

SAVAŞ SİMÜLASYONU: İNSAN İMPARATORLUĞU VE ORK LEJYONU

Özet

Bu projede iki ırkın (İnsan İmparatorluğu ve Ork Lejyonu) savaşının simülasyonu yapılmaktadır. Savaş; kahramanlar, canavarlar ve araştırmalar ile ekstra güç kazanan birliklerin arasında gerçekleşir. Birlikler, kahramanlar, canavarlar ve araştırmalara ait bütün veriler **JSON** tipindeki dosyalardan okunmuştur. Gerçekleşen savaşın adımları “**savas_sim.txt**” isimli bir dosyaya yazdırılmıştır. Savaşın başlangıç ve bitiş durumları, **raylib** kütüphanesi kullanılarak 20x20 bir ızgarada görselleştirilmiştir. Senaryo verileri, **curl** kütüphanesi ile ilgili URL’lerden dinamik olarak indirilmiş ve birliklerin sayıları bu dosyalardan sağlanmıştır.

1. Giriş

Bu proje, iki tarafın savaşını modellemek için bir simülasyon geliştirmeyi amaçlamaktadır. Savaş öncesi ve sonrası durumlar, **raylib** kütüphanesi kullanılarak 20x20 boyutunda bir ızgarada görselleştirilmiş ve senaryo parametreleri dinamik olarak **curl** kütüphanesi ile çekilmiştir. Birliklerin güçleri **JSON** dosyalarından okunup struct yapılarına manuel olarak ayrıştırılmıştır. Savaş, adım adım hesaplanmıştır. Kahramanlar, canavarlar ve araştırma seviyeleri gibi faktörler, birliklerin performansını etkileyen unsurlar olarak oyuna dahil edilmiştir. Savaş adımları ise **TEXT** dosyasına detaylı olarak yazdırılmıştır.

2. Yöntem

2.1 Birliklerin Güçlerinin Ayırıştırılması

Birliklerin güçleri “*unit_tyes.json*”, “*heroes.json*”, “*creatures.json*” ve “*research.json*” dosyalarında belirtilmiştir. Proje, bu dosyaları “*fgets()*” ve “*sscanf()*” fonksiyonları ile okuyunmuştur. Bu verilerin daha sonra kullanılabilmesi için struct yapılarına kaydedilmiştir.

2.2 Senaryo Dosyalarının İndirilmesi (curl)

Simülasyon, farklı senaryoları desteklemek için **curl** kütüphanesini kullanarak URL’lerden JSON dosyaları indirir. Kodun her çalıştırılmasında kullanıcıdan bir senaryo numarası alınır ve bu numaraya karşılık gelen URL’den ilgili dosya indirilir.

Dosya indirildikten sonra, içerik **manuel olarak ayrıştırılarak** simülasyona dahil edilir.

2.3 Savaş Mekanikleri ve Hesaplamalar

Saldırı gücü şu formülle hesaplanmıştır:

$$\text{Saldırı Gücü} = \text{Birlik Başına Saldırı Gücü} \times \text{Birlik Sayısı}$$

Saldırı gücü şu formülle hesaplanmıştır:

$$\text{Savunma Gücü} = \text{Birlik Başına Savunma Gücü} \times \text{Birlik Sayısı}$$

Net hasar şu formülle hesaplanmıştır:

$$\text{Net Hasar} = \text{Saldırı Gücü} (1 - (\text{Savunma Gücü} / \text{Karşı Birliğin Saldırı Gücü}))$$

Saldırı sonrası yeni sağlık şu formülle hesaplanır:

$$\text{Yeni Sağlık} = \text{Sağlık} \times (\text{Net Hasar} / \text{Birlik Sayısı})$$

Bir birliğin sağlık durumu sıfıra ulaştığında, o birlik tamamen yok olmuş sayılır ve savaşa devam edemez. Sağlığı sıfırlanan birliklerin savaştaki etkisi sona erer ve savaş dışı kalırlar. Birlik sayısı ne olursa olsun, eğer sağlık durumu sıfıra inerse, o birlik savaş dışı kalmış kabul edilir.

Kritik vuruşun kaç vuruşta bir yapılacağı şu formülle hesaplanmıştır:

$$\text{Kritik Vuruş Sayısı} = 1 / \text{Kritik Vuruş Oranı}$$

Kritik vuruş bu formülle hesaplanmıştır:

$$\text{Kritik Vuruş} = \text{Net Hasar} \times 1.5$$

Yorgunluk, her 5 turda bir saldırı ve savunma güçlerini %10 oranında azaltır.

Yeni saldırı gücü şu formülle hesaplanır:

$$\text{Yeni Saldırı Gücü} = \text{Eski Saldırı Gücü} \times 0,9$$

Yeni savunma gücü şu formülle hesaplanır:

$$\text{Yeni Savunma Gücü} = \text{Eski Savunma Gücü} \times 0,9$$

2.4 20x20'lik Izgara Görselleştirme (raylib)

Savaşın başlangıç ve bitiş durumları, **raylib** kütüphanesi ile görselleştirilmiştir. Her bir hücrede yalnızca bir birlik türü bulunur ve hücre başına maksimum 100 birlik yerleştirilebilir. Eğer bir hücrede bu sınır aşılsa, fazla birlikler otomatik olarak bir sonraki boş hücreye aktarılır. Savaş sonrasında ise yok olan birlikler ızgaradan tamamen silinir. Birliklerin sağlıkları **raylib** kütüphanesi ile savaş öncesi ve sonrası olarak ayrı ayrı gösterilmiştir. Sağlık durumları renkli barlarla yansıtılmıştır:

- **Yeşil:** Sağlık %100 - %50
- **Sarı:** Sağlık %50 - %20
- **Kırmızı:** Sağlık %20 ve altı

3. Deneysel Sonuçlar

Her senaryoda ilk saldırıyı İnsan İmparatorluğu yapmıştır. Kahramanlar, canavarlar ve araştırmalar savaşın seyrini değiştirebilmiş, kazananı doğrudan etkilemiştir.

“savas_sim.txt” dosyasındaki çıktılarda her adımda saldıran birlikte toplam saldırı gücüne hangi birimin ne kadar katkı yaptığı, savunan birlikte ise toplam savunma gücüne hangi birimin ne kadar katkı yaptığı gösterilmiştir. Ek olarak saldıran birliğin toplam saldırı gücü, verdiği net hasar ve savunan birliğin toplam savunma değeri ile saldırı sonrası kalan sayı ve sağlık değerleri yazdırılmıştır.

1, 7, 8, 9 numaralı senaryolarda **Ork Lejyonu** kazanmıştır.

2, 3, 4, 5, 6, 10 numaralı senaryolarda ise **İnsan İmparatorluğu** kazanmıştır.

4. Sonuç

Bu proje, savaş mekaniklerini gerçekçi bir şekilde modelleyerek kahramanlar ve canavarlar gibi unsurların seyir değiştirecek kadar önemli olduğunu göstermiştir. **curl** kütüphanesi ile dinamik olarak senaryo dosyaları indirilebilmiştir. Savaş, adım adım doğru bir şekilde hesaplanabilmiştir. **raylib** kütüphanesi kullanılarak savaş öncesi ve sonrası durumlar başarılı bir şekilde görselleştirilmiştir. Ayrıca savaş adımları başarılı ve detaylı bir şekilde **TXT** dosyasına yazdırılmıştır. Araştırma seviyelerinin ve yorgunluk mekanizmasının savaş performansına etkileri net bir şekilde gözlemlenmiştir.

5. Yazar Katkıları

- *Ali Berke Erenler (230202069)*: Curl entegrasyonu ve JSON dosyalarının manuel olarak ayrıştırılmasını sağladı.
- *Ömer Faruk Toyçu (230202040)*: Raylib ile görselleştirme ve savaş mekaniklerini geliştirdi.

6. Kaynakça

- [1] Kocaeli Üniversitesi Proje Yönergesi, 2024.
- [2] Raylib Documentation, [Çevrimiçi]. Erişim: <https://www.raylib.com>.
- [3] Curl Documentation, [Çevrimiçi]. Erişim: <https://curl.se>.
- [4] JSON dosyaları: unit_types.json, heroes.json, creatures.json, research.json.