



Project Number 17

Mobile Edutainment Game for White Rabbit (Alice Connex)

Presented by

1. Ms. Pannatat Supharatnodom 55070503428
2. Mr. Pittawat Jungsanguansit 55070503435

Advisor

Khajonpong Akkarajitsakul, Ph.D.

Co-advisor

Assoc. Prof. Tiranee Achalakul, Ph.D.

I have read this report and approve its content.

.....

(.....)

Advisor



Mobile Edutainment Game for White Rabbit (Alice Connex)

Ms. Pannatat Supharatnodom
Mr. Pittawat Jungsanguansit

A Project Submitted in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Engineering
Department of Computer Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi
Academic Year 2015

Mobile Edutainment Game for White Rabbit (Alice Connex)

Ms. Pannatat Supharatnodom
Mr. Pittawat Jungsanguansit

A Project Submitted in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Engineering
Department of Computer Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi
Academic Year 2015

Project Advisor

..... Advisor
(Khajonpong Akkarajitsakul, Ph.D.)

Project Co-advisor

..... Co-advisor
(Assoc. Prof. Tiranee Achalakul, Ph.D.)

Project Committee

..... Committee
(Asst. Prof. Nuttanart Facundes, Ph.D.)

..... Committee
(Kurt. T. Rudahl)

Project Title	Mobile Edutainment Game for White Rabbit (Alice Connex)
Project Credit	3 credits
Project Participant	Ms. Pannatat Supharatnodom Mr. Pittawat Jungsanguansit
Advisor	Khajonpong Akkarajitsakul, Ph.D.
Co-advisor	Assoc. Prof. Tiranee Achalakul, Ph.D.
Degree of Study	Bachelor's Degree
Department	Computer Engineering
Academic Year	2015

Abstract

Mobile devices such as smartphone and tablets have become as a part of our daily life. Then, in the project, we are going to develop a mobile game on Android. The key purpose of our game is to entertain players, to indirectly educate them via our game play, and to help promote ALICE Connex, a mobile volunteer computing platform for ALICE at CERN. Particularly, our mobile game has been designed to be a 2D multiplayer real-time interactive game. Players can fight with others by using items which are designed based on particle physics. To achieve in the game, each player may need to learn how different a type of particles is in order to powerfully attack or defense opponents. The story of the game begins with the discovery of "Magic Particles". These particles are used to own by Alice, a greatest angle. Once the human is attacked by a demon, Alice gives these particles to human and teaches them how to protect themselves from the demon. However, the demon is always trying to steal those particles. Thus, human decide to prepare themselves for a fight with the demon by holding the "Particle Shot Tournament". In the tournament, there are two teams per round. Each team can have at most three players. Each player has to select one item, each item has a different attack power and defense power. The tournament will continue until one of the teams loses all the team members. Moreover, to promote ALICE and ALICE Connex, all CPU cycles donated to ALICE experiments via ALICE Connex can be redeemed to in-game credits. Players can use their credits to acquire game items or real world items. This can help increase the number of volunteers as our expectation.

บทคัดย่อ

อุปกรณ์สื่อสาร เช่น โทรศัพท์มือถือ (Smartphone) และแท็บเล็ต (Tablet) ได้เข้ามามีบทบาทในชีวิตประจำวันมากขึ้น เราจึงการพัฒนาแอปพลิเคชันเกมบนมือถือในระบบปฏิบัติการแอนดรอยด์ (Android) โดยเป้าหมายหลักของแอปพลิเคชันคือการสร้างความเพลิดเพลินให้แก่ผู้เล่น สอดแทรกความรู้เกี่ยวกับคุณสมบัติของอนุภาคฟิสิกส์ให้แก่ผู้เล่นโดยอ้อม และประชาสัมพันธ์โครงการ ALICE Connex ซึ่งเป็น Mobile Volunteer Computing Platform ในโครงการ ALICE ของ องค์การวิจัยนิวเคลียร์ยุโรป (CERN)

เกมมือถือถูกออกแบบให้เป็นเกม 2 มิติ (2D) ที่สามารถรองรับผู้เล่นหลายคนในเวลาเดียวกัน ผู้เล่นสามารถต่อสู้กับผู้เล่นอีกฝ่ายโดยใช้อุปกรณ์ (Item) ซึ่งถูกออกแบบให้มีความสอดคล้องกับคุณสมบัติของอนุภาคฟิสิกส์ ในการเอาชนะผู้เล่นฝ่ายตรงข้าม ผู้เล่นควรมีความรู้เกี่ยวกับอนุภาคฟิสิกส์เพื่อใช้ในการวางแผนการโจมตีหรือป้องกันคู่ต่อสู้

นอกจากนี้ จำนวน CPU cycle ทั้งหมดที่ผู้เล่นบริจาคให้แก่โครงการ ALICE ผ่านแอปพลิเคชัน ALICE Connex ผู้เล่นสามารถนำมาแลกเปลี่ยนคะแนนสะสมเพื่อใช้แลกซื้อสินค้าพิเศษภายในเกมหรือแลกรับของที่ระลึกได้

เนื้อเรื่องของเกมเริ่มต้นจากมนุษย์ได้ค้นพบอนุภาคเวทมนตร์อันทรงพลัง ซึ่งเป็นของนางฟ้านามว่าอลิซ เนื่องจากโลกมนุษย์ถูกรุกรานโดยอสูรกาย อลิซจึงมอบอนุภาคเวทมนตร์และสอนวิธีใช้ให้กับเหล่ามนุษย์เพื่อป้องกันอันตรายจากอสูรกายและปกป้องโลก ทว่าอสูรกายที่โหดร้ายยังรอคอยโอกาสที่จะขโมยอนุภาคเวทมนตร์และทำลายเหล่ามนุษย์อยู่เสมอ เหล่ามนุษย์จึงได้จัดการประลองอนุภาคเวทมนตร์ขึ้นเพื่อเป็นการเตรียมพร้อมต่อสู้กับอสูรกาย โดยประลองเป็นทีม ครั้งละ 2 ทีม โดยแต่ละทีม ผู้เล่นจะทำการเลือกอุปกรณ์ที่ต้องการใช้ได้เพียงหนึ่งอย่างเท่านั้น โดยอุปกรณ์จะมีคุณสมบัติในการป้องกันและโจมตีที่แตกต่างกัน การต่อสู้จะเป็นการเล่นแบบรอบ (Round base) การต่อสู้จะดำเนินไปเรื่อยๆจนกว่าทีมใดทีมหนึ่งจะเสียชีวิตผู้เล่นทั้งหมด หรือผู้เล่นสูญเสียพลังชีวิตจนหมดทุกคน

Acknowledgements

This project could have not been completed without the supports from our advisors Dr. Khajonpong Akkarajitsakul and Assoc. Prof. Tiranee Achalakul, Ph. D. They spent lots of their time to guide us and help us in many situations to make our project run smoothly with the right direction. Furthermore, they also provided necessary equipment that we need for completing this project. We truly appreciate their helps.

We would like to express our special gratitude to Dr. Kachanon Nirunpong for taking his time to give us many expert advices about particle physics and related knowledge. He also gave important information that we need to integrated with our project

Our project would lack of beauty and creativity without Mr. Gantaphon Chalumporn. We really appreciate his generosity. He spent his time to help us design the great characters, weapons, characters' costumes, and user interface of our game.

Finally, we have to appreciate our committees for their valuable comments and suggestions which are significant for the learning opportunity and encouragement, which lead us to improve our project in the right direction.

Table of Contents

Contents	Page
Chapter 1 Introduction	1
1.1 Problem Statement and Approach	1
1.2 Objectives.....	2
1.3 Scope	3
1.4 Tasks and Schedule	3
1.5 Gantt chart	5
Chapter 2 Background, Theory and Related Research	8
2.1 Collaboration with CERN	8
2.1.1 Conseil Européen pour la Recherche Nucléaire (CERN).....	8
2.1.2 ALICE (A Large Ion Collider Experiment)	9
2.2 The concept of serious game	9
2.3 Natural Funativity Theory	11
2.4 Particle physics for edutainment game design.....	11
2.4.1 Particle Physics Table	13
2.5 Game Technology	13
2.5.1 Unity3D	13
2.5.2 Photon Turn-based	14
2.5.3 Playfab.....	15
2.5.4 Facebook SDK for Unity	16
2.5.5 Photon Server Architecture	17
Chapter 3 Design and Methodology	18
3.1 Hardware & Software requirements	18
3.2 Game Design	18
3.3 Feature list	19
3.4 Architecture diagram	20
3.6 Use case diagram	22
3.6.1 Use case narrative	22
3.7 Sequence diagrams	27
3.7.1 Scenario: Login	27

3.7.2 Scenario: Matchmaking.....	27
3.7.3 Scenario: Playing	28
3.8 Game story prologue	28
3.9 User interface screen layout design.....	29
3.10 Table of items properties.....	32
3.11 Refutation table	33
3.12 Database schema	34
3.13 Entity-Relationship diagram.....	35
Chapter 4.....	36
4.1 Game screenshots.....	36
4.1.1 Game title screenshots.....	36
4.1.2 Login/Register & Lobby screenshots	37
4.1.3 Store & inventory screenshots.....	41
4.1.4 Donated CPU's cycle redemption screenshots.....	44
4.1.5 Leaderboard screenshots.....	44
4.1.6 Waiting room screenshots.....	45
4.1.7 Battle field screenshots	47
4.2 Performance data	53
Chapter 5	55
Conclusion.....	55
5.1.1 Completion table	55
Reference	58

List of Figures

Figures

Figure 1.1 Worldwide Quarterly Mobile Phone Trackers (IDC).....	2
Figure 2.1 CERN logo	8
Figure 2.2 CERN particle accelerators and detectors	9
Figure 2.3 Serious Games Demand Circle.....	10
Figure 2.4 Curricular integration and alignment of Sustain City.....	10
Figure 2.5 Quark Particle	12
Figure 2.6 Gluon Particle	12
Figure 2.7 Particle Physics Table.....	13
Figure 2.8 Unity logo	13
Figure 2.9 Photon logo.....	14
Figure 2.10 PlayFab logo.....	15
Figure 2.11 Photon High Level Architecture.....	17
Figure 3.1 Architecture diagram	20
Figure 3.2 Stack diagram	21
Figure 3.3 Use Case diagram	22
Figure 3.4 Scenario: Login	27
Figure 3.5 Scenario: Matchmaking.....	27
Figure 3.6 Scenario: Playing	28
Figure 3.7 Shop.....	29
Figure 3.8 Rewarding.....	29
Figure 3.9 Character design and Weapon	30
Figure 3.10 Waiting Room	30
Figure 3.11 Character orientation	31
Figure 3.12 Start battle.....	31
Figure 3.13 ER-diagram.....	34
Figure 4.1 Game story1.....	35
Figure 4.2 Game story2.....	36
Figure 4.3 Login/Register scene	36

Figure 4.4 Register dialog1	37
Figure 4.5 Register dialog2.....	38
Figure 4.6 Register complete	38
Figure 4.7 Lobby scene.....	39
Figure 4.8 Lobby scene2.....	39
Figure 4.9 Inventory	40
Figure 4.10 Inventory 2.....	40
Figure 4.11 Store	41
Figure 4.12 Confirm purchase	41
Figure 4.13 Purchase succeed	42
Figure 4.14 Purchase failed.....	42
Figure 4.15 Redemption	43
Figure 4.16 Leader board.....	43
Figure 4.17 Waiting room.....	44
Figure 4.18 Waiting room2.....	44
Figure 4.19 Waiting room3.....	45
Figure 4.20 Weapon detail.....	45
Figure 4.21 Battle field scene(Large)	46
Figure 4.22 Battle field scene (Small)	46
Figure 4.23 “Aim-and-fire” control	47
Figure 4.24 Fired action	47
Figure 4.25 Fired action scene (Small)	48
Figure 4.26 Win	48
Figure 4.27 Lose	49
Figure 4.28 Push force1	49
Figure 4.29 Push force2	50
Figure 4.30 Pull force1	50
Figure 4.31 Pull force2	51

Chapter 1

Introduction

1.1 Problem Statement and Approach

The rapid technology development increases computational capability for human, which leads to more complicated experiments and researches. For example, ALICE (A large Ion Collider Experiment) [1] which is a heavy ion-detector from CERN or The European Organization for Nuclear Research is used to detect and study the properties of Quark-Gluon Plasma through high energy particles collision. The output of the experiment is million gigabytes of data in each year [2]. However, to process huge amount of data also requires high computational power, which exceeds the available internal resources. Therefore, volunteer computing is a way for researchers to gain computational resources and also a way for people to participate and give their support to the experiment that they are interested in. However, the ALICE volunteer computing platform is currently not widespread, most people do not know about this platform and some of them might not be interested in the particle physics. This causes delay in their research which can be helped by mobile volunteer computing.

Mobile volunteer computing is an interesting computing platform because nowadays, millions of mobile devices are used everywhere in the world. According to data from the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker [3], the worldwide smartphone market has growth 13.0% year over year in 2015 Q2, with 341.5 million shipments. Moreover in 2017, global game market will has Compound Annual Growth Rate (CAGR) of 8.1% while smartphones and tablet gaming will take 27.8% of the global games market in 2016, up from 17.4% in 2013 [4]. This means that the trend of user behavior on mobile devices are changing and leads to alternative accessible channel to people through gaming on mobile devices and also an opportunity to approach and introduce a mobile volunteer computing platform along with the game. Therefore, we have decided to promote the volunteer computing platform name ALICE Connex via 2D mobile edutainment strategy game. This game is a multiplayer real-time interaction on single turn-based action, while it integrates seamless edutainment with basic particle physics content through the mechanism and items design. In the game, the player who can understand characteristics of each particle will get an advantageous in this game. Thus, how players will automatically gain the knowledge of particle physics is what is interested in our game content.

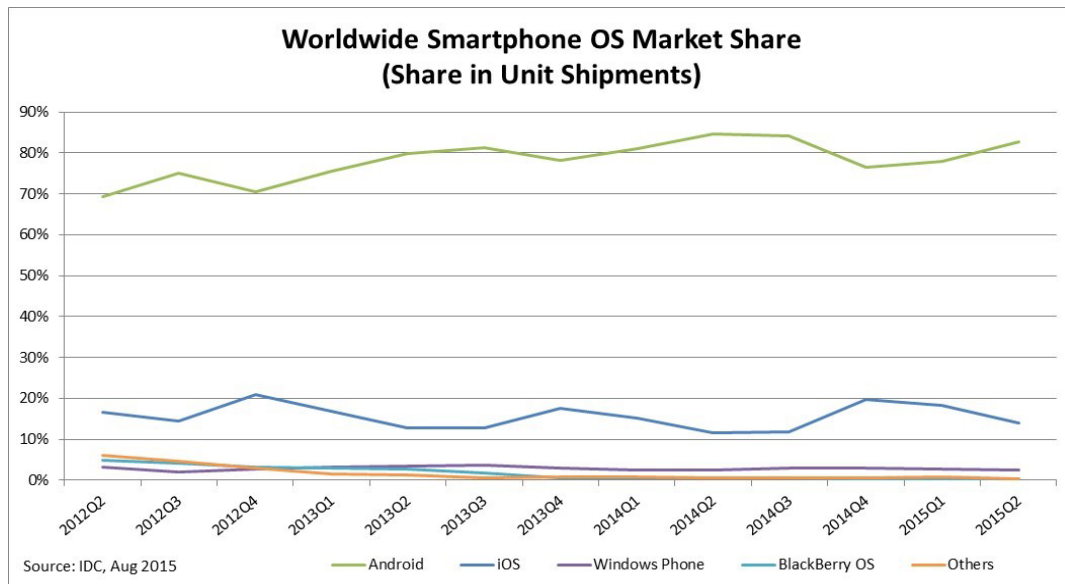


Figure 1.1 Worldwide Quarterly Mobile Phone Trackers (IDC)
 [Source: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>]

To develop our mobile game, we plan to do it on Android first because it has the highest market share as shown in the figure and flexibility than others OS. We expect that after our game is launched, the number of volunteers will significantly increase, we estimate the number of ALICE Connex volunteers will increase at least 20 percent each year, approximately one-half of our game players. Furthermore, player will gain particle physics knowledge and entertain with their friends. From the figure above, Android dominated the market with an 82.8% share in 2015 Q2 and trend to be the OS that has highest share in the future.

Our aims of the project can be divided in the following categories.

Fulfilling stakeholder needs: Stakeholders is the Conseil Européen pour la Recherche Nucléaire (CERN), which would like to promote a new mobile volunteer computing platform named ALICE Connex and to induce volunteers to donate computing resources via our mobile edutainment game

Providing social value: This project can be considered as a potentially powerful learning application which can improve educational effectiveness. Together, both entertainment and education can be merged as edutainment which be able to deliver beneficial knowledge of particle physics with enjoyment to people.

1.2 Objectives

- To promote ALICE Connex on mobile volunteer computing platform
- To raise the number of volunteers at least 20 percent every year
- To interfere particle physics knowledge to people
- To entertain people via our game

1.3 Scope

- An Android mobile game is implemented with particle physics and ALICE information integrated.
- This game supports multiplayer with 2 teams and 3 players / team
- This game can be connected to the ALICE Connex mobile volunteer computing server in order to exchange rewards from volunteered CPU cycles.
- **Software scope**
 - Be able to handle multiplayer connections
 - Be able to handle 20 connections in the first phrase
 - Be able to authenticate users via Facebook
 - Have at least 5 items available
 - Have at least 5 characters available
 - Be able to be connected to ALICE Connex (the mobile volunteer computing of ALICE)
- **Software limitations**
 - Run on Android 4.4 or higher only
 - Require internet connection

1.4 Tasks and Schedule

Semester 1

No.	Operation	Predecessors
1	Related research review	
2	Project Proposal Submission	
3	Necessary program, library installation and software environment setting	
4	Software Design I -Interface Design	
5	Software Design II -Back-end System Design	1
6	Implementation -Character	4
7	Implementation -Weapon and skill	4, 5
8	Implementation -Map	4, 5

Semester 2

No.	Operation	Predecessors
9	Implementation -Back-end system and logical process such as ranking, damaging, items dropping etc...)	4, 5
10	Game server and database server implementation	5
11	First draft report submission	
12	Presentations I	
13	Implementation -Transaction module	5
14	Implementation - User account module	5
15	Program testing & live run	6-10, 13-14
16	Game distribution on Play store	15
17	Final report submission	
18	Presentations II	

Chapter 2

Background, Theory and Related Research

2.1 Collaboration with CERN

This project is collaboration between KMUTT and CERN. The mobile game is developed in order to promote ALOCE (one of main accelerators at CERN), to educate people about particle physics, and to induce people who have mobile devices to donate their ideal CPU cycles to the ALICE experiment via ALICE Connex. Note that ALICE Connex (formerly White Rabbit) is another collaboration project between KMUTT and CERN. The key propose of the ALICE Connex project is to build a mobile volunteer computing platform for the ALICE experiment. Our proposed game is a part to be linked with ALICE Connex. Players of the game can redeem game items or real-world items by using credits obtained when they donate their CPU cycles to ALICE Connex.

More details about CERN and ALICE are as follows:

2.1.1 Conseil Européen pour la Recherche Nucléaire (CERN)



Figure 2.1 CERN logo represents the accelerators overall image

[Source: <http://design-guidelines.web.cern.ch/sites/designguidelines.web.cern.ch/files/u6/CERN-logo.jpg>]

CERN, the European Organization for Nuclear Research, was established in 1954. However, the concept of CERN was proposed by French physicist Louis de Broglie in 1949. His aim is to create a European laboratory at the European Cultural Conference.

In 1957, the first accelerator called ‘Synchrocyclotron’ was built. At first, the scientist focused on particle and nuclear physics. However, in 1964, they focused on only nuclear physics and left the particle to the newer accelerator called Proton Synchrotron. Proton Synchrotron, a world’s highest energy particle accelerator, has been used to accelerate protons since 1959 until now. The anti-nuclei were found in 1965 by observing the antineutron, a nucleus of antimatter made out of an antiproton plus an antineutron. The first Proton collision was detected in 1971 in the project named Intersecting Storage Rings (ISR).

In 1975, the Super Proton Synchrotron (SPS), the first of CERN's giant underground rings, was built. It crosses between the Franco and Swiss border. SPS has been used until now to handle many particles and to study about the inner of particles. [5]



Figure 2.2 CERN particle accelerators and detectors.

[Source: <http://scienceblogs.com/startswithabang/files/2012/07/LHC.jpeg>]

2.1.2 ALICE (A Large Ion Collider Experiment)

ALICE is one of the seven detectors at CERN. This detector is optimized to study heavy-ion collisions. The purpose is to produce quark–gluon plasma, a state that quarks particle and gluons particle are freed. This experiment focuses on strongly interacting matter at extreme energy densities. Quark–gluon plasma and its properties are the important key for their experiments and research [6]

2.2 The concept of serious game

Game is an activity which provides entertainment to people who play the game. However, new idea and technology, for example, a fill in the blank game or mathematical game in these days makes the game not only entertain players but also educate them. A game which gives players knowledge can be classified as “Serious Game”. Serious game can be divided into 6 subtypes which are advergaming, educational games, health games, political, religious, and social games. The serious game is developed with education and researches by following the demand cycle in the Figure 2.3.1

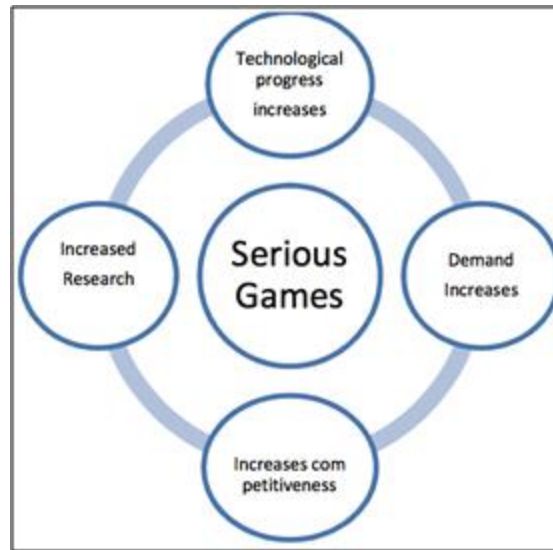


Figure 2.3 Serious Games Demand Circle

However, the Serious Games have to balance between fun and learning. The main goal is to combine the knowledge student learnt in the classroom or textbook with the entertainment. If the game is too educational, it will be too boring. If the game is too fun, it would not meet the goal of edutainment game. With the balance of these 2 factors, the players can enjoy the game and learn in the same time. The game does not limit the number of topics used in the game. For example, in Figure 2.3.2, the game is integration between many subjects - Digital Electronics, Civil Engineering and Architecture, Principle of Engineering, and Biotechnical Engineering.

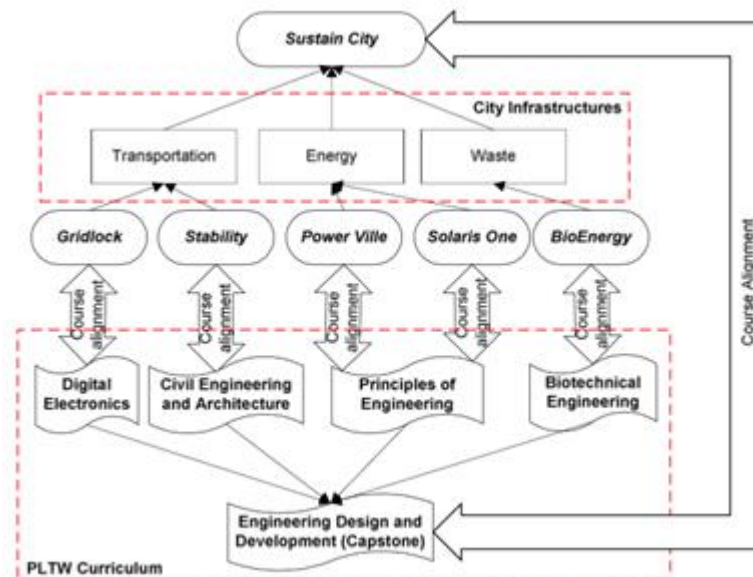


Figure 2.4 Curricular integration and alignment of Sustain City

2.3 Natural Funativity Theory

Natural Funativity is a theory of how entertainment and games relate to learning about survival skill by Noah Falstein. This theory is divided into three categories – physical fun, social fun and mental fun.

Physical activity is a part of human evolutionary, for example, sports. Sports like running or body building have an advantage in human survival. Many people enjoy playing sports with their friend or sometimes involving in a competition. While some people prefer sports, many people might prefer exploring places and enjoy engaging with new culture and activities. This shows that physical activities give pleasure to those who engage with the activity.

People also prefer to be involved in a group and watch other people do. A multiplayer game is best describes the kind of game which is social engagement. It creates virtual communities, group or sometimes lead to marriage. This shows that social engagement is an important part of a game. Many games allow players to challenge other players on the leaderboard and share their activities on a social media.

With only physical and social fun, the game might become boring. The game should be challenging and teach us something useful for survival. For example, recognize the shape and match to the proper space accurately in Tetris. It also should be easy to learn but keep adding new patterns to make it new and fun to play.

2.4 Particle physics for edutainment game design

In this section, some types of particles will be presented. The different characteristics or properties of different types of particles will be used in our game design. Our items in the game will be based on those particles. Players in the game will be indirectly educated about particle physics through the items and the results when players use those items to attack or to defense their opponents. Some combinations among particles may increase attack damage or decrease attack damage.

The details of some particles are shown as the following.

1) Quark-Gluon plasma (QGP)

A quark–gluon plasma or quark soup is a state of matter in quantum chromo dynamics (QCD) hypothesized to exist at an extremely high temperature, density, or both temperature and density. Scientists have been thinking **that** this stage consists of free quark particles and gluon particles that is the basic blocks of substance. The universe was in a quark–gluon plasma state, so both particles might be the important key to study the universe formation [7]

2) Quark

Quark is a primary particle and a fundamental composition of substance. Combination of Quark can be formed a Hadrons particles that consist of protons particle and neutrons particle (the components of atomic nuclei). Furthermore, Quark is the most stable particle compare with others. [8]

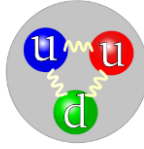


Figure 2.5 Quark Particle

[Source: <https://en.wikipedia.org/wiki/Quark>]

3) Antiparticle

Antiparticle or antimatter is used to call a particle that has the same mass and opposite charge. This term can be used to call any particle that has the opposite particle. For example, the antiparticle of the positron (a particle with positively charge) is the electron (a particle with negative charge). [9]

4) Gluon

Gluons is a primary particle that acts as an the exchange particle for the strong force between quarks, similar to the exchange of photons in the electric force between two charged particles. [10]

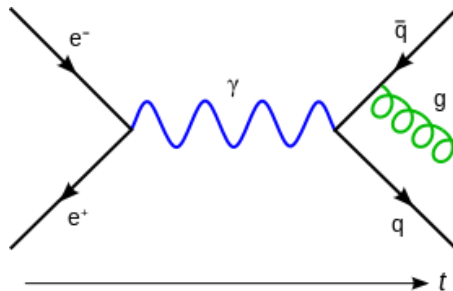


Figure 2.6 Gluon Particle

[Source: <https://en.wikipedia.org/wiki/Gluon>]

5) Hadron

Hadron is a composite particle made of quarks held together by the strong force. Hadrons are categorized into two families: baryons, made of three quarks, and mesons, made of one quark and one antiquark. [11]

6) Neutrino

Neutrinos have tiny mass compared to other subatomic particles. Neutrino is an electrically neutral elementary particle. Since Neutrinos do not carry any electric charge so they are not affected by the electric force that acts on charged particles. [12]

7) Meson

Mesons are hadronic subatomic particles consisting of one quark and one antiquark, bound together by the strong force. These particles are unstable, with the lifetime for a few hundredths of a microsecond. [13]

2.4.1 Particle Physics Table

Particle	symbol	charge	rest mass / MeV
electron	e^- β^- e	-	0.510999
positron	e^+ β^+ \bar{e}	+	0.510999
proton	p	+	938.257
antiproton	\bar{p}	-	938.257
neutron	n	0	939.551
antineutron	\bar{n}	0	939.551
neutrino	ν	0	0
antineutrino	$\bar{\nu}$	0	0

Figure 2.7 Particle Physics Table

[Source: <http://physicsnet.co.uk/a-level-physics-as-a2/particles-radiation/particles-antiparticles-photons/>]

2.5 Game Technology

2.5.1 Unity3D



Figure 2.8 Unity logo

[Source: <http://www.mmag.bplaced.net/images/mmag/unity3d-logo.jpg>]

Unity is a cross-platform game engine which is one of the most popular game engines in the market. It gains almost 45 percent of the game engine market share. Game developers use unity to create both 2D and 3D games. Some popular games in the market such as Temple Run and Angry Birds 2 have been also developed by using Unity. [14]

The program is divided into 2 parts which are designing and programming. For the designing part, developers can import their images (for 2D game) or models (for 3D game) into Unity directly. Those images and models are called “GameObject”. Then, we can attach the

components and scripts to each GameObject to do some action such as interactive with user or interactive with another GameObject. Scripts used in Unity are written in Object-oriented programming languages which are C# and Javascript.

Unity provides 2 versions of the application. [15]

1) Personal Edition (Free version)

Unity Personal Edition includes all Unity features, royalty-free, and support all platforms such as Android and iOS. However, people who use Unity Personal Edition should not use it to gain some revenue or to have a budget for educational reason more than \$100,000 per year.

2) Professional Edition

Unity Professional Edition includes all features in Personal Edition with much more features such as Unity Cloud Build Pro for 12 Months, Unity Analytics Pro, Prioritized bug handling, Game Performance Reporting, and Beta access to an update in Unity 5. Developers have to pay about 75 USD per month for the license fee.

2.5.2 Photon Turn-based



Figure 2.9 Photon logo

[Source: http://www.gamasutra.com/db_area/images/news2001/36304/Photon.jpg]

Photon is the world's first independent cross-platform network engine. More than 100,000 developers are using Photon in their games and more than 60 million users per month. It can be also integrated with Unity called Photon Unity Networking or PUN. PUN is used to create a real time multiplayer game. However, we use Photon Turn-based instead of PUN as it is more compatible with our project because PUN is not compatible with the turn-based game which needs to handle an inactive players.

Photon turn-based is one of Photon networking tools which is used to build both synchronous and asynchronous game. It is divided into 3 parts which are client, server, and cloud. Client means players which interact with each other whether they are Android, iOS, or other platform users. Server is a place where the data is updated and sent to all players. Photon

stores all data in cloud for a faster and worldwide real time access. In Asia, the cloud storage is in Singapore which is close to Thailand.

Photon turn-based is free for any game which has less than 20 connections. If the game has a larger number of connections, there are fees starting at \$65 for up to 100 connections, \$65 per month for 500 connections, and \$125 per month for each 1,000 connections up to 10,000 connections. [16]

Photon turn-based provides an SDK for developers to use in their projects. The SDK supports not only Unity but also support other game engines such as Unreal Engine and Playfab. Photon uses C# as the programming language.

The Photon Server core supports 4 protocols which are reliable UDP (based on eNET) and specially tuned for Client-2-Server architectures, binary TCP, web sockets, and HTTP. The transfer protocol is very lean and slim. Photon wraps up the networking layer of each client platform. Our game client can communicate cross-platform and across protocols.

Our game sits on top of a server development framework that solves common tasks such as RPC calls are simply and flexible mapped to operation instances and threading problems are solved using message passing through fibers.

Photon is designed to saturate bandwidth before the CPU becomes the bottleneck. [17]

2.5.3 Playfab



Figure 2.10 Playfab logo

[Source: <https://12893-presscdn-0-34-pagely.netdna-ssl.com/wp-content/uploads/2015/10/PlayFabLogo.png>]

Playfab is a backend service for a real-time multiplayer and turn-based game. It is recommended to work with Photon. By using Playfab with Photon, developers can use Prefab's Cloud Script to make an event such as creating or joining room. Also, Photon is free for developers who use premium and enterprise tier of Playfab. [18]

Playfab is a cloud-based service as same as Photon. Playfab provides services for game developers which are game database, player authentication, virtual economy, and matchmaking.

- Game database: Playfab will store data in a game such as player accounts, player status and actions made by that player, and item information.
- Player authentication: Developers can use Playfab service to authenticate player whether the player is registered by signing up directly or by using their Facebook account.
- Virtual Economy: If a game has a function which allows player to purchase some item from the game, virtual economy service is necessary. Playfab provides virtual economy features in its manager for an easier way to monitor and control the currency in the game.
- Matchmaking: Playfab provides a custom game server hosting and matchmaking services for an asynchronous and synchronous multiplayer game.

PlayFab has 4 choices for each level of development. [19]

1. FREE: Secure authentication and in-app purchases at no cost.
2. STANDARD: Add virtual currencies, leaderboards, push notifications, and etc.
3. PREMIUM: Complete multiplayer functionality plus the flexibility of Photon and Cloud Script.
4. ENTERPRISE: Custom features and pricing to fit your game and budget.

From our research, we have found that Photon recommends developers to use Photon with Prefab. Photon provides an SDK for developers to access to PlayFab. All PlayFab features are authentication, account management, friends, leaderboard, virtual currency, and catalog which will be called through API.

2.5.4 Facebook SDK for Unity

Facebook SDK for Unity [22] is a tool that Facebook provided for a developer for linked their application with Facebook

The game application can request name, email, birthdate, profile picture and others data that user allowed for registration and display in the game via their Facebook account. For more information see <https://developers.facebook.com/>

2.5.5 Photon Server Architecture

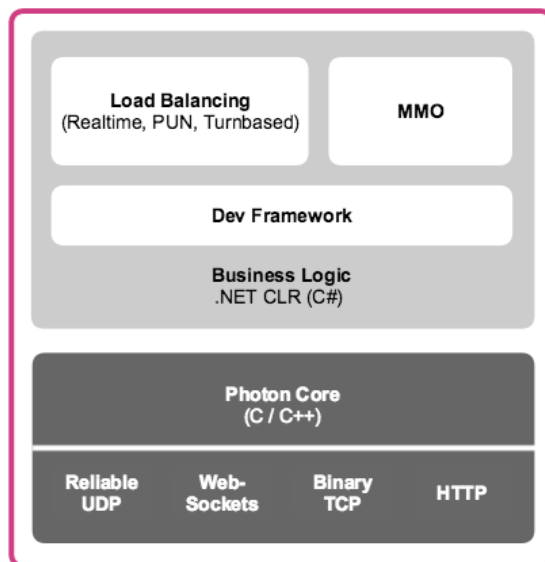


Figure 2.11 Photon High Level Architecture

[Source: <https://doc.photonengine.com/en/onpremise/current/getting-started/photon-server-intro>]

Photon Server Architecture

The core of the server (Backend System) was written in native C++ programming language for performance reasons [17]

- Uses Input/output completion port (IOCP), IOCP is an API (set of routines, protocols, and tools for building) for performing multiple simultaneous asynchronous input/output operations because IOCP has high performance socket handling
- Implements the heavy "lifting" of the protocols
- Supports UDP, TCP, HTTP and Websockets protocols

The business logic (a part of the program that determine how data can be created, displayed, stored, and changed) executes in a .NET CLR [20] hosted by the core.

- Contains the applications running on Photon
- Written in C# programming language or any other .NET programming language

Chapter 3

Design and Methodology

3.1 Hardware & Software requirements

- Software
 - Unity3D
 - MonoDevelop
 - Photon turn-based
 - Playfab
 - Photoshop
 - Android version 4.4.2 (API level 19) or higher
- Hardware

Minimum hardware requirement

Smartphone or Tablet with the following specification

 - Processor: Dual-core 1.2 GHz Cortex-A9
 - GPU: Broadcom VideoCore IV
 - Ram: 2GB
 - Storage: at least 2GB

3.2 Game Design

- Registration / Login system (User Account)
 - Create new account
 - Link with Facebook account
 - Recover forgotten password
- Game Mode
 - Quick start: A player joins an existing room immediately.
 - Room creation: A player can create a room to build a team.
 - Real time turn-based game: A player takes turn to attack.
 - One player can pick one character at a time and there will be 3 players in a team.
 - A player can invite friends to join the team
 - Training: A player can train with a boss which has a higher ability
- In-game Currency
 - Main currency: can be gained from playing and exchanging from volunteer credits.
 - Volunteer credits: can be gained only from donating the mobile phone's CPU cycles.
- Item (Weapon & Armor)
 - Normal Item: A player can purchase an item by using main currency.

- Special Item: A player can purchase an item by using only volunteer credit.
- Characters: Each one derived from physics particle. Each character has 3 attack methods.
- ALICE Connex connection
 - Acquire players' CPU cycle information donated via ALICE Connex in order to exchange special currency in this game
- Support Android OS, the game is designed to be compatible with Android version 4.3 [Jelly Bean] with API level 18 or above.
- Interfere basis of particle physics knowledge in the game content, couple of particle that reinforcement and cancellation with each other

3.3 Feature list

- Player
 - Able to login to the game
 - Able to login using Facebook account (in case that the user registered with Facebook account).
- Character
 - Able to change character
- Quick join
 - Able to join the room within 10 seconds
- Room creation
 - Able to create room
- Practice mode
 - Able to fight with the greater monster
- Shop
 - Able to buy items from the shop

3.4 Architecture diagram

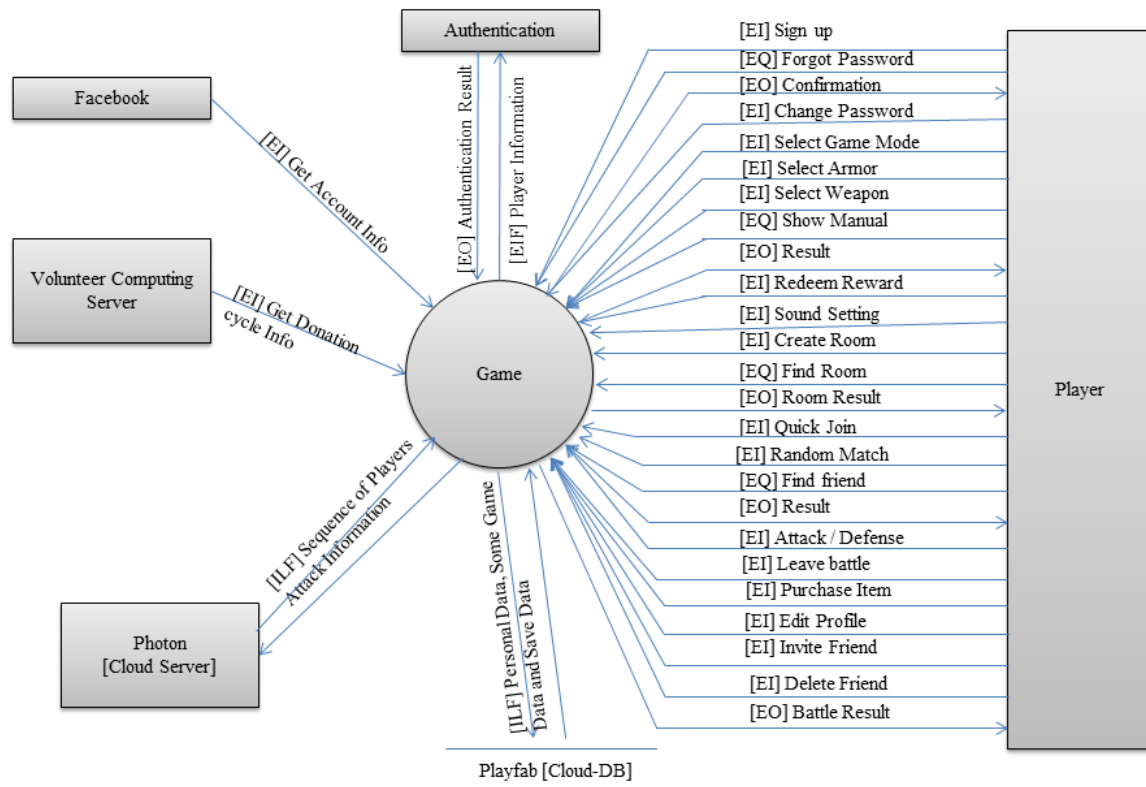


Figure 3.1 Architecture diagram

We separate the game into six major parts with one actor. The main part is a game engine, communicating with authentication system, Facebook, ALICE Connex server, Cloud Database, and Cloud Server

3.5 Stack Diagram

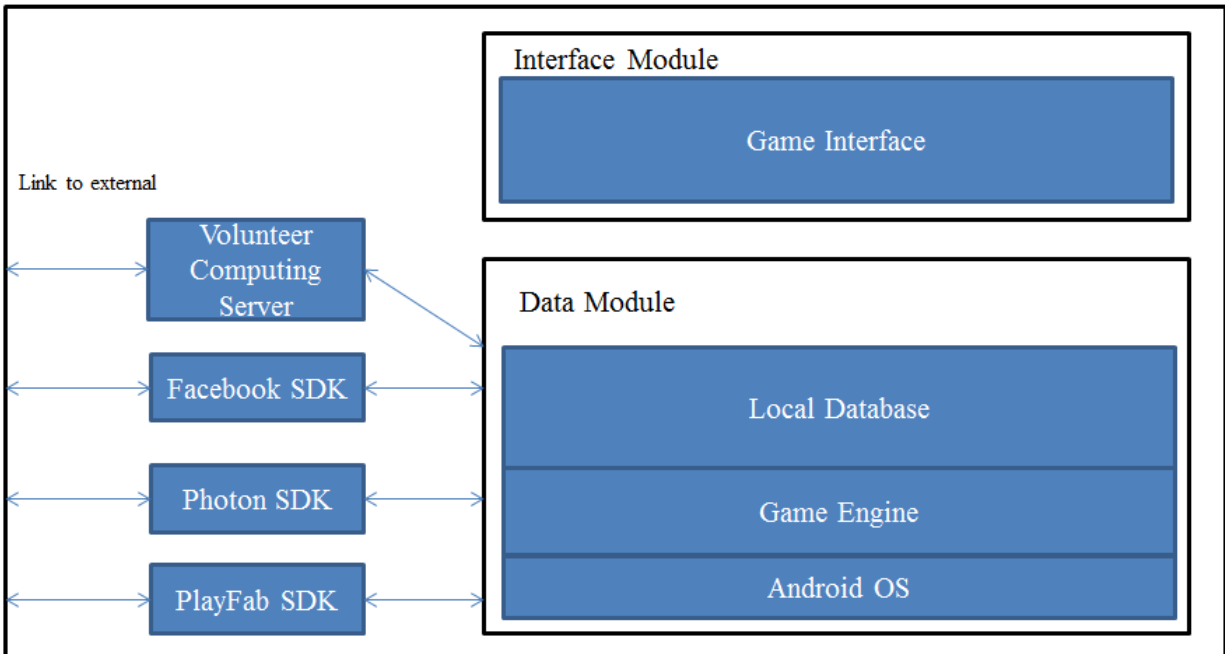


Figure 3.2 Stack diagram

This diagram shows the overview of our system, it can be divided into data module and interface module. Data module is a game back-end system that can communicate to other application via SDK (Software development kit) [21]. Interface module is use for interaction with player.

3.6 Use case diagram

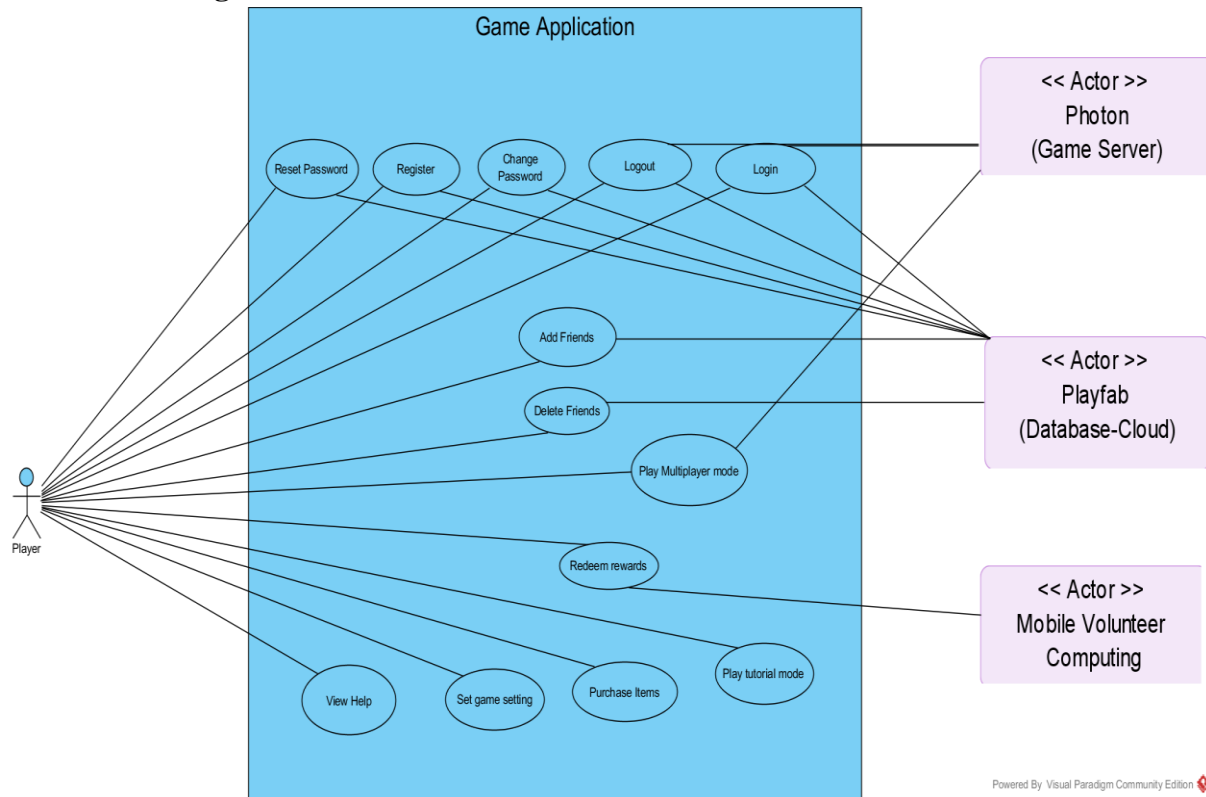


Figure 3.3 Use Case diagram

3.6.1 Use case narrative

Scenario 1: Register using a game account

Goal: To be a member of the system

Preconditions: -

Main success scenario:

1. A user presses "Register"
2. A user enter a new account information
3. A user press "Confirm Registration"
4. The system shows a success message

Alternative scenario:

- a. A user enter his/her existing username
 1. The system shows an error message
 2. The system restarts the registration process

Scenario 2: Login

Goal: To login into game

Preconditions: User must have a game account

Main success scenario:

1. A user press “Login with a Game Account”
2. A user enter login information
3. A user press “Login”
4. The system shows a login success message
5. The system redirects to the main menu page

Alternative scenarios:

- a. A user enters wrong login information
 1. The system shows an error message
 2. The system redirects back to the login page

Scenario 3: Logout

Goal: To logout from game

Preconditions: User must already login

Main success scenario:

1. A user press “Logout”
2. The system shows a confirmation dialog
3. A user press “Logout”
4. The system shows a login success message
5. The system redirects to the main menu page

Alternative scenarios:

- a. A user press cancel button at the logout confirmation dialog
 1. The system closed the logout dialog

Scenario 4: Change password

Goal: To change user password

Preconditions: User must have game account and already login

Main success scenario:

1. A user presses “Change password”
2. A user enter old password
3. A user enter new password
4. A user select submit
5. The system shows a success message

Alternative scenarios:

- a. A user enters wrong old password
 1. The system shows error message
 2. The system clears password and let user enter again

- b. A user enters new password with wrong format
 - 1. The system shows an error message
 - 2. The system clears password and let user enter again

Scenario 5: Reset password

Goal: To reset user password [In case of user forgot password]

Preconditions: User must have game account

Main success scenario:

- 6. A user presses “Forgot Password”
- 7. A user enters their E-mail
- 8. A user presses “Submit”
- 9. The system shows success message
- 10. A user enter temporary password from an E-mail sent to the user
- 11. A user enters a new password

Alternative scenarios:

- a. A user enters wrong Email
 - 1. The system shows error message
 - 2. The system redirects back to the login page
- b. A user enters new password with wrong format
 - 1. The system shows an error message
 - 2. The system clears password and let user enter again

Scenario 6: Play Multiplayer Mode

Goal: To play game

Main success scenario:

- 1. A user presses “Play”
- 2. A user selects game mode
- 3. A user selects matchmaking method (Quick join or Create room)
- 4. A user selects his/her hero’s weapon and clothes
- 5. A user presses “Ready”
- 6. Players in the room are in the ready status
- 7. The system enable start button to clickable
- 8. Room header presses start button
- 9. The system redirects to battle field scene

Alternative scenarios:

- a. Room header does not press start after all players ready for a while
 - 1. Room header is kicked off
 - 2. The system selects new room header
- b. A user does not select his/her hero’s weapon and clothes and press “Start”
 - 1. The system shows warning message
 - 2. A user selects “Confirm” or “Cancel”

Scenario 7: Play Tutorial Mode

Goal: To play in tutorial mode

Main success scenario:

1. A user presses “Play”
2. A user selects game mode
3. A user presses “Start”
4. The system redirects to battle field scene

Scenario 8: Add Friend

Goal: To add friends with existing user

Main success scenario:

1. A user presses “Social”
2. A user enters player id to add
3. A user presses “Add”
4. The system shows friend information and confirmation message
5. A user presses “Confirm”

Alternative scenarios:

- a. A user enters wrong friend id
 1. The system shows an error message
 2. The system redirects back to social page

Scenario 9: Delete Friend

Goal: To remove friends with existing user

Main success scenario:

6. A user presses “Social”
7. A user select player id to delete
8. A user presses “Delete”
9. The system shows friend information and confirmation message
10. A user presses “Confirm”

Scenario 10: Redeem Rewards

Goal: To redeem In-game special currency from cycle donation

Main success scenario:

1. A user presses on the “Redeem Reward”
2. The system shows current reward points and number of donated cycles
3. A user presses on the “Redeem”
4. User selects amount of donated cycles to exchange
5. A user presses “Redeem”
6. System deducts the donated cycles
7. The system shows a success message

8. System added In-game currency

Alternative scenarios:

- a. A user does not have enough points to redeem
 1. The system shows an error message

Scenario 11: Purchase Items

Goal: To purchase items (Weapons or Clothes)

Main success scenario:

1. A user presses “Shop”
2. A user selects item that want to purchase
3. A user presses “Purchase”
4. The system shows a confirmation message
5. A user presses “Confirm”
6. The system deducts the money or special currency
7. The system shows a success message
8. The system sends the item to user inventory

Alternative scenarios:

- a. A user does not have enough money to redeem
 1. The system shows an error message
- b. A user does not have enough special currency to redeem
 1. The system shows an error message

Scenario 12: Set game setting

Goal: To set sound and music of game

Main success scenario:

1. A user presses “Setting”
2. A user adjusts sound or music of game
3. A user presses “Save”

Alternative scenarios:

- a. A user adjusts volume but not presses “Save” before exit that page
 1. The system shows a warning message
 2. A user selects whether to confirm or discard their setting

Scenario 13: View Help

Goal: To see game help

Main success scenario:

4. A user presses “Help”
5. The system shows a game help information
6. A user presses “Ok”
7. The system redirects back to main menu page

3.7 Sequence diagrams

3.7.1 Scenario: Login

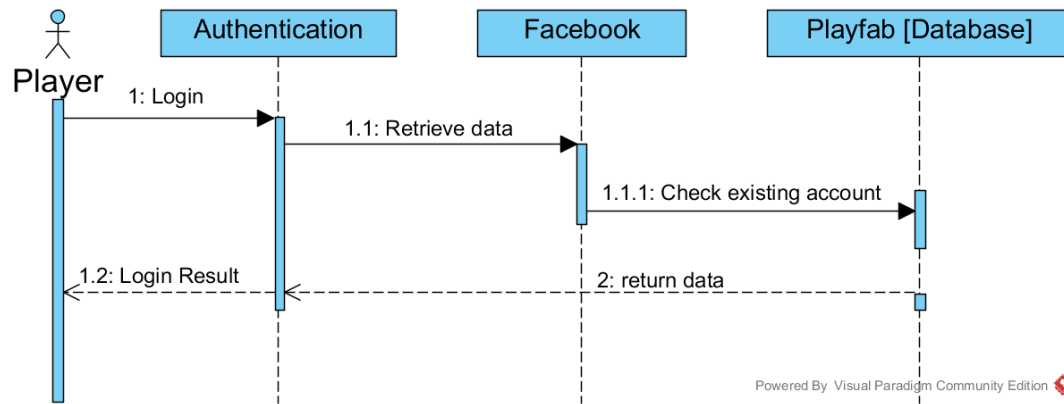


Figure 3.4 Scenario: Login

3.7.2 Scenario: Matchmaking

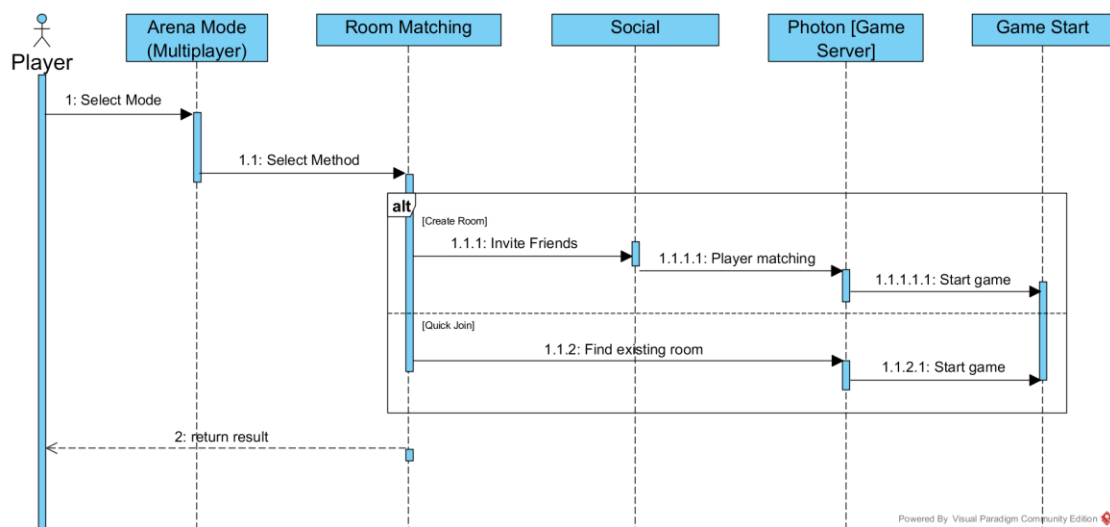


Figure 3.5 Scenario: Matchmaking

3.7.3 Scenario: Playing

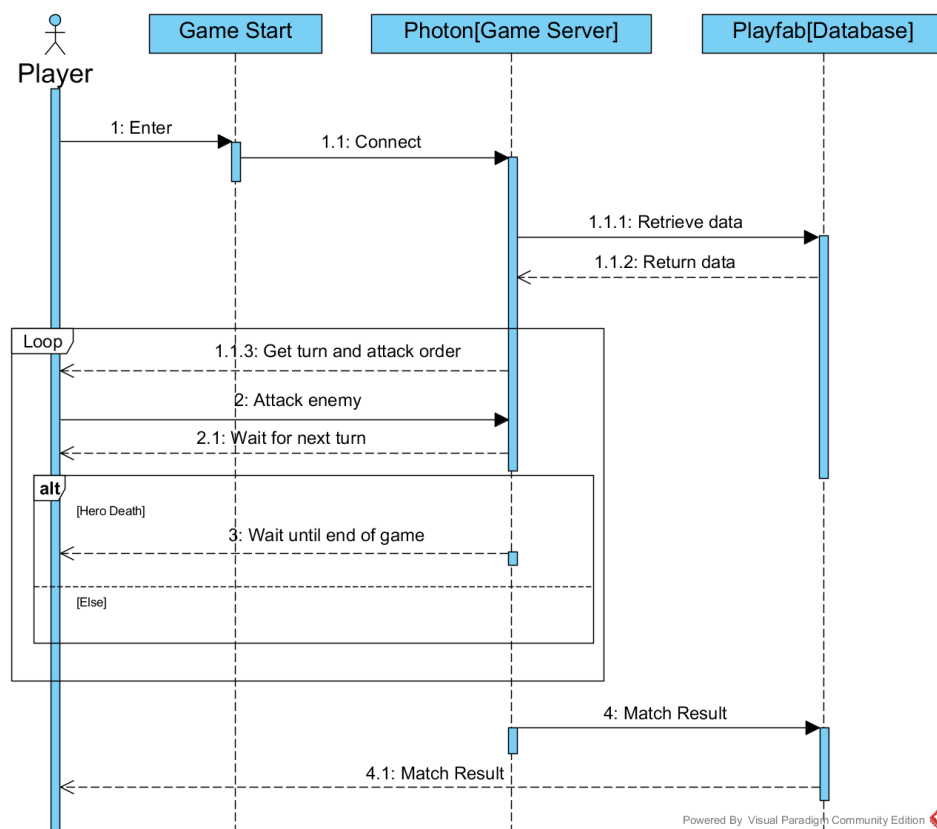


Figure 3.6 Scenario: Playing

3.8 Game story prologue

Long time ago, there are brave adventures, who seeking for power to help the word from the dark. In the deepest dungeon they found the existing of 'Magic Particle'. They feel the great power inside it. But not only human who seeking for power. Awaken of the force also attract the demon. Human can't resist the power of demon. It take the power of Magic Particle and try to destroy human. However, goddess ALICE who has watched and protect human appear to help. She use her power to drive out demon. The demon escape and disappear into the shadow. Peace has now return, but ALICE warn human to prepare if the demon is come back for revenge. ALICE give human knowledge about Magic Particle and teach them how to use it power to fight against demon. This is the beginning of great competition for human to practice their skill and knowledge so they can use to fight with demon once it return. And now it is your time to prove your skill in this arena. Are you ready?

3.9 User interface screen layout design



Figure 3.7 Shop

This scene show the items shop, sell costume. There are two currency in this game, gold can use to purchase ordinary costume and diamond can use to purchase special costume.

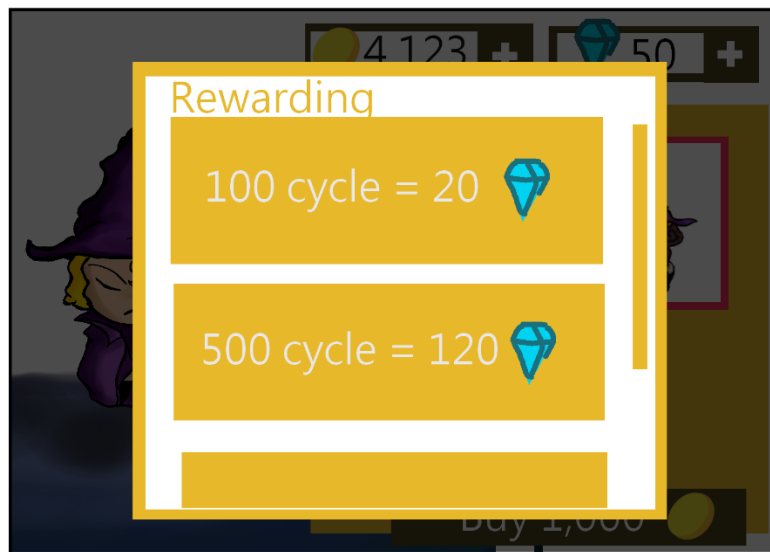


Figure 3.8 Rewarding

This scene is the rewarding, player can exchange the donated cycles to the special currency that represent as a diamond.



Figure 3.9 Character design and Weapon

Each character consist of three main parts

- Clothes (Green color): Character decoration, not affect the attack power and defense power
- Attack items (Pink color): Increase attack power, different for each item
- Defense items (Blue color): Increase defense power, different for each item

Player need to concern about items weight, total weight for both items (Attack items and Defense items) cannot exceed the maximum limit

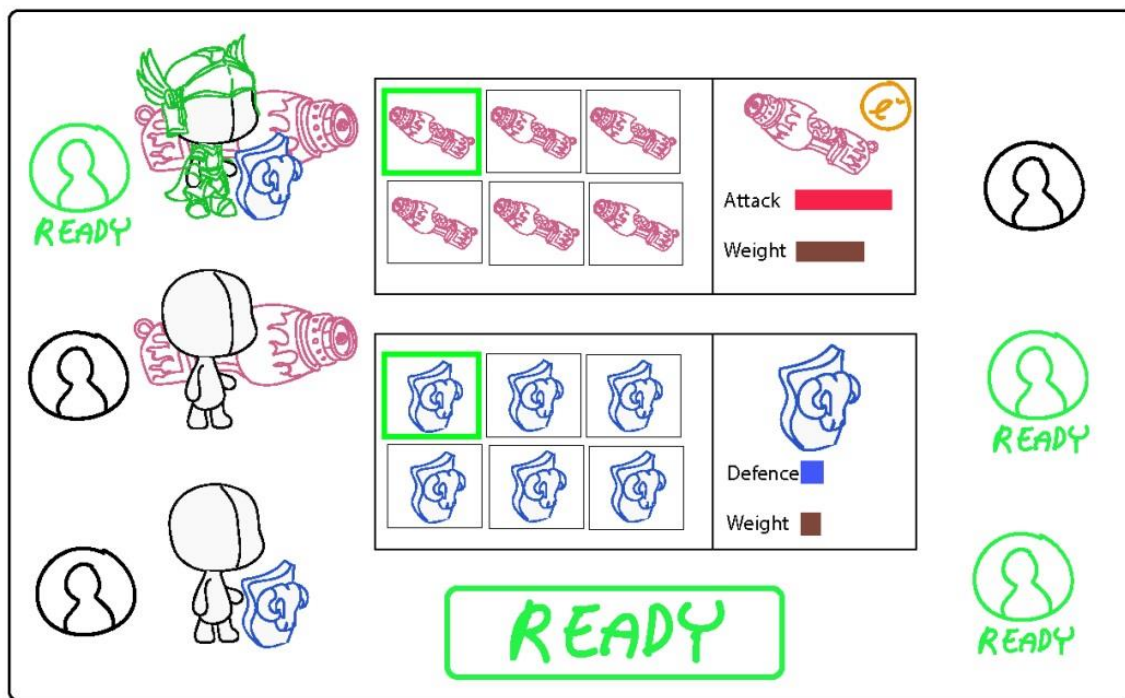


Figure 3.10 Waiting Room

Player has two ways to play a game

- Quick Join: randomly find a public room and join
- Create Room: Create new room, 2 teams with 3 player/team in each room

Player need to wait until everyone in the room are in the “Ready” status indicated by a green icon, then room header can start the game, player can invite their online friend to join the room

Before game start, players need to wait in the waiting room. In this room, player can change attack item and defense item of their character, each character can select only 1 attack item and 1 defense item, each item has different properties that related to the properties of particle physics. Player need to concern total weight of both selected items, total weight cannot exceed the maximum weight

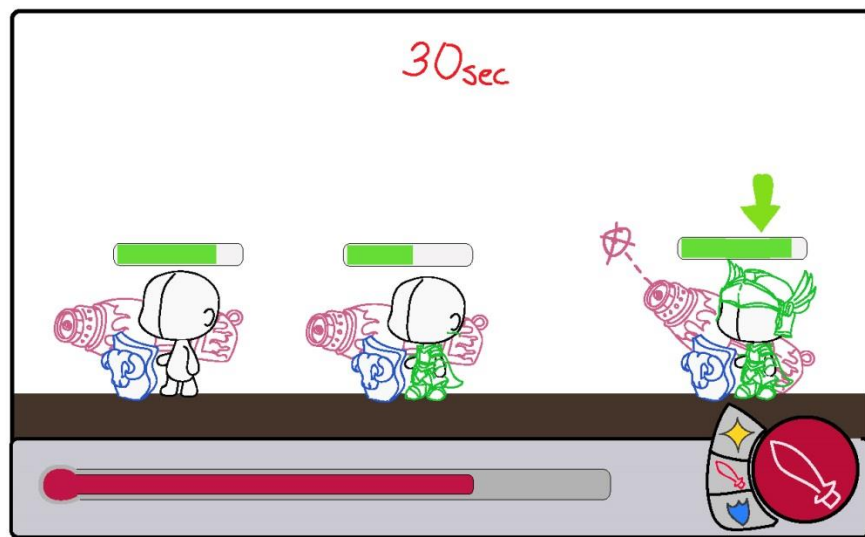


Figure 3.11 Character orientation

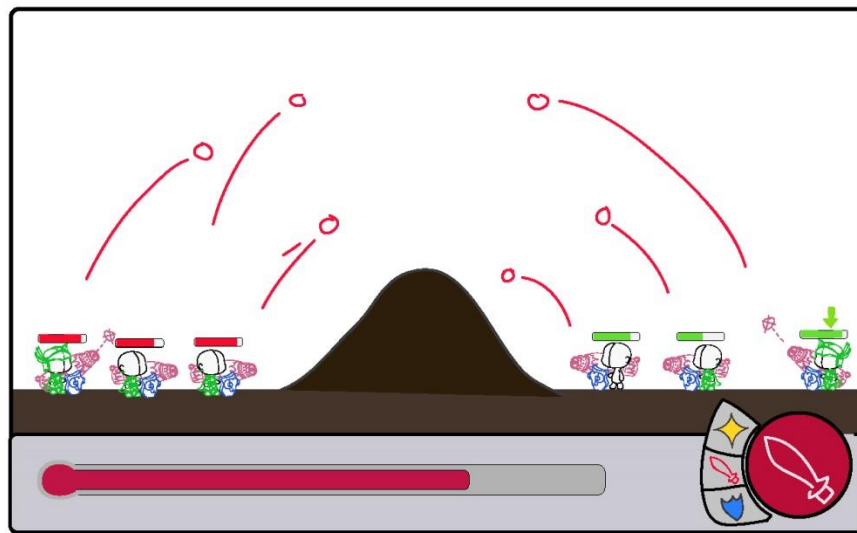


Figure 3.12 Start battle

After game was started, positions of each character are randomly assign that mean the player does not know their position before game start, the weapon of each player will be set to the first attack method by default, player can choose attack skills by press the button at the bottom of screen

The first attack method is a basic method for every weapon that has the damage equal to the default status of each weapon

The second attack method is a special method that has a higher damage than the first method, player can use this method once for each match

The player can adjust the shooting angle by touching at the aim and slide in the upward or downward direction. The angle range is between 90° and -90° . After that, the player can press and hold “Shoot” to shoot the enemy. The longer button hold the higher initial velocity of the shooting object, show as a bar in the bottom of screen but if the player does not release until the velocity reach the maximum value, the initial velocity will be set as the maximum value.

Each player has 30 seconds for selecting an attack method and shoots or confirms. If the player neither selects to attack nor to defense within 30 seconds, the system will automatically pass that player and no attack action occurred.

There are three conditions to end each match

- Condition: “Victory”
Match will end with the “Victory” condition if your team can eliminate all of the heroes of the rival team
- Condition: “Defeat”
Match will end with the “Defeat” condition if all of your team heroes were killed by the rival team before the time is up
- Condition: “Draw”
Match will end with the “Draw” condition if both the teams cannot eliminate each other within the match time

3.10 Table of items properties

Item	Types	Charge	ATK	DEF	weight	Skills	Cool down
Proton cannon	Attack	+	25	-	45	Normally attack	
						Double attack	3
Antiproton cannon	Attack	-	25	-	45	Normally attack	
						Special attack (Increase damage 40%)	3
Electron cannon	Attack	-	35	-	55	Normally attack	

						Attack and increase attack power 15% for 3 turns	4
Positron cannon	Attack	+	35	-	55	Normally attack	
						Attack and increase attack and defense power 25%	4
Neutrino cannon	Attack	0	20	-	35	Normally attack	
						Attack and increase attack power 20% of all allies	4
Antineutrino cannon	Attack	0	20	-	35	Normally attack	
						Attack and increase attack and defense power 10% of all allies	3
Proton shield	Defense	+	-	40	40	Recover health point of player 25%	4
Electron shield	Defense	-	-	50	60	Increase defense power of all allies 20%	3
Neutron shield	Defense	0	-	65	75	Increase defense power and recovery health point 20%	4

3.11 Refutation table

Weapon charge	Armor charge	Attack damage
+	+	Decrease 50%
+	-	Increase 25%
+	0	100%
-	-	Decrease 50%
-	0	100%
0	0	100%

3.12 Database schema

3.10.1 Player

Primary Key: ID

Foreign Key: CharacterID

The player database stores the necessary information about the player such as ID, Email, Facebook account, EXP, and characters that the player has.

3.10.2 Character

Primary Key: CharacterID

Foreign Key: PlayerID

The character database stores the necessary information of each character such as name, health, cloth and Item which will be set to default at the beginning of the game.

These can be customized by the user later.

3.10.3 Item

Primary Key: ItemNo

Foreign Key: Type

The item database stores the item information such as name, value, and price.

3.10.4 ItemType

Primary Key: Type

The type of item store the item type which are attack and defence. Each character can have both type of item at the same time but only 1 item per type for each character.

3.13 Entity-Relationship diagram

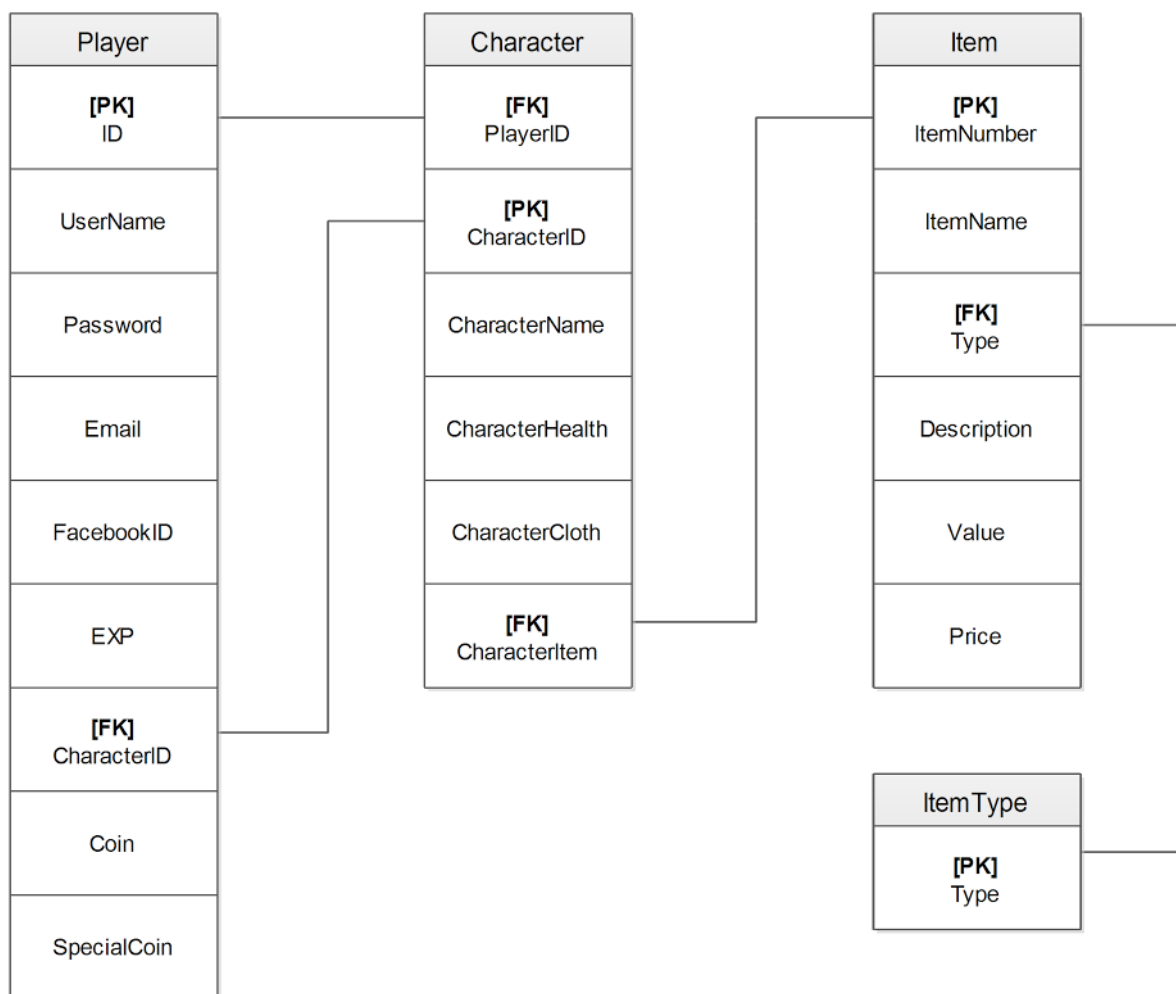


Figure 3.13 ER-diagram

Chapter 4

Results and Discussion

4.1 Game screenshots

4.1.1 Game title screenshots

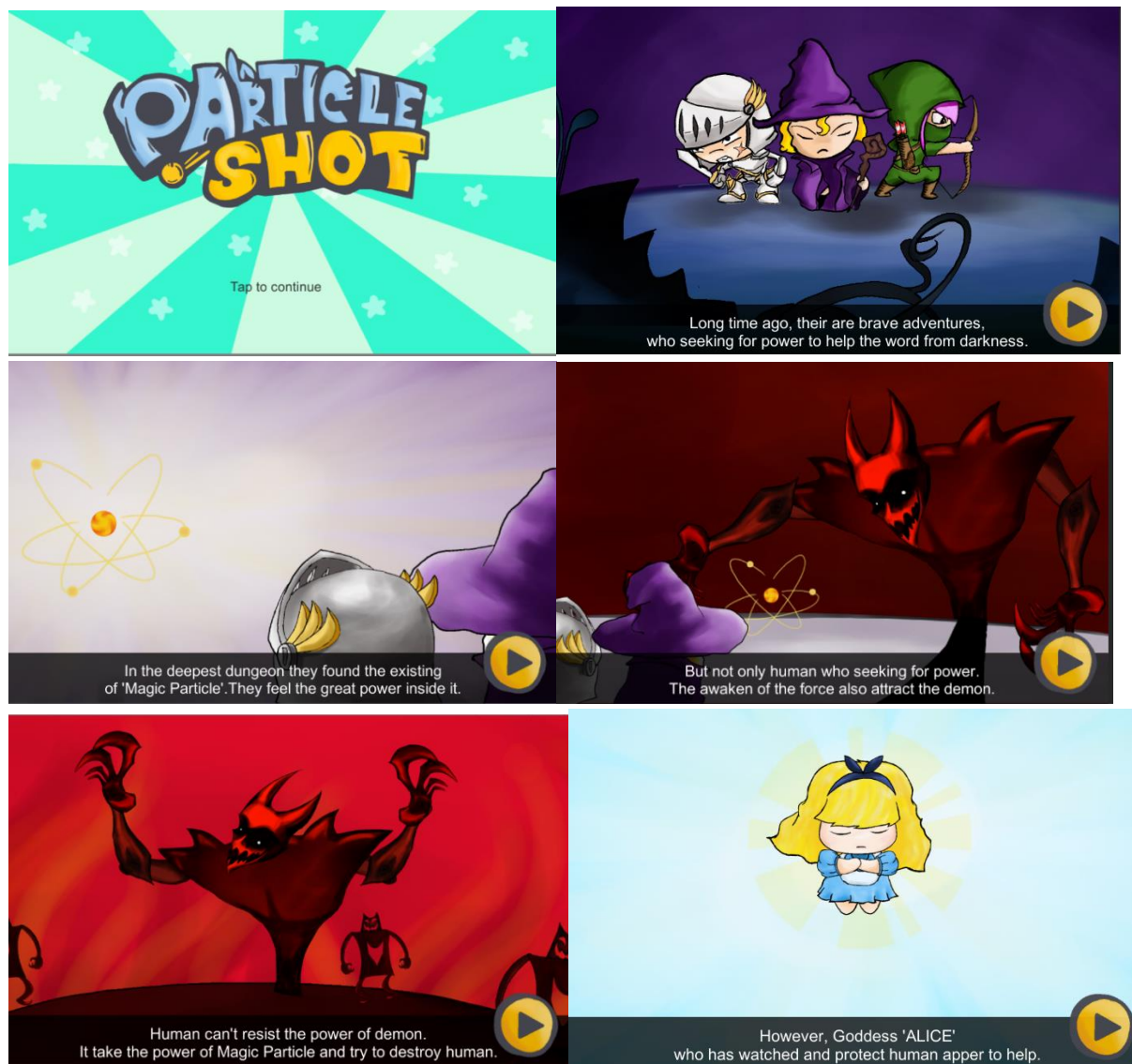


Figure 4.1 Game story 1

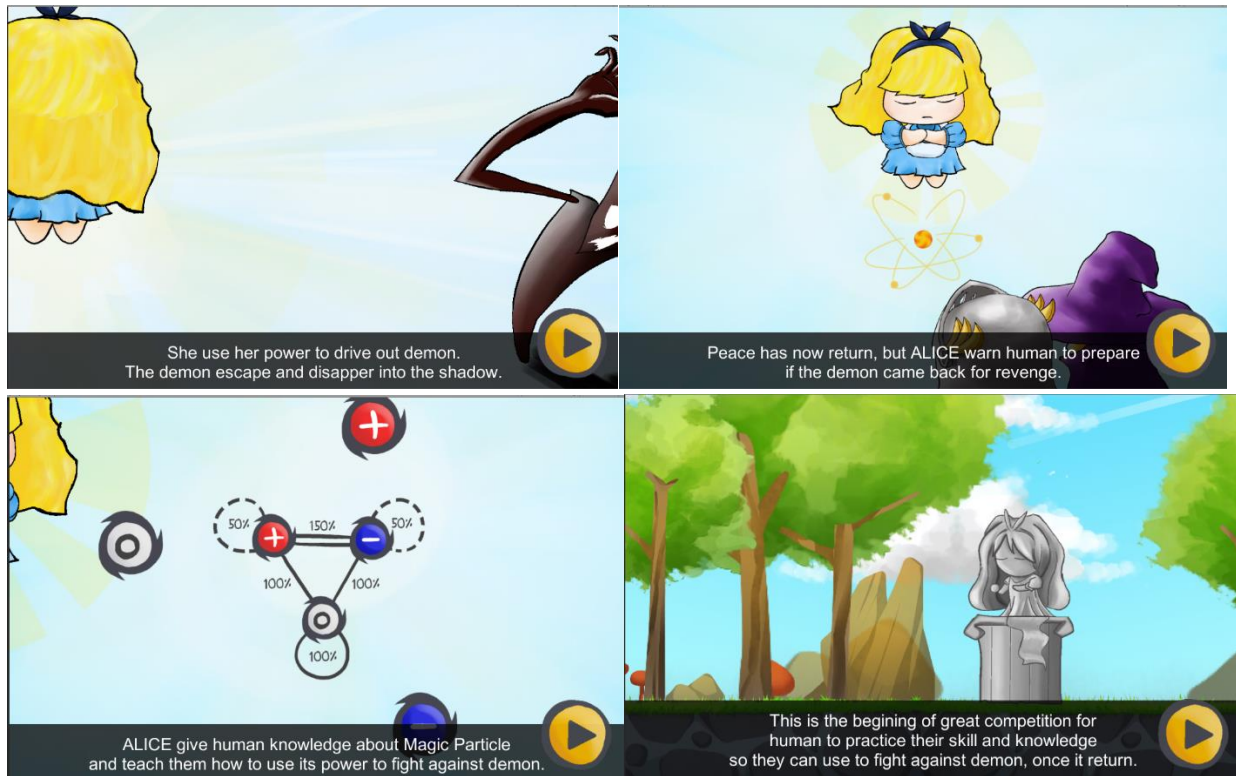


Figure 4.2 Game story 2

The first screenshot of the game is the title screen which a player needs to touch it to continue to the next screen. Other scenes after the first one are the animated story of the game described in Chapter 3 (prologue). The player needs to touch the “>” button in order to forward to the next scene.

4.1.2 Login/Register & Lobby screenshots



Figure 4.3 Login/Register scene

Before the player starts the game, the player must first login to our game. If the user has already registered, the player can fill in her username [1] and password [2], and press the login button [4] or press Log in with Facebook button [5]. However, if the player has never registered, the player has to press the register button [3] or registered via Facebook account button [5] for the game registration.



Figure 4.4 Register dialog 1

In the first step of registration process, the player must fill in the following information

- [1] Username - this will be used to login to the game.
- [2] Password - this will be used to login to the game.
- [3] Confirm the password – this will be used to check whether the password is typed correctly by comparing it with the previous one.
- [4] E-Mail address – this will be used for sending news or information of the game to the player.

The player can press the back button [5] to cancel the registration process or press the next button [6] to go to the next step.



Figure 4.5 Register dialog 2

Then, the player has to define his or her display name [1]. The display name will be shown to other players while the player is playing the game. The player can press the back button [2] to go to edit the player information or press the next button [3] in order to confirm the registration.



Figure 4.6 Completion of registration process

After the registration is complete, there will be the message “Registration Complete!!! Welcome to our land” [1] to notice that the player has completely registered. The player then presses the ok button [2] to further login to the game.



Figure 4.7 Lobby scene

The above picture shows the main menu of the game which is composed of the following components.

- [1] Play button - Pressing this button to go to the scene of the game mode selection.
- [2] Volunteer computing button - Pressing this button to redeem a gift from the mobile volunteer computing platform.
- [3] Leaderboard button - Pressing this button to view the leaderboard.
- [4] Setting button - Pressing this button to adjust sound volume or log out.
- [5] Inventory & store - Pressing this button to view the inventory or to purchase new items.



Figure 4.8 Lobby scene2

After the user presses the play button, the user has to choose one of the play modes.

- [1] Quick join - The player will join an available room randomly and automatically.
- [2] Create room - The player can create a new room and then wait for other players to join.

4.1.3 Store & inventory screenshots

Inventory



Figure 4.9 Inventory



Figure 4.10 Inventory2

This scene shows items in the player's inventory. There are two characters which are a boy and a girl when the player plays the game for the first time. The first character will be set as the default character. The player can change his or her default character by touching one of the characters shown in the right panel [1]. The selected character will be shown in the left panel as the default one [2]. The player can select only one default character at a time and this character will be used in the game. More characters are available to be purchased in the store [3]. The user can close the inventory by touching the "X" button at the top-right corner [4]. The balance of the user's virtual currency is also shown at the top of page [5] [6]

Store



Figure 4.11 Store

The player can buy new or exclusive characters in the store. Currently, for the first version of our game, the items in the store are characters with costumes only. Each item's attributes are name, description, and price. The player can purchase a character by touching the “buy” button [1]. Most of the characters can be purchased by gold coins [2] (earned by playing in the game) but some characters must be purchased by diamonds [3] (only earned by donating CPU's cycles for the volunteer computing). The player can close the store by touching the “X” button [4] at the top-right corner.

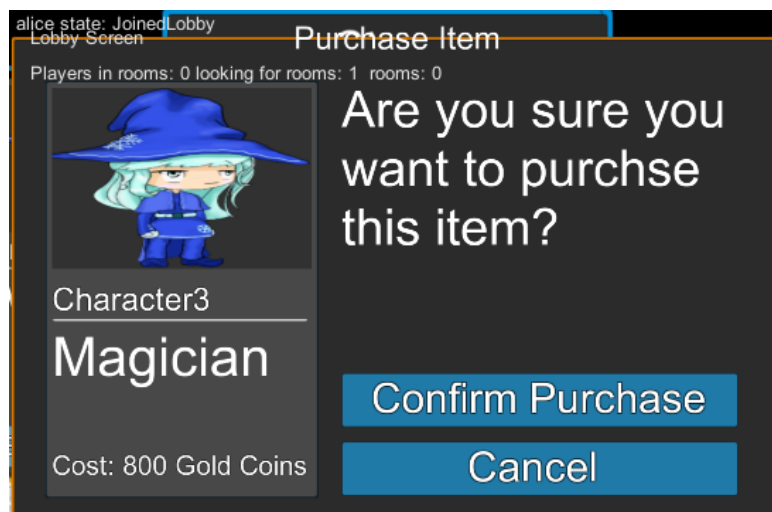


Figure 4.12 Confirm purchase

This purchase confirmation dialog will be popped up after the user presses the “buy” button [1] on the selected item. The player then choose to either purchase the item or cancel it.

Purchase result

Purchase succeed

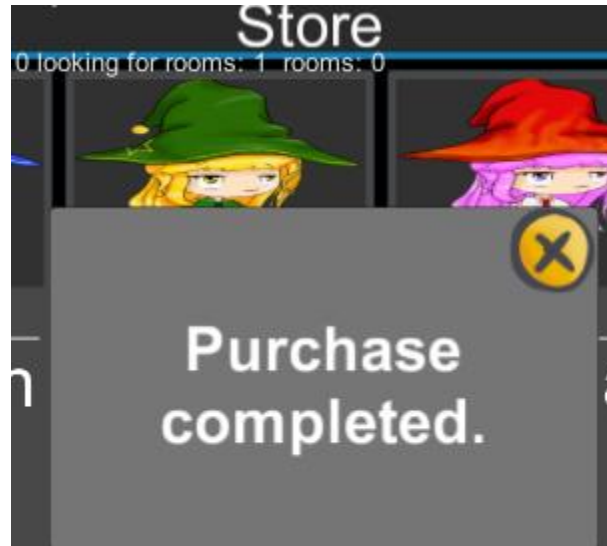


Figure 4.13 Purchase succeed

The purchase will be completed only if the player has enough coins or diamonds to purchase that item.

Purchase failed

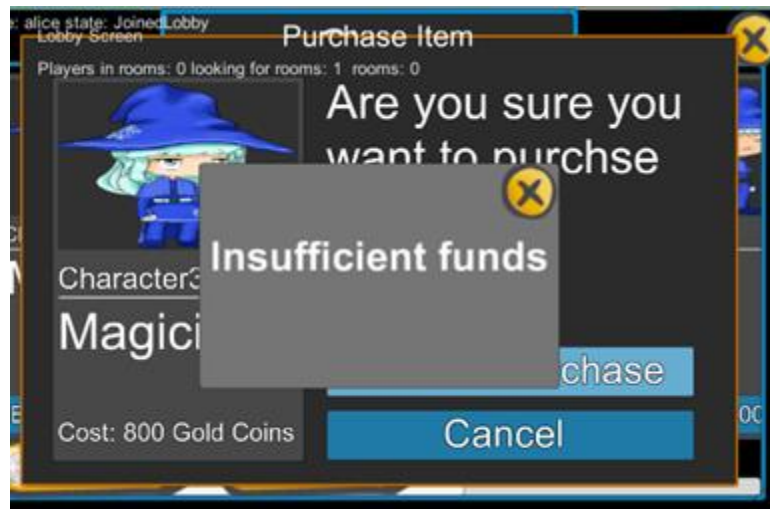


Figure 4.14 Purchase failed

If the user is unable to purchase the selected item, the user will see an error dialog showing a reason of failed purchase, e.g. not enough virtual currency and user purchased that character already.

4.1.4 Donated CPU's cycle redemption screenshots

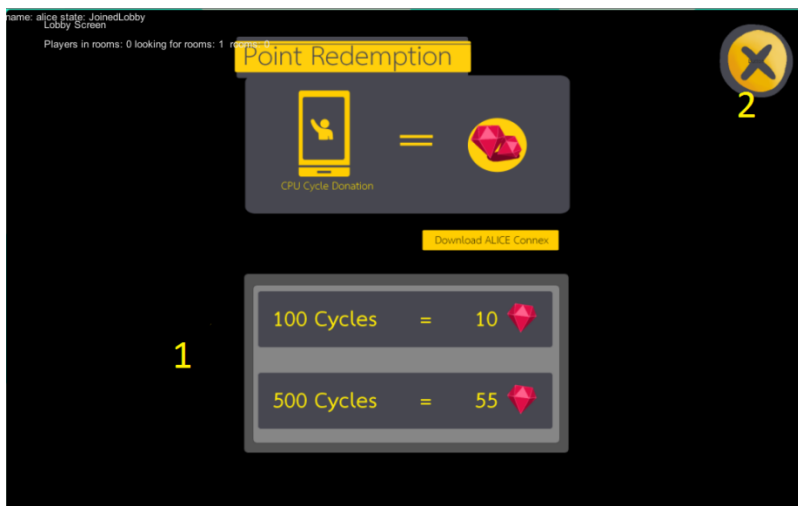


Figure 4.15 Redemption

The player can redeem donated CPU's cycles for diamonds which can be used for purchasing exclusive characters [1]. The player can close this page by touching the "X" button on the top-right corner [2].

4.1.5 Leaderboard screenshots

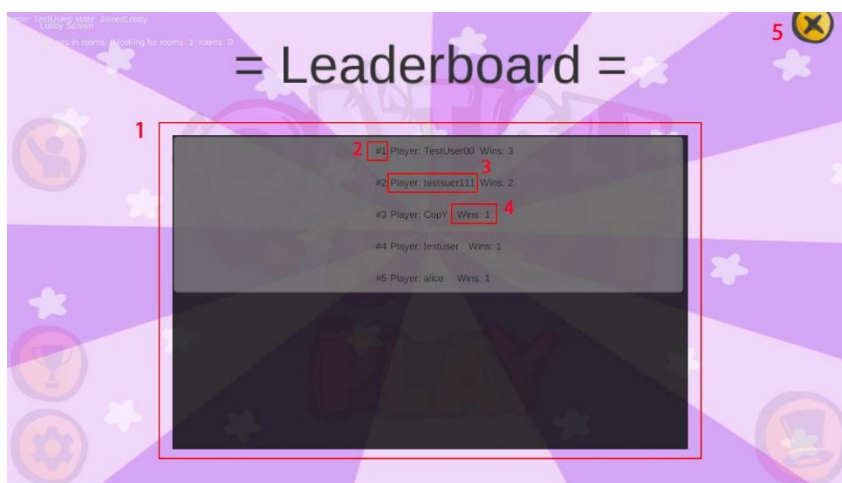


Figure 4.16 Leader board

The leaderboard is used to show the player ranking. In the middle of the screen, there is a section [1] which is used to display the list of the players. In each rank, there are several components.

[2] The rank of the player.

- [3] The player name.
 [4] The number of wins which is used to rank all the players
 The player can close this leaderboard by pressing the exit button [5].

4.1.6 Waiting room screenshots



Figure 4.17 Waiting room

In the waiting room, the room header will always get the 1st position and the next joined player will get the position 2,3,4,5, and 6 respectively



Figure 4.18 Waiting room2

Each player slot displays the detail of the player such as player's name and selected weapon [1]. For an empty slot, "Empty" will be set as the name and the weapon

will leave blank [2]. The player can select a weapon at the weapon panel [3]. The current selected weapon will be shown at [1] and [4]



Figure 4.19 Waiting room3

The room header can control player slots in the waiting room to be closed or open. A closed slot will be shown as a master key image and other players will be not allowed to take that position. Each player presses the ready button when the player is ready for playing the game. After the user is in the ready state, the user cannot change his or her weapon or leave the waiting room. The game can be started only if there are two or more players in the room and all the players are in the ready state.



Figure 4.20 Weapon detail

To see the detail of the selected weapon, the user needs to touch the “?” button [6]. Then a dialog showing the weapon description, charge, and attack/defense power will be popped up.

4.1.7 Battle field screenshots



Figure 4.21 Battle field scene (Large)

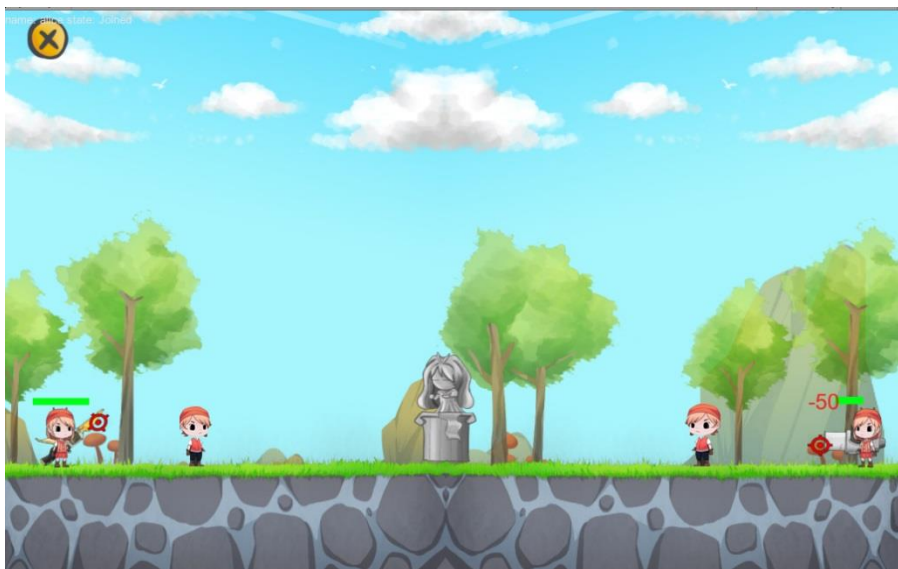


Figure 4.22 Battle field scene (Small)

In the battle field scene, the player can see other players in the same team. However, the player can view the opponents by using two-finger dragging through the screen and zoom in-out by pinching in-out [Figure 4.21].



Figure 4.23 “Aim-and-fire” control

This screenshot shows the “aim-and-fire” control of the leftmost player. Each player can control the aim angle by pressing and holding the aim point [1]. After that the player needs to press and hold the power button [2] in order to choose the level of attack power. The attack power bar will increase continuously until the player releases the button. Both the angle and power should be set in order respectively. The player has only 15 seconds for selecting the aim and attack power. If player does not submit the action in time, the game will skip the player’s action in this playing round; i.e., no action will be automatically chosen. Note that the player will be unable to change the power and aim after releasing the button.

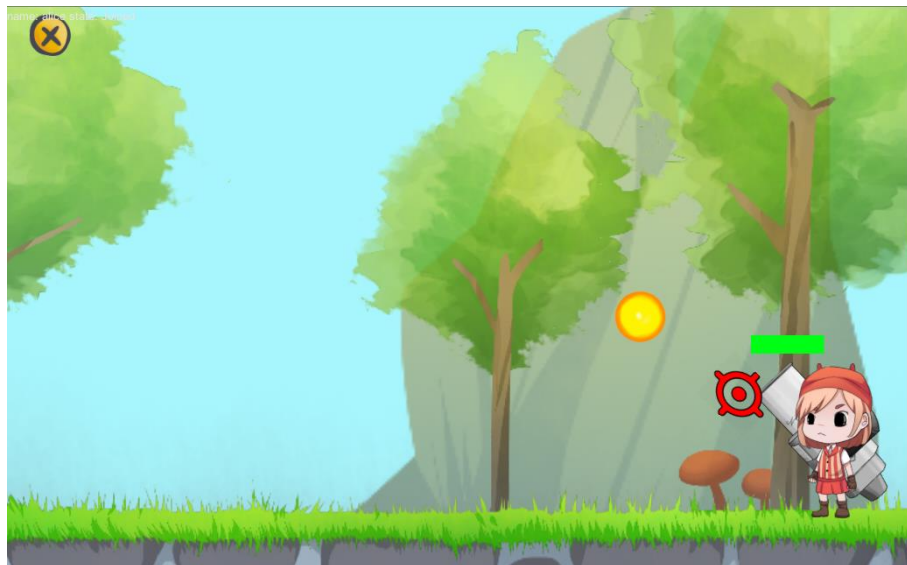


Figure 4.24 Fired action

Shooting action will be performed only if all the players submit their actions by releasing the power button or the time is out. After that all the players will simultaneously shoot and then the current round ends.

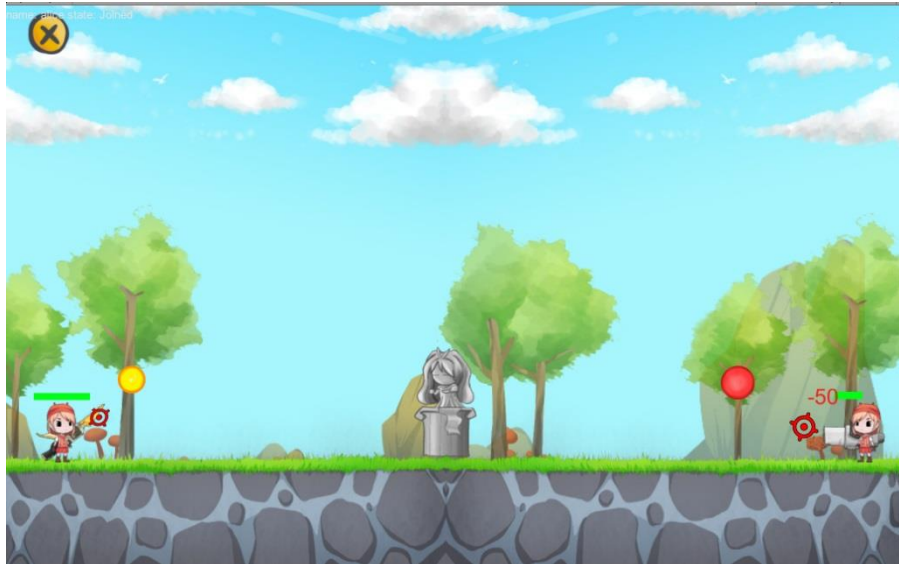


Figure 4.25 Fired action scene (Small)

If any player is shot, the damage will be shown at the player. If any player loses all of his or her health points or HP, the player will be unable to shoot in the game anymore and that user's character will be removed from the current playing scene. For any player that was attacked, the player's HP will decrease but the player will still be able to play in the next round. The game will be repeatedly played until one team loses all the members.

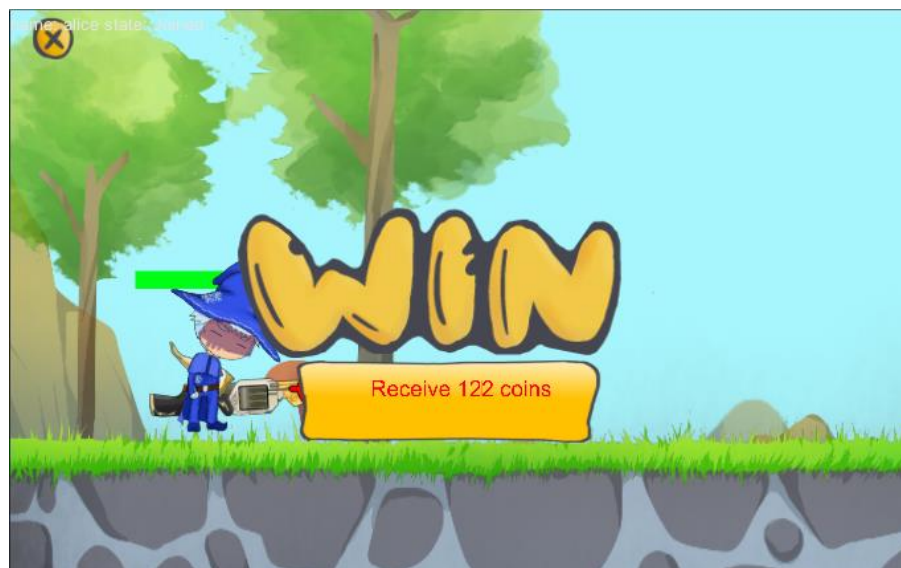


Figure 4.26 Win

At the end of the game, there will be a popup on the winner team screen indicating that the team wins the game and an amount of coins that each player has earned.

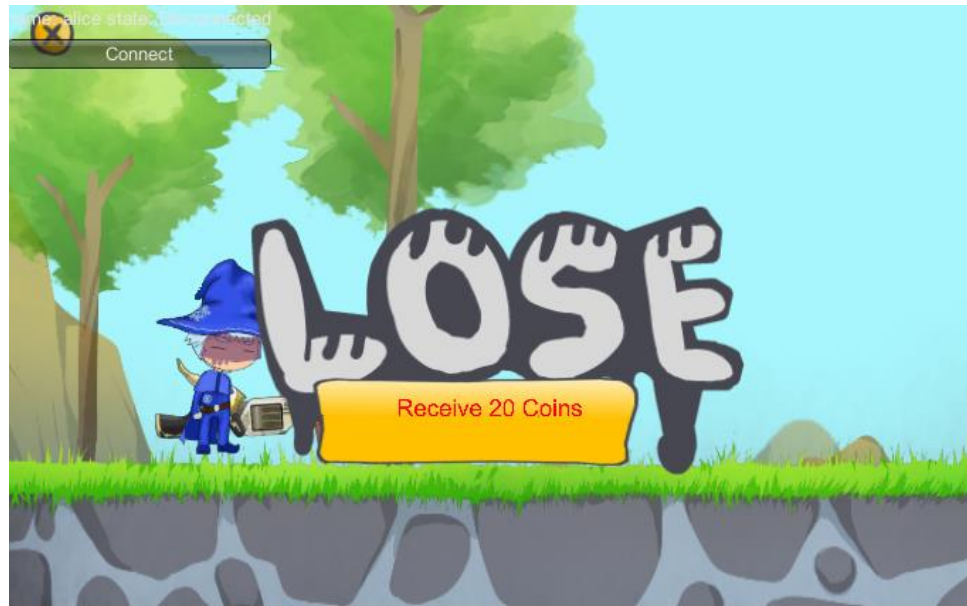


Figure 4.27 Lose

On the other hand, there will be a popup on the loser team screen indicating that the team loses the game and an amount of coins that each player has earned.

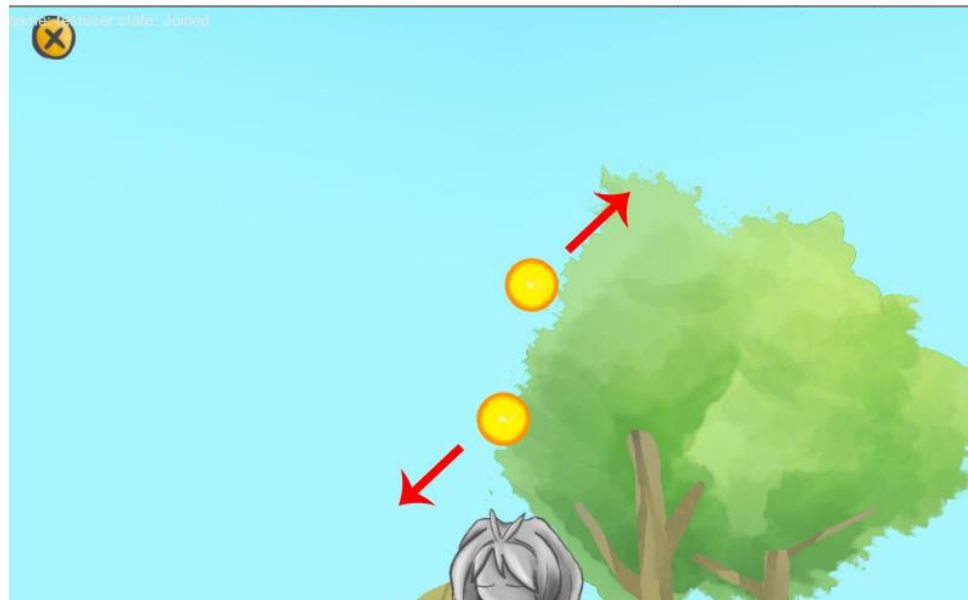


Figure 4.28 Push force1

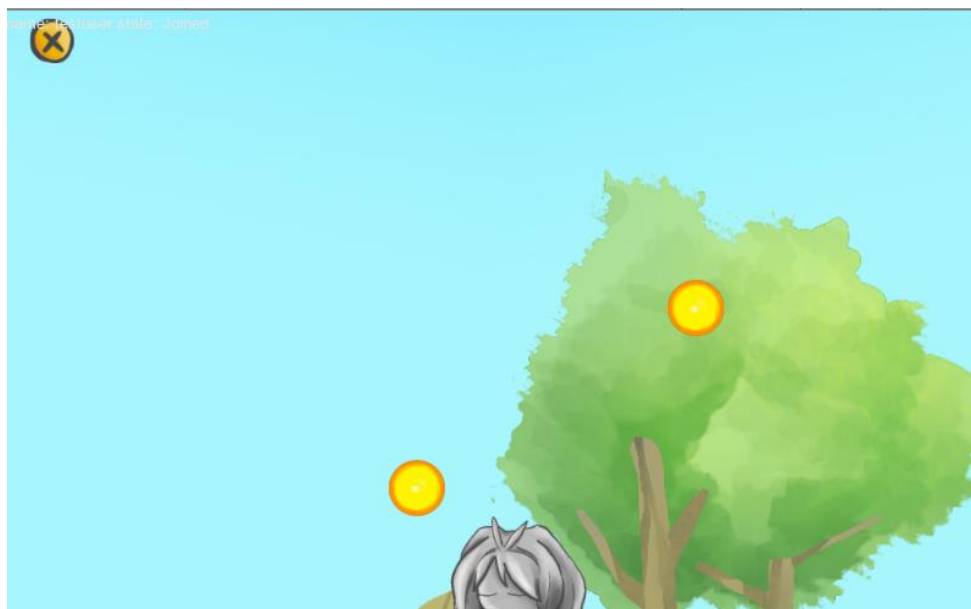


Figure 4.29 Push force2



Figure 4.30 Pull force1

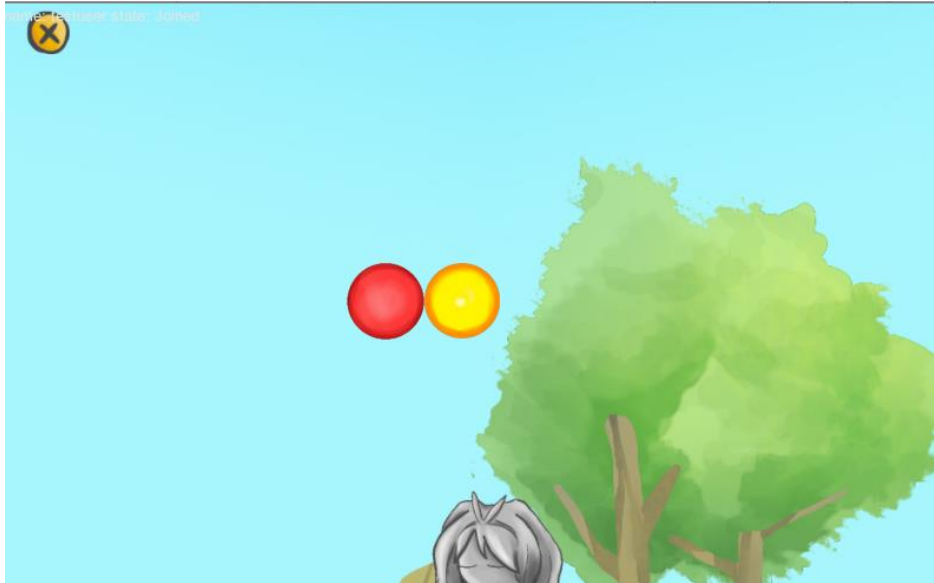


Figure 4.31 Pull force2

Particles with electric charges will interact with each other when they are close together. In this game, we simulate the electric force of when a particle interacts with the other one. If the particles have the same electric charges, it will push each other to the opposite direction [Figure 4.28 and 4.29]. On the other hand, if the particles have the different electric charges, it will pull the other to come closer [Figure 4.30 and 4.31]. The changing of speeds and angles of both the particles will conform to the Newton's laws of motion.

4.2 Performance data

There are our game performance criteria.

- Photon server's initialization and connection time : 1 - 2 s
- Processing time for creating a room: 10 - 20 ms
- Processing time for joining a room 5 - 10 ms
- Processing time for starting a battle : 3 - 6 s
- In-game connection delay : 500 - 1000 ms
- PlayFab database initialization and connection time 2 – 3 s
- Processing time for registering an account 2 - 3 s
- Processing time for logging in 1 – 2 s
- Processing time for completing a purchase transaction 3-4 s.
- Response time for opening the inventory page 200 – 300 ms
- Response time for opening the store and loading data from PlayFab 3-4 s
- Response time for opening the reward redemption page 200 – 300 ms
- Processing time for ending a battle and summarizing the battle result : 3 - 5 s

Since we are using Photon to manage the game room and synchronize among users in the game, Photon will distribute all actions to all users. However, when users sign in to the game not at the same time, some users may be in a different session from others. For example, on one player's screen, this user are dead and disappear but the user character still appear on the others' screens. To solve this problem, we will delay to start the game to make all users sign in into the same session.

There are some problems during our developing period. Although our game style is round based - mixing between turn based and real time style. However, the server provider has only two types - turn based server and real time server. We have decided to use a turn-based game server because it uses less bandwidth to exchange data and it causes less delay compared to a real-time game server. The turn-based game will store the data of each round and the data will be shared among other users.

Another problem is about Playfab. Since we are new to PlayFab API, we need to find several sample projects and try to understand a lot of functions and syntax. Furthermore, most of manuals in the PlayFab website are simple documents and do not describe any method to integrate with the Unity game engine. So, we need to find a lot of sample projects and see a lot of tutorials to apply Playfab in our game.

However, we come up with several problems with turn-based game server. First, the difference between round-based game and turn-based game makes coding more difficult. The logic of round-based game is very new. Normally, the turn based game will end each turn when a player choses her action. However, the round-based game will end each turn when all players in the game scene chose the actions. Second, it is difficult to send the data to all users without a data manager. Since the Photon server is only a way to connect users together, it does not have any manager to store and manage the incoming data from users. Each user has to update the data

of others by itself which causes a lot of processes. This problem leads to another problem when we implement the waiting room. The available position should be broadcasted to a new user who are going to join the room. We have tried to update the position among the players in the room and update it to the new user. However, if the new user joins the room before the data is updated, that user might be located at an incorrect position. We solve this problem by making a room leader. The room leader will update the room status and manage the user who is joining the room.

Since our game is almost complete, we experimented our game with a sample group and we found some problems about the user experience. First problem is that the user interface has too many colors. Some players cannot separate between the game logo and touchable buttons. The second problem is that there are too many levels to access to the game. For example, a user has to touch the play button and has to select the play mode in order to access to the game. They suggested that the play button is not necessary.

For the first user experience problem, we have solve them by redesigning the buttons with less colors and making different between touchable and untouchable areas. For the second problem, after a discussion among the group members, we have concluded that keeping the play button in the main menu screen is necessary in order to follow the standard game menu which has a play button.

According to the problems above, we spent a lot of time to find the best solutions for fixing them, especially on the serious problems about the game server and cloud database. Furthermore, we redesigned and re-implemented the system that the stakeholders need to change. These are causes affecting our project to d not follow the plan.

Now the implementation of game server is completed. This game server can handle several clients and rooms at the same time. We use PlayFab API for implementing several modules shown as follows:

- User account and authentication system: completed
- Virtual currency system: completed
- Store and inventory: mostly completed
- Leaders board and user statistic: mostly completed

However this project is currently in the implementing phased. So, this project does not have any contribute to promote the ALICE Connex project until the launch phase.

From the progress above, the stakeholders are satisfied with the progress. However, they need to change the wind module to an electric charge interaction module. This module is used for controlling an interaction between particles when the particle is close enough or collide with each other. This changing might be considered out of our project scope but they really want it. Furthermore some modules such as GUI, characters, and skill effects have to be modified to be more user friendly and attractive. After all the modules have been done, the next process is to deploy to the real server for playing and to get feedbacks back in order to improve our game.

Chapter 5

Conclusion

5.1.1 Completion table

No.	Operation	Status
1	Related research review	Completed
2	Project Proposal Submission	Completed
3	Necessary program, library installation and software environment setting	Completed
4	Software Design I -Interface Design	Completed
5	Software Design II -Back-end System Design	Completed
6	Implementation -Character	Completed
7	Implementation -Weapon and skill	Completed
8	Implementation -Map	Completed
9	Implementation -Back-end system and logical process such as ranking, damaging, items dropping etc...)	Completed
10	Game server and database server implementation	Completed
11	First draft report submission	Completed
12	Presentations I	Completed
13	Implementation -Transaction module	Completed
14	Implementation - User account module	Completed
15	Program testing & live run	Partially Completed
16	Game distribution on Google Play store	Preparing
17	Final report submission	Completed

During developing our project, we have encountered several problems as follows:

1. The problem of the scope and design for our game causes the delay in our project schedule. The design has been changed often due to a lack of communication among students, professors, and related persons. Thus, we have changed our schedule and started working on other components which are database and network server. However, after the scope is specified, we can continue working on the design and developing our game.

2. The problem occurs when we were using Unity to develop our game. Since we are new to Unity, we do not know how to use the provided functions efficiently. For example, Unity separates the UI layer and canvas layer allowing developers to separate between buttons which are placed in the UI layer and objects which are placed in the canvas layer. However, the UI layer is always on the top of the screen which causes the problem when we would like to display the effect on top of the object on the UI layer. To solve this problem, we have created an empty UI object and placed it on top of the previous UI layer. Then, this new object can display the effect.

3. The problem is about bad network connection that can result in some unexpected events in the game. For example, due to the bad network connection, a room header of the game may not be able to finish creating a room with in a short period of time before other players can see and join the room. This will make the game crash. To prevent this problem, we have overridden some classes of the game service to detect this kind of time overlapping problem.

4. The problem of lacking of knowledge about particle physics causes the delay in our game development. To solve this problem, we have invited an expert to give us some good advices.

5. The problem of using Facebook API is one of the key problems found during our project development since a user can register to the game by using her Facebook account. The first problem of using Facebook API is about key hash. Key hash is a set of numbers which is required for an application to connect with Facebook. Since at first we used a user-generated key hash which is recommended by many other developers but the errors of that Facebook could not find the key hash always came up. At the end, we were noticed that the key hash provided in the error message is not the same as the one that we used. Then, we tried that key hash appeared in the error message and it could work properly. The second problem is about Facebook user verification. At first we used a User ID which is generated from Facebook API to verify the user. However, it always said that the user is not found. So, instead of using User ID from Facebook API, we changed to User Access Token. Then the problem was resolved.

6. The problem of building an APK file of the game for running on Android devices had occurred during the testing phase. The cause of the building failure is from our wrong configuration of the AndroidManifest file which the API level required is not compatible with the one we have used. The other cause is from some methods in Playfab were duplicated which leads to the error of "unable to convert classes to DEX format". To solve this problem, some duplicated JAR files were deleted from our project. Others problems and solutions have already been discussed in Chapter 4.

The first thing we have learnt during the making of this project is to be flexible. Since this project is belonging to a real world stakeholder, we need to meet our customer in order to discuss about requirements and project information. Sometimes, we have to deal with inconstant requirements while we must keep on track in order to submit our work on time. This situation has changed our way that we used to work when we do the project with our friends. When we work with our friends, we can talk within our group and we do not change the scope of the project so much. However, when we work with other persons, we have to create our plan carefully, make a spare time for the delay which may occur, and try to balance our life in order to be not too tired which may cause our health problem. This situation has taught us how to handle stress from work, how to think carefully, how to analyse the situation, and how to communicate to others mannerly and properly.

The second thing is about working life and how to deal with problems. During the making of our project, we have encountered a lot of problems that we cannot solve them at the first time. We cannot find their solutions for some of them. We have done so by trial and error many times which makes us bored and tired. However, most of problems have been solved on time. This proves that a problem always has its solution and this helps us deal with a bigger problem which we might encounter in the future.

The main purpose of this project is to link with the ALICE's volunteer computing platform in order to exchange the related data. We have provided a module for communication between our game and the volunteer computing server after the volunteer computing platform is completed.

Since our game is the first version, there are only the necessary modules and completed backend systems with the basic UI screens. Others modules can be added later to make it more attractive. In the next version, developers can add maps of other planets and environments such as solar wind, electric field, and other environments that do not exist on the earth into the game. More game characters, weapons, armors, and particles can be added, deleted, and modified easily via a dashboard. Moreover, since Facebook account has been used in our game. We can add such as a social system and game invitation system to the game. This project is not available for download yet. It will be presented to CERN in July this year.

Reference

- [1] *ALICE Collaboration*, 2008, “*The ALICE experiment at the CERN LHC*,” 2008 JINST 3 S08002.
- [2] *D. Duellmann*, 2014, *Big Data and Storage Management at the Large Hadron Collider* [Online], Available:
<http://openlab.web.cern.ch/sites/openlab.web.cern.ch/files/presentations/dirkdfuturegovs14.pdf>
 [2015, March 5]
- [3] *K. Jones*, 2014, *The 2014 Mobile Landscape: 25 Statistics That Will Drive The Future of Mobile Marketing* [Infographic] [Online], Available:
<http://www.searchenginejournal.com/2014-mobile-landscape-25-statistics-willdrive-future-mobile-marketing-infographic/89507> [2015, March 5]
- [4] *M. Arslan*, 2012, “*Computing While Charging: Building a Distributed Computing Infrastructure Using Smartphones*,” *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pp. 193- 204.
- [5] *Timeline.web.cern.ch*, (2015). *The history of CERN | CERN timelines*. [online] Available at:
<http://timeline.web.cern.ch/timelines/The-history-of-CERN> [Accessed 9 Nov. 2015].
- [6] *Wikipedia*, (2015). *ALICE: A Large Ion Collider Experiment*. [online] Available at:
https://en.wikipedia.org/wiki/ALICE:_A_Large_Ion_Collider_Experiment [Accessed 10 Nov. 2015].
- [7] *Wikipedia*, (2015). *Quark gluon plasma*. [online] Available at:
https://en.wikipedia.org/wiki/Quark%E2%80%93gluon_plasma [Accessed 10 Nov. 2015].
- [8] *Wikipedia*, (2015). *Quark*. [online] Available at: <https://en.wikipedia.org/wiki/Quark>
 [Accessed 10 Nov. 2015].
- [9] *Wikipedia*, (2015). *Antiparticle*. [online] Available at:
<https://en.wikipedia.org/wiki/Antiparticle> [Accessed 10 Nov. 2015].
- [10] *Wikipedia*, (2015). *Gluon*. [online] Available at: <https://en.wikipedia.org/wiki/Gluon>
 [Accessed 10 Nov. 2015].
- [11] *Wikipedia*, (2015). *Hadron*. [online] Available at: <https://en.wikipedia.org/wiki/Hadron>
 [Accessed 10 Nov. 2015].

- [12] *Wikipedia*, (2015). *Neutrino*. [online] Available at: <https://en.wikipedia.org/wiki/Neutrino> [Accessed 10 Nov. 2015].
- [13] *Wikipedia*, (2015). *Meson*. [online] Available at: <https://en.wikipedia.org/wiki/Meson> [Accessed 10 Nov. 2015].
- [14] *Wikipedia*, (2015). *List of Unity games*. [online] Available at: https://en.wikipedia.org/wiki/List_of_Unity_games [Accessed 9 Nov. 2015].
- [15] *Unity3d.com*, (2015). *Unity - Get Unity*. [online] Available at: <https://unity3d.com/get-unity> [Accessed 9 Nov. 2015].
- [16] *Photonengine.com*, (2015). *Photon Turnbased Pricing Plans / Photon: Multiplayer Made Simple*. [online] Available at: <https://www.photonengine.com/en/Turnbased/Pricing> [Accessed 9 Nov. 2015].
- [17] *Doc.photonengine.com*, (2015). *Photon Server Intro / Exit Games*. [online] Available at: <https://doc.photonengine.com/en/onpremise/current/getting-started/photon-server-intro> [Accessed 9 Nov. 2015].
- [18] *Playfab*, (2015). *Features*. [online] Available at: <https://playfab.com/features/> [Accessed 9 Nov. 2015].
- [19] *Playfab*, (2015). *Pricing*. [online] Available at: <https://playfab.com/pricing/> [Accessed 9 Nov. 2015].
- [20] *Parmar, C.* (2004). *Understanding .NET Framework at a glance - CodeProject*. [online] *Codeproject.com*. Available at: <http://www.codeproject.com/Articles/7333/Understanding-NET-Framework-at-a-glance> [Accessed 11 Nov. 2015].
- [21] *Wikipedia*, (2015). *Software development kit*. [online] Available at: https://en.wikipedia.org/wiki/Software_development_kit [Accessed 9 Nov. 2015].
- [22] *Facebook Developers*, (2015). *Facebook SDK for Unity - Unity SDK - Documentation - Facebook for Developers*. [online] Available at: <https://developers.facebook.com/docs/unity> [Accessed 11 Nov. 2015].