|  |
| --- |
| 172114203 & 152113001 |
| Unity3D Lightweight Render Pipeline ile MiniMax Algoritması kullanan Yapay Zeka ile Mirror Networking tabanlı çok oyunculu mobil ve bilgisayar destekli strateji oyunu |
| Düzce Üniversitesi 2020 Yaz Okulu Tez Projesi |
|  |
| **Özgür Özbek & Atilla Çoruhlu** |
| **1.01.2020** |
|  |

|  |
| --- |
| Tez projesi için sunulan en son geliştirmeleri ve teknolojileri kullanarak çok oyunculu bir oyun yapımı hakkında gerekenlerin temel olarak açıklanıp çeşitli sorunların nasıl çözüldüğünün açıklanması. |

**Kullanılan Teknolojiler, Araçlar ve Açıklamaları**

**Unity3D:** Bir oyun motoru. Oyun yapmak için gereken ses sürücüleri, ışık sekmesi takibi yazılımı, kod derleyici, ekrana çizdirme, sürücü takibi gibi bir çok yazılımı içerisinde barındıran bir motor yazılım. Bu dökümanda Unity olarak geçecek. Unity araçları ve metodları Türkçe’leş-tirilmeyecek ve asıl isimlerini koruyacak.

**Visual Studio Code & Monospace:** Kod yazmak için kullanılan çalışma ortamınına bağlı çalışan, içerilerinde olan Linter, Parser, Error Handling ve benzeri kod yazmayı kolaylaştıran yazılımlar ile bir entegre geliştirme ortamı.

**Adobe Illustrator CS6:** Oyun içinde kullanılacak iki boyutlu neredeyse her doku için (buton ikonları, basit yazılar vb.) bir dijital çizim uygulaması.

**Lightweight Render Pipeline:** Unity içinde sunulan son geliştirmelere sahip, ekran kartı ve işlemciye yük bindirmemek adına önceden yazılım desteği tamamlanmış bir görüntü işleme hattı. Bunun sayesinde Post-Processing, Shaders ve bir çok Rendering efekti kullanabiliyoruz. Bu dökümanda LWRP olarak kısaltılacak.

**MiniMax Algoritması:** Olabilecek bütün adımları bir iterasyona kadar deneyip en yüksek ve en düşük sonucu elde etmeye yarayan bir yapay zeka algoritması. Oyunda, rakip olmadığı durumlarda rakibin yerine tatmin edici bir sanal rakip oluşturmak için kullanılıyor.

**Yazılım:** Unity ile haberleşmek için C# ve Java, Dökümantasyon için Markdown, Yapay Zeka ile iletişim kurmak için C# ve Python, Networking için C++, Versiyon takibi için Git kullanılmakta.

**Oyun Teorisi:** Geliştirme aşamasında bir çok rekabetçi kararı almak için John F. Nash tara-fından ileri sürülmüş rasyonel kararları vermeye yardımcı matematiksel formüller kullanıldı.

**Steam & Google Play:** Oyun geliştirme, sürdür-me ve yayınlanması hakkında sunulan bir çok dökümandan faydalanıldı. Geliştirme hem mobil hem bilgisayar için yapılıp yayınlandı.

**Asset Store:** Çeşitli ses, model ve benzeri ürün elde edebilmek için online olarak editör içeri-sinde sunulan bir market.

**MS Office:** Sunum için dökümantasyon oluş-tururken ve kendi aramızda not takibi yaparken kullanıldı.

**Trello:** Proje takibi için bulletin board.

**Sahneler ve Açıklamaları**

**Menu Scene:** Oyuna ilk girdiğimizde karşımıza çıkan ekran. Buradan ses açıp kapatmak gibi çeşitli ayarları yapıp, olduğumuz versiyonu görüp, Mobil ise oyuna puan verebilip, oyun lobisi başlatma ekranına geçebildiğimiz veya oyundan çıkabildiğimiz ekran. Oyunun başlığı da bu ekranda yer alıyor.

**Lobby Scene:** Yeni bir oda kurabiliyor, kurulu odaya katılabiliyor, ve genel olarak Mirror kulla-narak multiplayer için sistem şartlarını sağlı-yoruz. Versiyonda eğer varsa sanal rakibe karşı offline lobi kurma imkanı sunuluyor.

**Game Scene:** Oyunun temel mantık ekranı.

**Credits Scene:** Oyun içerisinde kullanılan varsa açık kaynaklı yazılımların kaynaklarının gösteril-diği, yazarların isimlerinin verildiği, ve varsa su-nulması gereken teşekkürlerin sunulduğu ekran.

**Tutorial Scene:** Oyunun nasıl oynandığını anlatan bir ekran. Yazılar ve resimler ile des-teklenmiş bir Slide Show mantığı ile kullanıcıya basit ama gerekli bilgiler veriliyor.