

RUSH RUN: AN PC RUSH RUN GAME

BY

Mahmudul Hassan

181-15-10709

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

Supervised By

Nazmun Nessa Moon

Associate Professor

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

24 JANUARY 2023

APPROVAL

This Project/internship titled “**Rush Run: An Pc Rush Run Game**”, submitted by Mahmudul Hassan, ID No: 181-15-10709 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment 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 *24 january.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

Abdus Sattar

Assistant Professor

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



Internal Examiner

Fatema Tuj Johra

Senior Lecturer

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

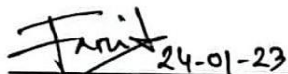


Internal Examiner

Dr. Dewan Md Farid

Professor

Department of Computer Science and Engineering
United International University



External Examiner

DECLARATION

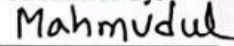
I hereby declare that, this project has been done by us under the supervision of **Ms. Nazmun Nessa Moon, Associate professor, Department of CSE Daffodil International University**. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



Ms. Nazmun Nessa Moon
Associate professor
Department of CSE
Daffodil International University

Submitted by:



Mahmudul Hassan
ID: 181-15-10709
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

In the beginning, I would like to express our heartiest gratitude to the Almighty for enabling us to prepare the final year project successfully.

Now I want to mention our Supervisor **Ms. Nazmun Nessa Moon, Associate professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “Windows” to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, errors and correcting them at all stages have made it possible to complete this project.

Afterwards, I would like to express our heartiest gratitude to **Dr. Touhid Bhuiyan, Professor & 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.

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

Abstract

Many of us are loved playing pc game, fall guys game lovers are also there. Most of the pc games are represented urban areas graphics and environment. But I tried to make a change and came out from this concept. Think differently that, how was that if it could have been related to learning something? From this concept I built a pc game in different way. Changing in outside graphics of map and adding some ice places . In this level player face difference obstacle in difference places . Player should move forward by using own technical. Therefore, I think it becomes very interesting for every gamer.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
List of Figures	viii
List of Tables	ix
CHAPTER	
CHAPTER 1: INTRODUCTION	1-2
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	1
1.4 Expected Outcome	2
1.5 Report Layout	2
CHAPTER 2: BACKGROUND	3-8
2.1 Introduction	3
2.2 Related Works	3
2.3 Comparative Studies	6
2.4 Scope of the Problem	6
2.5 Challenges	7
CHAPTER 3: REQUIREMENT SPECIFICATION	9-13
3.1 Business Process Modeling	9
3.2 Requirement Collection and Analysis	10
3.3 Use Case Modeling and Description	10

3.4 Logical Data Model	12
3.5 Design Requirements	13
CHAPTER 4: DESIGN SPECIFICATION	14-29
4.1 Front-end Design	14
4.1.1 Rush Run Racing Track in Blender	14
4.1.2 Making Game Environment	16
4.1.3 Setup the Character	17
4.2 Back-end Design	18
4.2.1 Character movement	18
4.2.2 Camera Movement	19
4.2.3 Apply Jump	20
4.2.4 Limit Walk Speed	20
4.2.5 The Starting Zone	21
4.2.6 The Circle	22
4.2.7 The Tasers	22
4.2.8 The Dumb rolls	23
4.2.9 The balls and Pendulum	24
4.2.10 Road Preview Camera	24
4.2.11 The Character Shock Effect	26
4.2.12 The Turbines	26
	28
4.3 Interaction Design and UX	28
4.4 Implementation Requirements	29

CHAPTER 6: IMPLEMENTATION AND TESTING	30-32
5.1 Implementation of Database	30
5.2 Implementation of Front-end Design	30
5.3 Implementation of Interactions	31
5.4 Testing Implementation	31
5.5 Testing Results and Reports	32
CHAPTER 6: CONCLUSION AND FUTURE SCOPE	33-33
6.1 Discussion and Conclusion	33
6.2 Scope for Future Development	33
REFERENCE	34
APPENDIX	
Appendix A: Project Reflection	35
PLAGIARISM REPORT	36

LIST OF FIGURES

FIGURES	PAGE
Figure 2.1: Fall Guys	4
Figure 2.2: Stumble Guys	5
Figure 2.3.: Human : Fall Hunts	5
Figure 2.4: Character movement challenge	8
Figure 3.1: BPM of Tour Map	9
Figure 3.2: Use case diagram	11
Figure 3.3: ER diagram	12
Figure 4.1: Draft design of rush run racing track	14
Figure 4.2: Rush run racing track on Blender	15
Figure 4.3: Final output of run rush racing track	16
Figure 4.4: Initial game environment	16
Figure 4.5: Importing Character on Unreal Engine	17
Figure 4.6: Character Movement	18
Figure 4.7: Camera Movement	19
Figure 4.8: Back-end code for applying Jump	20
Figure 4.9: Limit max walk Speed	21
Figure 4.10: Backend Code for Character Movement	21
Figure 4.11: The Split Zone	22
Figure 4.12: Working for the Circle	23
Figure 4.13: Activate the Tasers	23
Figure 4.14: The Dumb rolls	24
Figure 4.15: The Ball	25
Figure 4.16: The Pendulum	25
Figure 4.17: Road Preview Camera	26
Figure 4.18: Taser Effect	27
Figure 4.19: Show inner Skeletal	27
Figure 4.20: The Turbines	28
Figure 5.1: Main Menu user interface	30

List of Tables

Tables	Page
Table 2.1: Comparison between Race and Learn and competitors	6
Table 5.1: Testing Implementation	31

CHAPTER 1

INTRODUCTION

1.1 Introduction

We are familiar with computers in our daily life. We play many kinds of computer games. This project seeks to build a rush run game where recreation meets learning. Learning is about that particular ice location. Most games have foreign graphics and urban environments. I've included some of his ice cream flavor cards here to give players a different feel.

1.2 Motivation

If you've played a Fall Guys game, you may have noticed that the obstacles, game environment, and other graphics are mostly urban. Although playing games is a waste of time. It will inspire you to think differently. So I decided to do something new. Players can reach the next level only by using techniques.

1.3 Objectives

- Learning through entertainment.
- Briefly outline the ice skating rink as a quick note.
- Add an obstacle section for players so they don't waste time.
- Get people to know about this place

1.4 Expected Outcome

The purpose of gambling is entertainment. But rush run game with learning is more than just entertainment. We hope players will enjoy it and know about this place. Learning through entertainment is more interesting than learning in general.

1.5 Report Layout

Chapter 1: Introduction

In this chapter, we have discussed the project introduction, motivations, goals and expected results. The layout of the report was followed later.

Chapter 2: Background

Describe the background of the project. We will also talk about similar works, comparisons with other project systems, and problems and challenges of the project.

Chapter 3: Requirement Specification

This chapter provides recommendations for professional process structures, requirements gathering and analysis, project models and descriptions, and logical data models and design requirements.

Chapter 4: Design Specification

This chapter describes the project blueprint. Front-end design for the visual part, back-end design for the coding part, interface design and UMG, and minimum requirements for implementation

Chapter 5: Implementation and Testing

This chapter covers the implementation flow of the game, the visual part of the front-end, the interactions, the implementation of the tests, and the test results of the game.

Chapter 6: Conclusion and Future Scope

I talked about the conclusions and the scope of further developments that essentially arise from the project

CHAPTER 02

BACKGROUND

2.1 Introduction

We designed a run rush game with a difference. Before, everyone knew about games whose main vision was only to play. The result of this type of game is entertainment only. In this project, you can also play some educational games for fun. As a 3D run rush game, this game was developed with his Unreal Engine 5. Some obstacles are made in 3D software Blender. The game environment within the map is also created in Unreal Engine. Photoshop is used for resizing textures, creating images, and basic editing of downloaded images. The core languages are C plus plus and Blueprint.

2.2 Related Works

Of course, since it's a last-minute game, there are parts that are common to many games. But it's different in a roadside environment, as we've shown you at some ice spots. And compare that to learning, and there's no game where the two meet. If you just look at the racing part outside the learning section, there are some big differences in graphics for characters, levels and appearances. Related works for game environments are considered below.

1. Fall Guys [1]
2. Stumble Guys [2]

Fall Guys

Figure 2.1 below is for the Fall Guys game environment title Fall Guys. Ultimate Knockout. A parallel to our project is the failure dynamics.



Figure 2.1: Fall Guys

Stumble Guys

This is another game similar to obstacle game environment. Figure 2.2 shows outdoor graphics.



Figure 2.2: Stumble Guys

Human: Fall Flat

This is another game similar to obstacle game environment. Figure 2.3 shows playback outside the graphics during playback.



2.3 Comparative Studies

The rush run game above had certain similarities with the game environment, but also had some significant differences. The rush run game part theme is similar to all rush games, but the environment and appearance graphics are different. Here are the differences between Rush Run and Learn.

Table 2.1: How Race and Learn compares to competitors

No	Rush Run	Competitors
1	Study part available	No study part
2	Outside graphics contains ice	Only countryside graphics
3	Less disk space required	More disk space required
4	No paid version	No Free versions

2.4 Scope of the Problem

We offer this as a free version. Anyone can access this game. I didn't have to sign up or register. Windows has become one of the most used operating systems for personal computers. As such, there are many developers working on developing PC games. After completing a project, most of the time it seems that I didn't meet all the requirements of my needs. Are there any limits in the ever-changing world of technology?

Most of his UMG interfaces in current computer games are not user friendly and have limitations. For this reason, our main goal was a simple user interface. After entering the game, you will be presented with some information about the location. Next, the user has to answer some basic questions. After that, the game continues.

2.5 Challenges

Create games that run on the fly without any prior knowledge of how the process will unfold. Game development is a huge field and beginners are faced with many new problems, so getting started has never been easier. I am confident enough to take this as a challenge that I am determined to take on. Some challenges are like

- Familiarize yourself with Skeletal's new Blender [4] user interface. I'm a beginner so I can make a skeleton 4 times in a row.
 - Set up your game environment with various textures and Unreal Engine tools.
 - Next, you will also need to learn Photoshop basics for basic editing of downloaded textures and creating images in Photoshop [5].
 - Includes main editor Unreal Engine 5 [6]. You're already familiar with Android Studio, but there are some big differences between the two when compared to Unreal Engine. So it's finally time to tackle the Unreal Engine UI.
 - I have no experience using his Unreal Engine intrinsics for C++[7] or other Rush Race games[9]. This really helps reduce coding and is required in many aspects.
 - Working with different coordinates, their values and rotation angles was very confusing.
 - Obstacle movement, camera movement with the character, rotation and blur were challenges.
- According to Figure 2.4

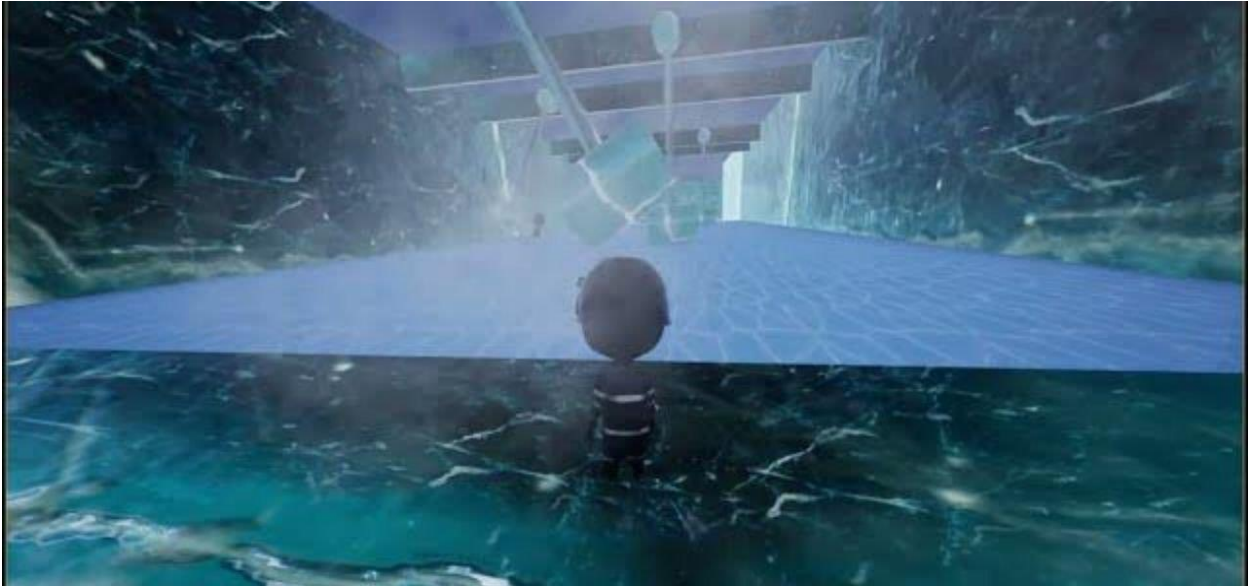


Figure 2.4 : Character movement challenge

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Business Process Modelling

BPM stands for Business Process Modeling and is a blueprint for creating a structural view of a system or process flow. It contains flowchart-like symbols and conditions. Click on new game user to open a new level

The BPM model is shown in Figure 3.1.

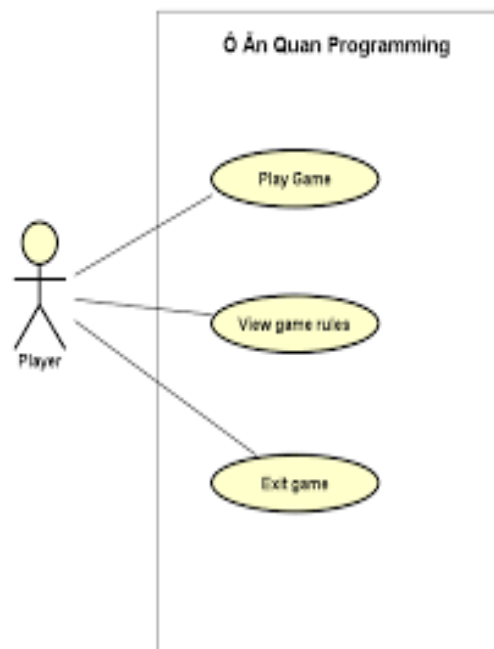


Figure 3.1: BPM of Tutor Map

3.2 Requirements Collection and Analysis

Gather key requirements, including requirements gathering and analysis, before starting a project. A fast-paced racing game that requires a lot of resources to satisfy your gaming cravings. I use programming languages C++ and JavaScript. Editors are Unreal Engine 5, Blender and Photoshop. I play games i.e. Windows OS. There are his two types of requirements: functional and non-functional [10]. Functional requirements are similar to systems. From the system's point of view, when you start the game, you will see a popup with some information.

Non-functional requirements define the efficiency, safety and performance of the system. Our game UI is very user-friendly and beautiful, resulting in a great user experience.

3.3 Use Case Modelling and Description

A use case model graphically describes the interactions between the elements of the system in various phases. A technique used in systems analysis to define, clarify, and organize system requirements. At the same time, find the conditions for achieving your goals. And the conditions for not reaching the goal. This model is shown in Figure 3.2 below.



Figure 3.2: Use case diagram

Description of the use case for the game Rush Run

Use case: Rush run game

Actor: Player, system.

Prerequisites: Read the information displayed.

Purpose: Start the game.

Description:

1. First, the user must enter the game and click on the New Game option.
2. Then some information will be displayed in one place on the ribbon. Players should.
3. Next, you will be asked some questions based on the information presented previously.
4. If the player answers the question correctly, the game begins.
5. If the answer is wrong, the information will reappear

Post condition:

- If the user answers correctly.

3.4 Logical Data Model

The term logical data modeling is used to define and define the requirements necessary to support formal processes in the information systems that serve this domain within an organization. An entity-relationship model or entity-relationship diagram (ERD) is a logical data model containing entities, attributes, and relationships. The map is modifiable and is therefore a multivalued attribute of the histogram.

The logical data model is shown in Figure 3.3 below.

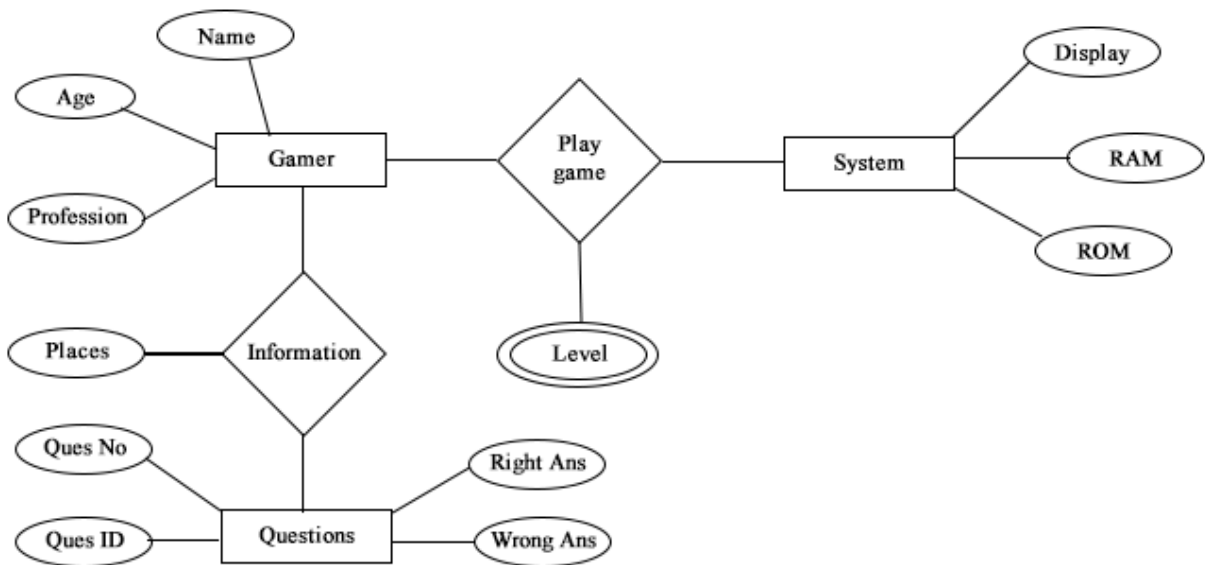


Figure 3.3: ER Diagram

3.5 Design requirements

The following goals are kept in mind when designing the system:

Make the system easy and flexible for users

That's our first goal. Users of the system have good control over their own goals to achieve their goals.

Make my system compatible

It should be integrated with the overall system and should require less maintenance and improvements in the future.

Effective

that's the most important thing. The system must run and operate without delay with the minimum available software resources of the hardware. Try to make it lighter so it's more responsive on all devices.

CHAPTER 4

DESIGN SPECIFICATION

4.1 Front-end Design

The importance of front-end design is the visual part of a system. Through which users can interact with the system. When developing the game, I tried to make it user-friendly. My intentions have been kept as simple as possible. Front view of my project refers to game environment design, obstacle creation, character design and all other visual graphics. This means that I have considered the non-coding part as our front end design. Also, I point out a place where there is an obstacle.

4.1.1 Rush Run Racing Track in Blender

I started my journey towards making a fast racing game from here. Before doing this, I first designed it in Paint, which is how the track would be designed on Blender. Figure 4.1 shows our demo project. Because doing the final exit directly is not easy as I am at a beginner level.

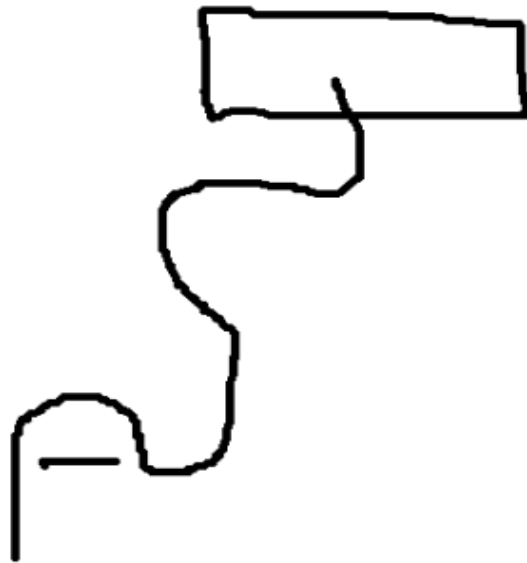


Figure 4.1 Draft design of Rush Run racing track

Then I go to Blender, with the help of bezier curves the edges are drawn. After doing some basic operations on the bezier curve like scaling, rotating, extruding the curve, changing the position on the ground, I was able to draw it as my draft demo. Figure 4.2 shows the running software demonstration.

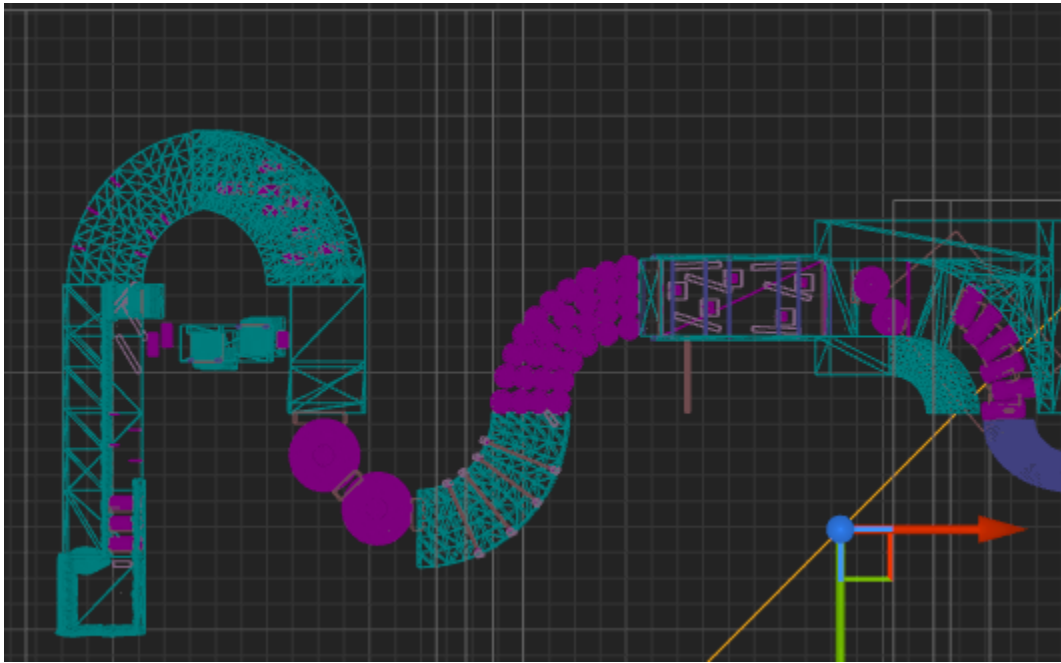


Figure 4.2: Rush Run racing track on Blender

I almost finished my track, after giving it a really fast track view, need to place a downloaded texture[11] on the track. And also the outer part of the road. Many edits were made in Adobe Photoshop before being used on Blender. From editing UV[12] and Blender's face option selected, I imported two textures for the road, one for the road and one for the curb. Again, scale and ground position are done to correct the images on the plotted line. The original extension of the Blender file was .blend, I also saved my file in this format. Then I exported it in .fbx format because I needed to work with it on Unreal Engine 5. In the end I got the output as shown in Figure 4.3

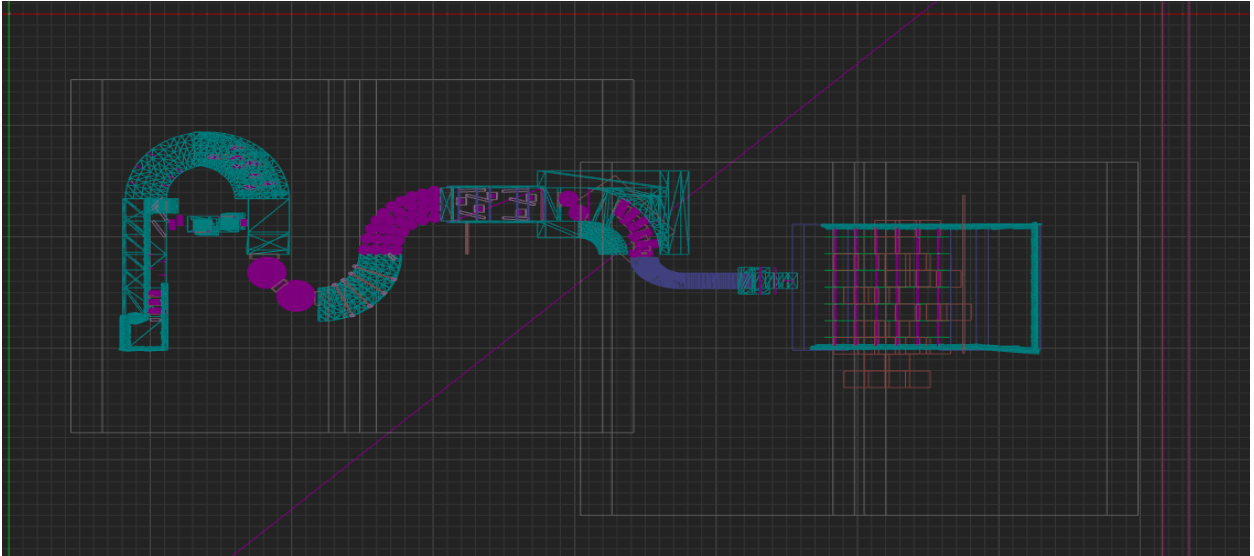


Figure 4.3: Final output of Rush Run racing track

4.1.2 Making Game Environment

Since I've created my environment based on mainline and inner lane, I first import the .fbx file into Unreal Engine 5. Then I fix my position on the road if necessary. Using different brushes on Unreal Engine 5, I designed the entire terrain environment. Figure 4.4 shows the environment with the top image and the bottom image the game view. Then I updated this by adding ice textures and other materials related to the environment.

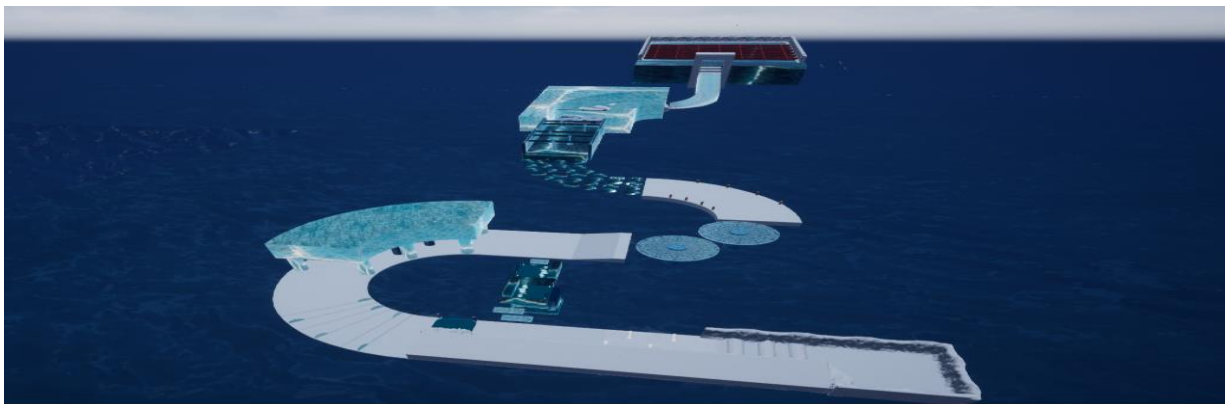


Figure 4.4: Initial Game environment.

4.1.3 Setup the Character

First I think you just need to create a character in Photoshop then import it into Unreal Engine and that would be the way to configure the character. But it's a 3D project and I also need a 3D character. Not only do I need a 3D character, but along with the vehicle, I need the Unreal Engine asset for the character to work on it. So a character was uploaded [13] for this project with bogus motivational content and other material. Then I changed the shape and color of her body to match our project in Photoshop. Then, some character textures such as the character's body, the character's eyes, and the martial arts were assigned. The character is then entered into a bogus engine, and Figure 4.5 shows it. With this activity, our front part is complete.

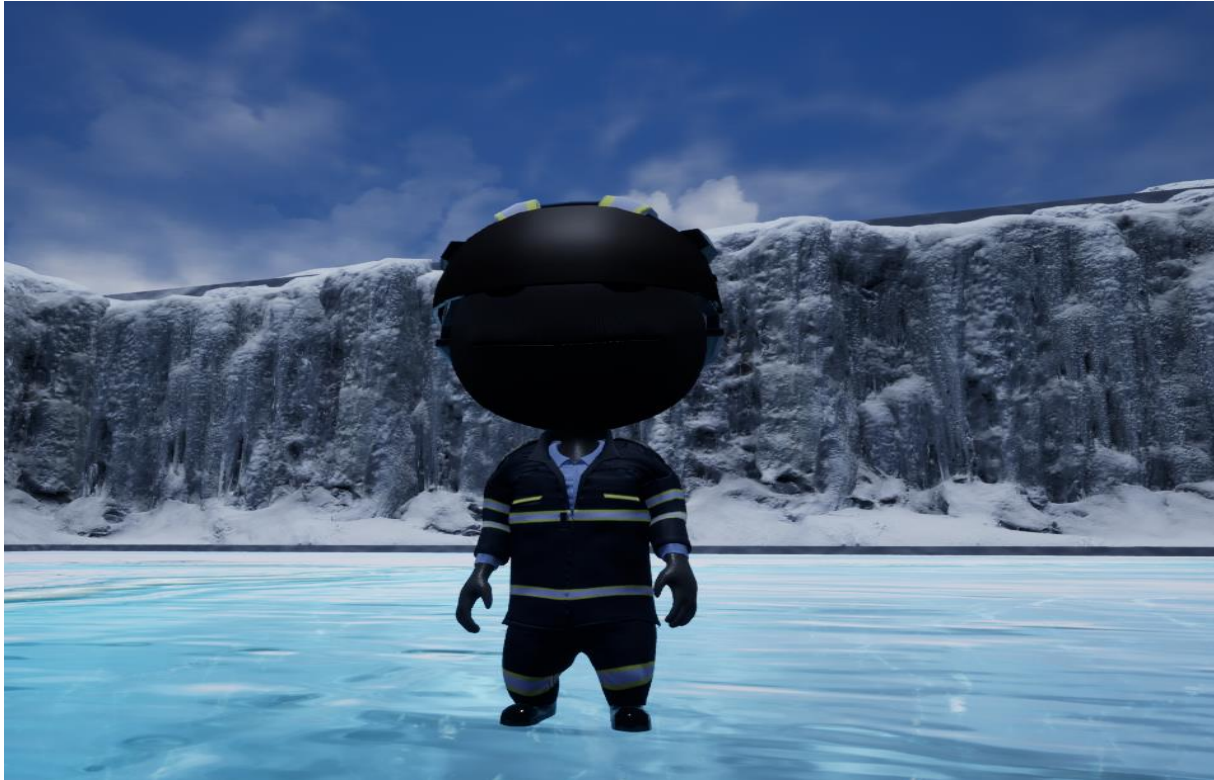


Figure 4.5: Importing Character on Unreal Engine

4.2 Back-end Design

In this section, I will talk about coding. Most of the time I use C++ and blueprint. The back-end is the part where all the logic is operated behind the system. The user cannot interact with this section or anything. Back-end design is where the real work happens.

4.2.1 Character movement

First, I create two motion protected variables since they are assigned to us from outside in Unreal Engine, they are MoveForward and MoveRight. Here, movForward is the character of the word forward and backward. Similar to MoveRight, for left and right characters. In the future I get the input axis from the project settings panel. When it started moving to make sure it followed the actual physics with the surface, I used [14]. Then he moved slowly, to move faster I assigned a maximum walking speed of 600m/s to 10. Figure 4.6 shows the main character's movement code.

```
protected:
    virtual void BeginPlay() override;

    void MoveForward(float value);
    void MoveRight(float value);

    void BeginCrouch();
    void EndCrouch();

    void Fire();

62
63 void ASCharacter::MoveForward(float value)
64 {
65     AddMovementInput(GetActorForwardVector() * value);
66 }
67
68 void ASCharacter::MoveRight(float value)
69 {
70     AddMovementInput(GetActorRightVector() * value);
71 }
72
73
74
75 void ASCharacter::Tick(float DeltaTime)
76 {
77     Super::Tick(DeltaTime);
78 }
79
80
81 void ASCharacter::SetupPlayerInputComponent(UInputComponent* PlayerInputComponent)
82 {
83     Super::SetupPlayerInputComponent(PlayerInputComponent);
84
85     //Character Movement
86     PlayerInputComponent->BindAxis("MoveForw", this, &ASCharacter::MoveForward);
87     PlayerInputComponent->BindAxis("MoveRight", this, &ASCharacter::MoveRight);
88     PlayerInputComponent->BindAxis("LookUp", this, &ASCharacter::AddControllerPitchInput);
89     PlayerInputComponent->BindAxis("Turn", this, &ASCharacter::AddControllerYawInput);
90
91     //Character Crouch
92     PlayerInputComponent->BindAction("Crouch", IE_Pressed, this, &ASCharacter::BeginCrouch);
93     PlayerInputComponent->BindAction("Crouch", IE_Released, this, &ASCharacter::EndCrouch);
94     PlayerInputComponent->BindAction("Jump", IE_Released, this, &ASCharacter::Jump);
95
96     // Fire Weapon by Call Fire function From SWeapon class
97     PlayerInputComponent->BindAction("Jump", IE_Released, this, &ASCharacter::Fire);
98 }
```

Figure 4.6: Character Movement

4.2.2 Camera Movement

Initially, when the character starts to move, the camera won't move with the character because I haven't written any code for it. It is stationary and the character moves as it is encoded. To work, I need to set the focus area of the camera, the position of the character change the way it moves, etc. for its focus area I used the search function. Another thing is the position of the camera, the distance of the character and the height of the surface of the shape. After fixing this problem, I realized that the camera movement was working but the movement was not normal. For smooth motion, I use prop configuration with [15] component. Also, every time the character moves faster, the camera will zoom out to make the user look a little faster, and when the speed drops, the camera will zoom in towards the character. When not pressing the rotate button, zoom in. Figure 4.7 shows the necessary codes for the above description.

For Character.h

```
ASCharacter::ASCharacter()
{
    PrimaryActorTick.bCanEverTick = true;

    SpringArmComp = CreateDefaultSubobject<USpringArmComponent>(TEXT("SpringArmComp"));
    SpringArmComp->SetupAttachment(RootComponent);
    SpringArmComp->bUsePawnControlRotation = true;
    GetMovementComponent()->GetNavAgentPropertiesRef().bCanCrouch = true;

    PlayerCamera = CreateDefaultSubobject<UCameraComponent>(TEXT("PlayerCamera"));
    PlayerCamera->SetupAttachment(SpringArmComp);

    // WeaponAttachSocketName = "weapon_r_muzzle";
}
```

For Character.cpp

```
protected:

    UPROPERTY(EditDefaultsOnly, BlueprintReadOnly, Category = "Camera")
    UCameraComponent* PlayerCamera;

    UPROPERTY(EditDefaultsOnly, BlueprintReadOnly, Category = "Camera")
    USpringArmComponent* SpringArmComp;
```

Figure 4.7: Camera Movement

4.2.3 Apply Jump

In the last semester, I signed up for a move character. Now to apply bread, I used another function called Jump[16] which will steal characters within a limit.. This operation will be performed when user press a key, to control from PC, I used spacebar. To do this, I created a boolean function variable named jump. Then assign it with the spacebar command. Figure 4.8 shows the back-end code to apply Jump.

```
    }  
    void ASCharacter::SetupPlayerInputComponent(UInputComponent* PlayerInputComponent)  
    {  
        Super::SetupPlayerInputComponent(PlayerInputComponent);  
  
        //Character Movement  
        PlayerInputComponent->BindAxis("MoveForw", this, &ASCharacter::MoveForward);  
        PlayerInputComponent->BindAxis("MoveRight", this, &ASCharacter::MoveRight);  
        PlayerInputComponent->BindAxis("LookUp", this, &ASCharacter::AddControllerPitchInput);  
        PlayerInputComponent->BindAxis("Turn", this, &ASCharacter::AddControllerYawInput);  
  
        //Character Crouch  
        PlayerInputComponent->BindAction("Crouch", IE_Pressed, this, &ASCharacter::BeginCrouch);  
        PlayerInputComponent->BindAction("Crouch", IE_Released, this, &ASCharacter::EndCrouch);  
        PlayerInputComponent->BindAction("Jump", IE_Released, this, &ASCharacter::Jump);  
  
        // Fire Weapon by Call Fire function From SWeapon class  
        PlayerInputComponent->BindAction("Jump", IE_Released, this, &ASCharacter::Fire);  
    }  
}
```

Figure 4.8: Back-end code for applying Jump

4.2.4 Limit Walk Speed

To limit the walking speed, the first condition is to know the walking speed. Then set the value for maximum speed. When the current speed exceeds the maximum speed limit, the speed will automatically decrease. And if the speed is lower than the maximum speed, the current speed will increase by the maximum walking speed in case the user continuously increases the current speed. Figure 4.9 shows the result of the back-end code for these operations.

Mass	100.0
Default Land Movement Mode	Walking
Default Water Movement Mode	Swimming
Advanced	
Character Movement: Walking	
Max Step Height	45.0 cm
Walkable Floor Angle	44.765083 °
Walkable Floor Z	0.71
Ground Friction	8.0
Max Walk Speed	600.0 cm/s
Max Walk Speed Crouched	300.0 cm/s
Min Analog Walk Speed	0.0 cm/s
Braking Deceleration Walking	2048.0
Sweep While Nav Walking	<input checked="" type="checkbox"/>
Can Walk Off Ledges	<input checked="" type="checkbox"/>

Figure 4.9: Limit max walk Speed

```

void APawn::AddMovementInput(FVector WorldDirection, float ScaleValue, bool bForce /*=false*/)
{
    UPawnMovementComponent* MovementComponent = GetMovementComponent();
    if (MovementComponent)
    {
        MovementComponent->AddInputVector(WorldDirection * ScaleValue, bForce);
    }
    else
    {
        Internal_AddMovementInput(WorldDirection * ScaleValue, bForce);
    }
}

```

Figure 4.10: Backend Code for Character Movement

24.2.5 The Split Zone

The area marked in the image is where the player has a choice. They can take shortcuts or long distances. Figure 4.11 the split zone

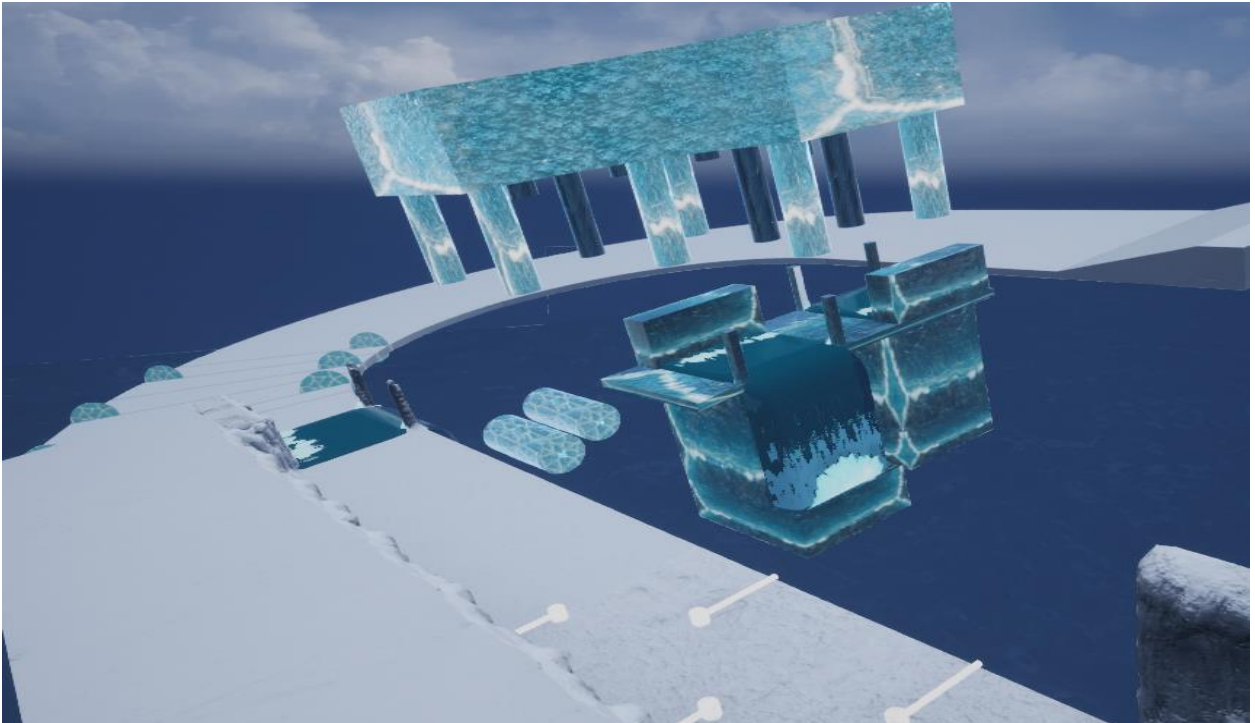


Figure 4.11: The Split Zone

4.2.6 The Circles

The area marked in the picture is where the player must jump on the spins to cross Figure 4.12 shows Circle area.

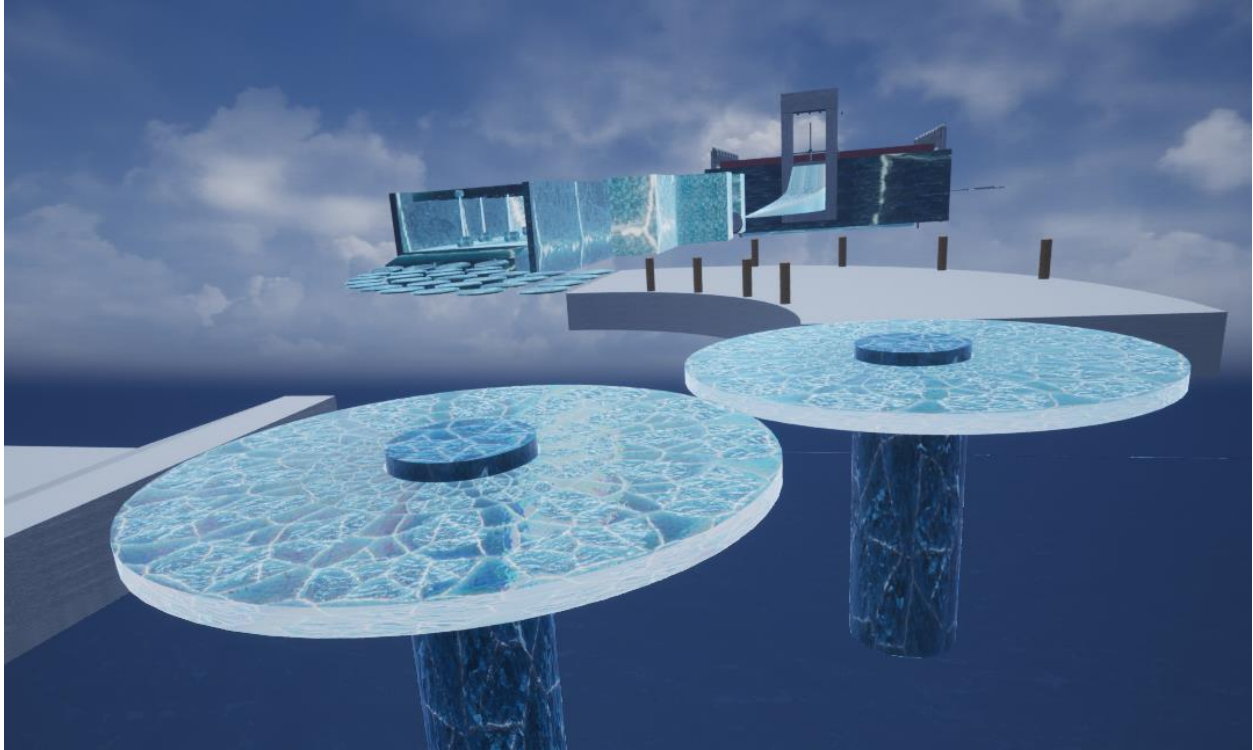


Figure 4.12: Working of the Circle

4.2.7 The Tasers

The marked area on the picture is where the player has to avoid the lightning beams to get across safely.. Figure 4.13 show the operation.

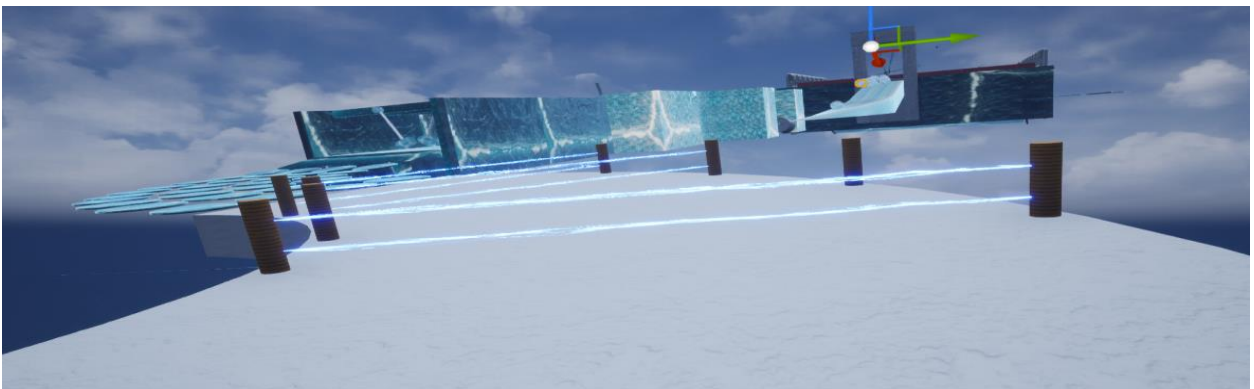


Figure 4.13: Activate the Tasers

4.2.8 The Dumb rolls

In this area player should across this place .This area have some moving dumb rolls that move up to down .Player should move carefully otherwise player will ragdoll mode 4.14 shows The Dumb rolls.

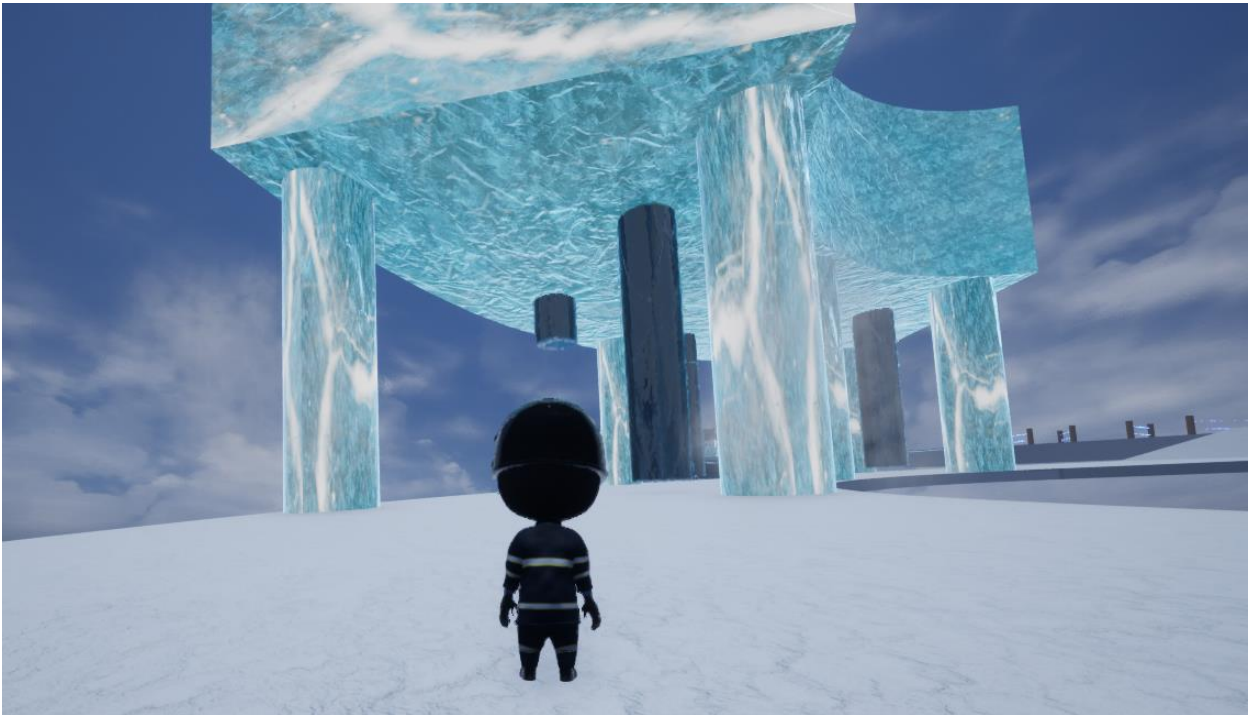


Figure 4.14: The Dumb rolls

4.2.9 The Balls and Pendulam

The marked area on the picture is where the player has to avoid balls. If ball touch to character ,character go to ragdoll mode. Figure 4.15 shows that.

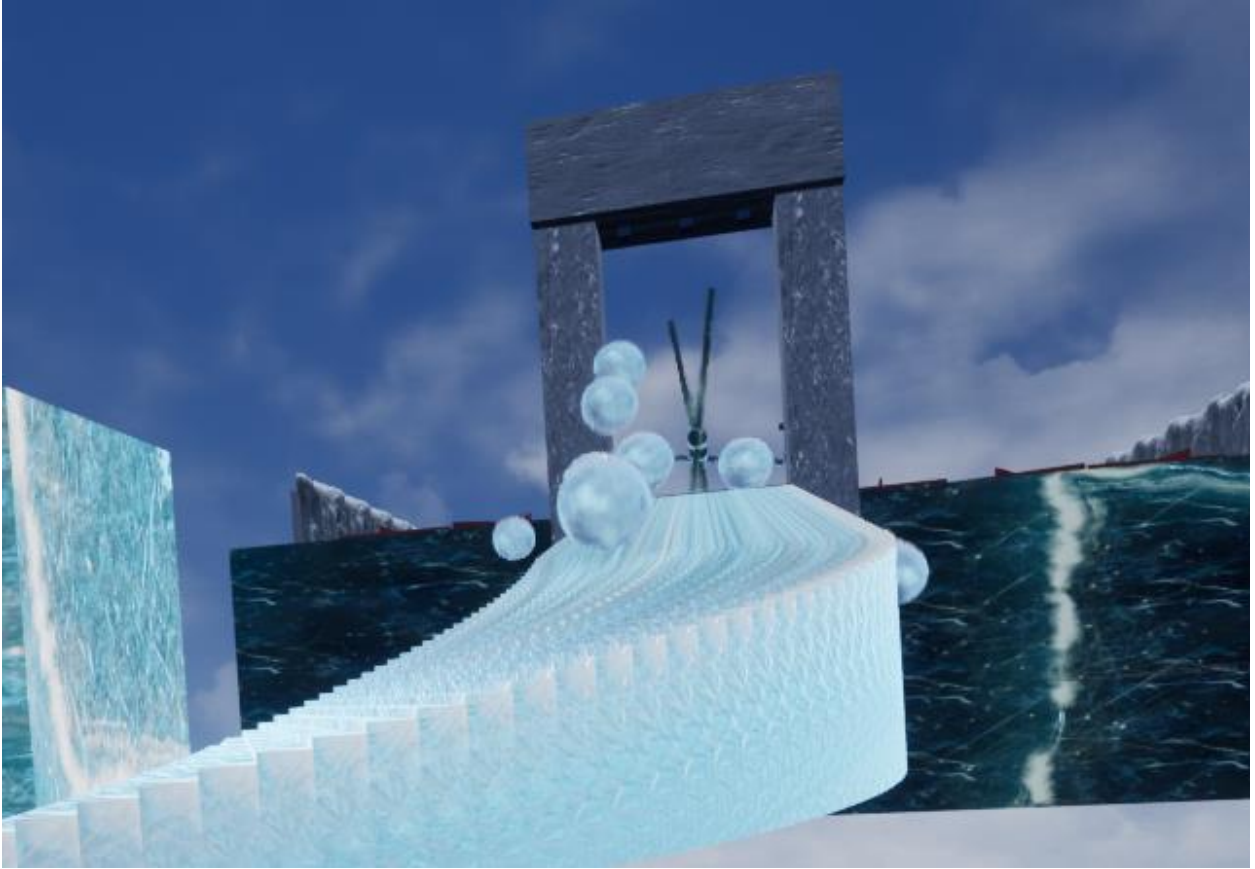


Figure 4.15: The Ball



Figure 4.16: The pendulum

4.2.10 Road Preview Camera

What to do next and where will your next destination be? To answer these questions, I added another camera so you can find these instructions. In this part complete by unreal engine blueprint. Figure 4.17 shows back-end code.

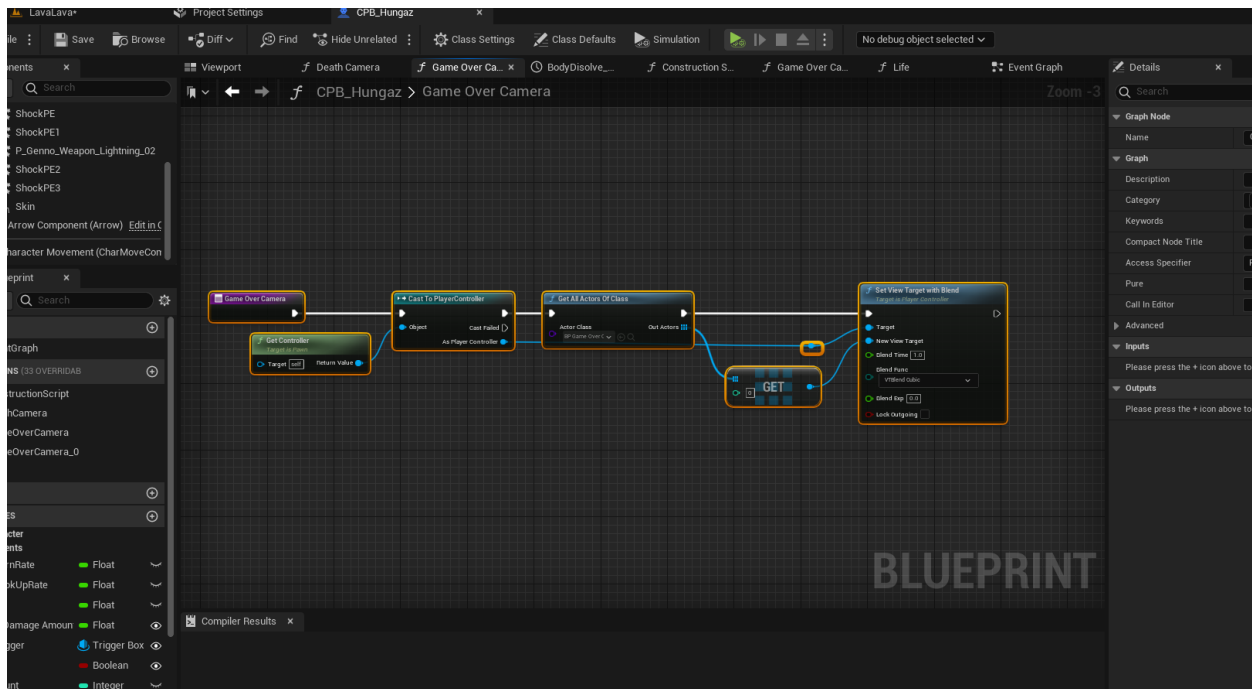


Figure 4.17: Road Preview Camera

4.2.11 Character Shock Effect

For making the Character more realistic Shocked add skeletal when character touch tasers inside skeletal mesh shown .. Figure 4.18 shows the tasers



Figure 4.18: Taser Effect



Figure 4.19: Show inner Skeletal

4.2.12 The Turbines

At the picture this turbine rotation .Player should across this turbine .if player hit this turbine character going to ragdoll mode. The marked area on the picture is where the player has to avoid the turbines. Following figure 4.19.

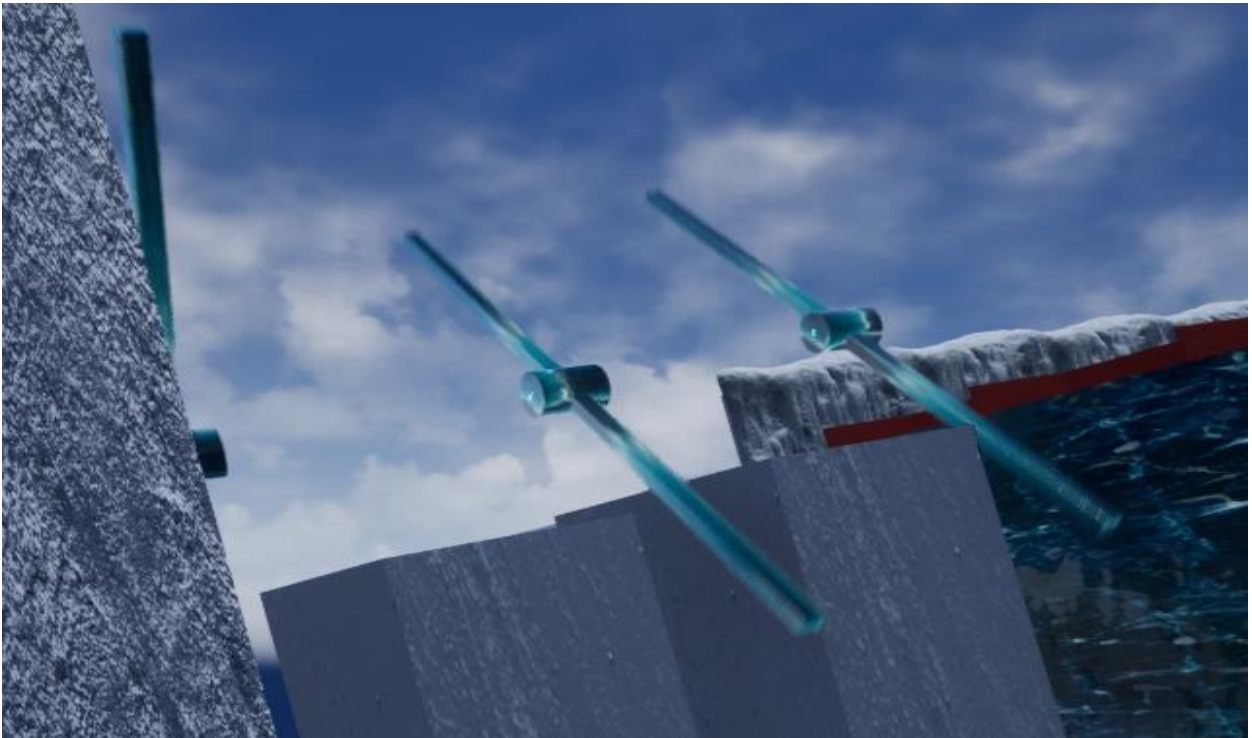


Figure 4.20: The Turbines

4.3 Interaction Design and UX

It represents the UMG of the application that uses it. Interaction design deals with the overall process of the system as it is performed by the user. A better interaction design depends on the overall outcome of the system.

UX stands for user experience. This term is very important for the design of any system. Because it changes over time. The old design or UI makes their mind monotonous.

To get rid of it, UI modifications can be applied. In my game, I try to maintain the elements that make the user happy.

4.4 Implementation Requirements

To create this Rush Race game, I used Unreal Engine 5, Blender and Photoshop. All this is not easy for someone at a beginner level. To make the Obstacle Blender. Next, convert the blend file to .fbx. The game environment and backend coding were then implemented on Unreal Engine 5. To create the image, Photoshop was used, to create the material using textures for the environment. I am working on this project using these softwares. Browse the content store for different grid assets if needed.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation of Database

Since my project is not information storage I am not using any database here. My project theme is made from the game Rush Run. Instead, little of our location information is stored as a string.

5.2 Implementation of Front-End Design

As I discussed the front end design of my game, here I am talking about the main menu section. Figure 5.1 shows this part.

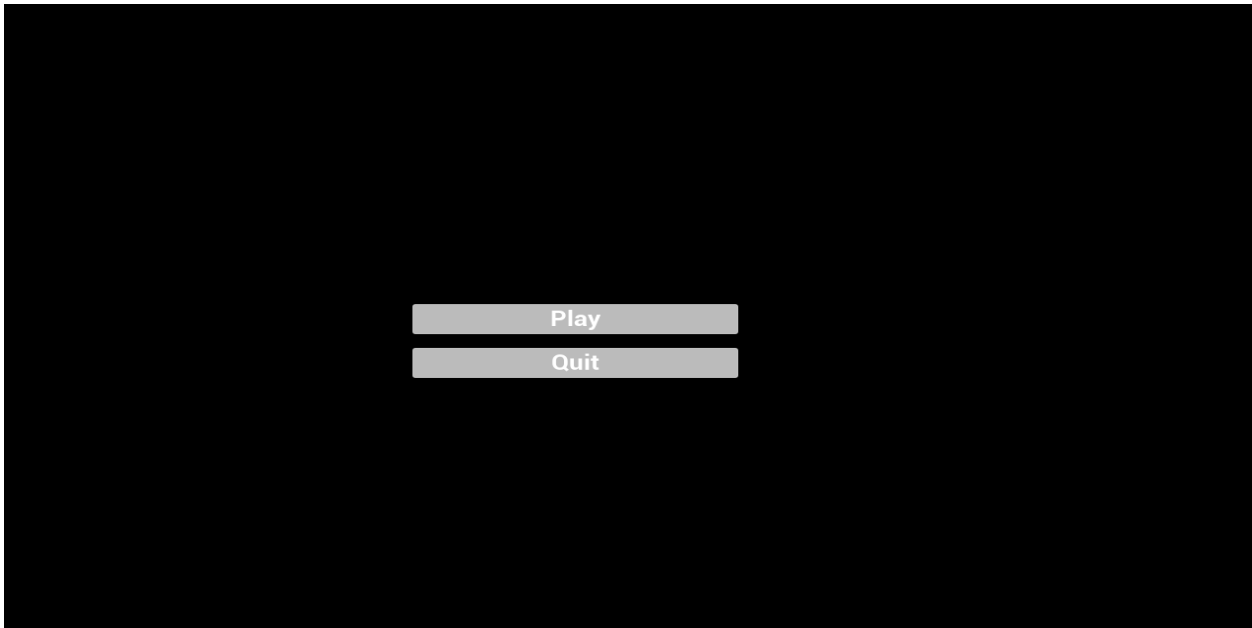


Figure 5.1: Main Menu user interface

5.3 Implementation of Interactions

User interaction with a system depends on certain factors. The more it is used, the more interactive the application becomes. I tried to make our game more user-friendly. the user will have to re-read the script to answer correctly.

5.4 Testing Implementation

Here I will look at some cases where testers will look at the case and specs. Under these conditions play will be continued or interrupted. Table 5.1 shows the test execution.

Table 5.1: Testing Implementation

Test Case	Test Input	Expected Outcome	Actual Output	Result	Tasted on
1. Open game	Information Displayed	Game to be started	Not started	Failed	06-11-18
2. Play game	Wrong answer	Game to be started	Not started	Failed	06-11-18
3. Play game	Correct answer	Game to be started	Started	Successful	06-11-18

5.5 Test Results and Reports

I have tested my system under what conditions it will boot or not. Whenever the user can give the correct answer, the game starts the other way. The user will then have to read the information again to play the game.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Discussion and Conclusion

This Rush Race game has two platforms, one is entertainment and the other is Learn. Hopefully it can give players a different feeling when playing. Developing a racing game Rush Run is not an easy task, especially for beginners. In my view through this project I learned Unreal Engine 5, Photoshop and the use of Blender. One word that needs to be said about Unreal Engine 5 is that this IDE is very advanced compared to many other IDEs I have learned so far. Although all three are not easy to learn at first. Time and time again I ran into problems and by restoring them I became more compact.

Games can also prove to be a great platform for all users to learn through entertainment in their spare time. Finally, I can say that I have learned important things about topics that will help us in our professional lives.

6.2 Scope for Further Developments

I want to continue to develop this game, some of my plans are given below:

- Added more obstacles.
- Level up.
- Our ultimate goal is to make it professional in every way and keep it on Google Play Store.
- Matching menu system
- Point system
- Multiplayer system

REFERENCES

- [1] “FallGuys Season 3: Sunken Secrets Available <https://www.fallguys.com/en-US> [Last accessed on Nov. 1, 2022 at 12.31 am].
- [2] “Stumble Guys”. [Online]. Available: <https://play.google.com/store/apps/details?id=com.kitkagames.fallbuddies&hl=en&gl=US> [Last accessed on March. 17, 2022 at 09.40 pm].
- [3] “Human: Fall Flat on Steam”. [Online]. Available <https://play.google.com/store/apps/details?id=com.and.games505.humanfallflat&hl=en&gl=US>. [Last accessed on Dec. 17, 2022 at 10.11pm].
- [4] “Blender”. [Online]. Available: <https://www.blender.org/>. [Last accessed on Jan. 25, 2022 at 01.06 am].
- [5] “Photoshop”. [Online]. Available: <https://www.adobe.com/products/photoshop.html>. [Last accessed on Mar. 12, 20122 at 8.30 pm].
- [6] “Unreal Engine”. [Online]. Available: <https://www.unrealengine.com/en-US/> [Last accessed on Mar. 22, 2019 at 8.00 pm].
- [7] “C++”. [Online]. Available: [https://www.learn-c++.org.](https://www.learn-c++.org/) [Last accessed on Nov. 05, 2020 at 11.30 pm].
- [8] “Blueprints”. [Online]. Available <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/GettingStarted/>. [Last accessed on Nov.18, 2020 at 10.15 pm].
- [9] “Unreal Engine Character movement Function ”. [Online]. Available: <https://forums.unrealengine.com/t/changing-character-movement-speed-in-c/611354> [Last accessed on Jan. 01, 2022 at 09.32 am].
- [10] “Functional Vs Non-functional requirements”. [Online]. Available: <https://reqtest.com/requirements-blog/understanding-the-difference-between-functional-and-non-functional-requirements/>. [Last accessed on Oct. 07, 2020 at 11.20 pm].
- [11] “Ice texture”. [Online]. Available: <https://www.unrealengine.com/marketplace/en-US/product/ice-cool>: [Last accessed on Feb. 04, 2021 at 1.00 pm].
- [12] “UV editing”. [Online]. <https://www.youtube.com/watch?v=j8C6B-yAyPA>. [Last accessed on March. 04, 2021 at 11.00 am].
- [13] “Character”. [Online]. Available <https://www.mixamo.com/#/>. [Last accessed on Jan. 20, 2022 at 07.30 am].

APPENDIX

Appendix A: Project Reflection

The purpose of this appendix is about project reflection. From Fall 2022 semester I have started our journey to make this Rush run game. Our vision is making a different type of a game that I am normally used to play.

One can entertain and playing game at a time. There are a lot of struggles I faced while making this game as a beginner. I am going forward step by step. After many hard work and spending a lot of time finally I am able to reach my goal.

The project "Rush Run" will be very helpful for all types of Bangladeshi game lovers. Hopefully they can spend their leisure by learning and entertaining themselves. So, I believe that my game "Rush Run" will be a positive game and will be appreciated and accepted by all age's people.

Plagiarism Report

Checked by: <https://my.plagramme.com>

RUSH RUN: AN PC RUSH RUN GAME

ORIGINALITY REPORT

23%

SIMILARITY INDEX

23%

INTERNET SOURCES

0%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

dspace.daffodilvarsity.edu.bd:8080

Internet Source

23%

Exclude quotes On

Exclude matches Off

Exclude bibliography On