

KOCAELİ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ

PROGRAMLAMA LABORATUVARI

PROJE 1

STRATEJİ OYUNU VE SAVAŞ MEKANİKLERİ

Saffet Hakan KOÇAK – 230202058

Yusuf BÜLBÜL - 230202050

I. ÖZET

Bu proje, JSON dosyalarından JSON dosyasındaki verileri çeken hazır fonksiyon kullanılmadan manuel bir şekilde verilerin çekilmesini, paylaşılan linklerden indirme yapıp veri çekilmesini, çekilen verilerin kullanılarak savaş mekaniği hesaplamalarının gerçekleştirilmesini, savaş mekaniği kısmında her bir adımdaki bilgilerin bir txt dosyasına yazdırılıp kaydedilmesini, savaş öncesi ve sonrasının görselleştirilmesini hedeflemektedir.

II. GİRİŞ

Proje, C programlama dili kullanılarak Code::Blocks idesinde yazılmıştır. Projemizde kullanılan fonksiyonlar şunlardır:

- `sola_dayama()`: Stringin başındaki boşlukları temizleyerek bulunduğu satırı sola dayar.
- `satira_git()`: Dosya içerisindeki istenilen satıra gidilmesini sağlar.
- Kahraman, canavar ve araştırma etkisi fonksiyonları verilecek bonusun hangi birime, hangi türde ve hangi değerde etki ettiğini hesaplar.
- `efect_belirleme_insan/ork` fonksiyonları kahraman ve canavar etkilerini ilgili birime parlama efektiyle göstererek bu etkinin belirtilmesine yardımcı olur.

III.YÖNTEM

Bu projede, İnsan İmparatorluğu ve Ork Lejyonu arasındaki savaş mekaniği simülasyonunun geliştirilmesi amaçlanmıştır. Ana savaş unsuru olarak çeşitli birlik türleri, kahramanlar, canavarlar ve araştırmalar yer almaktadır.

A. Proje Hakkında

Projede Code::Blocks idesinde C dilinde yazılmış fonksiyonları, komutları yaptıran kodlar bulunmaktadır. Projeyi çalıştırmak için 10 farklı senaryo linki bulunmaktadır. Kullanıcı girişinin isteğine göre link indirilip JSON dosyası haline getirilip dosyadan senaryo verileri çekilmektedir. Çekilen veriler savaş mekaniğine sokularak çeşitli hesaplamalar yaptırılmaktadır. Yapılan hesaplamalar sonrası istenen değerler txt dosyasına yazdırılıp kaydedilmektedir. Savaş öncesi ve savaş sonrası görseller proje çalıştırıldığında ekranda gösterilmektedir. Elimizden geldiğince projeye uygun adımlar atma gayreti içerisinde olduğumuzu belirtmek isteriz.

B. Struct Yapıların Oluşturulması

Kodun anlaşılır, düzenli ve işlevli olması için struct yapıları kuruldu. İç içe struct yapıları kullanıldı. Bunlardan bir tanesini aşağıda açıklayacağız: Struct insan imparatorluğu içerisinde 4 tane struct yapısı bulunmaktadır. Bunlar struct research, struct creatures, struct heroes, struct unit_I structlarıdır. 4

struct yapısı arasından struct unit_I'den ilerleyelim. Değişken adı i_unit olarak verildi. Bu struct da içerisinde 4 adet struct bulunmaktadır. Bu structların hepsi struct unit_ortak türündedir. Bunlar aynı struct türü olduğu için bu 4 değişken de aynı özelliklere sahip olacaktır. Değişken isimleri piyadeler, okcular, süvariler ve kusatma_makineleridir. Her birimin özellikleri struct unit_ortakta tanımlanmış olan int veri tipinde "saldiri", int veri tipinde "savunma", double veri tipinde "kritik_sans" ve double veri tipinde "saglik"dir.

```

struct seviye
{
    int deger;
    char etkiledigi_birim[50] ;
    char etki_turu[50] ;
};

struct seviyeler
{
    struct seviye level1;
    struct seviye level2;
    struct seviye level3;
};

struct research
{
    struct seviyeler savunma_ustaligi;
    struct seviyeler saldiri_gelistirmesi;
    struct seviyeler elit_egitim;
    struct seviyeler kusatma_ustaligi;
};

struct unit_ortak
{
    int saldiri;
    int savunma;
    double saglik;
    double kritik_sans;
};

struct unit_I
{
    struct unit_ortak piyadeler;
    struct unit_ortak okcular;
    struct unit_ortak suvariler;
    struct unit_ortak kusatma_makineleri;
};

struct insan_imparatorlugu
{
    struct research i_research;
    struct creatures_I i_creatures;
    struct heroes_I i_heroes;
    struct unit_I i_unit;
};

```

Fig.1 İç İç Structlar

C. Json Dosyalarından Veri Çekilmesi

4 adet JSON dosyasından birim, kahraman, canavar ve araştırma ile ilgili gerekli olan veriler çekildi. Senaryo JSON dosyası için ise bize verilen 10 adet linkten kullanıcıya hangisini çalıştırmak istediği sorularak seçilen JSON dosyasının verileri çekildi. Çekilen veriler yukarıda da anlatılmış olduğu gibi struct yapılarında tutuldu. Verileri çekerken kendimizin yazdığı satira_git() ve sola_dayama() fonksiyonları, string kütüphanesinden strstr(), strcpy(), sscanf(), fgets() fonksiyonları kullanıldı. JSON dosyalarındaki verilerin yerlerinin farklı olduğu durumlarda da veri çekme işlemimiz doğru bir biçimde çalışacak şekilde koşul blokları, döngüler, fonksiyonlar kullanılarak ayarlandı.

E. Savaş Mekaniği

JSON dosyalarından çekilen verilerin tutulduğu struct yapıları ve kullanılacak değişkenler tanımlanarak savaş mekaniği yazılmaya başlandı. İlk olarak while döngüsü şart bloğu içinde her adımda kontrol edilecek koşullar yazıldı. Döngünün içerisinde her adımda tekrar hesaplanacak şekilde birimlerin toplam saldırı - savunma güçlerini, sağlıklarını, sayılarını hesaplayan; kahraman, canavar, araştırma etkilerini hesaplayan; yorgunluk durumunu kontrol edip hesaplayan; kritik vuruş şansına göre kritik vuruş tur sayısını hesaplayan; ilk adımda insanlar orklara saldırarak şekilde sırasıyla birbirine saldırtan; başlangıç bilgilerini ve her adım sonrasında birimlerin sayısını, toplam saldırı gücünü, toplam savunma gücünü, sağlığını, insan imparatorluğunun ve ork lejyonun toplam saldırı ve savunma gücünü savas_sim.txt dosyasına yazdıran kodlar yazıldı. Saldırının gerçekleştiği her turda, elde edilen toplam hasar, savunan tarafın savunma gücüne oranla birimlere dağıtılmıştır.

F. Görselleştirme

Görselleştirme için Raylib Kütüphanesi kullanıldı. İnsan İmparatorluğu ve Ork Lejyonu'nun hayali bir savaş arenasında savaştığı resim üzerinde 20*20'lik kare ızgara tasarlandı. Izgara ikiye bölündü, sol taraf İnsan İmparatorluğu'na sağ taraf ise Ork Lejyonu'na ayrıldı. Hem ızgara üzerine yazdırılan yazıyla hem de ızgara içinin doldurulduğu renklerle savaş alanı belirgin hale getirildi. İnsan İmparatorluğu ve Ork Lejyonu birlikleri ızgaranın belirli hücrelerine yerleştirildi. Her bir birlik türü için ayrı PNG kullanıldı ve yerleştirilen hücrelerde bu PNG'ler ilgili birlikler için gösterildi. İnsan İmparatorluğu tarafında yer alan kahraman ve canavar karakterleri ızgaranın sol kısmında, Ork Lejyonu tarafında yer alan kahraman ve canavar karakterleri de ızgaranın sağ kısmında gösterildi. Senaryoda bulunan karakterlere göre görseller ve altında hangi karakterin olduğu yazılarak ayarlandı ve koda entegre edildi. Ayrıca kahraman ve canavar karakterlerinin savaş alanında ilgili birliklere verdiği bonus etkiler savaş öncesi görselinde belirli efektlerle gösterildi. JSON dosyalarından çekilen verilerle İnsan İmparatorluğu ve Ork Lejyonu birliklerinin savaş öncesi sayıları, sağlık barları ve sağlık yüzdeleri ilgili birimin etrafında gösterildi. Böylelikle savaş öncesi görsel tamamlanmış oldu. Savaşın başlamasıyla beraber savaş mekaniği hesaplamaları devreye girdi ve savaşan taraflar arasında birim kayıpları ve sağlık azalması olayları gerçekleşti. Savaş sonrası görselde savaşı kaybeden taraf tamamen yok oldu. Savaş sonrasında kaybeden tarafın kahraman ve canavarları ekranda gösterilmedi. Savaşı kazanan tarafın ilgili birimlerinin savaş sonrası sayılarının ne kadar kaldığı ve güncel sağlık barları yine aynı şekilde ilgili hücrenin etrafında gösterildi.



Fig.2 Oyun Menüsü

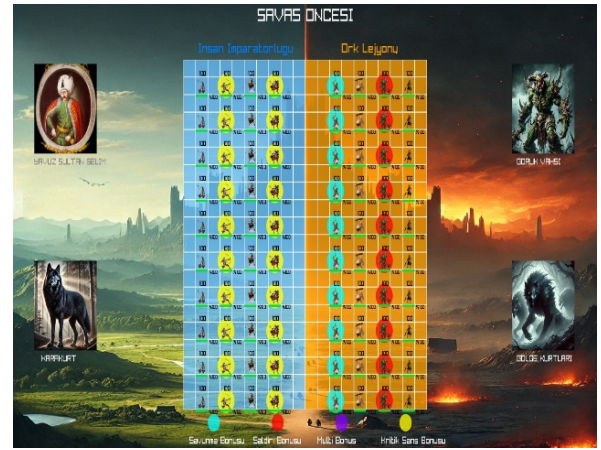


Fig.3 Savaş Öncesi Görsel

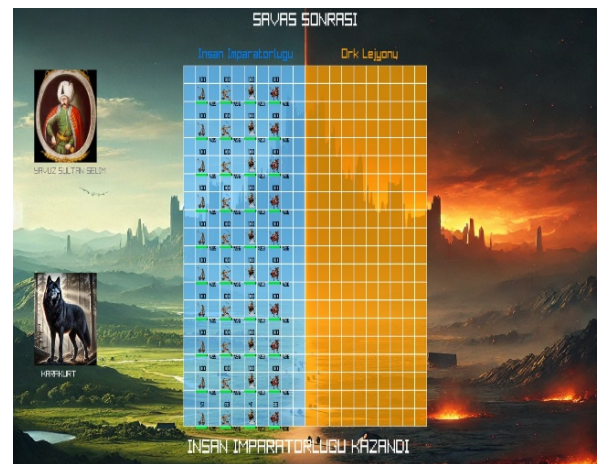


Fig.4 Savaş Sonrası Görsel

G. Projede İstenilen İsterler

Bu projede JSON dosyalarından stringler ayrıştırılarak açıklamalardaki kullanılacak verilerin çekilmesi, internetten indirilecek URL adresinin kullanıcı girişine göre indirilecek senaryo verisinin çekilmesi, çekilen verileri savaş mekaniği kısmında hesaplamalara sokarak çeşitli formüller kullanarak birimlerin saldırı - savunma güçleri, sağlıkları, sayıları vs. adım adım hesaplanarak her adımın bir txt dosyasına yazdırılıp kaydedilmesi ve savaş öncesi-sonrasının görselleştirilmesi isteniliyor.

```
Senaryoya Ait Başlangıç Bilgileri:
*****ORK LEJYONU*****

-----BİRİMLER-----

Ork Dovusculeri(Saldiri Gucu: 35,Savunma Gucu: 25,Saglik: 100,Kritik Vurus Sansi: 9.20)

Toplam sayi: 1000
Toplam saldiri gucu: 42000
Toplam savunma gucu: 25000

Mizrakcilar(Saldiri Gucu: 45,Savunma Gucu: 20,Saglik: 90,Kritik Vurus Sansi: 5.75)

Toplam sayi: 1000
Toplam saldiri gucu: 45000
Toplam savunma gucu: 20000

Varg binicileri(Saldiri Gucu: 55,Savunma Gucu: 35,Saglik: 130,Kritik Vurus Sansi: 6.90)

Toplam sayi: 1000
Toplam saldiri gucu: 55000
Toplam savunma gucu: 40250

Troller(Saldiri Gucu: 70,Savunma Gucu: 40,Saglik: 200,Kritik Vurus Sansi: 5.75)

Toplam sayi: 1000
Toplam saldiri gucu: 70000
Toplam savunma gucu: 40000

Ork Lejyonu Toplam Saldiri Gucu: 212000
Ork Lejyonu Toplam Savunma Gucu: 125250

-----KAHRAMAN-----

Goruk Vahsi ("ork_dovusculerine %20 "saldiri", gucu bonusu saglar)

-----CANAVAR-----

Golge Kurtlari ("varg_binicilerine %15 savunma", gucu bonusu saglar)

-----ARASTIRMA-----

Elit Egitim (Seviye 3): tum_birimlerin kritik % 15 artirir.
```

Fig.4 Senaryoya Ait Başlangıç Bilgileri

```
*****ORK LEJYONU*****

-----BİRİMLER-----

Ork Dovusculeri

Toplam sayi: 0
Toplam saldiri gucu: 0
Toplam savunma gucu: 0
Toplam saglik: 0

Mizrakcilar

Toplam sayi: 0
Toplam saldiri gucu: 0
Toplam savunma gucu: 0
Toplam saglik: 0

Varg binicileri

Toplam sayi: 0
Toplam saldiri gucu: 0
Toplam savunma gucu: 0
Toplam saglik: 0

Troller

Toplam sayi: 0
Toplam saldiri gucu: 0
Toplam savunma gucu: 0
Toplam saglik: 0

Ork Lejyonu Toplam Saldiri Gucu: 0
Ork Lejyonu Toplam Savunma Gucu: 0

-----INSAN IMPARATORLUGU Savasi Kazandi-----
```

Fig.5 Senaryoya Ait Savaş Sonu Bilgileri

IV. DENEYSEL SONUÇLAR

Kurulan iç içe struct yapılarının yazılan koddaki işlevi önemlidir.

Senaryo dosyası verilerinin çekilip structlara doğru bir şekilde atanması kodun tutarlı olması açısından önemlidir.

Savaş mekaniğinde kullanılan formüller

Savaş sürecindeki her bir adımdaki bilgilerin dosyaya doğru ve anlaşılır bir şekilde yazdırılıp kaydedilmesi önemlidir.

Genel hatlarıyla yapmış olduğumuz isterler tam anlamıyla çalışmaktadır.

V. SONUÇ

Projeyi bizden istenilen şekilde tamamlayabilmemiz bizi mutlu etti. İlk projemizde verdiğimiz emeklerin karşılığını aldığımız için mutluyuz. Bu projenin takım çalışma, problem çözme algoritmik düşünme, araştırma yapma ve sabırlı olma becerimizi geliştirdiğini düşünüyoruz.

VI. YAZAR KATKILARI

Yusuf Blbl : JSON dosyalarından manuel şekilde veri çekme işleminde görev aldı. Savaş mekaniği hesaplamalarında rol aldı. Main fonksiyonunda kendine düşen kısımları yazdı. Görselleştirmede yapılan efektler için çalışmalar yaptı.

Saffet Hakan Koçak : Proje için gerekli kütüphanelerin kurulumunu yaptı. Savaş mekaniği ve görselleştirme adımlarında rol aldı. Ayrıca JSON ayrıştırma ve main fonksiyonunda ilgili kısımları yazmada görev aldı. Hata tespit ve ayıklama sürecinde yer aldı.

Her iki yazar proje boyunca iş birliği yapmış ve süreci sabırlı ve istikrarlı bir şekilde yönetmişlerdir.

VII. KAYNAKÇA

<https://www.youtube.com/watch?v=a2IUxJFr8sg>

https://www.youtube.com/watch?v=OWyotSHKd8M&list=PLY20HpF-ruiK1cFSUqhP_XJSc_T0LWRh6t

<https://youtu.be/eLRXgpNB4dk?si=dxvSnZ9FKwf3FsZq>

<https://www.oracle.com/tr/database/what-is-json/>

<https://www.raylib.com/>

<https://archman.org/topluluk/belgeler/nasil-wget-nedir-wget-komutlari-nelerdir-wget-nasil-kullanilir/>

<https://youtu.be/rn-wJqJmTWo?si=cIqcjBmLoRC0ODAD>