

ALTIN KAPMACA OYUNU

Alparslan Beraat Özdemir - 170202045
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
Kocaeli, Türkçe
beraat78@gmail.com

ÖZET

Bu bir oyun programıdır. 4 oyuncudan oluşur ve oyuncular kendi kendilerine oynarlar, kullanıcı etkileşimi yer almamaktadır. Program çalıştıktan sonra kullanıcıdan oyunun özelliklerini belirtmesi istenir, belirtilen özelliklere göre oyun tahtası ve oyuncu özellikleri ayarlanarak oyun otomatik olarak oynanmaya başlar. Her oyuncunun kendine has bir hamle stratejisi vardır.

I. GİRİŞ

Altın Kapmaca oyunu, içerisinde 2 farklı frame içermektedir. Oyunu başlatmak için kullanılan ve başlangıç değerlerinin alındığı frame hariç bütün proje tek bir oyun class'ı içerisinde tamamlanmıştır. Temel olarak Şu fonksiyonlardan oluşmaktadır:

Başlatma fonksiyonu, çalıştırma fonksiyonu, frame güncelleme fonksiyonu ve 4 oyuncu için ayrı ayrı hedef belirleme ve hamle yapma fonksiyonları yer almaktadır.

Değer belirlenen pencere, değerleri aldıktan sonra başlat fonksiyonu ile değerleri oyun sınıfına iletir ve başlat fonksiyonu değerleri işleyerek çalıştır fonksiyonu ile oyunu çalıştırır.

Oyun otomatik olarak oynanmaya başlar ve bitene kadar yapılan her işlem cikti.txt adlı bir dosyada kaydolur.

Program timer ile multi-thread hale getirilmiştir.

Harita oluşturmada ve animasyonda butonlar kullanılmıştır.

II. TEMEL BİLGİLER

Proje gelişiminde;

Tümleşik geliştirme ortamı olarak Apache Netbeans IDE 12.0 kullanılmıştır.

Program Windows 10 İşletim sisteminde test edilmiştir. Program Java dilinde geliştirilmiştir.

III. TASARIM

Altın kapmaca oyununun programlanma aşamaları altta belirtilen başlıklar altında açıklanmıştır.

A. Oyun bilgilerini kullanıcıdan almak

Bu aşama, Netbeans IDE üzerinde swing destekli sürükle bırak pencere tasarlayıcısı kullanılarak, kullanıcıların değerleri girebileceği bir pencere oluşturulmak suretiyle tasarlanmıştır. İlgili textboxlara default veriler ön tanımlı olarak gelecek şekilde ayarlanmıştır, Baslangic isimli Class'ta bu textboxlardan alınan verilerin Oyun sınıfına gönderilmesi sağlanır. Oyun sınıfına verilerin gönderilmesi, bu sınıfa ait "baslat" isimli fonksiyon yardımıyla olur. Veriler gönderildikten sonra bu pencere kendi kendini otomatik olarak kapatır.

B. Fonksiyonlar

Baslat fonksiyonu:

Bu fonksiyon, ilk pencerede yer alan bütün girdileri parametre olarak alarak Oyun sınıfına taşınmasını sağlar. Oyun sınıfında public static olarak tanımlanmış değişkenlere bu değerleri yazarak bütün fonksiyonların parametre almaksızın her yerden erişebilmesi için hazır hale getirir.

Calistir fonksiyonu:

Oyunu hazır hale getirip başlatan ana fonksiyondur. Girilen boyutlarda haritanın oluşmasını, altınların ve gizli altınların belirtilen oranda rastgele haritaya dağılmasını, gereken çıktıların yazılacağı dosyanın hazır hale getirilmesini, Oyuncuların köşelere yerleştirilmesini ve oyunun temeli için gereken diğer bütün değişkenlerin başlangıç değerlerinin atamasını sağlar.

Oyunun gösterileceği frame yani pencere, burada tanımlanır ve oluşturulur.

Bütün oluşturmalar bittikten sonra, oyuncuların hamlelerinin sırası ile yapılmasını sağlayan ve oyunun bitene kadar sürekli çalışmasını sağlayan fonksiyon budur.

Oyuncular hamle yaparken aynı Thread içerisinde oyun tahtası da eşzamanlı güncellenemediğinden, Timer ve TimerTask kullanılarak oyuncuların hamlelerini farklı bir thread'da yapması sağlanmıştır. Bu sayede main thread, pencereden sorumlu olabilir ve tahtayı daima anlık olarak güncel tutabilmektedir.

Timer, her oyuncunun sırası ile hamle yapmasını sağlarken, oyuncuların hamlelerinin daha net anlaşılması için, hamlelerin olduğu thread kısa aralıklarla Thread.sleep(300) şeklinde 300 ms gecikme yaratarak bekletilmiştir.

Oyuncu1hamle fonksiyonu:

Bu fonksiyon, oyuncu1'in hedefinin olup olmadığını, hedefi var ise ulaşana kadar hedefe yönelik belirtilen hamle sayısı kadar hamle yapmasını, hedefi yoksa oyuncu1hedef fonksiyonu çağırılarak hamle belirlemesini sağlar Oyuncu1'in davranışlarının tamamından sorumludur.

Oyuncu1hedef fonksiyonu:



Oyuncu1 hedefe sahip değilse çağırılan fonksiyondur. Oyuncu1'in belirtilen özelliklerine ve hedef belirleme stratejisine göre yeni bir altın hedefli belirlemesini sağlar.

NOT: Bütün oyuncular yukarıda oyuncu1 için bahsedilen fonksiyonlara sahiptir. Bu fonksiyonlar her 4 oyuncu için de oyuncu2hedef, oyuncu2hamle, oyuncu3hedef, oyuncu3hamle gibi ayrı ayrı fonksiyonlar halinde programda yer almaktadır. Hepsinin temelde yaptığı görev, oyuncuların hareket etmesini sağlamak olduğu için tek ele alınmayacaktır.

Her hedef fonksiyonu, aşağıdaki kurala göre geliştirilmiştir:

Oyuncu 1 için: En yakın altın.

Oyuncu 2 için: Maliyetler göz önüne alınarak en kârlı altın.

Oyuncu 3 için: Hedef belirlemeden önce 2 gizli altın görünür yapıldıktan sonra maliyetler göz önüne alınarak en kârlı altın.

Oyuncu 4 için: Diğer oyuncuların kendi hedeflediği altına kendisinden önce ulaşip ulaşamayacağı da göz önüne alınarak maliyet hesabı da yapıldıktan sonra en kârlı altın.

Frameguncelle Fonksiyonu:

Bu fonksiyon, her hamleden sonra haritadaki değişiklikleri tarar ve haritayı son konuma göre tekrar düzenler.

C. Algoritma

- 1- Başla
- 2- Kullanıcıdan Verileri Al.

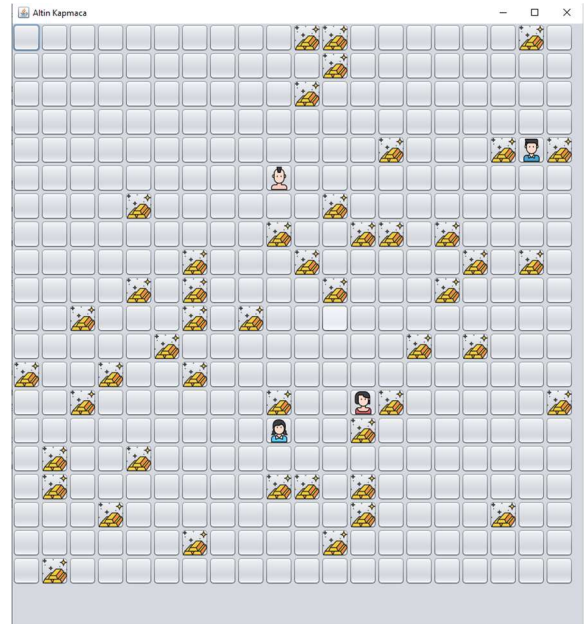
- 3- Oyun Sınıfına gönder ve bu verilerle haritayı oluşturup oyunu kurgula.
- 4- Sürekli çalışan zamanlayıcıyı aktif hale getir.
- 5- Oyuncuların tamamının altınları bitene ya da oyunda altın kalmayana dek her oyuncu hamle fonksiyonunu sıra ile belirlenen hamle sayısı kadar çağır.
- 6- Oyuncu hamle fonksiyonunda oyuncunun hedefi var mı kontrol et.
- 7- Hedefi varsa 1 kare hamle yap
- 8- Eğer oyuncu hedef belirlediyse, Bir kere daha fazladan çağır.
- 9- Eğer oyuncu hedefe ulaştıysa, tur hakkı geçmiş sayıldığı için tekrar çağırma.
- 10- Oyun bitene kadar bütün oyuncuları burada yer alan sıra ile çağır.
- 11- Oyun bitince oyunu sonlandır.

IV. SONUÇLAR VE EKRAN ÇIKTILARI

Oyun ilk açıldığında kullanıcıyı karşılayan, özelliklerin belirlendiği sade arayüz şekil 1'de gösterilmiştir.

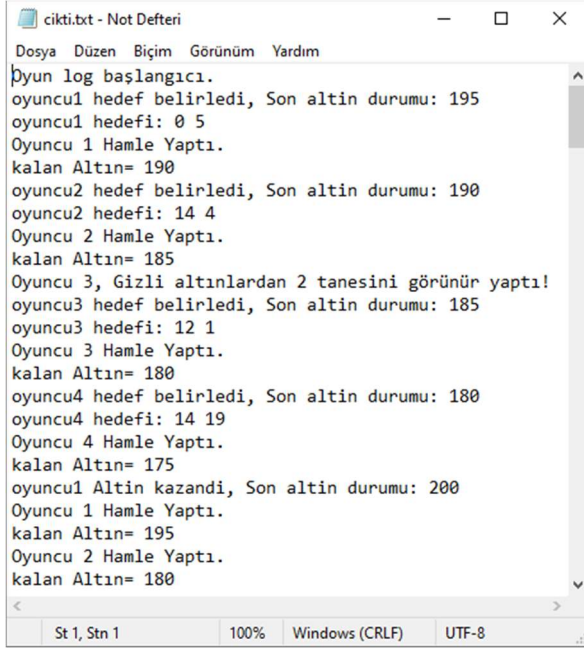
Şekil 1.

Oyunu Başlat'a basıldığı zaman başlayan oyun akarken çekilmiş bir ekran görüntüsü şekil 2'de belirtilmiştir.



Şekil 2.

Oyun çıktısından örnek bir görüntü Şekil 3'te belirtilmiştir.



```
ciikti.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
Oyun log başlangıcı.
oyuncu1 hedef belirledi, Son altın durumu: 195
oyuncu1 hedefi: 0 5
Oyuncu 1 Hamle Yaptı.
kalan Altın= 190
oyuncu2 hedef belirledi, Son altın durumu: 190
oyuncu2 hedefi: 14 4
Oyuncu 2 Hamle Yaptı.
kalan Altın= 185
Oyuncu 3, Gizli altınlardan 2 tanesini görünür yaptı!
oyuncu3 hedef belirledi, Son altın durumu: 185
oyuncu3 hedefi: 12 1
Oyuncu 3 Hamle Yaptı.
kalan Altın= 180
oyuncu4 hedef belirledi, Son altın durumu: 180
oyuncu4 hedefi: 14 19
Oyuncu 4 Hamle Yaptı.
kalan Altın= 175
oyuncu1 Altın kazandı, Son altın durumu: 200
Oyuncu 1 Hamle Yaptı.
kalan Altın= 195
Oyuncu 2 Hamle Yaptı.
kalan Altın= 180
```

şekil 3.

KAYNAKÇA

- [1] <https://youtu.be/844Q40XXXKXc> Cross Toast Gaming youtube kanalı, Java Game Programming – Creating a Tile Grid videosu
- [2] <https://serdarkuzucu.com/javada-timer-ve-timertask-siniflarinin-kullanimi/>
- [3] <https://stackoverflow.com/questions/36779354/java-creating-a-grid-of-buttons-using-a-2d-array>

EKSİKLİKLER

Yazılım, oyunun sonunda bir skor tablosu göstermiyor.