

ALONE ASSAILANT

BY

**MD. MOHSIN
ID: 183-15-2254**

AND

**IMRUL HAMID SHOHAG
ID: 183-15-2256**

This Report Presented in Partial Fulfilment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Dr. S.M. Aminul Haque
Associate Professor
Department of CSE
Daffodil International University

Co-Supervised By

Md. Mahfujur Rahman
Lecturer
Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
FEBRUARY 2023

APPROVAL

This Project titled “**Alone Assailant**”, submitted by Md. Mohsin and Imrul Hamid Shohag to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 06/02/2023.

BOARD OF EXAMINERS



Dr. Touhid Bhuiyan
Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

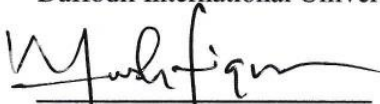
Chairman



Arif Mahmud
Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Mr. Mushfiqur Rahman
Senior Lecturer

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Mohammad Shorif Uddin
Professor

Department of Computer Science and Engineering
Jahangirnagar University

External Examiner

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Dr. S.M. Aminul Haque, Associate professor, Department of CSE, Daffodil International University.** We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



Dr. S.M. Aminul Haque
Associate Professor
Department of CSE
Daffodil International University

Co-Supervised by:

Md. Mahfujur Rahman
Lecturer
Department of CSE
Daffodil International University

Submitted by:



(Md. Mohsin)
ID: 183-15-2254
Department of CSE
Daffodil International University



(Imrul Hamid Shohag)
ID: 183-15-2256
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Dr. S.M. Aminul Haque Associate Professor**, Department of CSE, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “Data Mining, Machine Learning (ML)” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project

We would like to express our heartiest gratitude to **Dr. Touhid Bhuiyan, Professor and Head**, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

There are many games in this game world, but most of them are for the entertainment purpose. So, we made a game that people can use to learn educational stuff Game level completion. An Open world 3D desktop game where players can find different species. From math problems and MCQ questions you have to solve and switch to math. Our game's name is "Alone Assailant". The name was chosen for the game section where players can step-by-step complete game levels consisting of various math tasks. "Alone Assailant" is a game that is a structured form of play, usually done for entertainment and learning math. Developing thinking skills, decision making, understanding challenges, and real-time interaction are the key elements of this game. Therefore, this game is designed for entertainment and educational purposes both and can be used on your computer. The advantage of this type of testing indicates that students show more interest in this type of class testing. Such a concept can make leisure time more attractive and enjoyable. We know math can be too depressing when students are just starting to learn it. This is why smart students don't want to go through this complicated and tedious learning process. The main purpose of this project is to make math more fun and enjoyable, and more students are voluntarily entering this field.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
CHAPTER 1: INTRODUCTION	1-3
1.1 Introduction	1
1.2 Motivation	1-2
1.3 The Rationale of the study	2
1.4 Objectives	2
1.5 Expected Output	3
1.6 Project Finance	3
1.7 Report Layout	3
CHAPTER 2: BACKGROUND STUDY	4-6
2.1 Introduction of open world game	4
2.2 Open-world gaming relate to education	4-5
2.3 Related Works	5-6
2.4 Massive Challenges with Open-World Games	6
CHAPTER 3: REQUIREMENTS	7-17
3.1 Introduction	7
3.2 Hardware Requirements	7
3.3 Software Requirements	7
3.3.1 Unreal Engine	8-10

3.3.2 Blueprints	10-11
3.3.3 Metahuman	12-13
3.3.4 Blender	14-15
3.3.5 Mixamo	15-17
CHAPTER 4: DESIGN & SPECIFICATION	18-27
4.1 Agile methodology for Game	18-19
4.2 Frontend	19-22
4.3 Backend	22-25
4.4 Special Feature	26-27
CHAPTER 5: IMPLEMENTATION AND TESTING	28-29
5.1 Project Implementation	28
5.2 Project Testing	28-29
5.2.1 Experiment of game appliance	29
5.2.2 Bug fixing	29
5.2.3 Result	29
CHAPTER 6: IMPACT ON SOCIETY, ENVIORNMENT AND SUSTAINABILITY	30-32
6.1 Impact on Society	30
6.2 Impact on Environment	30-31
6.3 Ethical Aspects	31-32
6.4 Sustainability Plan	32
CHAPTER 7: DISCUSSION & FUTURE SCOPE	33-34
7.1 Discussion	33
7.2 Future Work	33-34
7.3 Conclusion	34
REFERENCES	35-36

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.3.1: Unreal Engine 5	8
Figure 3.3.2: Basic Blueprints Example	10
Figure 3.3.3.1: Main Character of 'Alone Assailant'	12
Figure 3.3.3.2: Mesh to Metahuman to Unreal Engine	13
Figure 3.3.4: Thickening 3D models with Blender	14
Figure 3.3.5: Mixamo 2.0 Animation	16
Figure 4.1: Agile methodology in game development	18
Figure 4.2.1: Loading Screen of game	20
Figure 4.2.2: Start menu of the game	20
Figure 4.2.3: Player Spawned in Last saved location	21
Figure 4.2.4: Player when get near to his/her vehicle	21
Figure 4.2.5: Player while driving his/her vehicle	22
Figure 4.3.1: A part of the city in backend	22
Figure 4.3.2: Whole city of our game in backend	23
Figure 4.3.3: Main Character in Backend	23
Figure 4.3.4: Non-player character people and vehicle	24
Figure 4.3.5: Vehicle Light Setup Blueprints	24
Figure 4.3.6: Vehicle Movement Blueprints	25
Figure 4.3.7: Vehicle Selection Menu	25
Figure 4.4.1: Basic Math solving station	26
Figure 4.4.2: Question of basic math	26
Figure 4.4.3: When player choose the wrong answer	27
Figure 4.4.4: When player choose the correct answer	27

LIST OF TABLES

TABLES	PAGE NO
Table 1.7: Report Layout	3
Table 5.2.1: Experiment of game appliance	29

CHAPTER 1

INTRODUCTION

1.1 Introduction:

We are working on games. In today's modern environment, games, are becoming more popular and beneficial day by day. It indicates that if we want to learn anything new, it will provide us with certain activities that are based on a learning topic during our learning periods. This includes some amusing material "If it's a game." With more complex visuals, we attempted something in mathematics. We created an Open world desktop game to assist you in learning how to solve difficulties with Puzzle and mathematics. It's called " Alone Assailant." Alone Assailant is a third-person perspective game in which players can play as that character, and complete various mission jobs. During the development process, we consider pupils and how we can amuse them with this project. We hope that by playing the game "Alone Assailant," Gamer would be able to learn simple puzzle & math while having fun. We think that whenever a learner learns anything new, he or she can memorize or understand it for a short period. However, if he/she interestingly studies a topic, he/she believes he/she can memorize or grasp that topic for a long time.

As a result, we created this style of game in which players may enjoy the game by playing about the characters story by completing game levels. This project's foundational topic is Open world environment. However, we can expand this sort of project to cover various themes by just altering the levels of difficulties in the game.

1.2 Inspiration:

There are several games accessible right now in the online stores like Steam, Epic Games. The majority of them are for enjoyment. We created an Open world desktop game in this era of video games where individuals may learn math by finishing the stages of the game. As a result, we wish to create a game where everybody can learn math in fascinated way.

We make certain that the gamers (students) learn something while having fun. We believe that if someone learns anything interesting, he or she will remember it for a long time, thus we began developing this game project with this motive in mind. We know that many individuals are bored when they play games and that playing games can lead to addiction, which is tragic for our society.

1.3 The Rationale of the study:

A huge number of pupils spend their valuable time playing various forms of games. It's entertaining, but they have no idea how much time they are wasting. However, this is making instructional games more popular and valuable in today's environment. This indicates that if we want to learn anything new, it will provide us with some assignments that are based on a learning topic with some pleasurable items throughout these learning moments. We created this one So that, Students learn something new or review content while playing this game.

1.4 Objectives:

This project's primary goals are as follows:

1. To make life easier and more fun for gamers.
2. Playing this game will help you create your first learning environment (Alone Assailant).
3. To understand the situation at that moment (variable, condition, loop & etc.)
4. Throw this project to make a fun learning experience.
5. To have various levels with a variety of quiz questions and difficulties to complete.

1.5 Expected Output:

Because this is an exploratory project, our primary goal was to establish the fun learning platform on a related topic. We feel that, this is the greatest technique to create an enviable learning platform. So, we expect that if any of our pupils learn something new, he or she will be able to memorize or grasp it for a short period. However, if he or she

enjoys the game, he or she believes he or she can memorize or grasp that topic for a long time. The desired effect is that the instructor will encourage this topic of study.

1.6 Project Finance:

We think that one day, all learning resources will be free. Anyone who wants to learn anything through the internet or another platform can do so. So, from this perspective, we created this gaming project. This is a fully free Open world computer Game. Because this is based on our own experiences and thoughts, as well as knowledge gained from other sources, it is an Open world story type of game. If we want to play such paid games, we must pay to do so.

1.7 Report Diagram:

The following table 1.7 shows that, we discussed about the chapter's description. That means, in which chapter what we had included and the discussion about all of them.

Table 1.7: Report Layout

Chapters	Chapter Description
01	Introduction, Inspiration, The coherent of the study, Objectives, Expected Output, Project Finance, Report Diagram
02	Introduction of open world game, Open-world gaming relate to education, Related Works, Massive Challenges with Open-World Games
03	Introduction, Hardware Requirements, Software Requirements
04	Agile methodology for Game, Frontend. Backend
05	Project Implementation, Project Testing
06	Discussion, Future Work, Conclusion

CHAPTER 2

BACKGROUND STUDY

2.1 Introduction of an open-world 3D game:

“In video games, an associate degree open world could be a virtual world within which the player will approach objectives freely, as against a world with a lot of linear and structured gameplay. whereas games have used open-world styles since the Eighties, the implementation in theft machine III (2001) set a regular for the construct that has been used since.

The games with an open or free-roaming worlds generally lack level structures like walls and latched doors or invisible walls in additional open areas that stop the player from dare on the far side them; solely at the place of associate degree open-world games can players be restricted by geographic options like large oceans or untraversable mountains. Players generally don't assignation loading screens usual in linear level styles once motile concerning the sports world, with the open-world game victimization strategic storage and memory techniques to load the sports world in an exceedingly dynamic and seamless manner. An open-world 3D games still enact several narrowness within the game setting, either thanks to great limitation of technical methods or in-game limitations imperative by a game's dimensionality”.

“Although the sincerity of the game world is an important prospect of open-world games, as long as a player with autonomy is the main draw – that’s mean not to such a great extent the freedom to do whatever a player will want in the game (which is nearly impossible with current computing technology), but the ability to approach the game and its challenges in the order and manner that the player desires while still constrained by gameplay rules”.

2.2 Open-world gaming relate to education:

Independent inquiry and discovery are vital parts of a child's developing mind long before she or he reaches the school setting. Brown University, A study discovered that

allowing young infants to connect with their surroundings autonomously first, rather than merely witnessing someone else accomplish a task and then participating in some type of interaction, results in improved accuracy in learning causal structures. Indeed, self-motivated exploration appears to play an important role in learning and development, as it helps youngsters to make more intimate connections with their settings and the information they are actively processing.

Given the emphasis on customized ways of discovery and engagement in open-world games, it appears that open-world settings have the potential to give young students the opportunity to explore and learn at their own speed in a personalized manner. When students interact with an educational open-world game, they are essentially given the option to participate in the organic discovery processes that they should actively engage in as part of the learning experience.

The visual character of video games as instructional aids is perhaps the most essential feature of open-world gaming in education. Visual learners, or those who learn best by actively imagining the content they encounter, make up around 65 percent of the population. While visual learners may struggle to keep up with a spoken lecture or conversation, they frequently thrive at more interactive tasks, making video games an important instructional tool for this student demographic. Teachers and parents may design more effective learning techniques that are customized to each student's specific learning style by embracing the active and autonomous exploration that is encouraged in open-world gaming.

2.3 Related Works:

1. "**Grand Theft Auto 5**" is one of those games that refuses to die. What's the point? It's critically praised, profitable, and features a vibrant open world to explore. It's yet another of those games that build an open environment based on reality. It's energetic and dynamic, with non-playable characters who don't follow predetermined pathways. They are interactable, and there is always a mission to complete or something new to uncover. When you combine the ability to explore on foot, via vehicle, or in the air, the possibilities for gameplay seem limitless.

2. Quantic Dream's project, "**Detroit: Become Human**", builds on the studio's tradition of riveting, narrative-driven experiences. In a compelling, an emotional voyage of choice and consequences, discover what it truly means to be human. Set in 2036 Detroit, the city has been rejuvenated by the development and adoption of Android robots into everyday life. However, when Android robots begin to act as if they are alive, events begin to spiral out of hand. Take on the roles of the story's three major playable characters, each with their own distinct viewpoint on their new way of life.
3. **Watch Dogs**: A player take on the role of Aiden Pearce, a talented hacker and former thug whose criminal past has resulted in a horrible family tragedy. As the player searches for individuals who have wronged the player's family, he will be able to watch and hack every single one's around him by controlling everything linked to the city's network. Such as access ubiquitous surveillance cameras, download personal data to track down a target, and manage traffic lights and public transit to thwart the adversary.

2.4 Massive Challenges with Open-World Games:

There are several features of the genre that might detract from the overall quality of a game. There are several examples of games that were made only to placate the market or to maximize play. Simply said, not every game has to be open-world. When it works, it can be incredible. At times, though, it was entirely unneeded, and these are the flaws of the genre as a whole. Here are ten major issues with open-world games that we need to admit.

- Typically Have the Most Bugs and Glitches
- Disconnection between the story and the actions of the players
- The World Doesn't React Properly to What You Do
- Problem navigating
- Deviates from the Story Pacing
- Limits Storytelling Possibilities

CHAPTER 3

REQUIREMENTS

3.1 Introduction:

This is a fun project to learn new things. The main purpose of this project for players to learn math. Players must select "Start New Game" or "Load Game". "Load Game" allows the player to continue the game from the last saved game. Our target audience is students and anyone who wants to learn math in an interesting way. This project is of great interest to students and all kinds of gamers who play games for entertainment. So, there is a lot of room for this technology.

3.2 Hardware Requirements:

- Windows OS
- Processor: Intel® Core™ i5 Processor / AMD X8 FX-8350 or better
- Ram: Minimum 8 GB DDR4 RAM
- Graphics: NVIDIA GTX 660 2GB / AMD HD 7870 2GB or better
- Storage: 30 GB available space

3.3 Software Requirements:

- Unreal engine
- Blueprint
- Quixel Bridge
- Metahuman
- Blender
- Mixamo

3.3.1 Unreal Engine:

Unreal Engine is the world's most open and advanced real-time 3D creation tool which is developed by a company named Epic Games. Unreal Engines is known for photorealistic visualizations and immersive experiences. Unreal Engine is the fastest developing skill requirement for real-time and 3D graphics. The skills have to learn for succeed in game development, architecture, film and television, automotive, and so on. Unreal Engine qualify game developers and creators in every industry to create next-generation real-time 3D content and sagacity with greater liberty, clearness and elasticity than ever before. Nanite, a virtualized micro polygon geometry system, and virtual shadow maps allow you to create games and worlds with an enormous amount of geometric detail. Directly import and duplicate millions of meshes without noticeable loss of fidelity while maintaining a real-time frame rate of 60 fps. These systems intelligently stream and process only recognizable details, greatly removing polygon count and draw call limitations.

The following figure 3.3.1 shows a demo of Unreal Engine 5.



Figure 3.3.1: Unreal Engine 5

Features of Unreal Engine:

- Collision Volume - A volume like sphere, cube, capsule, and so on, that can collide within the game world (Volume Reference).
- Primitive Component- In-game world a component which contains or generates geometry. It can be rendered or used as collision data. (UPrimitiveComponent)
- Physics Simulation - In a video game physical simulation calculations and collision calculations are handled by Physics Engine System
- Shape trace - This is a kind of trace. The start points and end point defined as Trace. Between these points and collect information about objects by using shape tracks to move shapes that collide with the passing shape. (Trace)
- Animation Montage - An asset containing multiple animated assets. This asset allows you to further define the functionality of your animation, such as attaching events. (Animation Montage Overview)
- Events - Events are functions that are triggered when something specific happens. For example, when the value of a variable changes, we notify the object of the value change.
- Skeletal Mesh - An animated object parade by a visual legation (a Skeletal Mesh).
- FPS - Frames per second. How many times the screen can be refreshed per second.
- Actor - Anything you can bring into the world is an actor. There can be many types of actors. Actors can be characters or inanimate objects. Actors are the base class for most of the game world. (Actor)
- Pawns and Characters – A base class where everything is controllable by either the player or the AI is called Pawn class. In pawn, character is a subclass, usually used for humanoid creatures. On the other hand, a pawn could be a vehicle or some sort of creature. Character classes only provide a more specific kind of behaviour. (Pawn; Character)
- Event - An event is a function that is called when something happens. Examples are the values of these changes. Instead of constantly checking if the value has changed, you can bind the event to the value. In code, this means that when a value changes, a function is called that announces the change to everyone bound

to that event. In UE4 this is done by a delegate. The delegate contains all the functions to call after the value has changed.

3.3.2 Blueprints:

Blueprints are an alternative coding method and a true programming language. This means that Blueprints are UE4's visual scripting method, a quick way to start prototyping your game instead of writing code line by line. But here, you can simply drop and drop nodes visually, set properties in the UI, and drag wires to connect them. This method is very important because it is complete, highly adaptable, and allows the designer to use almost all the concepts and tools normally available only to programmers.

In the following figure 3.3.2 shows, the blueprint from the beginning we can see that there were a bunch of nodes:

- On Component Begin Overlap (Box)
- Branch
- Get Player Pawn
- Equal Number
- Cast To Character
- Launch Character

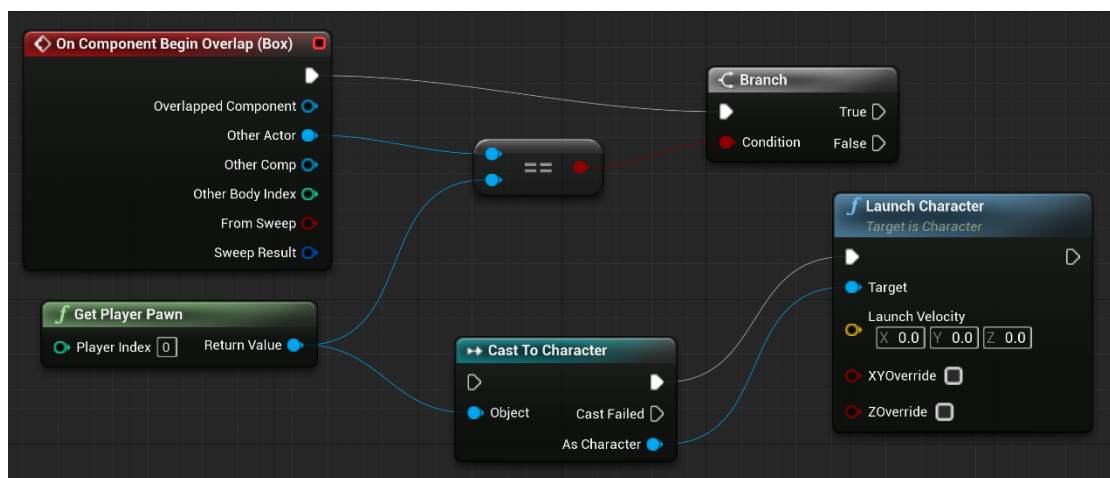


Figure 3.3.2: Basic Blueprints Example

Blueprints and programming: With the Unreal Engine, you have the choice of using Blueprints or programming in C++. Blueprints allow you to start game development without learning to code. If someone wants to create something in his Unreal Engine, in most cases Blueprints can be used to do it. Like player controllers, AI enemies, NPC cars, NPCs, etc. The Blueprint system is very powerful and has many nodes to work with. That's because in programming, you have to write all your code line by line. On the blueprint, on the other hand, you can simply enter a function node and connect that node to another node via a wire. But coding is great for creating complex interactions and behaviours, or heavy calculations, and Blueprints are great for editing and configuring current behaviours.

Blueprint works by connecting nodes, events, functions, and variables with each other by wires are the basic form. Basically, blueprints work by using graphs of nodes for many purposes such as – object erection, individual functions, and general gameplay events which are specified to one another illustration of it and in order to ingredient behaviour as well as other functionalities.

There are several kinds of Blueprints. Such as -

- Data - Only Blueprints
- Level Blueprint
- Blueprint Interface
- Blueprint Macro Library
- Blueprint Class

However, only two of these types are the most usually used Blueprint types, these are

- Level Blueprint
- Blueprint Class

3.3.3 Metahuman:

MetaHuman is a complete framework that allows creators to use highly realistic human characters in any way they can imagine. This includes his MetaHuman Creator, a free, cloud-based app that lets you create fully-manipulated, photorealistic digital humans in minutes.

The following figure 3.3.3.1 shows the main character of our game. So, actually our character looks like that. We made this character by the help of Metahuman, Mixamo, Blender. For now, this the only character of our game but in future we will make more realistic character like this one.



Figure 3.3.3.1: Main Character of 'Alone Assailant'

We created our main character face with MetaHuman.

The following figure 3.3.3.2 shows, how MetaHuman create a face from Mesh to original face.



Figure 3.3.3.2: Mesh to MetaHuman to Unreal Engine

- **Simple and Fun:** MetaHuman Creator is so easy to use that you may not even need to open the manual. Choose a starting point from the diverse offerings in our database, select a few more to contribute to her MetaHuman, and shuffle between them. Then use the modelling tools and controls to adjust your character by simply dragging objects.
- **Highly Variable:** Nearly infinite variations in facial features and skin tones, as well as like hair, eyes, makeup, and teeth, allow you to create a wide variety of faces for your projects. Choose the body type you want for your character and dress it up with different outfit sets with tools. You'll have a perfect MetaHuman in no time.
- **Physically Plausible:** MetaHuman Creator takes data from real-world scans and customizations are constrained to fit within the limits of the examples in the database, making it easy to create physically plausible MetaHumans. Other factors, such as careful selection of skin tone and hair colour, also affect accuracy.

3.3.4 Blender:

Blender is an open source 2D and 3D computer graphics software that enables 3D visualization such as still images, 3D animation, VFX shooting, video editing, rendering, simulation, rigging, motion tracking and many other useful features. Offers. In contrast, when it comes to Blender's 3D development suite, memory and disk specs are also relatively limited. The UI uses OpenGL to provide a balanced experience across all consistent hardware and platforms.

The following figure 3.3.4 shows an example of blender that how a blender works looks like. In blender, we make our custom vehicles part by part and then we joint the parts one by one. After making the vehicle, we had to rig the vehicle physics and after successful we had to make glass materials. And finally comes the colour section, we made the colour and paint it on the vehicle by blender.

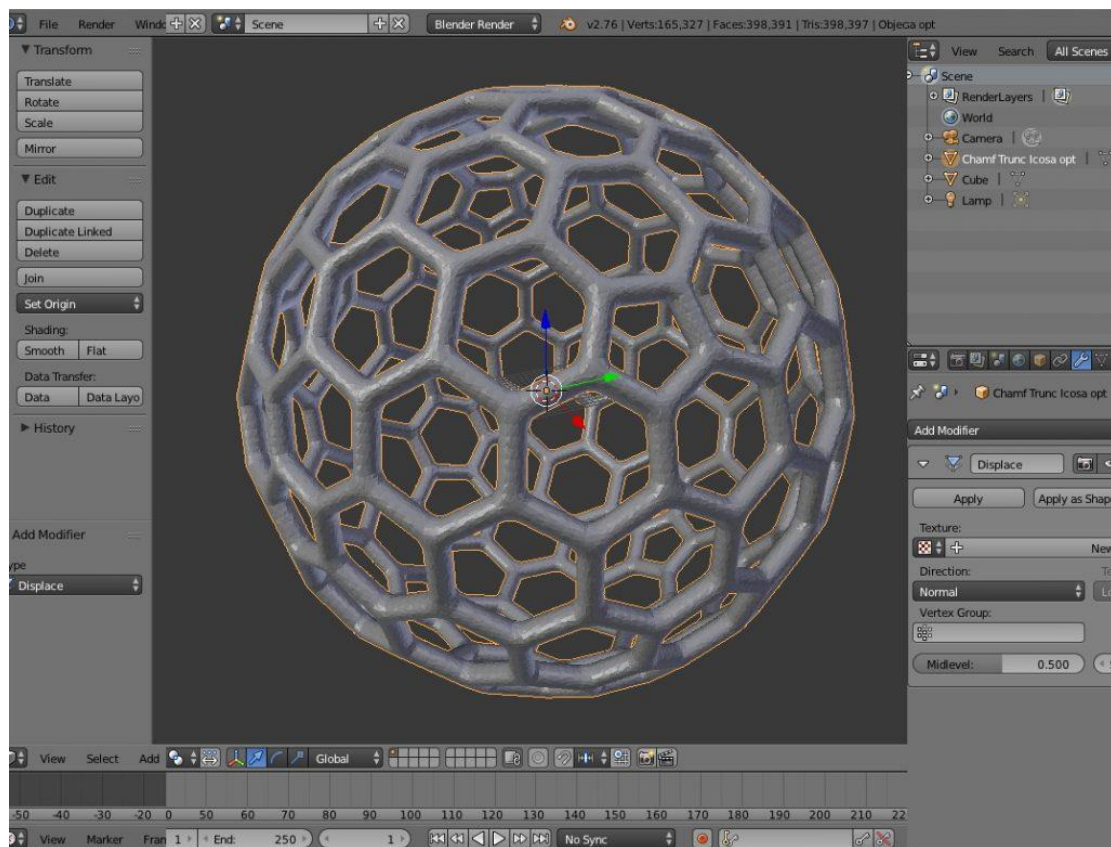


Figure 3.3.4: Thickening 3D models with Blender

There are few Blender's features. Some of them are:

- 1. Modelling:** Blender's wide range of modelling tools allow you to create, transform, shape and edit models.
- 2. Modifier:** Blender's modifiers allow non-toxic transformations of objects. This means you can adjust settings and later undo changes without affecting the original mesh. Blender technically has different kinds of modifiers that work on different kinds of objects.
- 3. UV unpacking:** UV unwrapping refers to the process of opening a mesh to create a 2D texture over a 3D object. Blender has two ways of calculating unwrap. These can be selected from the toolset on the 3D viewport toolbar.
- 4. Animation and Rigging:** It is an act of adding motions to a character or in a game development project with the help of factors, nonlinear animation can be easy, automatic walk cycles can be set, character animation editor is also available where we can instantly see the animation on our rendered screen with fast rigging options. There are some functionalities can be cast easily such as mirror functionality easy painting, skeleton and automatic skinning options, bone and spin making, and gaits. On the other hand, Rigging is the method of creating guides for gait in 3D objects. It has its essentials in hierarchical relationships between different movements in a human skeleton. Blender offers a great set of rigging tools.
- 5. Rendering:** At render time, the computer calculates the lighting in the scene to create the final image or animation. To calculate lighting, the rendering engine needs information from the scene.

3.3.5 Mixamo:

Mixamo is a 3D computer graphics technology company and an online database of characters and animations that anyone can access to be used in movies and games. Basically, this company develops and sells web-based services for 3D character

animation. The machine learning function is used for the character animation process and 3D modelling to rigging and 3D animation.

In the following figure 3.3.5 shows Mixamo's animation library. From here, we pick up some animations like walking, dancing, phone, driving, fighting and more. Because, the game companies made their animations by using green screen and more technology but we don't have these. For this reason, we picked up these animations for our game and these animations are working perfectly.

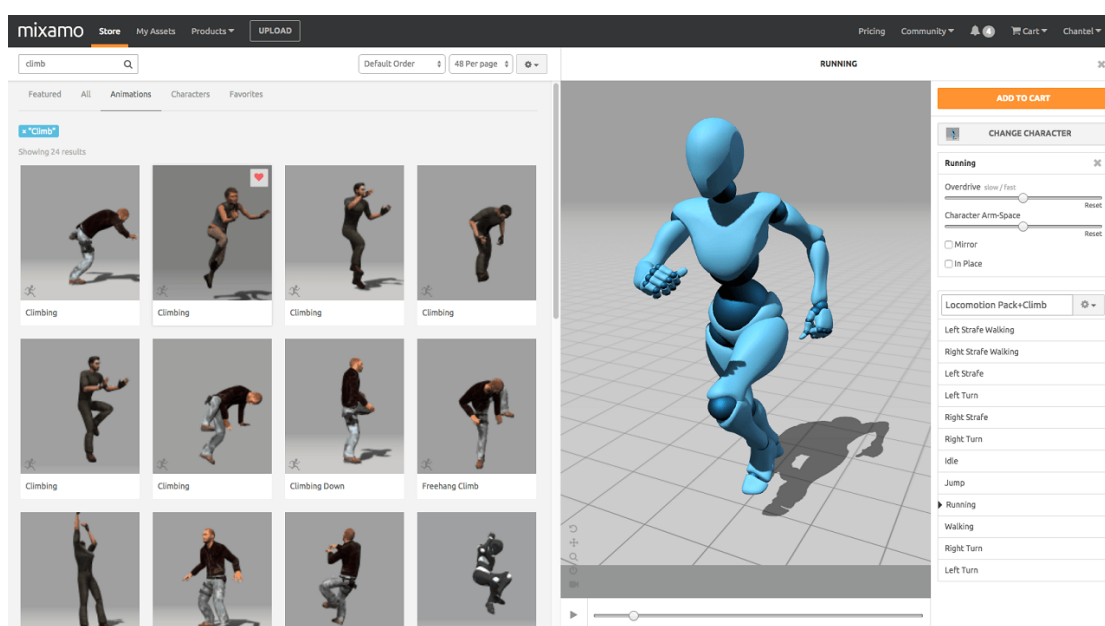


Figure 3.3.5: Mixamo 2.0 Animation

In Mixamo there are 3 main functions that serve when it comes to game development. So, it dispenses:

- Character models
- Animations
- Auto rigging tool

Character models: In Mixamo, there has a moderate selection of optimized game models which are already rigged and ready to be animated.

Animation: On the other hand, Mixamo has a bunch of animations that are free to use. The fact is this along with the Auto Rigger is where the real value is added when it comes to Mixamo.

Auto rigging tool: This is the most useful tool because where Mixamo has a huge amount of model libraries with rigged and provided that anyone makes their own model which is unrigged they can upload their model and put it through Auto Rigging Tool. It will be rigged automatically.

CHAPTER 4

DESIGN & SPECIFICATION

4.1 Agile methodology for Game:

Agile in game development means dividing the game creation process into short iterations. So instead of working on the whole project from start to release, development teams usually work on small projects called features.

In the following figure 4.1 shows agile methodology in game development.

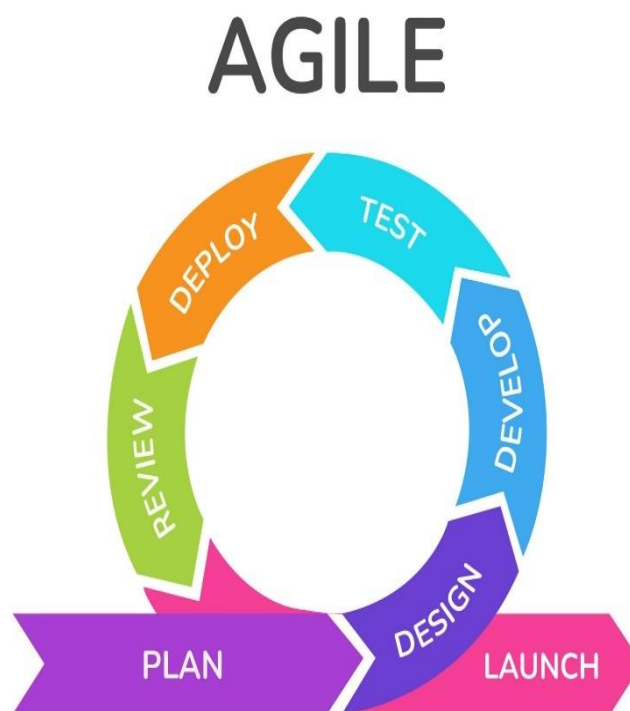


Figure 4.1: Agile methodology in game development

Divide and conquer:

A small set of tasks that need to be completed in a short amount of time per iteration allows developers to spend their time more efficiently and gives project managers control over every small step in the process. At the same time, the iteration includes all his four phases of project planning. Discover, design, develop, test. This also helps avoid bugs that may be missed when testing only the final version of the game. [1]

Scrum first, then ram:

Agile methodology in game development is primarily based on Scrum. However, Scrum is most effective after development. To show that everything in the game works fine. Scrum usually means working on bugs and improvements submitted after testing, player feedback, and the product backlog.

Scrum means working on bugs and improvements submitted after testing, Feedback and product backlog.[1] Because Scrum is most effective in post-development. this It provides faster build and release delivery, lower budgets, and team development. This model consists of five steps, also known as Scrum phases.

Scrum phase of Agile methodology of our project:

1. Planning
2. Designing
3. Developing
4. Testing
5. Releasing

4.2 Frontend:

Welcome page:

At first, the game will load with our custom loading screen. This is the loading screen of our game.

The following figure 4.2.1 shows our games loading screen looks like. We keep this as simple as we can. In this loading screen there has a background music and also some animations which the players will like.



Figure 4.2.1: Loading Screen of game

Then we have to select “Start” option to play the game.

The following figure 4.2.2 shows the starting menu of our game. When a player enters the game, he will get this menu and also when he clicks on esc on keyboard, he will also get this pop-up menu but that time he will get resume in the position of start. In option, he can change the settings of the game visualization as he prefers.

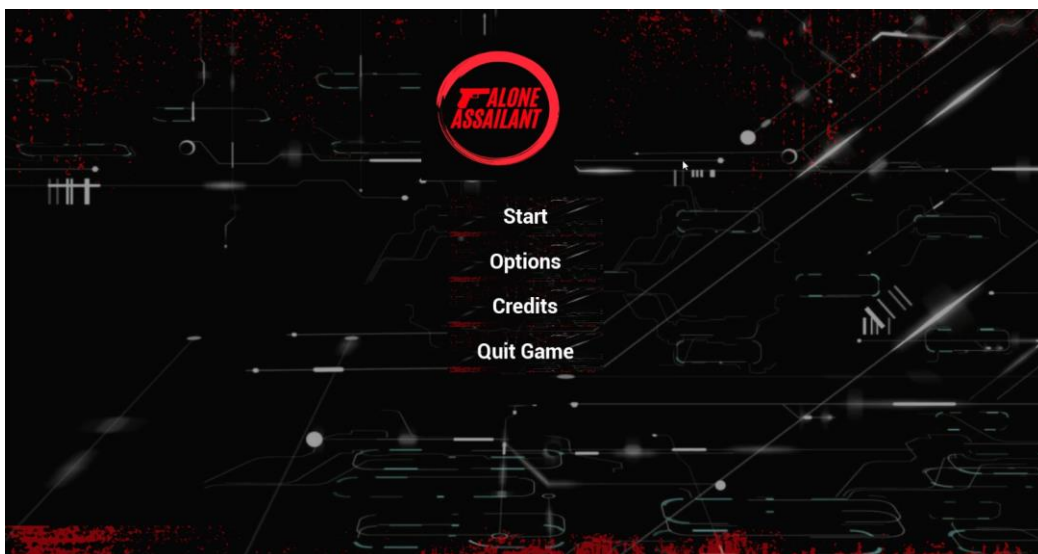


Figure 4.2.2: Start menu of the game

Main Game:

The following figure 4.2.3 shows that the last spawn location of the character. Such as a player quit the game wherever he was in the map. After when he again enters the game, he will resume and spawn the place wherever he left.



Figure 4.2.3: Player Spawned in Last saved location

The following figure 4.2.4 shows A rendered text has been popped up when a player will go near the vehicle and then the pop-up text will visible to him and it has its own rage which had been fixed by us.

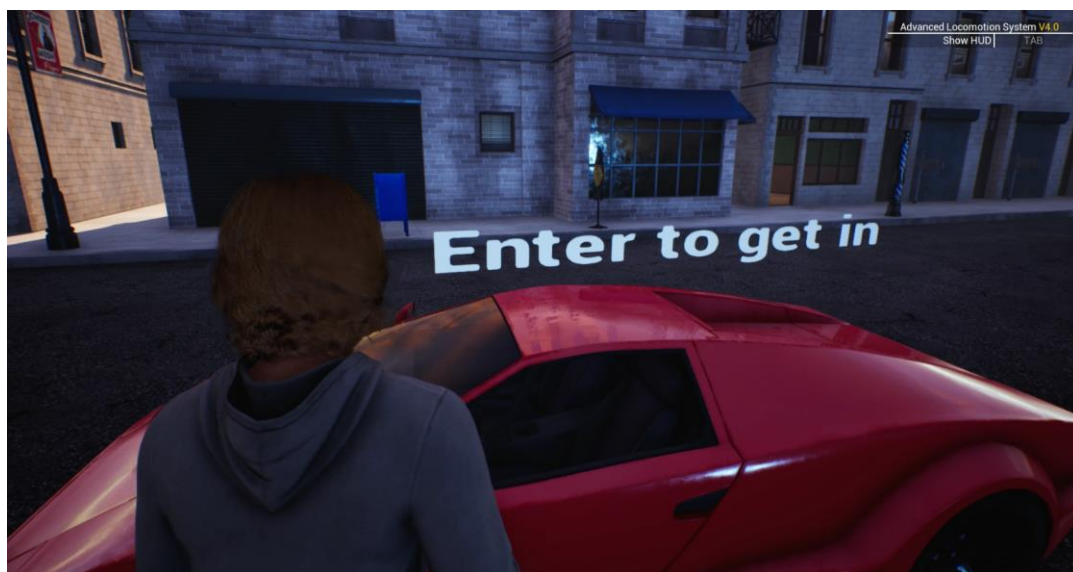


Figure 4.2.4: Player when get near to his/her vehicle

The following figure 4.2.5 shows we can also roam around the city. As we mentioned that our game is an open world game then a player can move around the map with vehicle as far as he want to explore the map. Just simple as the other open world game such as its key binding it like W, A, S, D and Space Bar. And for get enter and exit from the vehicle simply just click E.



Figure 4.2.5: Player while driving his/her vehicle

4.3 Backend:

The following figure 4.3.1 shows a part of the city that we generated in backend.



Figure 4.3.1: A part of the city in backend

The following figure 4.3.2 show the full world/city of our game that we generated in backend.



Figure 4.3.2: Whole city of our game in backend

The following figure 4.3.3 shows the character in backend of our game. We make this character as realistic as we can. We make every single thing much in detail. The hairs, eyes, nose, breads and so on. Also, in figure it shows the backend where we made and test our character.



Figure 4.3.3: Main Character in Backend

The following figure 4.3.4 shows about NPC character and NPC cars that's mean non-player characters and cars. These existed only for making the environment more realistic. In detail, when a player will enter in the map, he will never feel that the map is not a real-life map. We set the Non player character and vehicle in our own located target points.



Figure 4.3.4: Non-player character people and vehicle

The following figure 4.3.5 shows that how we made the light set up in the vehicle headlight and taillight. Firstly, we have to made some spot lights and point lights (as a variable) for the front and back side of the vehicle. Then, we drop and connect the nodes like the given figure blueprint. After that, we test the intensity of the light illumination.

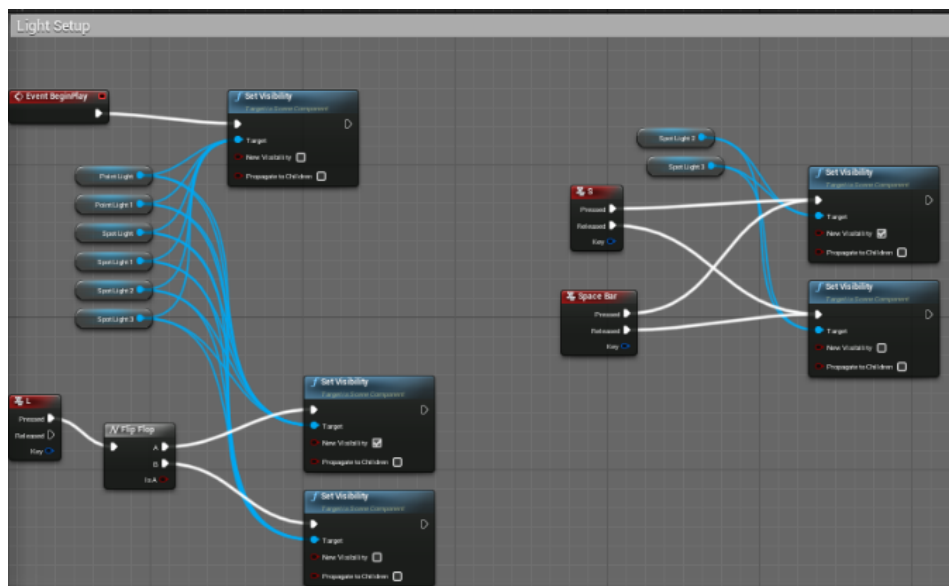


Figure 4.3.5: Vehicle Light Setup Blueprints

The following figure 4.3.6 shows a vital role for the game which is the vehicle movement. At first, we had to make our vehicle model. After made it, we had to make the input value for throttle, steering, braking and handbraking. Then drop the event throttle, steering, handbrake and connect with the vehicle movement to target.

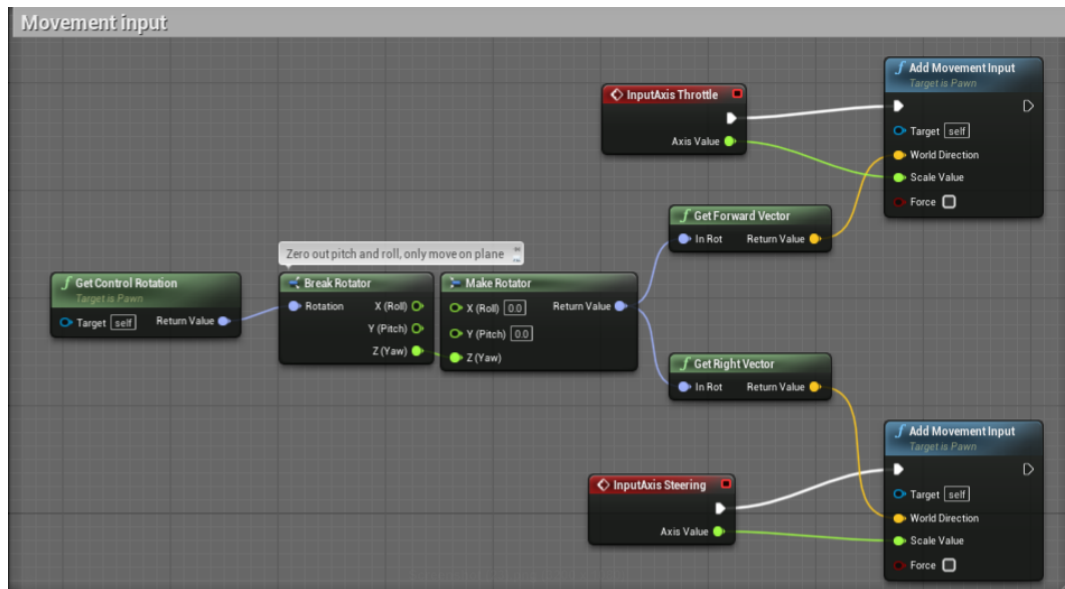


Figure 4.3.6: Vehicle Movement Blueprints

The following figure 4.3.7 shows a vehicle selection menu will pop up when a player enters the game. In this selection menu, we inserted the custom vehicle's which we made in the blender.



Figure 4.3.7: Vehicle Selection Menu

4.4 Special Feature:

There is a special feature that we have already add our project which is solve basic math in gameplay.

In the following figure 4.4.1 shows some math solving station where player can go and solve some basic math.



Figure 4.4.1: Basic Math solving station

In the following figure 4.4.2 shows the question of the math. In this phase, the game will pause and you can continue the game after solving the math.



Figure 4.4.2: Question of basic math

In the following figure 4.4.3 shows, if player select wrong answer it will show ‘Wrong’ in the upper left section. Player cannot continue the game without solving the math.

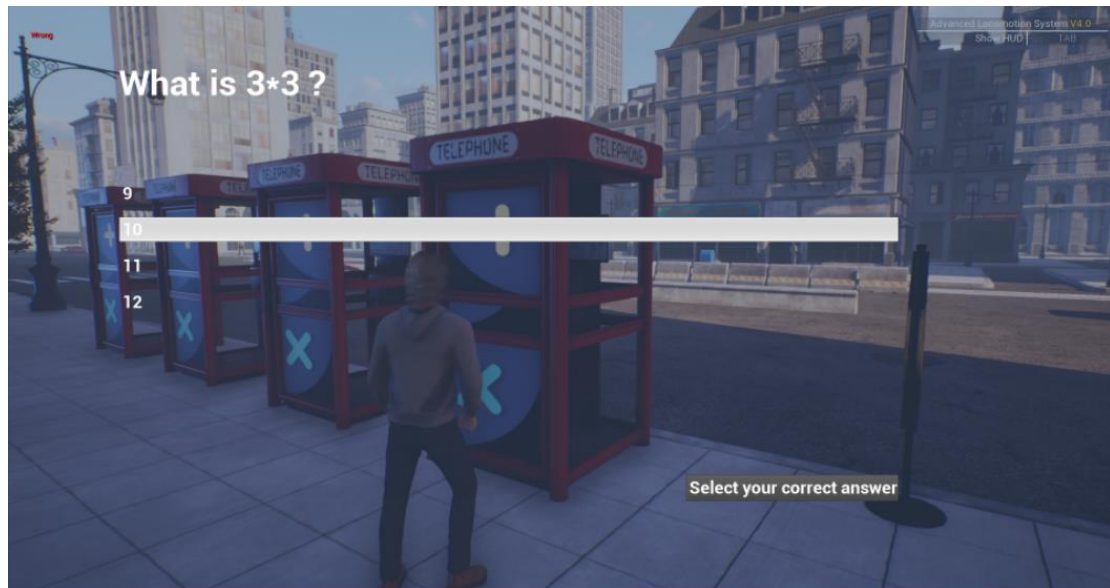


Figure 4.4.3: When player choose the wrong answer

In the following figure 4.4.4 shows, if player select correct answer it will show ‘Correct Answer’ in the upper left section. Player can continue the game after solving the math.



Figure 4.4.4: When player choose the correct answer

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Project Implementation:

At first, we take ideas from popular open-world games like the GTA series, Watch Dogs series, Gangster Vegas, and so on. Maximum players who play video games are well-known for these games. So, these games are all same category types just the main differences are the graphics and some aggressive missions such as fighting with each other, and all kind of robberies there is no educational stuff that existed because they made their games just for entertainment purposes.

So, we thought that to make our game entertaining and educational also. We take the minimum entertainment idea from these games but the educational system is fully implemented by us. Then in the second step of implementation, we choose the platform where we will make our game and answer is Unreal Engine and the programming language is Blueprint. We make our game character meta-human and animation of body language in Mixamo. The cars of this game are made in the blender.

5.2 Project Testing:

Testing is connected with the implementation that's mean 1st of all we make the character or vehicle by meta-human and blender or auto desk Maya then we create an external platform in Unreal Engine and implement the made things in there so that we can see that those are working fully ok.

Sometimes it's worked fully ok and sometimes it occurs some error. Then we find out the errors and fix them one by one by changing the node connection with another node. And also, we tested these things in another OS system and other pc users and when we see that all are ok, after that when the testing is fully functional and ready for final implementation, we set up these in our final workplace.

5.2.1 Experiment of game appliance: Below the given table, we have added the experiments outcome-

The following table 5.2.1 shows, we discussed about the test input, the testing date and the output of our game when we had tested.

Table 5.2.1: Experiment of game appliance

Experiment	Examine
Experiment input	User data
Consequence	Game appliance
Experimented on	04 December, 2022
Outcome	Passed

5.2.2 Bug fixing: Basically, when we made our game and do functionality experiment.

From this experiment we find lots of bugs. Such as –

- Character movement problem
- Character and vehicle animation or physics problem
- AI character movement problem
- AI car route problem
- Car lighting bug

We fixed these bugs during the experiment phase. By researching on these bugs finally we were succeeded in the bug fixing.

5.2.3 Result: Our game is very much interesting for the players. We faced some bug issues in the game that we have already fix up.

- Smooth character movement
- Smoot vehicle driving
- User-friendly map
- Entrainment missions
- Educational missions.

CHAPTER 6

IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

6.1 Impact on Society

In this fast-paced modern era, most people are liked to play video games. But the fact is most of the games were made for just giving entertainment such as the GTA series. This game is a 3D open-world game just like ours, but in the GTA series, there are no educational purposes existed. Everywhere the players were just fighting with each other. Yes, they had a story but also in the story, there had no educational thing. On the other hand, in our game, we made an educational item which is solving basic math by going to the booth location and also, we entered this solving via our story. By doing this, the player can get entertainment as well as learn basic math via solving the given problem and he can learn something from our game. In the end, we can say that most of the games were made for giving entertainment and also wasting time but not ours that's it.

6.2 Impact on Environment

The impact of our video games on the environment can be both positive and negative.

Positive impacts include:

- Promoting environmental awareness: Games that have an environmental theme can raise players' awareness of environmental issues and inspire them to take action.
- Encouraging sustainable behaviour: Games that reward players for making eco-friendly choices can encourage them to adopt sustainable habits in real life.

Among the negative effects are:

- Resource consumption: Video game and gaming equipment manufacture, distribution, and use all consume a substantial amount of energy and other resources, which worsens the environment.
- E-waste: Electronic waste is a problem since gaming technology is changing so quickly and because old equipment is discarded.

Every industry is under investigation for its influence on the environment and climate as the implications of climate change increase the pressure to decrease energy usage and greenhouse gas emissions globally. The energy needed to run sophisticated computers, fast internet connections, and cloud-based technologies is particularly expensive for the video game, media, and entertainment enterprises.

According to one research, 5 million automobiles' worth of energy were consumed globally by gamers. Computers and gaming consoles are among the 50 million tons of electronic garbage (E-Waste) that are created annually, just 20% of which are recycled, according to the UN Environmental Programme.

In conclusion, while video games can benefit the environment by fostering sustainable behaviour and increasing awareness, they can equally harm it by wasting resources and creating e-waste.

6.3 Ethical Aspects

In every open-world game, there should have a minimum size of violation acts. Through these acts, many players tried it in real life but would understand that a game is a game. In our game, we gave the narrowest size of violation acts than other open-world games. No matter how many educational games people made a game can never give the proper education to people, most children were wasted their time playing video games but they had to study first and then do the rest of things. They must study and try hard and soul. Playing a game is just a kind of taking entertainment in leisure time and our thought is

that the educational system must play a vital role in a video game that means when a player will play a game in their leisure time, their valuable time shouldn't be wasted.

6.4 Sustainability Plan

Our main plan is to make a game where a player not only can take entertainment but also can learn something. In the future, we will add more entertainment systems with the educational system, more visual items, custom cars, custom characters, customizable dress-up and so on.

CHAPTER 7

DISCUSSION & FUTURE SCOPE

7.1 Discussion:

In this modern era, most people elder or younger are preferred to play video games a lot. Because it's another way of spending their leisure time as well as provided that there was a video game which is an open world 3D where they can do everything even these things which they can't do in real life.

We preferred to make our game in Unreal Engine 4 and our programming language is the blueprint. Blueprint is an alternative to coding. Where we have to write a code line by line, on the other hand in the blueprint just we have to drop nodes and connect another node by wiring.

Our game is a 3D open world game where a player can do whatever he wants such as traveling the whole map for enjoyment, doing some interesting activity like solving puzzles, or something else, and provided that he wants to learn something about educational purposes he can do missions where he will get a clue of basic math's and solve the math. In detail, if a player enters the game and he wants to learn some basic math or coding with entertainment then he will go to the location where he can start the mission. After that, he will press the key for start then a pop-up window opens up, and step by step he can solve the mission.

So, we can say that most of the games are made for giving entrainment to the players. But we are making a game where a player can entertain himself and also learn some educational things.

7.2 Future work:

As a beginner, we are making not only a game but also an open-world game. Previously we said that our game is an educational game besides entertaining the players. So, this game has a future also. In the future, we will add more functions. For Example:

- At this moment our game is only a single-player game but in the future, we will make it multiplayer mode.
- Adding more missions both in single-player mode and multiplayer mode
- We will make and add custom real-life cars and a player can customize these cars as they want
- At this moment, our game is only for pc users, but we will also make our game compatible with phone users
- We will make and add a custom weapon system for the missions.

7.3 Conclusion:

Therefore, we can say that our motivation is not like most of the games these days waste, children's valuable time just for giving entertainment but not a goal is giving entertainment with educational experience. "Because these days, education has also been changed by video games through the use of new technologies that help teachers and students communicate in new methods through educational games like ours. Our game will increase good social skills, perform academically and build up better relationships with others. Players will get benefits like psychological and social. Also, our video game will help to tune up the brain's reasoning and problem-solving as well as control attention and visuospatial savvy skills. In our game, maximum game items made by us for example the vehicles, characters, weapons and some items are picked from some third party such as animation.

REFERENCES

- [1] Agile In Game Development, Available at <<https://meliorgames.com/game-development/agile-in-game-development>>, last accessed on 5 December 2021.
- [2] Unreal Engine 5, Available at <<https://www.unrealengine.com/en-US/unreal-engine-5>>, last accessed on 5 December 2022.
- [3] Bridge by Quixel - Unreal Engine, Available at <<https://www.unrealengine.com/en-US/bridge>>, last accessed on 5 December 2022.
- [4] 10 Massive Problems With Open-World Games, Available at <<https://gamerant.com/big-problems-open-world-video-games/>>, last accessed on 5 December 2022.
- [5] Benefits of Video Games For Kids & Adults, Available at <<https://www.geico.com/living/home/technology/9-reasons-to-give-video-games-a-try/>>, last accessed on 5 December 2022.
- [6] Exploring the relationship between gaming and civics, Available <<https://www.pewresearch.org/internet/2008/09/16/introduction-7/>>, last accessed on 5 December 2022.
- [7] Blueprints Visual Scripting, Available at <<https://docs.unrealengine.com/5.0/en-US/blueprints-visual-scripting-in-unreal-engine/>>, last accessed on 5 December 2022.
- [8] The Pros & Cons of Creating 3D Content with Blender Software, Available at <<https://www.epidemicsound.com/blog/blender-software/>>, last accessed on 5 December 2022.
- [9] An Introduction to Unreal Engine 4 (Focal Press Game Design Workshops), Available at <<https://www.pdfdrive.com/an-introduction-to-unreal-engine-4-focal-press-game-design-workshops-d158218397.html>>, last accessed on 5 December 2022.
- [10] Mixamo, Available at <www.mixamo.com>, last accessed on 5 December 2022.
- [11] blender.org - Home of the Blender project, Available at <<https://www.blender.org/>>, last accessed on 5 December 2022.
- [12] Planning Your First Video Game, Available at <<https://www.pluralsight.com/blog/creative-professional/practical-guide-to-planning-your-first-game>>, last accessed on 5 December 2022.
- [13] The Agile Development Process for Mobile Apps, Available at <<https://www.krasamo.com/agile-development-process/>>, last accessed on 5 December 2022.
- [14] An introduction to Unreal engine 4, Available at <<https://www.pdfdrive.com/an-introduction-to-unreal-engine-4-e182208125.html>>, last accessed on 5 December 2022
- [15] Jani Hämäläinen, Game Development with Unreal Engine 4, Theseus, 2020, pp. 6-10
- [16] Open world, Available at <<https://en.wikipedia.org/wiki?curid=18493095>>, last accessed on 5 December 2022
- [17] What is Mixamo and How Can it be Used in Games, Available at <<https://cgobsession.com/what-is-mixamo-and-how-can-it-be-used-in-games/>> last accessed on 5 December 2022
- [18] Documentation - Unreal Engine, Available at <<https://www.unrealengine.com/en-US/previous-versions/documentation?sessionInvalidated=true>> last accessed on 5 December 2022

[19] Epic Developer Resources Documentation, Available at

<<https://dev.epicgames.com/docs?sessionInvalidated=true>> accessed on 5 December 2022

[20] Unreal Engine | Oculus Developers. Available at

<<https://developer.oculus.com/documentation/unreal/unreal-engine/>> last accessed on 5 December 2022

Alone Assailant

ORIGINALITY REPORT

20%	18%	1%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	8%
2	Submitted to Daffodil International University Student Paper	3%
3	dbpedia.org Internet Source	2%
4	meliorgames.com Internet Source	1%
5	cgobsession.com Internet Source	1%
6	www.educba.com Internet Source	1%
7	gamerant.com Internet Source	1%
8	Fabrizio Banfi, Alessandro Mandelli. "Computer Vision Meets Image Processing and UAS PhotoGrammetric Data Integration: From HBIM to the eXtended Reality Project of	<1%