April 3, 2019

Project Certificate

This is to certify that **Brijesh Kamani [160303307041]** studying in 6th semester of Diploma Computer Engineering at *'ATMIYA Institute of Technology & Science For Diploma Studies'* affiliated to Gujarat Technological University, has successfully completed his/her project work entitled ***"Strike of Vengeance"*** during December 2018 to April 2019.

This work is submitted as a fulfillment towards the requirement of the degree of **DIPLOMA** for academic year 2016 – 2019.

We take this opportunity to wish his success in all his future endeavors.

Prof. Ankit Faldu
[Head]
Department Of Diploma Computer
Engineering.

Prof. Nirali Gondalia
[Guide]
Department Of Diploma Computer
Engineering

Examiner

Exam Date:

April 3, 2019

Project Certificate

This is to certify that **Dhaval Hariyani [160303307038]** studying in 6th semester of Diploma Computer Engineering at *'ATMIYA Institute of Technology & Science For Diploma Studies'* affiliated to Gujarat Technological University, has successfully completed his/her project work entitled ***"Strike of Vengeance"*** during December 2018 to April 2019.

This work is submitted as a fulfillment towards the requirement of the degree of **DIPLOMA** for academic year 2016 – 2019.

We take this opportunity to wish his success in all his future endeavors.

Prof. Ankit Faldu
[Head]
Department Of Diploma Computer
Engineering.

Prof. Nirali Gondalia
[Guide]
Department Of Diploma Computer
Engineering

Examiner

Exam Date:

April 3, 2019

Project Certificate

This is to certify that **Tushar Harsora [160303307051]** studying in 6th semester of Diploma Computer Engineering at *'ATMIYA Institute of Technology & Science For Diploma Studies'* affiliated to Gujarat Technological University, has successfully completed his/her project work entitled ***"Strike of Vengeance"*** during December 2018 to April 2019.

This work is submitted as a fulfillment towards the requirement of the degree of **DIPLOMA** for academic year 2016 – 2019.

We take this opportunity to wish his success in all his future endeavors.

Prof. Ankit Faldu
[Head]
Department Of Diploma Computer
Engineering.

Prof. Nirali Gondalia
[Guide]
Department Of Diploma Computer
Engineering

Examiner

Exam Date:

A Project Report On

Strike of Vengeance

Single Player FPS Game

Submitted in fulfilment for the Award of degree in

DIPLOMA IN COMPUTER ENGINNERING

[Batch 2016 – 2019]

Submitted by

Dhaval Hariyani [166030307038]
Tushar Harsora [166030307051]
Brijesh Kamani [166030307041]

Under the guidance of

Ms. Nirali Gondalia

Department Head

Mr. Ankit Faldu

Submitted to

Atmiya Institute of Technology & Science for Diploma Studies

Gujarat Technological University

Certificate

ACKNOWLEDGEMENT

It is high privilege for us to express our deep sense of gratitude to all those faculty members who helped us in the completion of the project, especially our internal guide **Ms. Nirali Gondalia** who was always there at hour of need. Our respected **HOD Mr. Ankit Faldu** who always inspire us during our study period.

Without their willing disposition, spirit of accommodation, frankness, timely clarification and above all faith in us, this project could not have been completed in due time. Their readiness to discuss all important matters at work deserves special attention.

Last but not the least, we would like to thank our colleagues and our project partners who helped us throughout the project.

Team Members:
Dhaval Hariyani
Tushar Harsora
Brijesh Kamani

Abstract

Strike of Vengeance is a 3D single player first-person shooter game creating using C# in Unity Game Engine. At starting of game, all the players are spawned on map at specific locations. There are two teams, each having equal number of players. They try to eliminate their opponents.

Also, they have to complete their specific game mode tasks during a round. The main goal of players is building strategy to complete the mission.

The aim of this document is to describe how the problem was analysed, development methods, development processes carried out for creating a working product and provides information on what went right and wrong about the project and the lessons learned from experiences that have been gained during the project period and our discussions.

Index

Chapter – 1	**Project Introduction**	6
1.1	Introduction	7
1.2	Purpose	7
1.3	Technical Description	8
Chapter – 2	**System Analysis**	9
2.1	Introduction of Proposed System	10
2.2	Hardware and Software Requirement	11
	2.2.1 Minimum Requirement	11
	2.2.2 Recommended Requirement	11
2.3	Project Model	12
Chapter – 3	**System Design**	18
3.1	System Environment	19
3.2	Swim Lane Diagram	20
3.3	State Diagram	24
Chapter – 4	**Project Management**	26
4.1	Glimpse Of Project	27
4.2	Test Cases	34
4.3	Project Planning	38
Chapter – 5	**Project Limitations & Future Enhancements**	39
5.1	Project Limitations	40
5.2	Future Plans	40
	• Conclusion	41
	• The Obstacles	41
	• The Achievements	42
	• Last Few Words	42
	• Bibliography	43

Chapter: 1
Project Introduction

1.1 Introduction

"Strike of Vengeance" is a first person shooter (FPS) game. The game revolves around two teams trying to eliminate each other, while also completing separate objectives depending on the team they are in. The Terrorists have to plant bomb and secure the bomb zone, while the Counter Terrorist have to prevent the bomb from planting or defuse the bomb in order to complete the round. At the end of each round, the players are rewarded achievements and points based on their individual performance. The match will end after certain amount of rounds are won by a particular team.

1.2 Purpose

The main aim of our game is to provide entertainment to players while also improving some skillset including –

- Precision
- Team Work
- Pressure Handling
- Working in Time Bound
- Improve Coordination
- Problem Solving
- Attention and Concentration (Focus)
- Multitasking
- Forward Thinking and Strategic Planning
- Creativity

1.3 Technical Description

	Company	Tool	Usage	We used For
	Unity Technologies	Unity3d	Game Engine	Backend activity
	Blender	Blender	3D modelling, Texturing, Graphics editing	Create and Animate 3D models
	Autodesk	Maya	Graphics Design and Animation	Create 3d Model
	Adobe	Photoshop	Picture Edit	Textures Editing and Icons
	Microsoft	Visual Studio	Programming	Making Programs

Chapter: 2
System Analysis

2.1 Introduction of Proposed System

- Strike of Vengeance is a FPS game, so like any other FPS game the user will be playing the character based upon the game mode and team.
- Players assume the roles of members of combating teams of the governmental counter-terrorist forces and various terrorist militants opposing them.
- To win the game a team must either eliminate all the players of opponent team or complete their proposed mission.
- The users will be rewarded with points and achievements at end of every round. This nature of the game encourages teamwork and trying to stay alive.
- So, this is the gameplay/story of many games, Yeah?

What Makes our game unique?

➢ **Assassination:** There will be one VIP character and the player will have to either kill or escort the VIP.

Case 1: If player choose to play as Terrorist

- The VIP will be guarded by 4 Counter-Terrorists and the Player have to find the VIP and kill him to win the round.
- If Player gets killed by Counter-Terrorist or time runs out, then he will fail.

Case 2: If player choose to play as Counter-Terrorist

- The VIP will be guarded by 4 Terrorists and the Player have to escort the VIP to safe zone to win the round.
- If Player gets killed by Terrorist or time runs out, then he will fail.

2.2 Hardware and Software Requirement

2.2.1 Strike of Vengeance Minimum Requirements

- CPU: Intel Core 2 Duo or AMD Athlon 1600+
- CPU SPEED: 1.7 GHz
- RAM: 2 GB
- OS: Windows 7 (32/64-bit)
- VIDEO CARD: DirectX 9 level Graphics Card
- VRAM: 512 MB
- SOUND CARD: Yes
- FREE DISK SPACE: 10 GB

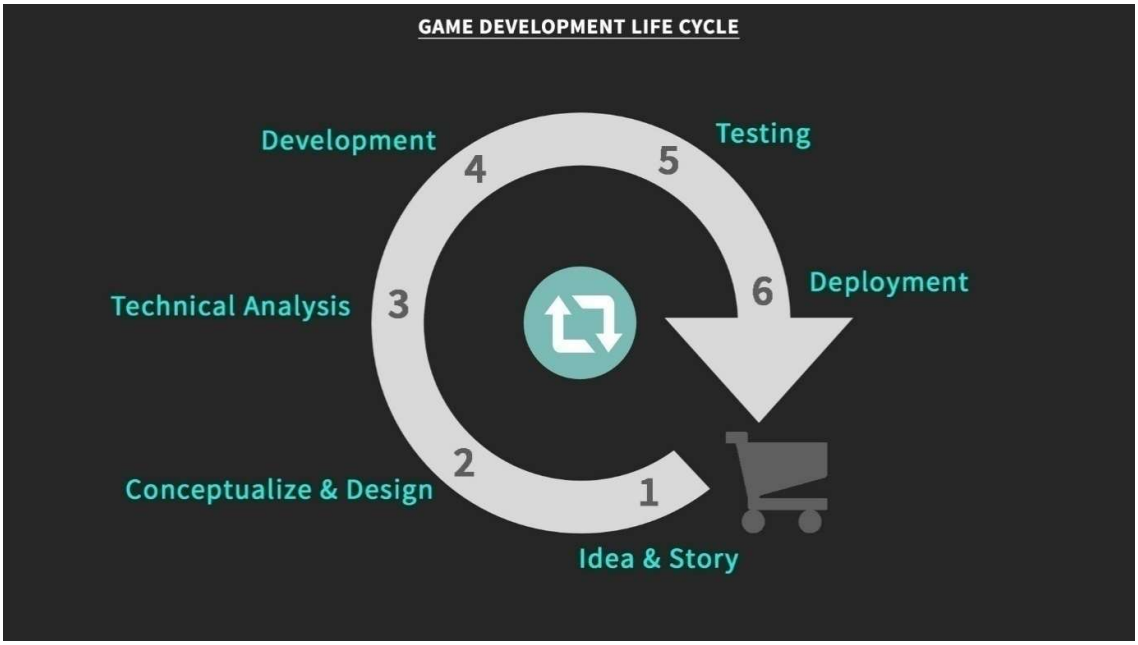
2.2.2 Strike of Vengeance Recommended Requirements

- CPU: Intel Core i3 or AMD Athlon 3600+ or Better
- CPU SPEED: 3.0 GHz or Higher
- RAM: 4 GB or Higher
- OS: Windows 10 (64-bit)
- VRAM: 1 GB or Higher
- VIDEO CARD: DirectX 11 level Graphics Card
- SOUND CARD: Yes
- FREE DISK SPACE: 10 GB

2.3 Project Model

Game Development Life Cycle (GDLC)

The game development life cycle (GDLC) as a life cycle of software development with the objective to entertain the users. Comparing to software development life cycle, developers face many challenges while developing games and hence they follow a different approach for developing games which are known as GDLC. Though we cannot define the lifecycle of every successful game explicitly but there is a common graph for developing games for pc, mobiles or console.



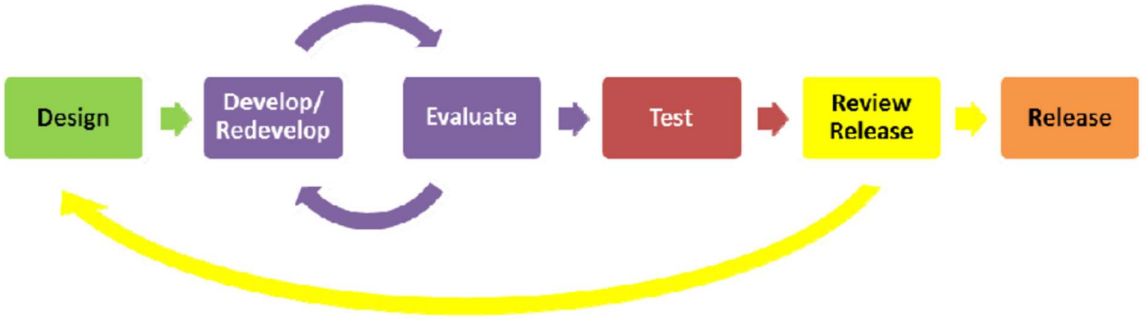
[Fig. 2.5.1 Game Development Life Cycle]

Why GDLC?

Software Development product always require a need for an existing problem and its developed for providing a solution, whereas, Game Development product is for entertainment to engage people to have fun, learn and spend good time. Hence the product will require an idea, story, innovation, creativity, writing skills and technical expertise.

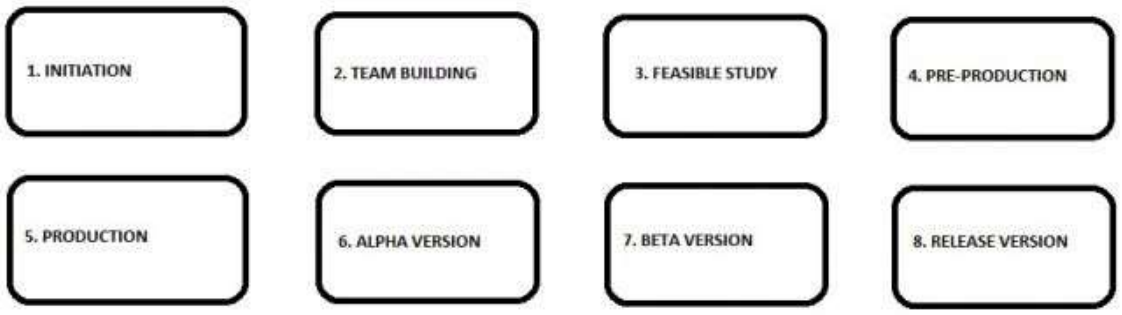
Hence, just following SDLC engineering principles as a product will have complications and iterations in design and development as a gaming product. Therefore, simply adopting the SDLC is not enough, as the developers face several challenges such as (Artwork, Visuals, Animations, Character Interactions, Collisions, Physics, Sounds, etc.) during its life cycle. To address the problem, we must use a specific approach called GDLC.

The flow of game development will be like:



[Fig. 2.5.3 The Flow of GDLC]

Fundamental Steps of GDLC:



[Fig. 2.5.2 Fundamental Steps of GDLC]

1. Initiation

It's all where it begins. Initiation is the part where you decide what kind of game you'll make, who's the hero, how to play the game, etc. It's the very basic of game design/game concepting.

To-do list:

- Define the game
- Look for publisher

Useful pointer: Answer these question first!

- **Brainstorming!**
- What **kind of game** you'll make?
- Who's the **hero/protagonist?**
- Will it be **2D** or **3D** game?
- What's the **theme?**
- What **features** it'll be?
- Who's your **potential player**/user? What **gender,** how **old** are they?

We have done finalising our game title, game type, game modes, gameplay, features, etc.

2. Team Building

Team building is essentially the same as staffing. At first, the team you have maybe consists of 3 to 10 people. However, you know that you need some extra muscles to do the job, and ensure that the whole team have the same vision as you are (assuming you're the project leader).

To-do list:

- Staffing/hiring
- Create team structure
- Delegating jobs to members
- Goal/vision synchronization

Our core team has 3 members. One for Coding, one for Designing, one for Documentation and other tasks. The goal is to build a fully working FPS game.

3. Pre-production

One of the most important thing before jump directly to code the game: Pre-production. It's the same as planning and designing the game itself.

To-do list:

- Brainstorming! (again)
- Design the gameplay (core gameplay, mini games, etc.)
- Define the art, style, and assets
- Choose the game engine and target platform
- Build prototype/mock-up/early design
- Create documentation: Art Design Document, Technical Design Document, Game Design Document
- Schedule planning

The characters and stages are final and are currently being made. The gameplay and concept documentation is done. We have chosen the Unity Game Engine and Blender for Graphics related work.

4. Production (We are currently at this stage)

It's time to pour everything you've design to become a game. The longest and exhaustive phase of the game development, production deals with the game codes and assets themselves.

To-do list:

- Coding (use framework/game engine to make it easier)
- Create assets (graphical and sound)
- Integrate source code and assets
- Evaluate
- Meeting

5. Alpha Version

Alpha version is the phase where the game itself is playable, but incomplete. For example, if the game has some playable level, it has already reach alpha version. Most of the core gameplay must be already included in the alpha version game.

To-do list:

- Alpha testing (testing by fellow developer/another team member)
- Bug fixing
- Continue production

Important note:

- Alpha version game means the **whole gameplay concept** is already **fixed.** No gameplay concept may change after a game is declared alpha, otherwise it's not alpha.
- Alpha testing is used to check usability error

6. Beta Version

Beta version is the phase where all the game is all set. It means, beta version game is already a full game.

To-do list:

- Beta testing (testing by 3rd party)
- Collecting user feedback
- Bug fixing
- Balancing

7. Release Version

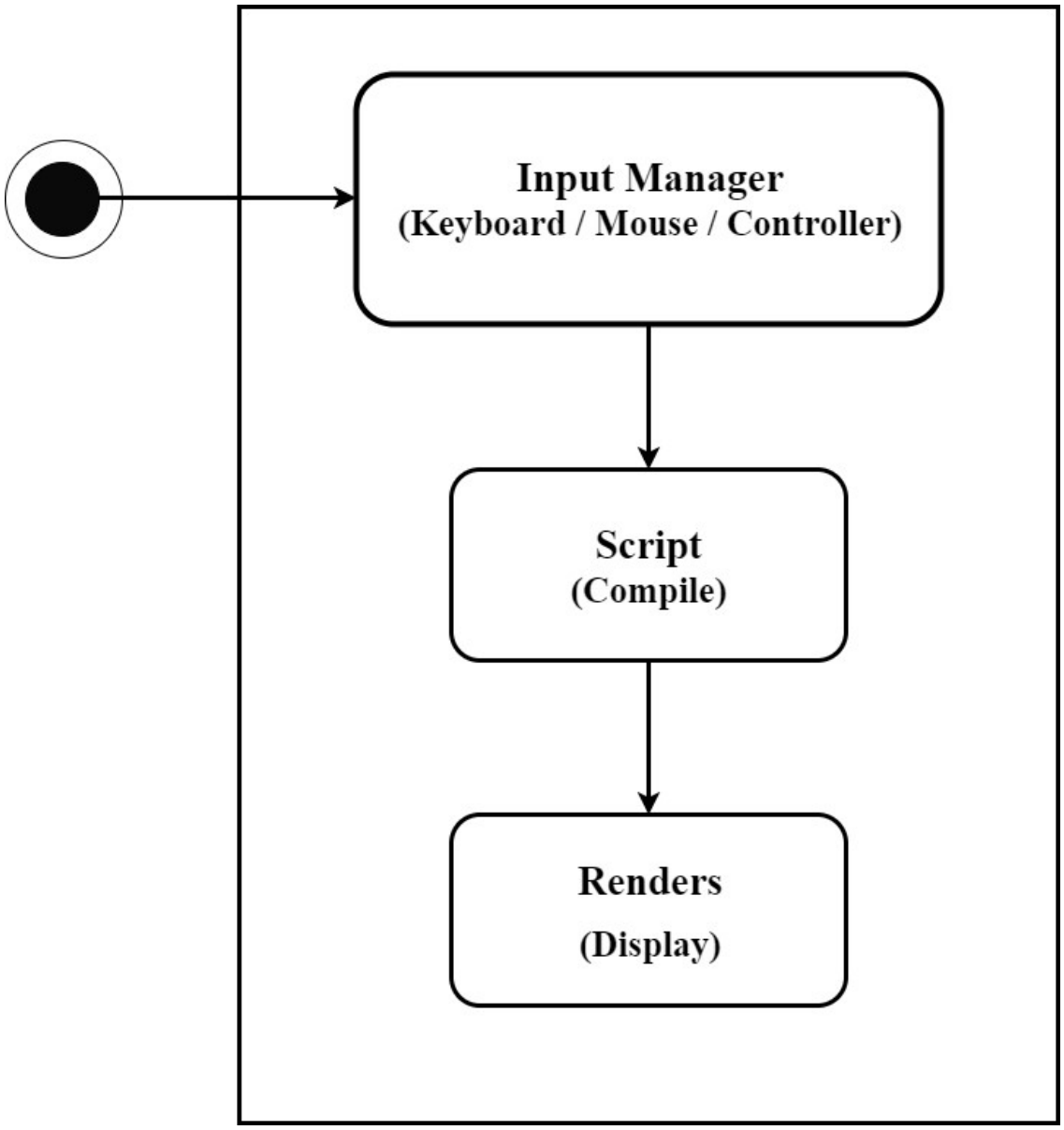
It's when all of the efforts rewarded. Release version means the game is ready to launch and to be shipped. Usually a game launching being held in some kind of party, but it's entirely optional. (but, we want to)

To-do list:

- Game launching
- Maintenance!

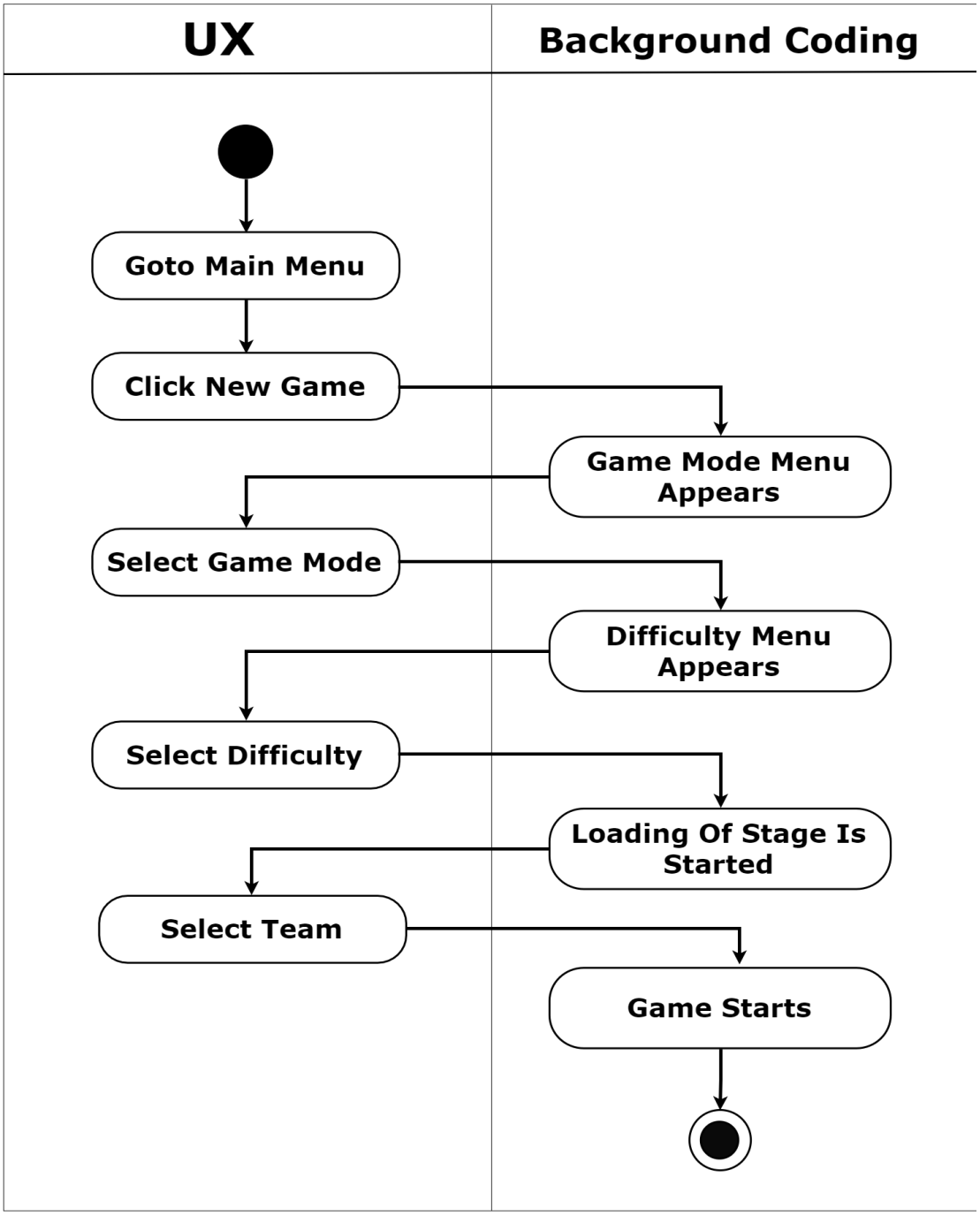
Chapter – 3
System Design

3.1 System Environment

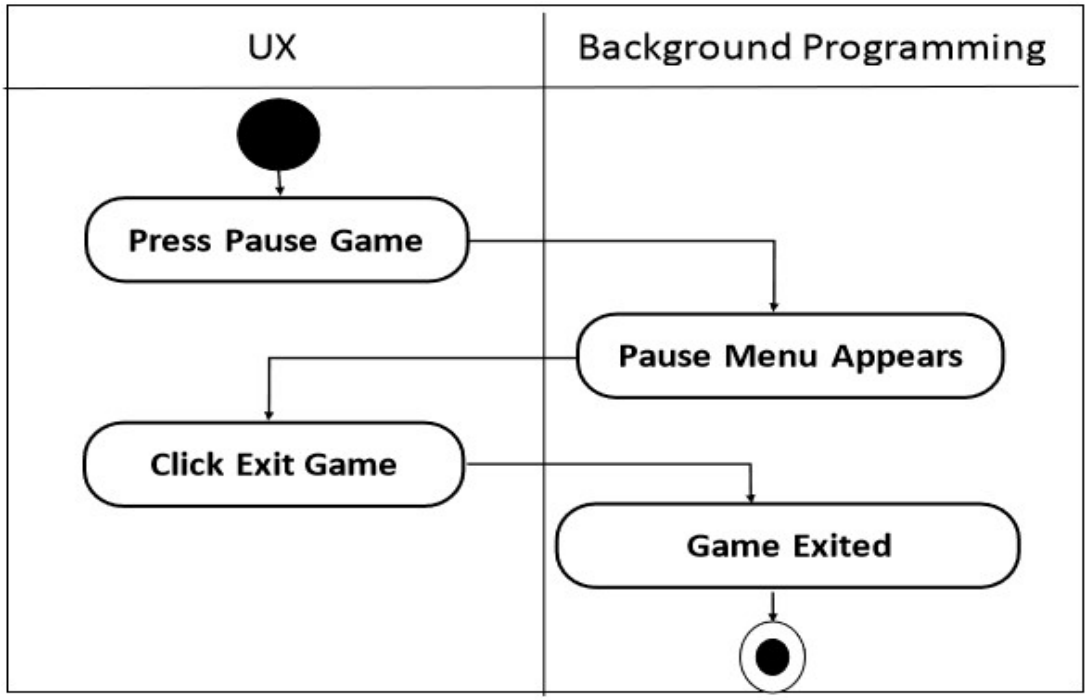


3.2 Swim Lane Diagram

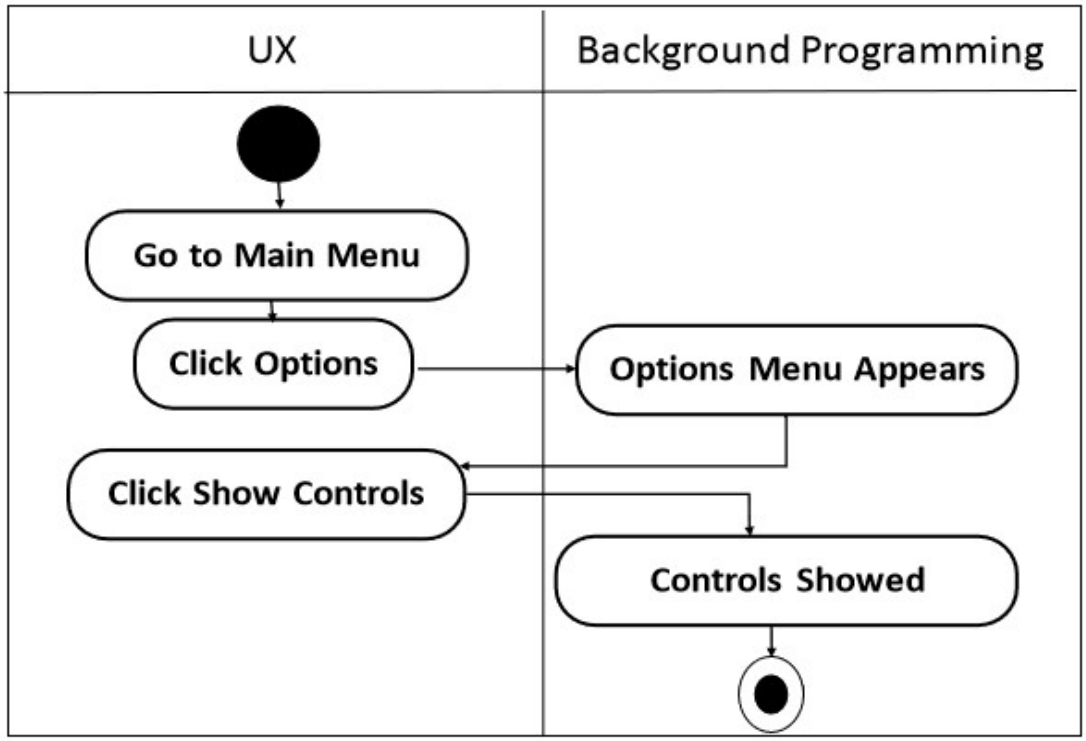
3.2.1 New Game



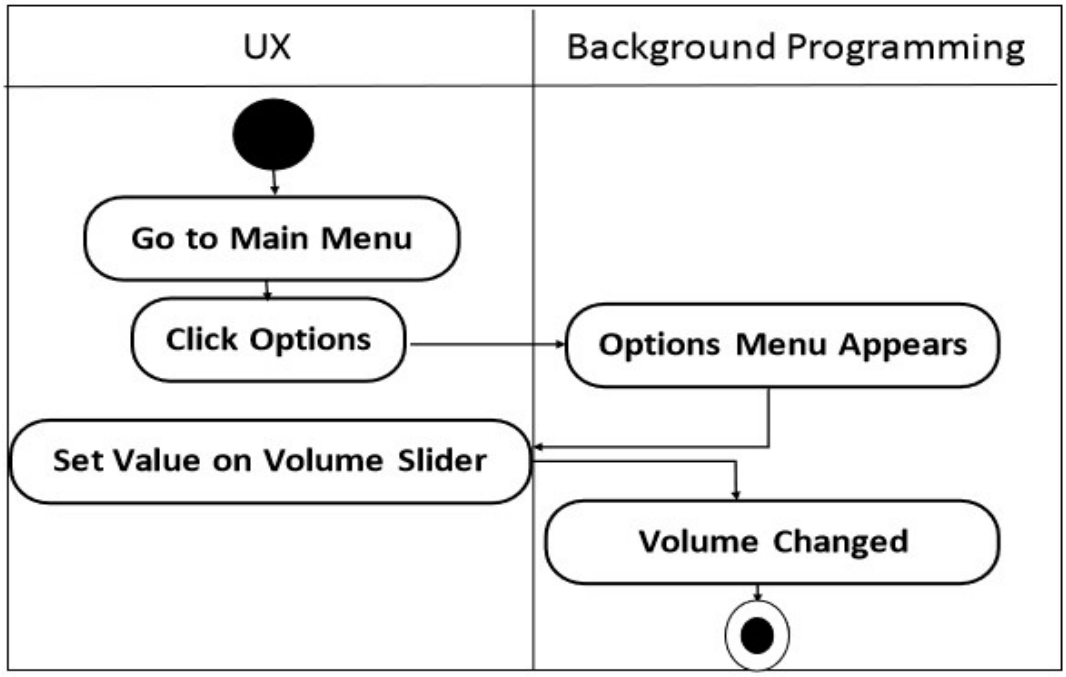
3.2.2 Pause Menu



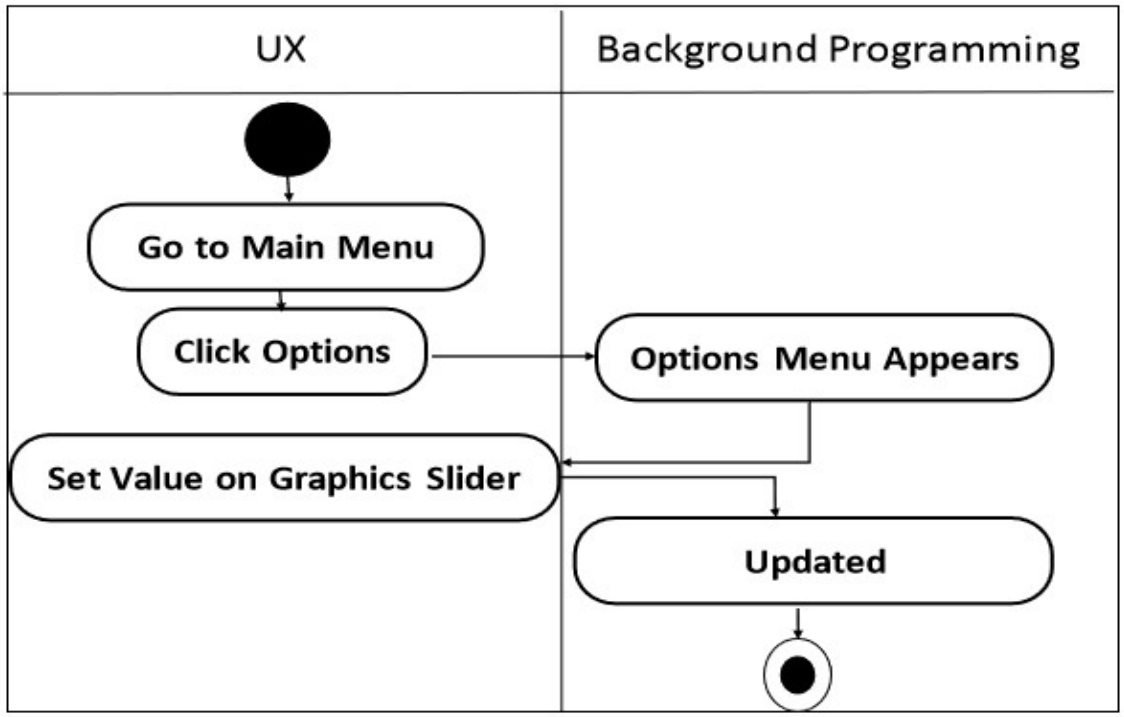
3.2.3 Controls Option Menu



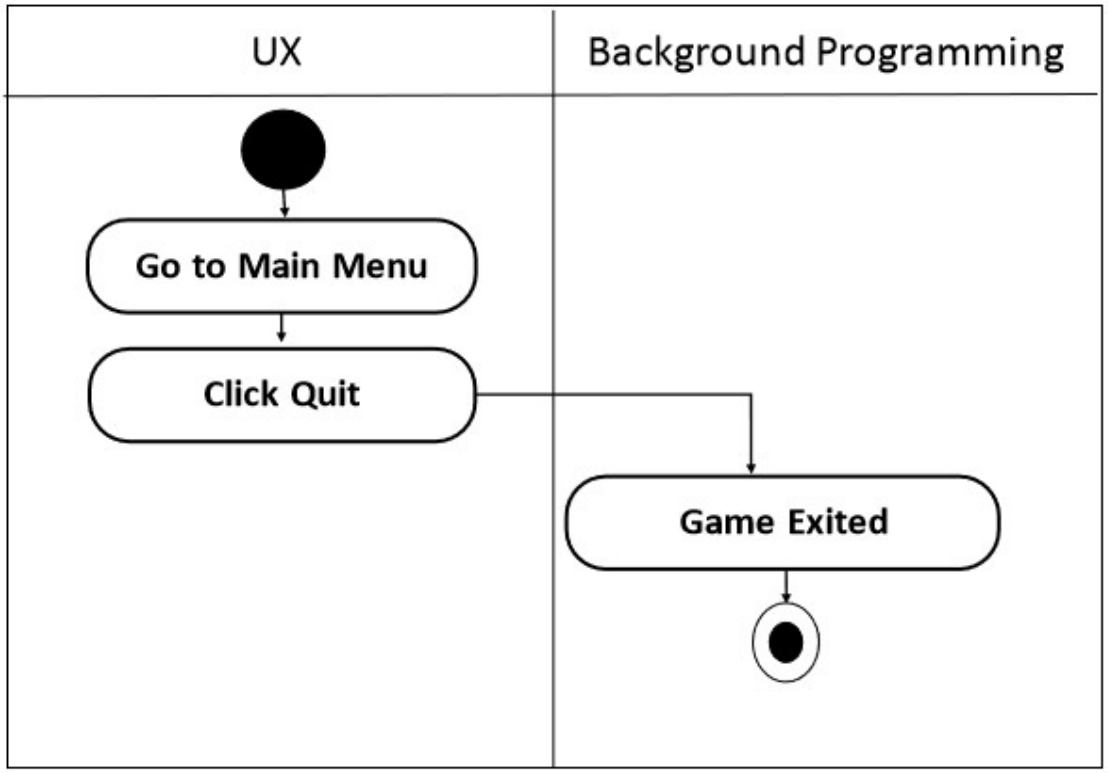
3.2.4 Music Option Menu



3.2.5 Graphics Option Menu

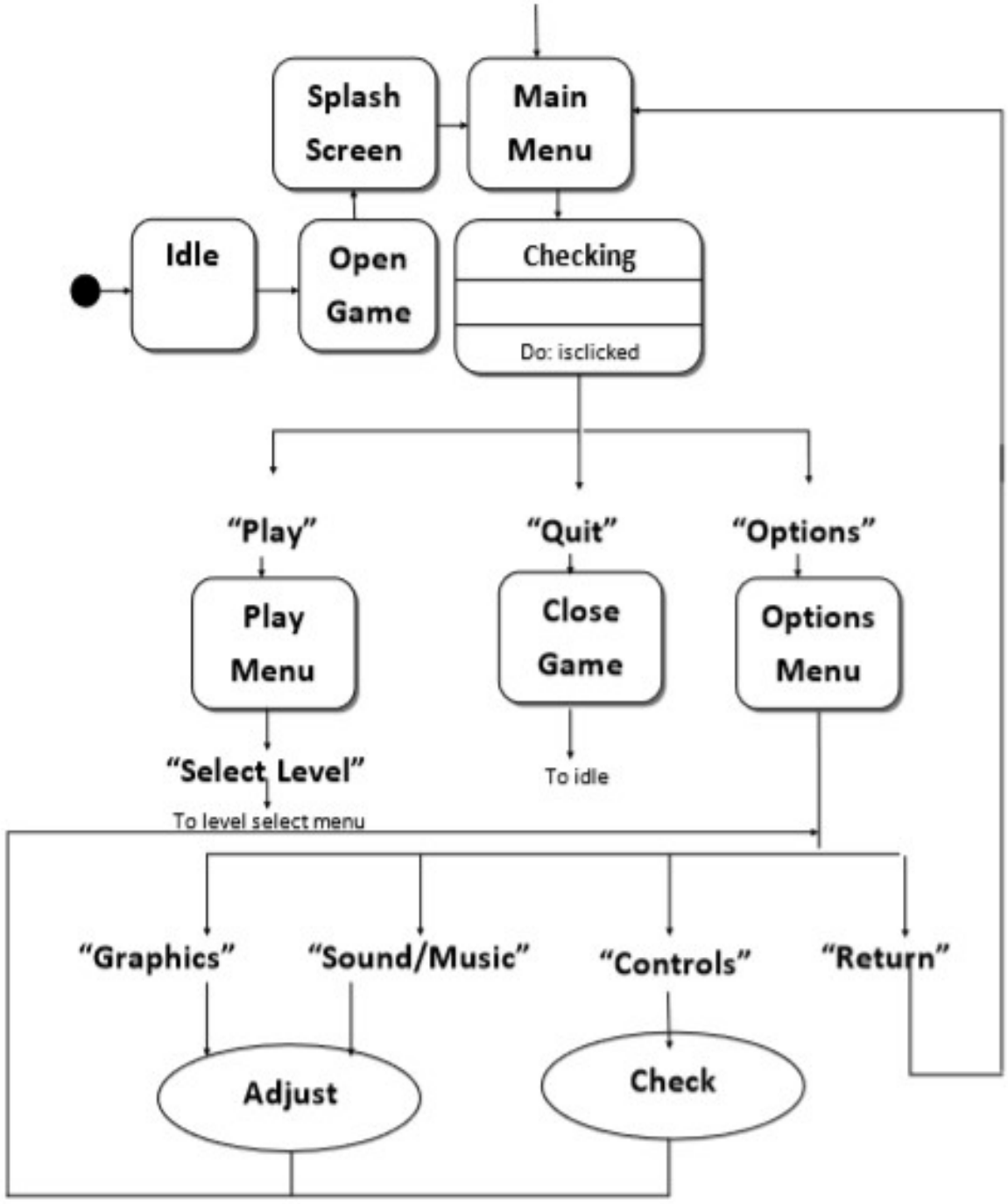


3.2.6 Quit

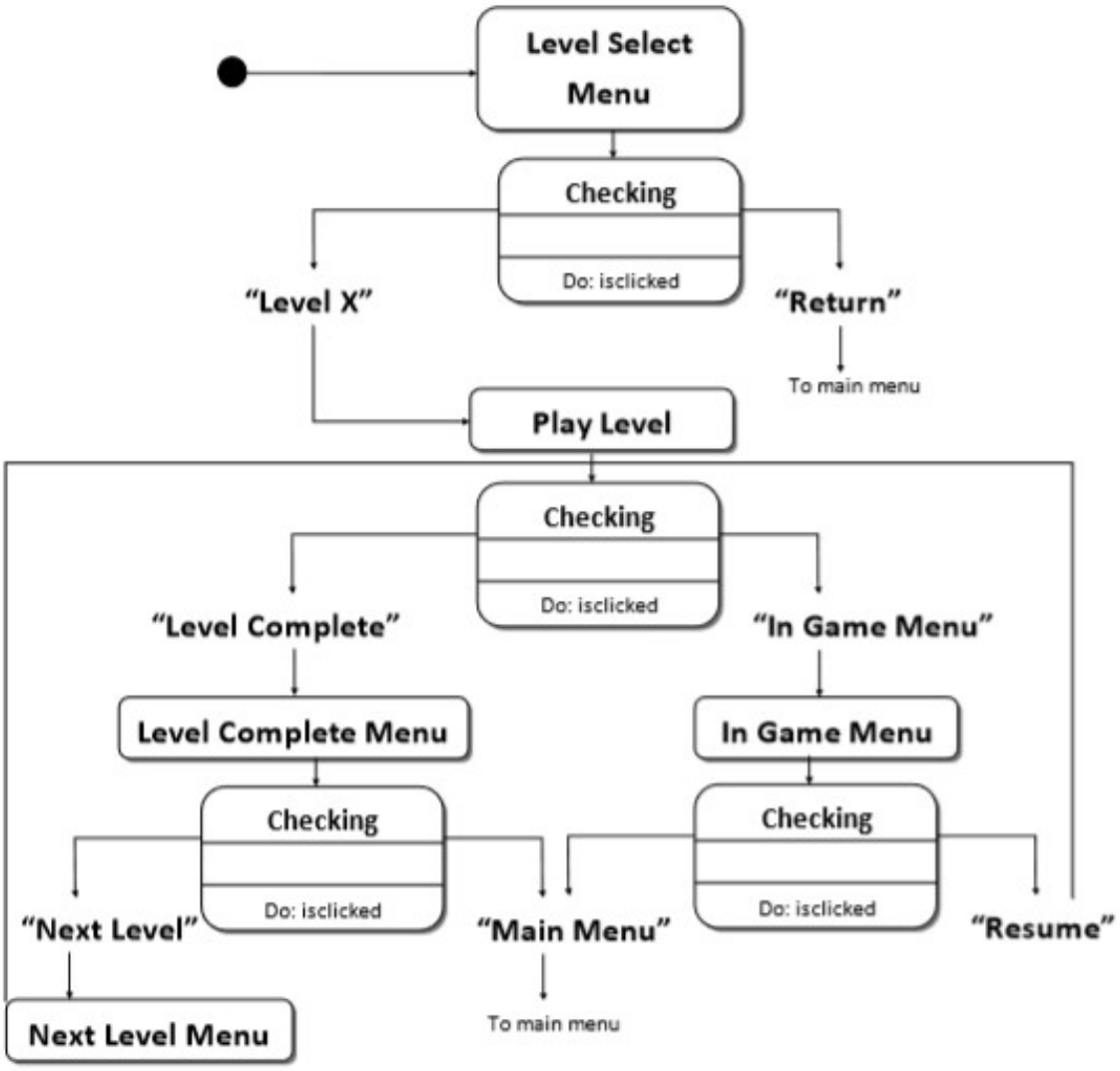

UX
Background Programming
Go to Main Menu
Click Quit
Game Exited

3.3 State Diagram

From Level Select, Level Complete, and in Game Menus in Play Level



Select Level State Diagram



Chapter: 4
Project Management

4.1 Glimpse of Project

Image 4.1.2: Game Logo

The main logo of our game.

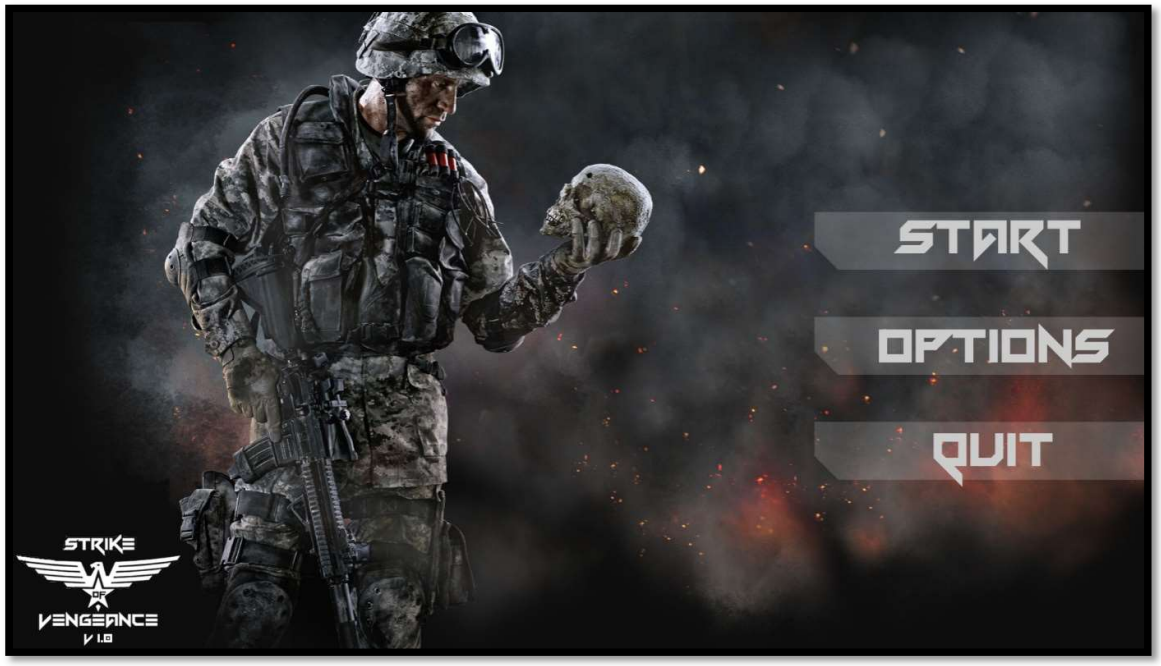


Image 4.1.2: Main Menu Screen

The main menu screen from where user can interact with other menus.

Image 4.1.3: Options Menu Screen

The options menu screen from where user can change in-game settings like audio, graphics, etc.

Image 4.1.4: Loading Screen

The Loading screen which appears while game is loading something.

Image 4.1.5: Enemy Encounter

The enemy aims at player.

Image 4.1.6: The Player Roams In The World

The Player can roam in the world freely.

Image 4.1.7: Spawn point

The player spawns at specific point.

4.2 Test Cases

This chapter includes some test cases for the game to check if the game works properly in various situations. We are giving four test examples for four different situations here.

Test Case 1

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.

Conditional Test : Again run scene.

Expected Result : Animation is working now.

Actual Result : Yes, it is working.

Accuracy : Accuracy depends on hardware configuration.

Test Case 2

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.

Test Case 3

Test Case : This test will check if the coding for player movement is working correctly or not.

Test Procedure : Add scripts of interaction in the objects that we want to interact with each other and make player controller. Run scene.

Expected Result : Player should be able to move the character with keys.

Actual Result : Player can control the character but some keys are misbehaving

Comment : Need to fix the keys coding and fix the controller.

Conditional Test : Run scene.

Expected Result : Player should be able to play.

Actual Result : Player controller is working fine and keys are working.

Accuracy : Perfectly accurate.

Test Case 4

Test Case : This test will check if the guns and other mechanisms are working or not.

Test Procedure : Add script to fire gun and movement the weapon.

Expected Result : Player should be able to fire the bullet from gun.

Actual Result : Gun id firing bullet.

Comment : No need to fix anything as the test is fully succeed.

Conditional Test : Run scene.

Expected Result : Player should be able to play and fire bullet from gun.

Actual Result : Player controller is working fine and keys are working.

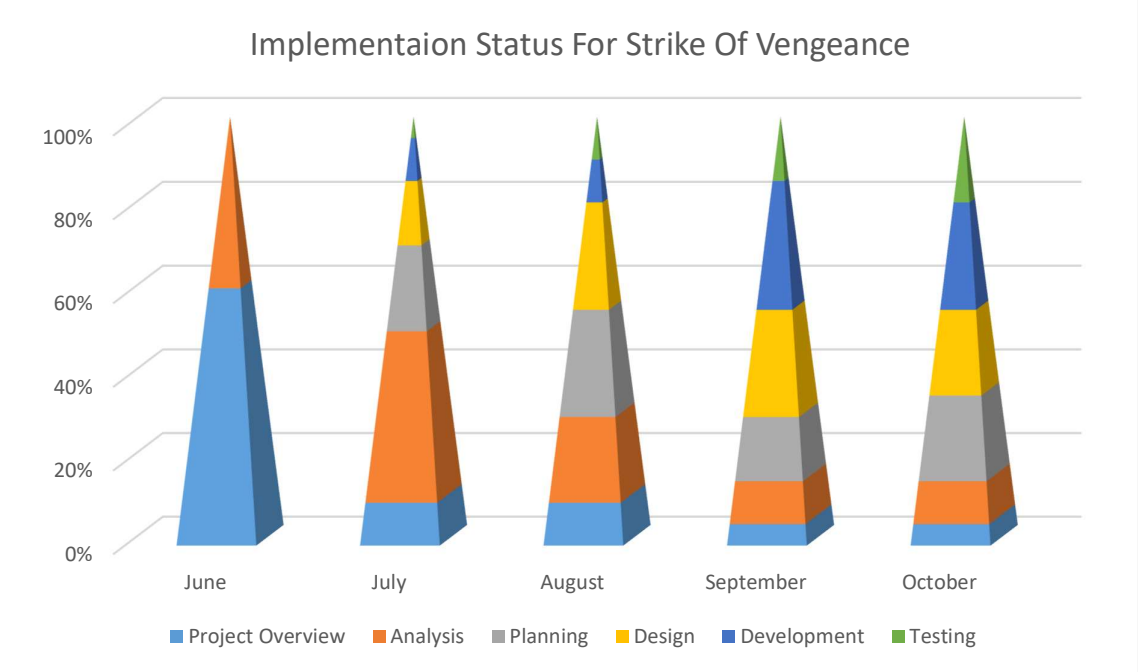
Accuracy : Perfectly accurate.

4.3 Project Planning

4.3.1 Roles and Responsibilities

Person	Roles
Tushar Harsora	Analysis, Design, Coding
Dhaval Hariyani	Analysis, Design, Coding
Brijesh Kamani	Analysis, Design, Coding

4.3.2 Implementation Status



Chapter: 5

Project Limitations & Future Enhancements

5.1 Project Limitations

- Only a single player can play the game.
- The user needs a specific or higher hardware to play the game.
- The bots are bit dumb.
- Weapons animation is not smooth.
- The game mechanism is bit older.

5.2 Future Plans

- Level Extension
- Improve Graphical Representation
- Introduce new game features
- Introduce new environment and scenes
- Take user response through website and produce web rank list

The Conclusion

A software project means a lot of experience. In this section we summarize the experience gained by project team during development of “Strike of Vengeance”.

It was a great and very interesting experience for us to make a game. We never thought that we would make a game while actually playing it. We are grateful to all the people who helped up to make our dream come true.

The Obstacles

1. Working with game engine completely a new experience for us. Normally we are working with different OO languages, DBMS, mark up languages etc.

2. We adopt these things by video tutorials, text tutorials, internet and learning materials given by the tools themselves. It's a matter of time, patience and hard work.

3. It is very sensible work and it demands much time because the game engines try to connect game environment with the real world.

4. Creating a 3d model is very difficult because you need to work with each and every point of the model.

5. The Exists game engines demands vast knowledge about its properties, sections and subsections.

The Achievements

- Now we know much more about game engines. How it works? The properties, objects and others.
- We 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.
- Co-Operation between group members.
- Develop communication skills
- Growing creative thinking and imagination capability.

Last Few Words

We learned a lot through this project. This project has sharpened our concept of Game engine, animation and the software-hardware interface.

We learned a lot about different documentation. The piece of software that we developed is intended to serve the gamers of the world.

The success of this project may give pleasure to billions of game lovers among the universe.

This project not only tested our technical skills but also our temperament.

There were times that we almost lost hope but we recovered through constant concentration and hard work.

If any kind of suggestion, improvements, more efficient development idea please feel free to communicate with us.

Bibliography

General References

1. http://www.mixamo.com/
2. http://www.unity3dstudent.com/
3. http://thefree3dmodels.com/stuff/
4. http://students.autodesk.com/
5. http://www.digitaltutors.com/11/index.php
6. http://library.creativecow.net/articles/3dsmax.html
7. http://www.good-tutorials.com/tutorials/3ds-max/animation

Special Thanks To

1. YouTube - http://www.youtube.com/
2. Archive3d - http://archive3d.net/
3. Unity3D - http://unity3d.com/
4. Unity Cookie - http://cgcookie.com/unity/
5. Unity Asset Store - http://assetstore.unity3d.com/