



BURSA TEKNİK ÜNİVERSİTESİ

MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
Bilgisayar Mühendisliği Bölümü

ALGORİTMALAR VE PROGRAMLAMA DERSİ PROJE ÖDEVİ RAPORU

MUHAMMED FATİH GÖRAL

22360859032

OKAN AYDINHAN

23360859056

ŞUBE: 2

ARALIK 2024

1. GİRİŞ

Bu ödevi yapan kişiler: MUHAMMED FATİH GÖRAL ve OKAN AYDINHAN
Kod Yazımında iş bölümü:

MUHAMMED FATİH GÖRAL:

MAİN FONKSİYONU

KAMP ALANI FONKSİYONU

ŞİFAHANE FONKSİYONU

HAN FONKSİYONU

OKAN AYDINHAN:

MACERA FONKSİYONU

MENÜYU GOSTER FONKSİYONU

SEVİYE ATLA FONKSİYONU

DURUM GÖSTER FONKSİYONU

OYUNDAN ÇIK FONKSİYONU

RAPOR YAZIMINDA İŞ BOLUMU:

Rapor kısmını yaparken her ikimiz de yazdığımız kodları birbirimize açıklayarak ve takıldığımız kısımları birbirimize anlatarak baştan aşağı birlikte yazdık.

2. TEKNİK DETAYLAR

Canının sahip olduğu temel ve ek ihtiyaçlara bu bölümde detaylı olarak degenilmelidir. Her bir ihtiyaç için birer alt başlık açılıp; canının bu ihtiyaca neden sahip olduğu, bu ihtiyacını hangi yollarla giderdiği detaylı olarak açıklanmalıdır.

BAŞLANGIÇ: SABİTLER VE FONKSİYON PROTOTİPLERİ

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

#define ALTIN 0
#define SEVIYE 1
#define TECRUBE 2
#define CAN 3
#define TOKLUK 4
#define UYKU 5
#define HIJYEN 6
#define SU 7
#define ENERJI 8
#define MORAL 9
#define GUC 10
#define CEVİKLİK 11
#define DAYANIKLILIK 12
#define KARIZMA 13
#define TOPLAYICILIK 14

float ozellikler[15] = {10, 1, 0, 100, 100, 100, 100, 100, 100, 100, 3, 3, 3, 3, 3}; //Karakterlerin özelliklerini bir dizide tuttu

char ad[50];
char calgi[50];

void menuGoster();
void kampAlani(float[]);
void sifahane(float[]);
void han(float[]);
void macera(float[]);
void seviyeAtla(float[]);
void durumGoster(float[]);
void oyundanCik();
```

Şekil 2.0: Sabitler ve fonksiyon prototipleri

Kodda yer alan #define ifadeleri, oyun sırasında kullanılacak sabitlerin isimlerini belirler. Bu sabitler, karakterin sahip olduğu özelliklerin dizideki indekslerini temsil eder:

- ALTIN (0): Karakterin başlangıçta sahip olduğu altın miktarını ifade eder.
- SEVIYE (1) ve TECRUBE (2): Karakterin oyun ilerleyişindeki gelişimini gösterir.
- CAN (3), TOKLUK (4), UYKU (5), HIJYEN (6), SU (7), ENERJI (8): Temel ihtiyaçlar ve sağlık durumu ile ilgili değerlerdir.
- MORAL (9): Karakterin ruh hali veya motivasyonunu temsil eder.

- GUC (10), CEVIKLİK (11), DAYANIKLILIK (12), KARIZMA (13), TOPLAYICILIK (14): Karakterin becerileri ve fiziksel özellikleridir.

ÇOK ÖNEMLİ NOT:

TEMEL NİTELİKLERE HİYYEN, SU, ENERJİ DEĞİŞKENLERİNİ BİZ EKLEDİK.

Karakterin ihtiyaçlarını daha kapsamlı bir şekilde yansıtmak ve oyunu daha gerçekçi hale getirmek için **hijyen**, **su**, **enerji** ve **moral** niteliklerini ekledik. Bu nitelikler, karakterin hayatı kalması ve oyun boyunca başarılı bir şekilde ilerlemesi için kritik öneme sahiptir.

1. Hijyen

- Karakterin temizliği
- Başlangıç Değeri: 100 (maksimum).
- Kullanım Alanları:
 - Şifahanede: Hijyen seviyesini artırmak için karakter nehirde yıkabilir.
 - Han: Hijyen seviyesi 20'nin altına düşerse, handa şarkı söyleyerek para kazanılamaz.
 - Düşük hijyen, karakterin moralini etkiler ve handa para kazanma fırsatını kısıtlar.
 - Hijyen seviyesinin düşmesi ile kullanıcıya uyarı mesajı verilir.

2. Su

- Karakterin susuzluğununu giderme durumu.
- Başlangıç Değeri: 100 (maksimum).
- Kullanım Alanları:
 - Kamp Alanı: Çalgı çalınca ve çadırda uyuyunca su seviyesi düşer. Nehirde uyuyunca da su seviyesi artmış olur.
 - Han: Yiyecek yiince su seviyesi azalır. İçecek içersen de su seviyen artmış olur.

- Su seviyesi kritik bir değer olan 20'nin altına düştüğünde kullanıcı uyarılır.
- Sıfıra düştüğünde karakterin canından 10 birim eksilir.

3. Enerji

- Karakterin fiziksel aktivitelere katılma gücünü temsil eder.
- Başlangıç Değeri: 100 (maksimum).
- Kullanım Alanları:
 - Seçenekler:
 - Kamp alanında şarkı söyleyip çalgı çalınca enerji azalır. Kamp alanında uyuyarak enerji seviyesi artırılabilir.
 - Han: Handa yemek yiince, enerji içeceği içince ve milkshake içince enerji seviyesi artar.
 - Macera: Maceraya çıkışınca şifalı bitki bulurken enerji seviyesi artar.
 - Enerji sıfıra düşerse karakterin canı her tur 25 azalır.
 - Enerji düşükken (20'nin altında) verilen hasarlar daha az etkili olabilir.

Moral

- Karakterin ruh hali ve motivasyonunu temsil eder.
- Başlangıç Değeri: 100 (maksimum).
- Kullanım Alanları:
 - Kamp Alanı: Moral seviyesini artırmak için kamp alanında şarkı söyleyebilir veya çadırda uyabilir.
 - Şifahane: Tedavi olununca moral seviyesi artar.
 - Han: Yemek yiince ve içecek içince moral seviyesi artabilir.
 - Macera: Yakın çevreden şifalı bitki bulup avlanınca moral seviyesi artabilir.
 - Düşük moral, karakterin performansını olumsuz etkiler. Örneğin, düşük moral ile yapılan savaşlarda hasar azaltılabilir veya başarı şansı düşebilir.
 - Moral seviyesi 20'nin altına düştüğünde oyuncuya uyarı mesajı verilir.

Float özellikler dizisi:

Bu dizi, karakterin tüm temel niteliklerini ve becerilerini depolar:

- İlk 3 eleman: Altın, seviye ve tecrübe.
- 4-9 arası elemanlar: Temel nitelikler (örneğin tokluk, uyku, hijyen).
- 10-14 arası elemanlar: Beceriler (örneğin güç, çeviklik, dayanıklılık).

Başlangıç değerleri, karakterin oyuna girişte sahip olduğu standart seviyeleri belirler:

- Örneğin, can, tokluk ve diğer temel nitelikler 100'den başlar.
- Beceriler varsayılan olarak 3 değerine sahiptir.

Ad ve çalgı değişkenleri ile kodumuzdaki ozanın adı ve çalgı adı tutulur

Kod, modüler bir yapı sağlamak için farklı işlemleri gerçekleştiren fonksiyonları önceden tanımlamıştır:

- **menuGoster**: Oyuncuya seçim yapabileceği ana menüyü gösterir.
- **kampAlani, sifahane, han, macera**: Oyunun farklı mekanlarında yapılacak işlemleri temsil eder.
- **seviyeAtla**: Karakterin tecrübe puanlarına bağlı olarak seviyesini yükseltir.
- **durumGoster**: Karakterin mevcut durumunu (özelliklerini) ekrana yazdırır.
- **oyundanCik**: Oyunu sonlandıran işlemleri içerir.

Bu bölüm, oyunun ana akışını ve kullanıcı etkileşimlerini düzenleyen kısımdır. Oyunun hikayesi, menü seçenekleri ve karakterin temel niteliklerinin kontrolü bu fonksiyon üzerinden gerçekleştirilir.

2.0 MAIN FONKSİYONU

```
//oyun tanitimi
printf("***** OZANIN MACERASI *****\n\n");
printf("Ozan, köy meydanında sessizce oturmuş, haval dunyasında geziniyordu.\n");
printf("Köyde herkes onu paseli sarkilarıyla tanııyordu, ama ozanın aklında daha büyük bir haval vardı:\nGerçek bir maceraya atılarak kendine köy halkına ispat etmek .\n\n");

printf("Bir sabah, köyün muhtarı ona yaklastı ve dedi ki:\n");
printf("\'Ozan, köyümüzün cevresindeki ormanlarda gariplik seyler oluyor.\n");
printf("Cesur birinin gidip bunu anastırması gerek.\n");
printf("Senin sarkılarınız ve cesaretiniz bize yardım edebilir.\'\n\n");

printf("Ozan bu atester gomlegi giydi,gazunu kararttı ve maceraya atılmak için köy meydanından ayrılmaya karar verdi.\n");
printf("Onunde keşfedilecek ormanlar atlatılacak tehlikeler ve köye dönüş yolunda anlatılacak yeni hikayeler vardı.\n");
printf("Simdi, onun macerası baslıyordu.\n");
printf("Havdi, sığ da bu eğlenceli ortak olmak için verilen talimatları uwarak oyun başlayın.\n\n");
printf("*****\n\n");

srand(time(NULL)); // Rastgele sayı üretimi

printf("Ozanın adını girin: ");
gets(ad);
printf("Ozanın kullandığı çalgıının adı: ");
gets(calgi);
```

Şekil 2.1: Hikâye tanıtımı

1. Hikâye Başlangıcı

Kod, oyuncunun temasını tanıtan ve hikâye giriş yapan metinlerle başlar:

- Ozan karakterinin köydeki durumu ve maceraya atılma motivasyonu, kullanıcıya hikâye havasında aktarılmıştır.
- Oyuncuya oyuna motive eden anlatımlarla, oyuncu dünyasına giriş sağlanır.

2. Kullanıcıdan Bilgi Alımı

- Kullanıcıdan ozanın adı (ad) ve kullandığı çalgının adı (calgi) gets fonksiyonu ile alınır.
- Bu bilgiler, hikâyede ve oyuncu akışında kullanılmak üzere saklanır.

```

int secim;
while (1) { //sensus dongu
    menuGoster();
    printf("Bir seçenek secin: ");
    scanf("%d", &secim);

    switch (secim) {
        case 1:
            printf("Kamp alanina gidiyorsunuz...\n");
            kampAlani(ozellikler);
            break;
        case 2:
            printf("Sifahaneye gidiyorsunuz...\n");
            sifahane(ozellikler);
            break;
        case 3:
            printf("Hana gidiyorsunuz...\n");
            han(ozellikler);
            break;
        case 4:
            printf("Macexa alanina gidiyorsunuz...\n");
            macera(ozellikler);
            break;
        case 5:
            printf("Sevive atlava gidiyorsunuz...\n");
            seviveAtla(ozellikler);
            break;
        case 6:
            printf("Durumlar gosteriliyor...\n");
            durumGoster(ozellikler);
            break;
        case 7:
            printf("Oyundan cikmak istiyorsunuz...\n");
            oyundanCik();
            break;
        default:
            printf("Gecersiz secim! Lutfen tekrar deneyin.\n");
    }
}

```

Şekil 2.2: Seçenek kısmı

```

void menuGoster() {
    printf("\n***** ANA MENU *****\n");
    printf("1. Kamp alanina git\n");
    printf("2. Sifahane git\n");
    printf("3. Hana git\n");
    printf("4. Macexaya atl\n");
    printf("5. Sevive atl\n");
    printf("6. Durumu goster\n");
    printf("7. Oyundan cik\n");
}

```

Şekil 2.3: Menu Göster fonksiyonu

```
***** OZANIN MACERASI *****

Ozan, koy meydaninda sessizce oturmus, hayal dunyasinda geziniyordu.
Koyde herkes onu neseli sarkilariyla taniyordu, ama ozanin aklinda daha buyuk bir hayal vardi:
Gercek bir maceraya atilarak kendine koy halkina ispat etmek .

Bir sabah, koyun muhtari ona yaklasti ve dedi ki:
"Ozan, koyumuzun cevresindeki ormanlarda garip seyler oluyor.
Cesur birinin gidip bunu arastirmasi gerek.
Senin sarkilarin ve cesaretin bize yardım edebilir.".

Ozan, bu atesten gomlegi giydi, gozunu karartti ve maceraya atilmak icin koy meydanindan ayrılmaya karar verdi.
Onunde kesfedilecek ormanlar, atlatilacak tehlikeler ve koya donus yolunda anlatilacak yeni hikayeler vardi.
Simdi, onun macerasi basliyordu.
Haydi, siz de bueglenceye ortak olmak icin verilen talimatlara uyarak oyun baslayin.

***** ANA MENU *****
1. Kamp alanina git
2. Sifahane git
3. Hana git
4. Maceraya atil
5. Seviye atla
6. Durumu goster
7. Oyundan cik
Bir secenek secin: 1
Kamp alanina gidiyorsunuz...
```

Şekil 2.4: Oyuncunun doğru seçim yapması

```
***** ANA MENU *****
1. Kamp alanina git
2. Sifahane git
3. Hana git
4. Maceraya atil
5. Seviye atla
6. Durumu goster
7. Oyundan cik
Bir secenek secin: Gecersiz secim! Lutfen tekrar deneyin.

***** ANA MENU *****
1. Kamp alanina git
2. Sifahane git
3. Hana git
4. Maceraya atil
5. Seviye atla
6. Durumu goster
7. Oyundan cik
Bir secenek secin: |
```

Şekil 2.5: Geçersiz seçim

3. Sonsuz Döngü ile Oyun Akışı

- Bir while (1) döngüsü, kullanıcı oyundan çıkışa kadar sürekli çalışır.

- Döngü içinde:
 - **Menü Gösterimi:** menuGoster fonksiyonu çağrılarak, oyuncunun seçim yapabileceği seçenekler listelenir.
 - **Kullanıcı Seçimi:** scanf ile kullanıcından menü seçimi alınır ve switch-case yapısı ile ilgili eylemler gerçekleştirilir:
 - **1:** Kamp alanına git (kampAlani).
 - **2:** Şifahaneye git (sifahane).
 - **3:** Hana git (han).
 - **4:** Maceraya atıl (macera).
 - **5:** Seviye atla (seviyeAtla).
 - **6:** Karakter durumu göster (durumGoster).
 - **7:** Oyundan çıkış (oyundanCik).
 - Varsayılan durumda, geçersiz girişler için uyarı mesajı görüntülenir.

Tekrar Oyuncunun önüne menü kısmı gelir ve tekrar seçim yapar

```

}
//Kritik niteliklerin 20'ye altına düşmesi
if (ozellikler[TOKLUK] <= 20)
    printf("Toksik seviyesi %.2f ye kadar dosta dikkat !!! \n",ozellikler[TOKLUK]);
if (ozellikler[UYKU] <= 20)
    printf("Uykuya seviyesi %.2f ye kadar dosta dikkat !!! \n",ozellikler[UYKU]);
if (ozellikler[HIJYEN] <= 20)
    printf("Hıjyen seviyesi %.2f ye kadar dosta dikkat !!! \n",ozellikler[HIJYEN]);
if (ozellikler[SU] <= 20)
    printf("Su seviyesi %.2f ye kadar dosta dikkat !!! \n",ozellikler[SU]);
if (ozellikler[ENERJI] <= 20)
    printf("Enerji seviyesi %.2f ye kadar dosta dikkat !!! \n",ozellikler[ENERJI]);
// Kritik niteliklerin kritik edilmesi
if (ozellikler[TOKLUK] <= 0 || ozellikler[UYKU] <= 0 || ozellikler[HIJYEN] <= 0 || ozellikler[SU] <= 0 || ozellikler[ENERJI] <= 0) {
    int canKaybi = 0;

    if (ozellikler[TOKLUK] <= 0)
        canKaybi += 20;
    if (ozellikler[UYKU] <= 0)
        canKaybi += 10;
    if (ozellikler[HIJYEN] <= 0)
        canKaybi += 15;
    if (ozellikler[SU] <= 0)
        canKaybi += 10;
    if (ozellikler[ENERJI] <= 0)
        canKaybi += 25;

    ozellikler[CAN] -= canKaybi;
    printf("Nitelikler kritik seviyede! Canınız %d azaldı.\n", canKaybi);

    if (ozellikler[CAN] <= 0) {
        printf("Canınız tükenmiş. Oyun sona erdi!\n");
        return 0;
    }
}
}

```

Şekil 2.6: Kritik nitelikler

```

***** KAMP ALANI *****
1. Kamp atesinin basinda calgi cal/sarki söyle
2. Nehirde yikan
3. Cadirina girip uyu
4. Koy meydanina don
Bir secenek secin: 1
Calginizi calmaya basladiniz...
Karakterin Temel Niteliklerinin Guncel Durumu -> Can:40.00 ,Tokluk: 100.00, Uyku: 0.00, Hijyen: 55.00, Su: 55.00, Enerji: 25.00
Tokluk seviyesi 0.00 ye kadar dustu dikkat !!!
Uyku seviyesi 0.00 ye kadar dustu dikkat !!!
Nitelikler kritik seviyede! Caniniz 30 azaldi.

```

Şekil 2.7: Kritik nitelikler

4. Kritik Nitelik Kontrolleri

- Karakterin temel nitelikleri (örneğin: tokluk, uyku, hijyen, su, enerji) sürekli kontrol edilir:
 - Bir niteliğin seviyesi 20'nin altına düştüğünde, kullanıcı uyarılır.

5.Koşul ile kontrol

- Niteliğin seviyesi 0 olduğunda, karakterin canı azalır. Bu azalma miktarı her nitelik için farklı tanımlanmıştır (örneğin, tokluk = -20 can).
- Karakterin can değeri 0'a ulaşırsa, oyun sona erdirilir.

6. Oyun Sona Erdirme

- Oyuncu çıkış yapmak isterse veya karakterin canı sıfırlanırsa, oyun uygun mesajlarla sonlandırılır.

ÖNEMLİ NOT

```

if(ozellikler[TOKLUK]>100)
    ozellikler[TOKLUK]=100;
if(ozellikler[TOKLUK]<0)
    ozellikler[TOKLUK]=0;
if(ozellikler[UYKU]>100)
    ozellikler[UYKU] = 100;
if(ozellikler[UYKU]<0)
    ozellikler[UYKU] = 0;
if(ozellikler[SU] > 100)
    ozellikler[SU] = 100;
if(ozellikler[SU] < 0)
    ozellikler[SU] = 0;
if(ozellikler[HİJYEN] > 100)
    ozellikler[HİJYEN] = 100;
if(ozellikler[HİJYEN] < 0)
    ozellikler[HİJYEN] = 0;
if(ozellikler[ENERJİ] > 100)
    ozellikler[ENERJİ] = 100;
if(ozellikler[ENERJİ] < 0)
    ozellikler[ENERJİ] = 0;
if(ozellikler[CAN] > 100)
    ozellikler[CAN] = 100;
if(ozellikler[CAN] < 0)
    ozellikler[CAN] = 0;

```

Şekil 2.7: Fonksiyonlardaki kritik nitelikler

```
if(ozellikler[MORAL] > 25)
    ozellikler[MORAL]= 25;
if(ozellikler[MORAL] < 3)
    ozellikler[MORAL]= 3;
if(ozellikler[GUC] > 25)
    ozellikler[GUC]= 25;
if(ozellikler[GUC] < 3)
    ozellikler[GUC]= 3;

if(ozellikler[CEVIKLIK] > 25)
    ozellikler[CEVIKLIK]= 25;
if(ozellikler[CEVIKLIK] < 3)
    ozellikler[CEVIKLIK]= 3;

if(ozellikler[DAYANIKLILIK] > 25)
    ozellikler[DAYANIKLILIK]= 25;
if(ozellikler[DAYANIKLILIK] < 3)
    ozellikler[DAYANIKLILIK]= 3;

if(ozellikler[KARIZMA] > 25)
    ozellikler[KARIZMA]= 25;
if(ozellikler[KARIZMA] < 3)
    ozellikler[KARIZMA]= 3;

if(ozellikler[TOPLAYICILIK] > 25)
    ozellikler[TOPLAYICILIK]= 25;
if(ozellikler[TOPLAYICILIK] < 3)
    ozellikler[TOPLAYICILIK]= 3;
```

Şekil 2.8: Fonksiyonlardaki temel beceriler

```
printf("Karakterin Temel Niteliklerinin Güncel Durumu -> Can: %f , Testlik: %f , Hizli: %f , Hizver: %f , Su: %f , Enerji: %f \n", ozellikler[CAN], ozellikler[HIZYEN], ozellikler[UZNU], ozellikler[SU], ozellikler[ENERJI], ozellikler[MORAL]);
```

Şekil 2.9Fonksiyonlarda kritik niteliklerin yazdırılması

Kamp alanı, Şifahane, Macera, Han fonksiyonlarında temel nitelikler 100 değerini aşmayacak 0 değerinin altına düşmeyecek şekilde if else yapılarıyla kontrol ediliyor. Beceriler de aynı fonksiyonlarda 25 değerini aşmayacak ve 3 değerinin altına düşmeyecek şekilde if else yapısıyla kontrol ediliyor. Aynı zamanda bu fonksiyonların sonunda temel nitelikler ekrana yazılır.

BU KISIM 4 FONKSİYONDA AYRI AYRI AÇIKLAMAMAK İÇİN BURDA TEK SEFERDE AÇIKLADIK.

2.1 KAMP ALANI

```
void kampAlani(float ozellikler[]) {
    int secim;
    do{
        printf("\n***** KAMP ALANI ****\n");
        printf("1. Kamp atesinin basinda cal/sarki söyle\n");
        printf("2. Nehirde vikan\n");
        printf("3. Cadirina gizip uyu\n");
        printf("4. Kov meydanina don\n");

        printf("Bir seçenek secin: ");
        scanf("%d", &secim);

        switch (secim) {
            case 1:
                printf("Calginizi calmaya basladiniz...\n");
                ozellikler[MORAL] += 10;
                ozellikler[ENERJI] -= 5;
                ozellikler[SU] -=5;
                ozellikler[UYKU] -=15;
                ozellikler[TOKLUK] -=15;
                break;
            case 2:
                printf("Nehirde vikanivorsunuz...\n");
                ozellikler[HIZYEN] += 20;
                ozellikler[SU] += 15;
                ozellikler[UYKU] -=10;
                ozellikler[TOKLUK] -=10;
                break;
            case 3:
                printf("Cadirinizda uyuyorsunuz...\n");
                ozellikler[UYKU] += 30;
                ozellikler[ENERJI] += 20;
                ozellikler[MORAL] += 20;
                ozellikler[DAYANIKLILIK] += 10;
                ozellikler[GUC] += 10;
                ozellikler[TOKLUK] -=20;
                ozellikler[SU] -=10;
                break;
            case 4:
                printf("Kov meydanina geri donuyorsunuz...\n");
                return;
            default:
                printf("Gecersiz secim! Lutfen tekrar deneyin.\n");
                return;
        }
    }while(secim<1 || secim>4);

    if(ozellikler[TOKLUK]>100)
        ozellikler[TOKLUK]=100;
    if(ozellikler[TOKLUK]<0)
        ozellikler[TOKLUK]=0;
    if(ozellikler[UYKU]>100)
        ozellikler[UYKU]=100;
    if(ozellikler[UYKU]<0)
        ozellikler[UYKU]=0;
}
```

Şekil 2.10: Kamp alanı fonksiyonu

```
***** KAMP ALANI *****
1. Kamp atesinin basinda calgi cal/sarki soyle
2. Nehirde yikan
3. Cadirina girip uyu
4. Koy meydanina don
Bir secenek secin: Gecersiz secim! Lutfen tekrar deneyin.

***** ANA MENU *****
1. Kamp alanina git
2. Sifahane git
3. Hana git
4. Maceraya atil
5. Seviye atla
6. Durumu goster
7. Oyundan cik
Bir secenek secin: |

***** KAMP ALANI *****
1. Kamp atesinin basinda calgi cal/sarki soyle
2. Nehirde yikan
3. Cadirina girip uyu
4. Koy meydanina don
Bir secenek secin: 4
Koy meydanina geri donuyorsunuz...

***** KAMP ALANI *****
1. Kamp atesinin basinda calgi cal/sarki soyle
2. Nehirde yikan
3. Cadirina girip uyu
4. Koy meydanina don
Bir secenek secin: 3
Cadirinizda uyuyorsunuz...

***** KAMP ALANI *****
1. Kamp atesinin basinda calgi cal/sarki soyle
2. Nehirde yikan
3. Cadirina girip uyu
4. Koy meydanina don
Bir secenek secin: 2
Nehirde yikaniyorsunuz...
Karakterin Temel Niteliklerinin Guncel Durumu -> Can:100.00 ,Toluk: 100.00, Uyku: 75.00, Hijyen: 100.00, Su: 95.00, Enerji: 25.00

***** ANA MENU *****
1. Kamp alanina git
2. Sifahane git
3. Hana git
4. Maceraya atil
5. Seviye atla
6. Durumu goster
7. Oyundan cik
Bir secenek secin: 1
Kamp alanina gidiyorsunuz...

***** KAMP ALANI *****
1. Kamp atesinin basinda calgi cal/sarki soyle
2. Nehirde yikan
3. Cadirina girip uyu
4. Koy meydanina don
Bir secenek secin: 1
Calginizi calmaya basladiniz...
Karakterin Temel Niteliklerinin Guncel Durumu -> Can:100.00 ,Toluk: 100.00, Uyku: 85.00, Hijyen: 95.00, Su: 95.00, Enerji: 25.00
```

Şekil 2.11: Kamp alanı fonksiyonu çıktısı

2. Menü ve Kullanıcı Seçim

- Menü oluştururlar:

Kullanıcıya sunulan seçenekler şunlardır:

1. Kamp ateşi başında çalgı çalmak veya şarkı söylemek.
2. Nehirde yıkamak.
3. Çadırda uyumak.
4. Köy meydanına dönmek.

- Kullanıcıdan giriş alınır:
- Kullanıcı seçimini yaptıktan sonra scanf ile secim değişkenine aktarılır ve switch-case yapısı kullanılarak ilgili işlem yapılır.

3. Eylemlerin Etkileri ve İşleyışı

Kullanıcının seçtiği eyleme bağlı olarak karakterin özellikleri güncellenir. Her bir seçenekin işleyışı ve etkileri:

Seçenek 1: Kamp Ateşi Başında Çalgı Çalmak veya Şarkı Söylemek

- Kullanıcı seçeneği seçtiğinde printf ile bir mesaj gösterilir.
- Etkiler:
 - **Moral:** +10 (artar).
 - **Enerji:** -5 (azalır).
 - **Su:** -5 (azalır).
 - **Uyku:** -15 (azalır).
 - **Tokluk:** -15 (azalır).

Seçenek 2: Nehirde Yıkamak

- İşleyiş: Kullanıcı seçimi yaptıktan sonra karakter nehirde yıkılır.
- Etkiler:
 - **Hijyen:** +20 (artırılır).

- **Su:** +15 (artırılır).
- **Uyku:** -10 (azalır).
- **Tokluk:** -10 (azalır).

Seçenek 3: Çadırda Uyumak

- İşleyiş: Kullanıcı uyuma eylemini seçtiğinde karakter dinlenir.
- Etkiler:
 - **Uyku:** +30 (artırılır).
 - **Enerji:** +20 (artırılır).
 - **Moral:** +20 (artırılır).
 - **Dayanıklılık:** +10 (artırılır).
 - **Güç:** +10 (artırılır).
 - **Tokluk:** -20 (azalır).
 - **Su:** -10 (azalır).

Seçenek 4: Köy Meydanına Dönmek

- Kullanıcı köy meydanına dönmeyi seçtiğinde fonksiyon sonlandırılır ve ana menüye geri dönülür.

2.1 ŞİFAHANE

```
void sifahane(float ozellikler[]) {  
  
    int secim;  
    do{  
        printf("\n***** SİFAHANE *****\n");  
        printf("1. Pahali tedavi. Sifacidan yaralarini治masini iste (3 altın, +30 Can)\n");  
        printf("2. Orta tedavi. Sifacidan merhem yapip surmesini iste (2 altın, +20 Can)\n");  
        printf("3. Ucuz tedavi. Sifacidan merhem isteyip yaralara kendin sur (1 altın, +10 Can)\n");  
        printf("4. Kov meydana don\n");  
  
        printf("Bir secenek secin: ");  
        scanf("%d", &secim);  
  
        switch (secim) {  
            case 1:  
                if (ozellikler[ALTIN] >= 3) {  
                    printf("Pahali tedavi alınıyor...\n");  
                    ozellikler[ALTIN] -= 3;  
                    ozellikler[CAN] += 30;  
                    ozellikler[GUC] +=4;  
                    ozellikler[DAVANIKLILIK]+=6;  
                    ozellikler[CEVIKLIK] +=5;  
                    ozellikler[MORAL] +=20;  
                } else{  
                    printf("Yeterli altınınız yok su kadar altınınız var : %.2f\n",ozellikler[ALTIN]);  
                }  
                break;  
        }  
    }  
}
```

```

        case 2:
            if (ozellikler[ALTIN] >= 2) {
                printf("Orta tedavi aliniyor...\n");
                ozellikler[ALTIN] -= 2;
                ozellikler[CAN] += 20;
                ozellikler[GUC] +=2;
                ozellikler[DAYANIKLILIK]+=3;
                ozellikler[CEVIKLIK] +=3;
                ozellikler[MORAL] +=15;
            } else{
                printf("Yeterli altininiz yok su kader altininiz var : %.2f\n",ozellikler[ALTIN]);
            }
        break;
    case 3:
        if (ozellikler[ALTIN] >= 1) {
            printf("Ucuz tedavi aliniyor...\n");
            ozellikler[ALTIN] -= 1;
            ozellikler[CAN] += 10;
            ozellikler[GUC] +=1;
            ozellikler[DAYANIKLILIK]+=2;
            ozellikler[CEVIKLIK] +=2;
            ozellikler[MORAL] +=10;
        } else{
            printf("Yeterli altininiz yok su kader altininiz var : %.2f\n",ozellikler[ALTIN]);
        }
    break;
    case 4:
        printf("Koy meydanina geri donuyorsunuz...\n");
        return;
    default:
        printf("Gecersiz secim! Tekrar deneyin.\n");
    }
}while(secim<1 || secim >4);

```

```

***** SİFAHANE *****
1. Pahali tedavi.Sifacidan yaralarini satmasini iste (3 altın, +30 Can)
2. Orta tedavi.Sifacidan merhem yapip surmesini iste (2 altın, +20 Can)
3. Ucuz tedavi.Sifacidan merhem isteyip yaralarina kendin sur (1 altın, +10 Can)
4. Koy meydanina don
Bir secenek secin: 4
Koy meydanina geri donuyorsunuz...

***** ANA MENU *****
1. Kamp alanina git
2. Sifahane git
3. Hana git
4. Maceraya atil
5. Seviye atla
6. Durumu goster
7. Oyundan cik
Bir secenek secin: |

***** SİFAHANE *****
1. Pahali tedavi.Sifacidan yaralarini satmasini iste (3 altın, +30 Can)
2. Orta tedavi.Sifacidan merhem yapip surmesini iste (2 altın, +20 Can)
3. Ucuz tedavi.Sifacidan merhem isteyip yaralarina kendin sur (1 altın, +10 Can)
4. Koy meydanina don
Bir secenek secin: 3
Ucuz tedavi aliniyor...
Karakterin Temel Niteliklerinin Guncel Durumu -> Can:100.00 ,Toluk: 100.00, Uyku: 90.00, Hijyen: 100.00, Su: 100.00, Enerji: 25.00

***** SİFAHANE *****
1. Pahali tedavi.Sifacidan yaralarini satmasini iste (3 altın, +30 Can)
2. Orta tedavi.Sifacidan merhem yapip surmesini iste (2 altın, +20 Can)
3. Ucuz tedavi.Sifacidan merhem isteyip yaralarina kendin sur (1 altın, +10 Can)
4. Koy meydanina don
Bir secenek secin: 2
Orta tedavi aliniyor...
Karakterin Temel Niteliklerinin Guncel Durumu -> Can:100.00 ,Toluk: 100.00, Uyku: 90.00, Hijyen: 100.00, Su: 100.00, Enerji: 25.00

***** ANA MENU *****
1. Kamp alanina git
2. Sifahane git
3. Hana git
4. Maceraya atil
5. Seviye atla
6. Durumu goster
7. Oyundan cik
Bir secenek secin: 2
Sifahaneye gidiyorsunuz...

***** SİFAHANE *****
1. Pahali tedavi.Sifacidan yaralarini satmasini iste (3 altın, +30 Can)
2. Orta tedavi.Sifacidan merhem yapip surmesini iste (2 altın, +20 Can)
3. Ucuz tedavi.Sifacidan merhem isteyip yaralarina kendin sur (1 altın, +10 Can)
4. Koy meydanina don
Bir secenek secin: 1
Pahali tedavi aliniyor...
Karakterin Temel Niteliklerinin Guncel Durumu -> Can:100.00 ,Toluk: 100.00, Uyku: 90.00, Hijyen: 100.00, Su: 100.00, Enerji: 25.00

```

Şekil 2.12: Şifahane fonksiyonu

Fonksiyonun Amacı

void sifahane(float ozellikler[]) fonksiyonu, oyuncunun karakterinin sağlık ve özelliklerini artırmak için kullanabileceğiniz bir şifa mekânı sunar. Şifahane üç farklı tedavi seçeneği sunar ve karakterin "altın" birikimine göre farklı derecelerde sağlık artırımı sağlar.

İşleyiş

1. Kullanıcı Seçimi

Kullanıcıya şifahane içinde dört seçenek sunulur:

1. Pahalı tedavi (3 altın karşılığında 30 CAN artışı).
2. Orta tedavi (2 altın karşılığında 20 CAN artışı).
3. Ucuz tedavi (1 altın karşılığında 10 CAN artışı).
4. Köy meydanına geri dönme.

2. Seçimin Kontrolü

Kullanıcı seçimini yaptıktan sonra:

- **Altın yeterli mi?**
- Eğer karakterin yeterli altını varsa, seçilen tedavi uygulanır ve özellikler güncellenir. Yetersiz altın durumunda kullanıcıya sahip olduğu altın miktarı gösterilir ve işlem yapılmaz.

3. Özelliklerin Güncellenmesi

Her tedavi, sadece CAN artışını değil, aynı zamanda karakterin diğer niteliklerini de etkiler:

- GÜÇ, DAYANIKLILIK, ÇEVİKLİK, MORAL gibi özellikler tedaviye bağlı olarak değişir.
- Örneğin, pahalı tedavi GÜÇ, DAYANIKLILIK ve MORAL gibi özelliklerde daha yüksek artış sağlar.

4. Köy Meydanına Dönme

Kullanıcı 4. seçeneği tercih ederse, işlem sonlandırılır ve karakter köy meydanına geri döner.

2.3 HAN

```
}

void han(float ozellikler[]) {
    int secim;
    do{
        printf("\n***** HAN *****\n");
        printf("1. Yiyecek satin al ve ye\n");
        printf("2. Icecek satin al, ic ve eglen\n");
        printf("3. Enstruman cal ve sarki soyle\n");
        printf("4. Koy meydanina don\n");

        int yiyecek_secimi;
        int icecek_secimi;
        printf("Bir secenek secin: ");
        scanf("%d", &secim);
```

Sekil 2.13: Han fonksiyonu

```

switch (secim) {
    case 1:
        do{
            printf("Xemek xiyeceklere...\\n");
            printf("Xiyeceklere ve fiyatlar\\n");
            printf("1-)Lehmesun=Tokluk 30 ruan , fiyat = 3 altin \\n");
            printf("2-)Iskender=Tokluk 50 ruan , fiyat = 5 altin \\n");
            printf("3-)Bog=Tokluk 10 ruan , fiyat = 1 altin \\n");

            printf("Hangi xiyecegi seciniz \\n");
            scanf("%d",&iyicecek_secimi);
            if(iyicecek_secimi == 1){
                if(ozellikler[ALTIN]>=3){
                    ozellikler[TOKLUK] += 30;
                    ozellikler[ALTIN] -= 3;
                    ozellikler[MORAL] += 10;
                    ozellikler[UYKU] -=10;
                    ozellikler[HİJYEN] -=20;
                    ozellikler[SU] -=15;
                    ozellikler[ENERJI] +=15;
                }else{
                    printf("Yeterli altininiz yok su kadar altininiz var : %.2f\\n",ozellikler[ALTIN]);
                }
            }else if(iyicecek_secimi == 2){
                if(ozellikler[ALTIN]>=5){
                    ozellikler[TOKLUK] += 50;
                    ozellikler[ALTIN] -= 5;
                    ozellikler[MORAL] += 10;
                    ozellikler[UYKU] -=10;
                    ozellikler[HİJYEN] -=25;
                    ozellikler[SU] -=20;
                    ozellikler[ENERJI] +=25;
                }else{
                    printf("Yeterli altininiz yok su kadar altininiz var : %.2f\\n",ozellikler[ALTIN]);
                }
            }
            else if(iyicecek_secimi == 3){
                if(ozellikler[ALTIN]>=1){
                    ozellikler[TOKLUK] += 10;
                    ozellikler[ALTIN] -= 1;
                    ozellikler[MORAL] += 10;
                    ozellikler[UYKU] -=5;
                    ozellikler[HİJYEN] -=10;
                    ozellikler[SU] -=5;
                    ozellikler[ENERJI] +=10;
                }else{
                    printf("Yeterli altininiz yok su kadar altininiz var : %.2f\\n",ozellikler[ALTIN]);
                }
            }
        }
    else
        printf("Hatali secim yaptiniz.Tekrar seciniz.\\n...");
    while(iyicecek_secimi<0 || iyicecek_secimi>3);

break;
}

```

Şekil 2.14: Han fonksiyonu Yiyecek Seçimi

```

***** HAN *****
1. Yiyecek satin al ve ye
2. Icecek satin al, ic ve eglen
3. Enstruman cal ve sarki soyle
4. Koy meydanina don
Bir secenek secin: 3
Hana Hosgeldiniz...
Sarki soyleyerken 12.25 altın kazandiniz!
Karakterin Temel Niteliklerinin Guncel Durumu -> Can:100.00 ,Tokluk: 75.00, Uykulu: 90.00, Hijyen: 90.00, Su: 100.00, Enerji: 25.00

```

```
***** HAN *****
1. Yiyecek satin al ve ye
2. Icecek satin al, ic ve eglen
3. Enstruman cal ve sarki söyle
4. Koy meydanina don
Bir seçenek secin: 1
Yemek yiyeorsunuz...
Yiyecekler ve fiyatları listesi...
1-)Lahmacun=Tokluk 30 puan , fiyat = 3 altın
2-)Iskender=Tokluk 50 puan , fiyat = 5 altın
3-)Muz=Tokluk 10 puan , fiyat = 1 altın
Hangi yiyecegi sectiniz
2
Yeterli altınınız yok su kadar altınınız var : 0.00
Karakterin Temel Niteliklerinin Guncel Durumu -> Can:100.00 ,Tokluk: 75.00, Uyku: 90.00, Hijyen: 90.00, Su: 100.00, Enerji: 25.00
```

```
***** ANA MENU *****
```

1. Kamp alanina git
 2. Sifahane git
 3. Hana git
 4. Maceraya atil
 5. Seviye atla
 6. Durumu goster
 7. Oyundan cik
- Bir seçenek secin: |

```
***** HAN *****
1. Yiyecek satin al ve ye
2. Icecek satin al, ic ve eglen
3. Enstruman cal ve sarki söyle
4. Koy meydanina don
Bir seçenek secin: 1
Yemek yiyeorsunuz...
Yiyecekler ve fiyatları listesi...
1-)Lahmacun=Tokluk 30 puan , fiyat = 3 altın
2-)Iskender=Tokluk 50 puan , fiyat = 5 altın
3-)Muz=Tokluk 10 puan , fiyat = 1 altın
Hangi yiyecegi sectiniz
2
Karakterin Temel Niteliklerinin Guncel Durumu -> Can:100.00 ,Tokluk: 75.00, Uyku: 90.00, Hijyen: 80.00, Su: 100.00, Enerji: 25.00
```

```
***** HAN *****

```

```
1. Yiyecek satin al ve ye
2. Icecek satin al, ic ve eglen
3. Enstruman cal ve sarki söyle
4. Koy meydanina don
Bir seçenek secin: 2
Icecek içiyorsunuz...
1-) Su (1 altın, Su +30)
2-) Enerji Icecegi (2 altın, Su +20, Enerji +30)
3-) Milkshake (3 altın, Su +10, Enerji +20)
Hangi icecegi sectiniz: 2
Enerji icecegi ildi. Enerji seviyeniz: 130.00
Karakterin Temel Niteliklerinin Guncel Durumu -> Can:100.00 ,Tokluk: 100.00, Uyku: 100.00, Hijyen: 100.00, Su: 100.00, Enerji: 25.00
```

Şekil 2.14: Han fonksiyonu çıktıları

Ana Menü Döngüsü:

- Kullanıcıya bir seçenek sunulur (1-4 arası). Menü aşağıdaki seçeneklerden oluşur:
 1. Yiyecek satın al ve ye.
 2. İçecek satın al, iç ve eğlen.
 3. Enstrüman çal ve şarkı söyle.
 4. Köy meydanına dön (fonksiyondan çıkış).
- Kullanıcının seçimine göre bir switch-case yapısı devreye girer.

Alt Menü ve İşlemler:

- **Yiyecek Menüsü (Seçenek 1):**
 - Kullanıcı, belirli yiyecekleri seçebilir (Lahmacun, Iskender, Muz).
 - Her yiyeceğin, karakterin özelliklerini nasıl değiştirdiği tanımlanmıştır.
Örneğin:
 - Lahmacun: TOKLUK artırır, ALTIN azaltır, HIJYEN düşürür.
 - Yeterli altını olmayan kullanıcıya uyarı verilir.
- **İçecek Menüsü (Seçenek 2):**
 - Kullanıcı, su, enerji içeceği veya milkshake alabilir.
 - Bu içecekler de karakter özelliklerini etkiler.
- **Enstrüman Çalma (Seçenek 3):**
 - Eğer HIJYEN seviyesi yeterliyse, kullanıcı şarkısı söyleyerek ALTIN ve TECRUBE kazanır.
 - TECRUBE seviyesi 100'ü aşarsa, seviyeAtla fonksiyonu çağrılır.
- **Köy Meydanına Dön (Seçenek 4):**
 - Kullanıcı köy meydanına geri döner, döngüden çıkarılır.

2.4 MACERA

1. Genel Fonksiyon Yapısı ve Amacı

Fonksiyonun Amacı:

Fonksiyon, oyuncunun bir macera seçmesi ve bu maceranın sonucunda, oyuncunun mevcut özelliklerinin (can, topluk, enerji gibi) değişmesini sağlamak amacıyla tasarlanmıştır. Oyuncunun yapacağı seçimler, oyunun gidişatını değiştiren faktörlerdir.

Fonksiyon Parametreleri:

Kodda float özellikler adlı bir dizi kullanılmıştır. Bu dizi, oyuncunun çeşitli özelliklerini tutar. Bu özellikler, oyuncunun sağlık topluk enerji şans gibi parametrelerinden oluşur ve oyuncunun oyun içindeki başarısını, seçimlerini ve ilerleyişini belirler. Bu dizi, her bir maceranın sonunda güncellenir.

```

printf("\n***** MACERA *****\n");
printf("1. Yakın çevreden şifalıl bitki topla ve avlan\n");
printf("2. Ormanı kesfe çık (Kolay)\n");
printf("3. Kayalıkları kesfe çık (Orta)\n");
printf("4. Vadisi kesfe çık (Zor)\n");
printf("5. Koy meydanına dön\n");

```

Şekil 2.15: Macera Menüsü

- 1. Şifalı Bitki Toplama ve Avlanması:** Oyuncu, yakın çevresinde şifalı bitkiler ve avlanmak için fırsatlar arar.
- 2-4. Keşif Görevleri:** Zorluk seviyelerine göre, orman, kayalıklar ve vadi keşifleri yapılır.
- 5. Koy Meydanına Dön:** Oyun sonlandırılır ve oyuncu ana köye döner.

Bu seçenekler, oyuncunun kararlarına bağlı olarak farklı yollar açar.

```

int sans = rand() % 100; // Rastgele sans değeri (0-99 arası)
double sifali_bitki_sans = (ozellikler[TOPLAYICILIK] * 4) / 100.0; // Yüzdelik oran hesaplama
int sifali_bitki_sans2 = sifali_bitki_sans * 100; // Ondalık değeri tam sayı yapmak için

if (sans < sifali_bitki_sans2) {
    // Sifalı bitki bulundu
    printf("Sifalı bitki bulundu! +10 Can\n");
    ozellikler[CAN] += 10;
    ozellikler[TOKLUK] += 10;
    ozellikler[UYKU] -= 15;
    ozellikler[ENERJI] += 15;
    ozellikler[MORAL] += 4;
    ozellikler[GUC] += 4;
    ozellikler[DAYANIKLILIK] += 7;
    ozellikler[CEVIKLILIK] += 4;
    ozellikler[TOPLAYICILIK] += 15;
}

```

Şekil 2.15: Macera menüsü şifalı bitki

`rand() % 100` fonksiyonu, 0 ile 99 arasında rastgele bir sayı üretir. Bu sayı, oyuncunun başarı oranını belirleyecek şans faktöründür.

Özellikler [TOPLAYICILIK] ifadesi, oyuncunun toplayıcılık yeteneğini temsil eder. Bu özellik, oyuncunun şifalı bitki bulma şansını etkiler. Oyuncunun bu yeteneği ne kadar yüksekse, şifalı bitki bulma şansı o kadar artar.

Eğer şans değeri, şifalı bitki bulma şansının sifali_bitki_sans2 değerinden küçükse, oyuncu şifalı bitki bulmuş olur.

Bu durumda oyuncunun özelliklerinde artışlar olur: CAN, TOKLUK , ENERJI , MORAL, GUC ,DAYANIKLILIK ,CEVİKLİK ,TOPLAYICILIK değerleri artar. Bu, oyuncunun sağlık, tokluk ve enerji durumunun iyileşmesi anlamına gelir.

```
// Meyve toplama gerçekleşirse yeni sansla avlanma kontrolü
int sans3 = rand() % 100; // Rastgele sans değeri (0-99 arası)
double av_sans = (ozellikler[TOPLAYICILIK] * 4) / 100.0; // Yüzelik oran hesaplama
int av_sans2 = (av_sans * 100)/2 ; // Ondalık değeri tam sayı yapmak için
if (sans3 < av_sans2) { //
    printf("Ve sanslısiniz! Bir av hayvanı avladiniz! +50 Tokluk\n");
    ozellikler[TOKLUK] += 50;
    ozellikler[UYKU]-=10;
    ozellikler[ENERJI]+=10;
    ozellikler[MORAL]+=3;
    ozellikler[GUC]+=2;
    ozellikler[DAYANIKLILIK]+=1;
    ozellikler[CEVİKLİK]+=3;
    ozellikler[TOPLAYICILIK]+=15;
    ozellikler[KARIZMA]+=7;
    ozellikler[TECRUBE]+=15;
    if (ozellikler[TECRUBE] >= 100)
        seviyeAtla(ozellikler);
}
} else {
    printf("hicbir sey bulamadiniz.\n");
}
break;
```

Şekil 2.16: Macera fonksiyonu şans

- Zincirleme kontrol yapıları, şifalı bitki bulunduktan sonra meyve toplama ve avlanma şanslarının kontrol edilmesini sağlar.
- Sans2 ve sans3 ile yapılan rastgele şans hesaplamaları, oyuncuya ekstra ödüller (meyve ve av) kazandırır.

Bu şekilde, her başarılı aksiyonun ardından bir sonraki aksiyon için yeni bir şans hesaplanır ve oyuncunun özellikleri artar.

```

int dusmanGuc = rand() % 3 + 1;
int dusmanCeviklik = rand() % 3 + 1;
int dusmanDayaniklilik = rand() % 3 + 1;
printf("Haydut: Guc: %d, Ceviklik: %d, Dayaniklilik: %d\n", dusmanGuc, dusmanCeviklik, dusmanDayaniklilik);
```

Şekil 2.17: Macera fonksiyonu random

Düşman özellikleri rastgele belirlenir. Düşmanın gücü, çevikliği ve dayanıklılığı, oyuncuya savaşırken karşılaşacağı zorlukları belirler. Bu değerler, savaşın dinamiklerini etkileyen önemli faktörlerdir.

```
// Ozan saldırıyor
int oyuncuSaldırı = 4 * oyuncuGuc;
dusmanCan -= oyuncuSaldırı;
printf("Ozan saldırdı: %d hasar verdi\n", oyuncuSaldırı);

if (dusmanCan <= 0) {
    break;
}

// Haydut savunuyor
int dusmanSavunma = 4 * dusmanDayanıklılık;
int alınanHasar = dusmanSavunma - (dusmanSavunma * (4 * oyuncuDayanıklılık / 100.0));
oyuncuCan -= alınanHasar;
printf("Haydut savundu, Ozan'a %d hasar verdi\n", alınanHasar);
} else {
    // Haydut saldırıyor
    int dusmanSaldırı = 4 * dusmanGuc;
    oyuncuCan -= dusmanSaldırı;
    printf("Haydut saldırdı: %d hasar verdi\n", dusmanSaldırı);

    if (oyuncuCan <= 0) {
        break;
    }

    // Ozan savunuyor
    int oyuncuSavunma = 4 * oyuncuDayanıklılık;
    int alınanHasar = oyuncuSavunma - (oyuncuSavunma * (4 * dusmanDayanıklılık / 100.0));
    dusmanCan -= alınanHasar;
    printf("Ozan savundu, Haydut'a %d hasar verdi\n", alınanHasar);
}

// Çevikliği büyük olan bir sonraki saldırımı var
oyuncuİlkSaldırı = 1 - oyuncuİlkSaldırı;
```

Şekil 2.17 Macera fonksiyonu savaş

Oyuncu ve düşman sırasıyla birbirlerine saldırır. İlk saldırıyı yapan taraf, çevikliği yüksek olan taraftır.

Oyuncu ve düşman arasında döngü devam eder, her iki tarafın canı sırasıyla azalır.

Bu mekanizma, oyuncunun saldırısı gücü ve düşmanın savunma gücü üzerinden belirlenir.

```

        }

        if (oyuncuCan > 0) {
            // Kazandildi
            printf("Kazandiniz!\n");
            ozellikler[TECRUBE] += 30;
            ozellikler[TOKLUK]-=10;
            ozellikler[UYKU]-=10;
            if (ozellikler[TECRUBE] >= 100)
                seviyeAtla(ozellikler);
            ozellikler[ALTIN] += (rand() % 11 + 15);
            printf("Haydut ganimetin: %.2f altin kazandiniz!\n", ozellikler[ALTIN]);
        } else {
            printf("Kaybettiniz!\n");
            ozellikler[CAN] = 0;
            oyundanCik();
        }
    }
}

```

Sekil 2.17: Macera fonksiyonu kazanma kaybetme durumu

Eğer oyuncuCan sıfırın altına inmezse, oyuncu savaşı kazanır ve belirli ödüller kazanır. Bunlar tecrübe puanı ve altın ödülündür. Eğer oyuncu kaybederse, oyundaki CAN değeri sıfırlanır ve oyun sona erer.

```

***** MACERA *****
1. Yakin çevreden sifalil bitki topla ve avlan
2. Ormani kesfe cik (Kolay)
3. Kayalıklari kesfe cik (Orta)
4. Vadisi kesfe cik (Zor)
5. Koy meydanina don
Bir secenek secin: 4
Vadiye zor bir kesfe ciktiniz!
Haydut: Guc: 9, Ceviklik: 8, Dayanıklilik: 7
Kacma sansi: 0, sans: 8
Kacmadiniz, savas basliyor...
Haydut saldirdi: 36 hasar verdi
Ozan savundu, Haydut'a 17 hasar verdi
Ozan saldirdi: 20 hasar verdi
Haydut savundu, Ozan'a 21 hasar verdi
Haydut saldirdi: 36 hasar verdi
Ozan savundu, Haydut'a 17 hasar verdi
Ozan saldirdi: 20 hasar verdi
Haydut savundu, Ozan'a 21 hasar verdi
Kaybettiniz!
Caniniz 0 a dustugu icin oyun sonlandi...
Oyun kapatiliyor...

```

```
***** MACERA *****
1. Yakin ceverden sifalil bitki topla ve avlan
2. Ormani kesfe cik (Kolay)
3. Kayaliklari Kesfe cik (Orta)
4. Vadisi kesfe cik (Zor)
5. Koy meydanina don
Bir secenek secin: 2
Ormanda kolay bir kesfe ciktiniz!
Haydut: Guc: 1, Ceviklik: 1, Dayanıklılık: 2
Kacma sansi: 0, sans: 60
Kacaamadiniz, savas basliyor...
Ozan saldirdi: 20 hasar verdi
Haydut savundu, Ozan'a 6 hasar verdi
Haydut saldirdi: 4 hasar verdi
Ozan savundu, Haydut'a 22 hasar verdi
Ozan saldirdi: 20 hasar verdi
Haydut savundu, Ozan'a 6 hasar verdi
Haydut saldirdi: 4 hasar verdi
Ozan savundu, Haydut'a 22 hasar verdi
Ozan saldirdi: 20 hasar verdi
Kazandiniz!
Haydut ganimet: 53.00 altın kazandiniz!
Karacterin Temel Niteliklerinin Guncel Durumu -> Can:100.00 ,Tokluk: 100.00, Uyku: 80.00, Hijyen: 100.00, Su: 100.00, Enerji: 25.00
```

Şekil 2.18: Macera Çıktıları

2.5 SEVİYE ATLA

Fonksiyonun Adı:

seviyeAtla

Amacı:

Bu fonksiyon, oyuncunun tecrübe puanlarının belirli bir eşeğe ulaştığında, bir üst seviyeye geçmesini sağlar. Seviye atlandığında oyuncuya çeşitli özelliklerini geliştirmek için puanlar sunulur.

Giriş Parametresi:

- ozellikler[]: Oyuncunun özelliklerini tutan bir dizi. Bu dizide, oyuncunun güç (guc), çeviklik dayanıklılık (karizma, toplayıcılık gibi becerileri ve tecrübe ile seviye bilgisi bulunur).

Tecrübe Kontrolü: Fonksiyon ilk olarak oyuncunun tecrübe puanını kontrol eder. Eğer ozellikler[TECRUBE] >= 100 ise oyuncu bir sonraki seviyeye geçer.

```
void seviyeAtla(float ozellikler[]) {
    if (ozellikler[TECRUBE] >= 100) {
        printf("Tebrikler! Seviye atladiniz.\n");
        ozellikler[SEVIYE]++;
        ozellikler[TECRUBE] = 0;
```

Şekil 2.17: Seviye atla fonksiyonu koşulu

- ozellikler[SEVIYE]++: Oyuncunun seviyesi bir artırılır.
- ozellikler[TECRUBE] = 0: Seviye atlandığında, mevcut tecrübe sıfırlanır.

Beceri Dağıtıımı: Seviye atlandığında oyuncuya 5 dağıtılabılır puan sunulur. Oyuncu, bu puanları özelliklerine eklemek için bir seçenek sunan bir döngüyle karşılaşır:

```

int puan = 5;
while (puan > 0) {
    printf("Beceri dagilimi icin kalan puan: %d\n", puan);
    printf("1. Guc\n2. Ceviklik\n3. Dayaniklilik\n4. Karizma\n5. Toplayicilik\n");
    printf("Bir secim yapin: ");
    int secim;
    scanf("%d", &secim);

    switch (secim) {
        case 1:
            ozellikler[GUC]++;
            break;
        case 2:
            ozellikler[CEVIKLIK]++;
            break;
        case 3:
            ozellikler[DAYANIKLILIK]++;
            break;
        case 4:
            ozellikler[KARIZMA]++;
            break;
        case 5:
            ozellikler[TOPLAYICILIK]++;
            break;
        default:
            printf("Gecersiz secim! Lutfen 1-5 arasinda bir secim yapin.\n");
            continue; // Hatali girdiste donguye devam et
    }
    puan--; // Her gecerli secim sonrasi puan azaltilir
}
else {
    printf("Seviye atlamak icin yeterli tescrubeniz yok!\n");
}

}

```

Şekil 2.17: Seviye atla fonksiyonu beceriler

Döngü, oyuncunun tüm puanlarını kullandığından emin olmak için çalışır.

Her geçerli seçim sonrası puan-- ile kalan puan sayısı azaltılır.

Kullanıcıdan alınan giriş (örneğin, secim) bir switch yapısıyla kontrol edilir:

Kullanıcı 1-5 arasında bir seçim yaptığından, ilgili özelliğin değeri artırılır.

Hatalı bir seçim yapılrsa, continue ile döngü yeniden başlar ve kullanıcıdan geçerli bir giriş yapılması istenir.

Tecrübe Eşini Karşılamayan Durum: Eğer oyuncunun tecrübeşi 100'den küçükse, seviye atlama işlemi yapılmaz ve bir uyarı mesajı görüntülenir:

```
Beceri dagilimi icin kalan puan: 1
1. Guc
2. Ceviklik
3. Dayaniklilik
4. Karizma
5. Toplayicilik
Bir secim yapin: 4
Haydut ganimet: 80.00 altın kazandiniz!
Karakterin Temel Niteliklerinin Guncel Durumu -> Can:100.00 ,Tokluk: 100.00, Uyku: 70.00, Hijyen: 100.00, Su: 100.00, Enerji: 25.00

***** ANA MENU *****
1. Kamp alanina git
2. Sifahane git
3. Hana git
4. Maceraya atil
5. Seviye atla
6. Durumu goster
7. Oyundan cik
Bir secenek secin: |
```

```
Razandiniz.
Tebrikler! Seviye atladiniz.
Beceri dagilimi icin kalan puan: 5
1. Guc
2. Ceviklik
3. Dayaniklilik
4. Karizma
5. Toplayicilik
Bir secim yapin: 4
Beceri dagilimi icin kalan puan: 4
```

Şekil 2.18: Seviye atla fonksiyonu beceriler

2.6 DURUM GÖSTER

```
void durumGoster(float ozellikler[]) {  
    printf("\n***** OZANIN DURUMU *****\n");  
    printf("Ad: %s\n", ad);  
    printf("Çalgi: %s\n", calgi);  
    printf("Altin: %.2f\n", ozellikler[ALTIN]);  
    printf("Seviye: %.2f\n", ozellikler[SEVIYE]);  
    printf("Tecrübe: %.2f\n", ozellikler[TECRUBE]);  
    printf("Can: %.2f\n", ozellikler[CAN]);  
    printf("Tokluk: %.2f\n", ozellikler[TOKLUK]);  
    printf("Uyku: %.2f\n", ozellikler[UYKU]);  
    printf("Hijyen: %.2f\n", ozellikler[HİJYEN]);  
    printf("Su: %.2f\n", ozellikler[SU]);  
    printf("Enerji: %.2f\n", ozellikler[ENERJI]);  
    printf("Moral: %.2f\n", ozellikler[MORAL]);  
    printf("Guc: %.2f\n", ozellikler[GUC]);  
    printf("Çeviklik: %.2f\n", ozellikler[CEVIKLIK]);  
    printf("Dayanıklılık: %.2f\n", ozellikler[DAYANIKLILIK]);  
    printf("Karizma: %.2f\n", ozellikler[KARIZMA]);  
    printf("Toplayıcılık: %.2f\n", ozellikler[TOPLAYICILIK]);  
}
```

Sekil 2.19: Durum Göster fonksiyonu

Fonksiyon Açıklaması: durumGoster

Bu fonksiyon, oyuncunun oyun içindeki tüm özelliklerini ekrana yazdırarak mevcut durumunu gösterir. Fonksiyonun amacı, oyuncunun fiziksel, duygusal ve yetenek seviyelerinin kontrol edilmesini sağlamaktır.

ozellikler[]: Oyuncunun tüm değerlerini (altın, seviye, güç, vs.) tutan dizi.

Ad ve çalğı değişkenleri, oyuncunun adı ve çalğı türünü temsil eder.

printf kullanılarak dizideki her özellik sırasıyla ekrana yazdırılır.

Gösterilen Özellikler:

- Ad, çalğı, altın, seviye, tecrübe, can, tokluk, uyku, hijyen, su, enerji, moral, güç, çeviklik, dayanıklılık, karizma ve toplayıcılık.

2.7 OYUNDAN ÇIK

```
}

void oyundanCik() {
    if(ozellikler[CAN]<=0) {
        printf("Caniniz 0 a düştüğü için oyun sonlandı...\n");
        printf("Oyun kapatılıyor...\n");
        exit(0);
    }
    printf("Çıkmak istediginizden emin misiniz?(1:Evet , 2:Hayır)\n");
    int onay;
    scanf("%d", &onay);

    if(onay == 1){
        printf("Oyun kapatılıyor...\n");
        exit(0);
    }
    else{
        printf("Oynaya devam ediliyor...\n");
    }
}
```

Şekil 2.20: Oyundan Çık fonksiyonu

Fonksiyon, oyuncunun can durumunu ve oyundan çıkış isteğini Kullanıcıdan onay alarak kontrol eder.

1. Can Kontrolü:

- o Eğer CAN değeri 0 veya daha düşükse:
 - Mesaj gösterilir: "Caniniz 0'a düştüğü için oyun sonlandı..."
 - Program sonlandırılır: exit (0).

```
***** ANA MENU *****
1. Kamp alanına git
2. Sifahane git
3. Hana git
4. Maceraya atıl
5. Seviye atla
6. Durumu göster
7. Oyundan çıkış
Bir seçenek seçin: 7
Oyundan çıkmak istiyorsunuz...
Çıkmak istediginizden emin misiniz?(1:Evet , 2:Hayır)
)2
Oynaya devam ediliyor...

***** ANA MENU *****
1. Kamp alanına git
2. Sifahane git
3. Hana git
4. Maceraya atıl
5. Seviye atla
6. Durumu göster
7. Oyundan çıkış
Bir seçenek seçin: |
```

```
***** ANA MENU *****
1. Kamp alanina git
2. Sifahane git
3. Hana git
4. Maceraya atil
5. Seviye atla
6. Durumu gostер
7. Oyundan cik
Bir secenek secin: 7
Oyundan cikmak istiyorsunuz...
Cikmak istediginizden emin misiniz?(1:Evet , 2:Hayir)
)1
Oyun kapatiliyor...
```

Şekil 2.21: Oyundan Çık fonksiyonu çıktısı

SENARYO:

Ozan'ın Hikayesi

Ozan, köy meydanında sessizce oturmuş, hayal dünyasına dalmış bir şekilde düşünüyordu. Köyde herkes onu neşeli şarkıları ve iyimserliğiyle tanııyordu. Ancak Ozan, sadece bir sanatçı olarak değil, aynı zamanda cesur bir maceracı olarak da kendini ispat etmek istiyordu.

Bir sabah, köyün muhtarı yanına gelerek endişeyle şu sözleri dile getirdi: "Ozan, köyümüzün çevresindeki ormanlarda sıra dışı olaylar yaşanıyor. Cesur birinin bu durumu araştırması gerekiyor. Senin yeteneklerin ve cesaretin, bu sorunları çözmek için çok değerli olabilir."

Bu sözlerden etkilenen Ozan, sorumluluğu kabul etti. Köy halkına yardım edebilmek için kararlılıkla hazırlıklarını tamamladı ve maceraya atılmaya karar verdi. Onu bekleyen zorluklar arasında keşfedilmeyi bekleyen ormanlar, tehlikeli mücadeleler ve kazandığı deneyimlerle köyüne döndüğünde anlatacağı ilham verici hikayeler vardı.

EKSİKLER VE GELİŞTİRMELER

EKSİKLER:

1- Geçersiz girişlerde programın beklenmedik şekilde çalışmasına yol açabilecek 1 hata var. Diğer kısımlar kontrol edildi hatası yoktur.

Hata olan kısım: ilk girişte menüden 7 seçenekten birini seçerken; geçersiz sayılar için if else koşulu koyduk fakat kullanıcı Harf girmesi durumunda kod hata veriyor. **Bu koşulu ASCII Kodu ile yapabileceğimizi biliyoruz** ama hem uzun olur hem de kullanıcının sayı girmesini istediğimiz için bu durumu yapmadık.

2-Konudan ve istenilen oyundan dolayı bazı kod bölümleri tekrar eden yapılar içeriyor. Örneğin, karakter özelliklerini sınırlamak için her fonksiyonun sonunda aynı kontrol kodu yer alıyor.

Örneğin 5 fonksiyonda if else koşulu ile değer aralığı tespit yapmak zorunda kaldık.

3-Karakter özelliklerinin statik bir dizi kullanılarak saklanması esnekliği azaltıyor. Sadece işlenilen yere kadar olan konular dahildir diye düşündük ve bu yüzden statik dizi kullandık.

GELİŞTİRMELER:

1-Dosyalalama işlemleri de oyuna eklerek oyunu kaydetme ve kaldığımız yerden devam etme eklenebilir.

2- Oyunun kod kısmı daha ayrıntılı hale getirilerek giriş kısmında kullanıcının zorluk seviyesi seçmesi sağlanabilir. Her zorluk seviyesinde daha farklı görev durum ve fonksiyonlarla birlikte oyun çok daha uzun ve eğlenceli hale gelebilir.

3- Kod tekrarlarından kaçmak için fonksiyon tanımlayabiliyorduk ama Ödev konsepti dışına çıkmak istemedik. Bu esneklik sağlanabilir.

4-Oyuncuların karakterlerini daha fazla kişiselleştirebilmesi sağlanabilir. Kullanıcı yeni özellikler ekleyebilir.

5-Oyun kodu daha da geliştirilerek tek kullanıcılı olmaktan çıkarılıp 2 kullanıcının karşılıklı oynayacağı moda getirilebilir.

6- İlerleyen zamanlarda C programlamada ustalaştıktan sonra görsel ve sesli ifadeler ekleyerek oyunun Siyah sıkıcı konsoldan ziyade herkesin oynarken eğlenebileceği bilgisayar ve telefon oyunu haline getirilebilir.

SONUÇLAR VE FARKLILIKLAR

1-Temel Niteliklere 3 yeni ekleme yapıldı: HIJYEN, SU, ENERJİ.

2- Oyuncunun seçimleri oyunun gidişatını doğrudan etkileyebilecek şekilde ve hikâye ilerleyişini şekillendirecek şekilde oyunumuzun kodunu yazdık. Eklediğimiz özelliklerin sürekli değişimi, oyuncuya stratejik kararlar almaya teşvik eder. Kullanıcı ozanı hayatı tutmak için farklı seçenekleri düşünmek zorunda kalacak ve oynarken eğlenecektir.

3-Oyuncuya gerçekçi ve sürükleyici bir oyun oynatmak için altın, tecrübe, hijyen gibi kaynaklarını dikkatlice yönetmesini sağlayacak şekilde kodumuzu yazdık.

4-Beceri ve nitelikleri tutmak ve puanlarını artırmak için dizi kullandık bu sayede fonksiyonları birbirine çok kolay şekilde bağladık.

5-Kodun ilk giriş kısmında #define ile sayı indekslerini tanımlayarak, dizilerde kullanımı kolaylaştırdık ve kodun okunurlüğünü artırdık. Bu sayede, indis yerine anlamlı kelimeler kullanarak hangi özelliğin artırıldığını veya azaltıldığını daha net bir şekilde anlamış olduk.

6-En Zor kısım olan macera fonksiyonu yazarken kurduğumuz farklılıklar:

A) En başta oyuncuya rastgele şans ve özelliklerine dayalı bir macera deneyimi sunduk.

B) Çeviklik statüsüne göre saldırı sırası belirlenmesi, her iki tarafın da dinamik bir şekilde etkileşime girmesini sağladık.

C) Ayrıca, savaş sırasında hasar hesaplamaları ve kazananın ödüllendirilmesi yaptık .

D) Savaşta oyuncunun ve düşmanın çeviklik değerlerine göre saldırı sırası belirlenmesini yaptık.

E) Zor keşif senaryosunda düşman güçleri rastgele belirlenerek her seferinde farklı bir mücadele oluşturuyor, bu da oyun deneyimini çeşitlendiriyor. Savaş sırasında saldırı ve savunma işlemleri, oyuncunun ve düşmanın çevikliğine göre dinamik bir şekilde değişiyor, bu da stratejiye dayalı kararlar alınmasını sağladık.

KAYNAKÇA

Boz, C. (2024). *Sıfırdan İleri Seviye Komple C Programlama Kursu* [Video]. Udemy.
https://www.udemy.com/course/sifirdan-ileri-seviyeye-komple-c-programlama-kursu/?srsltid=AfmBOorwrwzVa4ljV2oYctkwAbpwfSsQuK7YeO1DHayWuT1f7hD8_JVQ&couponCode=LETSLEARNNOW

TEŞEKKÜR

Bu projeyi hazırlarken aldığımız rehberlik ve destek için dersimizin hocalarına teşekkür ederiz.

Bu proje boyunca problem çözme ve oyun yapma gibi deneyimler kazandık. Özellikle, oyun hikayesi kurgulama ve mantıksal hataları çözme süreçlerinde çok şey öğrendik. Grup şeklinde ödevi yaptığımız için takım çalışmasını öğrendik. Ek olarak rapor hazırlama ve açıklamayı öğrendik

Bu deneyimler sayesinde, gelecekte karşılaşacağımız projelerde daha sağlam bir altyapı ile çalışabileceğimize inanıyoruz.

