A PROJECT REPORT

ON

<u>GAME DEVELOPMENT</u>

Submitted by

Mit Dasondi (150410107017)

Meet Soni (150410107042)

Mitravind Pattnaik (150410107049)

In partial fulfilment

Of

BACHELOR OF ENGINEERING

in

COMPUTER ENGINEERING

SARDAR VALLABHBHAI PATEL INSTITUTE OF TECHNOLOGY,

VASAD

Gujarat Technological University, Ahmedabad

SARDAR VALLABHBHAI PATEL INSTITUTE OF TECHNOLOGY, VASAD
COMPUTER ENGINEEERING DEPARTMENT

CERTIFICATE

Date: 08-04-2019

This is to certify that the project entitled "GAME DEVELOPMENT USING UNREAL ENGINE" has been carried out by Mitravind Pattnaik (150410107049), Mit Dasondi (150410107017), Meet Soni (150410107042) under my guidance in partial fulfilment of the project in Bachelor of Engineering in Computer Engineering 8th semester of Gujarat Technological University, during the academic year 2018-2019.

Internal Guide:

Dr. Neha Soni
HOD, CE
SVIT Vasad

Head of Department:

Dr. Neha Soni
CE Department
SVIT Vasad

Plagiarism Checker X Originality Report

Similarity Found: 13%

Date: Friday, April 12, 2019

Statistics: 447 words Plagiarized / 3523 Total words

Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

A PROJECT REPORT ON GAME DEVELOPMENT Submitted by Mit Dasondi 150410107017 Meet Soni 150410107042 Mitravind Pattnaik 150410107049 In partial fulfilment Of BACHELOR OF ENGINEERING in COMPUTER ENGINEERING / SARDAR VALLABHBHAI PATEL INSTITUTE OF TECHNOLOGY VASAD Gujarat Technological University Ahmedabad SARDAR VALLABHBHAI PATEL INSTITUTE OF TECHNOLOGY VASAD COMPUTER ENGINEEERING DEPARTMENT CERTIFICATE Date 2019 This is to certify that the project entitled "GAME DEVELOPMENT USING UNREAL ENGINE" has been carried out by Mitravind Pattnaik 150410107049 Mit Dasondi 150410107017 Meet Soni 150410107042 under my guidance in partial fulfilment of the project in Bachelor of Engineering in Computer Engineering 8th semester of Gujarat Technological University during the academic year 2018 2019 Internal Guide Head of Department Dr. Neha Soni Dr. Neha Soni HOD CE CE Department SVIT Vasad SVIT Vasad Acknowledgement Nothing can be done perfectly without a proper guidance. No task can be fulfilled in a land of selfish takers. We are highly indebted to all the people involved in the development of the project. Without their constant motivation it would have been next to impossible to carve this project to completion. We would like to thank Prof. Neha R Soni HOD CE Department who is our internal guide and gave us the opportunity to grow as software professionals and helped us to acquire such a valuable experience. Also she enthusiastically guided us and painstakingly followed up on our progress. Moreover we wish to acknowledge his unwavering support through the tenure of a degree program. But most of all we would like to thank him for believing in our abilities. Also, we would like to thank Prof. Keyur Suthar for encouraging us to think something out of the box. Finally, we would also like to thank other faculty members who provided us with moral support during the course of this project.

Acknowledgement

Nothing can be done perfectly without a proper guidance. No task can be fulfilled in a land of selfish takers. We are highly indebted to all the people involved in the development of the project. Without their constant motivation, it would have been next to impossible to carve this project to completion.

We would like to thank Prof. Neha R Soni (H.O.D C.E. Department) who is our internal guide and gave us the opportunity to grow as software professionals and helped us to acquire such a valuable experience. Also, she enthusiastically guided us and painstakingly followed up on our progress. Moreover, we wish to acknowledge his unwavering support through the tenure of a degree program. But most of all we would like to thank him for believing in our abilities.

Also, we would like to thank Prof. Keyur Suthar, for encouraging us to think something out of the box.

Finally, we would also like to thank other faculty members who provided us with Moral support during the course of this project.

Abstract

The purpose of this project is to develop a 3D, third person, RPG game which will have certain functionality like a player will control a character in an environment with A.I bots and powers which will help the player to fight tough battles. With help of blueprints we will provide the functions of our A.I and game logic and give output as according to our story-line. The software being used for this is going to be the unreal engine.

The Unreal Engine is a game engine developed by Epic Games, first showcased in the 1998 first-person shooter game Unreal. Although initially developed for first-person shooters, it has been successfully used in a variety of other genres, including stealth, fighting games, MMORPGs, and other RPGs. With its code written in C++, the Unreal Engine features a high degree of portability and is a tool used by many game developers today, with it being source-available. The most recent version is Unreal Engine 4, which was released in 2014.

The hardest part of making a game has always been the engineering. In times past, game engineering was mainly about low-level optimization—writing code that would run quickly on the target computer, leveraging clever little tricks whenever possible. But in the past ten years, games have ballooned in complexity. Now the primary technical challenge is simply getting the code to work to produce an end result that bears some semblance to the desired functionality. To the extent that we optimize, we are usually concerned with high-level algorithmic choices. There’s such a wide variety of algorithms to know about, so much experience required to implement them in a useful way, and so much work overall that just needs to be done, that we have a perpetual shortage of qualified people in the industry. Making a game today is a very different experience than it was even in 1994. Certainly, it’s more difficult.

TABLE OF FIGURES

Table of Contents

CHAPTER I: INTRODUCTION

1.1 SODH Yatra

Over the past few years, we have learnt many programming concepts, paradigms and languages. From the low-level languages i.e. assemble language to higher level languages like C, C++, Java. After doing some projects on these languages, we started to look for applications other than the conventional software development applications in which these can be used.

The gaming industry has shown much promise in the recent years and is sure last a long time in the IT sector and provide a great future to those aspiring to be a part of this industry for their careers. The computer and video game industry has grown from focused markets to mainstream. They took in about US$9.5 billion in the US in 2007, 11.7 billion in 2008, and 25.1 billion in 2010 (ESA annual report).

Though it has not been that much in trend in India, U.S. is home to major game development companies such as Activision Blizzard (Call of Duty, World of Warcraft), Electronic Arts (FIFA, Battlefield, Mass Effect), and Take-Two Interactive (Civilization, NBA 2K series, Grand Theft Auto). ZeniMax Media (Doom, Fallout, The Elder Scrolls) is the world's largest privately held video game company. Niantic (Ingress, Pokémon Go) is a notable mobile game developer.

Valve Corporation operates Steam, the largest computer gaming platform, with an active user base (~125 million) that rivals the combined user bases of the current console generation (~150 million). While not specifically focused on games, the largest mobile gaming platforms are operated by Google (Google Play), and Apple Inc.(App Store), with the majority of mobile revenue coming from Asia. Microsoft operates Xbox, the only major game console hardware franchise not controlled by a Japanese company. Sony established Sony Interactive Entertainment in Silicon Valley to run PlayStation, the world's largest and longest-running video game console franchise.

It has been in our minds for some time and with the sudden growth in this sector we were intrigued about how to apply the knowledge that we have gained up till now for game development. We found that these games are nothing but the applications of simple state diagrams, a programming language and some graphic designing.

This is when we thought of giving it a shot and took game development as our final
year project.

CHAPTER II: REQUIREMENT GATHERING

2.1 Functional Requirements

- The game follows a certain story line where the player must complete a set of scripted tasks sequentially in order to complete the game. The tasks range from solving puzzles, defeating foes, finding path to some secret locations etc. The story of the game unfolds in parts as the tasks assigned to its previous part gets completed.
- Initially when a game starts, a character with a weapon (Father's Sword "Kyuborichiku") is found. A character consists of various attributes such as Health, Agility, Strength to be 100 and an Armor of value 5. Here the player is introduced with basic movements and actions he can perform by pressing keys. These are:
  - W – Move Forward
  - S – Move Backward
  - A – Move Left
  - D – Move Right
  - Left Mouse Button – Light Attacks / Draw Sword
  - Right Mouse Button – Heavy Attacks / Draw Sword
  - Q – Sheath Sword
  - Left Shift – Evade (when sword not sheathed)
  - Spacebar – Jump
  - E – Interact
- The enemies should attack the player when he comes in a perimeter around the enemy. The players attack should cause damage to the enemy based on his attributes and enemy attacks should cause the player's health and armour to be decreased.
- As Boss 1 gets defeated, a player increases with its attributes such as Health by 30, Agility by 10, Armor by 1. Strength of a character remains constant in all the planets.
- Planet 2 portal will also get unlocked parallelly.
- As Boss 2 gets defeated, a player increases with its attributes such as Health by 70, Strength by 10, Armor by 1. Final Planet portal will also get unlocked parallelly.
- Now comes the Final Boss, as soon it gets defeated the game ends. After story and credits will be displayed on the screen.
- There must be an autosave feature in the game which:

        - As a player progresses in the game, it should be saved.
        - As a player progresses defeating first Boss in a game, it should be saved.
        - As a player progresses defeating second Boss in a game, it should be saved.
    - Meanwhile if a player is dead then "Game Over" should be displayed on screen. Game will directly get restarted from previously saved checkpoint. If the data is not saved then new game will be started.

2.2 Non-Functional Requirements

Entertaining

- The game should be exciting and thrilling for the player.
- The AI bosses should give the player hard time defeating them.
- The puzzles should be challenging for the player.
- The battle simulation should be realistic and in perfect synchronization.

Performance

- Game should run in low specification at minimum 60fps.
- It should be compatible with latest updating of graphic drivers.
- Real time response of player character with key input.

Robust

- It should run smoothly during multiple transition of game logics and level loading.
- Rendering should be done parallel with respect to how much system can handle.

Reliability

- Updated engine game code should be compatible with the older version of game.
- Game should run in any resolution display.

2.3 Hardware Requirement

- For development
    1. Processor: i7 5th Gen
    2. Ram: 8GB
    3. HDD: 1TB
    4. GPU: Geforce 920 m 4GB

- For running
    1. Processor: i5 Intel
    2. Ram: 8GB
    3. HDD: 10GB
    4. GPU: Geforce 920 m

2.4 Software Requirement

1. Unreal engine 4.
2. GPU drivers.
3. Blueprints supported IDE (included with UE4).
4. OS like windows, Linux etc.

CHAPTER III: Feasibility Study

3.1 Technical Study

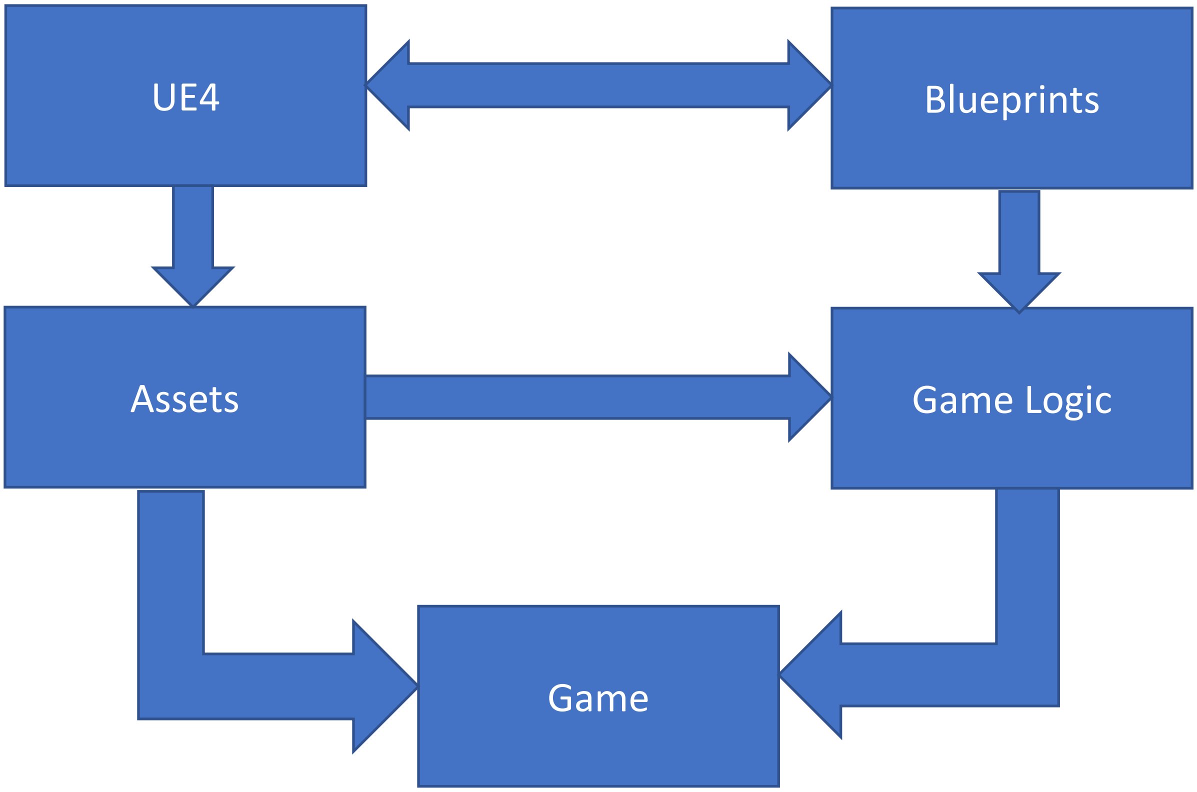
UE4

UE4 provides the basic assets contents and level design asset for our game design. It provides different types of assets such as landscape, game object, particle system etc.

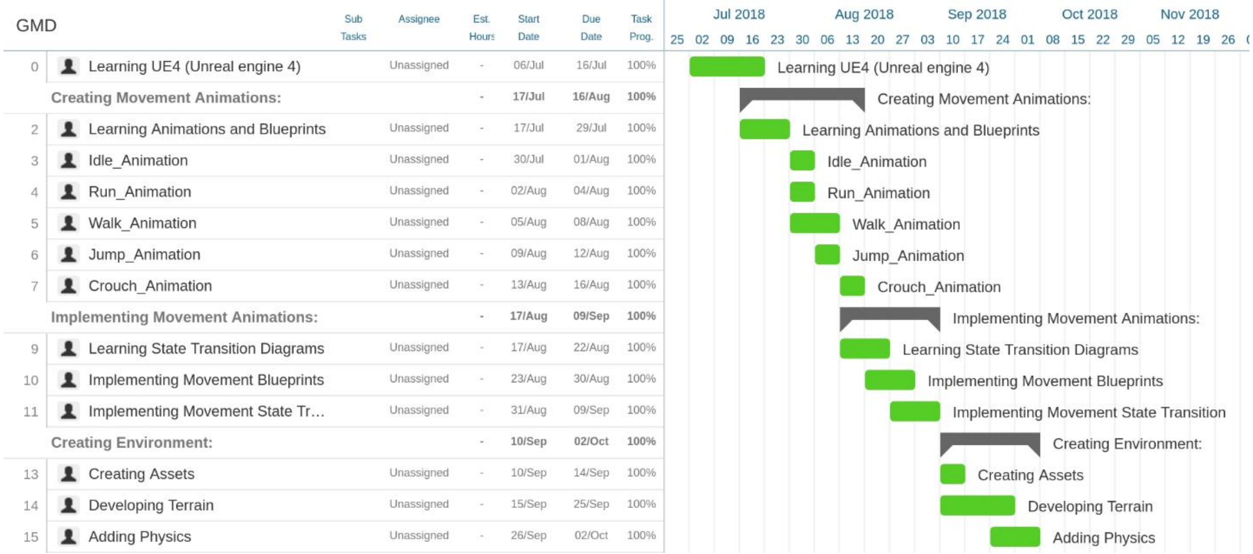
Blueprints

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.

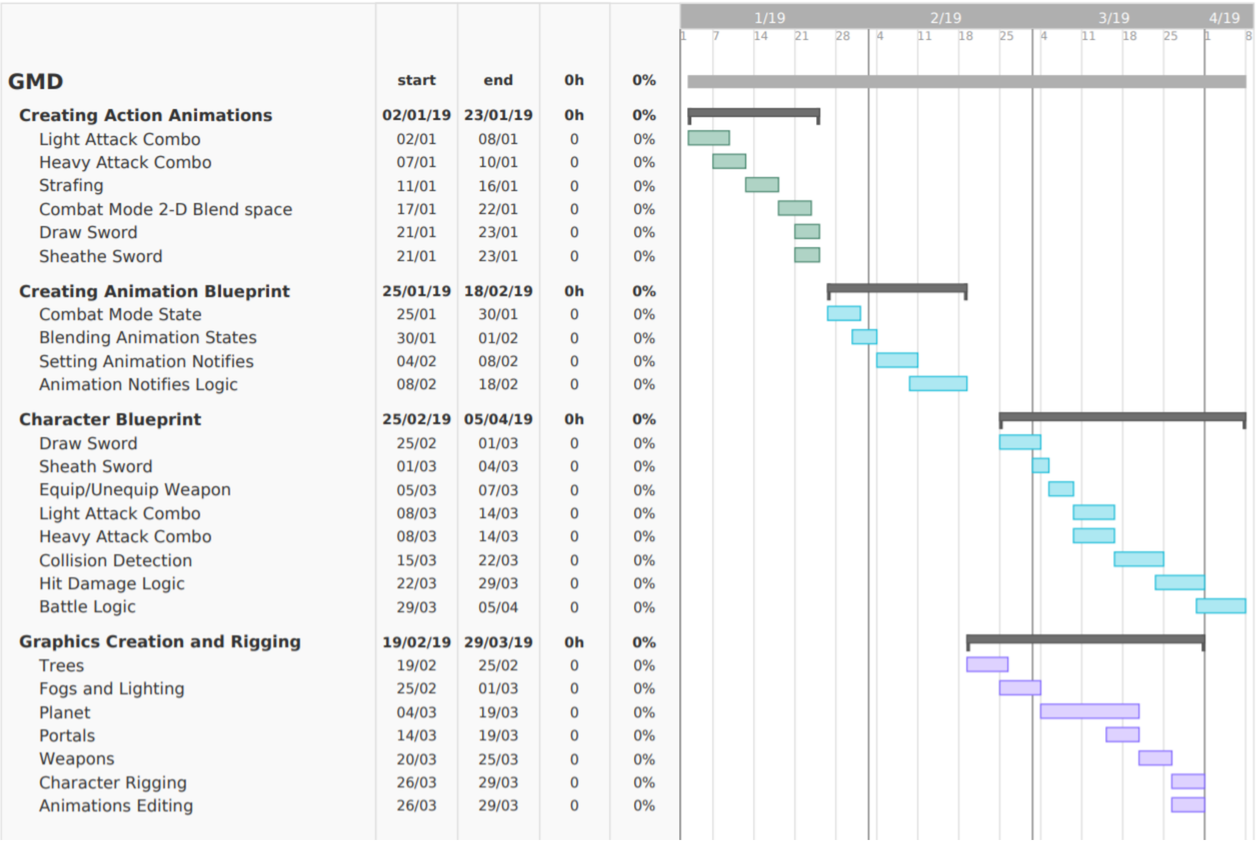
Blueprints provide the functionality of providing game logic and helps to assemble the game assets provided by engine.



3.2 Time Feasibility



	GMD	Sub Tasks	Assignee	Est. Hours	Start Date	Due Date	Task Prog.
0	Learning UE4 (Unreal engine 4)		Unassigned	-	06/Jul	16/Jul	100%
	Creating Movement Animations:			-	**17/Jul**	**16/Aug**	**100%**
2	Learning Animations and Blueprints		Unassigned	-	17/Jul	29/Jul	100%
3	Idle_Animation		Unassigned	-	30/Jul	01/Aug	100%
4	Run_Animation		Unassigned	-	02/Aug	04/Aug	100%
5	Walk_Animation		Unassigned	-	05/Aug	08/Aug	100%
6	Jump_Animation		Unassigned	-	09/Aug	12/Aug	100%
7	Crouch_Animation		Unassigned	-	13/Aug	16/Aug	100%
	Implementing Movement Animations:			-	**17/Aug**	**09/Sep**	**100%**
9	Learning State Transition Diagrams		Unassigned	-	17/Aug	22/Aug	100%
10	Implementing Movement Blueprints		Unassigned	-	23/Aug	30/Aug	100%
11	Implementing Movement State Tr…		Unassigned	-	31/Aug	09/Sep	100%
	Creating Environment:			-	**10/Sep**	**02/Oct**	**100%**
13	Creating Assets		Unassigned	-	10/Sep	14/Sep	100%
14	Developing Terrain		Unassigned	-	15/Sep	25/Sep	100%
15	Adding Physics		Unassigned	-	26/Sep	02/Oct	100%



GMD	**start**	**end**	**0h**	**0%**
Creating Action Animations	**02/01/19**	**23/01/19**	**0h**	**0%**
Light Attack Combo	02/01	08/01	0	0%
Heavy Attack Combo	07/01	10/01	0	0%
Strafing	11/01	16/01	0	0%
Combat Mode 2-D Blend space	17/01	22/01	0	0%
Draw Sword	21/01	23/01	0	0%
Sheathe Sword	21/01	23/01	0	0%
Creating Animation Blueprint	**25/01/19**	**18/02/19**	**0h**	**0%**
Combat Mode State	25/01	30/01	0	0%
Blending Animation States	30/01	01/02	0	0%
Setting Animation Notifies	04/02	08/02	0	0%
Animation Notifies Logic	08/02	18/02	0	0%
Character Blueprint	**25/02/19**	**05/04/19**	**0h**	**0%**
Draw Sword	25/02	01/03	0	0%
Sheath Sword	01/03	04/03	0	0%
Equip/Unequip Weapon	05/03	07/03	0	0%
Light Attack Combo	08/03	14/03	0	0%
Heavy Attack Combo	08/03	14/03	0	0%
Collision Detection	15/03	22/03	0	0%
Hit Damage Logic	22/03	29/03	0	0%
Battle Logic	29/03	05/04	0	0%
Graphics Creation and Rigging	**19/02/19**	**29/03/19**	**0h**	**0%**
Trees	19/02	25/02	0	0%
Fogs and Lighting	25/02	01/03	0	0%
Planet	04/03	19/03	0	0%
Portals	14/03	19/03	0	0%
Weapons	20/03	25/03	0	0%
Character Rigging	26/03	29/03	0	0%
Animations Editing	26/03	29/03	0	0%

3.3 Economic Feasibility

The UE4 is an open source software and most of the assets needed for a game are available at free of cost from the market as well.

3.4 Software description

The Unreal Engine is a game engine developed by Epic Games, first showcased in the 1998 first-person shooter game Unreal. Although primarily developed for first-person shooters, it has been successfully used in a variety of other genres, including stealth, MMORPGs, and other RPGs. With its code written in C++, the Unreal Engine features a high degree of portability and is a tool used by many game developers today. It has won several awards, including the Guinness World Records award for "most successful video game engine."

3.5 Hardware description

1. Processor: - i7 intel. 6900HQ as unreal engine needs very power full octa core processor i7 of the 6 series are needed for the running and development the game.

2. Ram: - 8 GB. The unreal engine needs 8gb as the need massive amounts memory are need

3. HDD: - 10 GB. The unreal engine requires 10gb or more for installing the whole software.

4. GPU: - 960m 4 GB. For the rendering 900 series are required for the hardware and any more powerful GPU.

3.6 Component description

Unreal engine The Unreal Engine is a game engine developed by Epic Games, first showcased in the 1998 first-person shooter game Unreal. Although primarily developed for first-person shooters, it has been successfully Blueprint editor 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.

CPU A central processing unit (CPU) is the electronic circuitry within a computer that carries out the instructions of a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions.

GPU A graphics processing unit (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. GPUs are used in embedded systems, mobile phones, personal computers, workstations, and game consoles.

CHAPTER IV: System Design

4.1 Game Plan

Once upon a time, there was an old man preaching his grandson the lessons of life. Suddenly the child ingenuously asked about his father to the old man. He was quite obstinate this time for an answer. It had been ages since the child last saw his father. The old man could no longer avoid the child’s questions and told him the truth. Giving the child, his father’s sword the old man started to tell the child, the tale of a gallant warrior who was martyred battling the fiends from the other world who now had taken over their world. The boy kept hearing glaring at the sword without uttering a single word. There was a hint of terror on the child’s face as the man spoke. The man went on with the tale and told the child that apart from their world the mysterious creature had taken over two other planets, once a nature’s bounty, that have become desolate in the rule of the fiends. The scared timid little boy never spoke about his father ever after. But there was a hidden power in the boy. Destiny had chosen the boy to be the one who will cast judgement upon the dark power. The boy grows to be a fine young man, unknown about his hidden power, sets out on the unprecedented journey to avenge his father and bring back peace to the world.

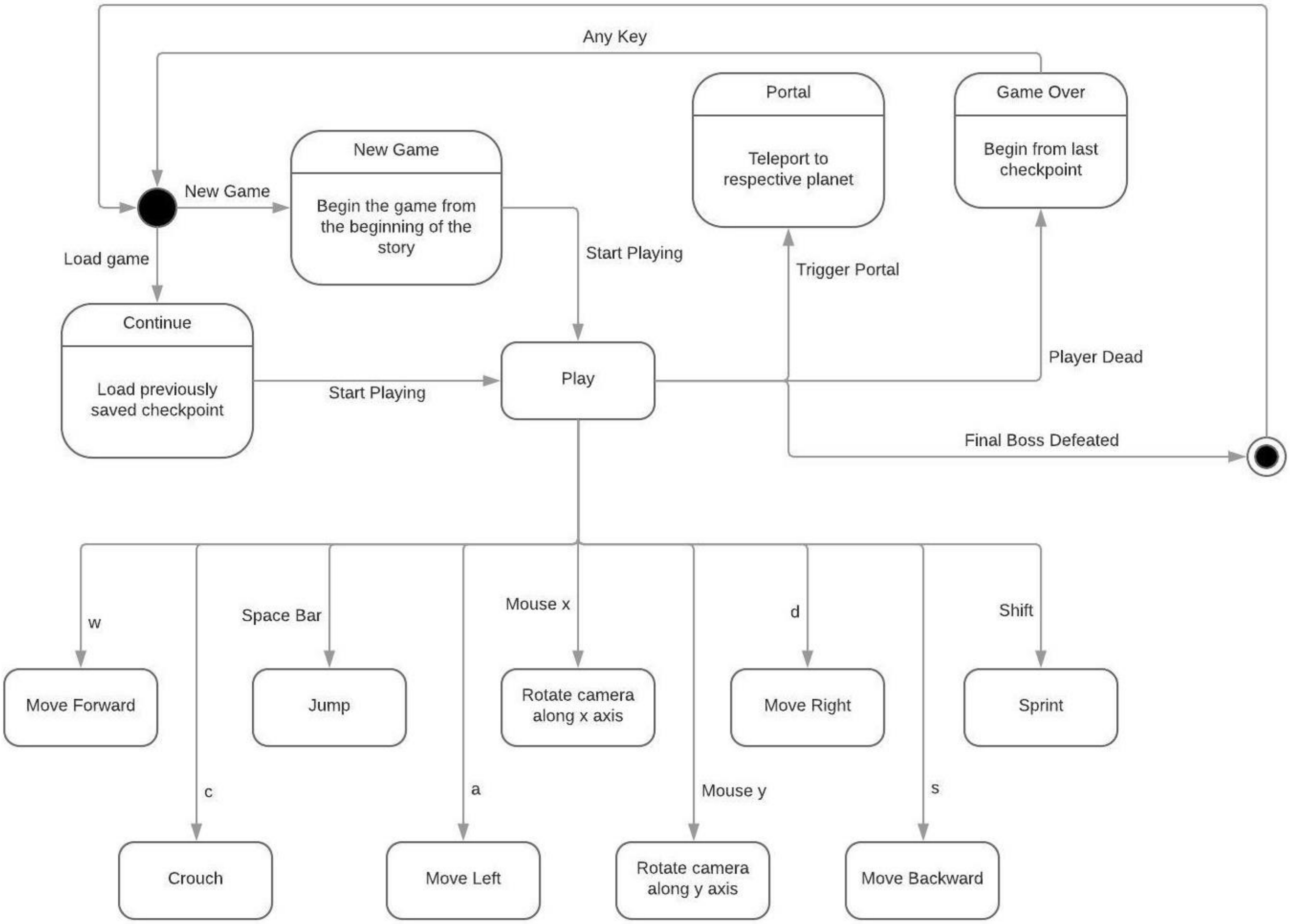
The game begins here. En route to avenge his father, the player must find the fiends of each planet and defeat them in order to unlock his potential. Each time he defeats them, he acquires an ability which eventually makes him stronger. Also, the player can defeat a particular fiend more than once to get even stronger. But each time the boss is defeated the difficulty to beat him increments depending on the player’s power.

As the story unfolds, the truth of the fiends is revealed to the boy. It turns out that these fiends are interconnected to each other and can only be stopped from being regenerated by defeating the source of their power. This source is nothing but the creator of these fiends known as “Abaddon” The King of the underworld.

There is a hidden chamber to reach the final boss, which unlocks only after the prior 3 demons have been defeated at least once. Then the player is free to go to this chamber and challenge the Demon King whenever he wishes. But the final boss will be pretentiously made harder to defeat in the first attempt. It will not be defeated easily in the first try, so the player will have to go back to the earlier planets, defeat the demons to get more stronger and then try again to beat the final boss.

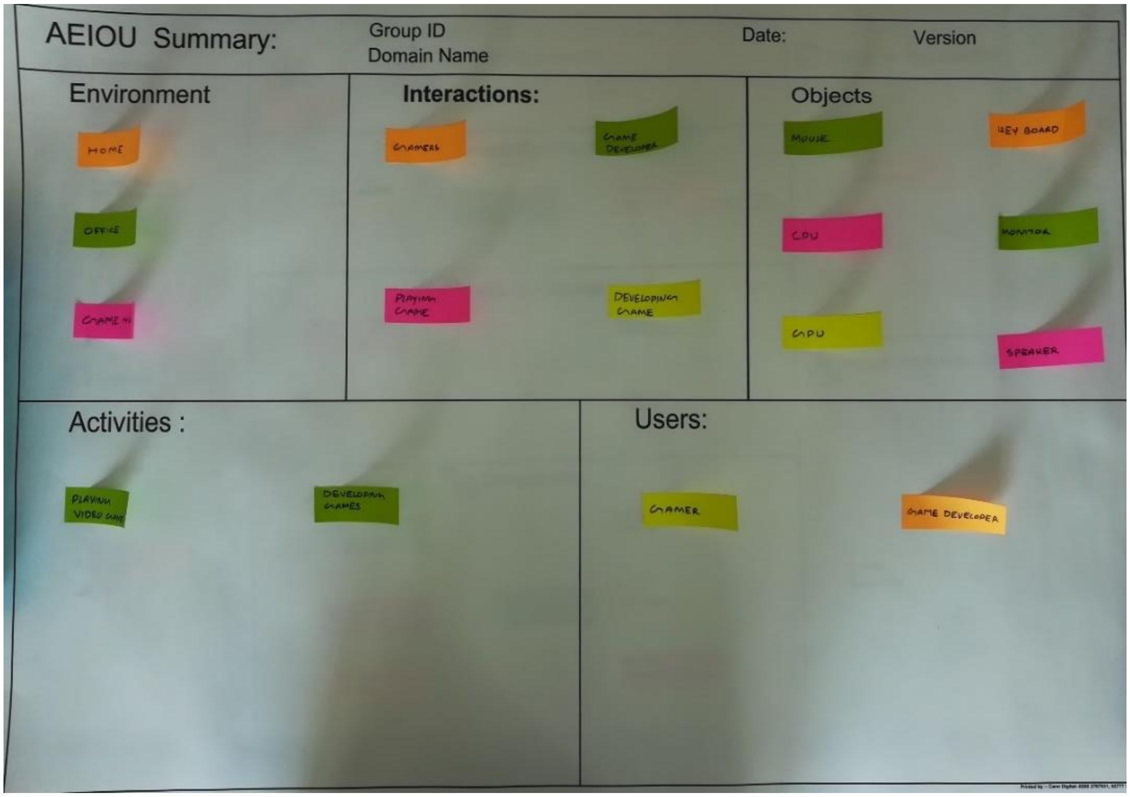
CHAPTER V: System Diagrams

5.1 State Diagram



CHAPTER VI: Canvas

6.1 AEIOU Canvas



AEIOU Summary:

Group ID

Domain Name

Date:

Version

Environment

HOME

OFFICE

GAME H

Interactions:

GAMERS

GAME DEVELOPER

PLAYING GAME

DEVELOPING GAME

Objects

MOUSE

KEY BOARD

CPU

MONITOR

GPU

SPEAKER

Activities :

PLAYING VIDEO GAME

DEVELOPING GAMES

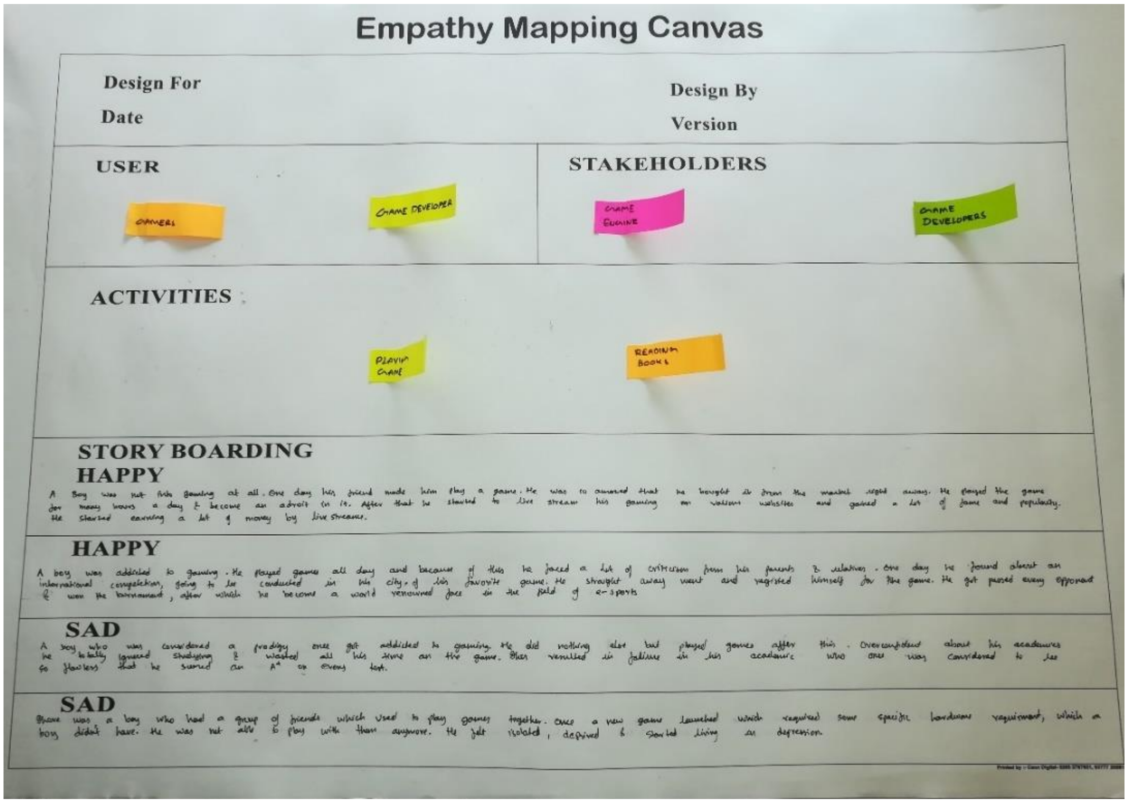
Users:

GAMER

GAME DEVELOPER

Fig. 6.1

6.2 Empathy Making Canvas



Empathy Mapping Canvas

Design For

Date

Design By

Version

USER

GAMERS

GAME DEVELOPER

STAKEHOLDERS

GAME ENGINE

GAME DEVELOPERS

ACTIVITIES :

PLAYING GAME

READING BOOKS

STORY BOARDING

HAPPY

A Boy was not into gaming at all. One day his friend made him play a game. He was so amazed that he bought it from the market right away. He played the game for many hours a day & become an adroit in it. After that he started to live stream his gaming on various websites and gained a lot of fame and popularity. He started earning a lot of money by live streams.

HAPPY

A boy was addicted to gaming. He played games all day and because of this he faced a lot of criticism from his parents & relatives. One day he found about an international competetion, going to be conducted in his city, of his favorite game. He straight away went and registed himself for the game. He got passed every opponent & won the tournament, after which he become a world renowned face in the field of e-sports

SAD

A boy who was considered a prodigy once got addicted to gaming. He did nothing else but played games after this. Overconfident about his academics he totally ignored studying & wasted all his time on the game. This resulted in failure in his academic who once was considered to be so flawless that he scored an A+ on every test.

SAD

There was a boy who had a group of friends which used to play games together. Once a new game launched which required some specific hardware requirment, which a boy didn't have. He was not able to play with them anymore. He felt isolated, deprived & started living in depression.

Fig. 6.2

6.3 Ideation Canvas

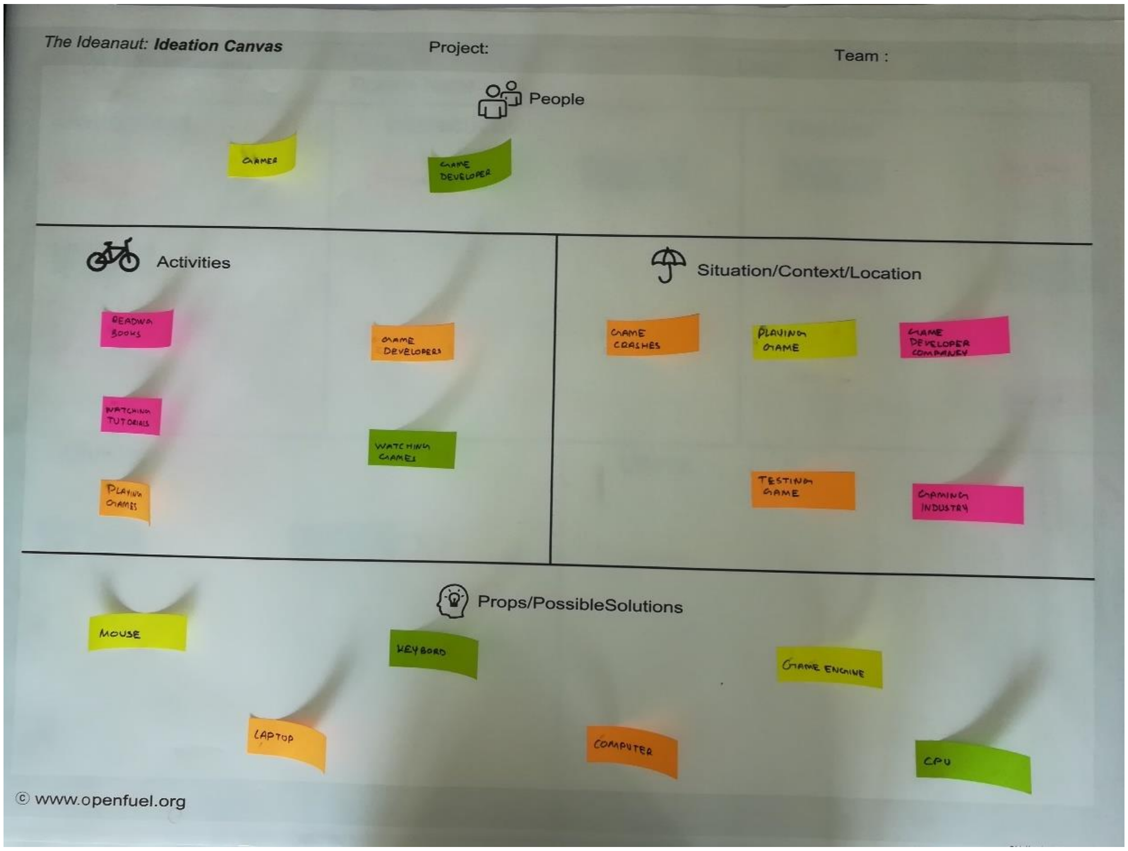


Fig. 6.3

6.4 Product Development

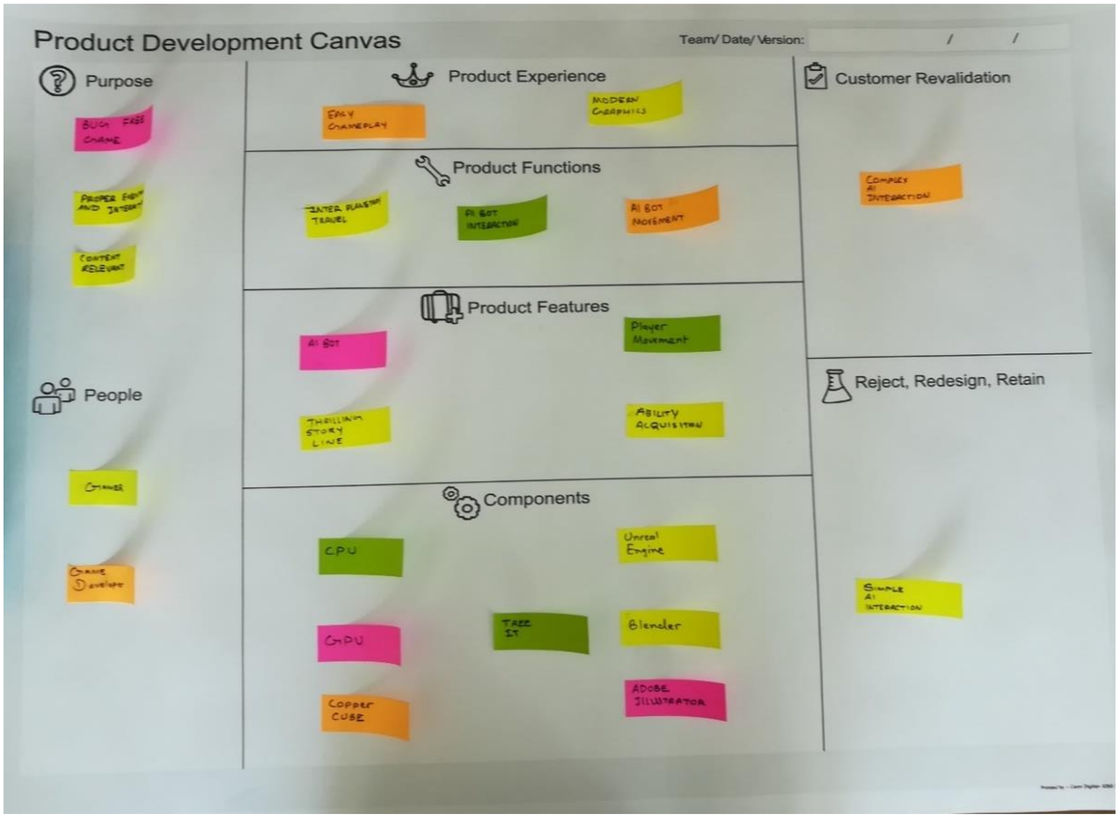
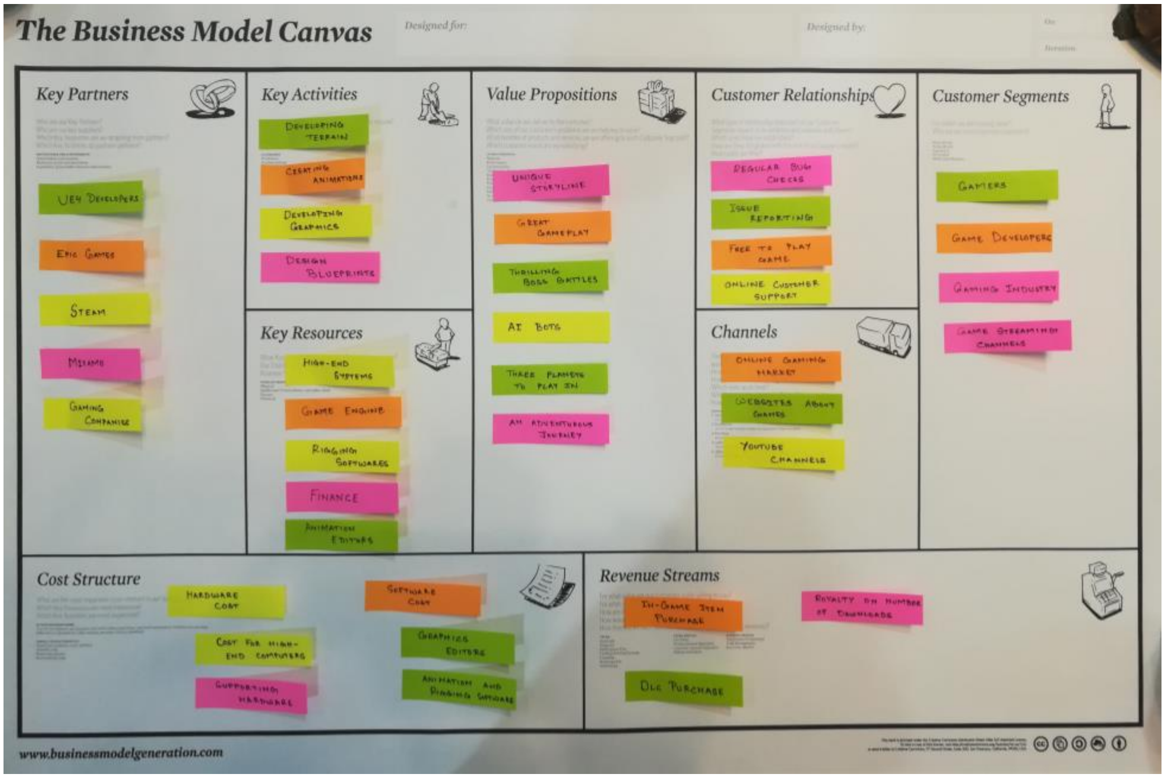


Fig. 6.4

6.5 Business Model Canvas



The Business Model Canvas

Designed for:

Designed by:

On:

Iteration:

Key Partners

UE4 DEVELOPERS

EPIC GAMES

STEAM

MIXAMO

GAMING COMPANIES

Key Activities

DEVELOPING TERRAIN

CREATING ANIMATIONS

DEVELOPING GRAPHICS

DESIGN BLUEPRINTS

Key Resources

HIGH-END SYSTEMS

GAME ENGINE

RIGGING SOFTWARES

FINANCE

ANIMATION EDITORS

Value Propositions

UNIQUE STORYLINE

GREAT GAMEPLAY

THRILLING BOSS BATTLES

AI BOTS

THREE PLANETS TO PLAY IN

AN ADVENTUROUS JOURNEY

Customer Relationships

REGULAR BUG CHECKS

ISSUE REPORTING

FREE TO PLAY GAME

ONLINE CUSTOMER SUPPORT

Channels

ONLINE GAMING MARKET

WEBSITES ABOUT GAMES

YOUTUBE CHANNELS

Customer Segments

GAMERS

GAME DEVELOPERS

GAMING INDUSTRY

GAME STREAMING CHANNELS

Cost Structure

HARDWARE COST

COST FOR HIGH-END COMPUTERS

SUPPORTING HARDWARE

SOFTWARE COST

GRAPHICS EDITORS

ANIMATION AND RIGGING SOFTWARE

Revenue Streams

IN-GAME ITEM PURCHASE

DLC PURCHASE

ROYALTY ON NUMBER OF DOWNLOADS

www.businessmodelgeneration.com

Fig. 6.5

CHAPTER VII: TESTING

7.1 Movement Testing

Player movement testing:

- The player is in idle state when no valid input for movement is given. In this state, the idle animation is played.

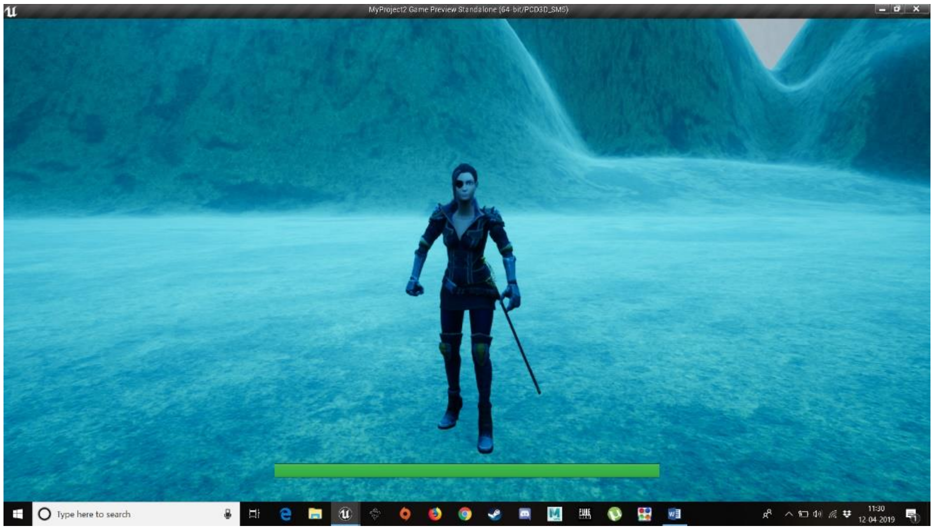


Fig. 7.1.1

- The player only moves on valid movement inputs. When pressed a respective key for movement, the transition from idle to run is smooth. No frame loss or glitch detected for forward, backward, left and right movement.
- The rotation orientation is set to player movement.

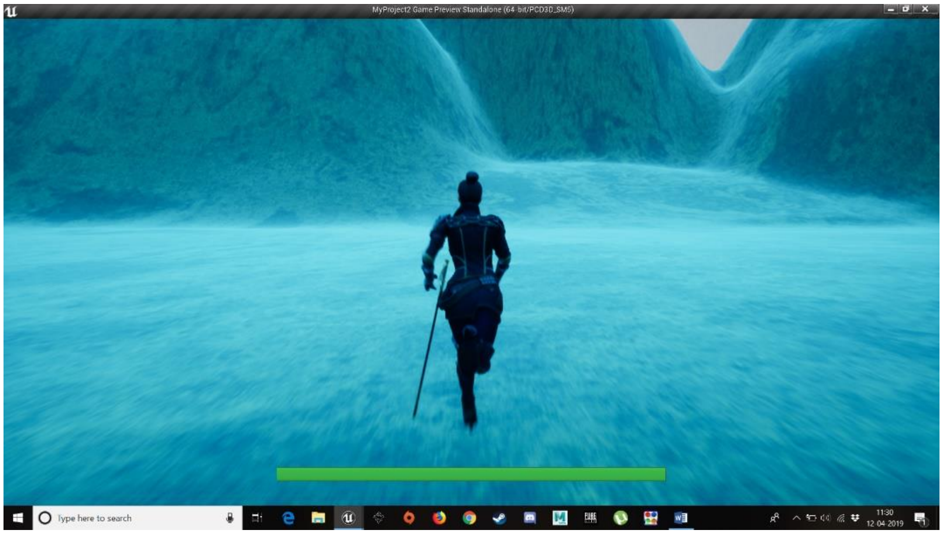


Fig. 7.1.2

- When space bar is pressed, the player enters into the jump state. The three animations assigned in this state get sequentially played as the character performs a jump.
- While character is in air, she can perform actions such as attacks, but all inputs for movements i.e. (w, a, s, d) have no effect.
- The transition between the animations is smooth and no delays or lags are encountered.

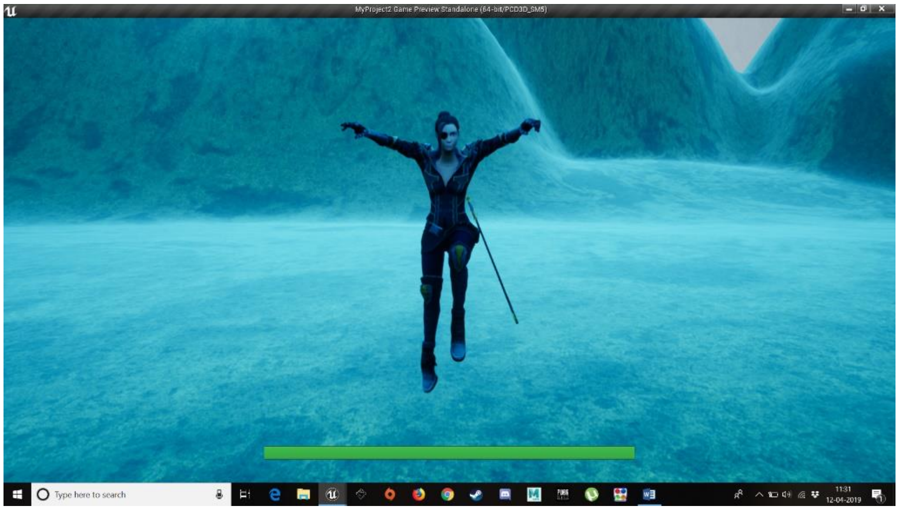


Fig. 7.1.3

7.2 Combat Mode Testing

Combat mode idle state:

- When the sword is drawn, the character enters in the combat mode animation state. The stance changes, the rotation is not oriented to movement anymore. Rotation controller is set to “Yaw”.
- When no input is given in this state, the idle animation for combat looped until a valid input is given.

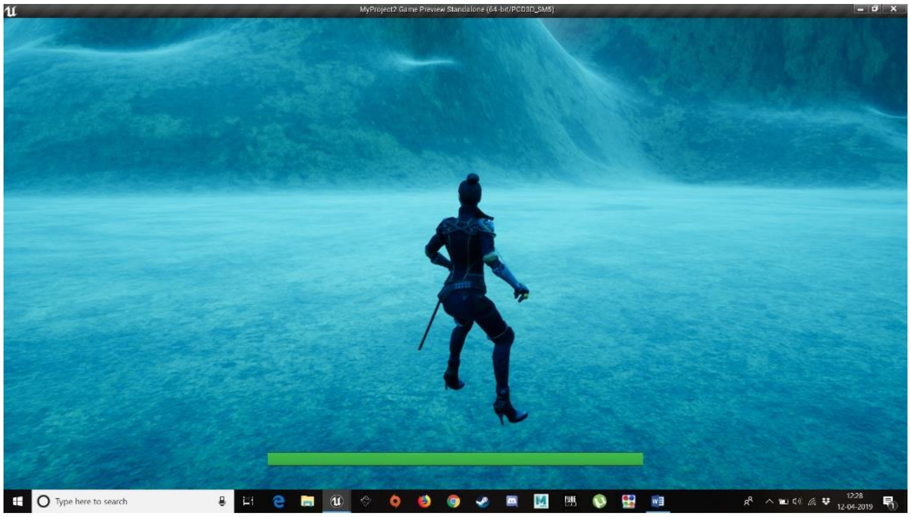


Fig. 7.2.1

- The movement style and speed decreases. The character starts strafing instead of running.

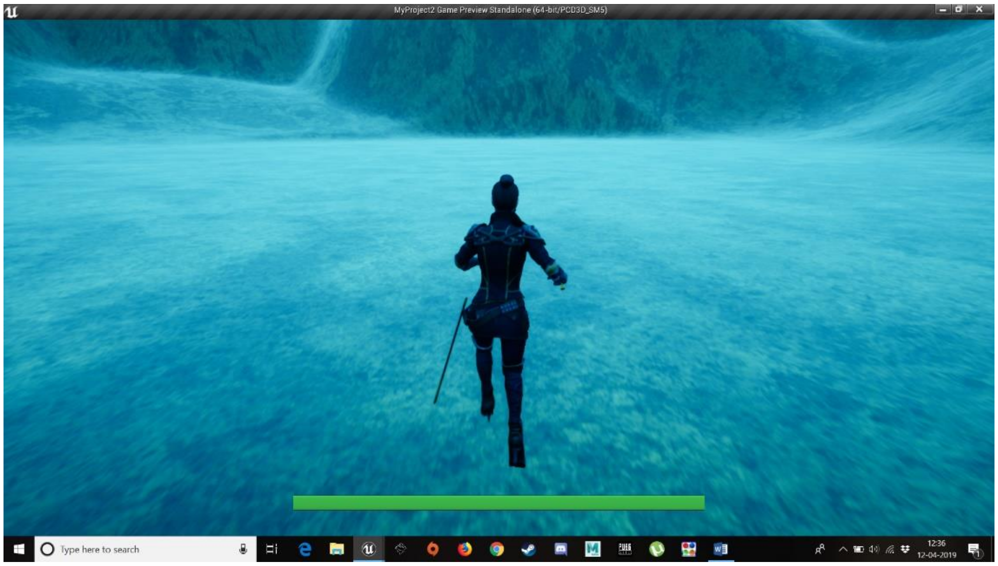


Fig. 7.2.2

- This is the state where the attacks can be performed by the character.

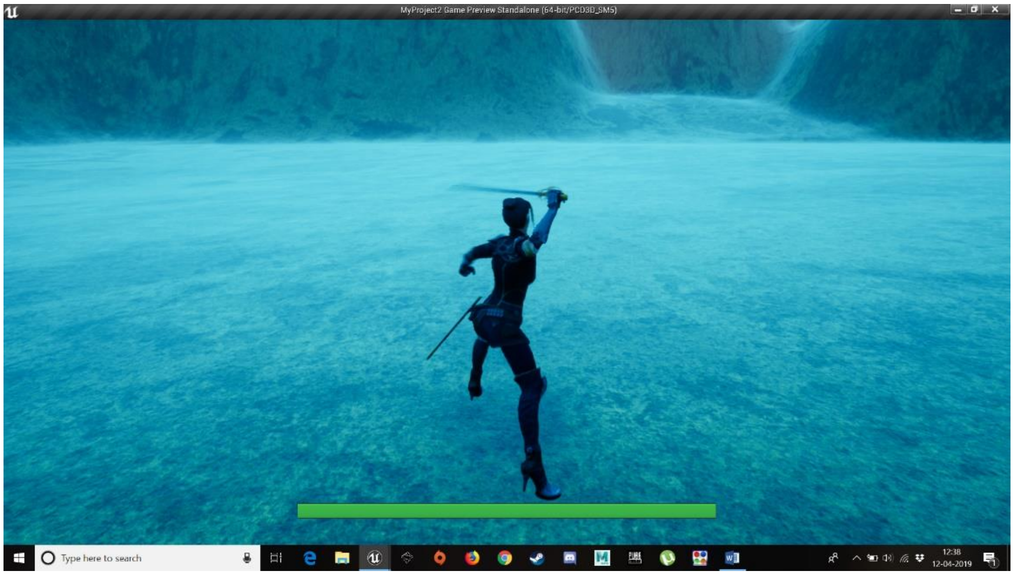


Fig. 7.2.3

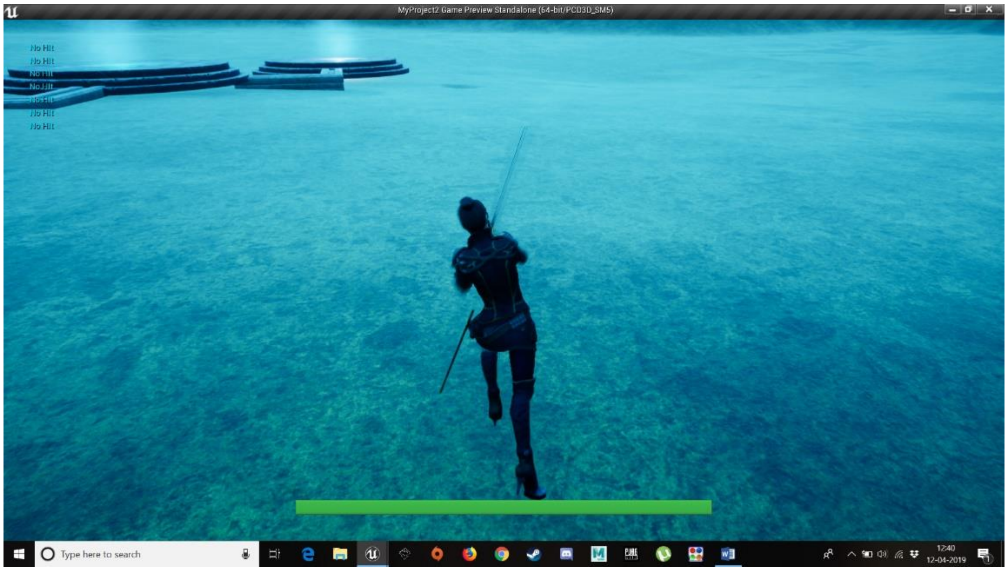


Fig. 7.2.4

7.3 Weapon Collision and Damage Testing

- The weapon is covered with boxes that detect collisions. During an attack, when the weapon comes in contact with the enemy, its health decreases depending on some parameters like:
  - Character’s agility, power and strength.
  - Weapon.
  - Type of attack.
- Light attacks deal less damage compared to heavy attacks.
- The collision is detected 70% of time.
- The damage dealt also depends on the how much of the sword penetrates the enemy’s body.
- Damage after a light attack

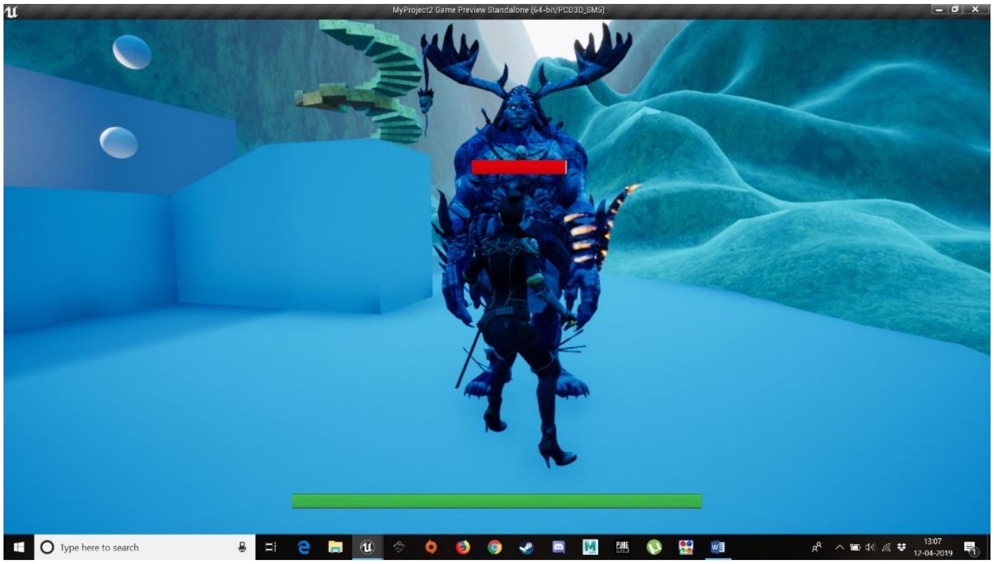


Fig. 7.3.1

- The damage dealt after a heavy attack.

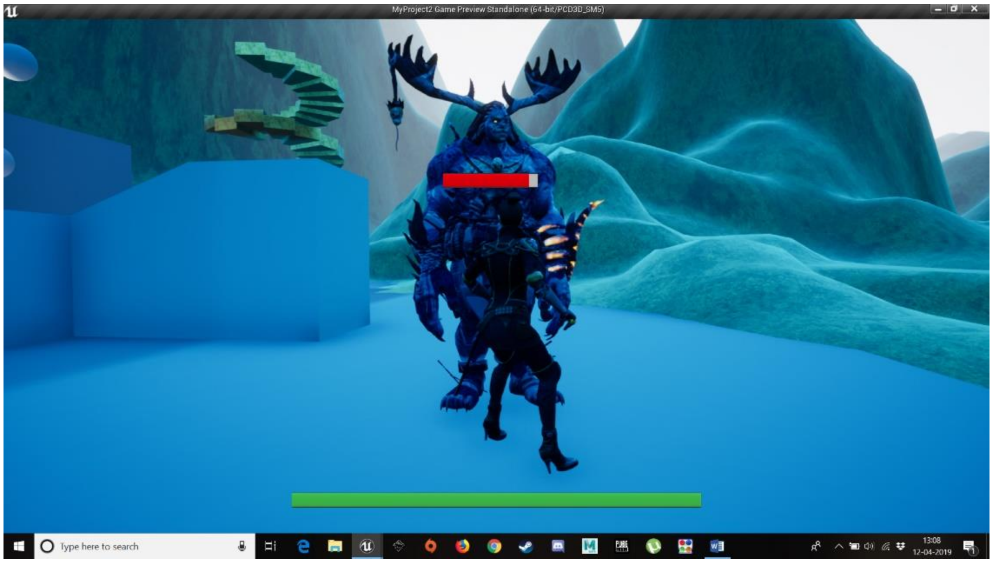


Fig. 7.3.2

7.4 Testing AI system

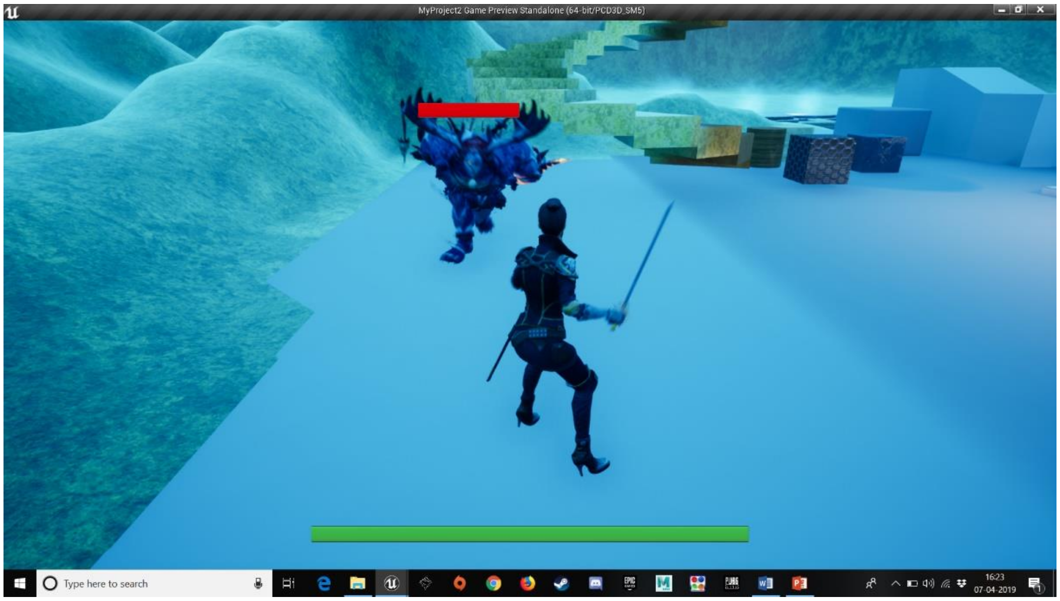


Fig. 7.4.1

The AI bots are able to detect the presence of main character and find the path to reach and attack her. Sometimes, some of the obstacles are hard for the AI bots to overcome so it stops there, but this happens only occasionally. For overcoming the obstacles, a better algorithm is needed.

Other than that, the AI works just fine.

7.5 Level Testing

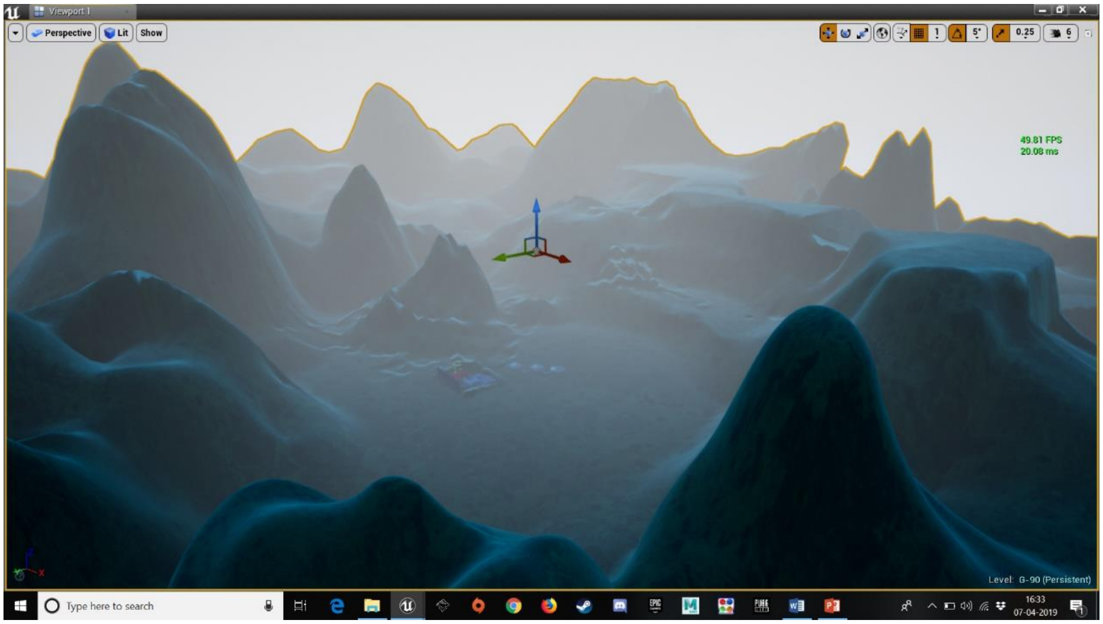


Fig. 7.5.1

The level renders at 40-40 fps with all the lightings, fogs, terrains and all the graphical elements. The environment’s ambiance is as per the requirements of the story line.

CHAPTER VIII: CONCLUSION

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. Video games themselves are a relatively modern form of entertainment. They are engaging and immersive on a level different from that of traditional board games and other forms of entertainment. The player actively contributes to the level of satisfaction he/she attains from this medium and thus is more invested and willing to engage in the elements of the video game. The amount of play time is also an important factor in the effects of gaming. Although excessive playtime can have negative consequence, gaming in moderation can be healthy, fun, and educational.

Having learnt the mechanics of the engine, its working and flow, developing assets and environment, creating animations and blueprints and implementing them through the state transition schemes, in this semester we used all the knowledge acquired during the previous semester and developed a fully functional working game.

CHAPTER IX: REFERENCE

- https://docs.unrealengine.com/latest/INT/GettingStarte d/index.html
- www.youtube.com
    - Red Blueprints
    - Titanic Games
- www.google.com