

28.12.2023

Nesneye Yönelik Programlama

Ödev Raporu

Metaverse Evreni ile karakter oluşturma oyunu

Ad: Umut Can

Soyad: YILDIZ

Projenin Genel Özetini

- ✓ **Metaverse** sınıfı, karakterlerin evreni, adı, TC numarası, para miktarı gibi özelliklerini içerir ve karakterlerin su seviyelerini, açlık seviyelerini kontrol eder.
- ✓ Karakterlerin hareketlerini yönetmek için hareket fonksiyonu bulunur.
- ✓ **kiyaslaPara** fonksiyonu, karakterlerin parasını karşılaştırır ve zenginlik durumunu belirtir.
- ✓ Para transferini, yemek yeme ve su içmeyi sağlayan fonksiyonlar da mevcuttur.
- ✓ Ana **main()** fonksiyonu, kullanıcıya karakter oluşturma imkanı sunar ve karakterler arasında para transferi, hareket etme, güç kıyaslaması yapma gibi işlemleri gerçekleştirir.

Algoritması:

1. Değişkenlerin Tanımlanması ve Kullanıcı Girdilerinin Alınması:

Konum yapısı tanımlanır.

Metaverse sınıfı tanımlanır.

Kullanıcıdan karakterlerin sayısı alınır ve karakterlerin özellikleri (evren, ad, tc, para) kullanıcıdan alınır.

Bu özelliklerle birlikte karakterler oluşturulur ve bir dizi içinde saklanır.

2. Ana Döngü:

Kullanıcı menüsü gösterilir.

Kullanıcı bir seçim yapar (para transferi, hareket etme, güç kıyaslaması, çıkış).

Seçime göre ilgili işlem yapılır.

3. Para Transferi İşlemi:

Kullanıcıdan para transferi yapmak istediği karakterlerin ve miktarın bilgileri alınır.

Belirtilen miktarda para transferi yapılır.

4. Hareket İşlemi:

Kullanıcı bir karakterin kimlik numarasını ve hareket miktarını belirtir.

Belirtilen karakter, belirtilen sayıda hareket eder.

Hareket işlemi için yön tuşlarına göre hareket fonksiyonu çağrılır.

5. Güç Kıyaslaması İşlemi:

Kullanıcıdan karşılaştırmak istediği karakterlerin bilgileri alınır.

Karakterlerin parası karşılaştırılır ve hangi karakterin daha fazla paraya sahip olduğu belirtilir.

6. Çıkış:

Kullanıcı çıkış yapmak istediğinde program sonlandırılır.

UML Gösterimi:

Metaverse
-konum: Konum
-evren: String
-ad: String
-tc: Float
-numara: Int
-para: Float
-su: Int
-aclik: Int
+Metaverse_const(String,string,string,float)
+ set_su(int)
+ set_id(int)
+ set_evren(string)
+ set_ad(string)
+ set_tc(float)
+ set_para(float)
+ set_aclik(int)

+ get_id():int
+ get_su():int
+ get_evren():string
+ get_ad():string
+ get_tc():float
+ get_para():float
+ get_aclik():int
+ acikma()
+ susama()
+ hareket(char)
+ kiyaslaPara(karakter1, karakter2)
+ yemek()
+ su_ic()
+ uyu()

```
#include <iostream>
```

```
#include <string>
```

```
#include <cstdlib>
```

```
#include <time.h>
```

```
#include <ctime>
```

```
#include <windows.h>
```

```
#include <conio.h>
```

```
#define YUKSEKLİK 10
```

```
#define GENISLİK 10
```

```
using namespace std;
```

```
// Konum yapısı, x ve y koordinatlarını içerir
```

```
struct Konum {
```

```
    int konum_x = 0; // x koordinatı
```

```
    int konum_y = 0; // y koordinatı
```

```
};
```

```
// Metaverse sınıfı
```

```
class Metaverse {
```

```
private:
```

```
    Konum konum; // Konum bilgisini içeren bir Konum yapısı
```

```
    string evren; // Karakterlerin bulunduğu evrenin adı
```

```
    string ad; // Karakterlerin adı
```

```
    int numara; // Karakter numaraları
```

```
    float para, tc; // Karakterlerin paraları ve TC numaraları
```

```
    int su = 100; // Su seviyeleri (başlangıç değeri 100)
```

```
    int aciklik = 100; // Açlık seviyeleri (başlangıç değeri 100)
```

```
public:
```

```
    // Kurucu fonksiyon (constructor)
```

```
    void Metaverse_const(string evren, string ad, float tc, float para) {
```

```
        this->evren = evren; // Evren adlarını atar
```

```
        this->ad = ad; // Karakter adlarını atar
```

```
        this->tc = tc; // TC numaralarını atar
```

```
        this->para = para; // Para miktarlarını atar
```

```
    }
```

// SET METODLARI: Değer atama metotları

void set_su(int su) { // Su seviyelerini ayarlar

 this->su = su;

}

void set_id(int id) { // Karakter numaralarını ayarlar

 this->numara = id;

}

void set_evren(string evren) { // Evren adlarını ayarlar

 this->evren = evren;

}

void set_ad(string ad) { // Karakter adlarını ayarlar

 this->ad = ad;

}

void set_tc(float tc) { // TC numaralarını ayarlar

 this->tc = tc;

}

void set_para(float para) { // Para miktarlarını ayarlar

 this->para = para;

}

void set_aclik(int aclik) { // Açlık seviyelerini ayarlar

 this->aclik = aclik;

}

// GET METODLARI: Değer döndürme metotları

int get_id() { // Karakter numaralarını döndürür

 return this->numara;

}

int get_su() { // Su seviyelerini döndürür


```

    return this->su;
}
string get_evren() { // Evren adlarını döndürür
    return this->evren;
}
string get_ad() { // Karakter adlarını döndürür
    return this->ad;
}
float get_tc() { // TC numaralarını döndürür
    return this->tc;
}
float get_para() { // Para miktarlarını döndürür
    return this->para;
}
int get_aclik() { // Açlık seviyelerini döndürür
    return this->aclik;
}

// Açlık durumunu kontrol eden fonksiyon
void acikma() {
    if ((this->aclik) > 10) { // Eğer açlık seviyeleri 10'dan büyükse
        this->aclik = (this->aclik) - 10; // Açlık seviyelerini 10 azaltır
    }
    else { // Aksi durumda
        yemek(); // Yemek fonksiyonunu çağırır
    }
}

// Susama durumunu kontrol eden fonksiyon

```

```

void susama() {
    if ((this->su) > 10) { // Eğer su seviyeleri 10'dan büyükse
        this->su = (this->su) - 10; // Su seviyelerini 10 azaltır
    }
    else { // Aksi durumda
        su_ic(); // Su içme fonksiyonunu çağırır
    }
}

// Hareket fonksiyonu, karakterlerin hareketlerini kontrol eder
void hareket(char hareket) {
    switch (hareket) {
        case 'w': { // Eğer hareket 'w' ise (yukarı gitme)
            cout << this->ad << " Bir Birim Yukarı Gitti" << endl; // Ekrana bilgi yazdırır
            this->konum.konum_y = (this->konum.konum_y) - 1; // Konum y koordinatını bir
azaltır
            cout << this->ad << "ın Konumu: (" << this->konum.konum_x << "," << this-
>konum.konum_y << ")" << endl; // Yeni konumu ekrana yazdırır
            acikma(); // Açlık kontrol fonksiyonunu çağırır
            susama(); // Susama kontrol fonksiyonunu çağırır
            break;
        }
        case 's': { // Eğer hareket 's' ise (aşağı gitme)
            cout << this->ad << " Bir Birim Aşağıya Gitti" << endl; // Ekrana bilgi yazdırır
            this->konum.konum_y = (this->konum.konum_y) + 1; // Konum y koordinatını bir
artırır
            cout << this->ad << "ın Konumu: (" << this->konum.konum_x << "," << this-
>konum.konum_y << ")" << endl; // Yeni konumu ekrana yazdırır
            acikma(); // Açlık kontrol fonksiyonunu çağırır
            susama(); // Susama kontrol fonksiyonunu çağırır
            break;
        }
    }
}

```



```

    }

    case 'a': { // Eğer hareket 'a' ise (sola gitme)

        cout << this->ad << " Bir Birim Sola Gitti" << endl; // Ekrana bilgi yazdırır

        this->konum.konum_y = (this->konum.konum_x) - 1; // Konum x koordinatını bir
azaltır

        cout << this->ad << "ın Konumu: (" << this->konum.konum_x << "," << this-
>konum.konum_y << ")" << endl; // Yeni konumu ekrana yazdırır

        acikma(); // Açlık kontrol fonksiyonunu çağırır
        susama(); // Susama kontrol fonksiyonunu çağırır
        break;
    }

    case 'd': { // Eğer hareket 'd' ise (sağa gitme)

        cout << this->ad << " Bir Birim Sağa Gitti" << endl; // Ekrana bilgi yazdırır

        this->konum.konum_y = (this->konum.konum_x) + 1; // Konum x koordinatını bir
artırır

        cout << this->ad << "ın Konumu: (" << this->konum.konum_x << "," << this-
>konum.konum_y << ")" << endl; // Yeni konumu ekrana yazdırır

        acikma(); // Açlık kontrol fonksiyonunu çağırır
        susama(); // Susama kontrol fonksiyonunu çağırır
        break;
    }
}

}

void kiyaslaPara(Metaverse& karakter1, Metaverse& karakter2) {

    float para1 = karakter1.get_para(); // Karakter 1'in parasını al
    float para2 = karakter2.get_para(); // Karakter 2'nin parasını al

    if (para1 > para2) { // Karakter 1 daha fazla para sahibi ise
        cout << karakter1.get_ad() << " daha zengin." << endl;
    } else if (para1 < para2) { // Karakter 2 daha fazla para sahibi ise
        cout << karakter2.get_ad() << " daha zengin." << endl;
    }
}

```

```

    } else { // Eğer ikisi de aynı miktarda paraya sahipse
        cout << "İkisi de aynı miktarda paraya sahip." << endl;
    }
}

// Yemek yeme fonksiyonu
void yemek() {
    this->aclik = 100; // Açlık durumunu sıfırlar
    cout << this->ad << " Yemek Yedi" << endl; // Ekrana karakterin yemek
yediğini yazdırır
}

// Su içme fonksiyonu
void su_ic() {
    this->su = 100; // Susuzluk durumunu sıfırlar
    cout << this->ad << " Su İçti" << endl; // Ekrana karakterin su içtiğini
yazdırır
}

// Para transferi fonksiyonu
void gonder_para(Metaverse& alici, float miktar) {
    if (this->evren != alici.evren) {
        cout << "Bulunduğunuz Evrende Böyle Bir Karakter Bulunmamaktadır"
<< endl; // Evrenlerin eşleşmediğini bildirir
    }
    else {
        if (miktar <= this->para) {
            alici.para = alici.para + miktar; // Alıcıya para ekler
            this->para = this->para - miktar; // Kendi paramızdan düşer
        }
    }
}

```

```

        else {

            cout << "Gönderecek Yeterli Paranız Bulunmamaktadır" << endl; //
Yeterli para olmadığını bildirir

        }

    }

}

};

// Uyku durumunu kontrol eden fonksiyon
void is_afk() {
    char yuz = 1;
    if (_kbhit() == 0) {
        cout << "Karakteriniz şuan uyuyor" << endl << yuz << endl;
        while (_kbhit() == 0) {

        }
        cout << "Karakteriniz uyandı" << endl;
    }
}

// Ana fonksiyon
int main() {
    setlocale(LC_ALL, "Turkish");
    srand(time(0));

    int getch();

    int _kbhit(void);

    char hareket;

    int N, kar_id, kar_har, kar_al, kar_ode;

```

```
string ad, evren;
```

```
float para, tc;
```

```
cout << "Oluşturmak İstedığınız Karakter Sayısını Giriniz : ";
```

```
cin >> N;
```

```
cout << endl;
```

```
Metaverse array[N];
```

```
// Karakterlerin oluşturulması
```

```
for (int i = 0; i <= N - 1; i++) {
```

```
    cout << "Oluşturmak İstedığınız Karakterin Bulunduğu Evreni Giriniz : ";
```

```
    cin >> evren;
```

```
    cout << "Oluşturmak İstedığınız Karakterin Adını Giriniz : ";
```

```
    cin >> ad;
```

```
    cout << "Oluşturmak İstedığınız Karakterin tcsini Giriniz : ";
```

```
    cin >> tc;
```

```
    cout << "Oluşturmak İstedığınız Karakterin Başlangıç Parasını Giriniz : ";
```

```
    cin >> para;
```

```
    array[i].Metaverse_const(evren, ad, tc, para);
```

```
    array[i].set_id(i);
```

```
    cout << endl << endl;
```

```
}
```

```
char devam = 'e';
```

```
while (devam != 'q') {
```

```
    // ... (Kodun geri kalan kısmı)
```

```
    cout << "İşlem Seçiniz: \n";
```

```

cout << "1. Para Transferi\n";

cout << "2. Hareket Etme\n";

cout << "3. Güç Kıyaslanması\n";

cout << "4. Çıkış\n";


int secim;

cout << "Seçiminizi yapınız: ";
cin >> secim;

switch (secim) {
    case 1: {
        cout << "Para Transfer Etmek İstedığınız Karakterin index Numarasını Giriniz: ";

        cin >> kar_al;
        cout << endl;

        cout << "Para Transferini Gerçekleştirecek Karakterin index Numarasını Giriniz: ";

        cin >> kar_ode;
        cout << endl;

        cout << "Transfer Etmek İstedığınız Miktarı Giriniz: ";
        cin >> para;
        cout << endl;

        array[kar_ode].gonder_para(array[kar_al], para);

        cout << array[kar_ode].get_ad() << " Kalan Para: " <<
array[kar_ode].get_para() << endl;

        cout << array[kar_al].get_ad() << " Kalan Para: " << array[kar_al].get_para()
<< endl;
    }
}

```

```

        break;
    }
    case 2: {

        // Hareket işlemleri

        cout << "Hareket Ettirmek İstedığınız Karakterin İndex Numarasını
Giriniz : ";

        cin >> kar_id;
        cout << endl;

        cout << "Karakterinizi Kaç Kere Hareket Ettirmek İstiyorsunuz : ";
        cin >> kar_har;
        cout << endl;

        cout << "Yukarı Gitmek İçin W, Aşağı Gitmek İçin S, Sağa Gitmek
İçin D ve Sola Gitmek İçin A Tuşlayınız (CapsLock Tuşunun Kapalı Olmasına Dikkat
Ediniz)" << endl;

        is_afk();
        for(int i=0;i<=kar_har;i++){

            hareket=getchar();

            array[kar_id].hareket(hareket);

        }

        system("pause");

    }

    case 3: {

        cout << "Güç kıyaslaması yapmak istediğin karakterin indexini gir: ";

        cin >> kar_al;

        cout << endl;

```



```
cout << "Güç kıyaslaması yapmak istediğin karakterin indexini gir: ";
cin >> kar_ode;
cout << endl;
array[kar_al].kiyaslaPara(array[kar_al], array[kar_ode]);

break;
}

case 4: {
    cout << "Programdan çıkılıyor..." << endl;
    devam = 'q';
    break;
}

default: {
    cout << "Geçersiz seçim!" << endl;
    break;
}
}

// Karakterin hareketi
for (int i = 0; i <= kar_har; i++) {
    hareket = getchar();
    array[kar_id].hareket(hareket);
}
```

```
system("pause");

return 0;

}
```

```
Oluşturmak İsteddiğiniz Karakter Sayısını Giriniz : 2
Oluşturmak İsteddiğiniz Karakterin Bulunduğu Evreni Giriniz : dünya
Oluşturmak İsteddiğiniz Karakterin Adını Giriniz : selin
Oluşturmak İsteddiğiniz Karakterin tcsini Giriniz : 123
Oluşturmak İsteddiğiniz Karakterin Başlangıç Parasını Giriniz : 500

Oluşturmak İsteddiğiniz Karakterin Bulunduğu Evreni Giriniz : dünya
Oluşturmak İsteddiğiniz Karakterin Adını Giriniz : umut
Oluşturmak İsteddiğiniz Karakterin tcsini Giriniz : 456
Oluşturmak İsteddiğiniz Karakterin Başlangıç Parasını Giriniz : 1000

İşlem Seçiniz:
1. Para Transferi
2. Hareket Etme
3. Güç Kıyaslanması
4. Çıkış
Seçiminizi yapınız: 1
Para Transfer Etmek İsteddiğiniz Karakterin index Numarasını Giriniz: 0

Para Transferini Gerçekleştirecek Karakterin index Numarasını Giriniz: 1

Transfer Etmek İsteddiğiniz Miktarı Giriniz: 300

umut Kalan Para: 700
selin Kalan Para: 800
İşlem Seçiniz:
1. Para Transferi
2. Hareket Etme
3. Güç Kıyaslanması
4. Çıkış
Seçiminizi yapınız: 3
Güç Kıyaslaması yapmak istediğin karakterin indexini gir: 0

Güç kıyaslaması yapmak istediğin karakterin indexini gir: 1

selin daha zengin.
İşlem Seçiniz:
1. Para Transferi
2. Hareket Etme
3. Güç Kıyaslanması
4. Çıkış
Seçiminizi yapınız: 2
Hareket Ettirmek İsteddiğiniz Karakterin index Numarasını Giriniz : 0

Karakterinizi Kaç Kere Hareket Ettirmek İstiyorsunuz : 3

Yukarı Gitmek İçin W, Aşağı Gitmek İçin S, Sağa Gitmek İçin D ve Sola Gitmek İçin A Tuşlayınız (CapsLock Tuşunun Kapalı Olmasına Dikkat Ediniz)
Karakteriniz şuan uyuyor
0
Karakteriniz uyandı
asd
selin Bir Birim Sola Gitti
selin'in Konumu: (0,-1)
selin Bir Birim Aşağıya Gitti
selin'in Konumu: (0,0)
selin Bir Birim Sağa Gitti
selin'in Konumu: (0,1)

İşlem Seçiniz:
1. Para Transferi
2. Hareket Etme
3. Güç Kıyaslanması
4. Çıkış
Seçiminizi yapınız: 4
Programdan çıkılıyor...
```

2018