



# Unity İle Hiper Gündelik Oyun Geliştirme

Yazılım Mühendisliği Ana Bilim Dalı  
Bitirme Projesi

Barış Çetin  
Y220240149

Proje Danışmanı : Dr. Öğr. Üyesi Serpil Yılmaz

Ağustos 2024

# Unity İle Hiper Gündelik Oyun Geliştirme

## ÖZ

Bu projenin amacı, kullanıcılara tekrar oynayabilecekleri, kolay öğrenebilecekleri bir oyun sunmaktır. Hiper-gündelik oyunlar kısa sürede öğrenebilen ve kolayca oynanabilen oyunlardır. Bu projede, Unity oyun motoru ile tasarlanmış ve geliştirilmiş bir oyunun teknik açıdan ve diğer yönlerden tanıtımı hedeflenmektedir.

Projenin bir diğer amacı ise, geliştirme sürecinde edinilen bilgi ve becerilerin özellikle oyun mekaniği tasarımı, kullanıcı arayüzü geliştirme, performans optimizasyonu ve mobil platformlara yönelik oyun yayını gibi alanlarda pekiştirilmesi ve uygulanmasıdır. Bu kapsamda, ideal bir oyun deneyimi sunulması amaçlanmıştır.

**Anahtar Sözcükler:** Hiper gündelik oyun, kullanıcı deneyimi, oyun mekanikleri, mobil oyun geliştirme, performans optimizasyonu

# Hyper Casual Game Development with Unity

## Abstract

The aim of this project is to provide users with a game that they can play again and learn easily. Hyper-casual games are games that can be learned and played easily in a short time. This project aims to present the technical and other aspects of a game designed and developed with the Unity game engine.

Another aim of the project is to consolidate and apply the knowledge and skills acquired during the development process, especially in areas such as game mechanics design, user interface development, performance optimization and game publishing for mobile platforms. In this context, it is aimed to provide an ideal game experience

**Keywords:** Hyper casual gaming, user experience, game mechanics, mobile game development, performance optimization

# İçindekiler

Öz.....	i
Abstract.....	ii
Şekiller Listesi.....	v
Giriş.....	1
Oyun.....	2
2.1 Oyun Kavramı .....	2
2.2 Oyun Kavramı .....	4
2.3 Oyunun Tarihçesi .....	5
2.4 Mobil Oyunlar .....	7
Oyun Sanatı.....	8
3.1 Oyun Geliştirme .....	8
3.1.1 Oyun Programlama .....	10
3.1.2 Oyun Tasarımı .....	12
3.2 Oyun Motorları.....	13
3.2.1 Unity .....	14
3.2.2 Unreal Engine .....	15
3.2.3 Godot .....	17
Günelik Türünde Oyun Projesi .....	18
4.1 Oyun Projesi Hakkında.....	18
4.2 Oyun Projesinin Tasarımı.....	19
4.2.1 Arkaplan Tasarımı .....	20
4.2.2 Karakter Tasarımı .....	21
4.2.3 Sahne Tasarımı .....	21
4.2.3 Post-Process Ayarları.....	22
4.3 Oyun Projesinin Arayüz Tasarımı .....	23
4.3.1 Giriş Ekranının Tasarımı .....	23
4.3.2 Puan Panelinin Tasarımı .....	23
4.3.3 Oyunun Tekrar Başlama Panelinin Tasarımı.....	24
4.4 Projenin Kodlanması .....	24
4.4.1 Karakter Kontrolü .....	24
4.5 Ses Efektleri .....	28

4.5.1 Zıplama Efektı Sesi.....	28
4.5.2 Süre Bitiş Efektı Sesi .....	28
4.5.3 Arka Plan Müziđi Sesi .....	29
4.6 Çalıştırılabilir Çıktı Alınması .....	30
4.6.1 Android Çıktı .....	30
4.6.2 Windows Çıktı .....	31
Sonuç.....	32
Kaynaklar .....	35

# Şekiller Listesi

Şekil 1 : Oyun Arkaplan Tasarımı .....	21
Şekil 2 : Karakter Tasarımı.....	21
Şekil 3 : Sahne Tasarımı.....	22
Şekil 4 : Post Process Ayarları.....	22
Şekil 5 : Giriş Ekranı Tasarımı .....	23
Şekil 6 : Puan Paneli .....	23
Şekil 7 : Oyuna Yeniden Başlama Paneli .....	24
Şekil 8 : Karakter Kontrol Kodları.....	26
Şekil 9 : Zıplama Sesi .....	28
Şekil 10 : Doğru Sesi .....	29
Şekil 11 : Arka plan Müziği .....	29
Şekil 12 : Android Çıktı Alınması.....	30
Şekil 13 : Windows Çıktı Alınması.....	32



# Bölüm 1

## Giriş

Mobil oyun sektörü, son yıllarda hızla büyüyen ve geniş kitlelere hitap eden bir alan haline gelmiştir. Özellikle hiper gündelik oyunlar, basit oynanış mekanikleri ve kısa sürede öğrenilebilir yapıları sayesinde geniş bir oyuncu kitlesi tarafından tercih edilmektedir. Bu oyun türleri, karmaşık hikaye veya derin stratejilerden ziyade, oyunculara kısa süreliğine eğlenceli bir deneyim sunmayı hedefler. Bu durum, hiper gündelik oyunları, günlük hayatın stresi ve yoğunluğundan uzaklaşmak isteyen kullanıcılar için ideal bir eğlence aracı haline getirmiştir.

Bu proje, hiper gündelik oyun türünün temel özelliklerini ve popülaritesini göz önünde bulundurarak, basit fakat bağımlılık yapıcı bir oyun geliştirmeyi amaçlamaktadır. Proje kapsamında, Unity oyun motoru kullanılarak geliştirilen oyun, minimal tasarımı ve kullanıcı dostu arayüzü ile oyunculara eğlenceli bir deneyim sunmayı hedeflemektedir.

Aynı zamanda bu proje, oyun geliştirme sürecinde kullanılan teknolojiler ve metodolojiler hakkında derinlemesine bir bilgi edinme ve bu bilgileri uygulamaya koyma fırsatı sunmaktadır. Oyun mekaniği tasarımı, kullanıcı arayüzü geliştirme, performans optimizasyonu ve mobil platformlara yönelik oyun yayını gibi önemli konular, proje süresince ele alınmış ve bu alanlarda önemli deneyimler kazanılmıştır.

Sonuç olarak, bu proje, hem oyunculara keyifli bir oyun deneyimi sunmayı hem de oyun geliştirme alanında pratik ve teorik bilgi birikimi sağlamayı amaçlamaktadır. Bu kapsamda, proje boyunca edinilen bilgi ve beceriler, hiper gündelik oyunlar için ideal bir oyun yapısı ve deneyimi oluşturmak için kullanılmıştır.

Oyuncular, karmaşık kurallarla ve uzun öğreticilerle uğraşmak zorunda kalmadan, doğrudan oyunun içine girebilir ve kısa sürede eğlenceye ulaşabilirler. Bu tür oyunlar, genellikle kısa süreli oturumlarla oynanmak üzere tasarlanmıştır ve oyuncuların tekrar tekrar geri dönmesini sağlayacak bir bağımlılık yaratır.



Bu proje, hiper gndelik oyunların bu temel zelliklerinden ilham alarak geliřtirilmiřtir. Amacımız, basit ama etkili bir oyun deneyimi sunarak, geniř bir kitleye hitap eden bir mobil oyun ortaya ıkarmaktır. Oyunun tasarım srecinde, kullanıcı dostu arayzler, sezgisel kontroller ve minimal bir grafik tasarımı n planda tutarak, oyunculara rahatlatıcı ve eęlenceli bir deneyim sunmayı hedefliyoruz.

Bunun yanı sıra, bu proje, oyun geliřtirme srecinin eřitli ařamalarında karřılařılan teknik zorlukları ařma konusunda da nemli bir ęrenme fırsatı saęlamıřtır. Proje boyunca, oyun mekanięi tasarımı, kullanıcı arayz geliřtirme, performans optimizasyonu ve mobil platformlar iin oyun yayını gibi kritik alanlarda bilgi ve beceri kazanılmıřtır. Ayrıca, bu srete edinilen deneyimler, gelecekteki oyun geliřtirme projelerinde de kullanılabilir deęerli bir bilgi birikimi oluřturmuřtur.

Sonuç olarak, bu proje, sadece oyunculara keyifli bir oyun deneyimi sunmakla kalmayıp, aynı zamanda oyun geliřtirme srecinde edinilen bilgilerin pekiřtirilmesi ve uygulanması aısından da nemli bir adım olmuřtur. Proje kapsamında geliřtirilen oyun, hiper gndelik oyunlar iin ideal bir model oluřturarak, bu tr oyunların temel ilkelerini bařarılı bir Őekilde hayata geirmeyi amalamaktadır.

## Blm 2

### Oyun

Bu blmde oyun geliřtirme ile ilgili detaylandırılmıř kısım bulunmaktadır. Bunlar bařlıklar halinde verilecektir.

#### 2.1 Oyun Kavramı

Oyun, insanlık tarihi boyunca kltrel, sosyal ve eęitsel aıdan nemli bir olgu olarak var olmuřtur. Genel anlamda oyun, belirli kurallar erevesinde oynanan, katılımcılara eęlence, rekabet, iřbirlięi ve ęrenme gibi deneyimler sunan bir etkinlik olarak tanımlanabilir. Oyunlar, insanların fiziksel, zihinsel ve duygusal yeteneklerini geliřtirmek, sosyal baęlar kurmak ve yaratıcı dřncelerini ifade etmek iin kullandıkları bir aratır.

Oyun kavramı, zaman ve mekan açısından sınırlandırılmamış, çok çeşitli biçimlerde ortaya çıkabilen bir yapıya sahiptir. Tarihsel olarak, oyunlar hem yetişkinler hem de çocuklar için toplumsal bir etkileşim aracı olarak kullanılmıştır. Geleneksel oyunlar, fiziksel etkinliklerden zeka oyunlarına, strateji oyunlarına kadar geniş bir yelpazeyi kapsar. Bununla birlikte, dijital devrimle birlikte oyunlar, sanal ortamlara taşınarak yeni bir boyut kazanmıştır.

Dijital oyunlar, bilgisayarlar, konsollar, akıllı telefonlar ve diğer elektronik cihazlar aracılığıyla oynanan oyunlardır. Bu tür oyunlar, etkileşimli ortamlar sunarak, oyunculara görsel ve işitsel deneyimler yaşatır. Dijital oyunlar, oyun kavramını genişletmiş, oyun tasarımında yeni yaklaşımlar ve mekanikler geliştirilmesine olanak tanımıştır. Bu oyunlar, yalnızca eğlence amacıyla değil, aynı zamanda eğitim, terapi, problem çözme ve sosyalleşme gibi amaçlarla da kullanılmaktadır.

Oyun kavramı, bireylerin öğrenme süreçlerine de katkıda bulunur. Oyunlar, oyuncuların zorluklarla başa çıkma, strateji geliştirme, takım çalışması yapma ve hızlı karar verme yeteneklerini geliştirmelerine yardımcı olur. Aynı zamanda, oyunlar sosyal bağları güçlendirebilir, çünkü insanlar oyunlar aracılığıyla ortak bir amaç doğrultusunda bir araya gelirler. Bu sosyal etkileşim, topluluklar arasında bağları kuvvetlendirir ve insanlar arasında işbirliğini teşvik eder.

Günümüzde oyunlar, mobil platformlar sayesinde daha da erişilebilir hale gelmiştir. Akıllı telefonlar ve tabletler, kullanıcıların her yerde oyun oynamasına olanak tanır. Mobil oyunlar, basit kontrolleri ve hızlı tempolu yapıları sayesinde geniş kitlelere hitap eder. Özellikle hiper gündelik oyunlar, kısa sürede öğrenilebilen ve oynanabilen, herkesin kolayca erişebileceği oyunlar olarak öne çıkmaktadır.

Özetle, oyun kavramı, insan hayatında önemli bir yere sahiptir ve farklı dönemlerde, farklı biçimlerde varlığını sürdürmüştür. Günümüzde ise dijital teknolojilerin etkisiyle oyunlar, daha geniş bir kitleye ulaşarak, eğlencenin ötesinde bir araç haline gelmiştir. Eğitici, sosyal ve rekreatif yönleriyle oyunlar, hem bireylerin gelişimine katkı sağlar hem de toplumsal bağları güçlendirir (Bogost, I. 2011).

## 2.2 Dijital Oyunlar

Oyun, insanlık tarihinin en eski ve evrensel faaliyetlerinden biridir. İnsanlar, binlerce yıl boyunca oyunları hem eğlence hem de toplumsal etkileşim aracı olarak kullanmışlardır. Oyunların kökeni antik çağlara kadar uzanır; örneğin, Mısır'da oynanan Senet ve Mezopotamya'da bulunan Ur Kraliyet Oyunu, bilinen en eski masa oyunları arasındadır. Bu oyunlar, sadece eğlence amaçlı değil, aynı zamanda dini ritüeller ve sosyal bağları güçlendirme amacıyla da oynanmıştır.

Antik Yunan ve Roma dönemlerinde oyunlar, toplumun her kesiminde popülerdi. Tavla, damalar ve zar oyunları gibi basit ama stratejik oyunlar, hem bireyler arasındaki rekabeti hem de sosyal etkileşimi artıran önemli etkinliklerdi. Orta Çağ boyunca, satranç gibi oyunlar, askeri strateji ve zeka geliştirme aracı olarak önem kazandı ve soylular arasında yaygınlaştı. Satranç, Hindistan'da ortaya çıktıktan sonra hızla dünya geneline yayıldı ve zeka oyunlarının temel taşlarından biri haline geldi.

Rönesans dönemi, Avrupa'da oyunların kültürel ve sosyal önemini arttırdığı bir dönemdir. Kart oyunları ve masa oyunları, bu dönemde entelektüel çevreler arasında popüler hale geldi. Özellikle kart oyunları, strateji geliştirme ve rekabet unsurları nedeniyle büyük ilgi gördü. Sanayi Devrimi ile birlikte oyunlar, geniş kitlelere ulaşabilecek ticari ürünler haline geldi. Bu dönemde Monopoly gibi ikonik masa oyunları ortaya çıktı ve oyunlar, sosyal etkileşimi teşvik eden, eğlence dolu etkinlikler olarak yaygınlaştı.

Yirminci yüzyılın ortalarına gelindiğinde, oyun endüstrisi hızla büyümeye başladı. Çocuklar için oyuncaklar ve oyunlar yaygın olarak üretilirken, yetişkinler için strateji ve kart oyunları da popülerliğini korudu. Ancak, gerçek anlamda bir devrim, dijital teknolojilerin gelişimiyle yaşandı. 1970'lerde dijital oyunların ortaya çıkması, oyun kavramını tamamen yeni bir boyuta taşıdı. Spacewar! gibi ilk bilgisayar oyunları, bilim insanları ve mühendisler tarafından geliştirildi ve bu oyunlar, dijital eğlencenin temelini oluşturdu.

1980'ler, dijital oyunların altın çağı olarak bilinir. Arcade oyunları ve ev video oyun konsolları, kısa sürede dünya çapında büyük bir popülerlik kazandı. Pong, Pac-Man

ve Space Invaders gibi oyunlar, dijital oyun dünyasının simgeleri haline geldi ve bu oyunlar, hem çocuklar hem de yetişkinler arasında büyük bir ilgi gördü. Bu dönemde, oyunlar sadece eğlence değil, aynı zamanda kültürel bir fenomen haline geldi (Schell, J. 2019).

1990'lar, bilgisayar ve konsol oyunlarının daha da karmaşık hale geldiği ve grafik teknolojilerinin büyük ilerlemeler kaydettiği bir dönemdi. Super Mario Bros, The Legend of Zelda ve Final Fantasy gibi oyunlar, bu dönemin en bilinen ve en çok sevilen yapımları arasında yer aldı. Bu dönemde oyunlar, sadece bir eğlence aracı olmaktan çıkıp, geniş ve çeşitli bir endüstri haline geldi (Adams, E. 2014).

Özetle, oyunların tarihçesi, insanlık tarihi kadar eski ve zengin bir geçmişe sahip olup, kültürel ve sosyal yaşamın her döneminde önemli bir yer tutmuştur. Modern dijital çağda ise oyunlar, teknolojinin sunduğu olanaklarla daha da gelişmiş ve küresel bir eğlence endüstrisi haline gelmiştir.

## 2.3 Oyunun Tarihçesi

Video oyunları, modern dijital çağın en önemli eğlence araçlarından biri haline gelmiştir. İlk olarak 1950'lerde ve 1960'larda bilim insanları tarafından geliştirilen basit bilgisayar oyunları ile başlayan bu serüven, bugün milyarlarca dolarlık bir endüstriye dönüşmüştür. Video oyunları, teknolojinin gelişmesiyle birlikte hızla evrim geçirmiş ve geniş bir kitleye hitap eden çeşitli türlerde oyunlar ortaya çıkmıştır.

Video oyunlarının kökeni, 1958 yılında William Higinbotham tarafından geliştirilen "Tennis for Two" adlı oyunla başlar. Bu oyun, bir osiloskop üzerinde oynanan basit bir tenis simülasyonuydu. 1962'de MIT'de Steve Russell tarafından geliştirilen "Spacewar!", ilk gerçek bilgisayar oyunu olarak kabul edilir. Bu oyun, üniversite laboratuvarlarında büyük ilgi gördü ve dijital oyunların temelini attı.

1970'ler, video oyunlarının ticari olarak başarı kazandığı dönem olarak bilinir. 1972 yılında, Atari şirketi, "Pong" adlı oyunu piyasaya sürdü. Pong, bir tenis oyununun basit

bir versiyonuydu ve büyük bir başarı elde etti, bu da video oyunlarının popülerleşmesinde önemli bir rol oynadı. Aynı dönemde, arcade oyunları da ortaya çıktı ve video oyunları, gençler arasında hızla popüler hale geldi.

1980'ler, video oyunlarının altın çağı olarak kabul edilir. Bu dönemde Pac-Man, Donkey Kong ve Space Invaders gibi ikonik arcade oyunları piyasaya sürüldü. Ayrıca, ev video oyun konsolları da bu dönemde büyük bir gelişme gösterdi. Nintendo'nun 1983'te piyasaya sürdüğü Nintendo Entertainment System (NES), video oyunlarını evlere taşıdı ve Super Mario Bros gibi oyunlarla oyun dünyasını derinden etkiledi.

1990'lar, video oyunlarının daha karmaşık hale geldiği ve üç boyutlu grafiklerin ortaya çıktığı bir dönemdi. Sony'nin PlayStation'ı ve Nintendo'nun Nintendo 64'ü, bu dönemde piyasaya sürüldü ve oyun dünyasına yeni bir soluk getirdi. Bu dönemde, Final Fantasy VII, The Legend of Zelda: Ocarina of Time ve Metal Gear Solid gibi oyunlar, oyun dünyasında devrim yarattı. Bu oyunlar, derin hikayeleri, karmaşık oyun mekanikleri ve çarpıcı görselleriyle oyuncuları büyüledi.

2000'ler ve sonrasında, video oyunları daha da gelişerek, çevrimiçi çok oyunculu oyunlar, mobil oyunlar ve sanal gerçeklik oyunları gibi yeni türler ortaya çıktı. World of Warcraft, Halo ve Grand Theft Auto gibi oyunlar, milyonlarca oyuncuyu bir araya getirerek büyük topluluklar oluşturdu. Aynı zamanda, bağımsız oyun geliştiricileri de bu dönemde yükselişe geçti ve Braid, Limbo ve Undertale gibi yenilikçi oyunlar geliştirdiler.

Bugün, video oyunları dünya genelinde milyarlarca insan tarafından oynanıyor. Oyunlar, sadece eğlence aracı olmaktan çıkıp, eğitim, terapi, spor ve sanat gibi birçok farklı alanda kullanılmaya başlandı. E-spor, profesyonel oyuncuların rekabet ettiği büyük bir sektör haline geldi ve oyun yayıncıları, milyonlarca izleyiciye ulaşan çevrimiçi platformlarda popülerlik kazandı.

Video oyunları, teknolojik ilerlemelerin bir sonucu olarak sürekli evrim geçiriyor ve sanal gerçeklik, artırılmış gerçeklik ve bulut oyunları gibi yeniliklerle geleceğe doğru ilerliyor. Bu, oyunların sadece eğlence için değil, aynı zamanda insan deneyiminin birçok yönünü zenginleştirmek için nasıl kullanılabileceğini gösteriyor. Gelişen grafik

teknolojileri, yapay zeka ve daha güçlü oyun motorları, video oyunlarının sınırlarını sürekli olarak genişletiyor ve oyunculara daha önce hiç yaşamadıkları türde deneyimler sunuyor.

## 2.4 Mobil Oyunlar

Mobil oyunlar, modern oyun endüstrisinin en hızlı büyüyen ve en etkili alanlarından biri haline gelmiştir. Akıllı telefonlar ve tabletlerin yaygınlaşmasıyla birlikte, oyun oynama alışkanlıkları da köklü bir değişim geçirmiştir. Mobil oyunlar, her yaştan ve her kesimden insana hitap eden, taşınabilirliği ve erişilebilirliği sayesinde kısa sürede büyük bir popülerite kazanmıştır.

Mobil oyunların tarihçesi, 1990'ların sonunda başlar. İlk mobil oyunlardan biri, 1997 yılında Nokia'nın 6110 model telefonunda yüklü olarak gelen Snake oyunuydu. Snake, basitliği ve bağımlılık yapıcı oynanışı ile mobil oyunların potansiyelini ortaya koydu ve dünya çapında milyonlarca insan tarafından oynandı. Bu dönemde mobil oyunlar, temel olarak basit grafikler ve sınırlı etkileşim sunan oyunlardı.

2000'li yılların başlarında, mobil teknolojilerin gelişmesiyle birlikte oyunlar da daha karmaşık ve çeşitli hale geldi. Java ve BREW platformları üzerinde geliştirilen oyunlar, renkli grafikler ve daha zengin içeriklerle mobil cihazlara taşındı. Ancak, mobil oyunlarda asıl büyük devrim, 2007 yılında Apple'ın iPhone'u piyasaya sürmesi ve App Store'un açılmasıyla başladı. iPhone, dokunmatik ekranı ve güçlü donanımı sayesinde mobil oyunların potansiyelini katladı. Angry Birds, Fruit Ninja ve Temple Run gibi oyunlar bu dönemde dünya çapında fenomen haline geldi (Nystrom, R. 2014).

App Store ve daha sonra Google Play Store'un açılmasıyla, mobil oyunlar küresel bir pazar haline geldi. Bağımsız geliştiriciler, büyük yatırımlar gerektirmeden oyunlarını milyonlarca kullanıcıya ulaştırma fırsatı buldu. Bu, mobil oyunların çeşitlenmesini sağladı ve çok sayıda yenilikçi oyun piyasaya sürüldü. Freemium modeli, yani ücretsiz indirilen ancak oyun içi satın alımlar içeren oyunlar, bu dönemde yaygınlaştı ve mobil oyun endüstrisini şekillendirdi.

Mobil oyunlar, basit bulmaca oyunlarından karmaşık strateji oyunlarına, aksiyon oyunlarından RPG'lere kadar geniş bir yelpazeye sahiptir. Ayrıca, mobil oyunlar sosyal medyanın ve çevrimiçi platformların entegrasyonu ile, oyuncuların birbirleriyle etkileşime geçebileceği sosyal oyunlar haline geldi. Candy Crush Saga, Clash of Clans ve Pokémon GO gibi oyunlar, mobil oyunların kitlesel popülerliğini artırdı ve oyunculara sosyal bir deneyim sundu.

Bugün, mobil oyunlar sadece eğlence için değil, aynı zamanda eğitim, sağlık ve hatta iş dünyasında kullanılmaktadır. Örneğin, bazı oyunlar çocukların eğitiminde kullanılırken, sağlık uygulamaları kullanıcıların fiziksel aktivitelerini takip etmelerine yardımcı olur. Ayrıca, artırılmış gerçeklik (AR) teknolojisi, mobil oyunların sınırlarını genişleterek, gerçek dünya ile sanal dünyayı birleştiren yeni oyun deneyimleri sunmaktadır. Pokémon GO, bu teknolojinin en başarılı örneklerinden biridir.

Mobil oyunlar, her geçen gün daha da gelişmekte ve oyun endüstrisinin önemli bir parçası olmaya devam etmektedir. Gelişen mobil cihaz teknolojisi, daha karmaşık ve görsel olarak çarpıcı oyunların geliştirilmesine olanak tanıırken, 5G teknolojisi gibi yenilikler, mobil oyunların geleceğini daha da parlak hale getirmektedir.

## Bölüm 3

### Oyun Sanatı

Bu bölümde oyun geliştirme ile ilgili bilgiler verilmiş, Oyun geliştirme süreçlerinden ve oyun motorlarından bahsedilmiştir.

#### 3.1 Oyun Geliştirme

Oyun geliştirme, bir oyunun konsept aşamasından nihai ürüne kadar olan süreçleri kapsayan karmaşık ve yaratıcı bir süreçtir. Bu süreç, bir fikrin oyun haline getirilmesi, oyun mekaniklerinin tasarımı, grafik ve seslerin oluşturulması, kodlama, test etme ve son olarak oyunun dağıtımını gibi birçok aşamayı içerir. Oyun geliştirme, disiplinler arası bir çalışma gerektirir; sanatçılar, yazılımcılar, tasarımcılar, ses mühendisleri ve

proje yöneticileri gibi farklı uzmanlık alanlarına sahip kişilerden oluşan ekiplerin işbirliğini içerir.

Geliştirmenin ilk aşaması, oyun fikrinin belirlenmesidir. Bu, genellikle yaratıcı bir sürecin sonucudur ve oyunculara ne tür bir deneyim sunulmak istendiği üzerine yoğunlaşır. Fikir aşaması, oyunun temasını, hikayesini, oynanış mekaniklerini ve hedef kitlesini belirlemeyi içerir. Bu aşamada, oyunun türü (örneğin, aksiyon, macera, strateji, bulmaca) ve oynanış tarzı (örneğin, tek oyunculu, çok oyunculu, çevrimiçi) gibi önemli kararlar alınır.

Bir sonraki aşama, oyun tasarımıdır. Bu aşamada, oyunun temel mekanikleri, seviye tasarımı, karakterler, hikaye akışı ve kullanıcı arayüzü gibi unsurlar detaylandırılır. Oyun tasarımcıları, oyuncuların oyunla nasıl etkileşime geçeceğini, hangi zorlukların ve ödüllerin olacağını ve oyunun genel yapısını belirler. Bu aşamada, oyun taslağı veya "game design document" adı verilen bir belge hazırlanır. Bu belge, oyun geliştiricileri için bir rehber niteliği taşır ve oyunun tüm yönlerini kapsar.

Oyun tasarımından sonra, geliştirme sürecinin teknik kısmı başlar. Programcılar, oyun motoru kullanarak oyunun kodunu yazar. Oyun motoru, grafiklerin işlenmesi, fizik simülasyonları, yapay zeka ve oyun mekaniği gibi temel bileşenleri yöneten yazılımdır. Unity, Unreal Engine ve Godot gibi popüler oyun motorları, geliştiricilerin bu süreçleri daha verimli bir şekilde yönetmelerine yardımcı olur. Programlama aşaması, oyunun temel işlevselliğini, performansını ve platform uyumluluğunu sağlamaya odaklanır.

Grafik ve ses tasarımı, oyunun görsel ve işitsel unsurlarını oluşturur. Grafik tasarımcıları, karakter modelleri, ortamlar, animasyonlar ve kullanıcı arayüzü gibi görsel bileşenleri oluşturur. Bu aşamada, oyunun estetik tarzı ve sanat yönü belirginleşir. Ses mühendisleri ise müzik, ses efektleri ve karakter seslendirmeleri gibi işitsel unsurları oluşturur. Ses, oyuncuların oyun dünyasına daha derinlemesine dalmalarına yardımcı olan önemli bir bileşendir.

Oyun geliştirme sürecinde, test etme aşaması kritik öneme sahiptir. Bu aşamada, oyun, olası hatalar ve performans sorunları açısından test edilir. Beta testleri, gerçek



oyuncular tarafından yapılan testlerdir ve oyunun son kullanıcı deneyimini değerlendirmek için kullanılır. Testler sonucunda, geliştiriciler oyunun hatalarını giderir, oyun dengesini ayarlar ve genel kullanıcı deneyimini iyileştirir.

Son aşama, oyunun piyasaya sürülmesi ve dağıtımıdır. Bu aşamada, oyunun yayınlanacağı platformlar (örneğin, Steam, PlayStation, Xbox, mobil uygulama mağazaları) belirlenir ve pazarlama çalışmaları başlatılır. Pazarlama, oyunun hedef kitlesine ulaşmasını sağlamak için kritik öneme sahiptir ve sosyal medya, fragmanlar, demo sürümler ve basın bültenleri gibi çeşitli araçlar kullanılarak gerçekleştirilir.

Oyun geliştirme süreci, oyunun piyasaya sürülmesinden sonra da devam edebilir. Güncellemeler, ek içerikler ve hata düzeltmeleri gibi işler, oyunun ömrünü uzatabilir ve oyuncu memnuniyetini artırabilir. Geliştiriciler, topluluk geri bildirimlerini dikkate alarak oyunu sürekli olarak iyileştirebilir ve genişletebilir.

Genel olarak, oyun geliştirme, yaratıcılık, teknik beceri ve işbirliği gerektiren karmaşık bir süreçtir. Bu süreç, bir fikri interaktif ve eğlenceli bir deneyime dönüştürmek için birçok disiplini bir araya getirir. Sonuç olarak, başarılı bir oyun, sadece teknik olarak iyi yapılmış bir ürün değil, aynı zamanda oyunculara unutulmaz bir deneyim sunan bir sanat eseri olarak kabul edilir.

### 3.1.1 Oyun Programlama

Oyun programlama, oyun geliştirme sürecinin temelini oluşturan teknik bir disiplindir ve yazılım geliştirme süreçlerinin oyun dünyasına uyarlanmış halidir. Oyun programcıları, oyun dünyasını inşa etmek, oyunun kurallarını belirlemek, oyuncu etkileşimlerini yönetmek ve oyunun düzgün çalışmasını sağlamak için gerekli olan kodları yazarlar. Bu süreç, birçok farklı teknik alanı kapsar ve oyun motorlarıyla doğrudan etkileşimli bir şekilde çalışmayı gerektirir.

Oyun programlamanın temel görevlerinden biri, oyun mekaniklerini oluşturmaktır. Oyun mekanikleri, bir oyunun nasıl oynandığını ve oyuncuların oyun dünyasıyla nasıl etkileşime girdiğini belirleyen kurallardır. Programcılar, oyuncunun karakteri nasıl kontrol ettiğini, oyunun zorluk seviyesinin nasıl arttığını ve oyundaki görevlerin nasıl

tamamlandığını kodlar. Bu süreç, genellikle oyunun tasarımcılarıyla yakın işbirliği içinde çalışmayı gerektirir, çünkü tasarımcılar oyunun genel yapısını ve hissini belirlerken, programcılar bu vizyonu hayata geçirir.

Fizik motorları ve simülasyonlar, oyun programlamasının bir diğer kritik alanıdır. Gerçekçi hareket, çarpışma ve diğer fiziksel olayların simülasyonu, özellikle 3D oyunlarda büyük önem taşır. Programcılar, fizik motorlarını kullanarak nesnelerin nasıl hareket edeceğini, yer çekimi ve sürtünme gibi kuvvetlerin nasıl uygulanacağını ve çarpışmaların nasıl yönetileceğini belirler. Bu tür detaylar, oyunun gerçekçiliğini ve oyuncunun deneyimini doğrudan etkiler.

Yapay zeka (AI) programlaması da oyun programlamanın önemli bir parçasıdır. Yapay zeka, oyun dünyasında NPC'lerin (oyuncu olmayan karakterler) davranışlarını yönetir. Bu karakterlerin oyuncuya nasıl tepki vereceği, kendi aralarındaki etkileşimler ve çevreye nasıl uyum sağlayacakları gibi unsurlar, yapay zeka algoritmalarıyla belirlenir. İyi tasarlanmış bir yapay zeka, oyunun daha dinamik ve zorlu hale gelmesini sağlar.

Ayrıca, oyun programcıları, grafik ve ses motorlarını entegre ederek oyunun görsel ve işitsel bileşenlerini de yönetirler. Grafik motorları, oyunun nasıl görüneceğini belirler; bu, ışıklandırma, gölgelendirme, animasyonlar ve efektler gibi unsurları içerir. Ses motorları ise müzik, ses efektleri ve karakter seslendirmelerinin nasıl oynatılacağını yönetir. Bu unsurlar, oyunun atmosferini yaratmak ve oyuncuyu oyunun dünyasına daha fazla çekmek için kritik öneme sahiptir.

Çok oyunculu oyunlarda, ağ programlaması da önemli bir rol oynar. Programcılar, oyuncuların internet üzerinden birbirleriyle nasıl etkileşime gireceğini ve oyun verilerinin nasıl senkronize edileceğini yönetirler. Bu süreç, düşük gecikme süreleri ve güvenilir bağlantılar sağlamak için optimize edilmelidir.

Son olarak, oyun programlama sürecinde, performans optimizasyonu ve hata ayıklama da kritik görevler arasında yer alır. Oyunların sorunsuz bir şekilde çalışabilmesi için kodun verimli olması ve sistem kaynaklarını etkili bir şekilde kullanması gerekir. Programcılar, performans sorunlarını tespit etmek ve çözmek için çeşitli testler ve

profilleme araçları kullanır. Hata ayıklama süreci ise, oyun içindeki hataların bulunması ve düzeltilmesi için kapsamlı bir şekilde yürütülür.

Özetle, oyun programlama, bir oyunun teknik altyapısını oluşturan ve oyun deneyiminin temel unsurlarını belirleyen karmaşık bir süreçtir. Programcılar, oyun tasarımını kodlara dönüştürerek, oyun dünyasını hayata geçirir ve oyunculara zengin, etkileşimli bir deneyim sunar.

### 3.1.2 Oyun Tasarımı

Oyun tasarımı, bir oyunun görsel ve işlevsel yapısını belirleyen süreçtir ve genellikle 3D ve 2D olarak iki ana kategoriye ayrılır. 3D oyun tasarımı, oyunun üç boyutlu bir ortamda geçmesini sağlar ve oyuncuya derinlik hissi verir. 2D oyun tasarımı ise iki boyutlu bir düzlemde gerçekleşir ve daha basit, düzlemsel grafikler kullanır. Her iki tasarım türü de kendine özgü avantajlar ve zorluklar sunar ve oyunun genel estetiği, oynanış mekanikleri ve kullanıcı deneyimi üzerinde büyük bir etkiye sahiptir.

3D oyun tasarımı, genellikle daha karmaşık ve gerçekçi bir oyun dünyası yaratmayı hedefler. Bu tür oyunlarda, oyunculara bir perspektif sunulur ve oyuncu karakteri, çevresindeki dünya ile etkileşime girerken derinlik algısını deneyimler. 3D oyunlarda, oyuncular farklı açılardan bakış açıları değiştirebilir ve dünya içinde serbestçe hareket edebilir. Bu tür bir tasarım, oyunculara daha sürükleyici bir deneyim sunar çünkü çevre detaylı bir şekilde işlenir ve oyuncunun bakış açısına göre dinamik olarak değişir. 3D tasarım, karakter modelleme, animasyon, fizik simülasyonu ve karmaşık ışıklandırma gibi teknik gereksinimleri içerir. Oyun tasarımcıları, bu unsurları bir araya getirerek gerçekçi ve etkileyici bir oyun dünyası oluştururlar.

2D oyun tasarımı ise daha basit ve genellikle daha hızlı bir geliştirme süreci sunar. 2D oyunlar, iki boyutlu bir düzlemde hareket eder ve genellikle yan kaydırma (side-scrolling) veya yukarıdan aşağıya (top-down) perspektiflerle sunulur. 2D tasarımda, oyun dünyası daha sade olabilir, ancak bu basitlik, oyunun oynanış mekaniklerine ve estetik tasarımına daha fazla odaklanılmasına olanak tanır. 2D oyunlar, genellikle retro tarzı grafikler ve animasyonlar kullanır ve bu da nostalji hissi yaratabilir. 2D oyun

tasarımı, özellikle bağımsız geliştiriciler arasında popülerdir çünkü daha düşük maliyetle yüksek kaliteli oyunlar üretmek mümkündür.

Her iki tasarım türü de farklı türde oyunlar için uygundur ve oyunun hedef kitlesine, konseptine ve teknik gereksinimlerine bağlı olarak seçilir. 3D oyunlar, genellikle açık dünya oyunları, simülasyonlar ve gerçekçilik gerektiren oyun türleri için tercih edilir. 2D oyunlar ise bulmaca, platform ve strateji oyunları gibi daha basit ama etkili oynanışa odaklanan oyun türleri için idealdir.

Sonuç olarak, 3D ve 2D oyun tasarımı, oyun geliştirme sürecinde temel rol oynayan iki ana yaklaşımdır. Her iki yöntem de oyun dünyasının nasıl görüneceğini, hissedileceğini ve oynanacağını belirler. Tasarımcılar, bu iki yaklaşımı kullanarak, oyunculara unutulmaz deneyimler sunan oyunlar yaratabilirler. Hangi tasarım türünün kullanılacağı, oyunun konsepti, hedef kitlesi ve teknik imkanlarına bağlı olarak belirlenir ve her iki yaklaşım da kendi içinde benzersiz zorluklar ve fırsatlar sunar.

## 3.2 Oyun Motorları

Oyun motorları, oyun geliştirme sürecinde kullanılan temel araçlardır ve oyunların oluşturulmasını, test edilmesini ve dağıtılmasını kolaylaştıran yazılım platformlarıdır. Bir oyun motoru, grafik işleme, fizik simülasyonu, ses yönetimi, yapay zeka, ağ altyapısı ve daha birçok teknik bileşeni yönetmek için gerekli olan araçları sunar. Geliştiriciler, oyun motorlarını kullanarak oyunlarını sıfırdan yazmak yerine, bu motorların sunduğu hazır araçlar ve kütüphaneler sayesinde oyun geliştirme sürecini büyük ölçüde hızlandırabilir ve basitleştirebilir.

Oyun motorlarının en büyük avantajı, oyun geliştirme sürecini daha erişilebilir hale getirmeleridir. Bir oyun motoru, karmaşık kodlama ve teknik ayrıntılarla uğraşmadan, görsel bir arayüz üzerinden oyun tasarımı yapmayı mümkün kılar. Bu da, hem bağımsız geliştiriciler hem de büyük oyun stüdyoları için önemli bir zaman ve maliyet tasarrufu sağlar. Oyun motorları, genellikle çeşitli platformlar (PC, konsol, mobil, VR) için çapraz uyumluluk sunar, bu da aynı oyun projesinin farklı cihazlarda çalışabilmesini sağlar.

Bir oyun motorunun temel bileşenleri arasında, oyun sahnelerini ve nesnelerini yöneten bir sahne yönetim sistemi, grafiklerin ekranda işlenmesini sağlayan bir render motoru, oyun dünyasında fizik kurallarını simüle eden bir fizik motoru, ses efektleri ve müzikler için bir ses motoru, ve karakterler veya nesnelere arasındaki etkileşimleri yöneten yapay zeka sistemleri bulunur. Ayrıca, birçok oyun motoru, çok oyunculu oyunlar için ağ altyapısı ve oyun içi ekonomiler veya mağazalar için veri yönetim sistemleri gibi ek özellikler de sunar.

### 3.2.1 Unity

Unity, oyun geliştirme dünyasında en popüler ve yaygın olarak kullanılan oyun motorlarından biridir. İlk olarak 2005 yılında piyasaya sürülen Unity, özellikle esnekliği, kullanıcı dostu arayüzü ve geniş platform desteği ile tanınır. Hem bağımsız geliştiriciler hem de büyük oyun stüdyoları tarafından tercih edilen Unity, 2D ve 3D oyunlar geliştirmek için güçlü bir araçtır.

Unity'nin temel avantajlarından biri, çok sayıda platforma oyun çıkışını desteklemesidir. Unity ile geliştirilen oyunlar, PC, macOS, Linux, iOS, Android, PlayStation, Xbox, Nintendo Switch, VR/AR cihazları ve daha birçok platformda çalıştırılabilir. Bu, geliştiricilere tek bir kod tabanıyla geniş bir oyuncu kitlesine ulaşma imkanı sağlar.

Unity, C# programlama dili kullanarak geliştirme yapar ve geliştiricilere güçlü bir geliştirme ortamı sunar. Unity Editor, projelerin yönetilmesi, oyun sahnelerinin oluşturulması, oyun nesnelerinin yerleştirilmesi ve oyun mantığının kodlanması için kapsamlı araçlar sunar. Bu araçlar, hem görsel hem de programlama odaklı geliştirme süreçlerini destekler.

Unity'nin bir diğer önemli özelliği, geniş bir varlık mağazasına (Asset Store) sahip olmasıdır. Unity Asset Store, geliştiricilerin oyunlarına modellemeler, animasyonlar, sesler, grafikler, scriptler ve diğer birçok varlığı hızlı bir şekilde eklemelerine olanak tanır. Bu mağaza, geliştiricilerin projelerini hızla ilerletmelerini sağlar ve aynı zamanda maliyet ve zaman tasarrufu sunar.

Unity, sadece oyun geliştirme için değil, aynı zamanda simülasyonlar, mimari görselleştirmeler, animasyonlar ve etkileşimli deneyimler için de kullanılabilir. Bu, Unity'yi oyun dünyasının ötesinde geniş bir uygulama alanına sahip bir araç haline getirir (Goldstone, W. 2020).

Son olarak, Unity'nin güçlü bir topluluk desteği ve çevrimiçi kaynakları bulunur. Unity Forumları, Unity Learn platformu, YouTube dersleri ve sayısız blog yazısı, geliştiricilerin sorunlarını çözmelerine ve yeni beceriler kazanmalarına yardımcı olur. Unity ayrıca, sürekli güncellemeler ve yeni özellikler sunarak oyun geliştirme teknolojisinin ön saflarında kalmaya devam eder.

Özetle, Unity, esnekliği, geniş platform desteği ve güçlü geliştirme araçları ile oyun geliştiricileri için vazgeçilmez bir oyun motorudur. Hem küçük bağımsız projeler hem de büyük çaplı oyunlar için uygun olan Unity, dünya genelinde milyonlarca geliştirici tarafından kullanılmakta ve oyun endüstrisinde önemli bir rol oynamaktadır (Unity Technologies. 2021).

### 3.2.2 Unreal Engine

Unreal Engine, Epic Games tarafından geliştirilen ve oyun endüstrisinde geniş bir şekilde kullanılan güçlü bir oyun motorudur. İlk olarak 1998 yılında piyasaya sürülen Unreal Engine, özellikle yüksek kaliteli grafikler, gerçekçi fizik simülasyonları ve güçlü araçlarıyla tanınır. Bu motor, hem büyük oyun stüdyoları hem de bağımsız geliştiriciler tarafından tercih edilir ve AAA kalitesinde oyunlar geliştirmek için sıkça kullanılır.

Unreal Engine, birçok özelliği ile oyun geliştirme sürecini destekler. En bilinen özelliklerinden biri, motorun sunduğu yüksek performanslı render motorudur. Unreal Engine, fotogerçekçi grafikler ve detaylı çevreler oluşturmak için gelişmiş aydınlatma, gölgelendirme ve malzeme sistemleri sunar. Bu özellikler, özellikle oyunların görsel kalitesini en üst düzeye çıkarmak isteyen geliştiriciler için önemli bir avantaj sağlar.

Unreal Engine'in bir diğer güçlü yönü, fizik motorudur. Gerçekçi fizik simülasyonları, oyun dünyasında nesnelere hareketlerini, çarpışmalarını ve etkileşimlerini daha

inandırıcı hale getirir. Bu motor, karmaşık fizik hesaplamalarını yüksek performansla gerçekleştirir ve oyunlarda daha etkileyici ve dinamik bir oynanış sunar.

Unreal Engine, C++ programlama dili kullanılarak geliştirilir, ancak motorun bir diğer önemli özelliği Blueprint adlı görsel programlama sistemidir. Blueprint, kod yazmadan oyun mantığını ve etkileşimleri oluşturmayı sağlar. Bu, özellikle programlama konusunda deneyimi az olan geliştiriciler veya hızlı prototipleme yapmak isteyenler için büyük bir avantajdır. Blueprint ile karmaşık oyun mekanikleri ve etkileşimler, sürükle-bırak yöntemiyle kolayca tasarlanabilir.

Unreal Engine, geniş platform desteği ile bilinir. Geliştiriciler, Unreal Engine kullanarak oyunlarını PC, macOS, Linux, iOS, Android, PlayStation, Xbox, Nintendo Switch, VR/AR cihazları ve daha birçok platformda yayınlatabilirler. Bu esneklik, oyunların geniş bir oyuncu kitlesine ulaşmasını sağlar.

Unreal Engine ayrıca, güçlü bir topluluk ve geniş bir çevrimiçi kaynak ağına sahiptir. Epic Games, Unreal Engine kullanıcılarına kapsamlı bir destek sunar ve düzenli olarak güncellemeler, yeni özellikler ve iyileştirmeler sunar. Unreal Engine Marketplace, geliştiricilerin projelerine hazır varlıklar, eklentiler ve araçlar eklemelerine olanak tanır, bu da projelerin hızla ilerlemesini sağlar.

Son yıllarda Unreal Engine, sadece oyun geliştirme için değil, aynı zamanda film prodüksiyonu, mimari görselleştirme, simülasyonlar ve sanal gerçeklik projeleri için de yaygın olarak kullanılmaya başlanmıştır. Özellikle gerçek zamanlı render yetenekleri, film ve TV endüstrisinde devrim yaratmış ve sanal prodüksiyon süreçlerinde önemli bir rol oynamıştır.

Özetle, Unreal Engine, güçlü grafik motoru, gerçekçi fizik simülasyonları ve geniş platform desteği ile oyun geliştirme dünyasında öne çıkan bir oyun motorudur. Hem büyük çaplı projeler hem de bağımsız oyunlar için uygun olan Unreal Engine, sunduğu esneklik ve yüksek performansla geliştiricilere olağanüstü oyunlar yaratma imkanı tanır.

### 3.2.3 Godot

Godot, açık kaynaklı ve ücretsiz bir oyun motorudur ve özellikle bağımsız geliştiriciler arasında popülerlik kazanmıştır. İlk olarak 2014 yılında piyasaya sürülen Godot, 2D ve 3D oyun geliştirme için güçlü ve esnek bir platform sunar. Godot'un en dikkat çekici özelliklerinden biri, kullanıcılara tamamen ücretsiz bir geliştirme ortamı sağlaması ve açık kaynaklı olmasıdır. Bu sayede, geliştiriciler motorun kodlarını inceleyebilir, değiştirebilir ve ihtiyaçlarına göre özelleştirebilirler.

Godot, hem 2D hem de 3D oyun geliştirme için optimize edilmiş araçlar sunar. Diğer birçok oyun motorunun aksine, Godot 2D oyun geliştirmeye özel olarak odaklanmış ve bu tür oyunlar için son derece verimli bir ortam sağlamıştır. Godot'un 2D motoru, pikselleme, aydınlatma ve animasyon gibi konularda yüksek performans ve esneklik sunar. Aynı şekilde, 3D oyunlar için de güçlü araçlar ve özellikler içerir, ancak özellikle basit ve orta düzey 3D projeler için idealdir.

Godot'un programlama dili GDScript'tir. GDScript, Python'a benzeyen, öğrenmesi kolay ve oyun geliştirmeye odaklı bir betik dilidir. Bu dil, özellikle hızlı prototipleme ve oyun mantığı yazma sürecini kolaylaştırır. Bunun yanı sıra, Godot C#, VisualScript ve NativeScript gibi diğer programlama dillerini de destekler, bu da geliştiricilere esneklik sağlar.

Godot'un sahne sistemi, geliştiricilere projelerini modüler bir şekilde yapılandırma imkanı tanır. Her sahne, kendi içinde bağımsız birimlerden oluşur ve bu sahneler birleşerek oyunun tamamını oluşturur. Bu modüler yapı, yeniden kullanılabilirliği artırır ve projelerin daha organize bir şekilde yönetilmesini sağlar. Örneğin, bir karakter, bir sahne olarak tanımlanabilir ve bu karakterin birden fazla yerde kullanılması gerektiğinde, aynı sahne farklı yerlerde tekrar kullanılabilir.

Godot'un bir diğer önemli özelliği, entegre bir görsel düzenleyici sunmasıdır. Bu düzenleyici, sahneleri, animasyonları, kullanıcı arayüzlerini ve diğer oyun bileşenlerini görsel olarak tasarlamayı sağlar. Düzenleyici, kullanıcı dostu bir arayüze sahiptir ve geliştiricilerin kodlama yapmadan oyunlarını inşa etmelerine yardımcı olur.



Godot, apraz platform desteęi sunar. Geliřtiriciler, Godot ile oluřturdukları oyunları Windows, macOS, Linux, Android, iOS, HTML5 ve dięer birok platformda yayınlatabilirler. Bu, oyunların geniř bir oyuncu kitlesine ulařmasını saęlar. Ayrıca, Godot'un kk boyutlu bir motor olması, dřk sistem gereksinimlerine sahip bilgisayarlarda bile sorunsuz alıřmasını saęlar.

Godot'un topluluęu, motorun en bk glerinden biridir. Godot, geniř bir topluluk desteęine sahiptir ve geliřtiriciler, evrimii forumlar, dersler, belgeler ve rnek projeler aracılıęıyla yardım alabilirler. Topluluk, motorun srekli geliřimine katkıda bulunur ve dzenli olarak yeni zellikler ve iyileřtirmeler sunar.

Sonuç olarak, Godot, baęımsız geliřtiriciler ve kk stdyolar iin mkemmел bir seenek sunan, gl, esnek ve tamamen cretsiz bir oyun motorudur. Hem 2D hem de 3D oyunlar iin uygun olan Godot, aık kaynaklı yapısı, kullanıcı dostu araları ve geniř topluluk desteęi ile oyun geliřtirme dnyasında nemli bir yer edinmiřtir.

## Blm 4

# Gndelik Trnde Oyun Projesi

Bu blmde Unity ile gndelik trnde bir oyun geliřtirilmiř ve oyun projesi alt bařlıklara ayrılarak teknik, grsel ve oynayıř anlamında tanıtılmıřtır.

### 4.1 Oyun Projesi Hakkında

Bu projede, oyuncuların reflekslerini ve renk algılarını test eden basit fakat eęlenceli bir oyun geliřtirilmiřtir. Oyunun temel amacı, oyuncunun bir topu ekrandaki karolar arasında doęru bir řekilde ynlendirmesidir. Oyun alanı, yan yana dizilmiř dokuz karodan oluřmaktadır. Oyuncunun kontrol ettięi top, belirli aralıklarla rastgele bir renge brnmektedir. Aynı řekilde, karolar da belirli srelerde renk deęiřtirmektedir.

Oyuncunun grevi, topun rengini takip ederek, topun rengine en uygun karonun zerine atlamasını saęlamaktır. Oyuncu, doęru renkli karoya atlayabilmek iin topu

dikkatli bir şekilde yönlendirmelidir. Her başarılı atlama, oyuncuya puan kazandırmakta ve oyun ilerledikçe zorluk seviyesi artmaktadır; bu, renk değişimlerinin hızlanması ve karoların daha sık renk değiştirmesi gibi unsurları içermektedir.

Bu oyun, hem reflekslerin hem de görsel algıların hızlı bir şekilde test edildiği bir oyun deneyimi sunmaktadır. Oynanış mekanikleri basit olmasına rağmen, artan zorluk seviyesi ve hızla değişen renk kombinasyonları sayesinde oyunculara sürekli bir meydan okuma sunulmaktadır. Oyunun amacına ulaşmak için, oyuncuların hızlı düşünmesi ve doğru zamanda doğru kararları alması gerekmektedir. Bu unsurlar, oyunu hem eğlenceli hem de bağımlılık yapıcı hale getirmektedir.

## 4.2 Oyun Projesinin Tasarımı

Oyun projesinin tasarımı, oyunun estetik ve işlevsel yönlerini belirleyen kritik bir süreçtir. Bu süreç, arka plan tasarımı, kullanıcı arayüzü (UI) öğeleri, ses efektleri, animasyonlar ve genel görsel stil gibi unsurları kapsar. Oyunun tasarımı, oyuncunun oyunu nasıl algıladığını, oyuna nasıl etkileşimde bulunduğunu ve genel oyun deneyimini doğrudan etkiler.

Arka plan tasarımı, oyunun atmosferini oluşturan temel unsurlardan biridir. Arka plan, oyunculara oyunun dünyasını tanıtır ve oyunun temasını, hikayesini ve tonunu belirler. Bu projede, arka planın sade ve dikkat dağıtıcı olmayan bir yapıda olması amaçlanmıştır, böylece oyuncuların dikkatleri oyun mekaniklerine ve renk değişimlerine odaklanabilir. Arka planın renkleri ve dokusu, oyunun genel estetiğiyle uyumlu olacak şekilde seçilmiştir.

Kullanıcı arayüzü (UI) öğeleri, oyunun oyuncu ile olan etkileşimini sağlayan görsel bileşenlerdir. Bu öğeler arasında menüler, butonlar, skor göstergeleri, zamanlayıcılar ve diğer etkileşimli unsurlar yer alır. UI tasarımında, kullanıcı dostu ve anlaşılır bir yapı oluşturmak temel hedeftir. Bu projede, UI öğeleri minimalist bir yaklaşımla

tasarlanmıştır; basit, net ve işlevsel olmalarına özen gösterilmiştir. Bu, oyuncuların oyun sırasında UI öğeleriyle hızlı ve sorunsuz bir şekilde etkileşim kurmalarını sağlar.

Renk paleti, oyunun genel tasarımında önemli bir rol oynar. Renklerin dikkatli bir şekilde seçilmesi, hem görsel uyumu sağlar hem de oyuncuların oyunu daha rahat algılamalarına yardımcı olur. Bu projede, renk değişimi oyun mekaniğinin temelini oluşturduğu için, renklerin kontrastı ve geçişleri büyük bir titizlikle belirlenmiştir. Bu, oyuncuların hızlı kararlar alabilmesi ve doğru renkli karoya atlayabilmesi için kritik öneme sahiptir.

Ses efektleri ve müzik, oyunun atmosferini güçlendiren ve oyunculara daha derin bir oyun deneyimi sunan unsurlardır. Ses tasarımı, her bir hareket, başarı veya hata için uygun ses geri bildirimleri sağlayarak, oyuncunun oyuna daha fazla dalmasını sağlar. Bu projede, ses efektleri, topun hareketi, karoların renk değiştirmesi ve başarılı atlamalar gibi önemli eylemleri vurgulamak için kullanılmıştır. Müzik ise arka planda oynayarak, oyunun temposunu ve ruh halini desteklemektedir.

Animasyonlar, oyun dünyasının daha canlı ve dinamik görünmesini sağlayan bir diğer önemli tasarım öğesidir. Topun hareketi, karoların renk değiştirmesi ve UI öğelerinin tepkileri gibi animasyonlar, oyunun akıcılığını ve görsel çekiciliğini artırır. Bu projede, animasyonlar, oyuncunun gözünü yormadan, oyunun hızını ve akışını bozmadan etkili bir şekilde uygulanmıştır.

Sonuç olarak, oyun tasarımı, bir oyunun görsel ve işlevsel unsurlarının bir araya gelmesiyle oluşur. Arka plan, UI öğeleri, renk paleti, ses efektleri ve animasyonlar gibi bileşenler, oyun deneyiminin kalitesini doğrudan etkiler. Bu projede, bu unsurlar dikkatlice tasarlanmış ve uyum içinde çalışacak şekilde entegre edilmiştir, böylece oyunculara keyifli ve etkileyici bir oyun deneyimi sunulması hedeflenmiştir.

#### 4.2.1 Arkaplan Tasarımı

Bu görselde, arka plan olarak düşük poli (low poly) stilinde tasarlanmış bir kale yerleşkesi tercih edilmiştir. Bu stil, basit ve sade bir görsellik sunarak oyunun performansını optimize ederken, aynı zamanda göz alıcı bir ortam yaratmayı amaçlar.

Kale, yüksek kuleler ve kalın duvarlarla çevrili olup, etrafındaki dağlık ve yeşil alanlarla birlikte, oyunculara çekici ve işlevsel bir oyun dünyası sunar.



Şekil 1 : Oyun Arkaplan Tasarımı

#### 4.2.2 Karakter Tasarımı

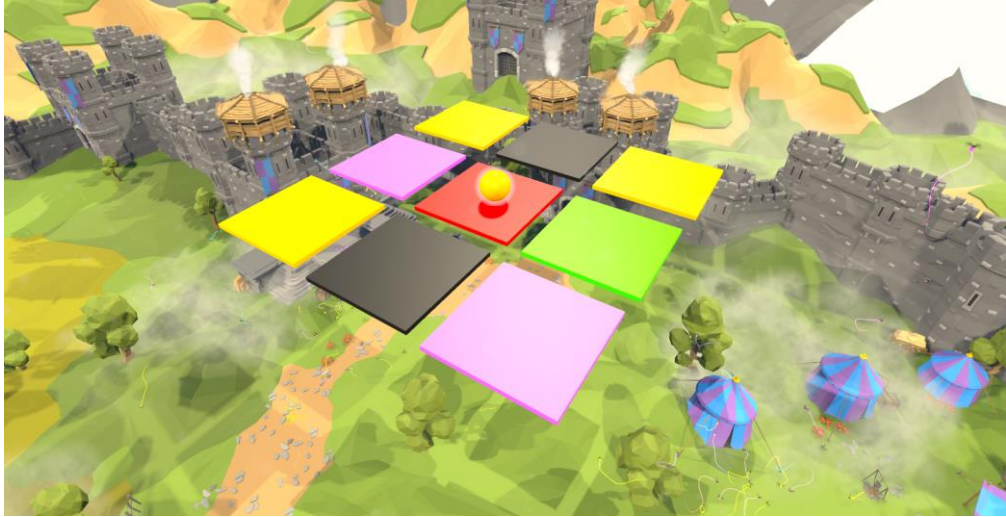
Bu projede, karakter olarak bir top tercih edilmiştir. Hem hareket kolaylığı, hem de kolay adaptasyon tercih edilmiştir.



Şekil 2 : Karakter Tasarımı

#### 4.2.3 Sahne Tasarımı

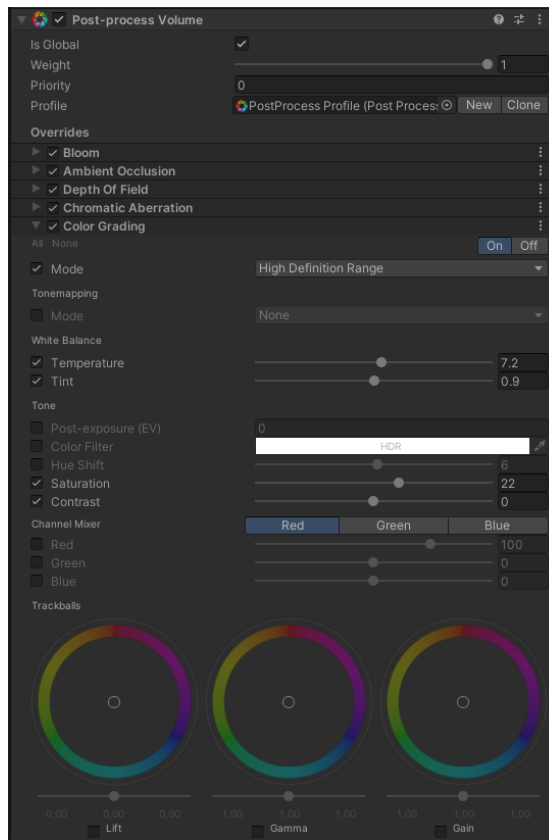
Bu projede sahne tasarımının ilgi çekici olması hedeflenmiş. Partekül efektleri, görsel öğeler, farklı renkler bolca kullanılmıştır



Şekil 3 : Sahne Tasarımı

### 4.2.3 Post-Process Ayarları

Bu projede renk ayarları ve diğer görsel öğeleri daha belirgin ve estetik yapmak için post-process kullanılmıştır.



Şekil 4 : Post Process Ayarları

## 4.3 Oyun Projesinin Arayüz Tasarımı

### 4.3.1 Giriş Ekranının Tasarımı

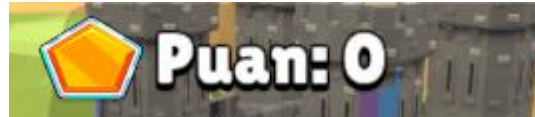
Bu projede kullanıcı oyuna eriştiğinde bir giriş ekranına ulaşması hedeflenmiş. Oyuna başlaması için bir buton eklenmiştir.



Şekil 5 : Giriş Ekranı Tasarımı

### 4.3.2 Puan Panelinin Tasarımı

Bu projede kullanıcının oyun içerisinde topladığı puanları görebilmesi için bir puan panelinin bulunması hedeflenmiş, buna göre aksiyon alınmıştır.



Şekil 6 : Puan Paneli

### 4.3.3 Oyunun Tekrar Başlama Panelinin Tasarımı

Bu projede kullanıcı, oyunda yanlış bir hamle yaptığında ve oyunu kaybettiğinde en iyi puanı görebilmesi ve yeniden başlayabilmesi için bir panel gösterilerek bir buton yardımı ile yeniden başlayabilmesi hedeflenmiştir.



Şekil 7 : Oyuna Yeniden Başlama Paneli

## 4.4 Projenin Kodlanması

### 4.4.1 Karakter Kontrolü

Şekildeki C# kodu, Unity oyun motorunda bir topun hareketini ve zıplamasını kontrol eden bir betik (script) olarak tasarlanmıştır. Kodun amacı, oyuncunun bir topu yatay olarak hareket ettirmesine ve belirli bir koşul sağlandığında topu zıplatmasına olanak tanımadır. Ayrıca, topun zıpladıktan sonra tekrar yere değip değmediğini de takip eder.

Kod, ilk olarak Unity'nin fizik motorunu kullanabilmek için gerekli olan "Rigidbody" bileşenini alır. Start fonksiyonu, oyun başladığında bir kez çalışır ve bu fonksiyon içinde topun Rigidbody bileşeni atanır ve topun yere değdiği varsayılır (yereDegdi değişkeni true olarak ayarlanır). Bu sayede, oyun başladığında topun yerde olduğu kabul edilir.

Update fonksiyonu, oyun çalıştığı sürece her karede tekrar tekrar çalışır. Bu fonksiyonda, oyun başladıysa (gameStart değişkeni true ise) topun hareketi ve zıplaması kontrol edilir. Eğer oyuncu boşluk tuşuna basarsa ve top yere değiyorsa, top yukarı doğru zıplatılır. Bu zıplama sırasında, bir ses efekti çalınır ve topun yere değmediği varsayılır (yereDegdi değişkeni false olarak ayarlanır).

Hareket fonksiyonu, oyuncunun topu yatay ekseninde hareket ettirmesini sağlar. Bu hareket, oyuncunun klavyedeki ok tuşlarına veya WASD tuşlarına basarak sağlanır. Kamera açısına göre topun hareket yönü belirlenir, böylece top ekranda doğru yönde hareket eder.

Son olarak, OnCollisionEnter fonksiyonu, topun başka bir nesneye çarpması durumunda çalışır. Eğer top, "Zemin" etiketi taşıyan bir nesneye çarparsa, topun yere değdiği kabul edilir ve yereDegdi değişkeni tekrar true olarak ayarlanır. Bu, topun tekrar zıplayabilmesi için gereklidir.

Bu kod parçası, temel bir oyun mekaniği sağlar; topun yatay olarak hareket etmesini ve zıplamasını kontrol ederken, oyuncunun doğru zamanda zıplama yapabilmesini sağlar. Bu mekanik, basit oyun projeleri için temel bir yapı taşını oluşturur (bakınız şekil 4.6).



```

using UnityEngine;

@ Unity Betiği (1 varlık başvurusu) | 0 başvuru
public class TopKontrol : MonoBehaviour
{
    public float hareketHizi = 10f;
    public float ziplaGucu = 5f;
    public Camera kamera;

    private Rigidbody rb;
    private bool yereDegdi = false;

    public AudioSource bounceSes;

    @ Unity İletisi | 0 başvuru
    void Start()
    {
        rb = GetComponent<Rigidbody>();
        yereDegdi = true; // Başlangıçta yere deđdiđini varsayıyoruz
    }

    @ Unity İletisi | 0 başvuru
    void Update()
    {
        if(RenkOyunu.gameStart == true)
        {
            Hareket();

            // Space tuşuna basıldıysa ve top yere deđiyorsa zıplama işlemi yap
            if (Input.GetKeyDown(KeyCode.Space) && yereDegdi)
            {
                bounceSes.Play();
                rb.AddForce(Vector3.up * ziplaGucu, ForceMode.Impulse);
                yereDegdi = false; // Zıpladıktan sonra yere deđmediđini varsayıyoruz
            }
        }
    }

    1 başvuru
    void Hareket()
    {
        float yatay = Input.GetAxis("Horizontal");
        float dikey = Input.GetAxis("Vertical");

        Vector3 kameraSag = kamera.transform.right;
        Vector3 kameraIleri = kamera.transform.forward;

        kameraIleri.y = 0f;
        kameraIleri.Normalize();

        Vector3 hareket = (kameraSag * yatay + kameraIleri * dikey) * hareketHizi;

        rb.AddForce(hareket);
    }

    // Top yere deđdiđinde yereDegdi'yi true yap
    @ Unity İletisi | 0 başvuru
    void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.tag == "Zemin")
        {
            yereDegdi = true;
        }
    }
}

```

Şekil 8 : Karakter Kontrol Kodları

Bu fonksiyonu sađlayan C# kodu, Unity oyun motorunda geliřtirilmiř bir renk tabanlı oyun projesinin iřleyiřini yönetmek için tasarlanmıřtır. Kod, oyun içindeki temel mekanikleri, karakterin davranıřlarını, zemin karolarının renk deęiřimlerini ve oyuncunun aldıęı puanları kontrol eder. Ayrıca, oyunun bařlangıcı, sona ermesi ve yeniden bařlatılması gibi kritik iřlevleri de ierir.

Oyun bařlarken, Start fonksiyonu devreye girer. Bu fonksiyon, bařlangı panelini ekranda gösterir ve mevcut puanı sıfırlayarak ekrana yansıtır. Aynı zamanda, daha önce elde edilmiř en yüksek puanı alarak ekranda gösterir ve oyuncunun bu puanı gemesi için motive olmasını saęlar. Oyunun bařında, karakterin arpıřma özellikleri de bu fonksiyon aracılıęıyla tanımlanır. Ayrıca, oyun sona erdięinde gösterilecek olan panel bařlangıta gizlenir.

Oyun devam ederken, zemin karolarının renkleri belirli zaman aralıklarıyla deęiřir. Bu iřlem, kod içinde tanımlanmıř bir coroutine ile gerekleřtirilir. Bu coroutine, sürekli alıřır ve her belirli zaman aralıęında zemin karolarının renklerini deęiřtirir. Renklerin rastgele bir sırayla karıřması ve karolar üzerinde eřitli animasyonların uygulanması saęlanır. Aynı zamanda, oyuncunun kontrol ettięi karakterin rengi de rastgele belirlenir ve zemin karolarının rengiyle eřleřip eřleřmedięi kontrol edilir.

Karakterin zemin üzerindeki doęru renkte olup olmadıęını kontrol eden bir iřlev de bulunmaktadır. Eęer karakter doęru renkte bir karo üzerinde duruyorsa, oyuncuya puan eklenir ve bu bařarıyı vurgulayan görsel ve iřitsel efektler devreye girer. Ancak, karakter yanlıř renkte bir karo üzerinde duruyorsa veya zemin üzerinde deęilse, oyun sona erer ve bu durumda yapılacak iřlemler bařlatılır.

Oyun sona erdięinde, oyunun renk deęiřim döngüsü durdurulur ve karakterin arpıřma özellikleri devre dıřı bırakılır. Oyuncunun aldıęı puan, daha önceki en yüksek puanla karřılařtırılır ve eęer yeni bir rekor kırılmıřsa, bu puan kaydedilir ve ekranda gösterilir. Bu, oyuncunun performansını artırmak için önemli bir motivasyon kaynaęıdır.

Oyuncu oyunu yeniden bařlatmak istedięinde, oyun sıfırlanır ve karakter bařlangı noktasına geri döner. Tüm puanlar sıfırlanır ve oyun yeniden bařlatılarak renk deęiřim döngüsü tekrar alıřtırılır.

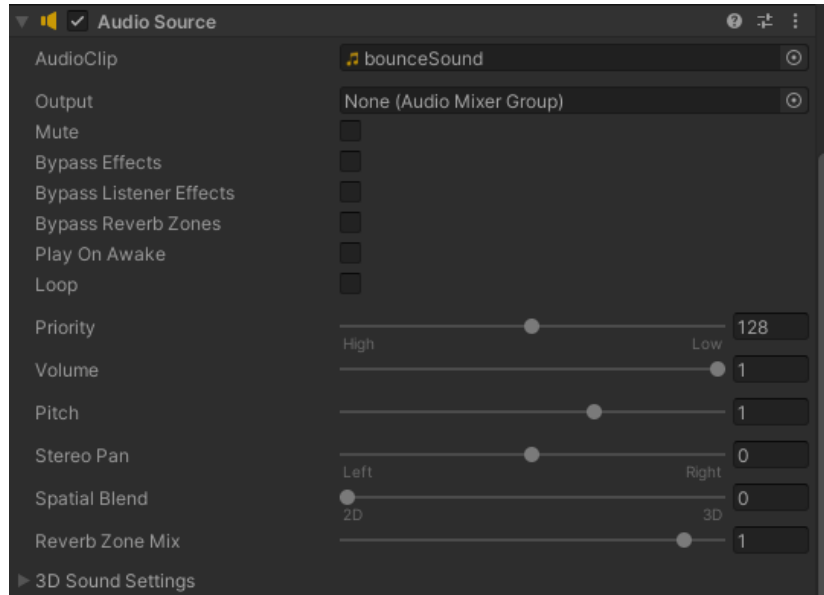
Son olarak, zemin karolarının renklerini rastgele bir sırayla karıştırılan bir işlev de kodda yer almaktadır. Bu işlev, zemin karolarının yan yana aynı renkte olmasını engellemek için renklerin karıştırılmasını sağlar ve bu şekilde oyunun zorluk seviyesi dengeli bir şekilde korunur.

Bu açıklamalarla, kodun genel işleyişi ve oyun mekaniği hakkında net bir anlayış elde edilebilir. Kod, oyuncunun reflekslerini ve görsel algısını test etmek üzere tasarlanmış bir oyun yapısını hayata geçirir ve bu süreçte gerekli olan tüm işlevleri içerir.

## 4.5 Ses Efektleri

### 4.5.1 Zıplama Efektı Sesi

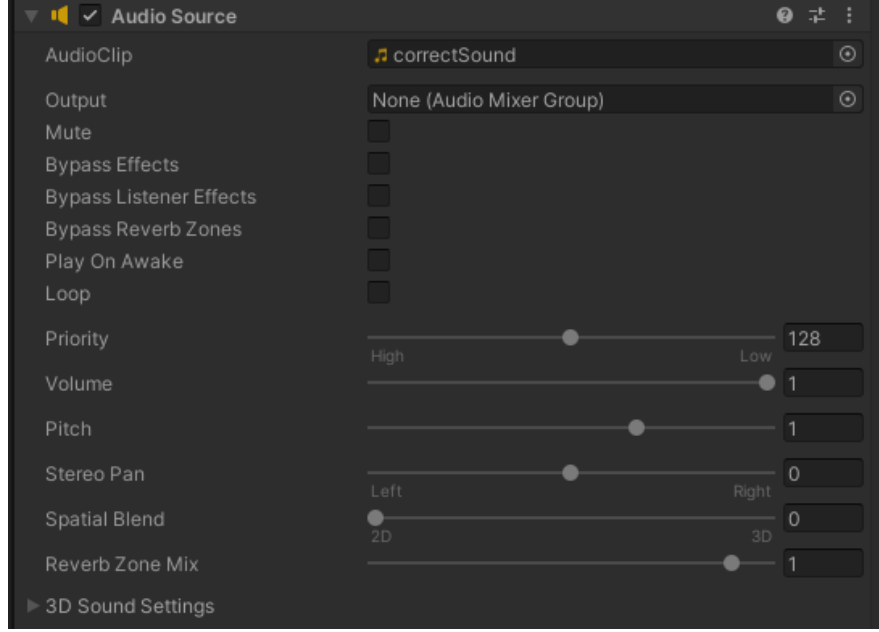
Bu projede, kullanıcı Space tuşuna bastığında zıplama sesinin çalınması için ses efekti eklenmiştir.



Şekil 9 : Zıplama Sesi

### 4.5.2 Süre Bitiş Efektı Sesi

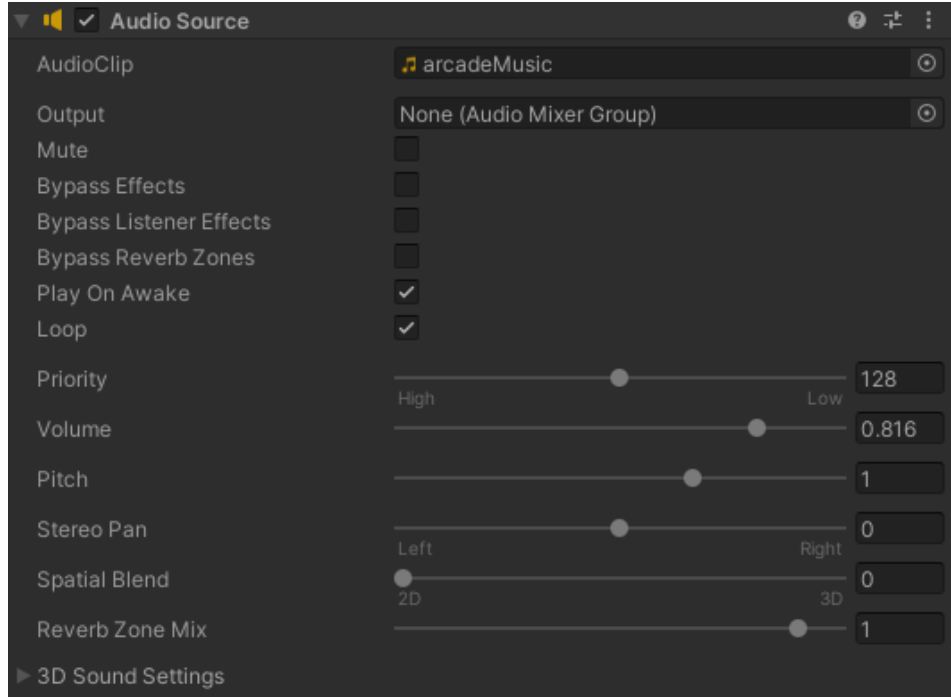
Bu projede, kullanıcının süre bittiğinde bundan haberdar olması için süre bitiş sesi eklenmiştir.



Şekil 10 : Doğru Sesi

### 4.5.3 Arka Plan Müziği Sesi

Bu projede, kullanıcının oyundan sıkılmaması ve oyunu daha eğlenceli kılmak için sürekli olarak tekrar eden bir arka plan müziği eklenmiştir.



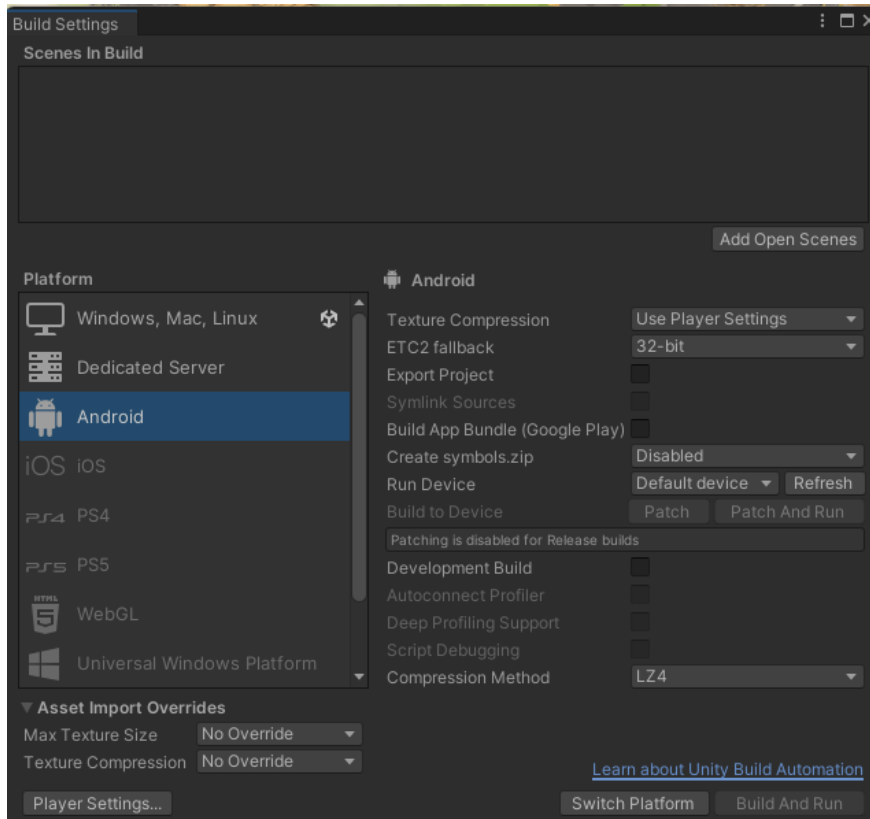
Şekil 11 : Arka plan Müziği

## 4.6 Çalıştırılabilir Çıktı Alınması

### 4.6.1 Android Çıktı

Unity ile Android platformuna oyun çıktısı almak için bazı gerekli ayarları yapmalıyız. İlk olarak, Android Build Support bileşenini yüklemeliyiz. Bu, Unity'nin Android için oyun derlemesine olanak tanır. Ayrıca, proje ayarlarında Android'e özgü seçenekleri düzenlemeliyiz. Bu aşamada, oyunun paket adı (package name), minimum ve hedef SDK sürümleri gibi önemli ayarları belirlemeliyiz.

Ayrıca, oyunun farklı Android cihazlarda sorunsuz çalışması için optimizasyonlar yapmalıyız. Gerekli ayarlamalar ve optimizasyonlar tamamlandıktan sonra, Unity'nin Build and Run özelliğini kullanarak APK dosyasını oluşturmalı ve bu dosyayı test cihazlarında denemeliyiz. Son olarak, oyunu Google Play Store gibi platformlarda yayınlamak için APK dosyasını hazırlamalı ve gerekli yükleme işlemlerini yapmalıyız. Bu adımlar, projemizin Android cihazlarda başarıyla çalışabilmesi için gereklidir.



Şekil 12 : Android Çıktı Alınması

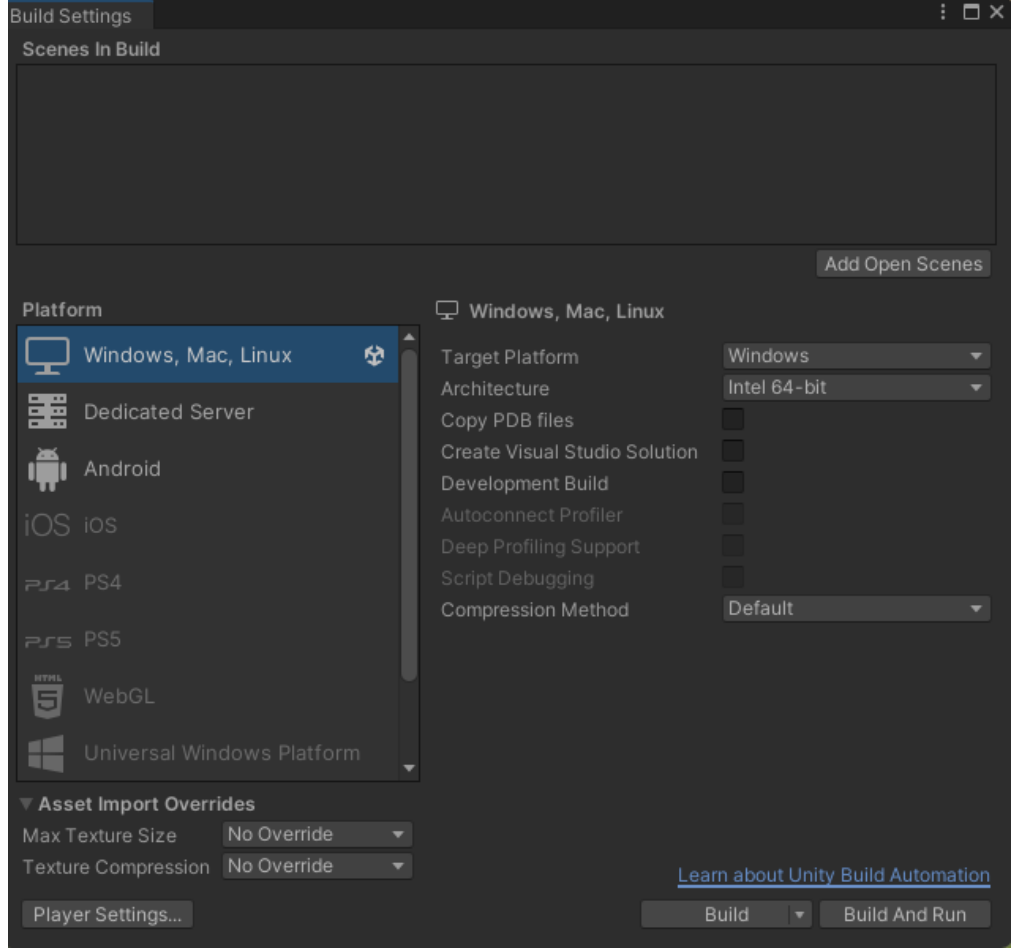
## 4.6.2 Windows Çıktı

Unity ile Windows platformuna oyun çıktısı almak için de bazı gerekli ayarları yapmalıyız. İlk olarak, Unity Editor'ün "Build Settings" menüsüne giderek, platform olarak "PC, Mac & Linux Standalone" seçeneğini seçmeliyiz. Burada, hedef platform olarak "Windows"u seçmeli ve oyunun 32-bit veya 64-bit sürümünü belirlemeliyiz.

Proje ayarlarında, çözünürlük, grafik ayarları ve giriş kontrolleri gibi Windows platformuna özgü seçenekleri düzenlemeliyiz. Bu, oyunun farklı ekran boyutlarında ve çeşitli donanım yapılandırmalarında sorunsuz çalışmasını sağlamak için önemlidir. Ayrıca, oyunun simgesini (icon) ve diğer Windows'a özgü meta verileri de ayarlamalıyız.

Ayarlar tamamlandıktan sonra, Unity'nin "Build and Run" özelliğini kullanarak oyunun .exe dosyasını oluşturmalıyız. Bu dosya, Windows işletim sisteminde çalıştırılabilir ve oyunun sorunsuz çalışıp çalışmadığını test etmek için kullanılabilir. Ayrıca, oluşturulan çıktı, oyunun dağıtımını için gerekli olan tüm dosyaları içerir, böylece oyunu farklı Windows cihazlarına kurmak ve çalıştırmak kolaylaşır.

Bu adımlar, projemizin Windows platformunda başarıyla çalışabilmesi için gereklidir.



Şekil 13 : Windows Çıktı Alınması

## Bölüm 5

### Sonuç

Bu çalışmada, dijital oyunların tarihçesi ve gelişim sürecinden başlayarak, mobil oyunların günümüzdeki önemine kadar geniş bir perspektif sunulmuş, ardından geliştirilen oyun projesi detaylı bir şekilde tanıtılmıştır. Oyun mekanikleri, kodlama süreçleri, tasarım unsurları ve platformlara yönelik çıktı alma işlemleri detaylandırılarak, oyun geliştirme sürecinin tüm aşamaları ele alınmıştır. Bu süreçte, oyun dünyasının evrimi, teknolojinin etkisi ve kullanıcı deneyiminin nasıl şekillendiği üzerinde durulmuştur.

Sonuç olarak, bu proje, basit bir oyun fikrinin nasıl kapsamlı bir dijital ürün haline getirilebileceğini göstermiştir. Geliştirilen oyun, temelinde bir eğlence aracı olmanın ötesinde, dijital kültürün bir parçası olarak, kullanıcıların görsel ve işitsel algılarını test eden bir araç haline gelmiştir. Bu süreçte, oyun tasarımının estetik değerleri ve kullanıcı dostu arayüzler oluşturmanın, oyun deneyiminin kalitesini ne denli etkilediği açıkça görülmüştür. Oyun programlama ve tasarımının, bir fikri somut bir ürüne dönüştürmede ne kadar kritik bir rol oynadığı, bu çalışmada ortaya konmuştur.

Ayrıca, bu projede kullanılan teknolojiler ve yöntemler, dijital oyun geliştirme alanında sağlam bir temel oluşturmuş ve bu alanın sürekli değişen dinamiklerine nasıl uyum sağlanabileceğini göstermiştir. Projenin başlangıcından sonuna kadar izlenen adımlar, oyun geliştirme sürecinin ne denli disiplinler arası bir yaklaşım gerektirdiğini ve her bir bileşenin uyumlu bir şekilde bir araya getirilmesinin önemini ortaya koymuştur.

Bu çalışmanın sonucunda, dijital oyunların sadece eğlence aracı olmanın ötesinde, kültürel ve teknolojik bir olgu olarak nasıl geliştiği ve bu süreçte geliştirilen oyun projesinin bu genel resim içinde nerede durduğu anlaşılmıştır. Yapılan araştırmalar ve pratik uygulamalar, mobil oyunların günümüzdeki önemini ve dijital çağın bir parçası olarak ne denli büyük bir etki yarattığını ortaya koymuştur.

Sonuç olarak, bu proje, bir oyunun fikir aşamasından nihai ürüne dönüşme sürecinde karşılaşılan zorlukların ve bu zorlukların nasıl aşılabileceğinin bir örneğini sunmaktadır. Oyun geliştirme sürecinde edinilen bilgi ve deneyimler, gelecekte yapılacak çalışmalar için sağlam bir temel oluşturmakta ve bu alanda daha ileri projelere zemin hazırlamaktadır. Bu çalışma, dijital oyunların tasarımı ve geliştirilmesi konusunda derin bir anlayış kazandırmakta ve oyun geliştirme sürecinin ne denli karmaşık ve çok yönlü olduğunu gözler önüne sermektedir. Bu süreç, bir yandan teknolojik bilgi birikimini gerektirirken, diğer yandan yaratıcı düşünce ve estetik duyarlılıkla desteklenmesi gereken bir alan olduğunu kanıtlamaktadır.

Sonuç olarak, bu çalışma, dijital oyunların ve özellikle mobil oyunların, günümüz kültüründe ve teknoloji dünyasında oynadığı rolü anlamak ve bu alanda nasıl etkili projeler geliştirilebileceğini görmek açısından önemli bir katkı sağlamaktadır. Proje,



dijital oyunların nasıl geliştirileceğine dair kapsamlı bir rehber sunmakta ve bu alanda çalışan veya çalışmayı planlayan bireylere yol gösterici nitelikte bilgiler sunmaktadır.

# Kaynaklar

Schell, J. (2019). *The Art of Game Design: A Book of Lenses*. CRC Press.

Adams, E. (2014). *Fundamentals of Game Design*. Pearson Education.

Unity Technologies. (2021). *Unity Manual*. Unity Technologies.

Goldstone, W. (2020). *Unity 2020 from Zero to Proficiency*. PACKT Publishing.

Bogost, I. (2011). *How to Do Things with Videogames*. University of Minnesota Press.

Nystrom, R. (2014). *Game Programming Patterns*. Genever Benning.