

Programlama Laboratuvarı 1.Proje Raporu

Sadık Gölpek

Kocaeli Üniversitesi
Bilgisayar Mühendisliği
Eskişehir, Türkiye
sadikgolpek@gmail.com

Abdullah Önder

Kocaeli Üniversitesi
Bilgisayar Mühendisliği
Malatya, Türkiye
abdullahonder726@gmail.com

I.Özet

Bu çalışmada, C dilinde geliştirilmiş bir strateji oyunu sunulmaktadır. Oyunda, insan imparatorlukları ile ork legleri arasındaki savaş mekanikleri simüle edilip görselleştirilmiştir. Savaşlar, ızgara tabanlı bir haritada gerçekleşirken, her bir birim farklı özelliklere sahiptir. Kahramanlar ve canavarlar, birimlerin performansını etkileyerek oyunun dinamiklerini zenginleştirir. Özellikle, kahramanların sahip olduğu bonuslar ve canavarların özel yetenekleri, savaş mekaniğinde kullanılıp savaşın sonucunu etkiler.

Ayrıca, araştırma seviyeleri ve bunların savaş üzerindeki etkileri de önemli bir rol oynamaktadır. Araştırmalar, birimlerin güçlerini, dayanıklılıklarını ve çeşitli yeteneklerini artırmak için kritik bir unsur olarak karşımıza çıkmaktadır. Üstelik her 5 turda etkili olan yorgunluk faktörü tüm birimlere etki eder ve kritik vuruş oranlarıyla birlikte hesaplanan adım sayılarında birimlerin saldırı güçleri 1.5 katına çıkabilir.

Görevimiz, JSON dosyalarından gelen verileri manuel olarak okuyarak bu savaş mekaniklerini hesaplamak, simüle etmek ve etkili bir şekilde görselleştirmektir. Projemizde, savaşın başlangıcı ve son durumu görselleştirilmiş; bu görselleştirme, savaş mekaniğinde hesapladığımız verileri doğru bir şekilde simüle etmemizi sağlamıştır.

sunmaktadır. Bu tür oyunlar, karmaşık kuralları, dinamik etkileşimleri ve çok çeşitli senaryoları içerir. Gelişen oyun teknolojileri, hem grafiklerin kalitesini hem de oyun mekaniklerinin karmaşıklığını artırarak daha çekici ve etkileşimli deneyimler sunmaktadır. Bu projede, savaş mekaniklerinin daha iyi anlaşılmasını sağlamak ve farklı simülasyonlar üzerinden görsellik sunmak amacıyla bir strateji oyunu geliştirmeyi hedefliyoruz.

Bu oyunda, insan imparatorlukları ve ork legleri arasındaki çatışmaların dinamikleri incelenmiştir. Savaş mekanikleri, sadece birimlerin güçleriyle değil, aynı zamanda kahramanların ve canavarların oyun içindeki rolleri ile de şekillendirilmektedir. Kahramanlar ve canavarlar, insan birimlerine ve ork birimlerine ekstra avantaj sağlar.

Veri yönetimi de bu projede önemli bir yer tutmaktadır. Dış kaynaklardan veri okuma süreci, oyunun temel unsurlarını belirlemek için kritik öneme sahiptir. Projede, birim özellikleri, kahraman bonusları ve araştırma değerleri gibi bilgileri içeren JSON formatında yapılandırılmış veri dosyaları kullanılmaktadır. Bu dosyalar, oyunun temel mekaniklerinin oluşturulmasına katkı sağlarken, oyunculara sunulacak deneyimin kalitesini artırmak için dikkatlice tasarlanmıştır. Verilerin düzgün bir şekilde işlenmesi, oyunun çalışabilmesi açısından da büyük önem taşımaktadır.

II.Giriş

Günümüzde, strateji oyunları oyunculara sadece eğlence sunmanın ötesinde, düşünme ve stratejik planlama yeteneklerini geliştirme fırsatı da

Bu proje bizim açımızdan, sadece savaş

mekaniklerinin simülasyonu ile sınırlı kalmayıp, oyun geliştirme sürecinde karşılaşılan zorlukları da ele almıştır. Özellikle, C dili gibi nesne yönelimli olmayan bir programlama dilinde oyun geliştirmek, bize deneyim kazandırmış ve verileri yönetmek için struct kullanımının, etkili fonksiyon kullanımının, kod optimizasyonunun ve proje yönetiminin ne denli önemli olduğunu göstermiştir. Bu süreçte karşılaştığımız sorunlar ve çözümler, yazılım geliştirme alanında bize önemli katkılar sunmaktadır.

III. Yöntem

Proje, C programlama dili kullanılarak geliştirilmiştir. Savaş mekaniği, ızgara tabanlı bir haritada simüle edilmiştir. Bizim görevimiz, ana JSON dosyalarından ve senaryo JSON dosyalarından veri okuyarak birimlerin performansını hesaplamakta ve savaş senaryolarını oluşturmaktır. JSON dosyaları, hazır bir kütüphane kullanılmadan manuel olarak ayrıştırılmıştır. Bu ayrıştırma işlemi, projenin temelini oluşturmakta ve oyun mekaniklerinin işleyişine dair bir anlayış sunmaktadır.

İlk olarak, dört ana fonksiyon tanımlanmıştır:

“unittypes_readparse()”: Bu fonksiyon, insan imparatorluğu ve ork legi birimlerinin saldırı, savunma, sağlık ve kritik şans verilerini JSON dosyasından kütüphane kullanmadan manuel olarak çekerek oluşturduğumuz global structlara atar. Bu aşama, birimlerin temel özelliklerinin doğru bir şekilde tanımlanmasını sağlamaktadır.

“heroes_readparse()”: Bu fonksiyon, insan ve ork legi birliklerine bonus veren kahramanların bonus türü, bonus değeri ve hangi birimlere bonus verdiği bilgilerini global structlara aktarır. Bu işlem de kahramanların stratejik önemini anlamamıza yardımcı olmaktadır.

“creatures_readparse()”: İnsan imparatorluğu ve

ork legi canavarlarından etki eden etki türü, etki değeri ve kime etki ettiği bilgilerini global structlara atarak birimlerin performansını etkileyen unsurları işler.

“research_readparse()”: Araştırma değerlerini ve bunların insan imparatorluğu ve ork legi birliklerine etkisini, yine JSON dosyalarından manuel olarak çekip global structlara atan bu fonksiyon, savaş mekanikleri için önemli bir bilgi kaynağıdır. Araştırmaların etkisi, savaş sonucunu belirleyebilen bir faktör olmaktadır.

Bu dört fonksiyondan sonra, savaş_mekanigi() fonksiyonu, 10 senaryo dosyasından rastgele birini seçerek bu dört ana JSON verisinin matematiksel denklemler ile işlenmesi için son adımı tamamlar. Arama algoritmaları kullanılarak çekilen JSON verileri, asker sayıları ve birimlerin simülasyondaki etkilerini belirler. Tüm bu veriler, manuel olarak çekildikten sonra savaş mekaniğini inşa etmeye başlanmaktadır. Bu süreçte kodumuzu optimize etmek ve çalıştırmak için çeşitli kütüphaneleri kullandık.

Kullanılan Kütüphaneler:

“<stdio.h>”: Giriş/çıkış işlemleri için standart kütüphane.

“<stdlib.h>”: Bellek yönetimi ve diğer yardımcı işlevler için kullanılır.

“<string.h>”: JSON verisinin işlenmesi için gerekli olan string işlemleri için.

“<math.h>”: Savaş mekaniğindeki matematiksel işlemler için.

“<curl/curl.h>”: İnternette veri çekmek için kullanılır.

“<raylib.h>”: Görselleştirme için kullanılır.

Projemizde çakışmaları önlemek amacıyla, aşağıdaki #define direktifleri kullanılmıştır:

```
#define LoadImageA LoadImage
#define DrawTextA DrawText
#define PlaySoundA PlaySound
#define DrawTextExA DrawTextEx
```

Görselleştirme sürecinde, raylib kütüphanesi kullanılarak savaş alanı 14*11 bir ızgara yapısı ile gösterilmiştir. Her hücreye maksimum 100 asker yerleştirilebilmekte ve hücredeki fazla birlik bir altındaki boş hücreye taşınmaktadır. Projenin geliştirme ortamı olarak Visual Studio Code kullanılmış, test ortamı için ise “Code::Blocks” tercih edilmiştir.

Temel algoritma olarak, main fonksiyonunda dört ana JSON dosyasını çekmek için dört fonksiyon ve savaş mekaniğini hesaplamak için bir fonksiyon oluşturulmuştur. İlk dört fonksiyon, verileri bir char dizisine aktardıktan sonra, ayrıştırma işlemi ile oluşturulan yapılar (struct) içerisine yerleştirilmiştir. Ayrıca, savaş mekaniği fonksiyonu içinde 10 senaryo dosyası bulunmaktadır. Bu senaryo dosyaları, curl kütüphanesi kullanılarak internetten veri talepleri yapılarak elde edilmiş, ardından yine aynı mantıkla veriler ayrıştırılarak yapılarına aktarılmıştır. Ayrıca bölümden arkadaşımız Çağatay Altıntopaç’tan görselleştirme kısmında kullanmak için müzik yapmasını istedik. O da özgün bir şekilde bir savaş müziği yapıp bizim müziği kullanmamızı sağladı. Son olarak da senaryolarda kazanan tarafın İnsan İmparatorluğu olması durumunda İnsan İmparatorluğunun kazandığına dair bir yazı, ork leglerinin kazanması halinde ise ork leglerinin kazandığına dair bir yazı savaş bitiminde görselleştirme kısmının en sonunda alt kısımda yazacak şekilde ayarlanmıştır.

IV. Deneyisel Sonuçlar

Projenin deneyisel aşamasında, farklı senaryolar test edilmiştir. Rastgele bir simülasyonda savaşın başlangıç durumu ve sonuç durumu şu şekilde görselleştirilmiştir:

Örnek bir senaryoda savaşın başlangıcı:



Aynı senaryoda savaşın sonu:



Başlangıçta insan ve ork legleri birbiri ile savaşır,simülasyonda kahraman ve canavarlar eğer insanların kahramanları ve canavarları ise insanların yanında yani solda,orkların kahramanları ve canavarları ise orkların yanında yani sağ tarafta dizilirler.

Sonda ise kaybeden birlik ve canavarları ve kahramanları yok olur,kazanan birlik ise her canları azalmış bir biçimde görselleştirilmede gösterilir.

Deneyisel sonuçlar, savaş mekaniklerinin dinamiklerini ve birimlerin performansını etkileme oranlarını ortaya koymuştur. Farklı senaryoların test edilmesi, oyun dengesi üzerinde önemli geri bildirimler sağlarken, aynı zamanda oyun tasarımında yapılan değişikliklerin etkilerini anlamamıza yardımcı olmuştur.

V. Sonuç

Bu proje, savaş mekaniklerini simüle eden bir strateji oyunu geliştirmiştir. İnsan imparatorlukları ve ork legleri arasındaki savaşı görselleştirmek bunları savaş mekaniğiyle denklemlere döküp ayarlamak bize teknik anlamda güzel bir deneyim oldu,Ayrıca bir proje nasıl yapılır proje yönetimi nasıl olmalıdır,takım çalışmasının önemi gibi bilgileri proje sayesinde kendimize kattığımızı düşünüyoruz. Proje, oyun geliştirme sürecinde JSON verileri ile çalışma deneyimi sağlamış ve görselleştirme teknikleri hakkında bilgi edinilmesine olanak tanımıştır.

Sonuç olarak, projemiz, ilk projemiz olması dolayısıyla bize önemli katkı sağlamaktadır.

VI. Yazar Katkıları

Bu projede Abdullah, görselleştirme kısmında kodun optimizasyonunda çalışırken; Sadık, savaş mekaniği ve JSON verilerinin çekimi üzerinde yoğunlaşmış ve projenin son kısımlarında olarak

kodu optimize etmiştir. Ayrıca, senaryo dosyalarından veri çekmek için curl kütüphanesi kullanarak internette veri talepleri gerçekleştirilmiştir. Projenin tüm aşamalarında iş birliği ve ortak çaba, projenin ilerlemesinde önemli bir rol oynamıştır. projede birbirlerimizle iletişim halinde çalışarak, kod üzerinde gerçekleştirdiğimiz optimizasyonlarla projenin performansını artırdık ve hata ayıklama süreçlerini birlikte yürüttük.Böylece projenin her aşamasında daha yüksek bir kalite ve uyum sağlanmıştı.

Aşağıda projemizi yaparken yararlandığımız kaynaklar gösteriliyor.

Kaynakça:

JSON: <https://www.json.org/json-en.html>

JSON dosyalarındaki verilerin hiyerarşisini anlamak için :

<https://jsoncrack.com/editor>

Curl : <https://curl.se/>

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

Stack Overflow. Çakışma Hataları Çözümü:

<https://stackoverflow.com/>

ChatGPT. Çakışma Hatalarının Çözümü İçin Kullanılan Bilgiler:

<https://chatgpt.com/>

Görseller için kaynak:

<https://www.artbreeder.com/create>

Görselleştirme olurken kullanılan müzik Çağatay Altıntopaç tarafından yapılmıştır.