

Altın Toplama Oyunu Projesi

Uğur Muhammed KARAYEL - 170202131

Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi

ugurkarayel41@gmail.com

Özet

Altın Toplama Oyunu; $m \times n$ boyutlu bir dikdörtgen tahta üzerinde farklı özelliklere sahip olan oyuncuların altın toplama yarışına dayanır.

Amaç:

1. Farklı kısıtlara sahip arama algoritmalarının birbirlerine karşı etkinliklerini gözlemlemek,
2. Arama algoritmalarını bir uygulama içerisinde kullanma ve kodlama becerisini geliştirmek,
3. Dinamik özelliklere sahip bir program geliştirmek.

1. Giriş

“Altın Toplama Oyunu Projesi” yazılım programı olarak C# nesne programlama dili tercih edilerek yazılmıştır. Program, girişinde ayarlar,yeni oyun ve çıkış şeklinde menu bulunmaktadır. Ayarlar menüsünde oyundaki değerleri değiştirebiliyoruz. Yeni oyun menüsü ile oyun tahtamız ekrana geliyor. Bu kısımda başla diyerek oyun otomatik bir şekilde oyuncular sırasıyla hamlelerini yaparak oynamaktadır. Oyun tahtasında altın biterse veya oyuncuların altınları biterse oyun biter. Çıkış butonuna basarak karşına sonuçlar ekranı gelmektedir bu kısımda oyuncuların hamleleri,topladıkları,harcadıkları ve kasalarındaki altın miktarını yazmaktadır. Oyun oynama aşamasında oyuncuların her hamlelerinde hedef seçme ve ilerleme olsun bütün hamleleri txt dosyasına kaydedilmektedir.

2. Temel Bilgiler

Programyazılım dili olarak C# Nesne Programlama “OOP”, Derleyeci olarak Visual Studio Kullanılmıştır. K-Nearest Neighbors Algoritması baz alınarak tasarlanmıştır.

3. Yöntem

Programda C# nesne tabanlı programlama dili ve Form arayüz formunun senkronize bir biçimde kullanılabilmesi hedeflenmiştir. Altın Toplama Oyunu Projesi programının programlama aşamaları altta belirtilen başlıklar altında açıklanmıştır.

3.1. Başlangıç Komutları

- 1- “Yeni Oyun”
- 2- “Ayarlar”
- 3-”Çıkış”

3.2. İkincil Komutlar

- 1- Başla
- 2- Çıkış

3.3. Komutların Çalışma Metotları

Oyun başlangıcında ayarlar sekmesinde, oyuncuların kasasındaki altın değerleri, hedef belirleme ücretleri, hamle yapma ücreti,oyun alanı boyutları,oyundaki altın sayısı,gizli altın sayısı gibi birçok değeri değiştirebiliyoruz.

Yeni oyun diyerek oyun arayüzü karşımıza gelmektedir. Arayüz formunda başla butonuna basarak oyunu başlatıyoruz. Oyun otomatik olarak oynamaktadır. Oyun bitimine kadar sürekli ekran yenilenerek güncel bilgiler paylaşılmaktadır. Oyunculardan altını biten oyuncu oyundan elenmektedir.

Oyun bitişi oyun alanında altın kalmadığında veya oyuncuların hepsi elendiğinde oyun bitmektedir. Oyun

bittiğinde çıkış butonuna basarak sonuc formu açılmaktadır. Bu kısımda oyuncuların yapmış olduğu hamle sayısı, topladığı altın, harcadığı altın ve kasalarında kalan altın miktarı yazmaktadır.

3.4. Algoritma

Oyun otomatik olarak oynanmaktadır. Başla düğmesine basıldığında oyun başlar.

-İlk oyuncu olan A oyuncusu ilk olarak en yakınındaki altını hedef olarak belirler ve varsayılan hamle sayısı kadar ilerleyerek altını almaya çalışır. Her hedef belirleme ve hamle yapması için belli bir ücret altın ödemektedir. Her hamle ve hedef belirlemesi ile A.txt dosyasına hamleleri yazılmaktadır.

-İkinci oyuncu B oyuncusu yakınındaki en karlı altını hedef olarak belirliyor ve varsayılan hamle sayısı kadar ilerleyerek altını almaya çalışır. Her hedef belirleme ve hamle yapması için belli bir ücret altın ödemektedir. Her hamle ve hedef belirlemesi ile B.txt dosyasına hamleleri yazılmaktadır.

-Üçüncü Oyuncu C oyuncusu hedef belirlemeden önce gizli altınlardan kendisine yakın olanı açar daha sonra yakınındaki en karlı altını hedef olarak belirler ve varsayılan hamle sayısı kadar ilerleyerek altını almaya çalışır. Her hedef belirleme ve hamle yapması için belli bir ücret altın ödemektedir. Her hamle ve hedef belirlemesi ile C.txt dosyasına hamleleri yazılmaktadır.

-Dördüncü Oyuncu D oyuncusu diğer oyuncuların hedeflerini tespit eder ve diğer oyuncuların hedeflerine daha önce varabiliyorsa o hedefe ilerlemektedir eğer diğer oyunculardan önce ulaşamıyorsa oyuncuların hedefleri dışında yakınındaki en karlı altını hedef olarak belirliyor ve varsayılan hamle sayısı kadar ilerleyerek altını almaya çalışır. Her hedef belirleme ve hamle yapması için belli bir ücret altın ödemektedir. Her hamle ve hedef belirlemesi ile D.txt dosyasına hamleleri yazılmaktadır.

Oyun bitiminde sonuc ekranında hamleleri ekranda gösterilmektedir.

Sözde kod (kaba kod, yalancı kod, pseudocode)

```
while(toplam altın>0)
{
    while
    {
```

A hedef varmı yoksa hedef seç.

```
        A hamle yap
    }
    while
    {
        B hedef varmı yoksa hedef seç.
        B hamle yap
    }
    while
    {
        C hedef varmı yoksa hedef seç.
        C hamle yap
    }
    while
    {
        D hedef varmı yoksa hedef seç.
        D hamle yap
    }
}
if(Oyun altını fazla olan kazanır)
{ MessageBox.show(“kazandı”) }
```

A Hedef seç fonksiyonu
if (kasadaki altın yeterli mi hedef seçme için)
yeterli değilse oyuncu durumu false
if (oyuncu durumu true ?)
int yol_max=max değerde..
for(...)
for(..)
alandaki altınları kontrol eder
yol_maxdan küçükse hedef olarak
seçilir ve yol_max’a eşitlenir. A_harcananaltın’a hedef
ücret eklenir ve kasadan altın azalır.

Oyun alanı x ve y ekseninde 4 bölgeden taranarak
altın hedefi belirlenir.

En düşük yol_max hedef belirlenir ve A oyuncu
hedef_x = i ve hedef_y = j ile hedef seçilir.

A yolmatrisinde hedef 2 olarak yazılır
A.txt dosyasına A yolmatris yazılır
}
A Hedef git fonksiyonu
{
if (kasadaki altın yeterli mi hedefe gitmek için)
yeterli değilse oyuncu durumu false

x = hedef x – konum x
y = hedef y - konum y mutlak değeri alınır
yapılacak hamle sayısı

if(x!= 0)
x ekseninde hareket eder.

```

        if(x == 0)
            if(y != 0)
                y ekseninde hareket eder.
        if(x == 0 && y == 0)
            hedefe varmıştır..
    }
    B Hedef seç fonksiyonu
    if (kasadaki altın yeterli mi hedef seçme için)
        yeterli değilse oyuncu durumu false
    if (oyuncu durumu true ?)
        int yol_max = max değerde..
        int yol_maliyet = max değerde.
        for(....)
            for(..)

                alandaki altınları kontrol eder
                yol_maxdan küçükse ve yol maliyeti
az ise hedef olarak seçilir ve yol_max'a eşitlenir.
B_harcananaltın'a hedef ücret eklenir ve kasadan altın
azalır.
Oyun alanı x ve y ekseninde 4 bölgeden taranarak
altın hedefi belirlenir.

```

En düşük yol_max hedef belirlenir ve B oyuncu hedef_x = i ve hedef_y = j ile hedef seçilir.

```

B yolmatrisinde hedef 2 olarak yazılır
B.txt dosyasına B yolmatris yazılır
}
B Hedef git fonksiyonu
{
    if (kasadaki altın yeterli mi hedefe gitmek için)
        yeterli değilse oyuncu durumu false

        x = hedef x – konum x
        y = hedef y - konum y mutlak değeri alınır
        yapılacak hamle sayısı

        if(x != 0)
            x ekseninde hareket eder.

        if(x == 0)
            if(y != 0)
                y ekseninde hareket eder.
        if(x == 0 && y == 0)
            hedefe varmıştır..
    }
}

```

```

C Hedef seç fonksiyonu
if (kasadaki altın yeterli mi hedef seçme için)
    yeterli değilse oyuncu durumu false
if (oyuncu durumu true ?)
    C gizli altın aç(varsayılan 2);
    int yol_max = max değerde..
    int yol_maliyet = max değerde.
    for(....)
        for(..)

            alandaki altınları kontrol eder
            yol_maxdan küçükse ve yol maliyeti
az ise hedef olarak seçilir ve yol_max'a eşitlenir.
C_harcananaltın'a hedef ücret eklenir ve kasadan altın
azalır.

```

Oyun alanı x ve y ekseninde 4 bölgeden taranarak altın hedefi belirlenir.

En düşük yol_max hedef belirlenir ve C oyuncu hedef_x = i ve hedef_y = j ile hedef seçilir.

```

C yolmatrisinde hedef 2 olarak yazılır
C.txt dosyasına C yolmatris yazılır
}
C Hedef git fonksiyonu
{
    if (kasadaki altın yeterli mi hedefe gitmek için)
        yeterli değilse oyuncu durumu false

        x = hedef x – konum x
        y = hedef y - konum y mutlak değeri alınır
        yapılacak hamle sayısı

        if(x != 0)
            x ekseninde hareket eder.

        if(x == 0)
            if(y != 0)
                y ekseninde hareket eder.
        if(x == 0 && y == 0)
            hedefe varmıştır..
    }
}

```

```

D Hedef seç fonksiyonu
if (kasadaki altın yeterli mi hedef seçme için)
    yeterli değilse oyuncu durumu false
if (oyuncu durumu true ?)
    if (Diğer oyuncuların hdeeflerini kontrol et)
    {
        daha önce varıyorsa hedef seç
    }
    int yol_max=max değerde..
    int yol_maliyet = max değerde.
    for(...)
        for(..)
            alandaki altınları kontrol eder
            yol_maxdan küçükse ve yol maliyeti
            az ise hedef olarak seçilir ve yol_max'a eşitlenir.
D_harcananaltın'a hedef ücret eklenir ve kasadan altın
azalır.

Oyun alanı x ve y ekseninde 4 bölgeden taranarak
altın hedefi belirlenir.

En düşük yol_max hedf belirlenir ve D oyuncu
hedef_x = i ve hedef_y = j ile hedef seçilir.
asa
D yolmatrisinde hedef 2 olarak yazılır
D.txt dosyasına D yolmatris yazılır
}
D Hedef git fonksiyonu
{
    if (kasadaki altın yeterli mi hedefe gitmek için)
        yeterli değilse oyuncu durumu false

    x = hedef x – konum x
    y = hedef y - konum y mutlak değeri alınır
    yapılacak hamle sayısı

    if(x!= 0)
        x ekseninde hareket eder.

    if(x = =0)
        if(y!=0)
            y ekseninde hareket eder.

    if(x==0 && y==0)
        hedefe varmıştır..
}

```

```

A.txt örnek
*****
1000000000
0000000000
0000000000
0000000000
0000000000
0000000000
0200000000
0000000000
0000000000
0000000000
0000000000
*****
1000000000
1000000000
0000000000
0000000000
0000000000
0000000000
0200000000
0000000000
0000000000
0000000000
*****
1000000000
1000000000
1000000000
0000000000
0000000000
0000000000
0200000000
0000000000
0000000000
0000000000
*****
1000000000
1000000000
1000000000
1000000000
0000000000
0000000000
0200000000
0000000000
0000000000
0000000000
*****

```

Arayuz.cs [Tasarım]

MirGame3 Arayuz

Ayarlar

Oyun Ayarları

X Boyut	20	Başlangıç Altın Değeri:	200
Y Boyut	20	A Hedef Ücreti:	5
Altın Oranı %	20	B Hedef Ücreti:	10
Gizli Altın Oran:	10	C Hedef Ücreti:	15
Adım ücreti	5	D Hedef Ücreti:	20
Hamle Sayısı:	3	C gizlialtın açma sayısı	2

Kaydet

Form1

Yeni Oyun

Ayarlar

Çıkış

D_Oyuncu.hedef_varmi = false; //MessageBox.Show("D oyuncunun hedefi alındı.."
D_Oyuncu.D_matris_yol[D_Oyuncu.D_hedef_x, D_Oyuncu.D_hedef_y] = 0; D_yaz();
A_Oyuncu.A_hedef_varmi = false; //MessageBox.Show("A oyuncunun hedefi alındı.."
; B_yaz();
A_Oyuncu.A_hedef_varmi = false; //MessageBox.Show("A oyuncunun hedefi alındı.."
; C_yaz();
A_yaz();
}

