

ABSTRACT

We are developing a Space Shooter game. Space Shooter games is a shoot'em up (shmup or STG) and an action pack "bullet hell" game where your mission is to shoot the enemy ships and win battles against big bad bosses and travel through the galaxy. Inspired by Raiden, the old school Japanese shooters, there are tons of unique enemy ships and bosses, each equipped with specialized weaponry and ability. While the game is easy to handle, you need more than fast reflexes to master the game. To survive in each level, we need to recognize attack patterns and use different power ups at appropriate time against the correct enemy. Upgrade your ships with several levels of lasers, spread bullets, homing missiles, torpedoes, speed power and powerful bombs. The power ups stay with you until you die and you will accumulate more power as you collect the same power up. The graphic is an exhilarating visual treat, with cool music and sound effects providing intense experience for hardcore gamers and casual gamers alike.

You have think what each power up does, how to gain extra live for free and how to get full power- ups without having to wait for it to appear.

ACKNOWLEDGEMENTS

First of all, we want to thank our Allah Almighty and then we are thankful to our parents who supported me in this whole study and always pray for our success and good health.

We would like to express our sincere gratitude and profound indebtedness to my supervisor Shamim Ahmed (Assistant Professor) for his guidance, valuable suggestions, commendable support and endless patience towards the completion of this project. We feel very proud to have worked with him. Without the inspiring enthusiasm and encouragement of our supervisor, this work could not have been completed.

We thank all the staffs and graduate students at Bangladesh University of Business and Technology (BUBT) and all the friends for their support and encouragement. We would also like to extend our elder and younger brothers.

Finally, I wish to express my gratitude to Bangladesh University of Business and Technology (BUBT) for providing an excellent environment for research and all the other facilities to complete the project successfully.

With Best Regards,

Md. Azizur Rahman

Mst. Moarium Akter

Md. Nazmul Hoque

DECLARATION

We declare that this project is our own work and been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published work of others has been acknowledged in the text and a list of reference is given.

Signature

Signature

Signature

Md. Azizur Rahman

Mst. Moarioum Akter

Md. Nazmul Hoque

CERTIFICATE

TO WHOM IT MAY CONCERN

This is to certify that, **Md. Azizur Rahman, Mst. Moarioum Akter and Md. Nazmul Hoque** of B.Sc. in CSE has completed their project work titled “**Space-Shooter Game**” in partial fulfillment for the requirement of B.Sc. in CSE. Bangladesh University of Business and Technology in the year 2018.

(Project Supervisor)

Shamim Ahmed

Assistant Professor

Department of Computer Science and Engineering

Bangladesh University of Business and Technology

Mirpur-2, Dhaka-1216, Bangladesh.

DEDICATION

Dedicated to our parents for all their love and inspiration.

APPROVAL

The project Designing and Implementation of Online based Website report is submitted by Md. Azizur Rahman ID-14153203054, Mst. Moarioum Akter ID-14153203059 and Md. Nazmul Hoque ID-14153203061 Department of Computer Science and Engineering, Bangladesh University of Business and Technology under the supervision of Shamim Ahmed, Assistant Professor, Department of Computer Science and Engineering has been accepted as satisfactory for the partial fulfillment of the requirements for degree of Bachelor of Science (B.Sc. Engineering) in Computer Science and Engineering.

Supervisor: -----

Shamim Ahmed
Assistant Professor
Department of Computer Science and
Engineering Bangladesh University of
Business and Technology Mirpur-2,
Dhaka-1216, Bangladesh.

Chairman: -----

Dr. M. Ameer Ali
Professor
Department of Computer Science and
Engineering Bangladesh University of
Business and Technology
Mirpur-2, Dhaka-1216, Bangladesh

COPYRIGHT

© Copyright by Md. Azizur Rahman(14153203054), Mst. Moarioum Akter (14153203059), and Md. Md. Nazmul Hoque (14153203062)

All Right Reserved

ABBREVIATIONS

List of abbreviations	Description
SSG	Space Shooter Game.
ER	Entity Relationship
DFD	Dataflow Diagram
DBMS	Database Management System
SDLC	Software Development Life Cycle
GUI	Graphical User Interface
RDBMS	Relational Database Management System
ERD	Entity Relationship Diagram
SCD	System Context Diagram
IDE	Integrated Development Environment
CLR	Common Language Runtime

Table of Content

	Page No
Abstract	i
Acknowledgements.....	ii
Declaration	iii
Certificate	iv
Dedication	v
Approval.....	vi
Copyright.....	vii
Abbreviations.....	viii
 Chapter- 1 Introduction	1-8
1.1 Introduction	1
1.2 Existing System	2
1.2.1 Existing System of Space Invaders	2
1.2.2 Existing System of Eve Online	2
1.2.3 Existing System Free Space 2	3
1.2.4 Existing System of Elite Dangerous.....	3
1.3 Problems of Existing System	3
1.4 Motivation	4
1.5 Objectives.....	5
1.6 Contributions	5
1.7 Organization of Project Report.....	6
1.8 Games Application in Bangladesh.....	6
1.9 Benefits of the old games Application.....	7
1.10 Limitations of the old games Application	7
1.11 Conclusions	8

Chapter- 2 Existing System and Literature Review	9-41
2.1 Introduction	9
2.2 Existing System	10
2.2.1 Space Invaders	10
2.2.2 Eve Online	14
2.2.3 FreeSpace 2	17
2.2.4 Elite Dangerous.....	20
2.3 Problem Statement.....	22
2.3.1 Graphics problem	22
2.3.2 Project estimation.....	22
2.3.3 Preproduction phase	22
2.3.4 Prototyping.....	23
2.3.5 Getting tired during the project	23
2.3.6 Polish	23
2.3.7 Communication.....	24
2.3.8 Advertising.....	24
2.4 Supporting Literature	24
2.4.1 Blueprint Language	24
2.4.2 How Do Blueprints Work?	26
2.4.3 Commonly Used Blueprint Types.....	26
2.4.4 Level Blueprint	26
2.4.5 Blueprint Class	27
2.4.6 Unreal Games Engine 4	27
2.4.7 Adobe Photoshop	29
2.4.8 Blender.....	29
2.4.9 Entity Relationship Diagram (ERD)	30
2.4.10 General Overview of ERD.....	31
2.4.11 Data Flow Diagram	35
2.4.12 Use Case Diagram	38
2.4.13 Relational Database (RDB).....	40
2.5 Conclusions	41

Chapter- 3 Proposed Solution/ Methodology	45-63
3.1 Introduction	45
3.2 Feasibility Study	45
3.2.1 Technical Feasibility	46
3.2.2 Economical Feasibility	47
3.2.3 Operational Feasibility	47
3.2.4 Behavioral Feasibility.....	48
3.2.5 Legal Feasibility	48
3.3 Key Features of Proposed System	48
3.3.1 Functional Requirements of the Proposed System.....	49
3.3.2 Non-Functional Requirements of the Proposed System.....	50
3.3.3 Advantages of Proposed System.....	51
3.4 System Design	52
3.4.1 Entity Relationship Model	52
3.4.2 Data Flow Diagram	53
3.5 Database Design	55
3.6 Implementation.....	57
3.6.1 Front End	57
3.6.2 Back End.....	59
3.7 Results Analysis	60
3.8 Conclusions	63
Chapter- 4 User Manual	64-98
4.1 Introduction	64
4.2 Testing.....	64
4.3 Testing Methods	65
4.3.1 Models of Testing.....	65
4.4 Test Cases.....	66
4.5 Requirement Analysis	68
4.5.1 Non-Functional Requirement.....	69
4.5.2 Functional Requirement	70
4.5.3 Software Requirement	70
4.5.4 Hardware Requirement.....	71
4.6 Result Analysis	71
4.6.1 Flow Chart	72

4.7 Graphical Interface	73
4.7.1 Level One.....	73
4.7.2 Level Two	79
4.7.3 Level Three	84
4.7.4 Level Four.....	89
4.8 Application	96
4.9 Conclusions	98

Chapter- 5 Conclusion and Future Work

99-101

5.1 Conclusions	99
5.2 Limitation.....	100
5.3 Future Scope.....	101

List of Figures

xiii-xiv

List of Tables

xv

References

102-104

List of Figures

	Page No
Figure 2-1: Space Invaders Game	11
Figure 2-2: Eve Online Game	15
Figure 2-3: FreeSpace 2 Game.....	18
Figure 2-4: Elite Dangerous Game.....	20
Figure 2-5: Level Blueprint	26
Figure 2-6: Blueprint Class.....	27
Figure 2-7: Unreal Game Engine4	28
Figure 2-8: Adobe Photoshop CS6.....	29
Figure 2-9: Blender	30
Figure 2-10: Entity	31
Figure 2-11: Weak Entity	32
Figure 2-12: Attribute.....	32
Figure 2-13: Multi Valued Attribute	33
Figure 2-14: One-to-One	33
Figure 2-15: One-to-Many	34
Figure 2-16: Many-to-One.....	34
Figure 2-17: Many-to-Many	34
Figure 2-18: DFD Component	36
Figure 2-19: 0-Level Data Flow Diagram of Customer Service System.....	37
Figure 2-20: Level-1 Data Flow Diagram of Customer Service System.....	37
Figure 2-21: Rectangle	39
Figure 2-22: Use Case	39
Figure 2-23: Actors	40
Figure 3-1: ER-Diagram for the Project	53
Figure 3-2: DFD Space Ship Games	55
Figure 4-1: Game Starting Menu	73
Figure 4-2: level-1 Enemy ships First Step.....	74
Figure 4-3: level-1 Enemy ships Second Step	75
Figure 4-4: Enemy ships Third Step.....	76
Figure 4-5: level-1 Enemy ships fourth Step	77
Figure 4-6: level-1 Enemy ships Boss	78
Figure 4-7: level-2 Enemy ships first Step	79
Figure 4-8: level-2 Enemy ships Second Step	80
Figure 4-9: level-2 Enemy ships Third Step	81
Figure 4-10: level-2 Enemy ships fourth Step	82
Figure 4-11: level-2 Enemy ships Boss	83
Figure 4-12: level-3 Enemy ships first Step.....	84
Figure 4-13: level-3 Enemy ships Second Step	85
Figure 4-14: level-3 Enemy ships Third Step	86
Figure 4-15: level-3 Enemy ships Fourth Step	87
Figure 4-16: level-3 Enemy ships Boss	88
Figure 4-17: level-4 Enemy ships first Step.....	89
Figure 4-18: level-4 Enemy ships Second Step	90
Figure 4-19: level-4 Enemy ships Third Step	91

Figure 4-20: level-4 Enemy ships Fourth Step	92
Figure 4-21: level-4 Enemy ships Boss	93
Figure 4-22: Blue Cube Rocket Power Up	94
Figure 4-23: Coin	95
Figure 4-24: Red Cube Bullet Power Up.....	96
Figure 4-25: Mobile games apps revenue	97

List of Tables

	Page No
Table 3-1: Score Save.....	56
Table 3-2: Coin Save	56
Table 3-3: Player	59
Table 3-4: Health.....	59
Table 3-5: Enemy1, Enemy2, Enemy Boss.....	59
Table 3-6: Coin.....	60
Table 3-7: Score	60
Table 4-4-1: Animation Testing	67
Table 4-4-2: Interaction between objects.....	67
Table 4-4-3: UI Testing	68
Table 4-4-4: Health and Coin.....	68

Chapter- 1

Introduction

1.1 Introduction

The game must allow the player to play the game, save and load the progress at any time, have score system to rate player performance. The game will be divided into stages. The Player can roam in an Open World when not on a mission. The player controls character movements over obstacles, defeat enemies, reaching end goal to finish one stage. Player character will loss a life or reduces its shields when collided with enemies or lethal obstacles[1].

In our project we selected Unreal Game Engine version 4 for development of the project. Unreal Game Engine 4 is a powerful cross platform 3D engine and it is user friendly development environment. Unreal Game Engine 4 is a easy to understand so anybody who want to easily create 3D games and application for mobile, laptop, computer, web etc. create 3D games and applications for mobile, desktop, the web and consoles[2].

The Entertainment Software Association (ESA) is the major trade association for the computer and video games industry. Each year, the association conducts a research survey which is the most comprehensive report of its kind. The 2016 report can be downloaded here. Some of the key findings of the 2016 report are:

63% of households in the U.S. have at least one person who regularly plays video games (defined as playing 3 or more hours per week) and 65% of households have at least one type of game playing device with 48% owning a console specifically for gaming.

Gaming is not just an activity exclusively for young boys. The average age of a gamer is 35 and the gender split is 59% male and 41% female. The average age of a female gamer is 44.

Game players span the age group spectrum with:

27% under the age of 18

29% between 18 and 35 years of age

18% between 36 and 49 years of age

26% are 50 or older.

The ESA's report also presents a picture of a group that is more socially engaged than the stereotype of gamers playing alone. 54% of gamers play with others and 51% of dedicated gamers play some type of multi-player mode at least once a week. Among the gamers surveyed, 53% said that the activity of gaming helped them to connect more with friends and 42% said the same for family.

1.2 Existing System

Existing System of Space Invaders is a 1978 arcade game created by Tomohiro Nishikado. It was manufactured and sold by Taito in Japan, and licensed in the United States by the Midway division of Bally. Within the shooter genre, Space Invaders was the first fixed shooter and set the template for the shoot 'em up genre. The goal is to defeat wave after wave of descending aliens with a horizontally moving laser to earn as many points as possible.

1.2.1 Existing System of Space Invaders

Space Invaders was an immediate commercial success; by 1982, it had grossed \$3.8 billion, with a net profit of \$450 million, making it the best-selling video game and highest-grossing "entertainment product" at the time. Adjusted for inflation, the many versions of the game are estimated to have grossed over \$13 billion in total revenue as of 2016, making it the highest-grossing video game of all time[2].

1.2.2 Existing System of Eve Online

Existing System of Eve Online (stylised EVE Online) is a space-based, persistent world massively multiplayer online role-playing game (MMORPG) developed and published by CCP Games[18]. Players of Eve Online can participate in a number of in-game professions and activities, including mining, piracy, manufacturing, trading, exploration, and combat (both player versus environment and player versus player). The game contains a total of 7,800 star systems that can be visited by players.

1.2.3 Existing System Free Space 2

Existing System Free Space 2 is a 1999 space combat simulation computer game developed by Volition as the sequel to Descent: Free Space – The Great War. It was completed ahead of schedule in less than a year, and released to very positive reviews. However, the game became a commercial failure, and was described by certain critics as one of 1999's most unfairly overlooked titles[2].

1.2.4 Existing System of Elite Dangerous

Existing System of Elite Dangerous is a space-flight simulation game developed and published by Frontier Developments. Piloting a spaceship, the player explores a realistic 1:1 scale open-world representation of the Milky Way galaxy, with the gameplay being open-ended. The game is the first in the series to attempt to feature massively multiplayer gameplay, with players' actions affecting the narrative story of the game's persistent universe, while also retaining single-player options. Elite Dangerous is the fourth game in the Elite video game series. It is the sequel to Frontier: First Encounters, released in 1995[5].

1.3 Problems of Existing System

I hear this argument a lot in terms of video game music and quality. And some of these same people, are also the ones who say that because the limitations are now gone, game music and game quality has gone down. However, I disagree. Yes, it's amazing to see the kind of music that came out of the SNES and Sega Genesis when it was pushed to its limits, but I would like to look at things from a different perspective[12]. If an artist is truly talented, than the quality of their work should not, and will not go down just because the limitations are off

The problems of the existing manual system of voting include among others the following:

No online or multiplayer

Not portable due to it being a large machine.

Couldn't save game progress on the arcade machine.

The games were limited to one game per machine e.g. Mario or Pac-Man.

Old hardware is still used in some arcade machines which made them unreliable and more prone to crash.

1.4 Motivation

In the current space shooter games. Movement in Mission Movements in mission is designed in such a way that confine the player to face forward towards the stage progression and control the space required to design the mission towards sure achievement if the player goes through all the enemies.

Movement in Open World Movement in Open World is given that user can travel in any direction in a 3D plane.

The enemies will be coded with basic artificial intelligence that can move towards enemies, some can ambiguity the attacks of the player and change their course midway on the playing area. The enemy can decrease player's health by either shooting at him or colliding with him. In either case the player's collider mesh will be affected by some other colliders. Electronic games are now an everyday part of childhood and adolescence. The debate has moved from whether children should play video games to how to maximize potential benefits and to identify and minimize potential harms. To do this, we must understand what motivates children to play electronic games and what needs the games meet. Drawing on a survey of 1,254 middle school children, focus groups with boys and their parents, and findings from other quantitative and qualitative research, the author describes a variety of motivations for video game play (including games with violent content) and how these may vary based on factors such as mood, environment, personality, and developmental stage. The findings are put into the context of normal development, and suggestions are given for parents, educators, and researchers[18].

1.5 Objectives

The game is developed for full-time entertainment and enthusiasms. It teaches the Gamer to be alert at every situation he/she faces, because if the Gamer is not fully alert and notice the saucer fire he/she must be hit by the enemy ships.

Though the proposed game is an action game, it doesn't involve direct violence. No zombie killing, animal killing, or human killing is performed in the game. So it can also be viewed as a nonviolence game. Kids can also play this game, because the design of the game is very simple[12]. controlling the game is very easy – pressing some neighboring keys of the keyboard.

The following are the objectives of this project:

1. Reviewing the existing/current games or approach in Bangladesh.
2. Coming up with an automated games application in Bangladesh.
3. Implementing an automated/games application.
4. Validating the system to ensure that only legible user are allowed to play.
5. To create desktop-based “online games application”.
6. To increase the efficiency and intelligence of user.
7. To integrate of all records of users’ score.
8. The system should be flexible enough. To provide a strong security system.
9. It provides “better and efficient” service to the user.

1.6 Contributions

We created games so that all developers can distribute their applications to users on an open platform. One of the best ways we can help games is to write cool system that user's love this games provides the information and tools we need to developed applications for compatible desktop and mobile devices[12].

- Attractive background
- Spaceship
- Different asteroid types
- Different enemy types
- Beautiful graphics

- Live score
- Background music

1.7 Organization of Project Report

In chapter 2, we describe the existing system and give the literature review on games. In existing system we will also discuss about the many types of existing games and functionalities of the existing games. In supporting literature we will describe about the all types of tools that we have used in our system. From analysis of existing system we will come to know proses and cons of System.

Chapter 3 gives us information about the system analysis and methodologies. In this chapter we discussed about software engineering methodologies and analysis of whole phase. In the proposed model chapter we discussed about the ERD, DFD, Use Case, database design and system design with details[13].

Chapter 4, here we discussed testing system and analysis result. We have discussed about the type of testing is needed to develop any games. We have also mentioned all our test cases.

Chapter 5 lets us discuss about the requirements like hardware and software and project overview interface with which user can understand. The concluding remarks are detailed in last chapter.

1.8 Games Application in Bangladesh

- The Mascoteers are a Bangladeshi game developer hub known for the worldwide arcade games 360 Degree, Swiperoo, Dropple and many more. They are a pioneer of Apple Watch Games, Apple TV Games and VR Gear games in Bangladesh.[15]
- Team 71 are the developers of the Liberation 71 game.
- Rise Up Labs or RUL is a successful mobile game developing company who created the worldwide famous mobile game Tap Tap Ants. They also created

some other Android and iOS games. Rooftop Frenzy and Highway Chase are two mobile games by RUL.

- Massive Star Studio (MSS) is the developer of the first open world racing game of Bangladesh Hatirjheel: Dream Begins.
- Dream World Studio (DWS) is a new developing team of young developers and the first multinational developing team in Bangladesh. The Hound6 series, Durjoy, and Crimact are the creations of the team.
- 143Play.com is a gaming and entertainment company which works with popular card games and MMORPGs along with augmented reality. This gaming house also provides gamification aspects in various domains.[16]
- GHOST Interactive is currently working on WAR71 and IGI: GHOST Fighter (Project IGI-3).
- Mindfisher Games INC: Developed Heroes 71 Game series.[17]

1.9 Benefits of the old games Application

- Multiple uses. A computer may cost more than a console, but by buying a good PC you kill multiple birds with one stone. ...
- Updating your gaming platform. ...
- Better graphics. ...
- Saving physical space. ...
- The game library is huge. ...
- Easy access to the history of video games. ...
- Increased control flexibility. ...
- Gaming PCs can be very expensive

1.10 Limitations of the old games Application

- The Can be complicated
- pricing of components
- Get outdated quicker
- The hardware limits the size and graphics of the game
- 3D Games are hard to run on the devices
- Not very portable

1.11 Conclusions

Even though the game might not the standards of many commercial games, given the resources and time frame. The game is easy to play and the visual effects make the game look graphically good. Each effect contributes to the appearance of the game. The game is easy to play and the visual effects make the game look best. Each effect donates to the look of the game own laptop. There are a lot of positive aspects to working together with team. When problems occur everyone can help, ideas can be discussed and you get to know each other better, making it more motivating and fun with members to work on a project. In hindsight, we should probably have utilized the rooms with computers provided by college in order to increase productivity. Video games are a form of media that is often associated with negative health consequences. However, when games are played in moderation and with mindfulness, they are a viable source of stress relief as well as a catalyst for mental health improvement and development of social skills.

Chapter- 2

Existing System and Literature Review

2.1 Introduction

Video gaming is a firmly established leisure pursuit, which continues to grow in popularity. This paper is an examination of what motivates people to play computer games, and the relevance of such factors to the positive and negative aspects of computer gaming. When all of an individual's motivations to play video games are for the pursuit of 'fun', it is said that an intrinsic motivation is the most prevalent motivation. However, the primary motivation for playing video games among periodic gamers is different from the primary motivation of regular gamers: periodic gamers are driven by extrinsic motivation, whereas regular gamers are driven by intrinsic motivation. The pursuit of a challenge is the prevalent motivation reported by regular gamers of both genders.

The Theory of Flow Experience, and the Attribution Theory have contributed to the understanding of why games may provide a safe medium, in which to learn about the consequences of actions through experience. Computer games may facilitate the development of self-monitoring and coping mechanisms. If the avoidance or escape from other activities is the primary motivation for playing video games, there tends to be an increased risk of engaging in addiction-related behaviors[12]. This paper reports on the findings of previous research (into the motivations for playing computer games), and on industry reports containing data relating to gamer motivations. The aim is to build a picture of what motivates people to play computer games, and how motivation is associated with the main positive and negative aspects of computer gaming.

2.2 Existing System

The related works and the theory/literature are detailed below.

2.2.1 Space Invaders

Space Invaders is a 1978 arcade game created by Tomohiro Nishikado. It was manufactured and sold by Taito in Japan, and licensed in the United States by the Midway division of Bally. Within the shooter genre, Space Invaders was the first fixed shooter and set the template for the shoot 'em up genre. The goal is to defeat wave after wave of descending aliens with a horizontally moving laser to earn as many points as possible.

Space Invaders was an immediate commercial success; by 1982, it had grossed \$3.8 billion, with a net profit of \$450 million, making it the best-selling video game and highest-grossing "entertainment product" at the time. Adjusted for inflation, the many versions of the game are estimated to have grossed over \$13 billion in total revenue as of 2016 making it the highest-grossing video game of all time[7].

Space Invaders is considered one of the most influential video games of all time. It helped expand the video game industry from a novelty to a global industry, and ushered in the golden age of arcade video games. It was the inspiration for numerous video games and game designers across different genres, and has been ported and re-released in various forms. The 1980 Atari VCS version quadrupled sales of the VCS, thereby becoming the first killer app for video game consoles. More broadly, the pixelated enemy alien has become a pop culture icon, often representing video games as a whole.

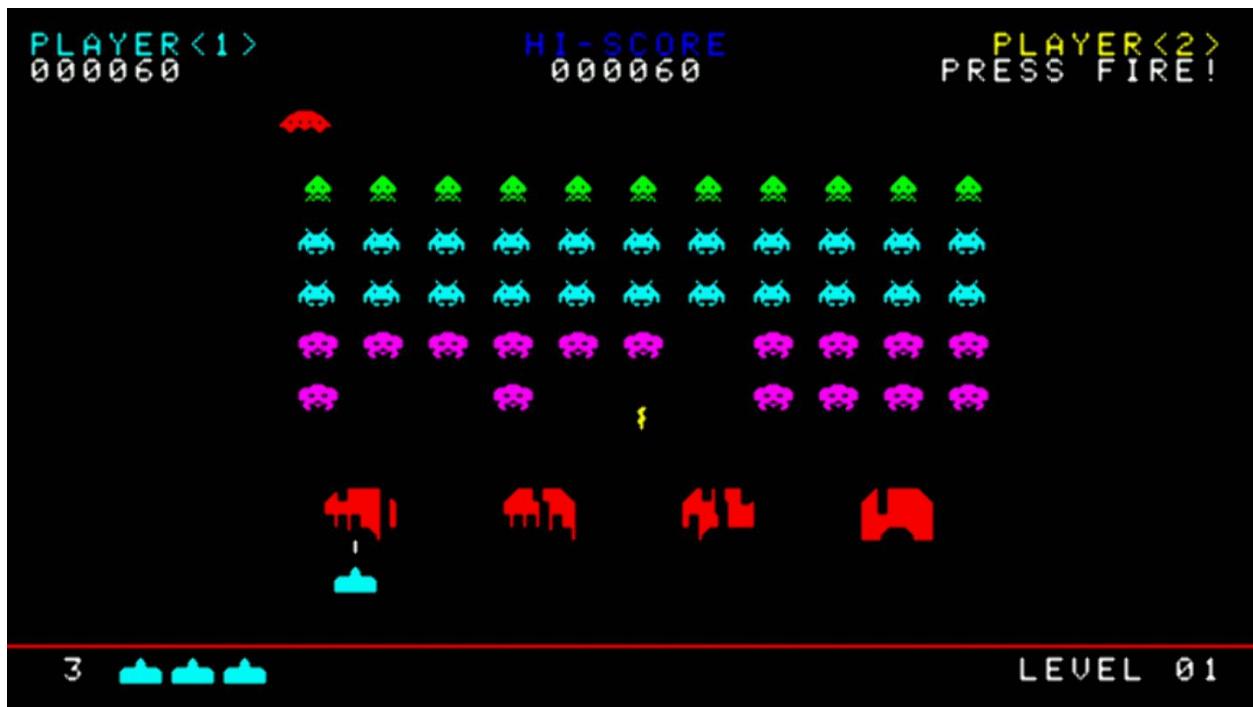


Figure 2-1: Space Invaders Game

The player-controlled laser cannon shoots the aliens as they descend toward the bottom of the screen.

Development: Space Invaders was created by Japanese designer Tomohiro Nishikado, who spent a year designing the game and developing the necessary hardware to produce it. The game's inspiration is reported to have come from varying sources, including an adaptation of the mechanical game Space Monsters released by Taito in 1972, and a dream about Japanese school children who are waiting for Santa Claus when they are attacked by invading aliens. Nishikado himself has cited Atari's arcade game Breakout as his inspiration. He aimed to create a shooting game that featured the same sense of achievement from completing stages and destroying targets, but with more complex graphics. The game uses a similar layout to that of Breakout but has altered game mechanics. Rather than bounce a ball to attack static objects, players are given the ability to fire projectiles at moving enemies. Early enemy designs for the game included tanks, combat planes, and battleships. Nishikado, however, was not satisfied with the enemy movements; technical limitations made it difficult to simulate flying. Humans would have been easier to simulate, but the designer considered shooting them immoral. After the release of the 1974 anime Space Battleship Yamato in Japan, and seeing a magazine feature about Star Wars, he thought of using a space theme. Nishikado drew inspiration for the aliens from a novel by H. G. Wells, The War of the Worlds, and created initial bitmap images after the octopus-

like aliens. Other alien designs were modeled after squids and crabs. The game was originally titled Space Monsters after a popular song in Japan at the time, "Monster", but was changed to Space Invaders by the designer's superiors.

2.2.1.1 Hardware

Because microcomputers in Japan were not powerful enough at the time to perform the complex tasks involved in designing and programming Space Invaders, Nishikado had to design his own custom hardware and development tools for the game. He created the arcade board using the latest microprocessors from the United States. The game uses an Intel 8080 central processing unit (CPU), displays raster graphics on a CRT monitor, and uses monaural sound hosted by a combination of analog circuitry and a Texas Instruments SN76477 sound chip. The adoption of a microprocessor was inspired by Gun Fight (1975), Midway's microprocessor adaptation of Nishikado's earlier discrete logic game Western Gun, after the designer was impressed by the improved graphics and smoother animation of Midway's version. Despite the specially developed hardware, Nishikado was unable to program the game as he wanted—the Control Program board was not powerful enough to display the graphics in color or move the enemies faster—and he ended up considering the development of the game's hardware the most difficult part of the whole process. While programming the game, Nishikado discovered that the processor was able to render the alien graphics faster when there were fewer on the screen. Rather than design in compensation for the speed increase, he decided to keep it as a challenging gameplay mechanism [11][15].

Space Invaders was first released in a cocktail-table format arcade cabinet with black-and-white graphics, while Midway released the Western version in an upright cabinet; it used strips of orange and green cellophane over the screen to simulate color graphics. The graphics are reflected onto a painted backdrop of a moon against a starry background. Later Japanese releases used rainbow-colored cellophane, such as T.T. Space Invaders in 1978, and were eventually followed by a version with a full-color display. The cabinet artwork features large humanoid monsters not present in the game. Nishikado attributes this to the artist basing the designs on the original title of "Space Monsters", rather than referring to the actual in-game graphics.

2.2.1.2 Music

Despite its simplicity, the music to Space Invaders was revolutionary for the gaming industry. Video game scholar Andrew Schartmann identifies three aspects of the music that had a significant impact on the development of game music:

Whereas video game music prior to Space Invaders was restricted to the extremities (i.e., a short introductory theme with game-over counterpart), the alien-inspired hit featured continuous music—the well-known four-note loop—throughout, uninterrupted by sound effects: "It was thus the first time that sound effects and music were superimposed to form a rich sonic landscape. Not only do players receive feedback related directly to their actions through sound effects; they also receive stimulus in a more subtle, non-interactive fashion through music[22].

The music interacts with on-screen animation to influence the emotions of the player: "That seemingly pedestrian four-note loop might stir us in the most primitive of ways, but that it stirs us at all is worthy of note. By demonstrating that game sound could be more than a simple tune to fill the silence, Space Invaders moved video game music closer to the realm of art[22].

The music for Space Invaders popularized the notion of variability—the idea that music can change in accordance with the ongoing on-screen narrative. The variable in Space Invaders, the tempo, is admittedly simple, but its implications are not to be underestimated. "Over the years, analogous strategies of variation would be applied to pitch, rhythm, dynamics, form, and a host of other parameters, all with the goal of accommodating the nonlinear aspect of video games[22]. At the deepest of conceptual levels, one would be hard-pressed to find an arcade game as influential to the early history of video game music as Space Invaders. Its role as a harbinger of the fundamental techniques that would come to shape the industry remains more or less unchallenged. And its blockbuster success ensured the adoption of those innovations by the industry at large.

— Andrew Schartmann, Maestro Mario: How Nintendo Transformed Videogame Music into an Art, Thought Catalog (2013)

2.2.1.3 Legacy

An urban legend states that *Space Invaders*' popularity led to a shortage of 100-yen coins in Japan[11][48][49]. In actuality, 100-yen coin production was lower in 1978 and 1979 than in previous or subsequent years[50][51]. Additionally, arcade operators would have emptied their machines and taken the money to the bank, thus keeping the coins in circulation.[51] Reports from those living in Japan at the time indicate "nothing out of the ordinary during the height of the *Space Invaders* invasion"[51].

As one of the earliest shooting games, *Space Invaders* set precedents and helped pave the way for future titles and for the shooting genre[52][53]. *Space Invaders* popularized a more interactive style of gameplay, with the enemies responding to the player-controlled cannon's movement[13] and was the first video game to popularize the concept of achieving a high score[3][48][52] being the first to save the player's score[52]. While earlier shooting games allowed the player to shoot at targets, *Space Invaders* was the first in which targets could fire back at the player[54]. It was also the first game where players were given multiple lives[55] had to repel hordes of enemies[16] could take cover from enemy fire, and use destructible barriers,[56] in addition to being the first game to use a continuous background soundtrack, with four simple diatonic descending bass notes repeating in a loop, which was dynamic and changed pace during stages like a heartbeat sound that increases pace as enemies approached[57] [58].

2.2.2 Eve Online

Eve Online (stylised EVE Online) is a space-based, persistent world massively multiplayer online role-playing game (MMORPG) developed and published by CCP Games. Players of Eve Online can participate in a number of in-game professions and activities, including mining, piracy, manufacturing, trading, exploration, and combat (both player versus environment and player versus player). The game contains a total of 7,800 star systems that can be visited by players[2][3].

The game is renowned for its scale and complexity with regards to player interactions – in its single, shared game world, players engage in unscripted economic competition, warfare, and political schemes with other players[4]. The Bloodbath of B-R5RB, a battle involving

thousands of players in a single star system, took 21 hours and was recognized as one of the largest and most expensive battles in gaming history[5]. Eve Online was exhibited at the Museum of Modern Art with a video including the historical events and accomplishments of the playerbase[6].

Eve Online was released in North America and Europe in May 2003. It was published from May to December 2003 by Simon & Schuster Interactive after which CCP purchased the rights and began to self-publish via a digital distribution scheme[7] [8]. On January 22, 2008, it was announced that Eve Online would be distributed via Steam[9]. On March 10, 2009, the game was again made available in boxed form in stores, released by Atari, Inc[10]. In February 2013, Eve Online reached over 500,000 subscribers[11]. On November 11, 2016, Eve Online added a limited free-to-play version[12].



Figure 2-2: Eve Online Game

2.2.2.1 Background

Set more than 21,000 years in the future, the background story of *Eve Online* explains that humanity, having used up most of Earth's resources through centuries of explosive population growth, began colonizing the rest of the Milky Way[13][14]. As on Earth, this expansion also led to competition and fighting over available resources, but everything changed with the discovery of a natural wormhole leading to an unexplored galaxy subsequently dubbed "New Eden." Dozens of colonies were founded, and a structure, a gate of sorts (which bears the inscription "EVE" on the New Eden side), was built to stabilize the wormhole that linked the colonies of New Eden with the rest of human civilization. However, when the wormhole unexpectedly collapsed, it destroyed the gate as well as the connection between the colonies of New Eden and the Milky Way. Cut off from the rest of humanity and supplies from Earth, the colonies of New Eden were left starving and disconnected from one another; many died out entirely. Over the millennia the descendants of the surviving colonists managed to rebuild their own societies, but by this time the memories and knowledge of humanity's origins, of Earth and the Milky Way galaxy, as well as the history of the settling of New Eden, was lost; what little information that survived transmission over the generations was misunderstood, lost in translation, and/or consigned to mythology. Five major distinct societies rose to prominence from the surviving colonies, each growing into interstellar spaceflight-capable civilizations. The states based around these societies make up the five major factions in *Eve Online*: the Amarr Empire, the Caldari State, the Gallente Federation, the Minmatar Republic and the Jove Directorate.

2.2.2.2 Gameplay

Players start the game by either selecting a previously created character or by creating a new one. Each Eve Online account allows for up to three characters[25]. When a player creates a new character, they start by choosing one of the four playable races – Amarr, Gallente, Minmatar, or Caldari. Each race is further divided into three bloodlines that give characters different pre-defined appearances, which can be finely tuned by the player.

Unlike many other MMOs, where there are numerous copies of the game universe intended to run at once (i.e., servers), Eve Online is functionally a single-universe game. There are

technically four copies of the universe running: the main server "Tranquility", the Chinese-based "Serenity", the event test server "Duality" that is a semi-public test server, and the test server "Singularity" (also "Sisi") which is a general, public test server[26] [27]. A new test server was announced called "Buckingham" to replace "Singularity" as the main EVE Online test server while "Singularity" was used for Dust 514/EVE Online joint testing. As DUST 514 is no longer active, "Singularity" is now the main test server again and "Buckingham" is a closed test server for the CCP developers[28].

2.2.2.3 Music

The Eve Online soundtrack was composed by Jón Hallur Haraldsson, also known as Real-X. Icelandic rap-rock group Quarashi also composed several tracks for the game. A digital soundtrack titled EVE Online: Original Soundtrack, Vol. 1 was released on iTunes on August 12, 2009. The soundtrack comes with an audio book track EVE Chronicle – Taught Thoughts. The soundtrack has since been removed from iTunes. The game itself contains an extensive in-game soundtrack. On December 4, 2012, the "Retribution" expansion of Eve Online was released. Among its features was the removal of the Jukebox, which enabled players to select their favorite songs to play[40]. In tandem with this, CCP Games announced that the entire game soundtrack (consisting of music in the game at the time) would be available to download for free from SoundCloud[41]. The soundtrack consists of 74 songs, with a running time of nearly seven hours.

2.2.3 FreeSpace 2

FreeSpace 2 is a 1999 space combat simulation computer game developed by Volition as the sequel to Descent: Free Space – The Great War. It was completed ahead of schedule in less than a year, and released to very positive reviews. However, the game became a commercial failure, and was described by certain critics as one of 1999's most unfairly overlooked titles[23].

The game continues on the story from Descent: Free Space, once again thrusting the player into the role of a pilot fighting against the mysterious aliens, the Shivans. While defending the human race and its alien Vasudan allies, the player also gets involved in putting down

a rebellion. The game features large numbers of fighters alongside gigantic capital ships in a battlefield fraught with beams, shells and missiles in detailed star systems and nebulae. Free multiplayer games were available via Parallax

Online which also ranked players by their statistics. A persistent galaxy was also available as SquadWar for players to fight with each other over territories.



Figure 2-3: FreeSpace 2 Game

2.2.3.1 Gameplay

FreeSpace 2's gameplay involves the player piloting a starfighter using mounted weapons to destroy enemy starfighters, performing reconnaissance behind enemy lines, or escorting other starships.[2][3] Its flight model is based on a looser interpretation of space physics instead of realistic Newtonian physics[4]. Hence, the ships are weightless and feel more responsive, though they require constant application of engine power to move[5]. The result is that the game plays more like a "WWII dogfight simulator" unaffected by gravity[6]. Although joysticks are the recommended controller for this game, the mouse is a viable alternative[7]. Single player mode is executed in the form of a campaign, which follows a story as a linear sequence of missions are executed.

The pre-mission briefing stage is where the player gets information on the background and objectives, and selects the ship and weapons.[8] The choices of ships and weapons increase as the player proceeds further along the campaign. Certain missions, however, will dictate certain ships and weapons to be used. Weapons can be classified into primary weapons and secondary weapons.[8] Primary weapons are kinetic and energy weapons, while missiles and torpedoes are classified as secondary weapons. Each weapon has its own specifications such as its rate of fire. They also inflict different damages on hulls (body of the ships) or shields (the protective energy fields surrounding the ships), or possess special effects such as shutting down specific electronic systems or propulsion.

The player flies around in a fighter with a first-person, in-cockpit view with a fully customizable fixed head-up display (HUD) as the visual interface[12][10]. The HUD displays video communications and relevant data on the ship's status and performance, weapons, objectives, and targets. It can also warn players from which direction missiles are locking onto them from, thus becoming an aide for launching countermeasures or taking evasive maneuvers[11]. Players have to maneuver into position and shoot through both shields and hull to destroy enemy ships[12]. While hull damage is unrecoverable, shields recharge over time. With the game supporting force feedback technology, joystick players will find their controllers vibrating or putting up resistance when they engage the afterburners or collide with objects.[19] Similarly, certain events, such as engaging afterburners and firing powerful weapons, will shake the screen as a form of visual feedback[11].

FreeSpace 2 has many helpful features available. The player can target enemies attacking a protected objective or match speeds with them. Power can be shunted between shields, engines, and weapons, thereby allowing faster recharge of shields, afterburners, and weapons at the expense of other subsystems. These features can be ignored without any detrimental effects on gameplay[14][15]. The mission parameters are not rigidly fixed, as there is an allowance for the failures of some primary objectives[11]. When the mission is concluded, a post-mission briefing will be conducted to discuss the mission, and the performance of the player, before the next mission can be taken on.

FreeSpace 2 allows multiplayer games to be played across a local area network (LAN) or over the Internet via the free services provided by Parallax Online (P XO)[16]. The player

can communicate with the other network players vocally through FreeSpace 2's own voice chat capability[17]. LAN play allows the players to play the standard player versus player modes such as deathmatch, or cooperate to complete multiplayer missions. They can even join in games which are already underway[20]. The same can be done over PXO but with the added incentive of having the players' statistics of kills and deaths being tracked on a ladder (ranking) system[13]. Players can also form up or join squadrons in SquadWar, an online persistent galaxy hosted by Volition on PXO, where squadrons fight each other for territories.

2.2.4 Elite Dangerous

Elite Dangerous is a space-flight simulation game developed and published by Frontier Developments. Piloting a spaceship, the player explores a realistic 1:1 scale open-world representation of the Milky Way galaxy, with the gameplay being open-ended. The game is the first in the series to attempt to feature massively multiplayer gameplay, with players' actions affecting the narrative story of the game's persistent universe, while also retaining single-player options. Elite Dangerous is the fourth game in the Elite video game series. It is the sequel to Frontier: First Encounters released in 1995[22].



Figure 2-4: Elite Dangerous Game

2.2.4.1 Gameplay

Beginning in the year 3300 upon its release in 2014, Elite Dangerous has now run to the year 3305 and has been running in sync with UTC, albeit 1286 years in the future. The game is set around 45 years after Frontier: First Encounters, the previous game in the series[22][29][16]. Elite Dangerous retains the basic premise of previous games – players start with a spaceship and a small amount of money and have to make their own way in an open galaxy, furthering themselves either legally or illegally, through trading, mining, exploration, bounty-hunting, piracy and assassination[30]. The game is the first in the series to feature online multiplayer, with players having access to a massively multiplayer persistent world, as well as an online-only single player mode[31]. Open Play gameplay is similar to Eve Online in that many actions which would be considered griefing in other multiplayer games are generally permitted here, so long as a valid roleplaying reason is attached. Examples include (but are not limited to) stealing from other players, extortion, and blocking off star systems via blockade or similar means[19]. However, some actions, such as "mob mentality" persecution of players, exploiting mechanics of the game (such as quitting the game in the middle of a fight to avoid death), and cursing are still not allowed, and could eventually result in a ban from the main server[22]. The player is able to explore the game's galaxy of some 400 billion star systems[23], complete with planets and moons that rotate and orbit in real time, resulting in dynamic day/night cycles[24]. Around 150,000 of the game's star systems are taken from real-world astronomical data while a few systems' fictional planetary systems as established in Frontier and First Encounters before significant numbers of exoplanets were discovered are carried over (e.g. none of the gas giants of the Fomalhaut system correspond with the detected properties of Fomalhaut b), and a handful of entirely fictional systems named in the original Elite and featured in later games are included (e.g. the original starting system Lave); the remainder are procedurally generated according to scientific models[32]. Throughout the galaxy, the player is able to dock with space stations and outposts to trade goods, purchase new spacecraft, re-arm their ship, effect repairs and to seek or complete missions from text-based station "Mission boards"[33]. The player may also find cargo or encounter other ships while in flight by investigating 'Unidentified Signal Sources'.

2.3 Problem Statement

2.3.1 Graphics problem

Graphics getting discoloring which almost looks like artifacting. Had this same problem with a few other old games to and was wondering is it that PCs these days are too powerful for these old games, because of newer drivers were the main issue to these problems.

2.3.2 Project estimation

Estimating any type of project can be hard, but it is especially true for video games. For an industry that is as fast changing as this, you have to keep a tight lid on schedule. Otherwise you'll end up always iterating and implementing new ideas until the end of time.

A cool new feature always sounds good, but do you have the time? Does the benefits outweigh the cost? Does much of the code have to be changed?

These are just some of the questions you have to answer before starting anything new.

There are countless development teams who failed to deliver because of poor time allocation. Having good management is key, someone who oversees every aspect of the development process and can evaluate the ideas and shut them down or postpone them to another project[18].

2.3.3 Preproduction phase

Or a lack thereof. This problem is common in almost every game development studio, small and large too. Usually when a game ships, the whole team jumps to the next project, skipping most of the preproduction phase. Without proper preproduction, many design decisions are made during proper game production phase, which is highly dangerous, and can be damaging to the product or to the schedule (in worst cases both). Lack of preproduction can cause lack of coherence in your product. You can create fun gameplay, beautiful graphics, great music and sound effects, but if these elements don't play well together, then you failed to create a great game. We try to eliminate this problem by starting the preproduction process with few people before the regular production of the previous game is finished. If you do your homework (preproduction) properly, then the whole team has a strong coherent vision from the beginning, which is essential in successful game development.

2.3.4 Prototyping

Deciding if a feature is valid can be a problematic process. When you implement it and it doesn't work do you keep polishing it or just cut it entirely? You have to weight the pros and cons. If you keep working on it you could loose a lot of work hours. But if you cut it, you might just lost a great feature that only needed more work.

This is where experience is really important. After a few projects you have a better overall understanding of how much time should you invest in it and have a better grasp of your limits[13].

2.3.5 Getting tired during the project

We really believe, if you work in the entertainment business (games, films, music, etc.), your work loudly speaks about your attitude. You really have to believe in the project to do a good job on it. If the only reason you do it, is to get a pay check at the end of the month, it will definitely show up in your work. Being excited for a project in the beginning is easy, the trick is to sustain this attitude for the whole duration of the development. It can be really difficult, when you are working on a huge scale project with very short deadline. In this situation the whole team is forced to crunch (working crazy long hours). For a few days it's manageable, but a few months of non-stop work going to burn and tire you out mentally and physically. Here we try to build a healthy working culture, working 8 focused hours, and crunching for short periods of time only, when its absolutely necessary[32].

2.3.6 Polish

Video games are carefully designed experiences. And nothing can ruin this more like a badly executed feature, weird anomalies and bugs. Some say, that the last 10% of development time is more important than the first 90%, and it's true to some extent. It can be really demanding to polish everything in time, when you are facing a tough release schedule.

The best way to avoid long crunch periods is to keep a tight lid on executing everything don't just put them off for later. The closer the deadline is, the more of the things you put off will come back to haunt you.

2.3.7 Communication

Nowadays making game is a more open process than anytime before. Thanks to the abundant communication channels, developers can get in touch easier with their audience than ever before. Releasing an alpha or a beta version of your product can yield important data, in both the game's state and the reception of it[26]. If something is not sitting well with your potential consumers, you'll have an immediate feedback. Don't be shy and don't get discouraged about negative response. The target audience can be especially demanding and sometimes even toxic, but if you sift through it, analysing the data can be beneficial.

2.3.8 Advertising

This is maybe the most crucial aspect of making a successful product. A game can be good, but if the people have no way of knowing about it, it is going to fail. Everyday hundreds of games are released on every platform. Making yours known is a tough job. Especially if you are on a budget[28].

Advertising is expensive but there are other ways. Social media, blogs and forums are a great jumping point. Know your target audience. Have dialogue. People will want their voices to be heard. Build a community. This way every time you start something new you already know where to start.

It is also a good idea to get in touch with the relevant press to have some coverage on your game.

2.4 Supporting Literature

2.4.1 Blueprint Language

The Blueprints Visual Scripting system in Unreal Engine is a complete gameplay scripting system based on the concept of using a node-based interface to create gameplay elements from within Unreal Editor. As with many common scripting languages, it is used to define object-oriented (OO) classes or objects in the engine. As you use UE4, you'll often find that objects defined using Blueprint are colloquially referred to as just "Blueprints."

This system is extremely flexible and powerful as it provides the ability for designers to use virtually the full range of concepts and tools generally only available to programmers.

In addition, Blueprint-specific markup available in Unreal Engine's C++ implementation enables programmers to create baseline systems that can be extended by designers.

The first generation Unreal Engine was developed by Tim Sweeney, the founder of Epic Games[34]. Sweeney, who already had experience on creating editing tools, such as those programmed for ZZT (1991) and Jill of the Jungle (1992)[4] began writing the engine in 1995 for the production of a game that would later become known as Unreal, a first-person shooter. After years in development, it debuted with the game's release in 1998[7] although licensees such as MicroProse and Legend Entertainment had possessed the technology much earlier, with the first licensing deal taking place in 1996.

Both software and hardware rendering were present in the foundational software, as well as collision detection, colored lighting, and a rudimentary version of texture filtering. The engine also provided a level editor, UnrealEd (formerly Unreal World Editor that had support for real-time constructive solid geometry operations as early as 1996, allowing mappers to change the level layout "on the fly"[13][14]. Other features implemented during the engine's development included real-time direct illumination and light sourcing, which were respectively integrated in 1995 and 1997[15][16]. In addition to having support for Microsoft Windows, Linux and Mac, Unreal Tournament also opened the platform to PlayStation 2 and, with the help of Secret Level, to Dreamcast. In 2000, Epic updated the engine with new improvements, including higher-polygon models and architecture, a skeletal animation system and large-scale terrain support[20].

By late 1999, The New York Times indicated that the number of external projects using Epic's technology was 16, naming the likes of Deus Ex, The Wheel of Time and Duke Nukem Forever,[20] the title from 3D Realms that was set to debut the series on the GameCube console[21]. While it cost around \$3 million to produce and licenses for up to \$350,000,[17] Epic gave modders the ability to create their own worlds with the incorporation of UnrealEd and a scripting language called UnrealScript in its games, sparking a community of enthusiasts around a game engine built to be extensible and improved over multiple generations of games.

2.4.2 How Do Blueprints Work?

In their basic form, Blueprints are visually scripted additions to your game. By connecting Nodes, Events, Functions, and Variables with Wires, it is possible to create complex gameplay elements. Blueprints work by using graphs of Nodes for various purposes - object construction, individual functions, and general gameplay events - that are specific to each instance of the Blueprint in order to implement behavior and other functionality[43].

2.4.3 Commonly Used Blueprint Types

The most common Blueprint types you will be working with are Level Blueprints and Blueprint Classes. These are just two of the Types of Blueprints, which also include Blueprint Macros and Blueprint Interfaces.

2.4.4 Level Blueprint

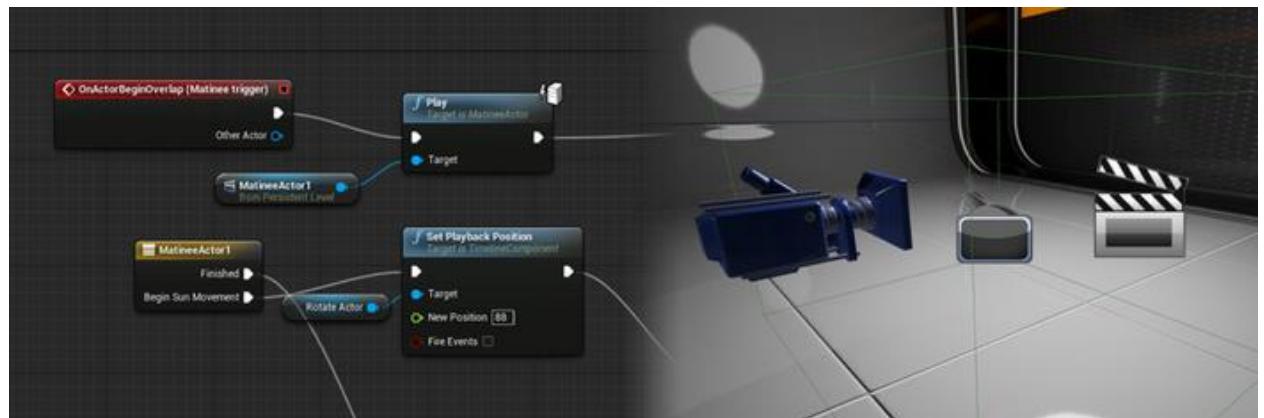


Figure 2-5: Level Blueprint

The Level Blueprint fills the same role that Kismet did in Unreal Engine 3, and has the same capabilities. Each level has its own Level Blueprint, and this can reference and manipulate Actors within the level, control cinematics using Matinee Actors, and manage things like level streaming, checkpoints, and other level-related systems. The Level Blueprint can also interact with Blueprint Classes (see the next section for examples of these) placed in the level, such as reading/setting any variables or triggering custom events they might contain.

2.4.5 Blueprint Class



Figure 2-6: Blueprint Class

Blueprint Classes are ideal for making interactive assets such as doors, switches, collectible items, and destructible scenery. In the image above, the button and the set of doors are each separate Blueprints that contain the necessary script to respond to player overlap events, make them animate, play sound effects, and change their materials (the button lights up when pressed, for example).

In this case, pressing the button activates an event inside the door Blueprint, causing it to open - but the doors could just as easily be activated by another type of Blueprint, or by a Level Blueprint sequence. Because of the self-contained nature of Blueprints, they can be constructed in such a way that you can drop them into a level and they will simply work, with minimal setup required. This also means that editing a Blueprint that is in use throughout a project will update every instance of it[43].

2.4.6 Unreal Games Engine 4

In August 2005, Mark Rein, the vice-president of Epic Games, revealed that Unreal Engine 4 had been in development since 2003.[58] Until 2008, development was "basically" done by Sweeney.[59] The engine targets the eighth generation of consoles, PCs, and Tegra K1-based devices running Android announced in January 2014 at CES.

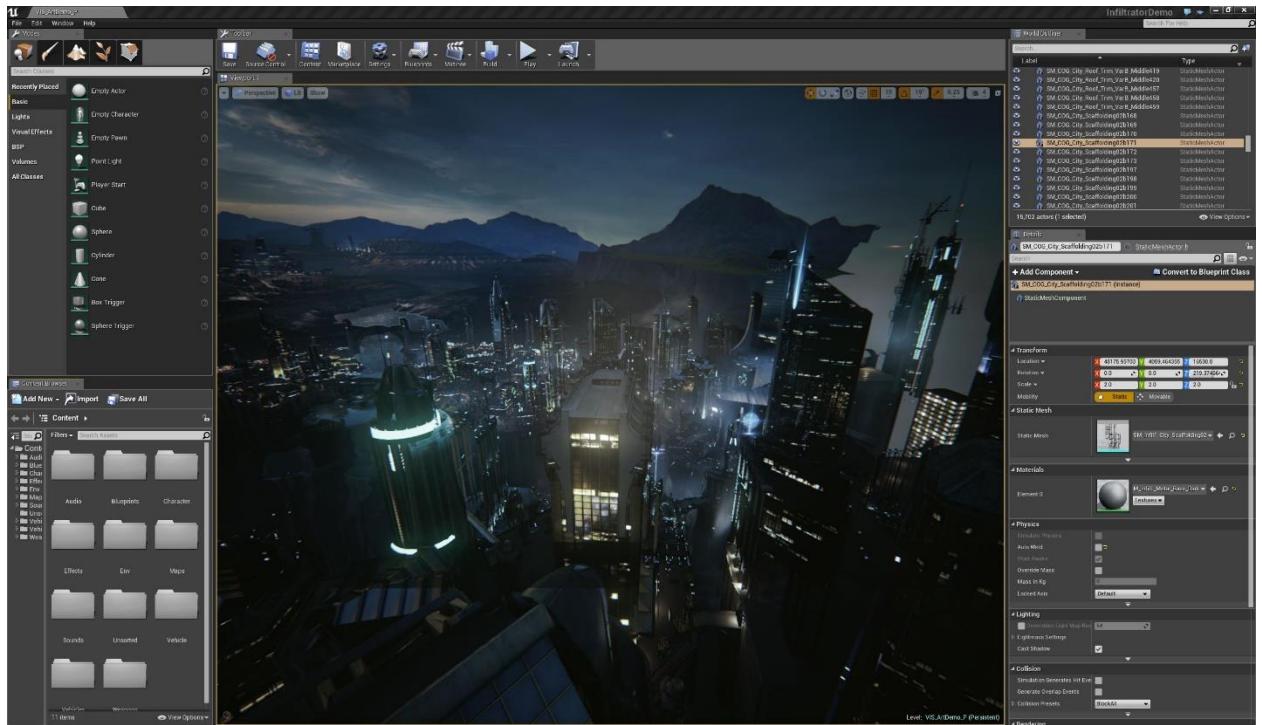


Figure 2-7: Unreal Game Engine4

In February 2012, Rein said "people are going to be shocked later this year when they see Unreal Engine 4"[60]. Unreal Engine 4 was unveiled to limited attendees at the 2012 Game Developers Conference, and video of the engine being demonstrated by technical artist Alan "Talisman" Willard was released to the public in June 7, 2012 via GameTrailers TV[61] [62].

One of the major features planned for UE4 was real-time global illumination using voxel cone tracing, eliminating pre-computed lighting[15]. However, this feature has been replaced with a similar but less computationally-expensive algorithm prior to release for all platforms including the PC due to performance concerns[63]. UE4 also includes new developer features to reduce iteration time, and allows updating of C++ code while the engine is running. The new "Blueprints" visual scripting system (a successor to UE3's "Kismet") allows for rapid development of game logic without using C++, and includes live debugging[65][66]. The result is reduced iteration time, and less of a divide between technical artists, designers, and programmers.

2.4.7 Adobe Photoshop

(For designing 2D objects like MainMenu) Adobe Photoshop is a popular image changing software package. It is widely used by photographers for photo editing (fixing colors, reducing noise, adding effects, fixing brightness/contrast) and by graphic designers and Web designers to create and change images for web pages[61].



Figure 2-8: Adobe Photoshop CS6

2.4.8 Blender

Blender (For creating 3D models) Blender is a professional, free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, interactive 3D applications and video games. Blender's features include 3D modules, UV unwrapping, texturing, raster graphics editing, rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, sculpting, animating, match moving, camera tracking, rendering, motion graphics, video editing and compositing. It also features an integrated game engine.

On January 1, 1998, Blender was released publicly online as SGI freeware[1]. NeoGeo was later dissolved and its client contracts were taken over by another company. After NeoGeo's

dissolution, Ton Roosendaal founded Not a Number Technologies (NaN) in June 1998 to further develop Blender, initially distributing it as shareware until NaN went bankrupt in 2002. This also meant, at the time, discontinuing the development of Blender[23].

In May 2002, Roosendaal started the non-profit Blender Foundation, with the first goal to find a way to continue developing and promoting Blender as a community-based open-source project. On July 18, 2002, Roosendaal started the "Free Blender" campaign, a crowdfunding precursor[37][35]. The campaign aimed for open-sourcing Blender for a one-time payment of €100,000 (US\$100,670 at the time) collected from the community[38]. On September 7, 2002, it was announced that they had collected enough funds and would release the Blender source code. Today, Blender is free and open-source software largely developed by its community, alongside two full-time and two part-time employees employed by the Blender Institute[15].

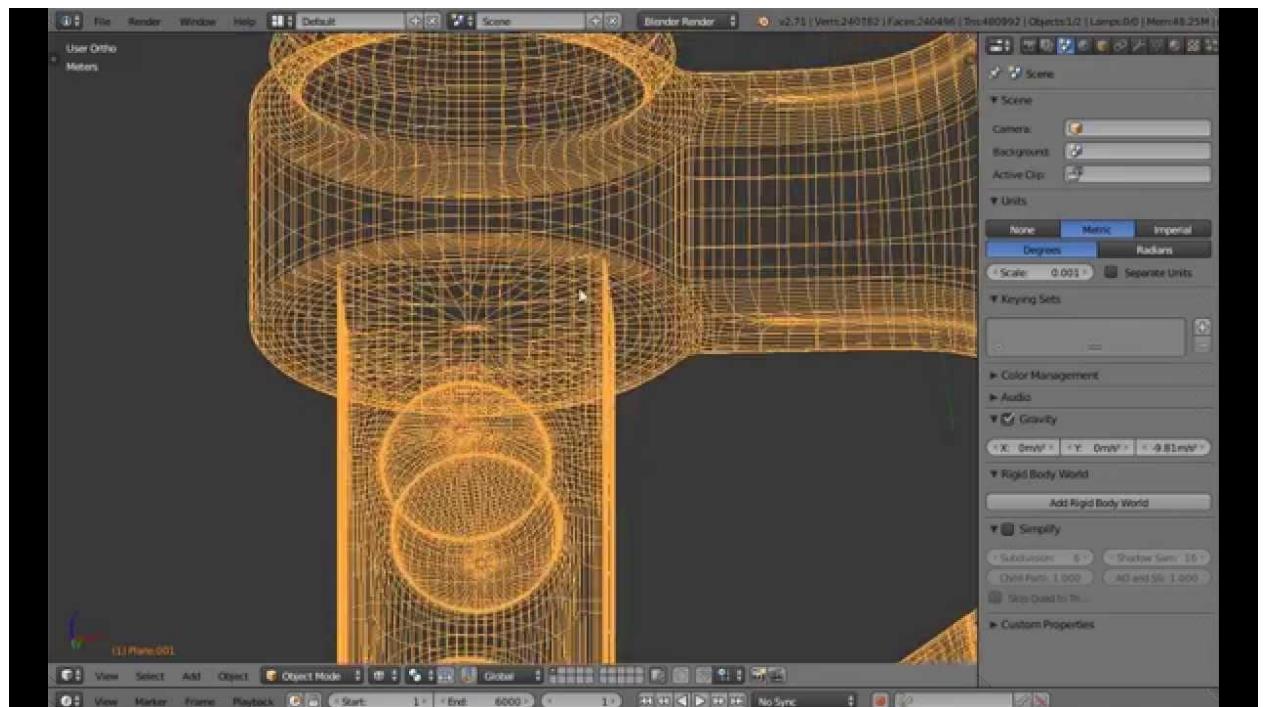


Figure 2-9: Blender

2.4.9 Entity Relationship Diagram (ERD)

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a data modeling

technique that can help define business processes and be used as the foundation for a relational database. Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later. However, while an ERD can be useful for organizing data that can be represented by a relational structure, it can't sufficiently represent semi-structured or unstructured data. It's also unlikely to be helpful on its own in integrating data into a pre-existing information system[18].

2.4.10 General Overview of ERD

An entity relationship diagram consists of several components. Components those are frequently used to represent an e-r diagram are

- Entity
- Weak entity
- Attribute
- Multi valued attribute
- Derived attribute
- Relationship

2.4.10.1 Entity

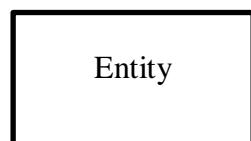


Figure 2-10: Entity

An entity can be a person, place, event, or object that is relevant to a given system. For example, a school system may include students, teachers, major courses, subjects, fees, and other items. Entities are represented in ER diagrams by a rectangle and named using singular nouns. It is represented by rectangle.

2.4.10.2 Weak Entity

A weak entity is an entity that depends on the existence of another entity. In more technical terms it can be defined as an entity that cannot be identified by its own attributes. It uses a foreign key combined with its attributes to form the primary key. An entity like order item is a good example for this. The order item will be meaningless without an order so it depends on the existence of order. It is represented by double rectangle.

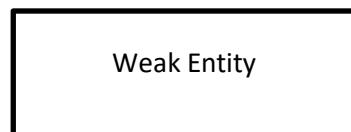


Figure 2-11: Weak Entity

2.4.10.3 Attribute

An attribute is a property, trait, or characteristic of an entity, relationship, or another attribute. It is represented by an ellipse.

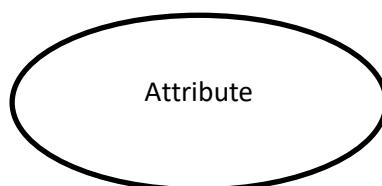


Figure 2-12: Attribute

2.4.10.4 Multi Valued Attribute

If an attribute can have more than one value it is called a multi valued attribute. It is important to note that this is different to an attribute having its own attributes. For example a teacher entity can have multiple subject values. It is represented by a double ellipse.

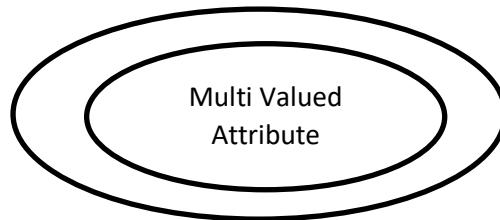


Figure 2-13: Multi Valued Attribute

2.4.10.5 Mapping Cardinality of ERD

Cardinality refers to the number of entity objects on each side of the relationship. In e-r diagram there are four types of mapping cardinalities.

For example: a customer can order products one after another.

- One-to-One
- One-to-Many or Many-to-One (dependent on the direction)
- Many-to-One
- Many-to-Many
-

2.4.10.6 One-to-One

A one-to-one relationship is the simplest relationship between two beans. One entity bean relates only to one other entity bean. For example: a customer can be kept only in one word/cell at a time.

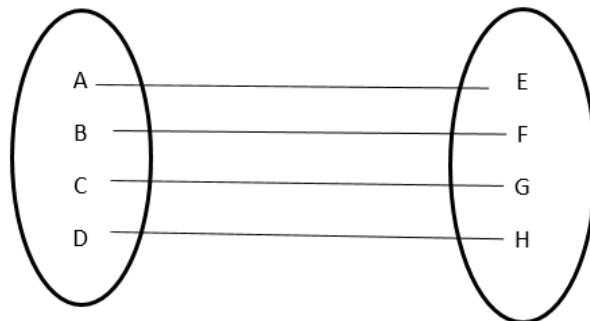


Figure 2-14: One-to-One

2.4.10.7 One-to-Many

In a one-to-many relationship, one object can reference several instances of another.

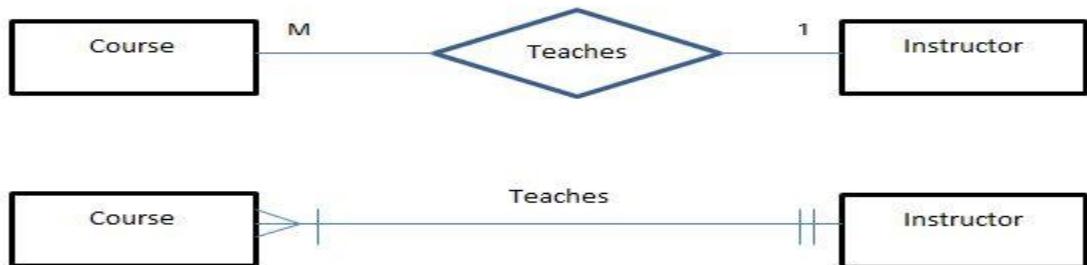


Figure 2-15: One-to-Many

2.4.10.8 Many-to-One

In a many-to-one relationship, many objects can reference one instance of another.

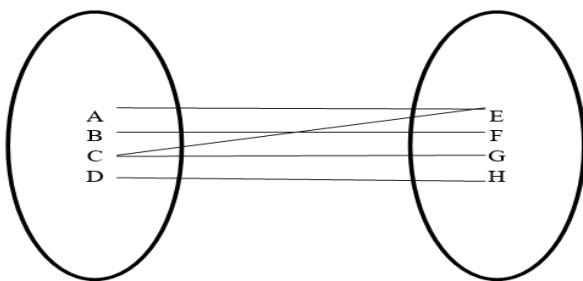


Figure 2-16: Many-to-One

2.4.10.9 Many-to-Many

A many-to-many relationship is complex. In a many-to-many relationship, many objects can reference many objects. This cardinality is the most difficult to manage.

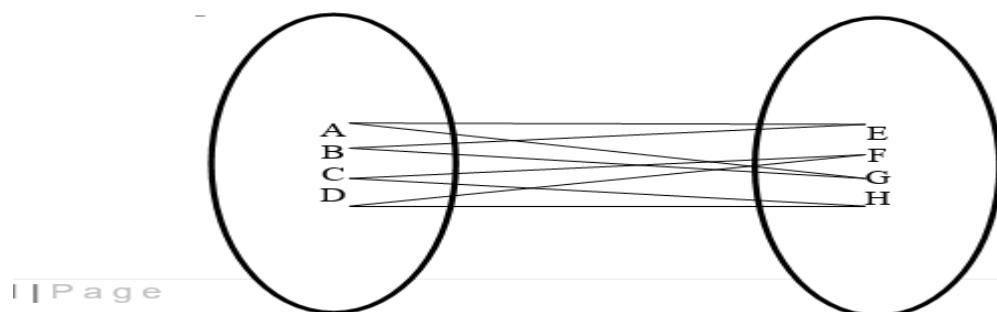


Figure 2-17: Many-to-Many

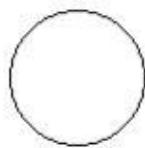
2.4.11 Data Flow Diagram

A Data Flow Diagram (DFD) is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describes the flow of data and the processes that change or transform the data throughout a system. This network is constructed by using a set of symbols that do not imply any physical implementation. It has the purpose of clarifying system requirements and identifying major transformations. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. DFD can be considered to an abstraction of the logic of information oriented or a process-oriented system flow-chart. For these reasons DFD's are often referred to as logical data flow diagrams[24].

DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structures of DFD allow starting from a broad overview and expand it to a hierarchy of detailed diagrams. DFD has often been used due to some reasons such as logical information flow of the system, determination of physical system construction requirements, simplicity of notation, establishment of manual and automated system requirements.

DFDs only involve four symbols. They are:

- Process
- Data Object
- Data Store
- External entity



Process

Transform of incoming data flow(s) to outgoing flow(s).



DataFlow

Movement of data in the system.



DataStore

Data repositories for data that are not moving. It may be as simple as a buffer or a queue or as sophisticated as a relational database.



ExternalEntity

Sources of destinations outside the specified system boundary.

Figure 2-18: DFD Component

2.4.11.1 DFD Rules and Tips

- Each process should have at least one input and an output.
- Each data store should have at least one data flow in and one data flow out.
- Data stored in a system must go through a process.
- All processes in a DFD go to another process or a data store.

2.4.11.2 DFD Levels and Layers

A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece. DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond.

1. DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance

view, showing the system as a single high-level process, with its relationship to external entities.

- DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. The main functions carried out by the system will be highlighted, as the high-level process of the Context Diagram is broken down into its sub processes.

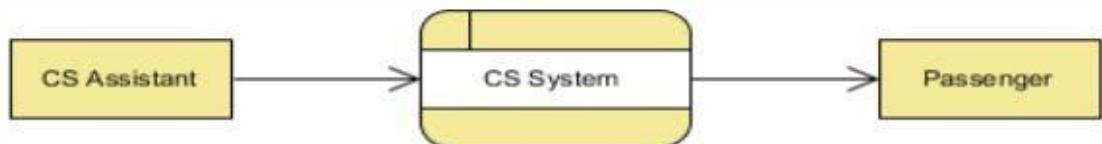


Figure 2-19: 0-Level Data Flow Diagram of Customer Service System

The figure above shows a context Data Flow Diagram that is drawn for a railway company's Customer Service System. It contains a process (shape) that represents the system to model, in this case, the "CS System". It also shows the participants who will interact with the system, called the external entities. In this example, CS Assistant and Passenger are the two entities who will interact with the system. In between the process and the external entities, there are data flow that indicates the existence of information exchange between the entities and the system.

Level-1 Data Flow Diagram:

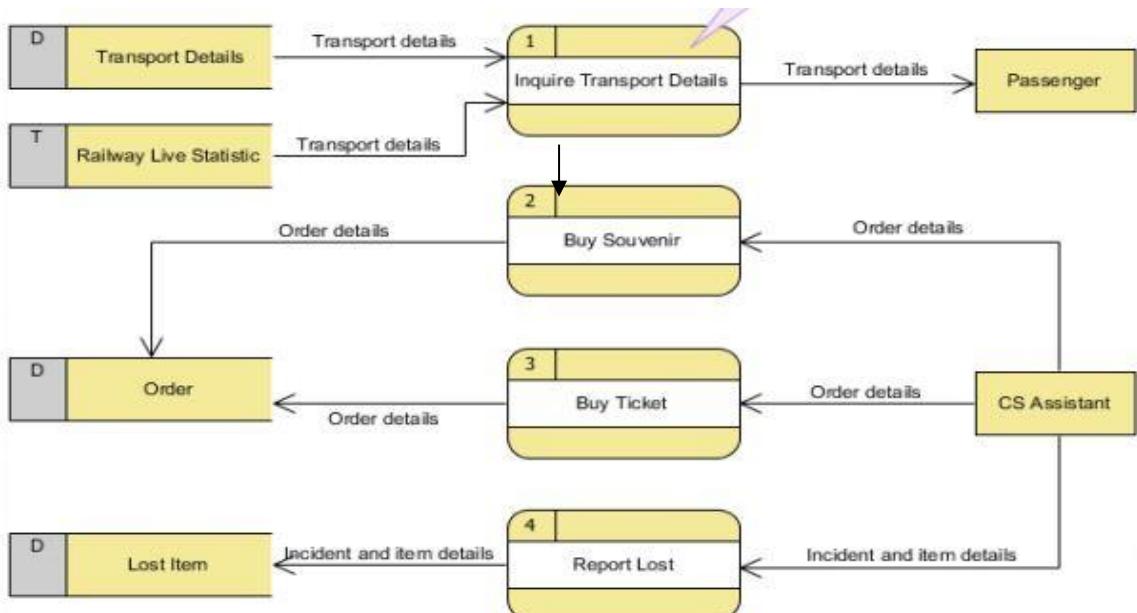


Figure 2-20: Level-1 Data Flow Diagram of Customer Service System

The figure above shows the level 1 DFD, which is the decomposition of the CS System process shown in the context DFD. Read through the diagram and then we will introduce some of the key concepts based on this diagram. The CS System Data Flow Diagram example contains four processes, two external entities and four data stores. Although there is no design guidelines that governs the positioning of shapes in a Data Flow Diagram, we tend to put the processes in the middle and data stores and external entities on the sides to make it easier to comprehend. Based on the diagram, we know that a Passenger can receive Transport details from the Inquiry Transport Details process, and the details are provided by the data stores Transport Details and Railway Live Statistic. While data stored in Transport Details are persistent data (indicated by the label "D"), data stored in Railway Live Statistic are transient data that are held for a short time (indicated by the label "T") [36]. A callout shape is used to list out the kind of details that can be inquired by passenger. CS Assistant can initiate the Buy Souvenir process, which will result in having the Order details stored in the Order data store. Although customer is the real person who buy souvenir, it is the CS Assistant who accesses the system for storing the order details. Therefore, we make the data flow from CS Assistant to the Buy Souvenir process. CS Assistant can also initiate the Buy Ticket process by providing Order details and the details will be stored again in the Order data store. Data Flow Diagram is a high level diagram that is drawn with a high degree of abstraction. The data store Order which is drawn here does not necessarily imply a real order database or order table in a database. The way how order details are stored physically is to be decided later on when implementing the system. Finally, CS Assistant can initiate the Report Lost process by providing the Incident and item details and the information will be stored in the Lost Item database.

2.4.12 Use Case Diagram

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system. Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities. They also help identify any internal or external factors

that may influence the system and should be taken into consideration. They provide a good high level analysis from outside the system. Use case diagrams specify how the system interacts with actors without worrying about the details of how that functionality is implemented.

2.4.12.1 Basic Use Case Diagram Symbols and Notations

2.4.12.1.1 System

Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.

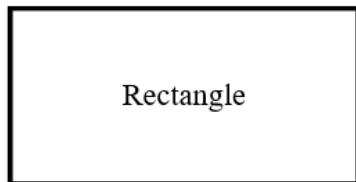


Figure 2-21: Rectangle

2.4.12.1.2 Use Case

Draw use cases using ovals. Label the ovals with verbs that represent the system's functions.



Figure 2-22: Use Case

2.4.12.1.3 Actors

Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.

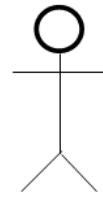


Figure 2-23: Actors

2.4.12.1.4 Relationships

Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.



2.4.13 Relational Database (RDB)

A relational database (RDB) is a collective set of multiple data sets organized by tables, records and columns. RDBs establish a well-defined relationship between database tables. Tables communicate and share information, which facilitates data search ability, organization and reporting. RDBs use Structured Query Language (SQL), which is a standard user application that provides an easy programming interface for database interaction. RDB is derived from the mathematical function concept of mapping data sets and was developed by Edgar F. Codd.

RDBs organize data in different ways. Each table is known as a relation, which contains one or more data category columns. Each table record (or row) contains a unique data instance defined for a corresponding column category. One or more data or record characteristics relate to one or many records to form functional dependencies. These are classified as one to one, one to many, many to one and many to many. Relational databases

are designed to make data entry and management simpler, quicker and less prone to data corruption by storing information so that each piece of data is stored only once, and referenced by other pieces of data when needed. This is achieved by creating relations between data sets, hence the term relational database.

2.5 Conclusions

Even though the game might not the standards of many commercial games, given the resources and time frame. The game is easy to play and the visual effects make the game look graphically good. Each effect contributes to the appearance of the game. The game is easy to play and the visual effects make the game look best. Each effect donates to the look of the game own laptop. There are a lot of positive aspects to working together with team. When problems occur everyone can help, ideas can be discussed and you get to know each other better, making it more motivating and fun with members to work on a project. In hindsight, we should probably have utilized the rooms with computers provided by college in order to increase productivity.

Chapter- 3

Proposed Solution/ Methodology

3.1 Introduction

The Space Shooter Games is a feature-complete game template made in 100% Blueprints that allows you to create a 3D space shooter similar to games like Starfox. Boss and level winning mechanism. Play in two game play modes - Track Mode and All-Range Mode. You can also switch between 3 camera angles - First-Person, Third-Person, and Back-facing. The Space comes with abilities, such as, Roll, Barrel Roll, Boost, Break, Somersault, and Uturn. Comes with a weapon-switching mechanic built-in to easily add your own weapons! It also has these pickups included: Health, Shield, Points, Points Cache, Laser Upgrade, and Bombs!

Comes with a fully functional HUD and menu system that includes game play and graphics settings. The template also comes with quit and replay functionality for 2 save slots and a Level-Linking System[44].

The template also comes with level-building tools like Procedural Debris, Road Builder, and Corridor Builder. It also comes with Actor Components that allow you to quickly add functionality to any actor in your scene.

A spaceship with configurable weapon turrets. 2 control modes: MOUSE and KEYS. Basic gun and homing missile weapons. Supports damage on crash. Supports enemy spawning and recycling. A simple HUD that displays game stats. Tested on PC and mobile.*

3.2 Feasibility Study

Feasibility study is a process to check possibilities of system development. It is a method to check various different requirements and availability of financial & technical resources.

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information

Such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to

establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards.

Before starting the process various parameters must be checked like:

- Estimated finance is there or not?
- The man power to operate the system is there or not?
- The man power is trained or not?
- All the above conditions must be satisfied to start the project. This is why in depth analysis of feasibility is carried out.
- There are three different ways feasibility can be tested
- Technical Feasibility
- Economical Feasibility
- Operational Feasibility
- Behavioral Feasibility
- Legal Feasibility

3.2.1 Technical Feasibility

In this, one has to test whether the system can be developed using existing technology or not. It is evident that necessary hardware and software are available for development and implementation of proposed system. We acquired the technical knowledge of working in languages, and then only we have started designing our project. The system is self-explaining and does not need any entire sophisticated training. A system has been built by concentrating on the graphical user interface concepts, the application can also be handled very easily with a novice uses. The overall time that a user needs to get trained is less than 15 minutes.

Technical feasibility performs following tasks.

- Analyzes the technical skills and capabilities of the software development team members.
- Determines whether the relevant technology is stable and established.
- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.
- It mentions new hardware requirements of proposed system.

- It Mentions Computer with new configuration requirements of proposed system.
- It mentions new software requirements of the proposed system.

3.2.2 Economical Feasibility

Economic feasibility is a measure of the cost-effectiveness of a project or solution. As a part of this, the costs and benefits associated with the proposed system are compared and the project is economically feasible only if tangible and intangible benefits outweigh the cost. The cost for proposed matrimony system is much less than traditional system as the eligible bachelor or their family doesn't have to go through the long process of meeting each other and having to choice so many times. The system also reduces the traditional or manual system will be replaced such "e-ballot"/doodle or various websites website can do. So, this system is economically feasible.

Economic feasibility also performs following tasks.

- Cost incurred on software development to produce long-term gains for an organization.
- Cost required conducting full software investigation (such as requirements elicitation and requirements analysis).
- Cost of hardware, software, development team, and training.
- Cost involves in purchase or rental of equipment.
- Cost of phones & mobile communication equipment.
- Cost of salaries of employee.
- Cost of maintenance of equipment.

3.2.3 Operational Feasibility

Operational feasibility means how much the system is user interactive. In this project, the management will know the details of each project where he may be presented and the data will be maintained as decentralized and if any inquires for that particular contract can be known as

Per their requirements and necessities.

Operational feasibility also performs following tasks.

- Determines whether the problems anticipated in user requirements are of high priority.

- Determines whether the solution suggested by the software development team is acceptable.
- Analyzes whether users will adapt to a new software.
- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.
- It finds if any job reconstruction is required or not?
- Watches the feelings of the customers as well as user.
- System should provide right & accurate information to user or customer at right place as well as at right time.

3.2.4 Behavioral Feasibility

Our system is capable of providing user interactive interface for all devices. It is very user interactive. Any people can easily get a clear idea about the whole system. Admin can easily handle the report and make their analysis because it is very important for the improvement of the ERP business. Report analysis will make us taking helpful decisions.

3.2.5 Legal Feasibility

Legal feasibility determines whether the proposed system conflicts with legal requirements, e.g. the Data Protection Act. It will be done by some legal advisors. The feasibility of Online-Voting is based on the widespread Internet access and the use of digital ID cards. These personal identification documents with the size of a credit card allow citizens and residents to digitally sign documents and use private and governmental online services that require secure authentication.

3.3 Key Features of Proposed System

Space Shooter: game puts you at the galaxy on fire with infinity shooting war. You will be faced an increasingly large number of enemies and deal with many striker bosses in space war.

- Highly customizable Player Blueprint with dozens of parameters to tweak its behavior to your liking!
- Level-linking system
- Amazing lighting and special effects
- Includes Power-ups and Bosses!
- PVP - online shooting games
- Kill space intruders
- Change your weapons after getting power.

3.3.1 Functional Requirements of the Proposed System

It is an online-based appointment system. Therefore, bulletin also plays an important role for the advertiser in helping them especially in the fields of product sales and marketing. Currently, use of different media will cause different marketing and communication results for consumers. Generally, advertising media is divided into five major categories: magazine/newspaper, TV, radio broadcasts, Internet, and also mobile communications. Advertising has changeover the years-it have gone from the rudimentary forms of television, radio, newspapers, and billboards to the emergence of mobile advertising witnessed in the last decade. The usage of voting website has increased significantly every year. This results in competition among developer to produce the application that fulfills the needs of the consumer. Online user subscriptions in Malaysia have shown an increasing trend in the past five years.

Functional requirements are the capabilities of the system and domain specific. The Online games System (UIBOVS) would have the following functional requirements:

- The system must provide the voters with accurate data
- The system must supply standard reports for decision making
- Audit trails of who made changes to the database must be maintained
- The system should allow voting administrators to make updates to the voter information database
- The system must provide standard error checking

- The system must provide data integrity checks to ensure data remains consistent and updated.
- The system should provide documentation to inform users of system functionality and any change to the system.
- Ease of GUI use that can be accessed via web browser must be established

3.3.2 Non-Functional Requirements of the Proposed System

Games is the most popular application in Bangladesh. Non-functional requirements specify the system's 'quality characteristics' or 'quality attributes'.

Many different stakeholders have a vested interest in getting the non-functional requirements right particularly in the case of large systems where the buyer of the system is not necessarily also the user of the system.

The importance of non-functional requirements is therefore not to be trifled with. One way of ensuring that as few as possible non-functional requirements are left out is to use non-functional requirement groups. For an explanation on how to use non-functional requirement group

Non-Functional requirements are constraints on the functional requirements or quality

Requirements. The non-functional requirements of the system include:

- Performance – for example Response Time, Throughput, Utilization, Static Volumetric
- Scalability
- Capacity
- Availability
- Reliability
- Recoverability
- Maintainability
- Serviceability
- Security
- Regulatory
- Manageability

- Environmental
- Data Integrity
- Usability
- Interoperability
- games is carried out from many computers using the internet
- Highly secure platform
- Backup data restore capabilities should be granted
- The system must conform to the requirements of the Government rules and regulations.

3.3.3 Advantages of Proposed System

Our online games system is a system where user can play games through online and also offline. The advantages of proposed system are listed below:

- Online gaming makes the kid sharper and mentally more active. The games generally have various levels or missions to be completed in limited times. This helps the kids in learning about time management.
- Mind and hand coordination. This is one of the main and important advantages that the kids experience while playing online games. He/she learns to coordinate his/her mind with the actions of his hands. While executing those actions, they also develop mental strength.
- Kids tend to become socially active, as they interact and play with complete strangers online. It helps them in their social life too.
- Secrecy - You can teach kids to keep their personal information a secret, especially not to share it on the unknown sites.
- Most importantly, online gaming should be played only be for fun, and not for any accessing age restricted activities like gambling.
- Develop empathy -- Since there are a multitude of group games, players have to make decisions and know that they can have consequences, both positive and negative, on the other players in the game. Therefore, it is a means that facilitates the development of empathy (Levine, 2009 Reyes-Hernández et al., 2014).
- Increase the sense of competence and self-esteem-- The use of video games also creates feelings of self-esteem and competition in players game after game. As you

overcome the different challenges and levels, you feel better about yourself and able to meet the following objectives.

3.4 System Design

Design is the first step into the development phase for any engineered product or System. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

3.4.1 Entity Relationship Model

An ER diagram (ERD), also called an entity relationship model, and is a graphical representation of a system's entities and their relationships to each other. Entities refer to people, things and actions, etc., while relationships refer to the inter-relations of the entities. For example, when the event in which "an employee places an order for a product" is applied to an ERD, the "admin" and the "registration" represent the entities and "permits" which connects these entities represents the relationship. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

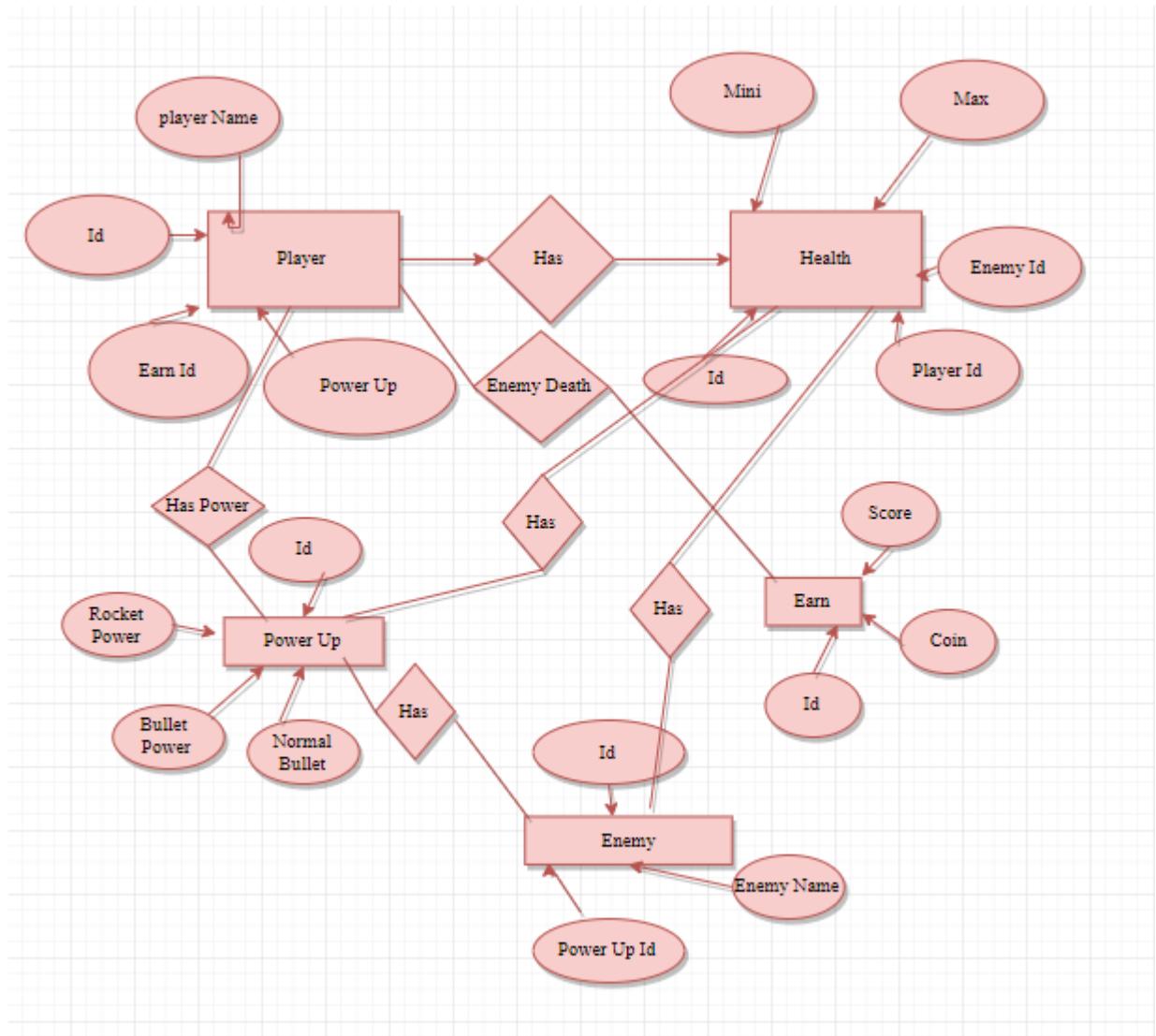


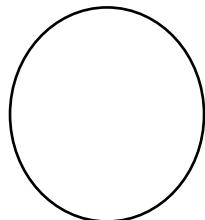
Figure 3-1: ER-Diagram for the Project

3.4.2 Data Flow Diagram

A picture is worth a thousand words. A Data Flow Diagram (DFD) is traditional visual representation of the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

Here, DFDs only involve four symbols. They are:

1. Process.
2. Data Object.
3. Data Store.
4. External entity.



Process

Transform of incoming data flow(s) to the outgoing flow(s)



Data Flow

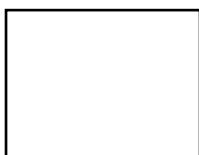
Movement of data in the system



Data Store



Data repositories for data that are not moving. It may be as simple as a buffer or a queue or a sophisticated as a relational database



External Entity

Sources of destinations outside the specified system boundary

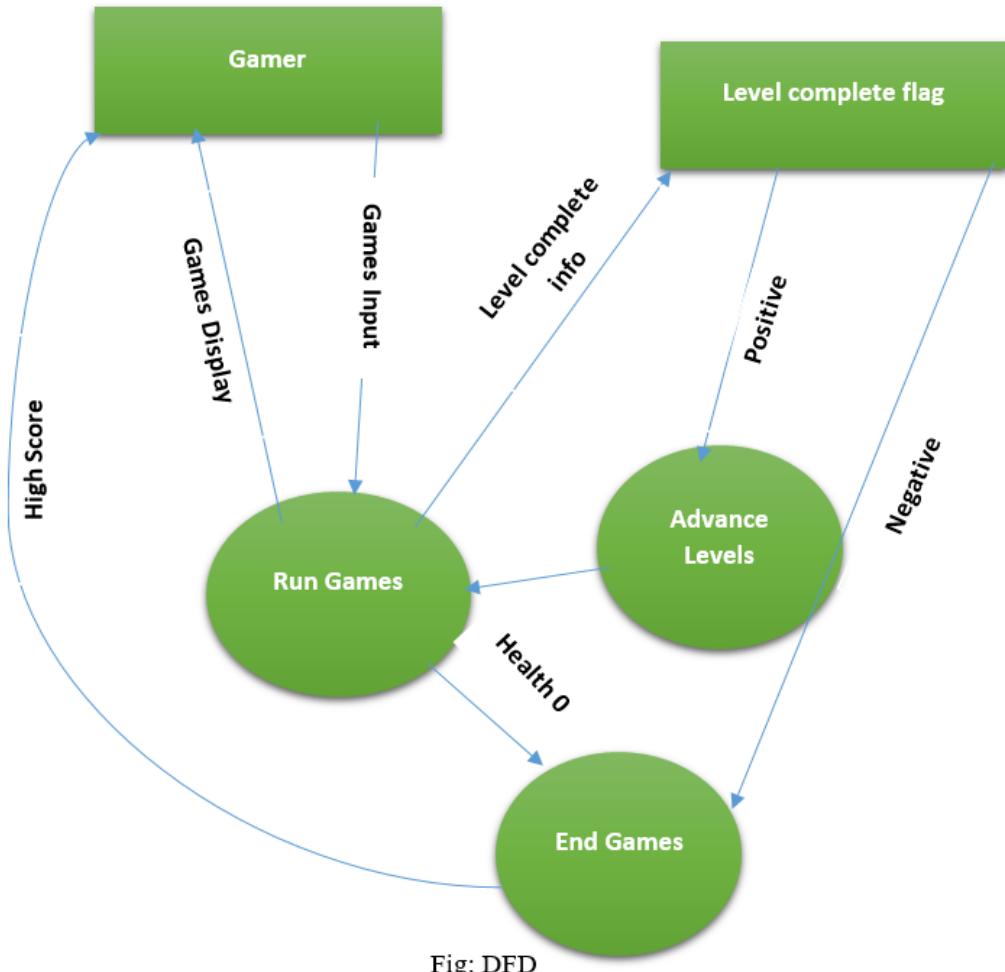


Figure 3-2: DFD Space Ship Games

3.5 Database Design

This section will describe how the system's database is designed. Database design is an important part of software design. The database design is directly related to the merits of the program to achieve the efficiency and simplicity of data access.

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

- Primary Key - the field that is unique for all the record occurrences.
- Foreign Key - the field used to set relation between tables. Normalization is a technique to avoid redundancy in the tables.

In this section, as if we used runtime score and coin method, below those functionality handle the score and coin saving.

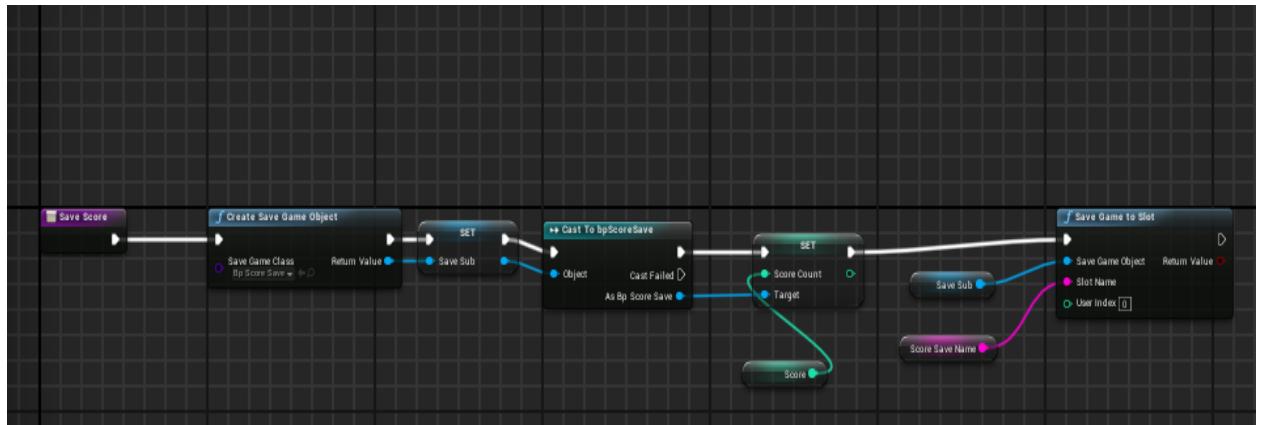


Table 3-1: Score Save

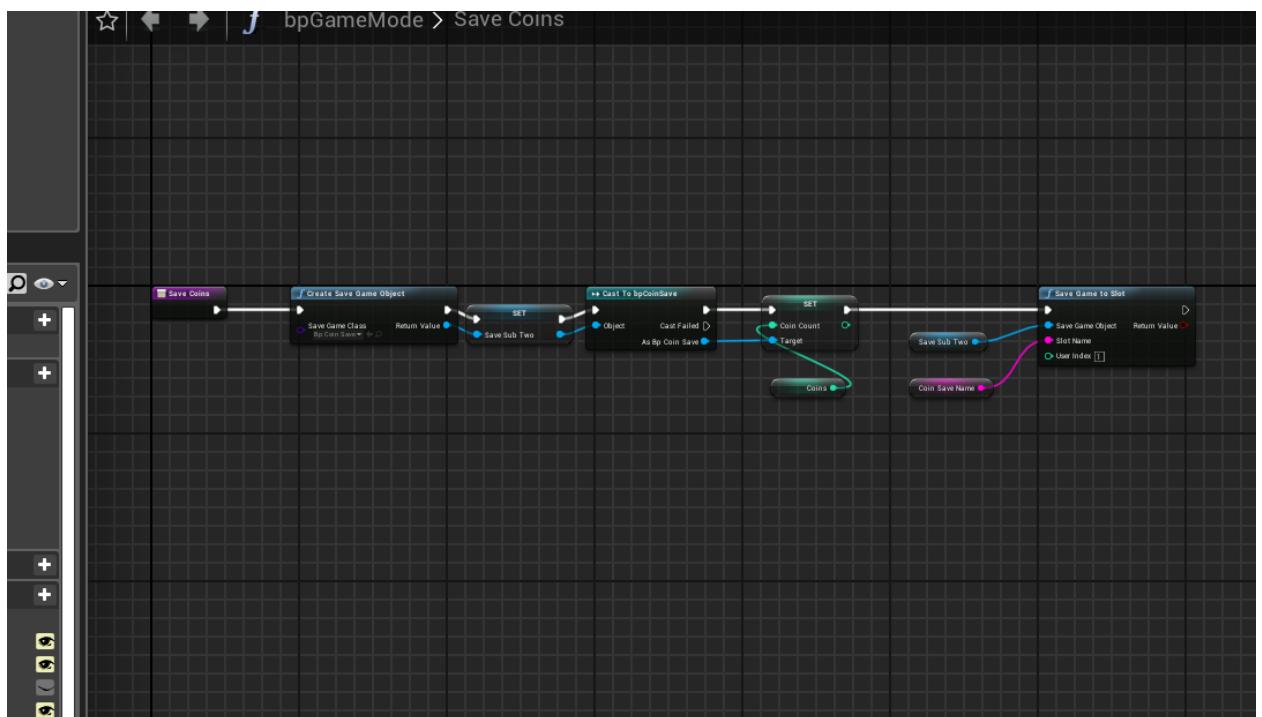


Table 3-2: Coin Save

When admin permits mutual accounts to connect, the data will be moved in this table. This table is also linked with profile information table. This table is giving us the information of the people that has already number exchanged.

3.6 Implementation

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users is that it will work efficiently and effectively. The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system. The more complex the system is being implemented, the more involved will be the systems analysis and design effort required just for implementation. The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed.

Our proposal is an offline system that handles games with security, privacy and accountability. In a specific implementation, our proposal could be a pure online multiplayer system or be offline single player games system. We discuss general ideas about an implementation regarding online multiplayer in a later section.

With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

This project involved the implementation of offline games at system that has a user and also has multiple enemy ships. During this project, past experiences in the field of offline games were documented. In fall of last year, the games system developed in the project using renowned cryptologists underwent a security review.

3.6.1 Front End

Unreal Frontend (UFE) is a tool intended to simplify and speed up daily video game development and testing tasks, such as preparing game builds, deploying them to a device, and launching them. UFE is designed to be the central interface for all game deployment, profiling, and testing tasks. Launcher Build, cook, deploy and launch your game. UFE allows you to simultaneously deploy to multiple target devices running different target platforms, either locally connected or anywhere on the network. You will be able to launch

multiple instances per device (on supporting platforms) and configure different roles per instance in order to drastically speed up your workflow for testing your single or multiplayer cross-platform games. Sessions remotely monitor and interact with any of your currently active game sessions running on the network. The new Session Browser will allow you to locate instances of your game currently running on the network. The Session Console shows you a real-time feed of each game instance's output log, and it allows you to remotely execute console commands on one or more target devices. It will soon be possible to share your active sessions remotely with other developers or artists in order to get a second opinion without them having to come to your desk. Automation automatically test code and content and verify their integrity. This is a new feature not available in previous versions of the engine. You will be able to write unit/feature/stress tests for your game's custom C++ classes and content, and you can create automated tests for verifying common workflows, such as verifying that all maps and packages are loading without errors. We are also working on integrating automation tests into our build system, so that it will be possible to run them automatically and detect potential problems as part of the continuous integration process. Profiling remotely analyze the performance of your game and locate bottlenecks. If you are familiar with our previous suite of profiling tools in UE3, you will be excited to hear that we are working on integrating all these disparate tools into a single user experience inside UFE. Device Management Manage your devkits, mobile devices and other test computers. Simple device discovery was already available in previous versions of UFE, but in UE4 it will be a lot more powerful. We are dropping the restriction that a target device has to be connected to the local PC, either physically or through the device's SDK. You will be able to detect locally connected and networked devices and share them with others. Windows and Mac computers are now also able to act as target devices for deployment, launching, and automation. Unreal Frontend is available as both a tab for Unreal Editor and a standalone application. While their functionality is equivalent, the lightweight standalone application is likely the preferred tool for QA personnel and programmers. Artists may prefer the in-editor version, because they already have Unreal Editor opened most of the day.

3.6.2 Back End

3.6.2.1 Database Design for Table

In this section, we will discuss about database which was used in our project. As if our space shooter games is under development. So this case we show up run time data only in this game. In future we will add firebase or SQL database.

Table 3-3: Player

SL NO.	Name	Type	Allow Nulls	Description
1	<u>Id</u>	Int	No	Primary Key
2	<u>Coin Id</u>	Int	No	
3	Score Id	Int	No	

Table 3-4: Health

SL NO.	Name	Type	Allow Nulls	Description
1	<u>Id</u>	Int	No	Primary Key
2	<u>Max</u>	Int	No	
3	Min	Int	No	
4	Plr_Id	Int	No	Foreign Key
5	Enm_OneId	Int	No	
6	Enm_TwoId	Int	No	
7	Boss_Id	Int	No	

Table 3-5: Enemy1, Enemy2, Enemy Boss

SL NO.	Name	Type	Allow Nulls	Description
1	<u>Id</u>	Int	No	Primary Key
2	Damaged_id	Int	No	

Table 3-6: Coin

SL NO.	Name	Type	Allow Nulls	Description
1	Id	Int	No	Primary Key
2	TotalCoin_id	Int	No	

Table 3-7: Score

SL NO.	Name	Type	Allow Nulls	Description
1	Id	Int	No	Primary Key
2	TotalScore_id	Int	No	

3.7 Results Analysis

In a game that focuses on shooting down enemies, or avoiding being shot down, a system to handle collision is vital. This section defines colliders, a way to detect colliding objects, our perceptive behind collision in a space-shooter game, results from the game, as well as a discussion about what could have been done differently. Collider components state the shape of an object for the purposes of physical collisions. A collider, which is invisible, need not be the exact same shape as the objects mesh and in fact, a rough estimate is often more efficient and indistinguishable in game play. The simple (and least processor-intensive) colliders the so-called primitive collider types. In 3D, there are the three types collider like Box Collider, Sphere Collider and Capsule Collider. In 3D technique, we can use the Box Collider 3D.

- **Lighting-effect:** There are many ways to represent light sources, common ones being point lights, directional lights, spotlights, and area lights. Both the spotlight, and the directional light sources have defined directions, so when the light source is in the middle of the plane, as in our game, they are not very suitable; an omnidirectional light source would be preferred. This leave the point light source and area light sources as likely candidates. A point light source is a light source where all the light comes from an infinitely at a small point. This approximates lighting from a light source that comes from an infinitely either very small, or far away.

However, when the light source covers a larger area, a point light source might be too imprecise, as it will not illuminate the surrounding objects as well as a large light actually would (see Figure 3). An area light source would solve this issue by simulating the larger area, commonly using multiple point light sources, which results in more proper lighting depend on the cost of computational power. One way to avoid this issue is to, for every fragment of the object on which to apply lighting, choose the best point of the light source, and use only this point, as a point light, for calculating the light.

- **Dynamics of Particle System:** Each particle has a predetermined lifetime, typically of a few times, during which it can undergo various changes. Dynamic particle system begins its life when it is generated or emitted by its particle system. The system emits particles at random positions within an area of space shaped like a sphere, hemisphere, cone, box or any arbitrary mesh. The particle is displayed until its time is out, at which point it is removed from the system. The system's emission rate indicates how many particles are emitted per second, although the exact times of emission are randomized slightly. The choice of emission rate and average particle lifetime calculate the number of particles in
- **Speed:** player (5000) has boundary that it can move up and down also straight right and left. Player has bullet every click will emit a bullet until get power up. When player gets power up. In red cube when touch player will get 5 bullets at a time. User should click right button of mouse. As same as red cube, the blue cube has rocket power, after getting touch every click 2 rocket will emit. Enemy one speed(-2000), enemy2 speed(-2200).
- **Coin Counter:** There will be Coin here and there in the scene for the player which he needs to collect. Coin Collection is used for the measurement of game coin how gamer earn. The more the player collects Coin, the more points he gets. Player notices coin in a certain place He moves plane to the place plane automatically collects coin when gets in touch of it. Calculate points depending on total coin collected.
- **Power up Red Cube:** There will be power here and there in the scene for the player which he needs to collect. Red Cube, Collection is used for the increasing the player

bullet. The player collects Red, the more power he/she gets. Player notices Red Cube in a certain place. He moves plane to the place. Plane automatically collects Red Cube when gets in touch of it. Automatically power will be up, press right button for throwing power up bullets.

- **Power up Blue Cube:** There will be power here and there in the scene for the player which he needs to collect. Blue Cube, Collection is used for the increasing the player bullet damage point power. The player collects blue cube, the more power he/she gets. Player notices Blue Cube in a certain place. He moves plane to the place. Plane automatically collects Blue Cube when gets in touch of it. Automatically power will be up, press right button for throwing power up bullets.
- **Accuracy:** It is quite impossible to make error free in the gaming system where people are involved using from more than 24 hours. People are directly involved in these process are tired and could not perform efficiently which make mistakes and errors in results.

Currently some gaming systems are already working. In my interview it was also part of discussion which one can solve the problem and fulfill the requirements. To validate the literature, we discussed online gaming system is covering most of the requirements;

3.8 Conclusions

This chapter deals with the details of the technologies we used to develop our project. We needed a hardware configuration with core i3 processor, 8 GB of RAM, 20 GB hard disk allocation. We also needed a software configuration of windows 7, 64-bit operating system, Android Studio to implement the application. We also used net beans IDE for the coding purpose. And the way we used these technologies are briefly described above.

We have discussed in this chapter about the feasibility study of online voting system. Apart from the requirement analysis, system design, front end and back implementations are described in this chapter. In system design the entity relationship diagram and data flow diagram are described. Moreover, database designs are described in back-end implementation. In requirement analysis here discussed about the functional and nonfunctional requirement of our system. Moreover, here

Discussed how we implemented our system. That's all about the proposed system chapter of our project.

Chapter- 4

User Manual

4.1 Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement. Testing can also improve overall security, but testing is not a simple process. Each day, there will be difficult challenges that involve coding and decoding. The testing process is an important phase during the software development because each small module must be tested to ensure its accuracy and validity[49].

In this chapter we have analyzed our system. We described the testing process in software development. By testing we can see the output of the different types of input by the user. By analyzing we have understood that out all the modules are working. An admin can handle our system by logging in successfully.

This chapter also describes the report displayed to admin only. All reports will give the idea of how reports will be displayed.

The objective of this project is to develop Designing and implementation Desktop based “Space Shooter Games”. Right now no need using any web server. if you are already downloaded games from web. Just install it on your pc and play smoothly.

4.2 Testing

Testing is the process of exercising software with the intent of finding of errors. Web app testing is a collection of related activities with a single goal: to uncover errors in web app content, function, usability, navigability, performance, capacity, and security.

Web based systems and applications reside on network and interoperate with many different

- Operating systems,

- Browsers,
- Hardware platforms,
- Communications protocols

4.3 Testing Methods

Testing presents an interesting anomaly for the software engineering activities, the engineer attempts to build software from an abstract concept to a tangible product. Now comes testing. The engineer creates a series of test case that are initiated to "demolish" the software that has been build. In fact, testing is the one step in the software process that could be viewed (psychologically, at least) as destructive rather than constructive.

4.3.1 Models of Testing

On the basis of testing methods there are two types of testing:

- Black-box testing.
- White-box testing.

Black-box tests are used to demonstrate that software functions are operational, that input is properly accepted and output is correctly produced, and that integrity of external information is maintained.

White-box tests are used to examine the procedural details. It checks the logical paths by test case. It can also check the conditions, loops used in the software coding. It checks that loops are working correctly on defined boundary value.

4.3.1.1 White-Box Testing

White-box testing sometimes called glass-box testing, is a test case design method that uses the control structure of the procedural design to drive the test case. Always we are thinking that there is no necessary to execute or checks the loops and conditions. And so large number of errors is uncovered. With using white-box testing methods, we have checked that; all independent paths within a function have been executed at least once. All logical decisions on their true and false side. All loops working correctly at their boundary values and within their specified conditions.

In our coding we test that all the loops work truly in each module. The one technique of white-box testing is basis path testing. It contains two parts, one is flow graph notation and the second is cyclometer complexity. In flow graph notation we are checking logical control of flow. By using cyclometer complexity, we find complexity of our project structure.

4.3.1.2 Black-Box Testing

Black-box testing focuses on the functional requirements of the software. That is black- box testing enables the software engineer to drive sets of input conditions that will fully exercise all functional Requirements for the program. Black-box testing is not an alternative to white-box testing techniques. Rather, it is a complementary approach that is likely to uncover a different class of errors than white-box methods. We use in our coding to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in database
- Performance errors
- Initialization and termination errors.

Unlike white-box testing, which is performed earlier in the testing process, black-box testing tends to be applied during later stages of testing. Because black-box testing purposely disregards control structure, attention is focused on the information domain. By applying black-box techniques, we derive a set of test cases that satisfy following criteria.

4.4 Test Cases

Once the coding is complete the system has to be thoroughly tested. Unit testing involves the testing of the game's modules. System testing aims to test the games as a whole and ensure its functionality for the user.

Table 4-4-1: Animation Testing

Test Case	This test will check if the animation is working correctly.
Test Procedure	Import a character model with animation in unity. Place character on the scene. Run the game.
Expected Result	Animation works perfectly in the environment
Actual Result	Animation is not working
Comment	Need to check character configuration on inspector window. The appropriate animation was not selected. Select it
Conditional Test	Again run scene.
Expected Result	Animation is working now
Actual Result	Yes, it is working.
Accuracy	Accuracy depends on hardware configuration

Table 4-4-2: Interaction between objects

Test Case	This test will check if the interaction between objects is working correctly
Test Procedure	Add scripts of interaction in the objects that we want to interact with each other. Run scene.
Expected Result	Objects are interacting
Actual Result	Run time exception
Comment	Need to add checking in the scripts for the objects that have a particular script.
Conditional Test	Run scene.
Expected Result	Interaction is ok now
Actual Result	Interaction is ok now
Accuracy	Perfectly accurate

Table 4-4-3: UI Testing

Test Case	This test will check if the dialogue box is working
Test Procedure	Add dialogue box in the scene. Run scene.
Expected Result	Dialogue box appears in the correct dimension.
Actual Result	Working perfectly
Comment	Tips and dialogues are working as expected.

Table 4-4-4: Health and Coin

Test Case	This test will check if the automatic health decreasing when player collide with enemy bullets and also coin are increasing when player collide coin ring.
Test Procedure	Set health 100% and check decreasing or not when collide with bullet of enemy. And also coin are increased or not when collide with coins come from the front side.
Expected Result	Health decreasing and coin increasing depending on collision with aspect object.
Actual Result	Working perfectly
Comment	Health working correctly it's make sense to play properly and getting coin it's means how much you are profitable plying this game.

4.5 Requirement Analysis

System Requirements are expressed in a software requirement document. The software requirement specification (SRS) is the official statement of what is required of the system developers. This requirement document includes the requirements definition and the requirement specification. The software requirement document is not a design document.

It should set out what the system should do without specifying how it should be done. The requirement set out in this document is complete and consistent.

The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually Architecturally Significant Requirements. Broadly, functional requirements define what a system is supposed to do and non-functional requirements define how a system is supposed.

4.5.1 Non-Functional Requirement

The major non-functional requirements are describing below:

- **Performance**

The system must be interactive and the delays involved must be less. So in every action response of the system, there are no immediate delays. In case of opening windows forms, of popping error messages and saving the settings or sessions there is delay much below 2 seconds, in case of opening databases, sorting questions and evaluation there are no delays and the operation is performed in less than 2 seconds for opening, sorting, computing, posting > 95% of the files.

- **Safety**

Information transmission should be securely transmitted to server without any changes in information

- **Reliability**

As the system provides the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.

- **Availability**

If the internet service gets disrupted while sending information to the server, the information can be sending again for verification.

- **Security**

The main security concern is for users account hence proper login mechanism should be used to avoid hacking. The tablet id registration is way to spam check for increasing the security.

4.5.2 Functional Requirement

A functional requirement defines a function of a system or its component. These functional requirements are needed to access the system.

- **Play Activity**

Play Activity is a simple and very interesting game at the same time. The game does not require any special skill. All you need is common sense and a sense of humor. The base of the game is explaining words and guessing of those words. Click play button and get the entry of game world.

- **Replay Activity**

Anyhow if player dead by the enemy shoot, there will be option appear that you want to play again. Actually it giving another chance to play games.

- **Level Activity (one to four)**

Every games has multiple level. In our games we have been created four level, a user can pass another level through the killing boss of the enemy. After passing same procedure will repeated until level 4.if user completed all level by killing enemy in the level 4, user will declared as a winner. And option will appear quit or replay.

Accidently if any user dead any level at that time games will restart from level one.

4.5.3 Software Requirement

The software is a set of procedures of coded information or a program when fed into the computer hardware enables the computer to perform various tasks. Software is like a current inside the wire, which cannot be seen but its effect can be felt.

- **Operating System:** Windows7/8/10
- **Development Language:** blueprint, C#
- **Database Server:** No
- **Framework:** Unreal Games Engine v4.21.2
- **Front End:** Blander V2.80, adobe Photoshop cs6, engine V4.21.2
- **Back End:** Blueprint run time database. Real time database is under development.

4.5.4 Hardware Requirement

The following are the minimum hardware specifications to run this package:

- **Operating System:** Windows 7, 8, 8.1, 10
- **Processor:** Intel 7th Generation Pentium Processor G4560((3M Cache, 3.50 GHz), Intel 8th Generation Core i3-8100 Processor.
- **ROM:** Western digital WD2003FYPS desktop hard disk drive has 2 TB storage capacity, 64 MB cache.
- **RAM:** Corsair Vengeance® LPX 4GB (1x4GB) DDR4 DRAM 2400MHz.
- **Display:** Samsung 20" wide-screen 1440 X 900px res LCD-TFT
- **System Type:** 32/64-bit Operating System

4.6 Result Analysis

In this chapter we have analyzed our system. By analyzing we can see the output of the different types of report such as registration information, profile information. By analyzing we have understood that out all the modules are working. An admin can handle our system by logged in successfully. Some of the reports are describing as follows.

4.6.1 Flow Chart

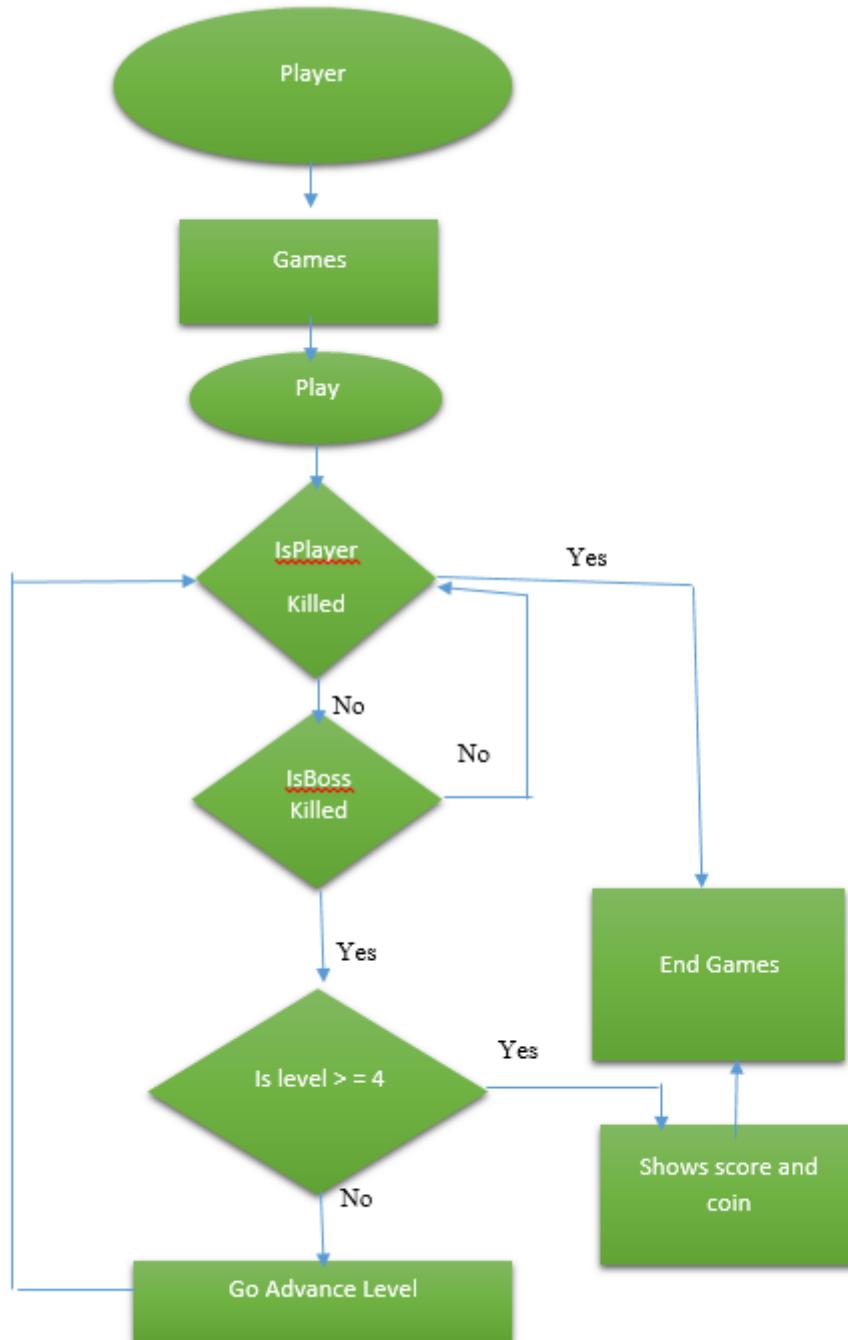


Figure 4-1: Flow Chart

4.7 Graphical Interface

4.7.1 Level One

A Gamer first interact with system UI to start playing. We provide play and quit button to all users so that he/she can easily understand about the playing procedure.

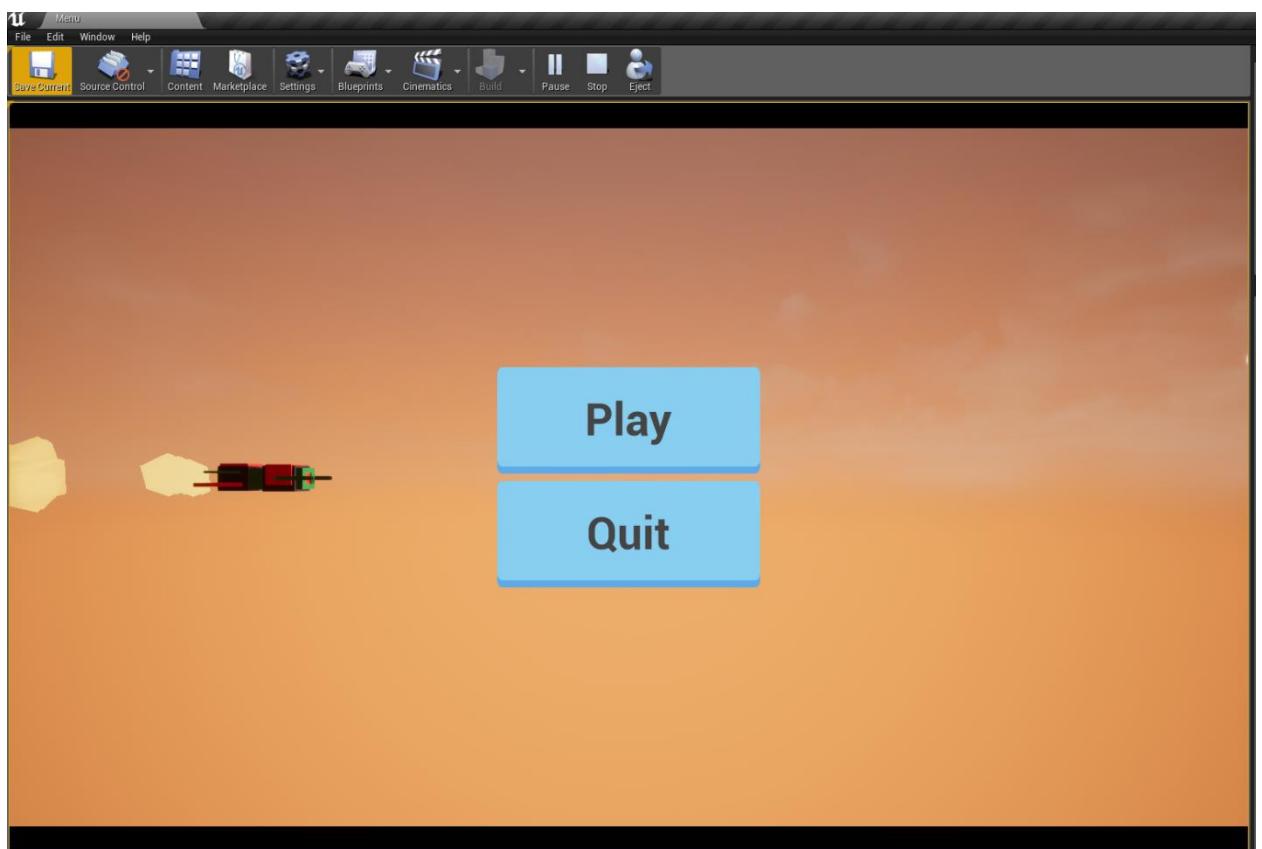


Figure 4-2: Game Starting Menu

At the beginning of the game will appear two options, one is play button and another one is quit. Play button for starting the game and also quit button is exit from the game.

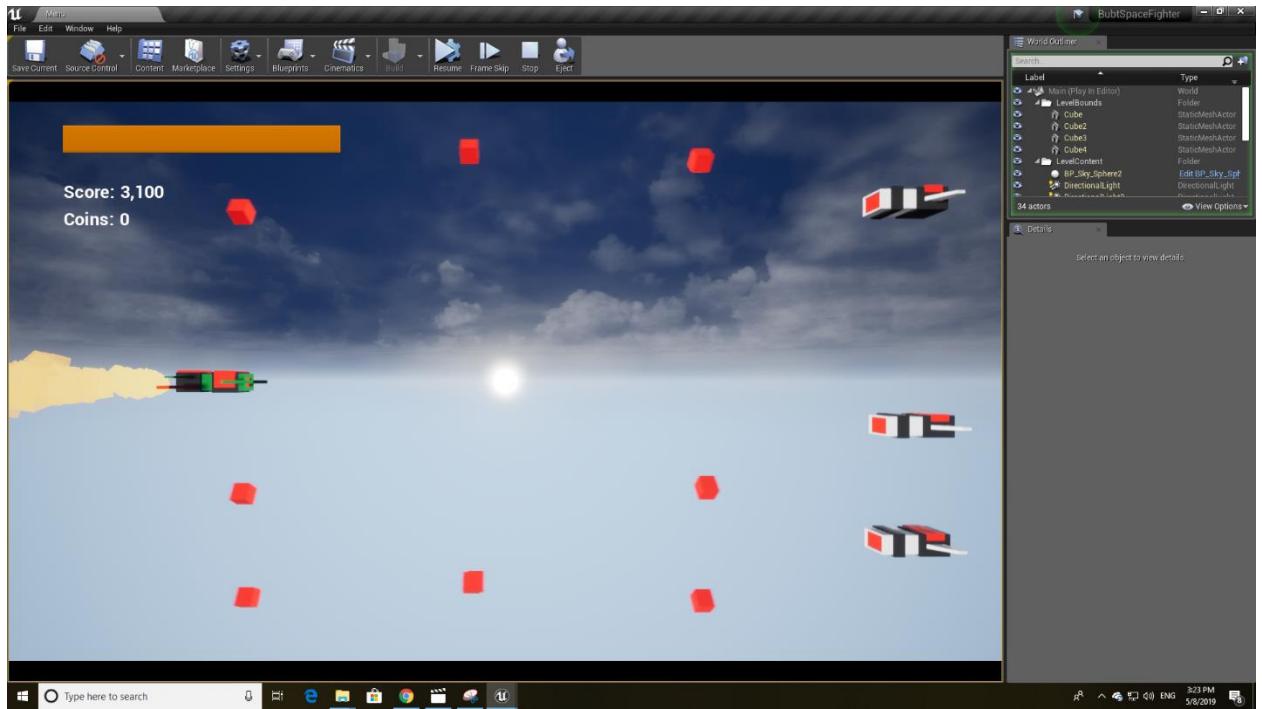


Figure 4-3: level-1 Enemy ships First Step

In the First step of level1, there will be appear three enemy (type one). Player has health 100, also enemy has health 100, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (5) damage point is 50. Besides of those enemy one bullet (1) damage point is 10.

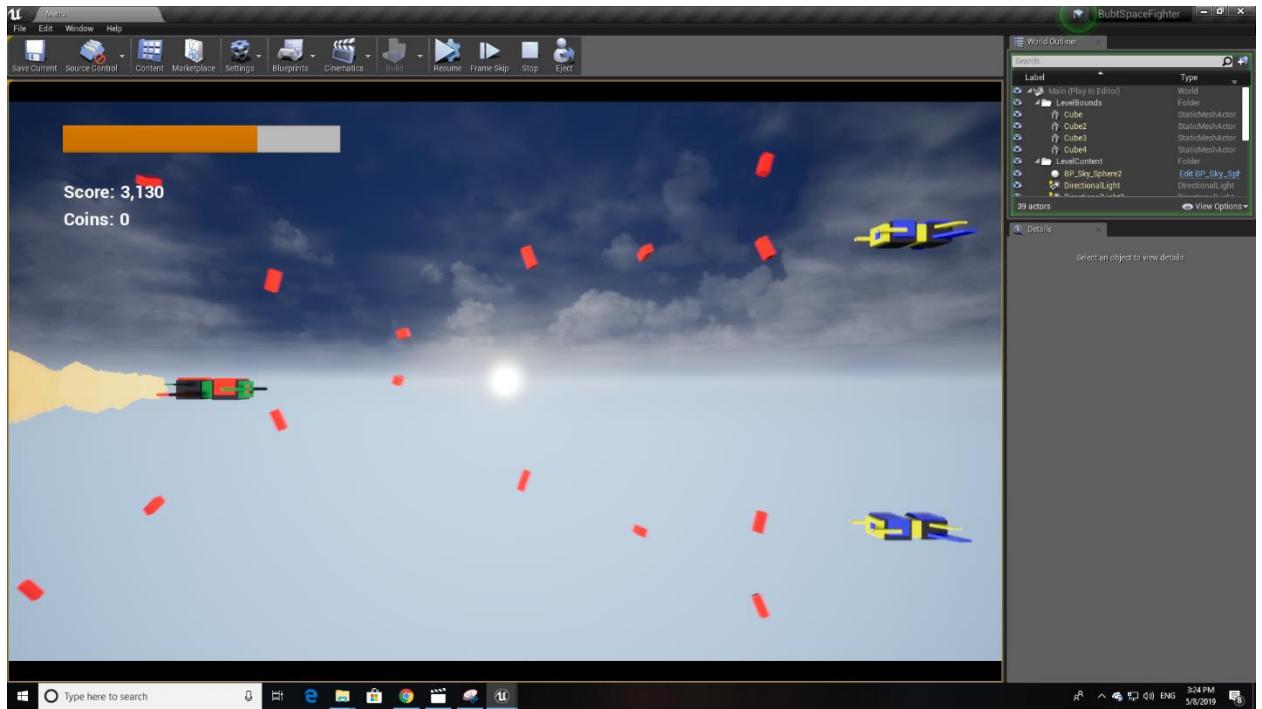


Figure 4-4: level-1 Enemy ships Second Step

In the Second step of level1, their will appear two enemy type two. Player has health 100, also enemy has health 200, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (5) damage point is 50. Besides of those enemy2 bullet (2) damage point is 20.

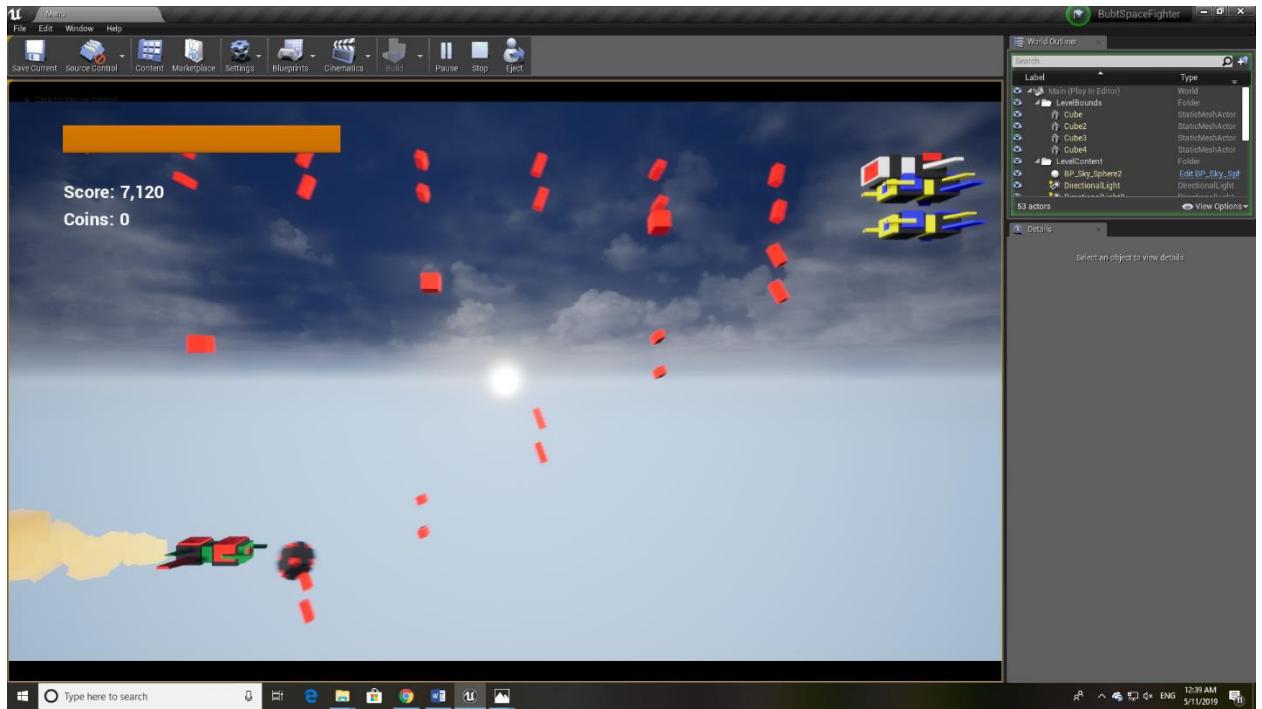


Figure 4-5: Enemy ships Third Step

In the third step of level1, there will appear three enemy type randomly. Player has health 100, also enemy1 has health 100, enemy2 has health 200, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (2) damage point is 50. Besides of those enemy1 bullet (1) damage point is 10 and enemy2 bullet (2) damage point is 20.

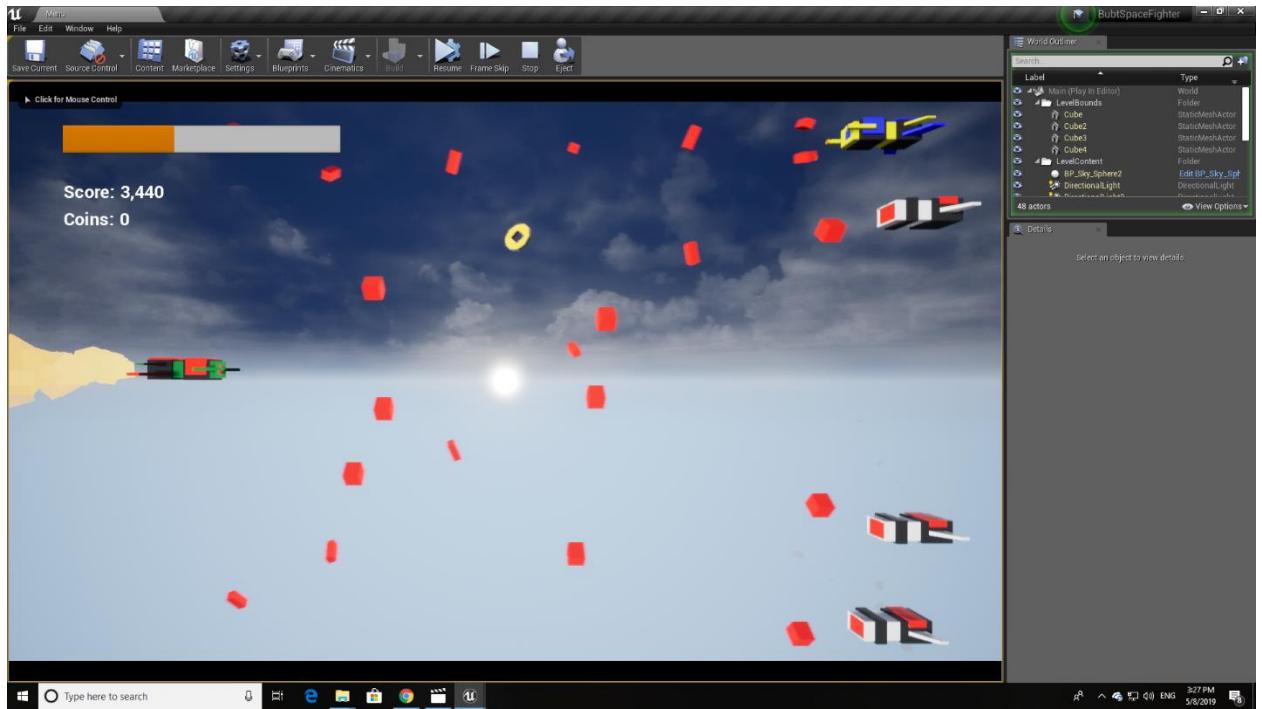


Figure 4-6: level-1 Enemy ships fourth Step

In fourth step of level1, their will appear four enemy type randomly. Player has health 100, also enemy1 has health 100, enemy2 has health 200, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (2) damage point is 50. Besides of those enemy1 bullet (1) damage point is 10 and enemy2 bullet (2) damage point is 20.

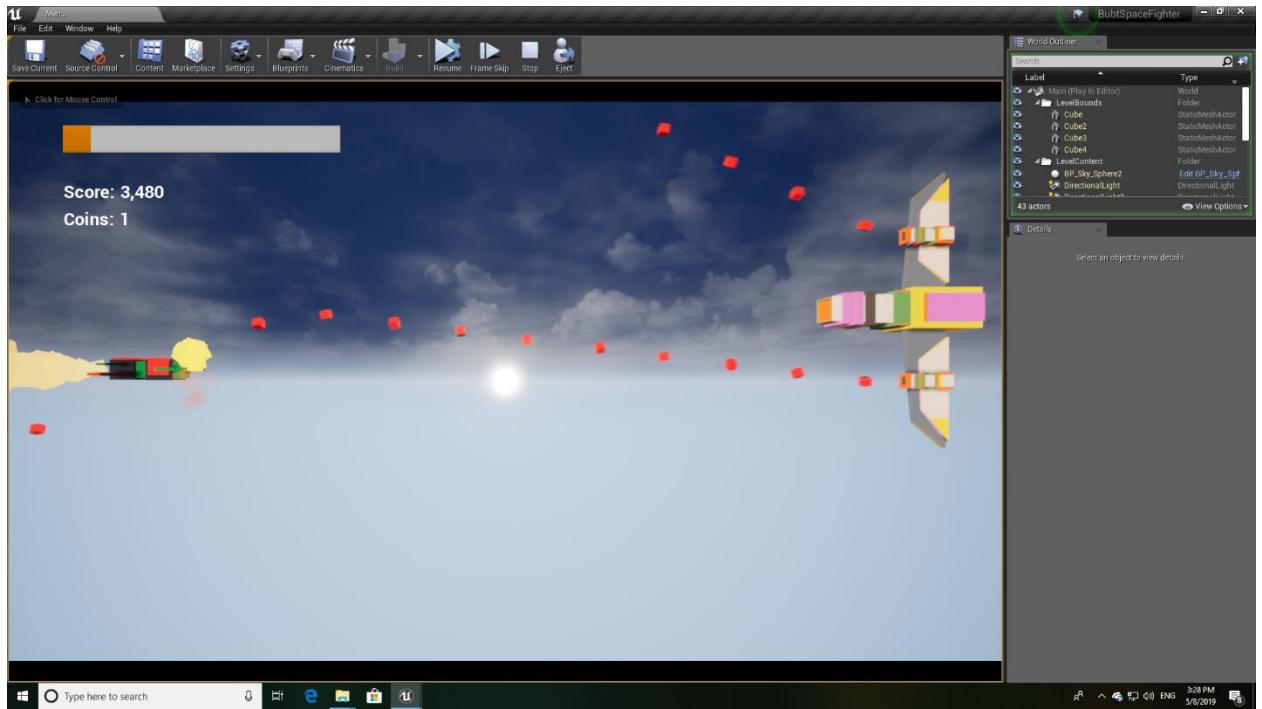


Figure 4-7: level-1 Enemy ships Boss

In the Last step of level1, there will be appear a boss, player health is 100, and boss health is 5000, player has normal bullet damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (5) damage point is 50. Boss bullet (2) damage point 5.

4.7.2 Level Two

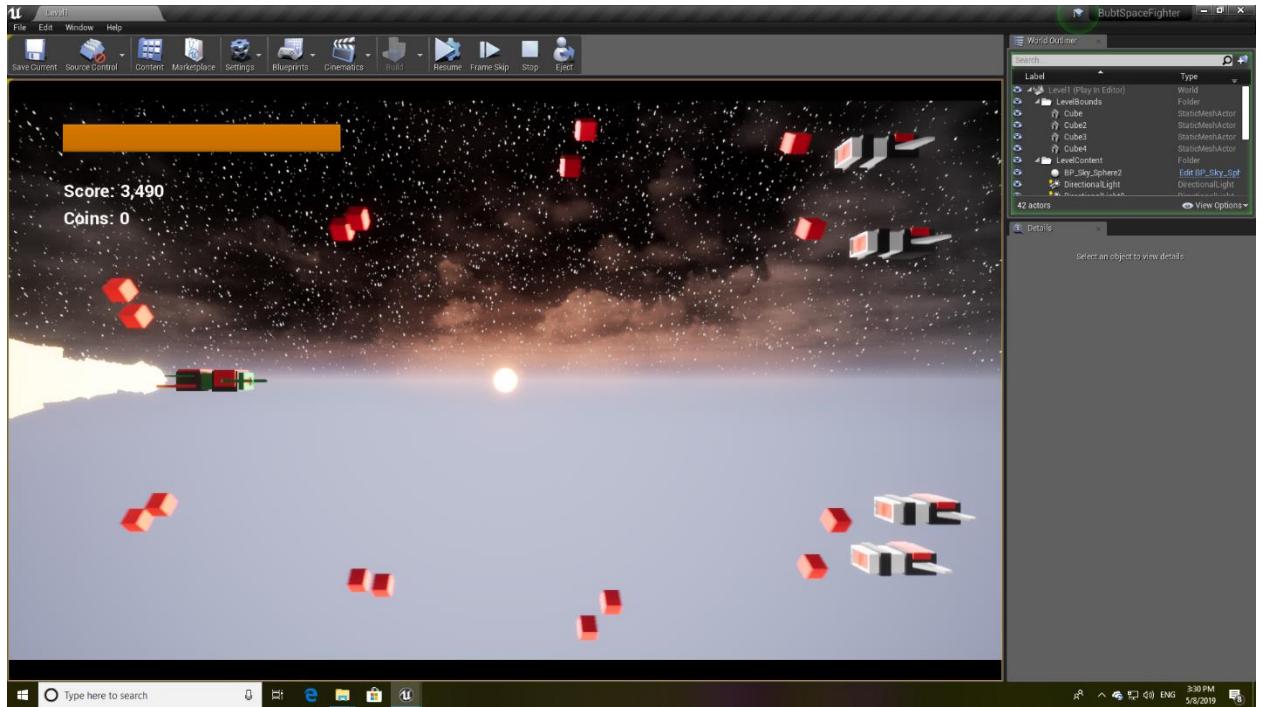


Figure 4-8: level-2 Enemy ships first Step

In the First step of level2, there will be appear four enemy (type one). Player has health 100, also enemy has health 100, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (5) damage point is 50. Besides of those enemy one bullet (1) damage point is 10.

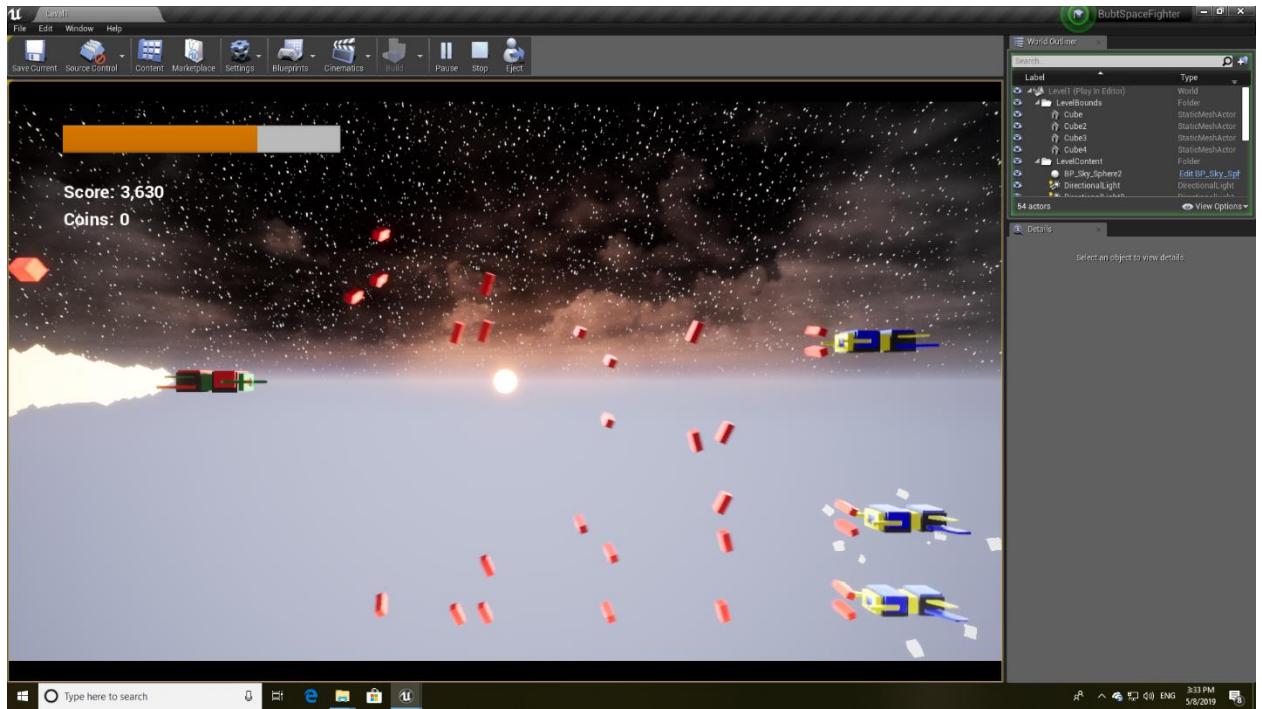


Figure 4-9: level-2 Enemy ships Second Step

In the Second step of level2, there will appear three enemy type two. Player has health 100, also enemy has health 200, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (5) damage point is 50. Besides of those enemy2 bullet (2) damage point is 20.

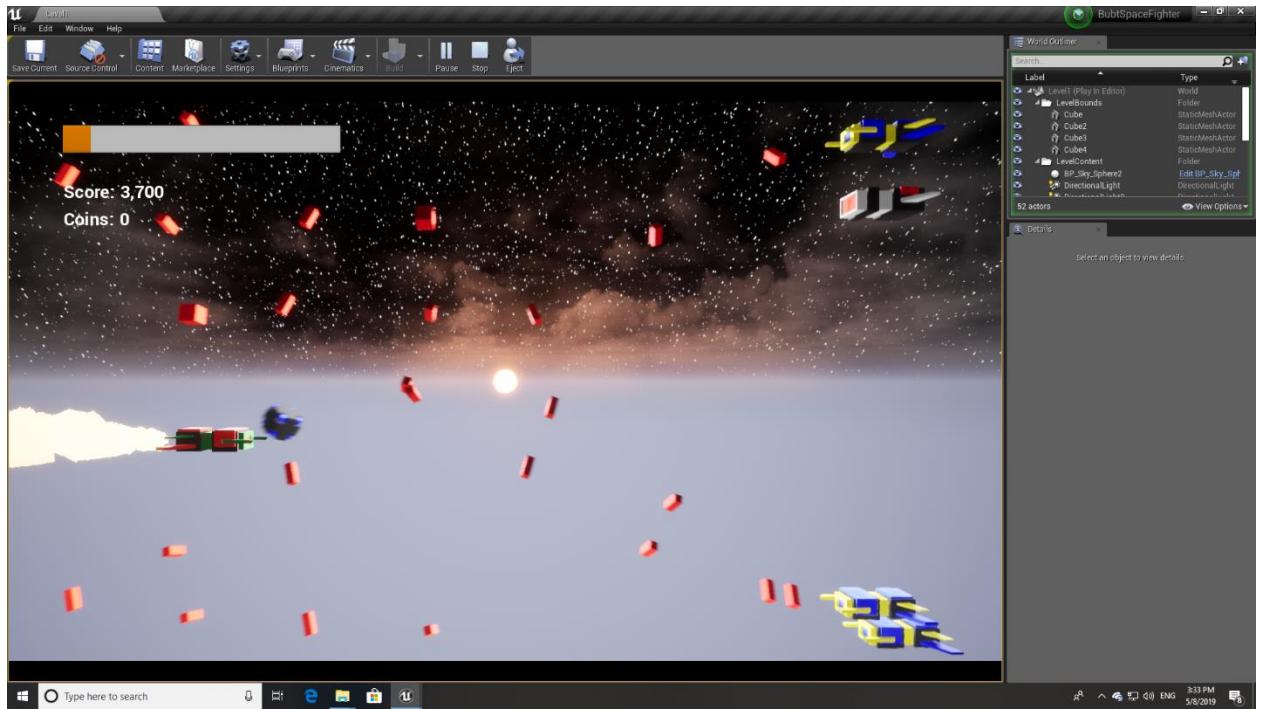


Figure 4-10: level-2 Enemy ships Third Step

In the Third step of level2, their will appear five enemy type randomly. Player has health 100, also enemy1 has health 100, enemy2 has health 200, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (2) damage point is 50. Besides of those enemy1 bullet (1) damage point is 10 and enemy2 bullet (2) damage point is 20.

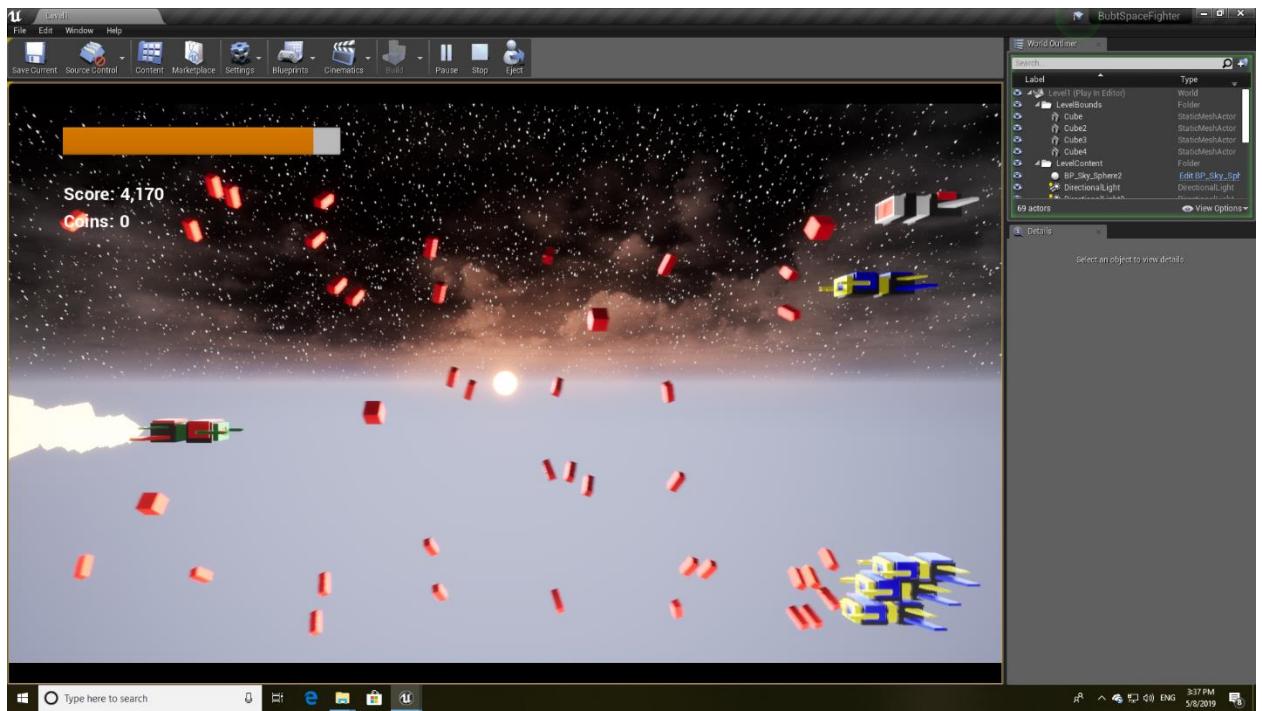


Figure 4-11: level-2 Enemy ships fourth Step

In the fourth step of level2, there will appear five enemy type randomly. Player has health 100, also enemy1 has health 100, enemy2 has health 200, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (2) damage point is 50. Besides of those enemy1 bullet (1) damage point is 10 and enemy2 bullet (2) damage point is 20.

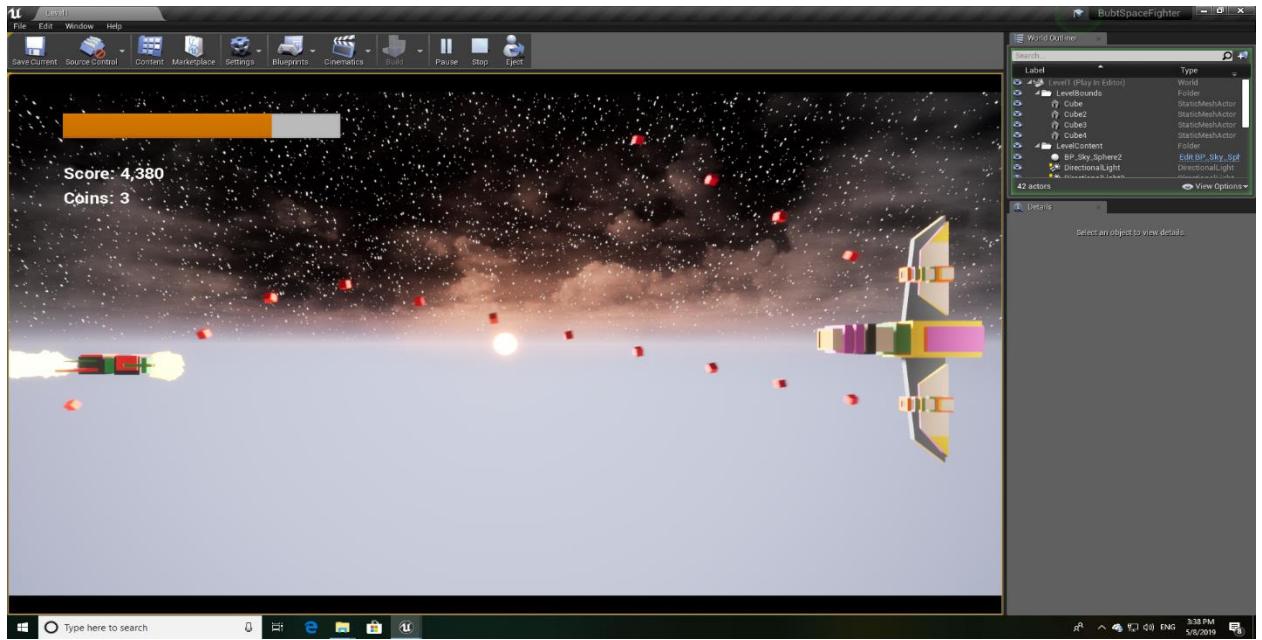


Figure 4-12: level-2 Enemy ships Boss

In the Last step of level2, there will be appear a boss, player health is 100, and boss health is 5000, player has normal bullet damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (5) damage point is 50. Boss bullet (2) damage point 5.

4.7.3 Level Three

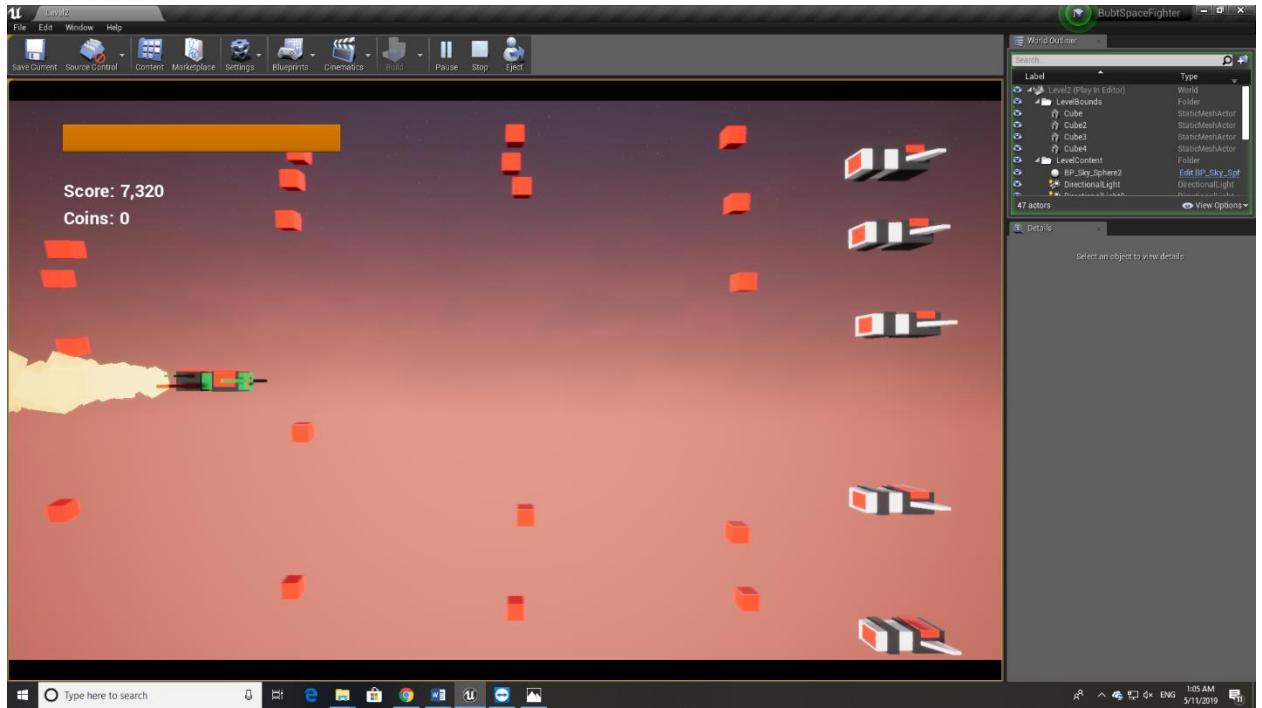


Figure 4-13: level-3 Enemy ships first Step

In the First step of level3, there will be appear five enemy (type one). Player has health 100, also enemy has health 100, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (5) damage point is 50. Besides of those enemy one bullet (1) damage point is 10.

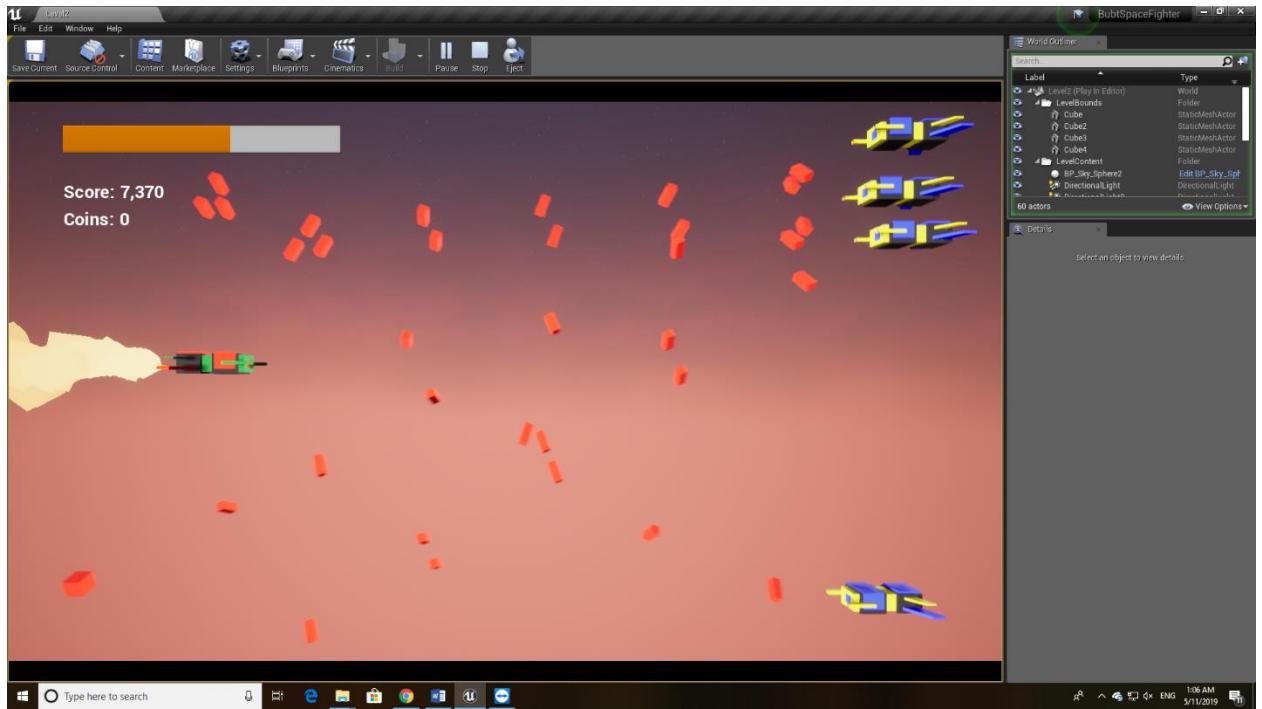


Figure 4-14: level-3 Enemy ships Second Step

In the Second step of level3, their will appear four enemy type two. Player has health 100, also enemy has health 200, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (5) damage point is 50. Besides of those enemy2 bullet (2) damage point is 20.

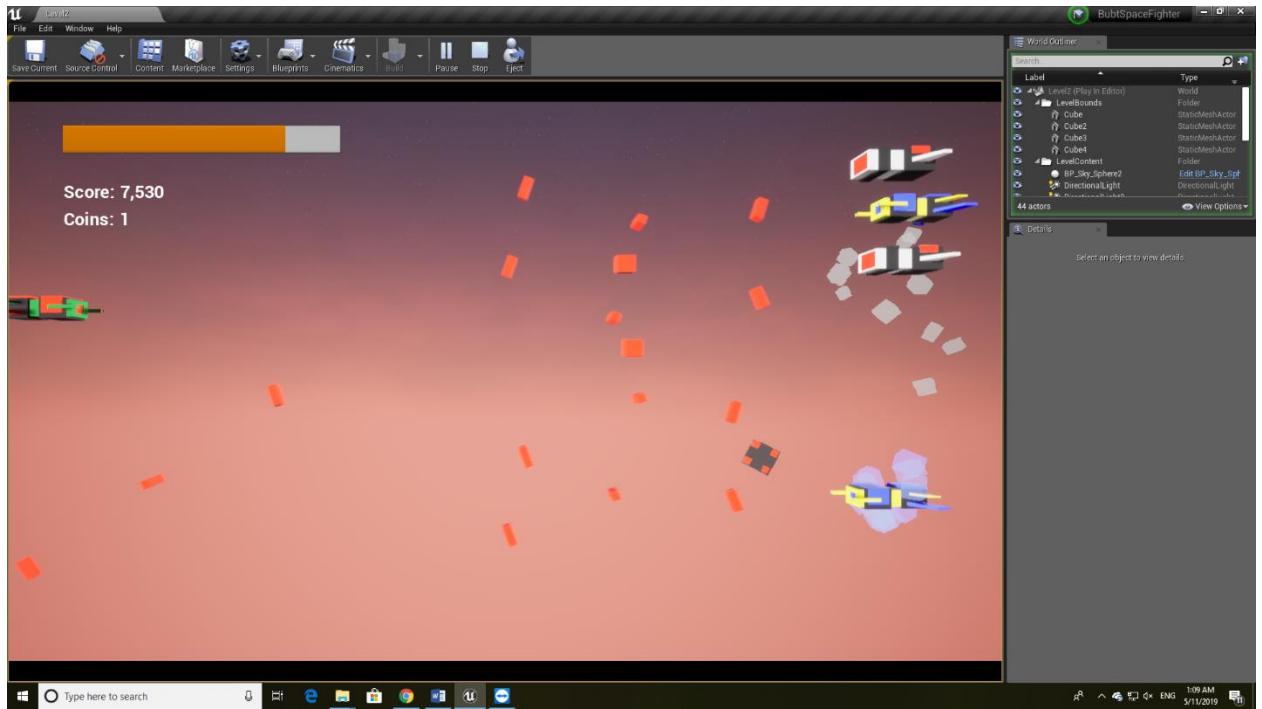


Figure 4-15: level-3 Enemy ships Third Step

In Third step of level3, there will appear five enemy type randomly. Player has health 100, also enemy1 has health 100, enemy2 has health 200, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (2) damage point is 50. Besides of those enemy1 bullet (1) damage point is 10 and enemy2 bullet (2) damage point is 20.

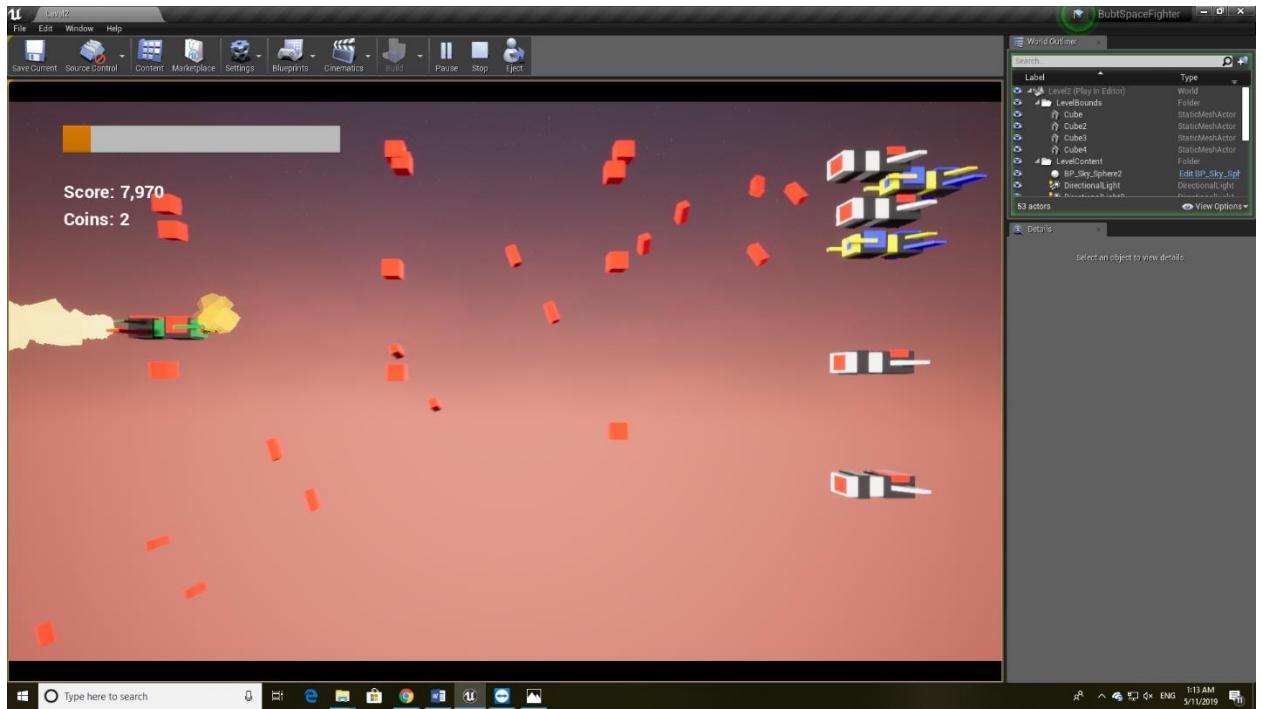


Figure 4-16: level-3 Enemy ships Fourth Step

In fourth step of level3, there will appear six enemy type randomly. Player has health 100, also enemy1 has health 100, enemy2 has health 200, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (2) damage point is 50. Besides of those enemy1 bullet (1) damage point is 10 and enemy2 bullet (2) damage point is 20.

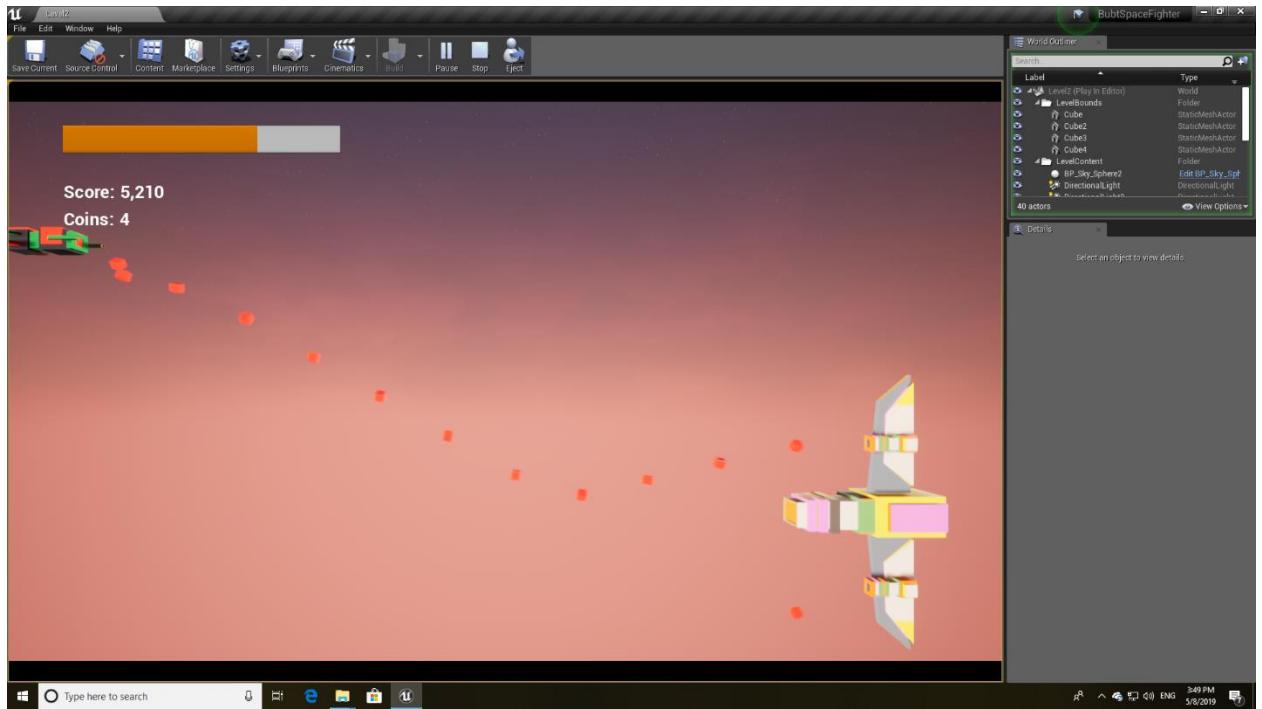


Figure 4-17: level-3 Enemy ships Boss

In the Last step of level3, there will be appear a boss, player health is 100, and boss health is 5000, player has normal bullet damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (5) damage point is 50. Boss bullet (2) damage point 5.

4.7.4 Level Four

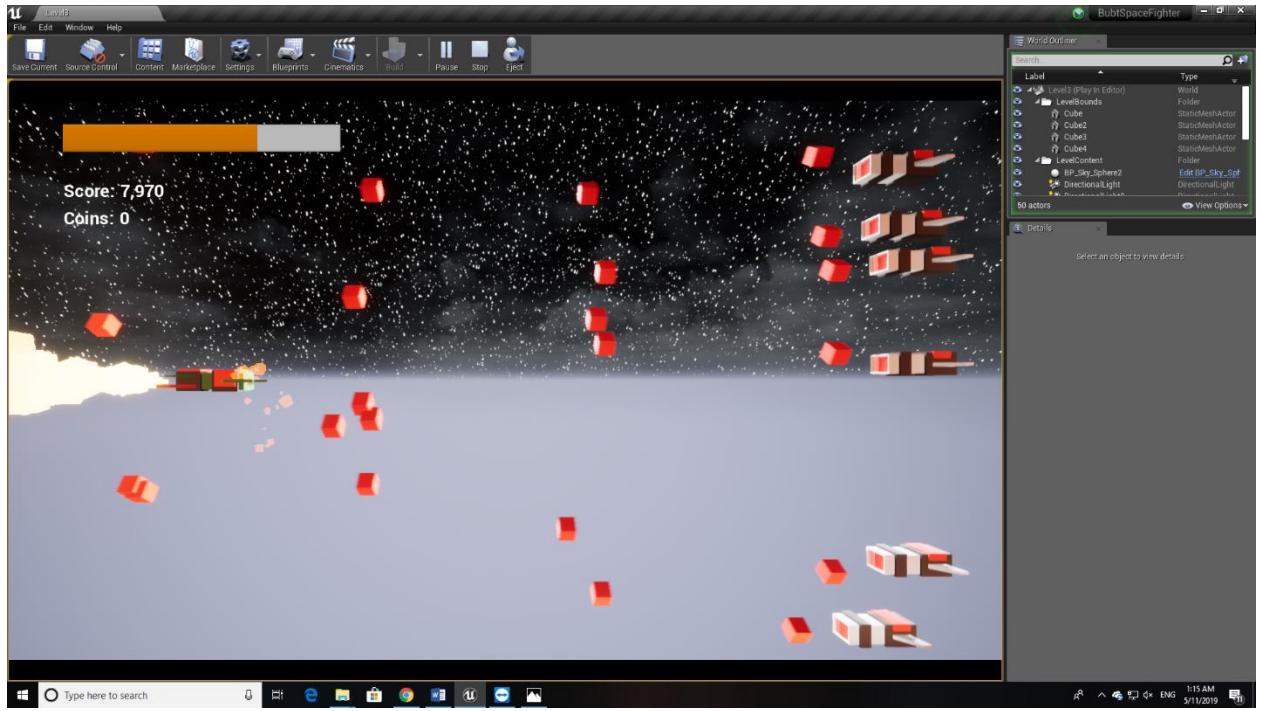


Figure 4-18: level-4 Enemy ships first Step

In the First step of level4, there will be appear six enemy (type one). Player has health 100, also enemy has health 100, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (5) damage point is 50. Besides of those enemy one bullet (1) damage point is 10.

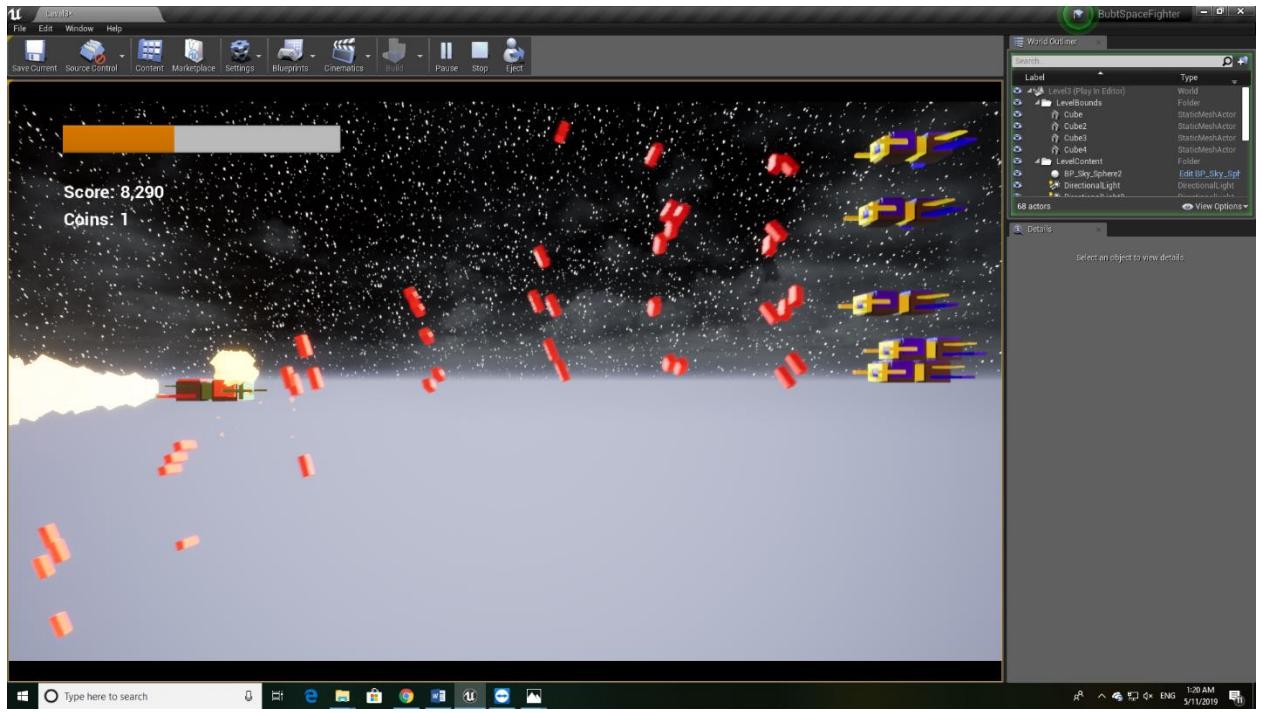


Figure 4-19: level-4 Enemy ships Second Step

In Second step of level4, their will appear five enemy type two. Player has health 100, also enemy has health 200, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (5) damage point is 50. Besides of those enemy2 bullet (2) damage point is 20.

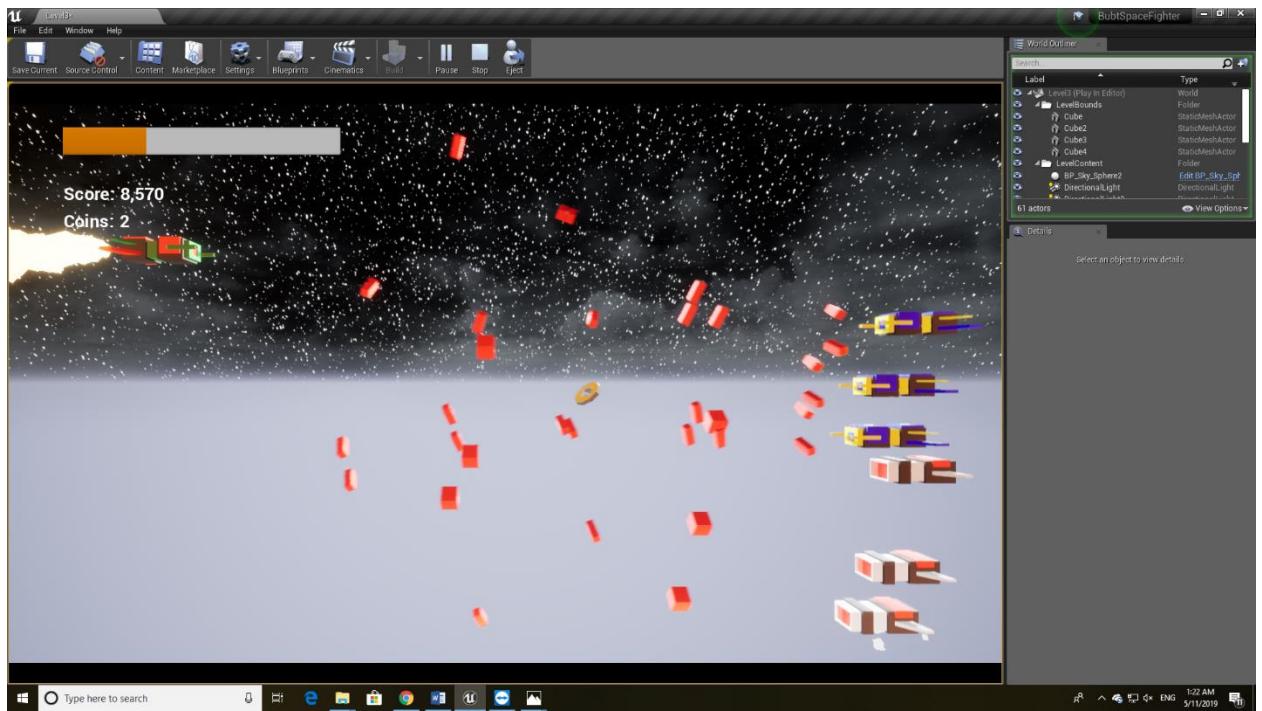


Figure 4-20: level-4 Enemy ships Third Step

In the Third step of level4, there will appear six enemy type randomly. Player has health 100, also enemy1 has health 100, enemy2 has health 200, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (2) damage point is 50. Besides of those enemy1 bullet (1) damage point is 10 and enemy2 bullet (2) damage point is 20.

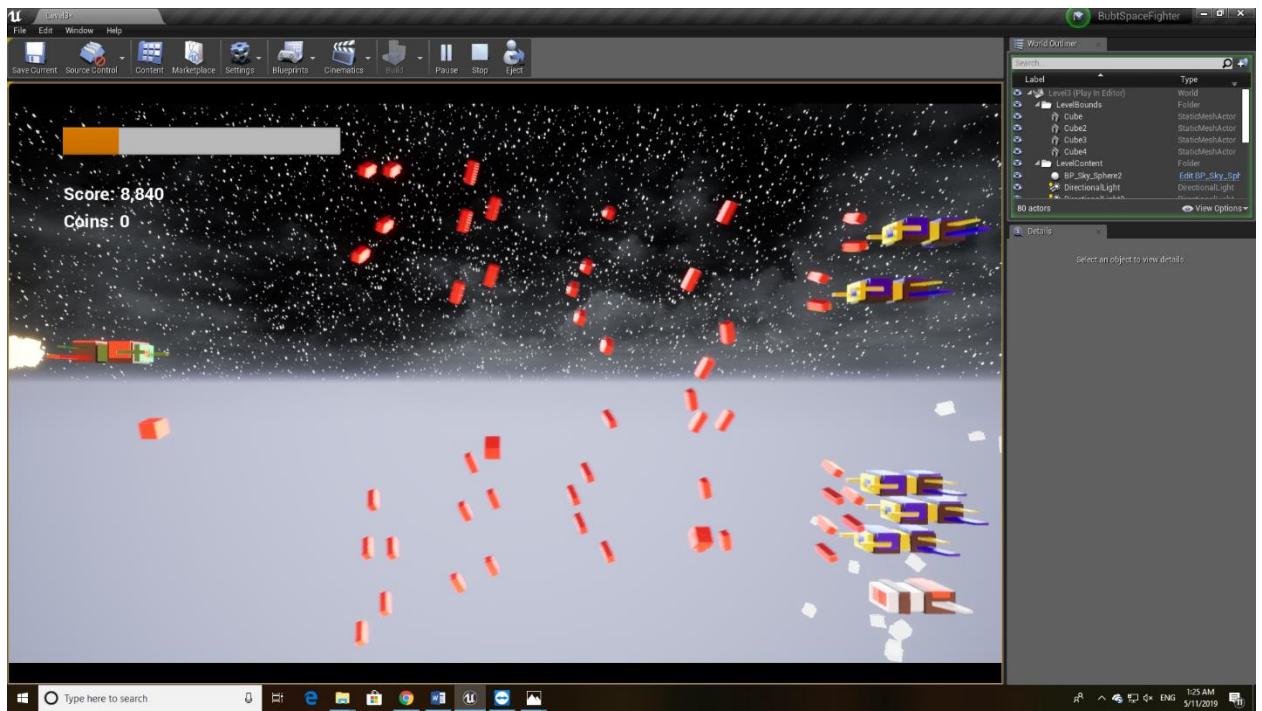


Figure 4-21: level-4 Enemy ships Fourth Step

In fourth step of level4, there will appear seven enemy type randomly. Player has health 100, also enemy1 has health 100, enemy2 has health 200, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (2) damage point is 50. Besides of those enemy1 bullet (1) damage point is 10 and enemy2 bullet (2) damage point is 20.

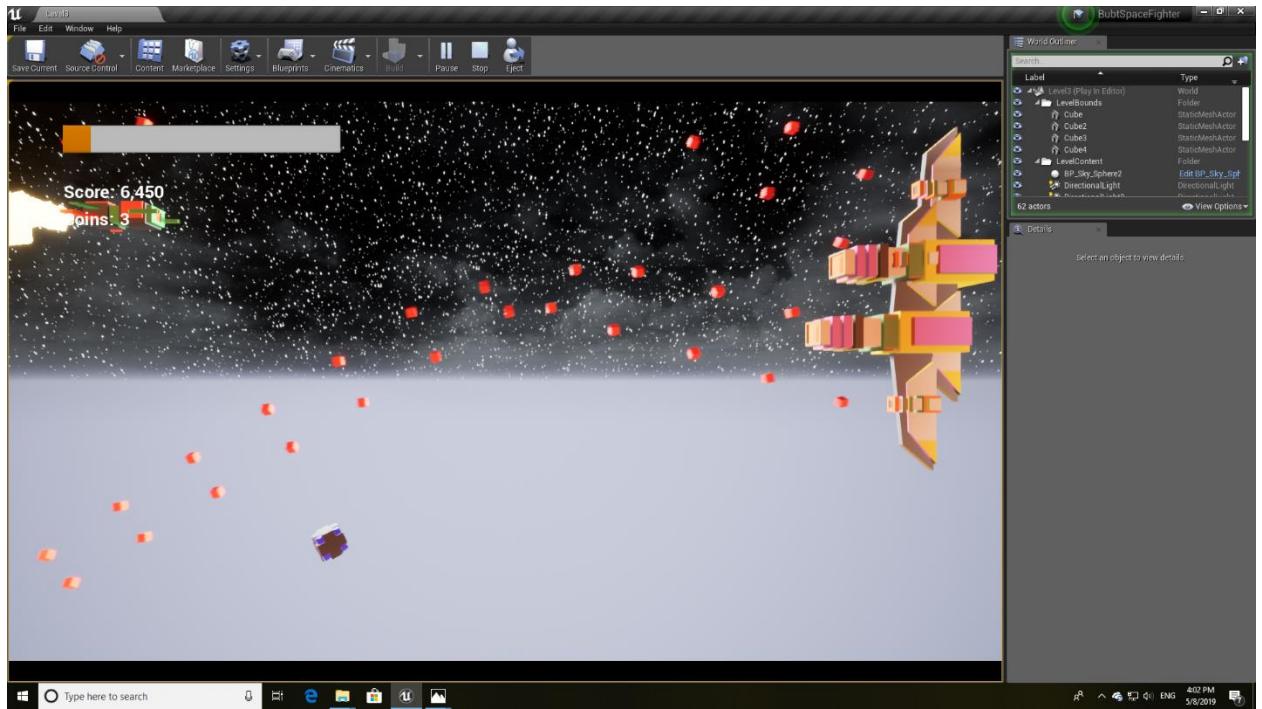


Figure 4-22: level-4 Enemy ships Boss

In the Last step of level4, there will be appear bosses, player health is 100, and bosses health is 5000, player has normal bullet (1) damage point is 30, when touch blue cube then bullet (2) damage point is 200, and also when touch red cube then bullet (5) damage point is 50. 2 Bosses bullet (2) damage point 5.

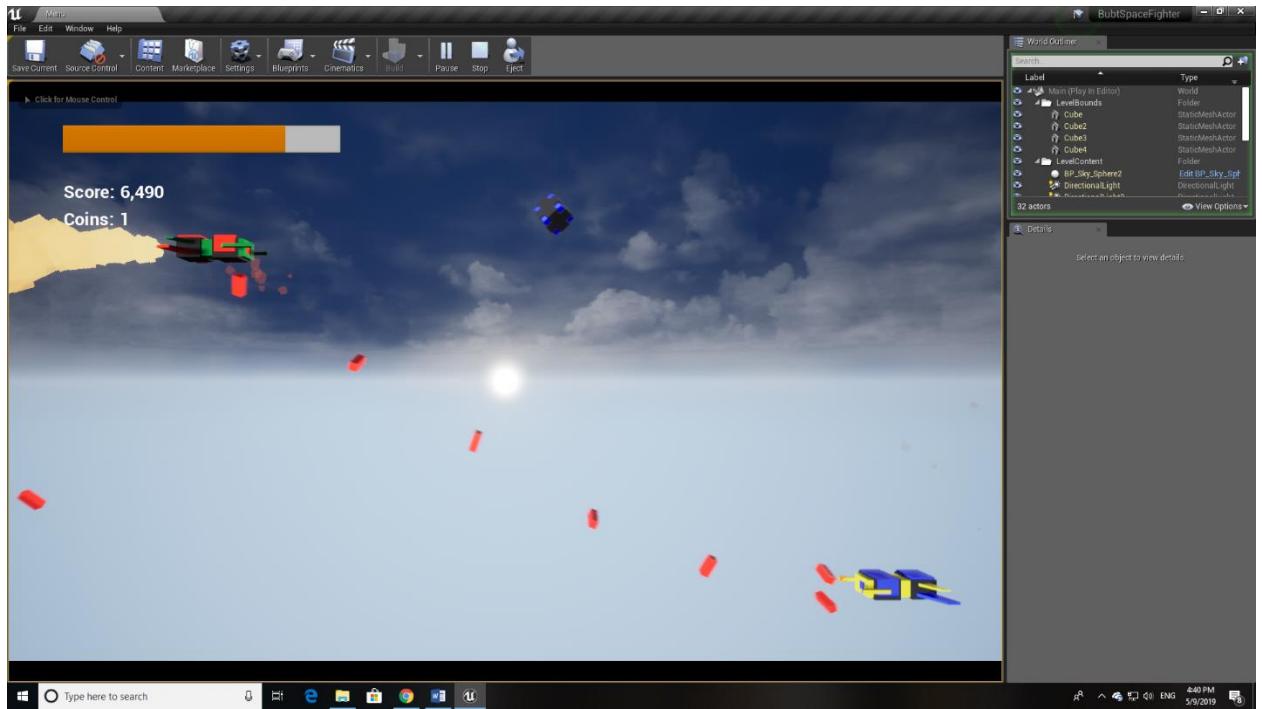


Figure 4-23: Blue Cube Rocket Power Up

There will be power here and there in the scene for the player which he needs to collect. Blue Cube, Collection is used for the increasing the player bullet damage point power. The player collects blue cube, the more power he/she gets.

Step 1: Player notices Blue Cube in a certain place

Step 2: He moves plane to the place

Step 3: plane automatically collects Blue Cube when gets in touch of it.

Automatically power will be up, press right button for throwing power up bullets.

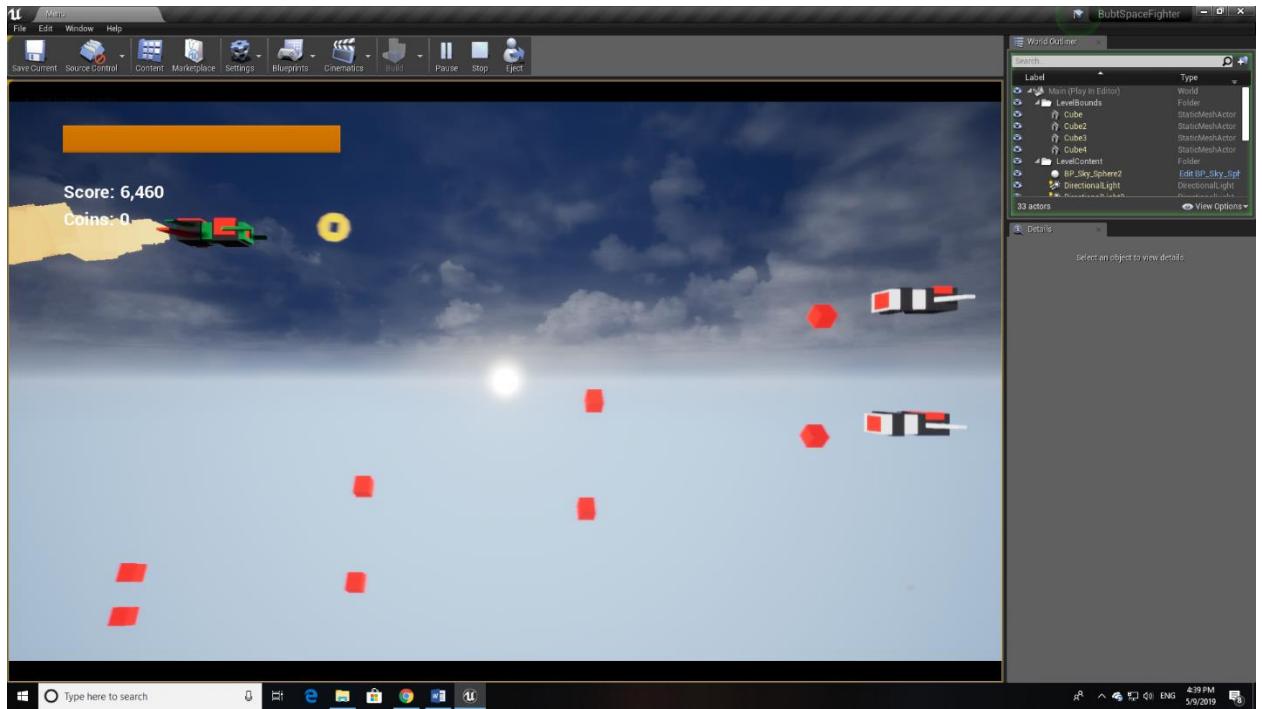


Figure 4-24: Coin

There will be Coin here and there in the scene for the player which he needs to collect. Coin Collection is used for the measurement of game coin how gamer earn. The more the player collects Coin, the more points he gets.

Step 1: Player notices coin in a certain place

Step 2: He moves plane to the place

Step 3: plane automatically collects coin when gets in touch of it.

REQ 2: Calculate points depending on total coin collected.

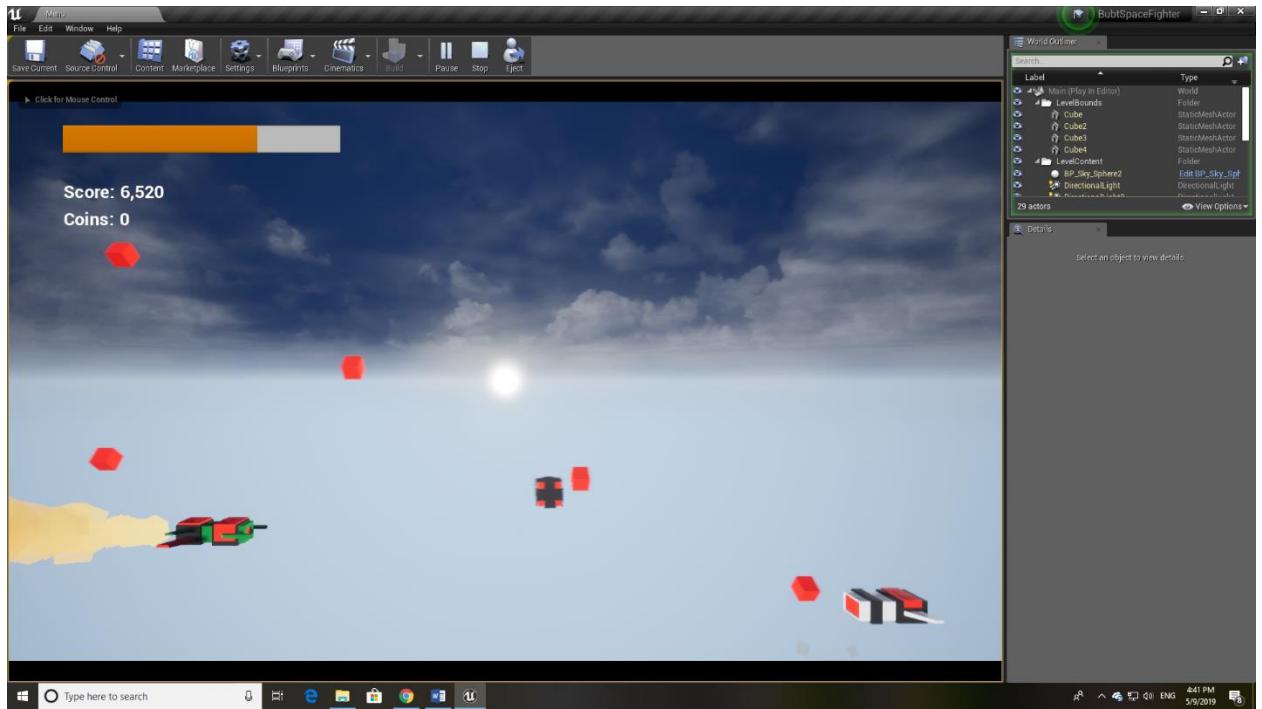


Figure 4-25: Red Cube Bullet Power Up

There will be power here and there in the scene for the player which he needs to collect. Red Cube, Collection is used for the increasing the player bullet. The player collects Red, the more power he/she gets.

Step 1: Player notices Red Cube in a certain place

Step 2: He moves plane to the place

Step 3: plane automatically collects Red Cube when gets in touch of it.

Automatically power will be up, press right button for throwing power up bullets.

4.8 Application

In 2011, Mobclix, an iOS advertising firm, released an infographic showing the average advertising revenue per user for different applications on different platforms. Their data was based on 50 top apps, each having at least 500,000 downloads or 75,000 daily active users. For iPhone games, the advertising revenue boiled down to \$4 per user each month. As a comparison, Android games using the same criteria only generated \$1.90 per user. Entertainment and utility apps on the iPhone generated more revenue than games, with \$6.70 and \$9.50 per user, respectively.

TOP GROSSING IOS MOBILE GAMING APPS 2016, RANKED BY DAILY REVENUE

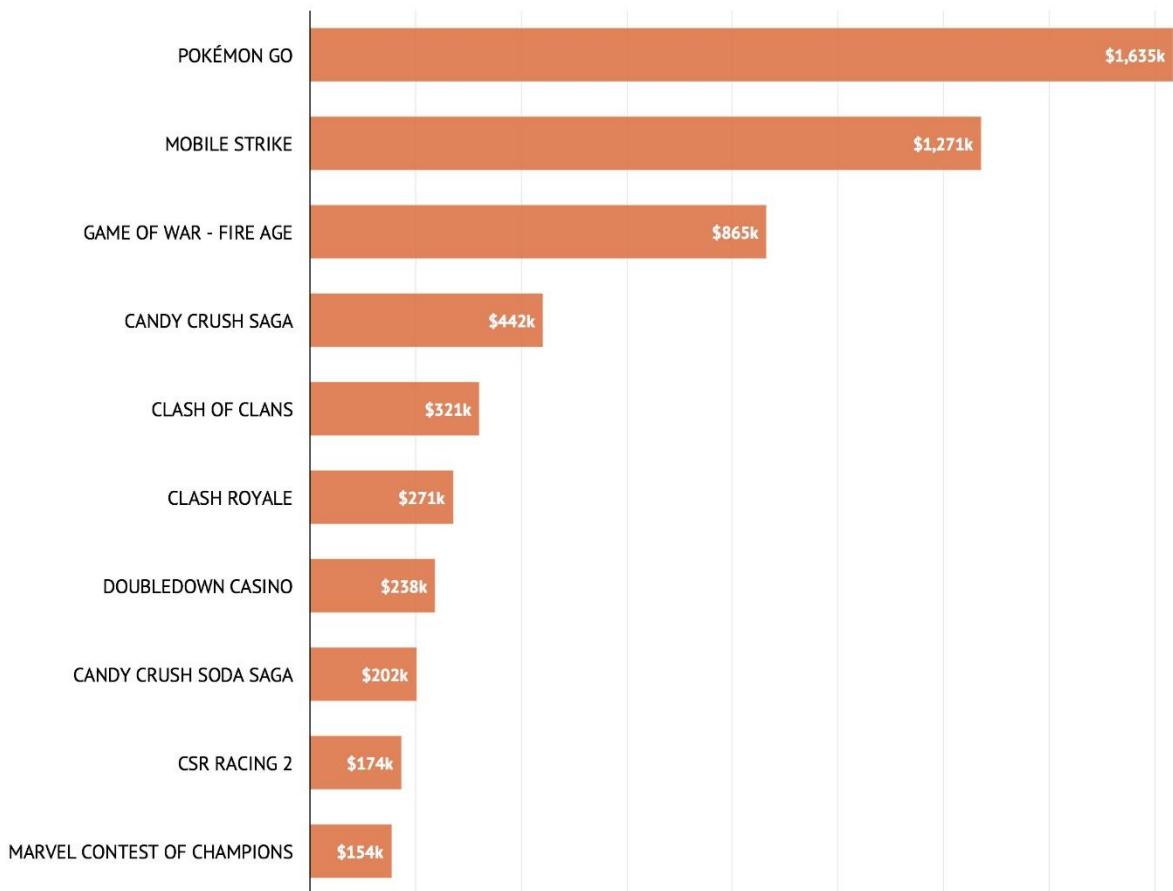


Figure 4-26: Mobile games apps revenue

Bangladesh is the third-largest video gaming market in South Asia and 61st globally in a ranking of 100 countries thanks to robust growth of internet users and smartphone penetration.

The country's video gaming market is worth about \$62.22 million or nearly Tk 500 crore a year, according to Newzoo, a leading provider of market intelligence of global games, e-sports, and mobile markets. In South Asia, Bangladesh lags behind India and Pakistan but is ahead of Sri Lanka. Nepal, Bhutan and Afghanistan could not grab a place in the top 100.

4.9 Conclusions

In this chapter we have discussed about the whole testing process. Testing is an important part of software development process. We have described the details of testing website application. Also we have shown all the test cases that we've tested on our website. In result analysis we've shown all the reports shown to admin. We've also described some application in our society

For setting up run time database to our games, we need to install unreal 4 games engine. Then Create a project into unreal4 and then add our project's table and others node as well.

We use blander for designing plane, we used Photoshop design materials.

Textures. Textures are images that are used in Materials. They are mapped to the surfaces the Material is applied to. However, some textures are generated within Unreal, such as Render Textures.

Chapter- 5

Conclusion and Future Work

5.1 Conclusions

We tried our level best to introduce a new gaming system that will be accurate, transparent, and faster and will ensure that games will give enjoyment and happiness when will kill enemies. Our proposed system has covered all of these issues successfully. Moreover, this system will provide boundary making chose the way of success. A better runtime database maintenance, automated playing system and the process of playing games using mouse or keyboard will further help us to fulfill our purpose. Our proposal enables a user to play games through internet or offline and additionally fast to access, highly secure, easy to maintain all information of games, highly efficient and flexible. Hence, by this user percentage will increase drastically. The using of online games has the capability to reduce or remove unwanted human decision. In addition to its reliability, online games can handle multiple modalities, and provide better scalability future path taking. Space ship games is also an excellent mechanism that does not require geographical proximity of the user. For example anyone can play easily without facing any difficulties[42].

Thinking of the game as a part of a bigger educational process is really in the core mindset that this project wants to promote. Games can do many things very well, but they certainly cannot do everything at once. Especially not without solid supporting structures around them. Throughout the project and the case studies we built this was true. As each teacher build her or his story these processes were discussed and reflected upon and we will be referring to these and link back to them. This project aimed as much at using alternative and innovative methods to teach through coding digital games and playing games as part of learning, as at developing the skills of teachers in extending academic goals to understand, support and include the whole child: not only their academic subject skills but also social, emotional and behavioral skills.

Some key factors have been identified which have enabled the teachers at the schools we saw to take steps towards better embedding games into their lessons. Many of these factors have been highlighted as the factors needed to bring about teacher change.

Senior management leader's team provided teachers with time to learn to use and play with technologies: this is helping to improve teachers' self-efficacy and confidence. Giving support to guide the use of technology and help build confidence. Giving teachers freedom and trust thus enabling teachers to take risks and experiment with ICT. Encouraging teachers to meet and support one another through sharing ideas and knowledge. Providing technical support. Recognizing and valuing teachers as professionals by giving more responsibility to enable CPD and career progression. Creating a culture of shared responsibility at all levels of the school, including the Head teacher, class teachers and support staff. Making funds available to purchase ICT. Encouraging teachers to use online spaces (Blogs, School websites) to share ideas and scale up[43]. Working with game engine completely a new experience for me. Normally i am working with different OO languages, DBMS, mark up languages etc. It is very sensible work and it demands much time because the game engines try to connect game environment with the real world. Creating a 3d model is very difficult because you need to work with each and every point of the model. The Exists game engines demands vast knowledge about its properties, sections and sub-sections.

Now I know much more about game engines. How it works? The properties, objects and others.

I know how a model is constructed and how it is animated. The main thing is that as a software engineer, skill and expertise to create a SRS document and an overall software product report is now better than before Develop communication skills Growing creative thinking and imagination capability.

5.2 Limitation

The data above indicates that there are limited resources to manage user and available information because we didn't used database. Also hasn't multiplayer games.

Results indicate that there is no clear dominate business and revenue models. There is a huge opportunity for voting system solution that will address the issue of trust and legitimacy.

- There is no admin panel.
- Built-in weapon few.
- Runtime score and health storage.

- Can be corrupted internally and externally.
- May have software issues.

5.3 Future Scope

There are some features that had to be dropped during the development of this project due to the lack of time. If the group were to continue further development, some of these features would be reexamined and implemented into the game. More content is something that would be the first thing to be added to the game is that multiple player can play the game at a same time. A proper development system, where the player receives better weapons or more health over time, has also been discussed. This could be implemented in a number of ways, either by collecting elevations dropped by enemies or by receiving an elevation when a certain number of enemies have been defeated by the player. Regardless of the specifics, this will give the player a feeling of development and thus will inspire them to keep on playing. Special powers like time rewind can also be implemented with extended time.

References

Articles

- [1]. From Internet different sources
- [2]. Hansen, Dusty (2016). *Game On! Video Game History From Pong and Pac-Man to Mario, Minecraft and More*. MacMillan Publishing Group, LLC. p. 11. ISBN 978-1-250-08095-0. Drain,
- [3]. CCP Games. "Patch Notes For Rhea". eveonline.com. Retrieved December 14, 2014.
- [4]. Savage, Phil. "The 50 most important PC games of all time".
- [5]. Brian (March 31, 2013). "EVE Evolved: EVE's Odyssey expansion could be incredible". Massively. Retrieved December 15, 2014
- [6]. Takahashi, Dean (2015). "Eve Online exhibit to become a permanent fixture at New York's Museum of Modern Art"
- [7]. CCP Games (March 6, 2003). "CCP Games Press Release: New Release Date for Eve Online: The Second Genesis Announced". ccpgames.com. Archived from the original on September 10, 2015. Retrieved January 15, 2008.
- [8]. CCP Games. "CCP Games Press Release: Eve Online Available for Download". ccpgames.com. Archived from the original on September 10, 2015. Retrieved December 14, 2014.
- [9]. "Eve Online Puts the MMO in Steam". January 22, 2008. Retrieved January 23, 2008.
- [10]. Matt 'Mix' Fox (1999-10-10). "Freespace 2". GameSpy. Archived from the original on 2007-10-15. Retrieved 2007-10-26.
- [11]. Staff (January 2008). "Classic GI: Space Invaders". Game Informer. Game Stop (177): 108–109.
- [12]. Khalil "SniperCloud" Wehbi (2004-05-12). "Freespace 2: Interplay 20th Anniversary Edition Review". XGP. Archived from the original on 2006-11-15. Retrieved 2007-10-26
- [13]. CCP Games. "Eve Online background story page". eveonline.com. Archived from the original on February 22, 2009. Retrieved January 15, 2008.
- [14]. CCP Games. "Timeline of the Amarr Empire". eveonline.com. Retrieved January 15, 2008.
- [15]. Kiphshidze, N; Zubiashvili, T; Chagunava, K (October 2005). "The Creation of Space Invaders". Edge. Future plc (154): 7–13.

- [16]. Marc Saltzman (n.d.). "FreeSpace 2". The Electric Playground. Archived from the original on 2007-10-10. Retrieved 2007-10-26.
- [17]. JB (1999-04-19). "Interplay Announces FreeSpace 2". IGN. Retrieved 2007-10-25.

- [18]. Mark 'KILLZAT' Jackson (1999-11-12). "FreeSpace 2". Speedy3D. Retrieved 2007-10-26.
- [19]. Stefan "Desslock" Janicki (1999-10-15). "Freespace 2". GameSpot. Retrieved 2007-10-26.
- [20]. Mark 'KILLZAT' Jackson (1999-11-12). "FreeSpace 2". Speedy3D. Retrieved 2007-10-26.
- [21]. "Elite: Dangerous Gameplay Demo - IGN Live: Gamescom 2014 - Sequel".
- [22]. "Elite: Dangerous Gameplay Demo - IGN Live: Gamescom 2014 - Sequel".
- [23]. "Blender History". Blender.org. Retrieved March 29, 2018.
- [24]. "Blender Foundation Launches Campaign to Open Blender Source". Linux Today. Retrieved 2017-01-22.
- [25]. "Eve Online Player Guide, Chapter 3, Character Creation". eveonline.com. Archived from the original on August 22, 2008. Retrieved September 3, 2008.
- [26]. "EVE Online Duality server". Archived from the original on March 23, 2011. Retrieved April 8, 2010.
- [27]. "EVE Online Singularity server". Archived from the original on March 14, 2010. Retrieved April 8, 2010.
- [28]. "Buckingham - EVElopedia". Archived from the original on March 10, 2013. Retrieved June 3, 2013.
- [29]. "Elite: Dangerous: the David Braben interview". 26 September 2014. Retrieved 14 November 2014.
- [30]. Cook, Dave (23 November 2012). "Elite: Dangerous: Braben's square peg". VG247. Retrieved 25 April 2014.
- [31]. Royce, Brianna (16 December 2014). "Elite: Dangerous' launch-day roundup". Joystick. Retrieved 17 December 2014.
- [32]. Parkin, Simon (9 July 2014). "The Video Game That Maps the Galaxy". The New Yorker. Retrieved 10 July 2014.
- [33]. "Elite: Dangerous review". gamesradar.com. Retrieved 13 January 2015.
- [34]. Sweeney, Tim (2005). "GPU Gems 2 – Foreword". Nvidia Developer. Archived from the original on October 3, 2017. Retrieved October 3, 2017.
- [35]. Free Blender campaign". 2002-10-10. Archived from the original on 2002-10-10. Retrieved 2017-01-22.
- [36]. "members". 2002-10-10. Archived from the original on 2002-10-10. Retrieved 2017-01-22.
- [37]. blenderfoundation (2011-07-09). "Blender 2.58 release notes". blender.org. Retrieved 2019-01-14.

- [38]. <https://www.blender.org/2-8/>
- [40]. "Atmospheric Audio". CCP Games. December 4, 2012. Retrieved June 24, 2014.
- [41]. "EVE Online - Soundtrack". CCP Games. December 4, 2012. Retrieved June 24, 2014.
- [42]. <https://www.gamesforlearning.se>
- [43]. <https://www.academia.edu>
- [48]. Craig Glenday, ed. (March 11, 2008). "Record Breaking Games: Shooting Games Roundup". *Guinness World Records Gamer's Edition 2008*. Guinness World Records. Guinness. pp. 106–107. ISBN 978-1-904994-21-3.
- [49]. Richards, Giles (July 24, 2005). "A life through video games". *The Observer*. UK: Guardian Media Group. Retrieved May 22, 2008.
- [55]. Brian Ashcraft & Jean Snow (2008), *Arcade Mania: The Turbo-charged World of Japan's Game Centers*, Kodansha International, ISBN 4-7700-3078-9, retrieved May 1, 2011, Space Invaders offered a novelty: players had three lives. Those who got good at the game could play for as long as they could keep from being blown to bits.
- [56]. Brian Ashcraft (January 20, 2010). "How Cover Shaped Gaming's Last Decade". *Kotaku*. Retrieved March 26, 2011.
- [57]. Karen Collins (2008). *From Pac-Man to pop music: interactive audio in games and new media*. Ashgate. p. 2. ISBN 0-7546-6200-4. Retrieved April 8, 2011.