyuncu için farklı algoritmalar kullanıldı.

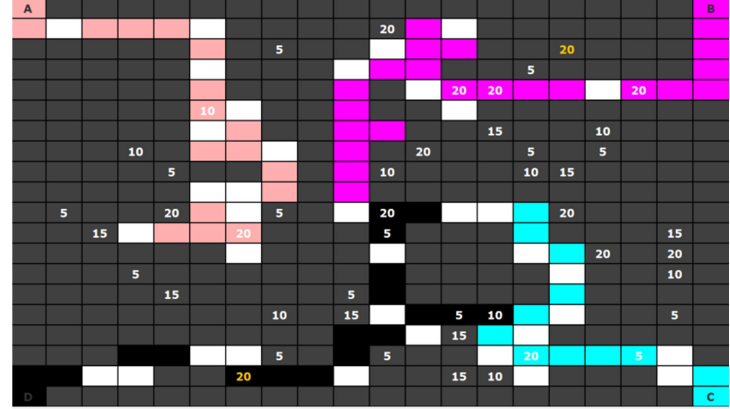
Ör: A oyuncusu için ek yakın mesafe algoritması kullanıldı. Her kullanıcı sınıfı için hedefBelirle() fonksiyonu oluşturuldu ve bu algoritmalar bu fonksiyonun içerisine yazıldı. Daha sonra tahtanın arayüzünü oluşturduğumuz Masa.java sınıfı içerisinde oyuncuların sırasıyla hareket ettiği CizdirmeA(), CizdirmeB(), CizdirmeC() ve CizdirmeD() metotları oluşturuldu. CizdirmeA() fonksiyonun çalışma örneği şekilde verilmiştir.(Şekil A.2)

Şekil A.2:



CizdirmeA(), CizdirmeB(), CizdirmeC() ve CizdirmeD() fonksiyonlarının birlikte çalışma örneği şekilde verilmiştir(Şekil A.3)

Şekil A.3:



(Şekilde verilen beyaz noktalar altınların olduğu noktalardır ve oyuncuların altını aldıklarını gösterir.)

Resimde sarı ile yazılmış noktalar açılmış gizli altınları gösterir.

Her bir adımdan sonra
TimeUnit.SECONDS.sleep(1);
komutunu kullanarak her adımın 1 saniyede bir atılmasını sağladık.

3. DENEYSEL SONUÇLAR

Öncelikle dinamik bir arayüz oluşturmakta zorlandık farklı layout çeşitlerini denedik ama bizim arayüzümüz için en uygun olanının GridLayout olduğuna karar verdik. Daha sonra B'nin algoritmasını

kullanırken zorlandık ama birçok denemenin sonucunda doğru algoritmayı kurduk. Bizim için en zorlayıcı kısımlardan biri hareketti. Hareket ettirirken sağa sola aşağı yukarı kararının oyuncu tarafından verilmesi gerekiyordu ve tüm oyuncular sırasıyla hareket etmeliydi. Daha sonrasında bunun algoritmasını da kurduk.

4.SONUÇ

Bu projede java ile nasıl dinamik bir arayüz oluşturabiliriz ve bir masaüstü uygulaması yapabiliriz onu gördük. Farklı kısıtlamalara sahip algoritmaların birbirlerine karşı etkinliklerini gördük. Arama algoritmalarını bir uygulama içerisinde kullanmayı öğrendik. GridLayout nasıl kullanılır bunu gördük ve bunun içerisine nasıl paneller yerleştireceğimizi gördük. Panellerimizi bir matris şeklinde yerleştirdik ve aslında kendi koordinat sistemimizi oluşturduk. Koordinat sistemine arama algoritmalarını uyumlu hale getirdik ve oyuncularımızı panellerde nasıl hareket ettireceğimizi random şekilde nasıl panellere altın ekleyeceğimizi ve bu altınlara göre nasıl hareket edeceğimizi öğrendik. Kodlama becerimizi geliştirdiğimizi düşünüyoruz.

5. KULLANDIĞIMIZ KÜTÜPHANELER

Projede kullandığımız kütüphanelerden bazıları:

- java.awt.Color
- java.awt.GridLayout;
- javax.swing.JFrame;
- javax.swing.JButton;
- javax.swing.JPanel;
- java.io.BufferedWriter;
- java.io.File;
- java.io.FileWriter;
- java.io.IOException;

6. KAYNAKÇA

[1].Web Site

<https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/javada-dosya-islemleri>

[2].Web Site

<https://www.geeksforgeeks.org/design-a-chess-game/>

[3].Web Site

<https://docs.oracle.com/javase/tutorial/uiswing/components/textarea.html>

[4].Web Site

<https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/javada-dosya-islemleri>

[5].Web Site

<https://docs.oracle.com/javase/tutorial/uiswing/components/formattedtextfield.html#:~:text=Formatted%20text%20fields%20provide%20a,inherited%20from%20the%20JTextField%20class>

[6].Web Site

<https://stackoverflow.com/questions/218384/what-is-a-nullpointerexception-and-how-do-i-fix-it>

UML DİYAGRAMI:

