



**ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT**

Project Report
Version 1

CENG 407
Innovative System Design and Development I

TEAM - 6
BEYBLADE AR

Ahmet Faruk İLOĞLU
Ayşe ÖZTAN
Doğa DÜZGÖL
Egemen CENGİZ
Erce ALTUN

Advisor: *Dr. Serdar ARSLAN*

Table of Contents

Table of Contents	1
Abstract	3
Özet:	3
1. Introduction.....	4
2. Literature Review.....	5
2.1 Beyblade Game	5
2.2 What is a Game Engine?	5
2.3 Unity.....	6
2.3.1 Unity Assets	6
2.3.2 Unity Physics	7
2.3.3 Unity Scripting.....	7
2.3.4 Unity Animations	7
2.4 Why are we using C# for scripting in our project?.....	7
2.4.1 Photon Unity Networking (PUN).....	8
2.4.2 Installing Photon Engine and Connecting to Unity Game Engine	8
2.4.3 Multiplayer.....	9
2.5 What is Augmented Reality (AR)?.....	9
2.5.1 History of AR.....	10
2.5.2 Examples of AR	11
2.6 What is Virtual Reality (VR)?.....	13
2.6.1 Virtual Reality (VR) vs Augmented Reality (AR)?	13
2.6.2 ARKit	14
2.6.3 ARCore	14
2.6.4 ARKit vs ARCore	15
2.6.5 ARToolKit	15
2.7 What is Artificial Intelligence (AI)?.....	21
2.7.1 Applications of AI.....	21
2.7.2 AI in Gaming.....	22
2.7.2.1 Techniques and Terms for AI in Games	22
2.8 AR, AI, and Games Relation.....	25
3. Conclusion	25
4. References.....	26

5.	Software Requirement Specification (SRS)	28
5.1	Introduction	28
5.1.1	Purpose of the Document	28
5.1.2	Scope of Project	28
5.1.3	Glossary	29
5.1.4	References	29
5.1.5	Overview of Documents	30
5.2	Overall Description	30
5.2.1	Product Perspective	30
5.2.2	User Characteristic	31
5.2.3	Requirements Specification	31
5.2.4	Functional Requirements	32
5.2.5	Performance Requirement	41
5.2.6	Software System Attributes	42
5.2.7	Safety Requirement	42
5.3	References	42
6.	Software Design Description (SDD)	43
6.1	Introduction	43
6.1.1	Purpose of This Document	43
6.1.2	Scope of the Project	43
6.1.3	Glossary (Definitions, Acronyms, and Abbreviations)	44
		44
6.1.4	Overview of This Document	44
6.1.5	Motivation	44
6.2	System Design	45
6.2.1	Architectural Design	45
6.2.2	User Interface Design	47

Abstract

Today, everyone knows that video games are one of the most loved entertainment industries in the world. As the game development field has grown, the expectations of gamers from the story and gameplay in the games have increased. In addition, since children are introduced to technology at an early age, the mobile game industry has developed itself tremendously. In the 80s and before, children used to go out to the streets and play with their stone toys. Over time, the toys derived from these tools gave way to Beyblade-type toys in the 2000s. Finally, all new-time children are officially in the game culture by playing digital games. In the past, children could not play alone when they did not have friends, now it is not even possible for children to be bored thanks to artificial intelligence. Artificial intelligence has been integrated into competitors and similar actors in the game in order to increase the enjoyment of the users in solo games, making the game more enjoyable. Thus, the players began to feel as if they were playing the game they were playing, not against artificial intelligence, but against a real player. Afterwards, in the near decade, augmented reality came into our lives and enabled technology to get out of abstraction and take a 3D concrete view. AR is now used in every part of our lives in the field of play, health, education, and entertainment. In this article, we explore the vast computer science literature to evaluate how artificial intelligence and augmented reality work in a vintage Beyblade game. By quoting from articles that address the topic, we will examine it in depth.

Özet:

Bugün herkes video oyunlarının dünyanın en sevilen eğlence endüstrilerinden biri olduğunu biliyor. Oyun geliştirme alanı büyüdükçe oyun severlerin oyunlarda hikâyeden ve oynanıştan beklentileri arttı. Ayrıca çocukların teknoloji ile erken yaşta tanışması ile mobil oyun sektörü de kendini müthiş bir şekilde geliştirdi. 80'ler ve öncesinde çocuklar sokaklara çıkar taş oyuncaklarıyla oynarlardı. Zamanla bu aletlerden türetilen oyuncaklar 2000'li yıllarda yerini Beyblade tipi oyuncaklara bıraktı. Son olarak tüm yeni zaman çocukları resmen dijital oyunlar oynayarak oyun kültürünün içindedir. Eskiden arkadaşları yokken çocuklar tek başlarına oynayamıyorlardı, şimdi yapay zekâ sayesinde çocukların sıkılması bile mümkün değil. Kullanıcıların solo oyunlarda daha fazla keyif almaları için oyundaki rakip ve benzeri aktörlere yapay zekâ entegre edilerek oyunu daha keyifli hale getirildi.

Böylece oyuncular oynadıkları oyunu yapay zekaya karşı değil de gerçek bir oyuncuya karşı oynuyormuş gibi hissetmeye başladılar. Daha sonra, yaklaşık on yıl içinde hayatımıza artırılmış gerçeklik girdi ve teknolojinin soyutlamadan çıkıp 3 boyutlu somut bir görünüm almasını sağladı. AR artık hayatımızın her alanında oyun, sağlık, eğitim ve eğlence alanında kullanılmaktadır. Bu makalede, eski bir Beyblade oyununda yapay zekâ ve artırılmış gerçekliğin nasıl çalıştığını değerlendirmek için geniş bilgisayar bilimi literatürünü araştırıyoruz. Konuyu ele alan makalelerden alıntı yaparak derinlemesine inceleyeceğiz.

1. Introduction

A childhood love of gaming has developed into an addiction for teenagers, adults, and even the elderly. In today's simple games, "players engage in artificial conflict that is defined by the rules and results in a measurable outcome," as the phrase goes. Players frequently need to complete specific objectives in artificial skirmishes used in video games in order to earn points. For instance, gathering items, improving one's health or abilities through interaction, critically striking an adversary, etc. By playing these games, the younger generation has developed skills like being able to easily access information at a young age and effectively use that information. They frequently have no trouble solving problems in their daily lives because of these abilities. [33]

Artificial intelligence has been incorporated into games, which has resulted in the addition of new features, modifications to the environments, and the use of a fluid progression method. Dynamic developments started to be presented to the players as the game went on. The developers made these adjustments to the game in order to increase player enjoyment, and these adjustments provided the players with new experiences.

With the addition of augmented reality to games, games have become more realistic, educational, and enjoyable. We witnessed the addition of augmented reality to games most clearly in 2016 when Pokémon GO came out and created a big wave.

Beyblade 1999 is a nostalgic strategy game where you fight as if you are fighting in the real world with augmented reality (AR), you fight with different types of artificial intelligence-integrated enemies and you can play with your friends, advance strategically, and give fun moments.

The main goal of the player is to have their Beyblade stay in the game longer than their opponent (AI) or disqualify their opponent from the game by using their Beyblade abilities.

2. Literature Review

2.1 Beyblade Game

Beyblade is a spinning top toy line released in Japan in July 1999. Both the toys and their names were inspired by Beigoma, a traditional spinning top. The toy series was introduced in 1999 with a manga series of the same name, and it has become one of the greatest joys of a generation. We will bring this nostalgic toy, which has become the indispensable toy of the children of the 2000s, to life in its most technological way. [1]



2.2 What is a Game Engine?

For the creation of games and multimedia material, game engines are sophisticated and multifaceted tools. Game engines should support a wide range of game development tasks, rendering, physics, audio, animation, artificial intelligence, and UI design. A game engine's primary objective is to abstract typical video game characteristics, enabling the reuse of code and game assets across many titles. [2]

2.2.1.1 What Features Should Game Engines Include?

A game engine should have the following functions:

A rendering engine, for 2D or 3D graphics, Input handling (for keyboard & mouse, touch devices or other hardware, etc.), Game loop (the internal routine that recalculates game events every frame), A physics engine, with collision detection and response, sound, A scene graph (which manages graphical elements and their relationships on the screen), Animation (for 2D sprites or 3D models), Memory management. Scripting, artificial intelligence, networking, streaming, localization support, and multi-platform publication are examples of additional features. [3]

Some Popular Game Engines:

- Unreal Engine
- Unity
- Amazon Lumberyard
- CryEngine
- GameMaker: Studio
- Godot
- Cocos2d [4]

	Installation & Ownership	2D/3D	Ease of Use	Integration & Compatibility	VR Support	Customer Support
Unreal Engine	***	Both	***	*****	Yes	***
Amazon Lumberyard	***	3D Only	*****	***	Yes	***
CryENGINE	***	Both	***	*****	Yes	***
Unity	***	Both	***	*****	Yes	***
GameMaker: Studio	*****	2D Only	*****	***	No	***
Godot	*****	Both	*****	***	No	*****
Cocos2d	*****	2D Only	*****	***	No	***

2.3 Unity

Unity is a cross-platform game engine and integrated development environment (IDE) for creating interactive media, usually video games and simulations, by Unity Technologies. Unity was first announced and released as a game engine for Mac OS X at Apple Inc.'s Worldwide Developers Conference in June 2005. The engine has since been gradually expanded to support a variety of desktop, mobile, console, and virtual reality platforms. As CEO David Helgason puts it, Unity is "a toolset for building games and the technology that runs the graphics, sound, physics, interactions, and networks." [9]

2.3.1 Unity Assets

An asset is any item in a Unity project that is used to build a game or application. Assets can represent visual or audio elements in the project, such as 3D models, textures, motion graphics, sound effects, or music. Assets can also represent more abstract elements such as color gradients, animation masks, or arbitrary text or numeric data for any use.

An asset can come from a file created outside of Unity, such as a 3D Model, audio file, or image. Some asset types can also be created in Unity Editor. Through the Unity Asset Store, creators can create user-generated assets and market them to other video game developers. [10]

2.3.2 Unity Physics

Unity physics can be simulated to ensure that objects in a Unity project accelerate and respond appropriately to collisions, gravity, and other forces. In Unity, one can choose from a variety of physics engine implementations, including 3D, 2D, object-oriented, and data-oriented options, depending on the needs of the project. [11]

2.3.3 Unity Scripting

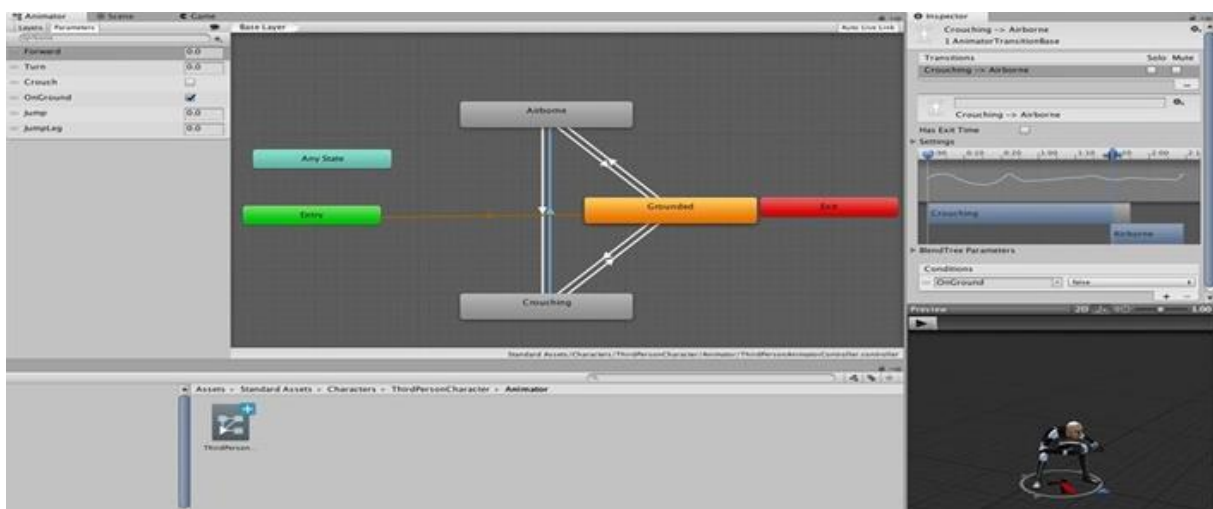
Unity provides three programming languages to its developers: Boo, C#, and Unity JavaScript (commonly referred to as UnityScript). The developers should be able to implement the same content regardless of their preferred language using any one of these scripting languages. [9]

2.3.4 Unity Animations

Animation features in Unity include retargetable animations, complete runtime control over animation weights, event calling from within the animation playback, complex state machine hierarchies and transitions, blend shapes for facial animations, and many more. In addition, the Animator widget in Unity contains elements that we choose the order of the animations and how, where, and when they play. [13]

2.4 Why are we using C# for scripting in our project?

C# is the more popular programming language than the other 2 languages used to create games in the Unity game engine. The C# scripts that power Unity and control every function of the engine are stored in code files. Scripts are still the best way to add unique interactions and actions to a game environment, despite the fact that many new tools now exist that enable developers to create games without them. Since C# can be implemented more easily and more effectively with an object-oriented program, we decided to script our game with C#. [12]



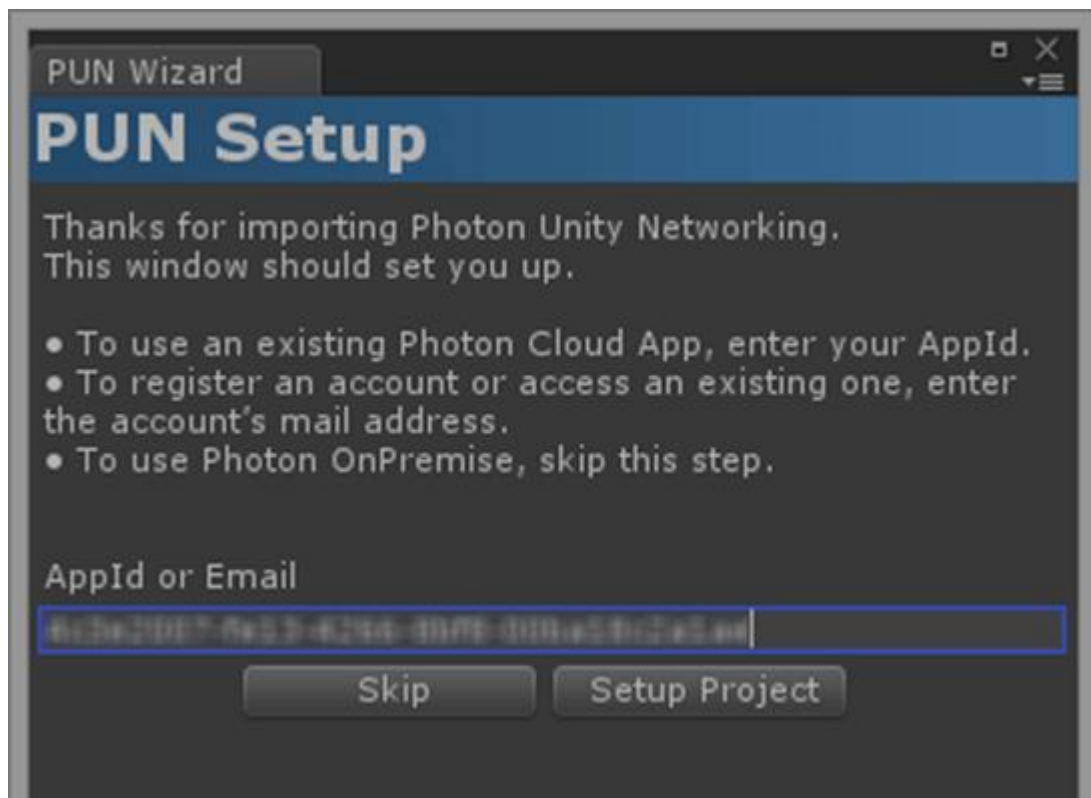
2.4.1 Photon Unity Networking (PUN)

Photon Unity Networking is the abbreviation for PUN. It enables gamers and game developers to quickly integrate Photon with the Unity framework. By doing this, you can create the foundations for a network that works and for servers that operate quickly. PUN is an excellent and quick way to transfer servers, rooms, and other features that make games operate smoothly in addition to real-time.

PUN connects to a dedicated server that provides player rooms, matchmaking, and in-room communication at all times. Multiple servers are used by Photon Unity Networking in the background: While several "Game Servers" control the actual rooms, a "Master Server" keeps track of all the rooms and match players (matches).

2.4.2 Installing Photon Engine and Connecting to Unity Game Engine

The first step is to sign up and sign in to a free Photon Unity (PUN) account. The PUN Wizard opens when you import the PUN into a new project. Alternatively, you can open it from the Window > Photon Unity Networking menu.



Next, we connect to the client by writing the code "PhotonNetwork.ConnectUsingSettings()". Then we make the server settings. [7]

Server settings are where user rooms configure the way matches and players connect. Users have the option to connect to regional servers, AppId, and game version from this window.

2.4.3 Multiplayer

There are certain methods used to develop multiplayer games. Thanks to the "Netcode" released by Unity in recent months, it is no longer necessary to spend a lot of time on low-level protocols and network frameworks. But at the moment Netcode is a network library that is not meant to be used for massive multiplayer game designs. It is mostly suitable for 2-8 player game designs. In addition, this library contains debugging tools to fix bugs faster.

All of your project's netcode-related settings are contained in the NetworkManager, which is a necessary Netcode for GameObjects (Netcode) component. NetworkManager has very useful features. Below are some of them.

Protocol Version: Set this value to help distinguish between builds when the most current build has new assets that could cause issues with older builds connecting.

Tick Rate: This value controls the network tick update rate.

Connection Approval: This enables connection approval when this is checked and the NetworkManager.ConnectionApprovalCallback is assigned.

LogLevel: Sets the network logging level

PlayerPrefab: When a prefab is assigned, the prefab will be instantiated as the player object and assigned to the newly connected and authorized client. [7]

This and all other NetworkManager features work together with Netcode. In order to download Netcode and add it to our project, it is necessary to download Unity versions that support Netcode. These are Unity 2020.3, 2021.1, 2021.2, and 2021.3 LTS. In addition, Netcode supports Windows, MacOS and Linux, iOS, and Android platforms. But Netcode does not support the WebGL platform because it does not allow access to IP Sockets. [8]

2.5 What is Augmented Reality (AR)?

Augmented reality (AR) is an enhanced version of the real physical world that is achieved through the use of digital visual elements, sound, or other sensory stimuli and delivered via technology.

One of the main objectives of augmented reality is to draw attention to specific physical world characteristics, understand these characteristics better, and derive clever and approachable insights that may be used in practical contexts. [14]

Augmented reality has started to be used in many fields such as education, security, defense, medicine, game development, and decoration.

2.5.1 History of AR

- 1968: Ivan Sutherland invents the head-mounted display and positions it as a window into a virtual world.
- 1975: Myron Krueger creates Videoplace to allow users to interact with virtual objects.
- 1980: Steve Mann creates the first wearable computer, a computer vision system with text and graphical overlays on a photographically mediated scene.
- 1987: Douglas George and Robert Morris create a working prototype of an astronomical telescope-based "heads-up display" system (a precursor concept to augmented reality) which superimposed in the telescope eyepiece, over the actual sky images, multi-intensity star, and celestial body images, and other relevant information.
- 1990: The term augmented reality is attributed to Thomas P. Caudell, a former Boeing researcher.
- 1992: Louis Rosenberg developed one of the first functioning AR systems, called Virtual Fixtures, at the United States Air Force Research Laboratory—Armstrong, that demonstrated benefit to human perception
- 1999: Frank Delgado, Mike Abernathy et al. report successful flight test of LandForm software video map overlay from a helicopter at Army Yuma Proving Ground overlaying video with runways, taxiways, roads, and road names.
- 2000: ARToolKit, an open-source SDK, was developed by a Japanese scientist Hirokazu Kato.
- 2004: Outdoor helmet-mounted AR system demonstrated by Trimble Navigation and the Human Interface Technology Laboratory (HIT lab)
- 2008: Wikitude AR Travel Guide launches on 20 Oct 2008 with the G1 Android phone.
- 2009: ARToolkit was ported to Adobe Flash (FLARToolkit) by Saqoosha, bringing augmented reality to the web browser.
- 2015: Microsoft announces Windows Holographic and the HoloLens augmented reality headset. The headset utilizes various sensors and a processing unit to blend high-definition "holograms" with the real world.
- 2016: Niantic released Pokémon Go for iOS and Android in July 2016. The game quickly became one of the most popular smartphone applications and in turn spikes the popularity of augmented reality games.
- 2019: Microsoft announces HoloLens 2 with significant improvements in terms of field of view and ergonomics.[34]

2.5.2 Examples of AR

Pokémon Go: The most well-known augmented reality gaming app is Pokémon Go. This program is renowned for its outstanding usability and performance. The game itself is very captivating, and upon its July 2016 release, it swept the globe by storm. There are several mobile AR applications that will undoubtedly take inspiration from this app. The gaming industry's turning point was

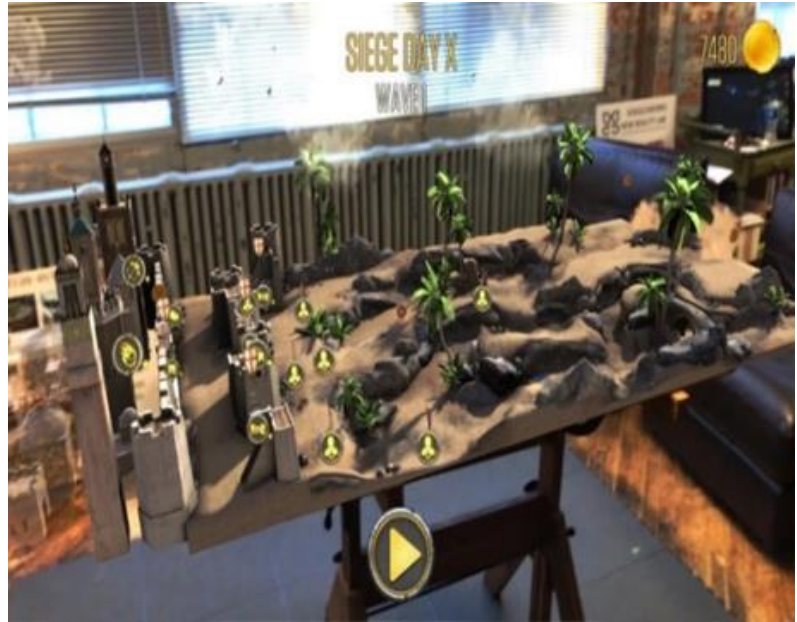


when it first appeared. The game provides a virtual battlefield and realistically brings virtual items to life. The users are able to view and catch Pokémon nearby.

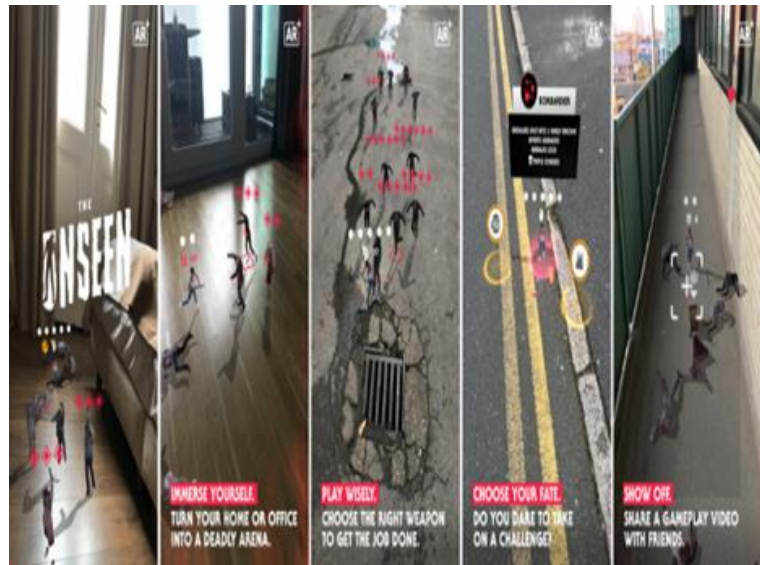
Harry Potter: Wizards Unite: The possibilities of reality are enhanced by an exceptional spiritual journey. A disaster has befallen Harry Potter's magical realm in the adventure and narrative game Harry Potter: Wizards Unite. To resolve this problem and defeat the adversaries, the Wizards of the world must band together. The wizard who plays the game must battle his way through adversaries and gather various components. This game is specifically made for fans of fantasy and adventure. For the fans, the augmented reality app incorporates several cameos of well-known figures.



Knightfall AR: A game that immerses the player in the realm of knights and warriors for battle strategy. The goal is to keep Acre safe from the adversaries (Mamluk Warriors). The video games throw the player onto a smooth surface. You can shoot at the enemy troops as they advance toward you using the user's viewpoint as a targeting system. In AR mode, you can defeat adversaries, acquire riches, bolster your fortifications, and drive back troops. You unlock features and take on new challenges as you progress and win battles. AR works its magic throughout games to give the user a real-time, exciting experience.



Zombie, Run! A free software that caters to individuals who enjoy a morning jog around the city and delivers exciting adventures to Android and iOS users. You won't be able to go for a morning run the same way after you start using this augmented reality game app. The player is immediately thrust into a world full of zombies where they must flee and gather objects along the way to survive the zombie invasion. The game also includes music and inspirational narration to help you get motivated to exercise in the morning.



Jurassic World Alive: Explore the primordial era when enormous creators walked the Earth. The Jurassic World will transport you to a setting where untamed dinosaurs roam freely among contemporary societies. You can investigate various facets of the Jurassic era, encounter mammoths, dinosaurs, tyrannosaurus, and Pteranodons, and gather DNA samples to produce your own raptor. Additionally, you can compete against other creatures in real time and share your creation with other people. The quick mechanism of the game and experience is designed to thrill the player. This exceptional augmented reality adventure app made it to this list of motivating apps thanks to its well-designed graphics. [22]



2.6 What is Virtual Reality (VR)?

Virtual reality (VR) is an artificial digital environment that completely replaces the real environment or "real world". Participants can navigate the virtual world. Thanks to VR technology, the user is in the virtual environment, the user can see, access, capture, and reshape this environment not only from a single angle but also from many different angles. Today, VR technology is used in many areas such as entertainment, automotive, education, health, tourism, and military. [15][16]

2.6.1 Virtual Reality (VR) vs Augmented Reality (AR)?

Augmented reality combines the existing real-world environment with a virtual world. Smartphones or tablets can be used for AR. The user directs the camera of this device to a point. Virtual reality puts the users in a completely virtual environment. Eye-covering headsets and headphones are used for VR. AR immerses the user partially in the action, while VR completely immerses the user in the action.

2.6.1.1 Advantages and Disadvantages of AR

Advantages of AR

A better experience is provided to users by augmented reality. As an illustration of this experience, when users hold their phones up to a building, you can view the building from various angles and utilize advanced maps to figure out where you are in relation to the building.

To use augmented reality, smartphone users simply need to point their camera at an object. [17]

Disadvantages of AR

The creation and upkeep of AR technology-based initiatives are expensive. In addition, making AR-based gadgets is expensive.

Applications based on augmented reality raise privacy concerns. [17]

2.6.2 ARKit

Apple's augmented reality software for iOS mobile devices is called ARKit. Developers can create AR experiences for iOS and iPadOS as a result. The AR object "appears" in the physical surroundings by overlaying virtual pictures to real-world things. With the 2020 update, Apple introduced the built-in LiDAR technology, which gives customers new experiences and improves the environment. LiDAR is used to calculate the distance between the device and the surface, the time it takes for a light pulse to travel from a device to the surface and back. A single dot is formed as a result of each light pulse. A topographic map of the user's surroundings is made using "point clouds," which are collections of these points. [18][19]

2.6.3 ARCore

Google's framework for creating augmented reality experiences is called ARCore. Your phone can perceive its surroundings, comprehend the outside world, and interact with information thanks to ARCore, which makes use of many APIs. To enable shared AR experiences, some APIs are accessible on iOS and Android.

To combine virtual material with the real world as seen through your phone's camera, ARCore makes use of three crucial capabilities:

- The phone can comprehend and track its position in relation to the outside world thanks to motion tracking.
- Through environmental knowledge, the phone can determine the size and placement of any kind of surface, including inclined, horizontal, and vertical surfaces like walls, coffee tables, and the ground.
- The phone can assess the amount of light present in the surrounding area.

When you move your phone, ARCore generates a universe in which it can place virtual things after memorizing the surroundings. It recognizes specific characteristics that enable it to track how these dots move in response to the movements of your camera using motion-tracking technology. When you set an object, it determines its position in relation to other objects, and when you come back to the same location, the object is reproduced. [18][19][20][21]

2.6.4 ARKit vs ARCore

- If you're looking to build AR apps that track user movements to manipulate on-screen images, ARKit will often be the more efficient option.
- ARKit offers faster, more accurate, and more powerful tools.
- Developers using ARCore can develop for Android as well as Apple devices. This is an advantage that ARKit developers don't have.
- ARKit is better for image recognition and certain iOS tasks, while ARCore is better for general graphics processing and gaming. [18][19][20][21]

2.6.5 ARToolKit

Augmented reality might not be as complex and exciting as a virtual reality simulation ride, it could end up being a highly practical tool in our daily lives. It provides the potential to enhance and glamorize the things we see, hear, and feel by bringing components of the virtual world into the real world. When ARToolKit originally appeared, it was a game changer for the majority of augmented reality (AR) developers because it was free and open source in addition to being an excellent tool for AR. [23]

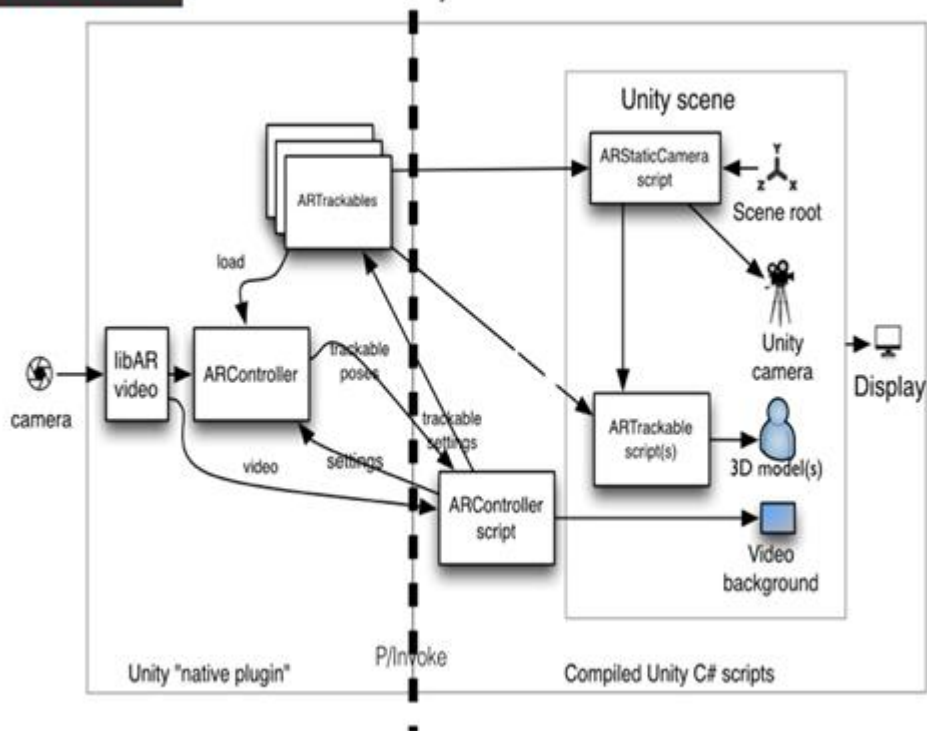
2.6.5.1 The History of ARToolKit

When Hirokazo Kato joined the HITLab in 1999, ARToolKit was created. The shared space project received its initial demonstration at SIGGRAPH 1999. Support for Windows DirectShow Video has undoubtedly contributed to the widespread adoption of many exhibitions. Significant changes have been made in recent years (multiplatform, better tracking algorithm).

- 1999 First public demonstration @ SIGGRAPH
- 2001 v1, with square tracking, released open source
- 2001 ARToolworks Incorporated, begins selling the Pro version
- 2004 NFT tracking integrated into ARToolKit
- 2004 v2 released open source
- 2008-10 iPhone 3g Support, Open-Source Innovation
- 2011-14 Android support added, further extending platform support
- 2015 Acquired by DAQRI, pro version open sourced
- 2017 ARToolKit-6 The big release in the AR industry. [23]

2.6.5.2 How ARToolKit Works

Although there are numerous stages required, most of them are rather simple in concept. Computer-generated images are often inserted into real-life images. An image is first captured by a camera and sent into the route to start the process. The image is then examined for distinctive, specific markers. If the pattern is detected, the computer will keep track of it or sort it out by comparing it to a number of other patterns in a pre-built database. If a match is discovered, the algorithm understands which object (3D Model) to connect to that particular pattern. When that solely distinct pattern is discovered, the 3D object that was saved in the database will be rendered or displayed.



For additional information of ARToolKit; the following are supported by the most recent version of ARToolKit, version 6.

For computer:

- MacOS
- Windows
- Linux

For mobile phones:

- iOS
- Android
- SymbianOS

Additionally, a Unity game engine plugin for ARToolKit is accessible. The plugin is compatible with OS X, Windows, Android, and iOS versions of Unity. [23]

2.6.5.3 Advantages of ARToolKit

- Caters Almost all available platform now
- Easy to implement
- Easy to use
- Have a wide range support
- Fully detailed documentation
- Free and open source [23]

2.6.5.4 How Pattern Can Be Track?

Planar textured surfaces are among the natural properties of images and documents that ARToolKit NFT can identify and monitor. However, in order to do this, ARToolKit NFT has to know in advance how the surface would look. As a result, in order for the system to recognize and track the surface, it must first be trained with a surface. A set of data that may be used for real-time tracking in applications employing the ARToolKit SDK is the training's output.

The surfaces that can be used with ARToolKit NFT are subject to the enumerated limitations.

- Rectangular images of the surfaces that need to be tracked must be provided.
- Jpeg format must be used for the photos.
- The surface needs to have a decent quantity of tiny detail, sharp edges, and texture (a low degree of self-similarity and high spatial frequency). Large swaths of one flat color in images, as well as images that are blurry or have soft details, will not track effectively, if at all. It is challenging to identify separate feature points in such photos.
- When the camera is closer to the image or when a higher resolution image is used, the ability to extract feature points at higher levels of detail will be enabled by larger or higher resolution images (more pixels).

To implement NFT tracking, the ARToolKit NFT tracker does not necessitate enhancing the image using Natural Feature Tracking with Fiducial Markers. However, when utilizing the NFT tracker, fiducial markers can also be included in the dataset together with NFT markers for more effectiveness and robustness. [23]



Figure 1: You could see a printed pinball picture system where other shape can be defined

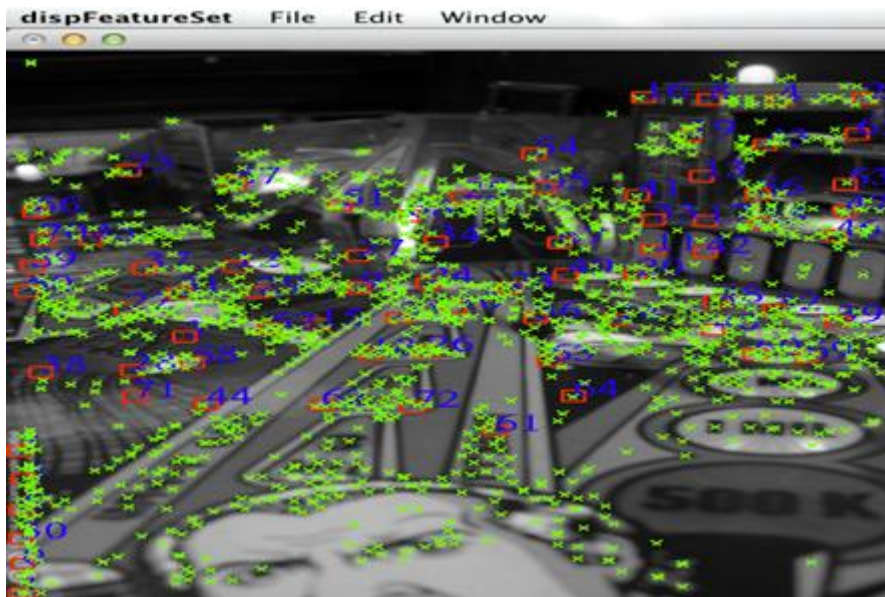


Figure 2: The pattern gets its own coordinate

2.6.5.5 How Trackable Image Pattern Process?

Every trackable visual pattern has its own coordinate system and understands when to display 3D models.

Define a new coordinate system with the origin at the middle of the pattern; that would be acceptable. As seen in figure 3, the z-axis may be outside the pattern's plane while the x and y-axes may be on it.

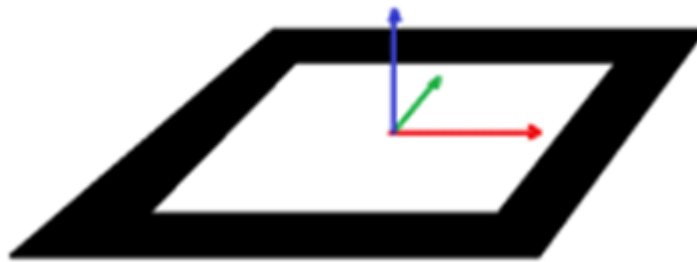


Figure 3: Having the x and y axis and point of origin

Therefore, as illustrated in figure 4, when a 3D Model (Item) is placed on top of the pattern, it will remember and obtain the coordinates of the object.

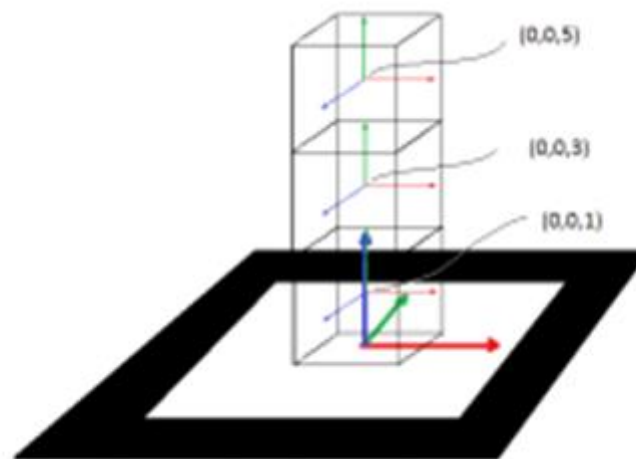


Figure 4: A Model is place on the pattern.

When the box is ultimately superimposed over the camera image, it appears as though it is actually standing on the pattern in real life, as seen in figure 5. The placement of the pattern in the camera image, as seen from the camera. [23]

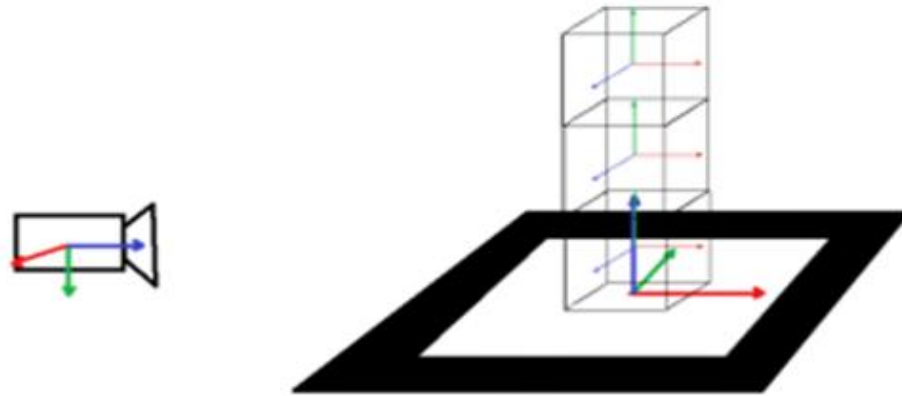


Figure 5: The camera has its own coordinate system where the pattern will be defined

2.7 What is Artificial Intelligence (AI)?

Artificial intelligence (AI) is the emulation of human intellect in devices that have been designed to behave and think like humans. The phrase may also be used to refer to any computer that demonstrates characteristics of the human intellect, such learning and problem-solving. Ability to reason and take actions that have the highest likelihood of reaching a certain objective is the ideal quality of artificial intelligence. Machine learning (ML), a subtype of artificial intelligence, is the idea that computer systems can automatically learn from and adapt to new data without human assistance. Deep learning algorithms allow for this autonomous learning by ingesting vast quantities of unstructured data, including text, photos, and video. Artificial intelligence is founded on the idea that human intellect can be described in a way that makes it simple for a computer to imitate it and carry out tasks of any complexity. [24]

2.7.1 Applications of AI

Artificial intelligence can be very helpful in a variety of fields, including astronomy to solve complex universe problems, health care to speed up diagnosis, finance for automation, chatbot, adaptive intelligence, and algorithm trading, social media for identify tags and user requirements, education, e-commerce, travel, and a countless number of other fields. Additionally, there is a significant influence on our topic, gaming. [25]

2.7.2 AI in Gaming

Video game responsiveness and adaptability are referred to as artificial intelligence. These AI-powered dynamic experiences are often created by NPCs, or non-player characters, that behave intelligently or imaginatively as though they were being directed by a human gamer. An NPC's behavior in the game environment is determined by AI. [26] It has several applications in the gaming business. For instance:

Adaptive games need reinforcement learning: A kind of machine learning called reinforcement learning enables the machine to adapt its behavior based on input from the outside world. Games provide a lot of promise for advanced learning.

Maintain high-quality player interaction: AI grows clever enough to react genuinely to live input rather than predetermined narratives, resulting in increased realism and natural interactions between players and environments. AI can also assist some fascinating marketing applications, but only provided a good user experience is maintained. [27]

Improving the Overall Gaming Experience: Modern gamers pay close attention to how colorful and dynamic the game is in every manner. AI has the capacity to play a significant role in elevating the game experience through ongoing scenario optimization.[28]

2.7.2.1 Techniques and Terms for AI in Games

Most games employ extremely basic AI strategies, some of which, from an academic perspective, wouldn't even be called AI in the first place. Because of how easy they are to use and how powerful they are structurally, finite state machines (FSMs) are the time favourite of game AI and are employed in every computer game. Due to their inherent simplicity, decision trees may also be seen in a lot of video games. These trees, however, are frequently concealed by implementations made up of nested if/else clauses. The constrained and hurried game production timelines are mostly to blame for the employment of such tactics [29]. Here are some in-depth explanations of game AI terminology and techniques:

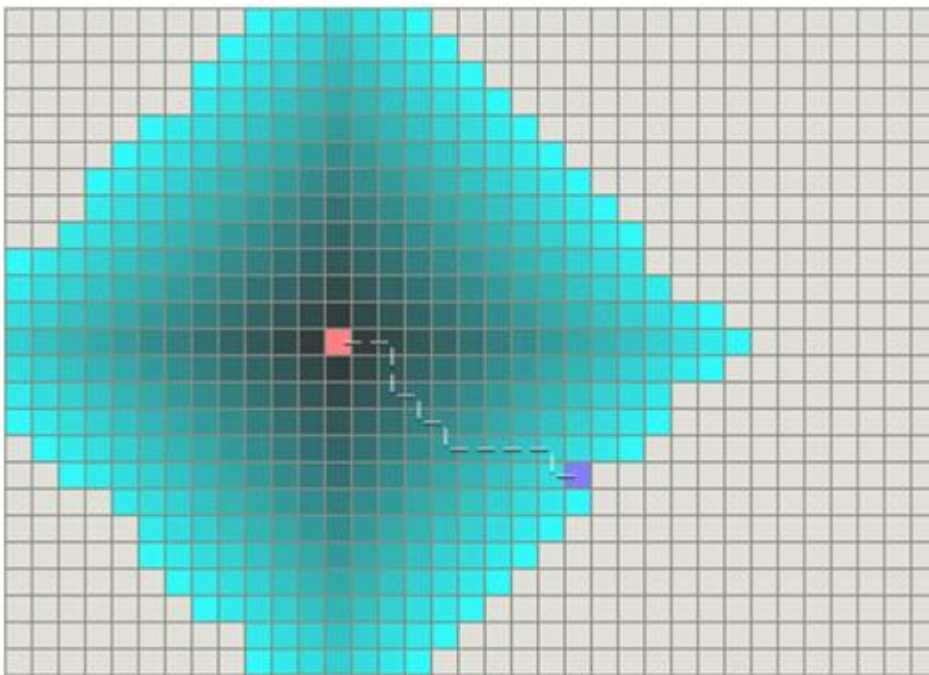
1. NPC Control

a. Agents: An agent is a type of object that has the ability to gather information about its surroundings and base decisions on it in order to accomplish a goal. An agent in a video game frequently depicts a non-player character (NPC), but it can also stand in for a more complex system, such a rival in a strategy game.

b. States: An agent's location's environment can be described as being in a particular state. When an agent (or the player) takes an action, the state may change. The set of potential states for an agent is referred to as its state space.

This is a crucial principle because the main goal of the majority of AI techniques is to browse or explore the state space in order to choose the optimal course of action given the current circumstances. Thus, the type of AI techniques that can be applied is influenced by the state space's properties.

c. Pathfinding: It is vital to specify the NPC's range of motion within the game world as well as how it can interact with its surroundings. Finding the shortest route between two points, or pathfinding, is thus frequently one of the fundamental components of sophisticated AI systems. The A* algorithm is the most used pathfinding algorithm. It is a quicker variant of the Dijkstra algorithm, which is an earlier algorithm. The basic idea behind Dijkstra's method is very straightforward: he will proceed through each grid cell in the order determined by how far it is from the starting cell until he finds the target cell.

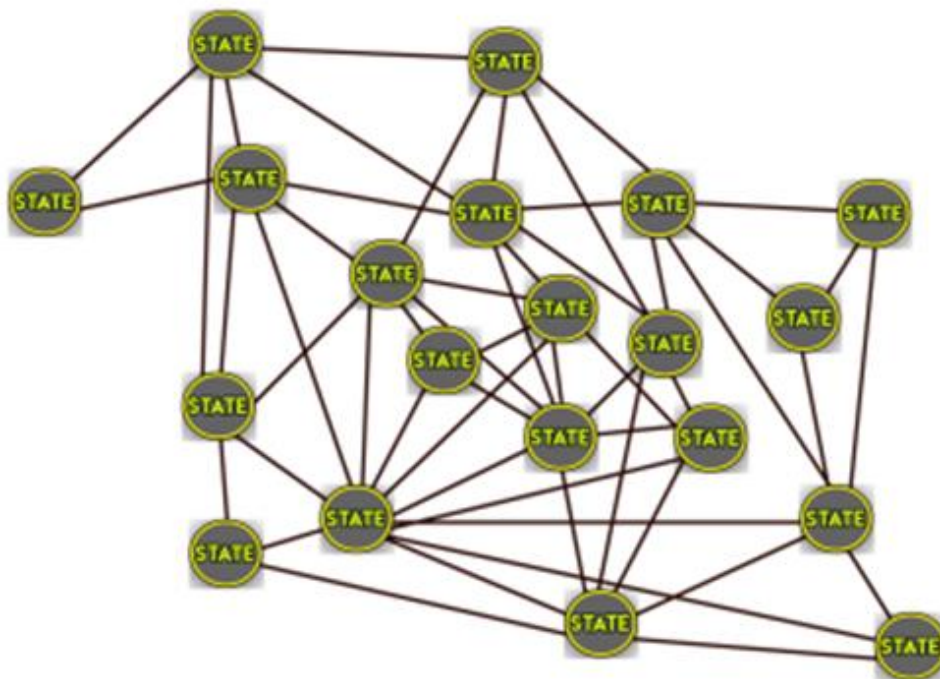


A* may only find an approximation of the solution, but Dijkstra always gets the best solution. However, he will complete the task far more quickly, which is important in a computer game. The method estimates roughly the general direction that he has to travel to get closer to the destination in addition to knowing the distance between each square and the starting square. Then, it will attempt to traverse the graph, always moving in the direction of the objective, which enables much quicker discovery of a solution.

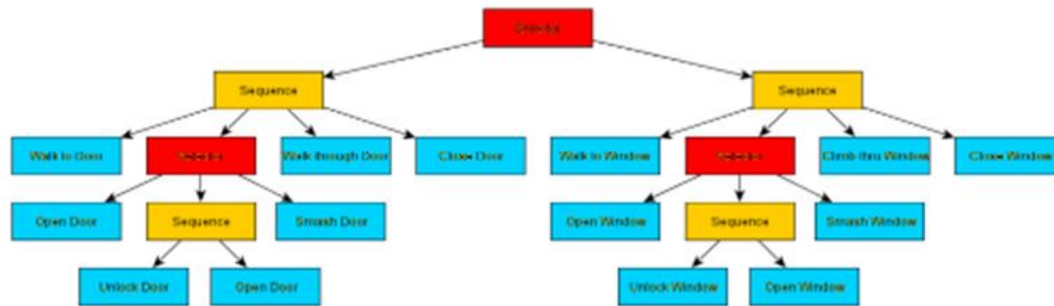
The Jump Point Search (JPS) approach was developed as an alternative to the A* algorithm. In contrast to A*, which can only search for its path square by square, it permits searching under specific conditions to skip multiple squares at once and save time. Today's pathfinding systems in video games are frequently modifications of A* or JPS that have been tailored to the unique challenges of each game. [30]

2. Ad Hoc Behavior Authoring: Ad Hoc methods are expert systems, where a set of principles that will be used to describe the AI behaviour must be manually defined. Therefore, the methods for navigating the state space are manual. Ad-hoc approaches, in contrast to search algorithms like A*, may not really be classified as traditional AI, but the industry of video games has long regarded them as such.

a. Finite state machine: A finite automata enables the manual definition of an agent's goals and stimuli in order to control its behavior. One or more states are described for each objective, and for each state, the NPC must take one or more actions. The set of an agent's states along with all of the transitions between them can therefore be represented by a finite automaton. The agent will always behave accordingly if the states and transitions are clearly stated. Since you only need to manually write the rules that define the many states and transitions, finite automata's key benefit is that they make it easy and rapid to simulate a relevant behavior. They are also useful since they provide you a visual representation of how an NPC could act. A finite automaton, on the other hand, lacks flexibility since, once it has been developed and put into use, the AI's behavior cannot change or evolve. The NPC will consistently do the same activities to get the same goal. Typically, a flow diagram is used to illustrate finite automaton. [30]



b. Behavior trees: Finite automaton and behavior trees are extremely similar, however behavior trees are based solely on actions (rather than states) and express the many transitions between actions as a tree. The representational structure of a trees make it considerably easier to implement complicated activities. Additionally, it is easier to maintain since browsing the tree rather than using a flow chart gives one a clearer notion of the agent's potential courses of action. [30]



2.8 AR, AI, and Games Relation

AI algorithms may be used by AR developers to deliver features like improved interaction with the actual environment.

Artificial intelligence (AI) techniques like machine learning and deep learning are ideal for AR settings because:

- Since cameras are continuously on, there is chance to gather additional data for the AI algorithm to learn from.
- Due to the fact that AR settings rely on several sensors, including GPS, the input to the AI system is extensive. Compared to systems that rely solely on a single sensor, this offers more reliability.[31]

Based on the information above, we can conclude that artificial intelligence is also very helpful for augmented reality games because these games use virtual assets that are superimposed over the real world, necessitating the use of artificial intelligence to detect and react to changes in the real world. One cannot presume a constant world that is only susceptible to the player's in-game choices. The aforementioned justifications for artificial intelligence in games and its applications also apply to augmented reality games. [32]

3. Conclusion

In summary, in an environment where the game industry is growing, it is inevitable that AR games will also be popular. AR-style games add more realism to games and draw users more in. For example, thanks to Pokemon GO, people have found the experience of reliving their childhood feelings thanks to AR game. Based on this idea, we redesigned the Beyblade game with AR technology and aimed to make it a game closer to a more realistic player experience with AI. In light of all this, our research focuses on AR Beyblade with AI to achieve the team purpose or desired results.

4. References

- [1] <https://en.wikipedia.org/wiki/Beyblade>
- [2] <https://core.ac.uk/download/pdf/84832291.pdf>
- [3] <https://pdfs.semanticscholar.org/b656/0c35ce1f0484cc2fc75cada34b580953c9ff.pdf>
- [4] <https://www.incredibuild.com/blog/top-7-gaming-engines-you-should-consider>
- [5] <https://documentation.help/Photon-Unity-Networking-1.91/general.html>
- [6] https://www.researchgate.net/profile/Pratik-Patil-11/publication/312591645_Cross-platform_Application_Development_using_Unity_Game_Engine/links/5884d39faca272b7b44a847f/Cross-platform-Application-Development-using-Unity-Game-Engine.pdf
- [7] https://www.theseus.fi/bitstream/handle/10024/108970/Jetsonen_Timo.pdf?sequence=1
- [8] <https://docs-multiplayer.unity3d.com/>
- [9] John K. Haas, A History of the Unity Game Engine, March 2014
- [10] <https://docs.unity3d.com/Manual/AssetWorkflow.html>
- [11] <https://docs.unity3d.com/Manual/PhysicsSection.html>
- [12] <https://hub.packtpub.com/harrison-ferrone-why-c-preferred-programming-language-building-games-unity/>
- [13] <https://docs.unity3d.com/Manual/AnimationSection.html>
- [14] <https://www.investopedia.com/terms/a/augmented-reality.asp>
- [15] <https://ieeexplore.ieee.org/abstract/document/666641>
- [16] https://techinnovations.info/why-every-organization-needs-an-augmented-reality-strategy/?gclid=EAIaIQobChMIutnf4MSc-wIViJeyCh0p1gljEAAAYASAAEgLcqfD_BwE
- [17] <https://www.myayan.com/advantages-and-disadvantages-of-ar-augmented-reality>
- [18] <https://www.modelry.ai/blog/arkit-vs-arcore#:~:text=Both%20kits%20have%20their%20strengths,alongside%20the%20hardware%20they%20serve>
- [19] <https://www.makeuseof.com/is-arcore-or-arkit-better-for-augmented-reality/>

- [20] <https://developers.google.com/ar/develop>
- [21] <https://medium.com/mindorks/what-is-arcore-by-all-you-need-to-know-to-get-started-b3715734cdd3>
- [22] <https://www.tekrevol.com/blogs/22-best-augmented-reality-games/>
- [23] <http://devfun-lab.com/1240>
- [24] <https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp>
- [25] <https://www.javatpoint.com/application-of-ai>
- [26] <https://www.arm.com/glossary/ai-in-gaming#:~:text=AI%20in%20gaming%20refers%20to,behavior%20in%20the%20game%20world.>
- [27] <https://bernardmarr.com/the-amazing-ways-ai-and-machine-learning-are-used-in-the-gaming-industry/>
- [28] <https://pianalytix.com/role-of-artificial-intelligence-in-gaming/>
- [29] https://www.michelepirovano.com/pdf/fuzzy_ai_in_games.pdf
- [30] <https://www.saagie.com/blog/artificial-intelligence-in-video-games/>
- [31] <https://research.aimultiple.com/ar-ai/#:~:text=AR%20software%20derives%20information%20about,text%20analysis%2C%20a nd%20scene%20labeling>
- [32] <https://mark-riedl.medium.com/augmented-reality-games-need-artificial-intelligence-b8f57ec4e5a9#:~:text=Because%20the%20game%20plays%20out,game%20actions%20of%20the%20player.>
- [33] K. Salen and E. Zimmerman, Rules of Play: Game Design Fundamentals, vol. 1, MIT Press, 2004.
- [34] https://en.wikipedia.org/wiki/Augmented_reality

5. Software Requirement Specification (SRS)

5.1 Introduction

Strategy games or strategy games are games in which the casual and often autonomous decision-making skills of the player are of great importance in determining the outcome. Almost all strategy games require an internal decision-tree style of thinking and usually a very high level of situational awareness. Strategy games are also considered descendants of war games and define strategy in terms of the context of war, although this is more partisan. The definition of a strategy game in its cultural context belongs to a tradition that goes back to war games, involves more strategy than the average video game, incorporates the conventions of a particular game, and is represented by a particular community. should be a game.[1] Strategy games use systems powered by artificial intelligence. That's why game manufacturers use artificial intelligence algorithms that can act in the most realistic way in this game genre. Our project will be to play the strategic augmented reality game. In this part of the report, we explained our purpose, the scope of our project, target audience characteristics, glossary, and finally an overview of our SRS document.

5.1.1 Purpose of the Document

In this project, our aim is to revive and develop the nostalgic Beyblade game with artificial intelligence and multiplayer using augmented reality. This strategic game will bring people of all ages back to the past and offer a competitive game experience. While doing the project, we design it to meet various software requirements such as Artificial Intelligence, Game Engine Physics and Augmented Reality for our game. In single-game mode, the user plays against the AI, and the AI uses damage against the user according to the user's skill usage or at his own will. In multiplayer mode, people launch their Beyblades and fight in real-time. This SRS document contains the project requirements and the Software Requirements Specification for Beyblade AR game with artificial intelligence.

5.1.2 Scope of Project

In our project, it is aimed to develop the strategic Beyblade game played on the arena, which we place on a 3D virtual plane integrated into the real world. Our project can be played both in single and multiplayer. In the single-player game, the difficulty of the game can be adjusted so that the artificial intelligence will fight the user accordingly.

The AI will use damage against the user according to the user's skill usage or at his own will. In multiplayer, players will be able to join the same lobby and play against each other. We will combine multiple engineering disciplines to address these applications. Since our game has both single and multiplayer features, we will learn the use of artificial intelligence and networking for game engines. We will use the Unity artificial intelligence library and algorithms when designing the opponent's brain, and we will use the Unity Photon Network tool and algorithms for multi-game lobbies and connections.

Our target audience in this project is those who are interested in the world of augmented reality and metaverse and want to have a strategic gaming experience. Thanks to the user-based design principles, people will enjoy playing the game without any difficulties from the moment they enter the game.

5.1.3 Glossary

Term	Definition
Player	The person who plays the game.
Unity 3D	A cross-platform game engine developed by Unity Technologies.
C#	It is a widely used programming language for website applications, desktop applications, and game creation in general.
Augmented Reality (AR)	It is the manipulation and augmentation of reality by the computer.
Singleplayer	A game mode in which only one player plays the game.
Multiplayer	A game mode in which many players play the game.
Artificial Intelligence (AI)	The ability of a computer or computer-controlled robot to perform tasks often associated with intelligent beings.
Game Engine	Software framework designed for the development of video games.
Agile Software	It is a specific approach to project management used in software development.

5.1.4 References

We used IEEE Std 830™-1998(R2009) Recommended Practice for Software Requirements Specifications.

5.1.5 Overview of Documents

The document consists of 3 main titles.

- The first title gives information about the document, such as the articles used as references and the terms used in the article.
- The second title introduces the system. Function properties used, user classes characteristics, and requirements for creating the system.
- The last title gives a detailed introduction to the requirements, such as the interface requirements used and the explanation of the requirements in detail.

5.2 Overall Description

This part will clarify of the principal aspect of Beyblade Augmented Reality system and necessities.

5.2.1 Product Perspective

Our system includes multiple software subsystems. We also used multiple computer engineering disciplines as we aimed to promote our system as a product. On the other hand, our project includes 4 main computer science concepts: augmented reality algorithms that detect 3D environment, artificial intelligence algorithms, network algorithms and advanced C# game algorithms. We will use the C# programming language for these techniques. Also, there are different tools and IDEs used for the development process. In this section, we have summarized which features are included in the software. Our Beyblade AR project is divided into two parts: single player mode and multiplayer mode.

Single player mode is a mode in which the player chooses the opponent's strength from the game difficulty selections and fights against artificial intelligence. Both players (user and AI) can use abilities and apply a strategic path.

Multiplayer mode is a mode where people enter a virtual room and play in real time.

5.2.1.1 Development Methodology

The project was to be developed using Scrum, an agile software development methodology. Scrum is incremental and iterative. In Scrum, major work is divided into sprints and must be completed within a specific time period (average 30 days). Since Scrum is an agile development methodology, each sprint should have the same iteration length. Each sprint contains tasks with their own story points and risk points. The development team should meet every morning for up to 15 minutes. Scrum has his three main roles: Product Owner, Scrum Master and Development Team. The Product Owner is the person who provides the requirements, and the Scrum Master is the person who leads the development team. A development team is a group of developers working on a project as planned. Scrum has several advantages.

The first advantage is that short sprints and continuous feedback make it easier to manage change. Another advantage is that the morning meeting allows us to resolve issues quickly. It also allows us to produce quality products within the planned time.

5.2.2 User Characteristic

5.2.2.1 Players

- Players must have a smartphone.
- Players should make sure their smartphones comply with the system requirements.
- The player must be able to read and understand the English language at a certain level.
- Players need to know the use of smartphones to a certain extent and log in to the application.

5.2.2.2 System

- The system starts the game when the player enters the game.
- When the player wants to change anything in the settings, the system adjusts the game according to the player's wishes.
- When the player wants to play multiplayer, it finds the opponent player in the background and starts the match.
- When the player wants to exit the game, it removes all the load on the system.

5.2.3 Requirements Specification

5.2.3.1 External Interface Requirements

5.2.3.1.1 User Interfaces:The user interface will be worked on smart phones(iOS and Android).

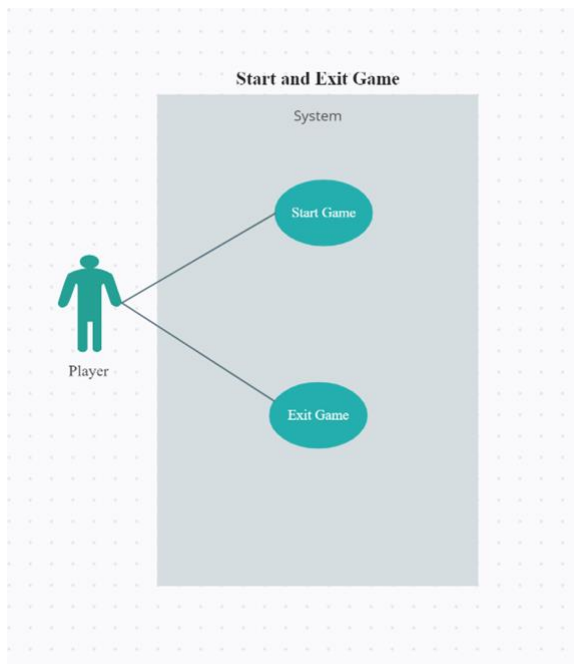
5.2.3.1.2 Hardware Interfaces:We will use smart phones' camera sensor.

5.2.3.1.3 Software Interfaces:We will use intellij, VS Code as editor and Unity.

5.2.3.1.4 Communication Interfaces:The application will communicate with the server via HTTP protocol over internet.

5.2.4 Functional Requirements

5.2.4.1 Start/Exit Game Use Case



Brief Description: This use case shows the game entry and exit use case. If the player downloads the game to his/her phone from the Appstore or Google Play, he/she can open the game by clicking the game icon on the main menu of the phone. The player may or may not choose to play at his discretion. To leave the game, the player must click on the 'Exit' button.

Initial Step by Step Description:

Use Case Name: Start and Exit Game

Use Case Number: UC 1

Actors: Players

Typical Flow Description:

- **Pre-Condition:** The player should have smart phone and download the game.
- **Post Condition:** The player logs in and out of game.

Main Flow

1. The player clicks on the game downloaded from Appstore or Google Play via the main menu. (A1)
2. The player clicks the 'Start Game' button. (A2)
3. After the round has finished, player clicks the 'Exit ' button. (A3)

Alternative Flow Description

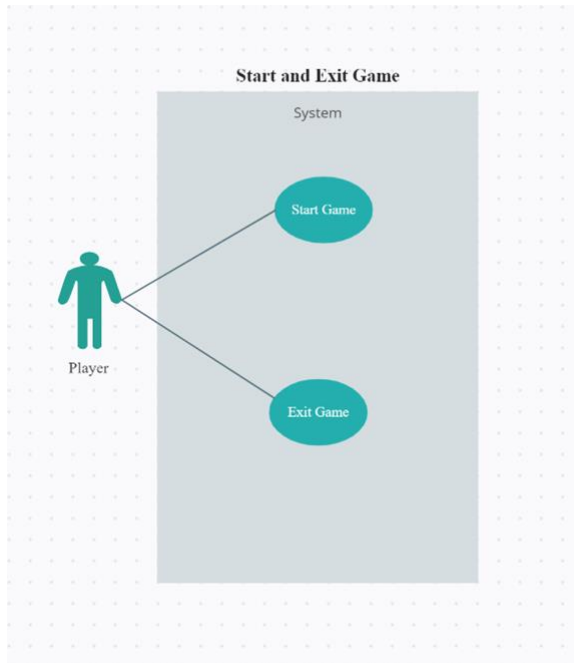
A1- If the player did not download the game, he/she must download it first.

A2- The player clicks on exit button directly.

A3.1- The player continues with 'Start Game' button.

A3.2- If the player wants to quit before the round has finished, he/she clicks 'Options' button and chose to exit the game or he/she chooses not to play at all and directly exit

5.2.4.2 Login and Register Game Use Case



Brief Description: This use case shows register and login to the system. The player has to register first and then login into his/her account which will be saved by the system.

Initial Step by Step Description:

Use Case Name: Login & Register the Game

Use Case Number : UC2

Actors: Players

Overview: This use case captures a player of the system registers and logs the system

Typical Flow Description:

- **Pre-Condition:** The player should have smart phone and download the game.
- **Post Condition:** The player registers and logs in the game.

Main Flow

1. The player clicks on the game downloaded from Appstore or Google Play via the main menu. (A1)
2. The player fills in the blank settings to enter his/her own information and create password. (A2)
3. After system admin authenticates the account, player can login to the game with e-mail address and password. (A3)
4. The system automatically saves account and make account stayed in.(A4)

Alternative Flow Description

A1- If the player did not download the game, he/she must download it first.

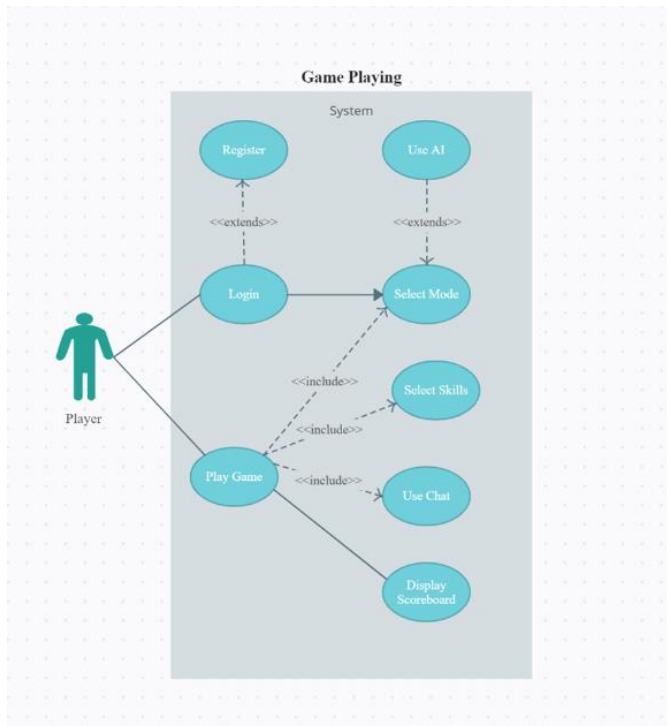
A2- If player enters his/her wrong email address, the system asks for valid e-mail address again.

A3.1- If player enters wrong information, the system asks again for information.

A3.2- If player forgets password, clicks 'Forget Password' button.

A4- If the player wants to change his/her login information or log out, can change it from options 'Account Settings'.

5.2.4.3 Game Playing Use Case



Brief Description: This use case shows single player and multiplayer game mode. In this game mode, the player who has downloaded and registered the game enters and starts the game. Before starting the game, he/she chooses 'noob' and 'pro' game difficulty modes. After started the game, the player can choose game skills. Also, the player can view game score. After the game is completed, the winner is given to the screen. The player can log out or continue with a new game again.

Initial Step by Step Description:

Use Case Name : Single Player Mode

Use Case Number : UC 3

Actors: Players, AI

Overview: This use case captures a player of the system enters and plays the game.

Related Use Cases: Select Difficulty, Select Skills

Typical Flow Description:

- **Pre-Condition:** The player should have smart phone and download the game.
- **Post Condition:** The player plays the game.

Main Flow

1. The player clicks on the game downloaded from Appstore or Google Play via the main menu. (A1)
2. The player clicks the 'Play Game' button. (A2)
3. The player selects 'Single Player Mode'.
4. The player selects game difficulty level. (A3)
5. The player selects the Beyblade according to his/her needs.
6. The player starts the game with 'Throwing Beyblade'.
7. While player is playing the game, he/she can select the skills which needed in game. (A4)
8. After the round has finished, the system shows winner via 'Scoreboard'. (A5)
9. Player clicks the 'Exit' button. (A6)

Alternative Flow Description

A1-If the player did not download the game, he/she must download it first.

A2- The player clicks on exit button directly.

A3.1-The player selects the level as "Noob" for easier game.

A3.2- The player selects the level as "Pro" for harder game.

A4- The player prefers not to use any skills.

A5- The player exits before round has finished from 'Options' menu.

A6- The player can continue with 'Start Game' button.

5.2.4.4 Multiplayer Mode Use Case

Initial Step by Step Description:

Use Case Name: Multi Player Mode

Use Case Number: UC 4

Actors: Players

Overview: This use case captures a player of the system enters and plays the game with other players.

Related Use Cases: Select Difficulty, Select Skills

Typical Flow Description:

- **Pre-Condition:** The player should have smart phone and download the game.
- **Post Condition:** The player plays the game.

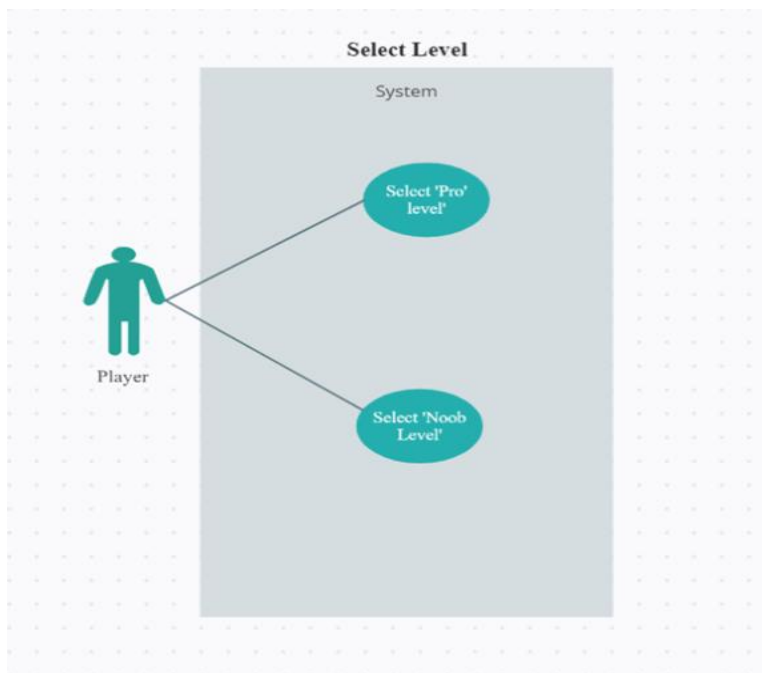
Main Flow

1. The player clicks on the game downloaded from Appstore or Google Play via the main menu. (A1)
2. The player clicks the 'Play Game' button. (A2)
3. The player selects 'Multiplayer Mode'.
4. The player selects the room. (A3)
5. The player clicks "I am ready to play" button. (A4)
6. In the playroom, the screen that allows users to send messages to each other appears. (A5)
7. The player selects the Beyblade.
8. The player throws the Beyblade.
9. While player playing the game, he/she can select the skills which needed. (A6)
10. After the round has finished, the system shows winner. (A7)
11. The player clicks the 'Exit Room' button. (A8)

Alternative Flow Description

- A1- If the player did not download the game, he/she must download it first.
- A2- The player clicks on exit button directly.
- A3- If the player does not want to select room, system assign room for user.
- A4- The player press "Back" button to not to play.
- A5- Players prefer not to use chat feature.
- A6- The player prefers not to use any skills.
- A7- The player exits before round has finished from 'Options' menu.
- A8- The players continue with 'Start Game' button.

5.2.4.5 *Select Level Use Case*



Brief Description: This use case shows selecting level of game difficulty. In this game mode, the player who has entered the game, can select game level according to his/her ability or what they want.

Initial Step by Step Description:

Use Case Name: Select Level

Use Case Number: UC 5

Actors: Players, AI

Overview: This use case captures a player of the system chooses the difficulty of game.

Related Use Cases: Single Mode Player, Multiplayer Mode

Typical Flow Description:

- **Pre-Condition:** The player must be entered the 'Single Player Mode'.
- **Post Condition:** The player selects the level of the game.

Main Flow

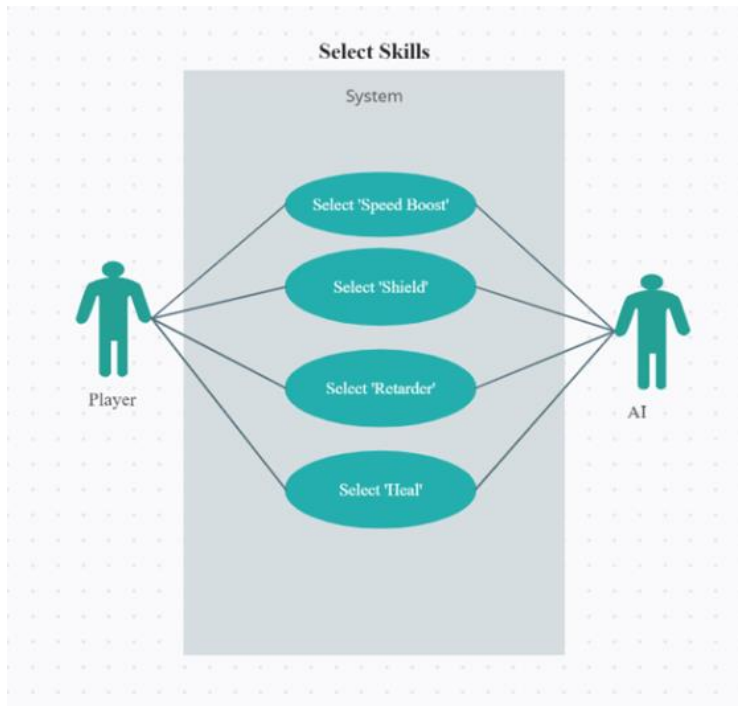
1. The player selects game difficulty level. (A1)
2. The player completes selecting level of the game difficulty.

Alternative Flow Description

A1.1- The player selects the level as "Noob" for easier game.

A1.2- The player selects the level as "Pro" for harder game.

5.2.4.6 Select Skill Use Case



Brief Description: This use case shows the selecting skills. In this game mode, the player who has started to play the game can attack enemy or defense his/her Beyblade with skills. There are four different skills which provide speed boost, shield, and heal for player's Beyblade and one for blocking the skill from enemy.

Initial Step by Step Description:

Use Case Name: Select Skills

Use Case Number: UC 6

Actors: Players, AI

Overview: This use case captures a player of the system chooses the difficulty of game.

Related Use Cases: Single Mode Player, Multiplayer Mode

Typical Flow Description:

- **Pre-Condition:** The player must be started the game.
- **Post Condition:** The player selects the skills to use in the game.

Main Flows

1. The player can choose the skill according to his/her needs in the game.

Alternative Flow Description

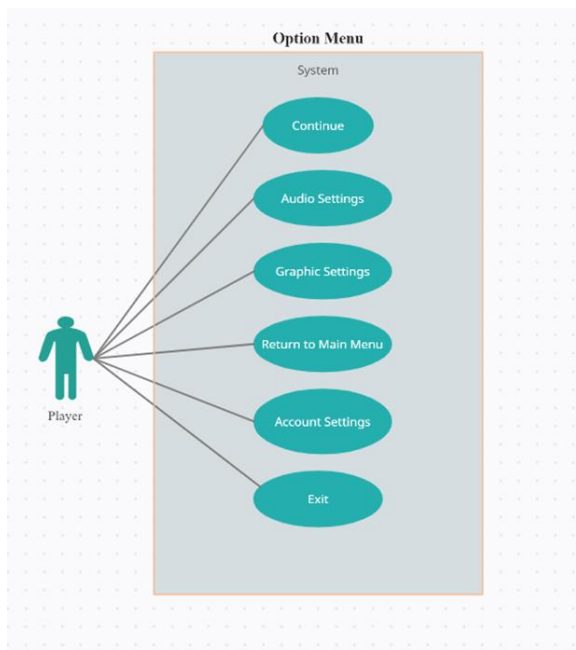
A1.1- The player selects the level as “Speed Boost” for boosting his/her speed in game.

A1.2- The player selects the level as “Shield” for protect his/her Beyblade from attacks from AI or other attackers’ skills.

A1.3- The player selects the level as “Retarder” for decreasing the speed of enemy Beyblade.

A1.4- The player selects the level as “Heal” for increasing Beyblade’s health.

5.2.4.7 Option Use Case



Brief Description: This use case shows option menu diagram. When players entered the game or staying in game lobby, he/she can display the options menu. The player can execute functions of continue, audio settings, graphic settings, account settings or return back to main menu and exit game according to their needs.

Initial Step by Step Description:

Use Case Name: Option

Use Case Number: UC 7

Actors: Players

Overview: This use case captures a player of the system changes the game settings.

Typical Flow Description:

- **Pre-Condition:** The player should have smart phone and download the game.
- **Post Condition:** The player changes the settings.

Main Flow

1. The player clicks on the game downloaded from Appstore or Google Play via the main menu. (A1)
2. The player clicks the 'Options' button.
3. The player chooses to continue to exit options menu.
4. The player clicks 'Audio Settings' to adjust the game volume and increase or decrease the game volume. (A2)
5. The player clicks 'Account Settings' to log out or change login information.
6. The player clicks 'Graphic Settings' to adjust the graphic settings and set higher or lower resolution. (A3)
7. The player clicks 'Return to Main Menu' to exit from game and go back to main menu.
8. The player exits the game. (A4)

Alternative Flow Description

A1- If the player did not download the game, he/she must download it first.

A2.1- The player increases volume of the game by selecting "+" button.

A2.2- The player decreases volume of the game by selecting "-" button.

A2.3- The player chooses to do not make any change and press 'Back' button and go back.

A3.1- The player selects resolution of the game as "Higher".

A3.2- The player selects resolution of the game as "Lower".

A3.3- The player chooses to do not make any change and press 'Back' button and go back.

A4- The player continues with 'Start Game' button.

5.2.5 Performance Requirement

5.2.5.1 Android

- Android 6 or above.
- Preferred resolution of 720×1280 pixels.
- Strong internet connection (Wi-Fi, 3G, 4G or 5G).
- 5.2.5.2. iOS
- iPhone 6s or above.
- iOS 12 or above.
- Strong internet connection (Wi-Fi, 3G, 4G or 5G).

We advise utilizing phones with 2GB or more RAM for the best Beyblade AR Game experience. We advise trying the following troubleshooting actions if your device is frequently crashing or underperforming while you are playing Beyblade AR Game:

- Make that your device is running the most recent version of the operating system (such as iOS or Android) available.
- While playing Beyblade AR Game, close any other open applications.

5.2.6 Software System Attributes

Attributes	Description
Portability	The game can be played on iOS, iPadOS and Android operating systems.
Performance	If the system requirements specified in the documentation are met, the player can play the game more comfortably.
Usability	Using AR technology, the player must create the ground on which player wants to play. Before the game starts, the player can choose one of the single, multi, or artificial intelligence options from the menu section. The player can choose the room in which player wants to play the game.
Adaptability	There is no requirement for adaptability because no data from the run time is collected and saved.

5.2.7 Safety Requirement

At least 1 square meter of space is required to play Beyblade1999. This space is required for AR technology to scan. When Beyblade1999 is played for a long time, it can cause sleep disorders, posture disorders, stress and eye problems.

5.3 References

[1] Software Requirements Specification: "Simulacrum: Simulated Virtual Reality for Emergency Medical Intervention in Battle Field Conditions"

6. Software Design Description (SDD)

6.1 Introduction

Strategy games or strategy games are games in which the casual and often autonomous decision-making skills of the player are of great importance in determining the outcome. Almost all strategy games require an internal decision-tree style of thinking and usually a very high level of situational awareness. Strategy games are also considered descendants of war games and define strategy in terms of the context of war, although this is more partisan. The definition of a strategy game in its cultural context belongs to a tradition that goes back to war games, involves more strategy than the average video game, incorporates the conventions of a particular game, and is represented by a particular community. should be a game. Strategy games use systems powered by artificial intelligence.

That's why game manufacturers use artificial intelligence algorithms that can act in the most realistic way in this game genre. Our project will be to play the strategic augmented reality game. In this part of the report, we explained our purpose, the scope of our project, target audience characteristics, glossary, and finally an overview of our SDD document.

6.1.1 Purpose of This Document

In this project, our aim is to revive and develop the nostalgic Beyblade game with artificial intelligence and multiplayer using augmented reality. This strategic game will bring people of all ages back to the past and offer a competitive game experience. While doing the project, we design it to meet various software requirements such as Artificial Intelligence, Game Engine Physics and Augmented Reality for our game. In single-game mode, the user plays against the AI, and the AI uses damage against the user according to the user's skill usage or at his own will. In multiplayer mode, people launch their Beyblades and fight in real-time. This SDD document contains architectural and user interface design.

6.1.2 Scope of the Project

In our project, it is aimed to develop the strategic Beyblade game played on the arena, which we place on a 3D virtual plane integrated into the real world. Our project can be played both in single and multiplayer. In the single-player game, the difficulty of the game can be adjusted so that the artificial intelligence will fight the user accordingly.

The AI will use damage against the user according to the user's skill usage or at his own will. In multiplayer, players will be able to join the same lobby and play against each other. We will combine multiple engineering disciplines to address these applications. Since our game has both single and multiplayer features, we will learn the use of artificial intelligence and networking for game engines. We will use the Unity artificial intelligence library and algorithms when designing the opponent's brain, and we will use the Unity Photon Network tool and algorithms for multi-game lobbies and connections. Our target audience in this project is those who are interested in the world of augmented reality and metaverse and also want to have a strategic gaming experience. Thanks to the user-based design principles, people will enjoy playing the game without any difficulties from the moment they enter the game.

6.1.3 Glossary (Definitions, Acronyms, and Abbreviations)

Term	Definition
Player	The person who plays the game.
Unity 3D	A cross-platform game engine developed by Unity Technologies.
C#	It is a widely used programming language for website applications, desktop applications, and game creation in general.
Augmented Reality (AR)	It is the manipulation and augmentation of reality by the computer.
Singleplayer	A game mode in which only one player plays the game.
Multiplayer	A game mode in which many players play the game.
Artificial Intelligence (AI)	The ability of a computer or computer-controlled robot to perform tasks often associated with intelligent beings.
Game Engine	Software framework designed for the development of video games.
Agile Software	It is a specific approach to project management used in software development.

6.1.4 Overview of This Document

The document consists of 3 main titles.

- The first title gives information about the architectural design, such as technology used, activity and class diagrams.
- The second title contains user interface design.
- The last title contains requirement matrix.

6.1.5 Motivation

We are a group of senior students in the computer engineering department that are enthusiastic in gaming and augmented reality technology. For a deeper grasp of the gaming industry, our crew has attended online courses on creative game creation and reviewed documentation. Our goal was to bring together artificial intelligence, augmented reality, and gaming. To create our project, we have decided to use the Unity game engine, with which all of the group members are already familiar or have an interest. We have obtained augmented reality technology, and we will use IntelliJ with VS Code as the editor.

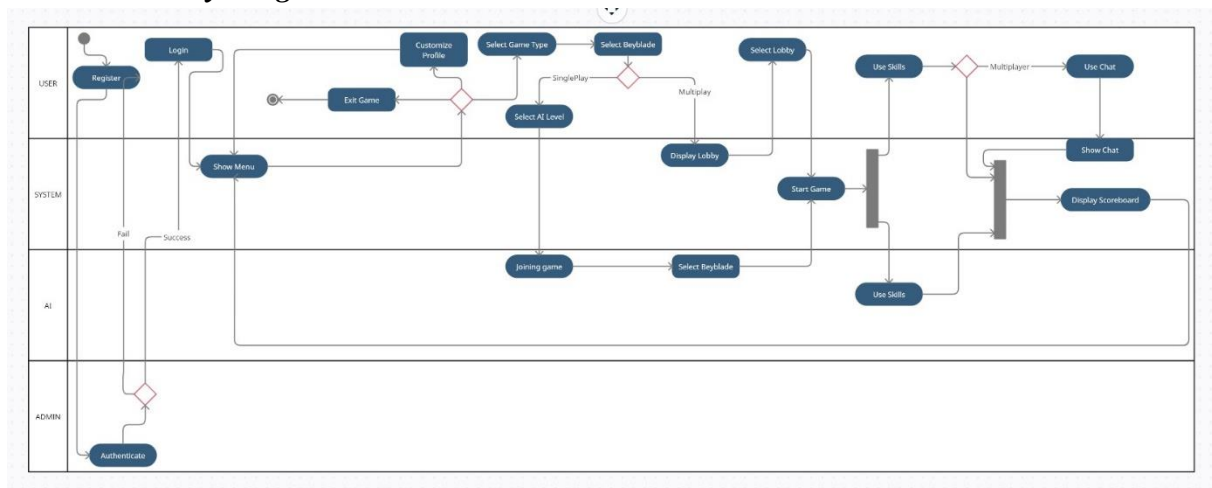
6.2 System Design

6.2.1 Architectural Design

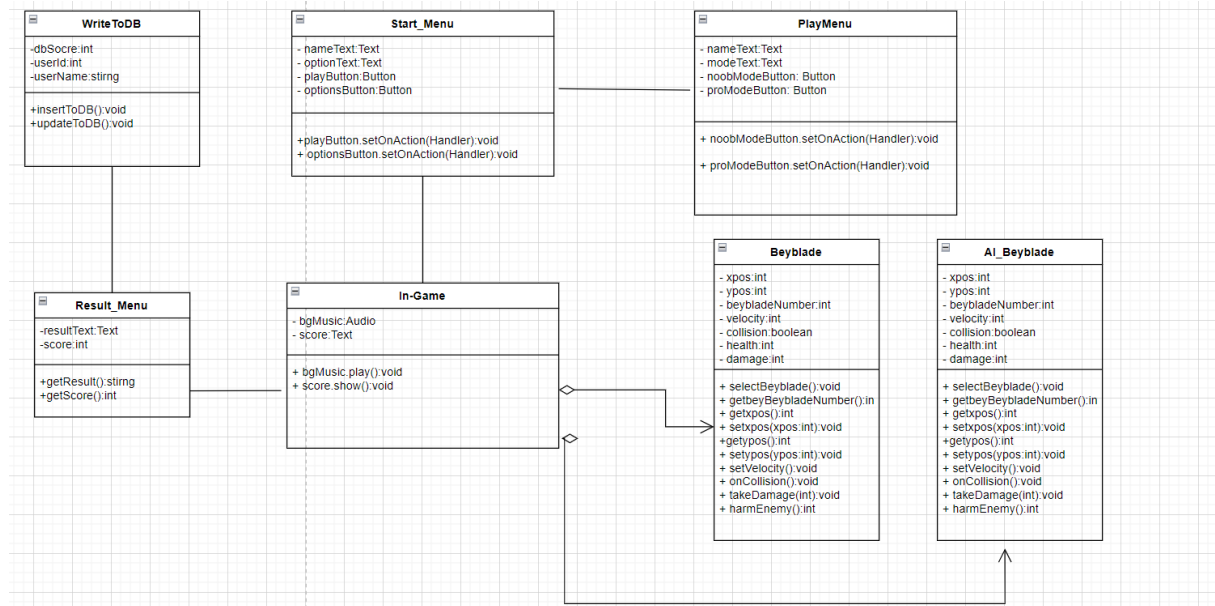
6.2.1.1 Technologies Used

- Unity Engine version 2021.3.12.
- JetBrains Rider IDE version 2022.2.3. We use JetBrains Rider to be able to run our code on our game project.
- C#. We use C# programming language for many mechanisms and workflow operations in our system.
- ARCore / ARKit Framework for implementing Augmented Reality.

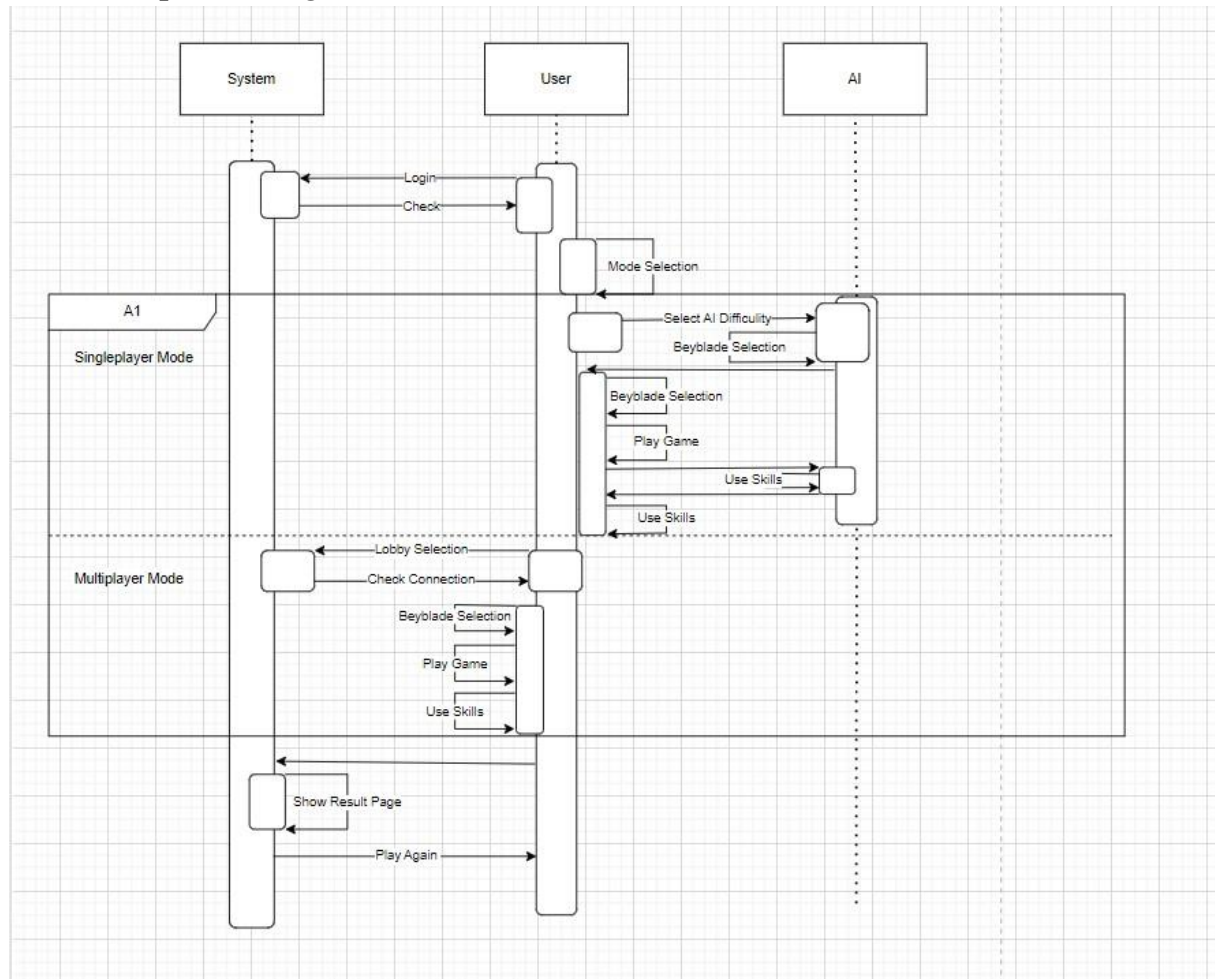
6.2.1.2 Activity Diagram



6.2.1.3 Class Diagram



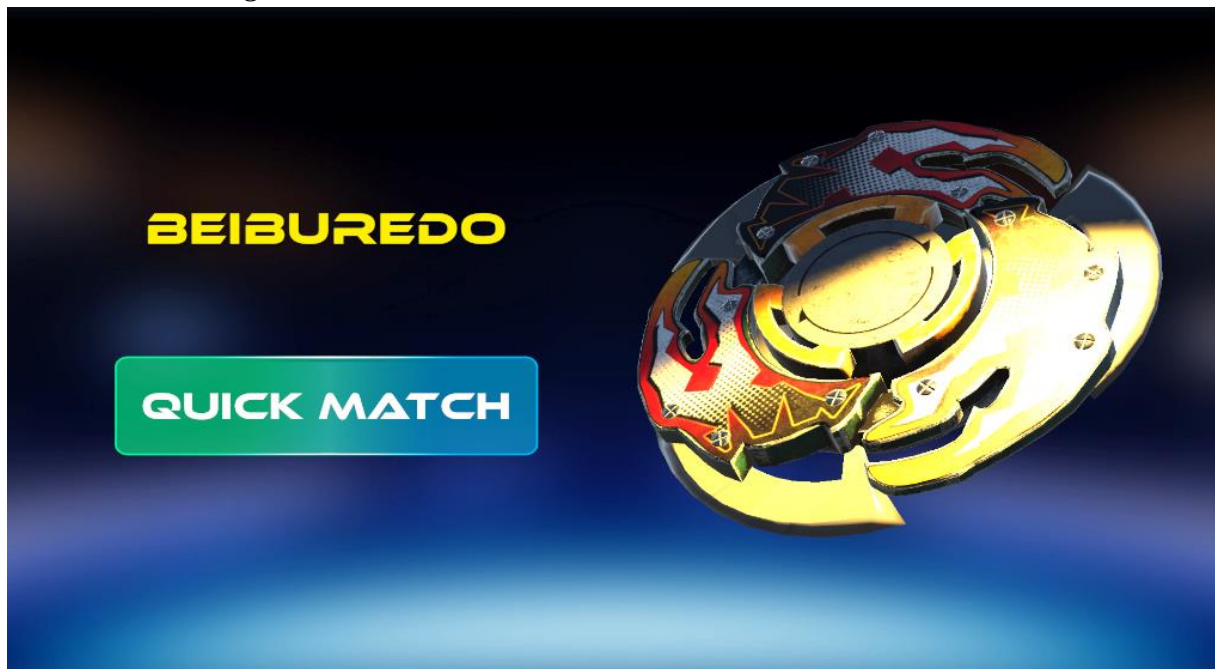
6.2.1.4 Sequence Diagram



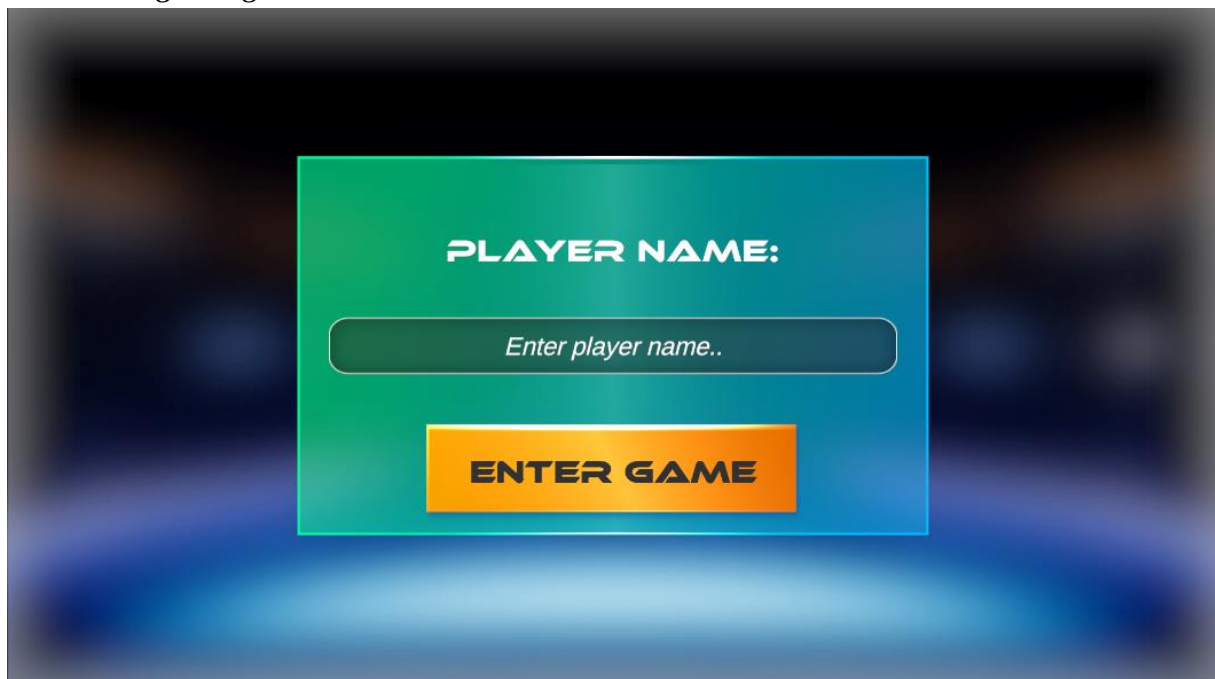
6.2.2 User Interface Design

In this section, we will state our designs for possible use cases and explain what features are included for each UI screen.

6.2.2.1 Home Page

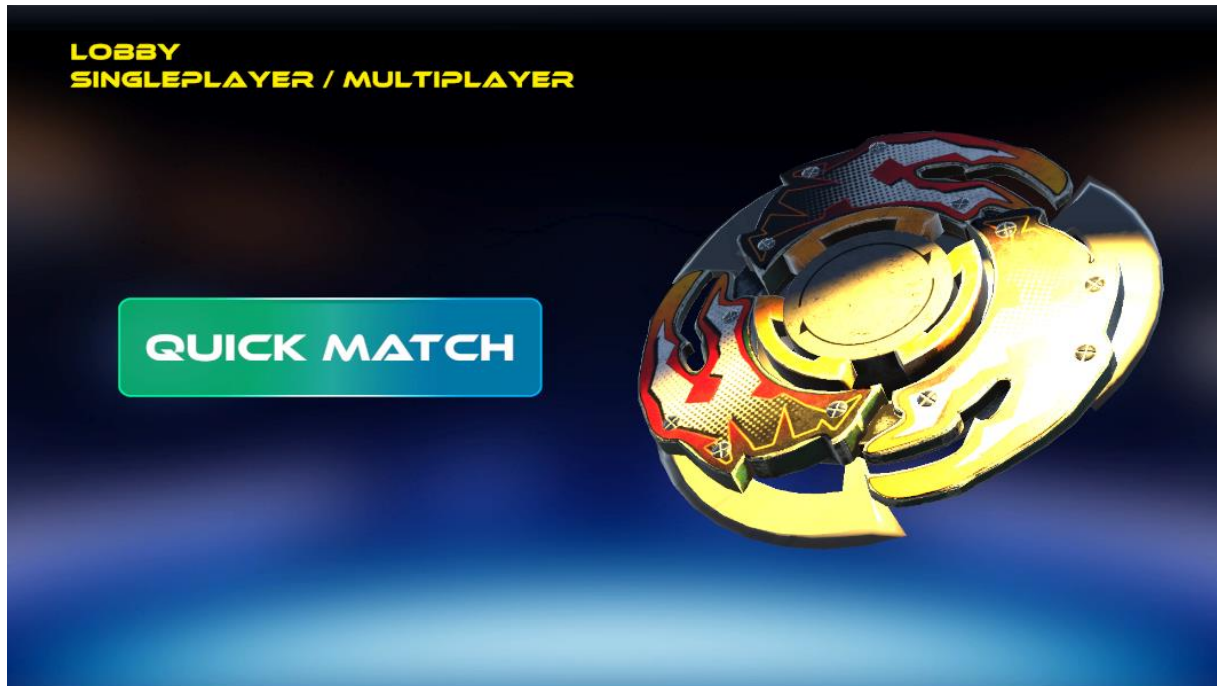


6.2.2.2 Login Page



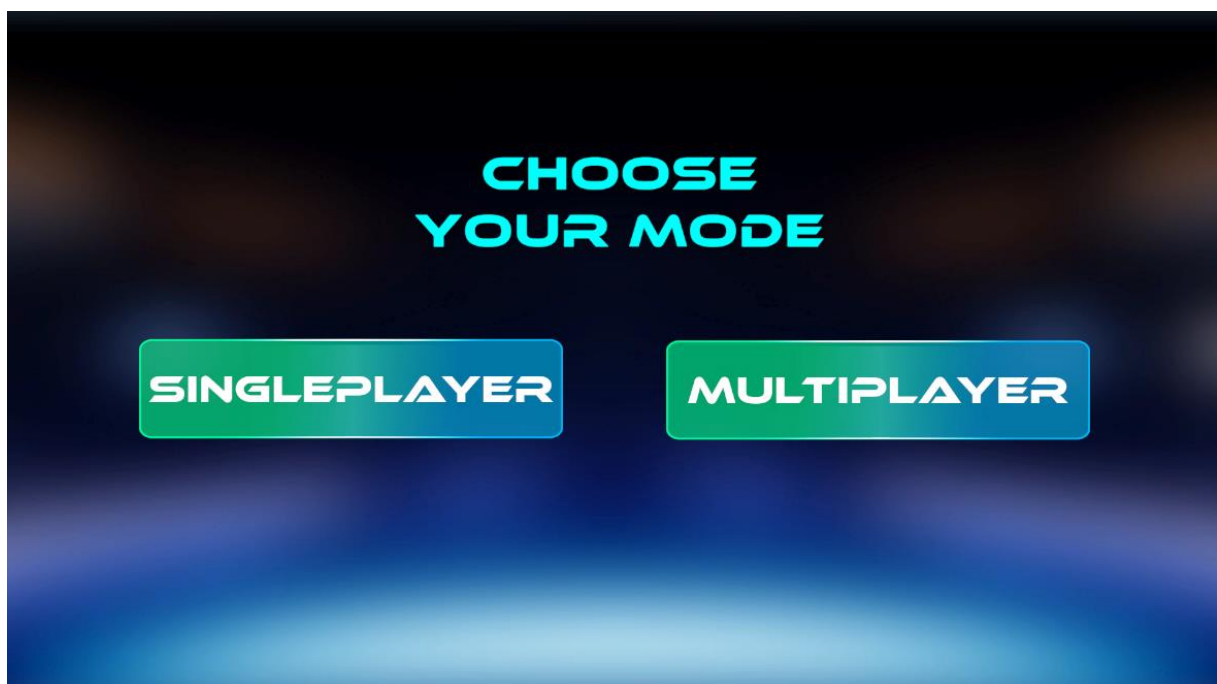
The user enters the user name specific to the game, if there is no username of the same name in the database, the user enters the room.

6.2.2.3 Lobby Page



In order to run single and multiplayer modes in our project, the user must choose which mode BeybladeAR will work in. Therefore, on the main page, we ask the user to choose a mode, single-player or multiplayer. Selecting any of these options will change the screen to one of the following screens.

6.2.2.4 Mode Selection

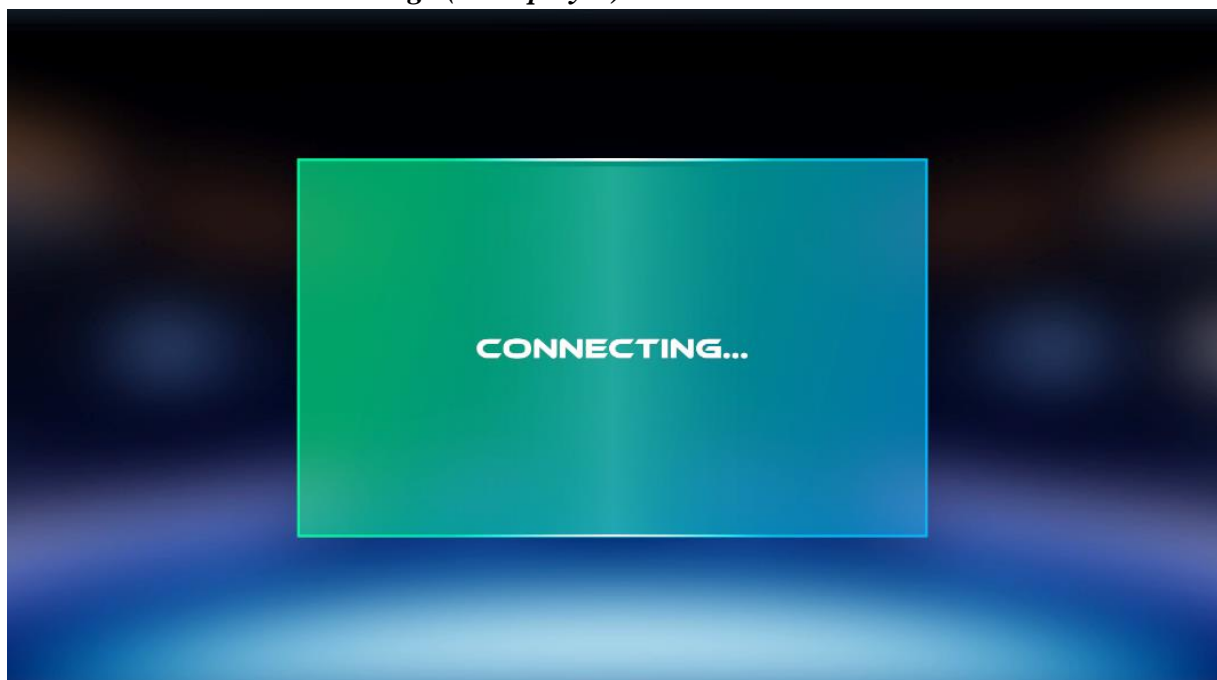


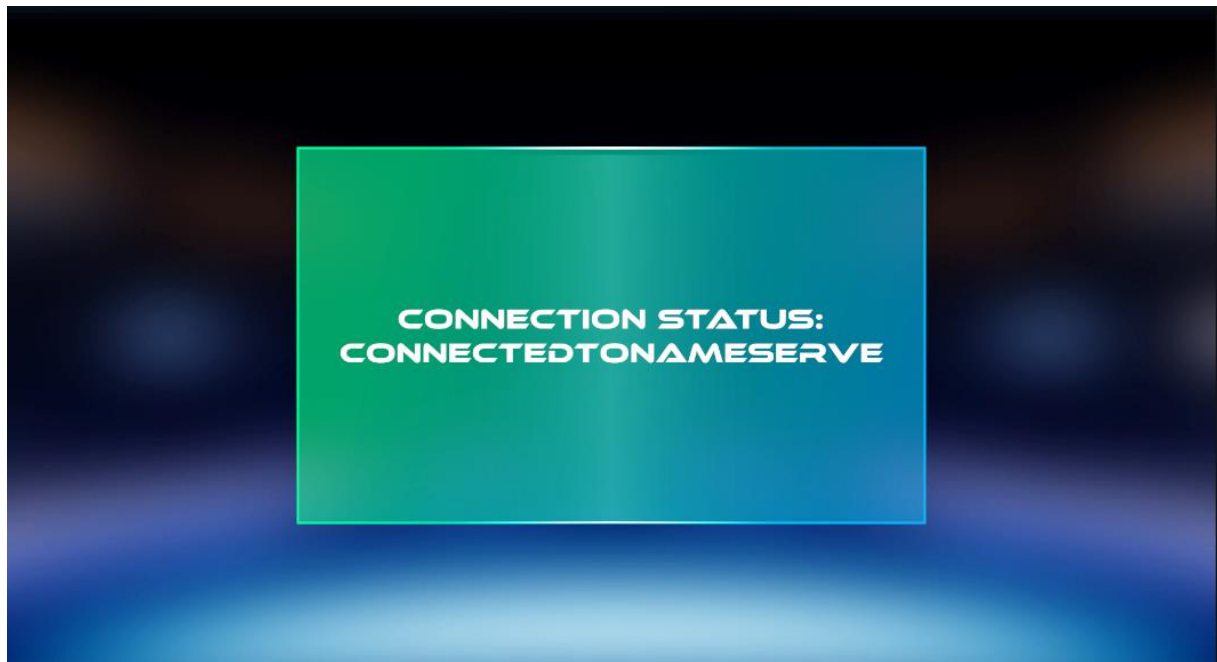
6.2.2.5 Singleplayer AI Difficulty Selection



Users choose according to the difficulty level.

6.2.2.6 Connection Status Page (Multiplayer)





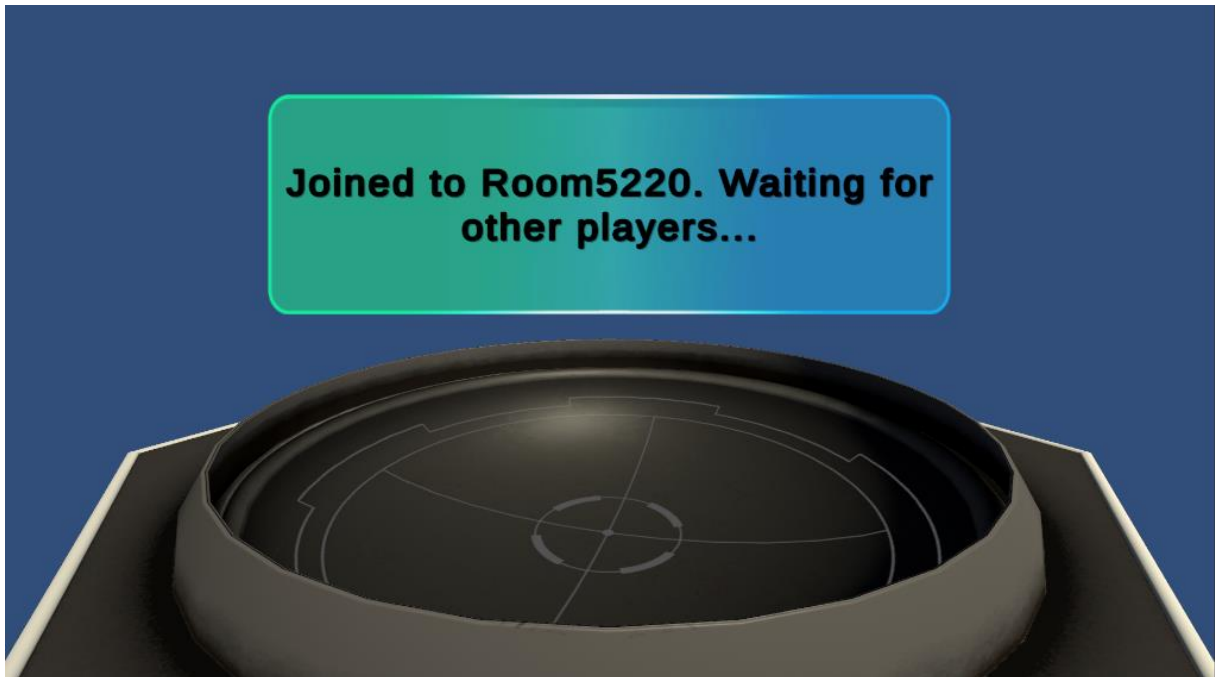
After the user selects the multiplayer mode and enters the user name, they are directed to the connection screen. If the user successfully connects to Photon servers, the login page below opens.

6.2.2.7 *Beyblade Selection*



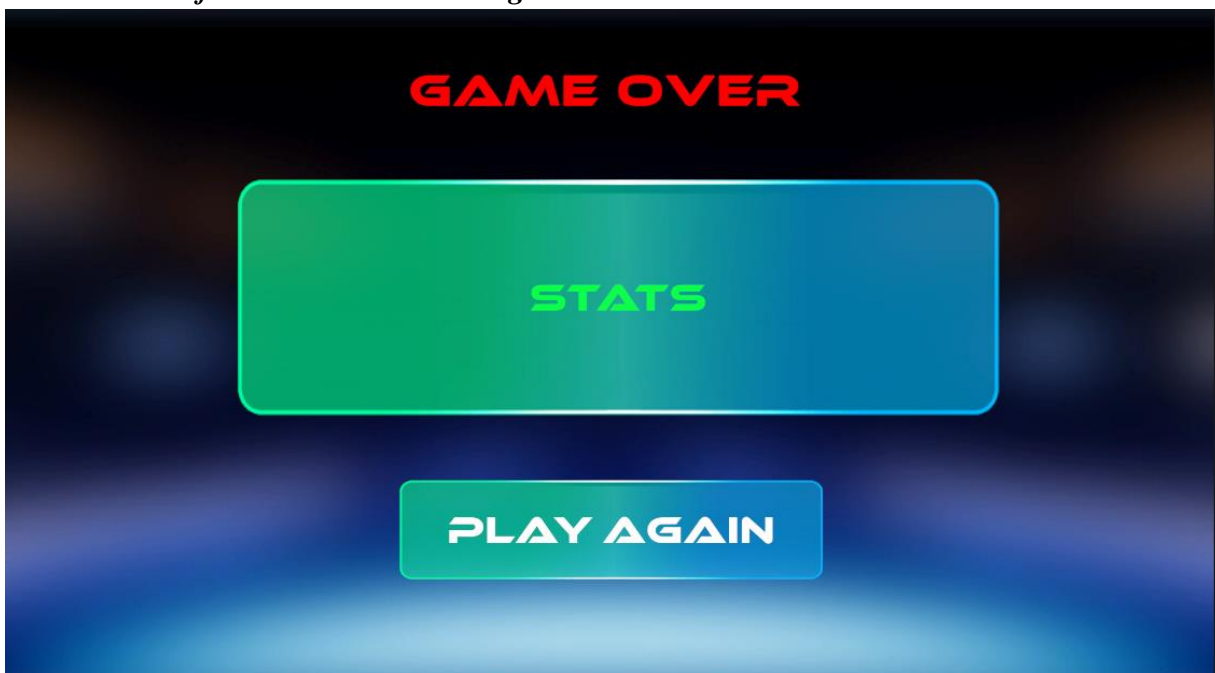
Users choose from the Beyblades on the screen.

6.2.2.8 *Joining Rooms (Multiplayer)*



After the user connects to the server, s/he is automatically directed to one of the rooms.

6.2.2.9 *End of the Game / Results Page*



Screen showing in-game score data and a replay button when the player or players' game is over