A MAJOR PROJECT REPORT
ON
3D Building Model Using UNITY 3D

SUBMITTED IN PARTIAL FULFILLMENT FOR THE AWARD OF
DEGREE OF
BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted By:
KAPIL SHARMA (9917102148)
SANCHIT GUPTA (9917102167)
SATYAM GUPTA (9917102181)

Under the guidance of :
Mr. VARUN GOEL

Department of Electronics and Communication Engineering
Jaypee Institute of Information Technology, Noida,
MAY 2021

CERTIFICATE

This is to certify that the major project report entitled, "**3D Building Model using UNITY 3D**" submitted by **"Kapil Sharma"** ,**"Sanchit Gupta"** and **"Satyam Gupta"** in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Electronics and Communication Engineering of the Jaypee Institute of Information Technology, Noida is an authentic work carried out by them under my supervision and guidance. The matter embodied in this report is original and has not been submitted for the award of any other degree.

_____

Signature of Supervisor(s)

Name:Mr. Varun Goel
ECE Department
JIIT Sec-128
Noida-201304

DECLARATION

We hereby declare that this written submission represents our own ideas in our own words and where others' ideas or words have been included, have been adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission.

Place : Noida

Name : Kapil Sharma
Enrollment No. : 9917102148

Name : Sanchit Gupta
Enrollment No. : 9917102167

Name : Satyam Gupta
Enrollment No. : 9917102181

ABSTRACT

We can see recently these days that information and communication technology support the development of human interaction with physical, computer and virtual environment such as science, banking, education, etc. 3D viewing of a building type model has been used by many for designing, constructing and enhancing the overall structure. Our project tries to create a 3D model of our institute building and focus on viewing it as a third person view. Our project aims to eliminate the physical effort by human to visit the architecture.
The project starts with discussing the problem we currently facing due to unavailability of 3D model of our college architecture. The project continues to analyze the problem and use the handful of software tools to get started. Using the information from around the internet and some papers help to gain knowledge and suitable yet effective ideas were implemented. The result of the project can be useful in viewing the 3D model of our college and can be used for further advancements in the architecture. The project concludes with the results which has been obtained and evaluation of the methods and the tools that has been used in this project.

ACKNOWLEDGEMENT

We would like to take this opportunity to express our profound gratitude and deep regards to our guide and mentor **Mr. Varun Goel** for his exemplary guidance , monitoring and constant encouragement throughout the course of this project.We owe the success of this project to him and are indebted to him for his blessings, help and guidance.

We would also like to extend our gratefulness to our parents and family for their persistent love and support.

Contents

List of Figures

Chapter 1

INTRODUCTION

The 3D modeling industry has had much growth in recent years. It's a great industry to dive into as it allows innovation, flexibility, creativity and freedom for the developers and designers. Here we get a handful chance to get familiar with all forms of media like animations, sound, design and programming. [1] On the same hand the progress in the on going researches on 3D technologies for building modeling types and platforms like "Unity 3D" game engine will lead to the implementation of great variety of 3D application on a bigger scale. Today it's become a part of designers and construction workers to work with 3D model of the space. Handiness and Easiness of 3D models for the business and authority operations directly highlights the benefits of 3D models. The ground floor of JIIT-128 campus has been created using snaps and 3D animations has been imported to move around the space as shown in Figure 1.1.

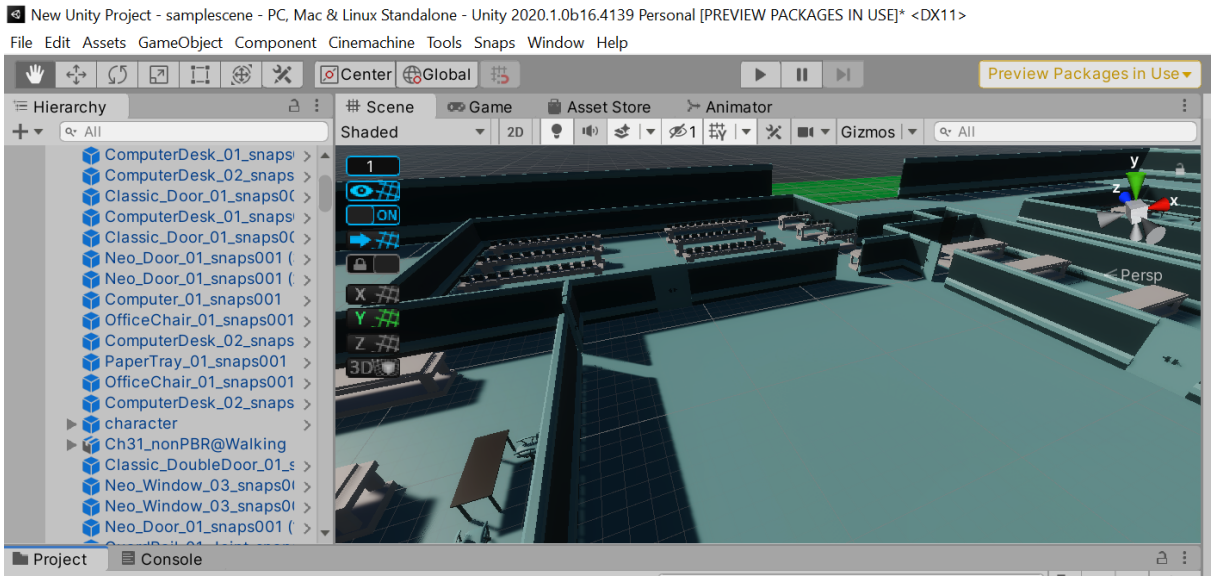


Figure 1.1: Ground floor prototype of JIIT-128

It provides a whole new experience for the user to view the 3D model of our college and move around it as a player in third person view.

1.1 Problem Definition

We know that working on the 2D architecture has become outdated in today's world where everything is becoming 3D. Even though there are certain applications for 2D modeling like architectural design, bioprinting in medical industry, car designing etc.

but they cost heavier in term of time consumed and workdone. So to ease the work and for the better experience the 3D models are being used by developers nowdays.

1.2 Existing Systems

Previously the user doesn't have faster 3D rendering software and they rely on assisted manpower for 2D understanding of the image. Geometry plays a huge role in 2D understanding whereas people with better spatial awareness go for 3D as a better option. Mathematics plays a huge role in 3D modeling.
Certain Setbacks for 2D modeling –
• Multiple models for each space view. When changes happen, they need to be reflected in all drawing views.
• Redundancy of the information is also the problem which causes confusion and is less accurate. 2D drawings aren't capable of capturing the complexities of product design.
• Less accessible for the user and consumes a lot more time than the 3D models. Designs may need to be recreated numerous times in different views to capture the details of single part.

1.3 Proposed Systems

In view of the need for the 3D model of our JIIT-128 campus, we created the ground floor looking approximately same as the real one. Interior architecture being one of the applications for 3D modeling, developers prefers using 3D rendering software as their first choice. The gaming engine used to develop the project was Unity 3D, as the large community and posts has helped us in solving software related queries.[2] 3D models allow us to utilize 3D printing and rapid prototypes. Rapid prototypes are the working models of a part that is used to evaluate and analyze the designs being created before going through manufacturing or any further changes. Using 3D models, photorealistic rendering is able to give a lot of information making it easier for a layman person to understand the design. 3D modeling software allow us to view a design from every angle.[3]
Steps to be followed –
• Importing prototype snaps packs from unity asset store to start prototyping.
• Placing of the objects using snaps in unity grid.
• Importing animated character from the Mixamo.
• Setting up the freelook camera from cinemachine for the third person viewing.
• Programming for the movement of the player across the space.
• Importing the animation movements and setting them in place.

Chapter 2

Hardware and Software Required

2.1 Hardware Required

1. RAM Capacity: Minimum 4GB
2. Graphics card: 1 GB
3. Accessories: Smartphone with AR support

2.2 Software Required

1. Android version: Android 8.0 or above
2. Tool 1: Unity-3D
3. Tool 2: Animations from Mixamo

2.3 Unity-3D

Unity-3D is a "game development ecosystem", it includes an environment for the development of interactive 2D and 3D content including a rendering and physics engine, a scripting interface to program interactive content, a content exporter for many platforms (desktop, web, mobile) and a growing knowledge Unity-3D introduces new tools that help artists & designers tell better visual stories, new ways for teams to collaborate more productively, and more features than ever to help you succeed in the gaming industry and interactive games development.[4]

The main editor window is made up of tabbed windows which can be rearranged, grouped, detached and docked. This means the look of the editor can be different from one project to the next, or one developer to the next, depending on personal preference and what type of work we are doing. The default arrangement of windows gives you practical access to the most common windows. If ones are not yet familiar with the different windows in Unity, they can identify them by the name in the tab. The Project window in unity helps us to displays our library of assets that are available to use in our project. When we import assets into your project, they appear here.[5]

2.4 Pro Builder and Pro Grids

Pro Builder is an tool for quickly modeling assets and levels within the unity editor. It is optimized for building simple geometry but capable of UV unwrapping and detailed editing.
We can easily snap objects on the Unity using pro grids, so that we have a whole bunch of interactive buttons to interact with the grids.[6]

2.5 HDRP and Mixamo

The High Definition Rendered Pipeline is a high fidelity scriptable rendered pipeline built by unity to target modern platforms. It gives us tool needed to create games, animation, technical demos and high graphical standards. HDRP utilizes physical based lighting techniques, linear lighting.[7]
Mixamo is a web-based service that helps the developers to animate the 3D characters by making rigging and animations easier. It is extremely user friendly and aimed at people to create animations and make character move.

Chapter 3

Work Done and Methodology

The project implementation consists of 4 modules :

1. Importing packages from asset store.
2. Developing prototype for user interface.
3. Controlling the character inside the space.
   - Input movement
   - Normalizing the vector
   - Rotation
   - Rigid Body
4. Setting up the camera look

3.1 Importing packages from asset store

Importing the packages from unity asset store like Pro grids, Pro Builder was pre-required. Snaps were the pre-made assets designed specifically for prototyping. The combinations of different variety of assets were required to create unique 3D levels. And unity gives us this huge flexibility to change or modify the assets according to our needs. Prototype snap pack which was used in our project was "Office" for creating the college type environment. The Unity-3D version which is used here was 'Unity 2020.1'.
After importing, all folders were automatically added to the project files. Thus we got all our structures like floors, ceilings and many other props to work upon. Then with the help of pro grids and pro builder, we can easily snap assets.

3.2 Developing prototype for user interface

Under prefabs, then structures the floor gets added. Further variations of floor were added to make it look better. After flooring was complete the walls were dragged and dropped. The walls were rotated and moved to two levels to make the ceiling a little up. Doors were rightly chosen and got them fit between the walls. Same way the whole structure was created with multiple variations of the objects.

3.2.1 Floors

As shown in Figure 3.1 & Figure 3.2, under the prefab component floor and ceiling tab was selected. The deck floor1 & deck floor2 was selected and dragged into the space. After clicking on the transform, we reset it to snap the floor. For this we open up tools and clicked on progrids window and enable snapping. The snap value of our grid was set according to the size of floor tile used.

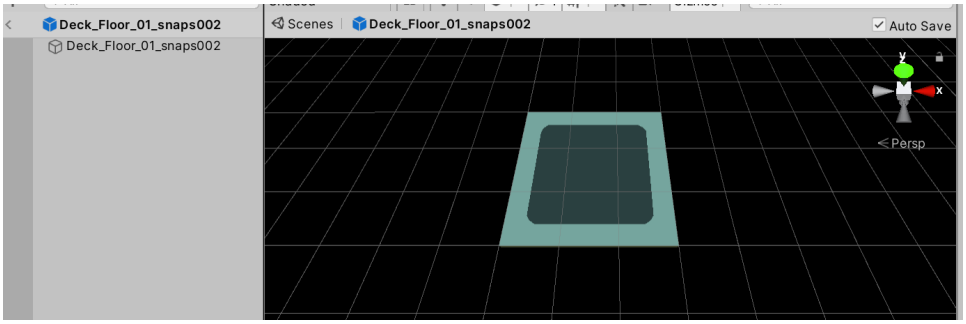


Figure 3.1: Deck Floor 01

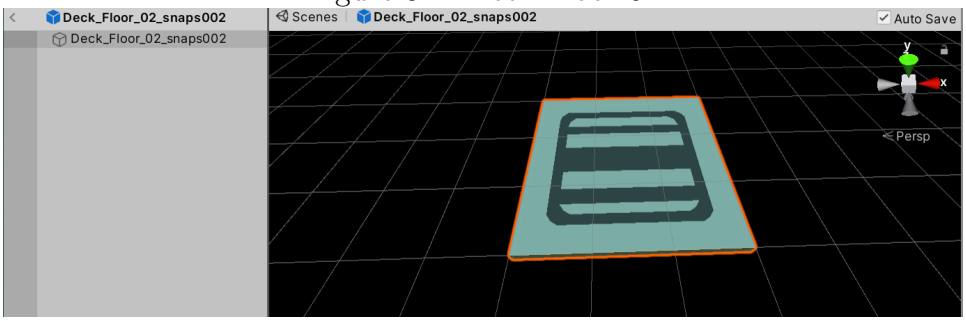


Figure 3.2: Deck Floor 02

3.2.2 Walls

As shown in Figure 3.3 & Figure 3.4, under the structure component we simply drag and drop the Gale wall2 and Deck wall. We duplicate the walls by ctrl+D short key. The spaces were leaved in between the walls so as to insert the doors or for the passage between the walls as to in and out. To switch to their rotation tools, walls were hold control while rotating by pressing keyword 'E'. To get the 2 level rooms i.e. in the case of MPH if JIIT-128, all the walls were selected and were moved up.

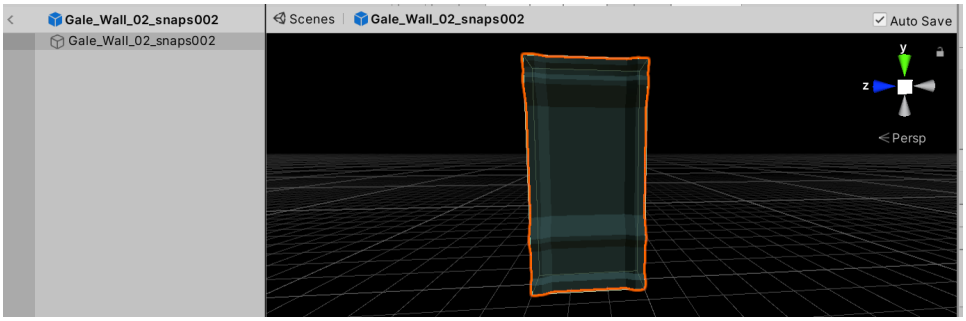


Figure 3.3: Gale Wall 02

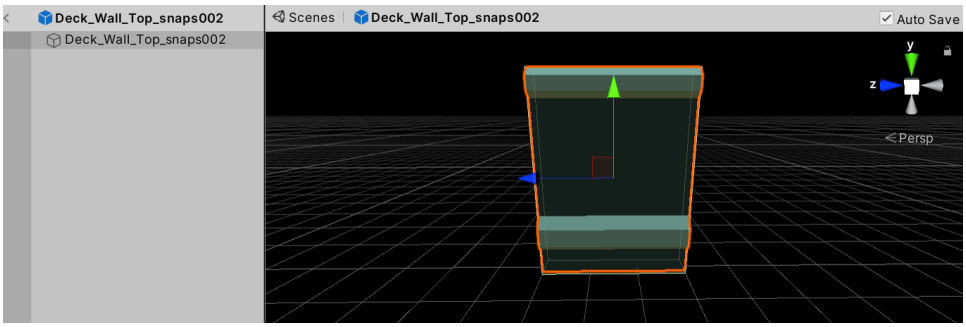


Figure 3.4: Deck Wall

3.2.3 Doors

As shown in Figure 3.5 & Figure 3.6 the Neo double door 01 & Neo door 01 were selected under the prefab component and were placed into the level. They were moved by avoiding snapping and so as to perfectly fits on the door frame. These modular real-time assets are great for use in any 3D productions like advertising, games, media etc.

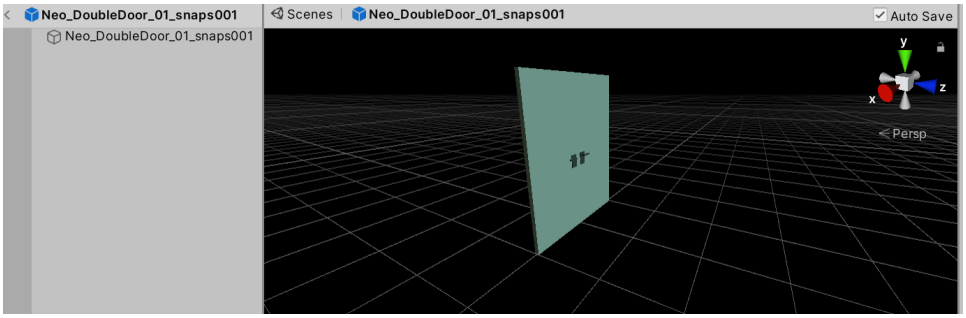


Figure 3.5: Neo Double Door 01

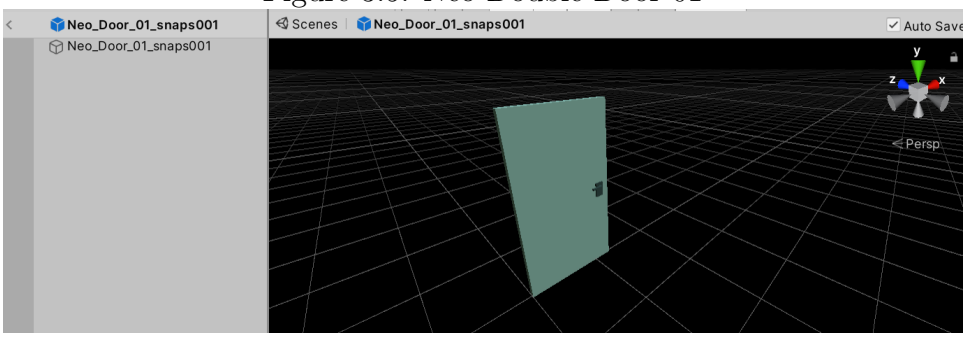


Figure 3.6: Neo Door 01

3.2.4 Rails

By choosing the prop component under prefab the bunch of objects were selected for guardrails as shown in Figure 3.7 & Figure 3.8. For example Guardrail joint 01 was selected and scattered around to create guardrails. After that Guardrail end 01 were placed at the end manually.

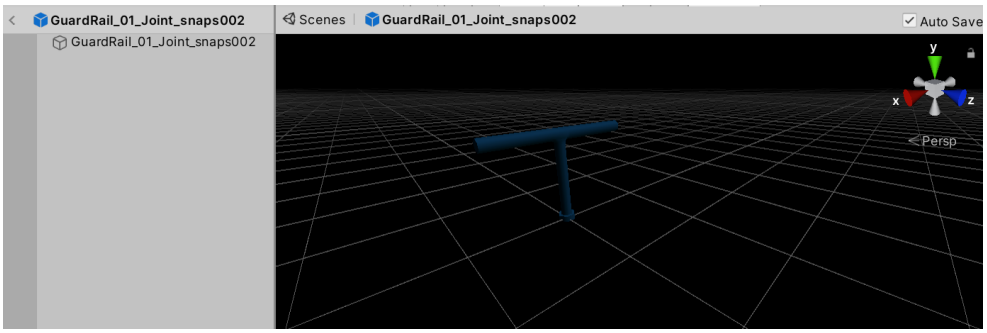


Figure 3.7: GuardRail Joint 01

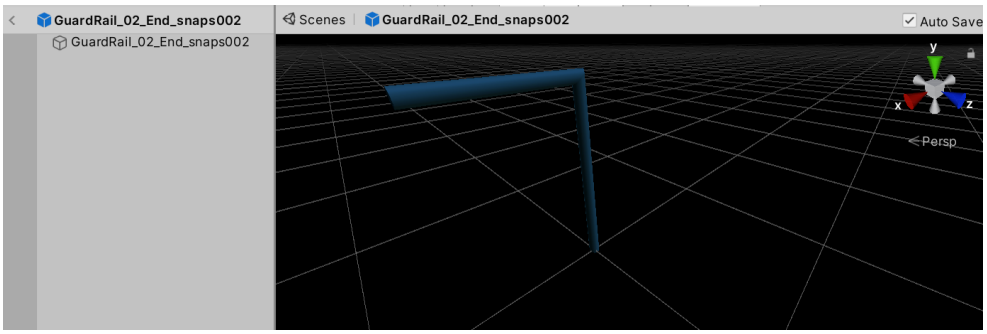


Figure 3.8: GuardRail End 01

3.2.5 Tables

As shown in Figure 3.9 the table was selected and was placed on the level. In this case the computer Desk 01 was placed in the computer lab and monitors were put on them. The size and the length of the table were adjusted accordingly. This object was fully textured and materials were tailored and applied for high-quality render results.

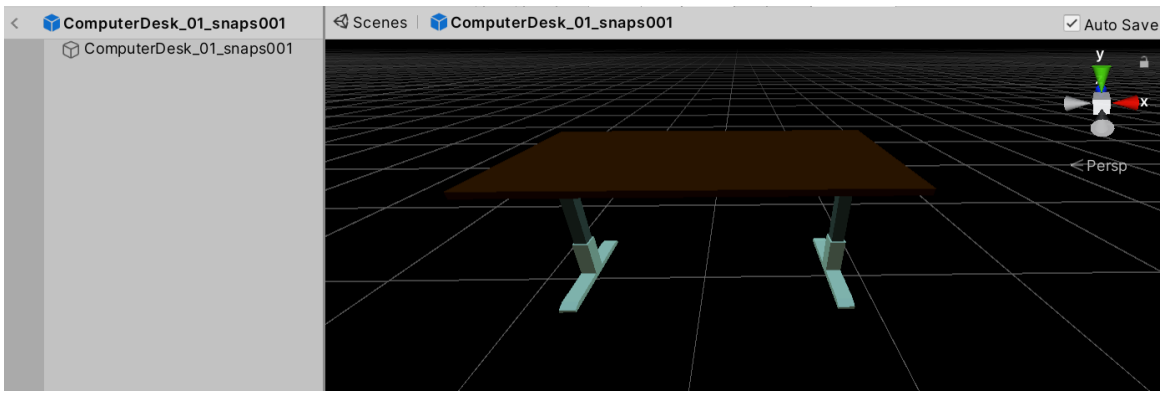


Figure 3.9: Computer Desk 01

To make the area lit up and to pop up the scene, a bunch of coloured point lights were used. Point lights shine out in all direction and the colour of the point lights were set according to the use.

3.3 Controlling the character inside the space

A character prefab was imported from 'Mixamo'. Character used here was "Leonard" which is of humanoid type. To move the character the basic step taken was to change transform positions in X, Y and Z axis.

3.3.1 Input Movement

float horizontal = Input.GetAxis("Horizontal");
float vertical = Input.GetAxis("Vertical");
Vector3 movement = new Vector3(horizontal, 0f, vertical);
After taking the input movement from "W-A-S-D" into 2 axes, we convert them into vector so that W-S go along vertical axis and A-D go along horizontal axis.

3.3.2 Normalizing the Vector

After that if magnitude of the movement is greater than zero, we normalize it just to turn it into a direction rather than a magnitude. Moving diagonally directly changes X & Z direction by 1 unit. Normalizing the vector will squeeze the value of X and Z approximately to 0.7 units, thus making the magnitude of our final movement vector to 1.
float velocityZ = Vector3.Dot(movement.normalized, transform.forward);
float velocityX = Vector3.Dot(movement.normalized, transform.right);
_animator.SetFloat("VelocityZ", velocityZ, 0.1f, Time.deltaTime);
_animator.SetFloat("VelocityX", velocityX, 0.1f, Time.deltaTime);
To move the object smoothly on the surface we multiply the speed with Time.deltaTime. It is the amount of time in sec since the last update / frame was called. Further translate it to move the character in the world space. Vector3.dot gives us the dot product of two vectors. For animation we created a blend tree and add different types of motion as shown in Figure 3.10.

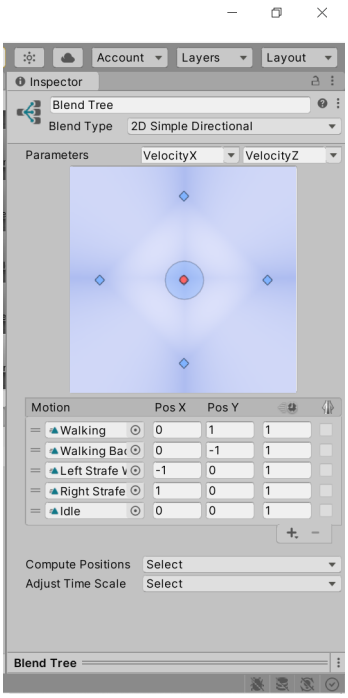


Figure 3.10: Position Values

3.3.3 Rotation

The rotation of the character to face the direction in which it was travelling is much needed. For this we got to know how much angle we must rotate our object in Y-axis using Atan2 function.
float targetAngle = Mathf.Atan2(movement.x,movement.z) * Mathf.Rad2Deg;
transform.rotation = Quaternion.Euler(0f,targetAngle,0f);
Atan2 is a mathematical function that returns the angle between x-axis and the vector (starting from origin to (x,y)) and further converting that angle from radians to degree. After getting that angle we can rotate the character by putting that rotation around y-axis in 'Quaternion.Euler' function, which allows to input rotation around x, y and z axis.

3.3.4 Rigid Body

To move the character according to physics we add rigid body component to it. Mass of the body is set to one as more mass means more force is required to move the character.

3.4 Setting up the camera look

Ray ray=Camera.main.ScreenPointToRay(Input.mousePosition);
In 'AimTowardMouse' function, we sent a ray to our input mouse position on the screen and get the direction by subtracting the hit direction to the current direction. After that we send forward to that direction.

Figure 3.11: Cinemachine Virtual Camera

For the interactive camera movement we used ‘Cinemachine’ . After importing it we created a cinemachine virtual camera as shown in Figure 3.11. This acted as a brain component for the main camera, to tell which direction to look on. We made the virtual camera to follow the character and constantly look at our character.

3.5 Flow Chart

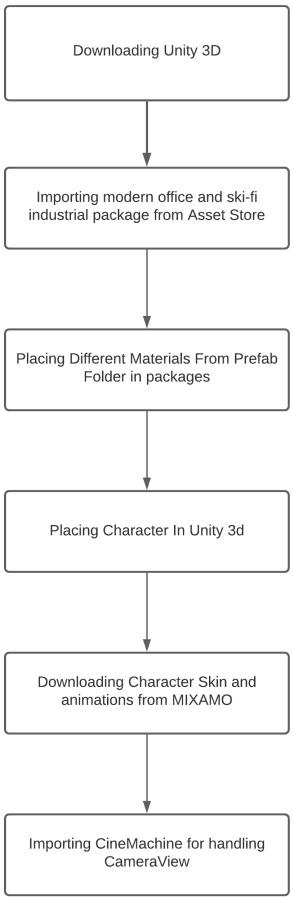


Figure 3.12: Flow-Chart Of Working Process Of the Project

As shown in Figure 3.12 there are multiple steps involved in the making of this project.
First Step was to download the unity 3d software as a platform to perform further steps.After downloading unity next step was to import modern office and ski-fi industrial packages from the asset store of unity after importing these packages next task was to place objects for example objects like Table, Chair, Walls and Computers etc. After making structure of JIIT-128 our next task was to make a robot player which will move in the campus so that the user will get a better and high class experience of the campus and also user can control the player with keyboard and can roam around the campus.
After importing the player next task was to give the player a user friendly look. For that we used mixamo for downloading animations and player skin. Now the next task was to handle the camera so that camera view will be connected to the player. For this we used cinemachine and imported it into the camera view of the player.

Chapter 4

Result and Discussion

As we can see in the Figure 4.1 the player is being controlled with the keyboard controls. The player was looked moving forward towards Annapurna and library. When we press ‘w’ the player moves forward, ‘s’ is for moving backward, ‘a’ is for left side movement, ‘d’ is for right side movement. Movement in any direction on the level can be achieved by multiple combination of keys. By using ‘Mixamo’ a particular skin was given to the player and with the same platform we used animations for movement, so that it will look real and user friendly.

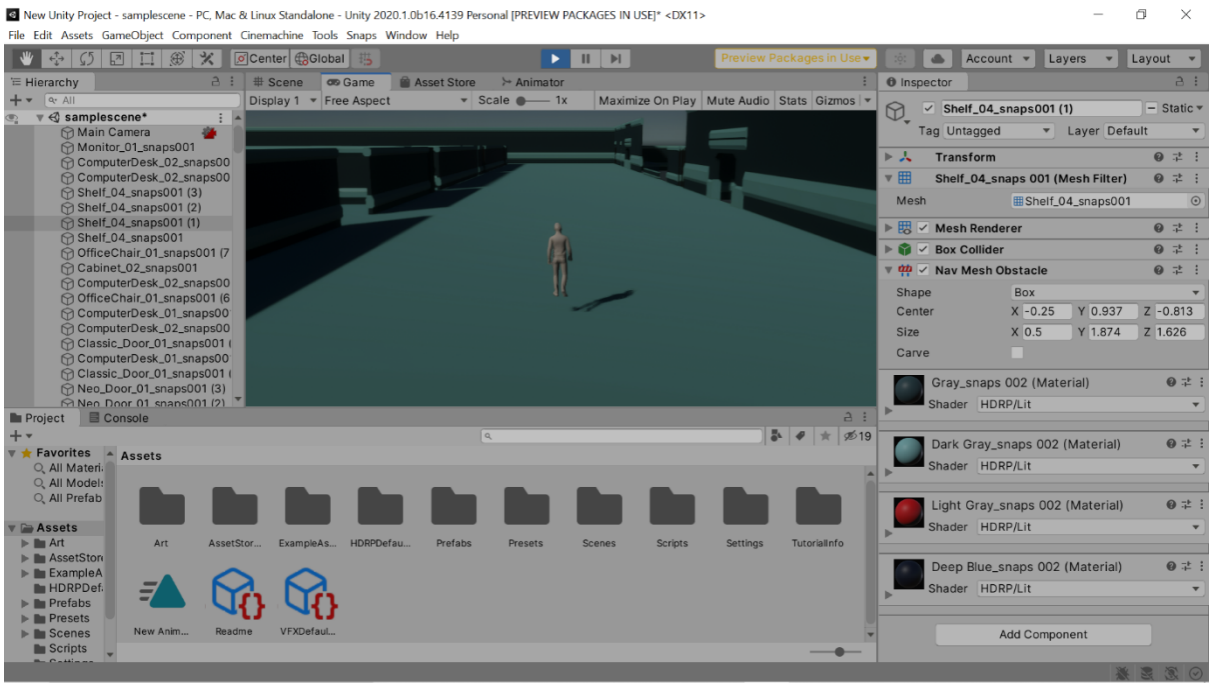


Figure 4.1: Player movement in JIIT-128 campus

As shown in Figure 4.2 the player was standing in front of Annapurna and we can move the player inside the serving area also using the same controls. Initially when the player is at rest it shows ‘Idle’ position, where no movement of the character can be traced. Further animations were shown according to the change in position of X & Z vectors. By this a holistic view of ground floor JIIT campus was obtained.

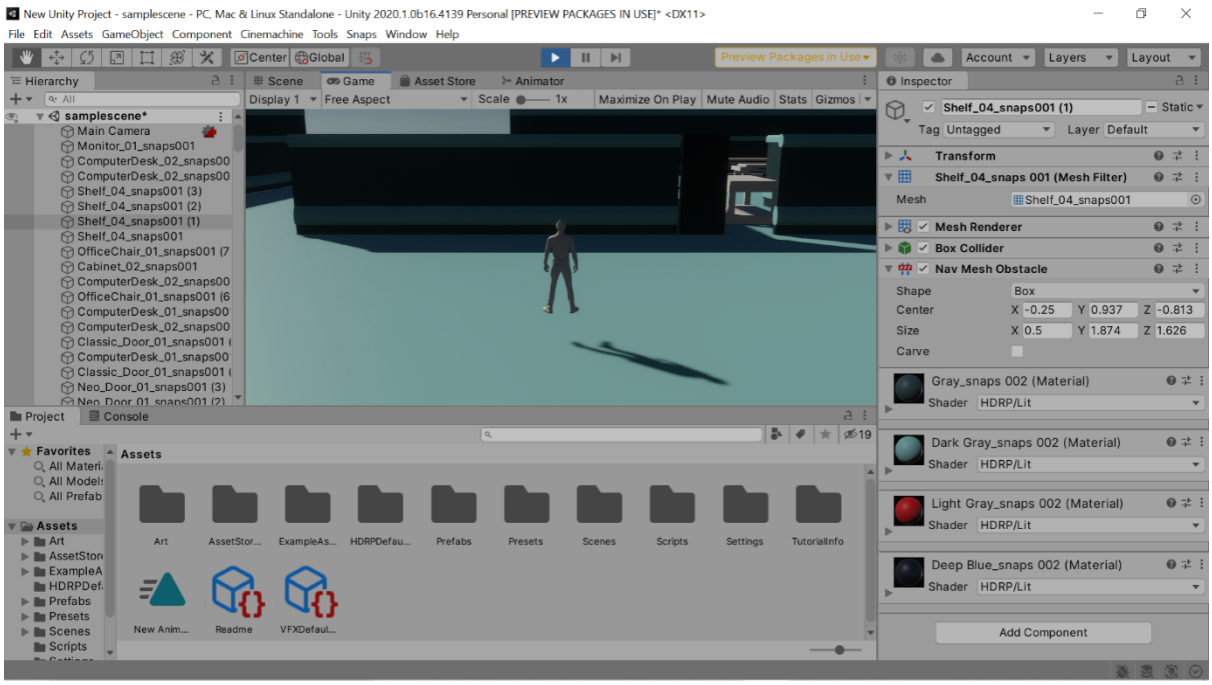


Figure 4.2: Player in front of Annapurna

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

The main objective of this "3d building model using unity 3d" is to provide the user the understanding of a place which he is interested in or want to be a part of it. 3d model building technology allows the users to decide and interact with the structure and can take a virtual tour of the place. It helps the user to view and understand the structure for his better understanding.
Prototyping is an experimental process where design teams implement ideas into tangible forms from paper to digital. Teams build prototypes of varying degrees of fidelity to capture design concepts and test on users. With prototypes, we can refine and validate our designs so we can release the right products.
Prototyping is a new evolving technology in the field of game development and will be much more helpful than traditional technologies.

5.2 Future Scope

In future, our "3d building model using unity 3d" dataset and scope will be scalable. The user might not only be able to move a player in that place but they can also take a virtual tour of that place. It can also be used for various applications in virtual tour of a famous place or provide a better understanding of a certain place using Virtual Reality. New technology may come into existence in future that will help in developing 3D models and make them more realistic.

Bibliography

[1] M. Fachri and A. Khumaidi, "Performance analysis of navigation ai on commercial game engine: Autodesk stingray and unity3d," vol. 4, pp. 61–68, May 2020.

[2] G. Lu, G. Xue, and Z. Chen, "Design and implementation of virtual interactive scene based on unity 3d," *Advanced Materials Research*, vol. 317-319, pp. 2162–2167, Aug. 2011. DOI: 10.4028/www.scientific.net/AMR.317-319.2162.

[3] F. Messaoudi, G. Simon, and A. Ksentini, "Dissecting games engines: The case of unity3d," in *2015 International Workshop on Network and Systems Support for Games (NetGames)*, 2015, pp. 1–6. DOI: 10.1109/NetGames.2015.7382990.

[4] J. Xie, "Research on key technologies base unity3d game engine," in *2012 7th International Conference on Computer Science Education (ICCSE)*, 2012, pp. 695–699. DOI: 10.1109/ICCSE.2012.6295169.

[5] P. Xia, "3d game development with unity : A case study: A first-person shooter (fps) game," 2014.

[6] C. Bartneck, M. Soucy, K. Fleuret, and E. B. Sandoval, "The robot engine — making the unity 3d game engine work for hri," in *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2015, pp. 431–437. DOI: 10.1109/ROMAN.2015.7333561.

[7] I. Buyuksalih, S. Bayburt, G. Buyuksalih, A. P. Baskaraca, H. Karim, and A. A. Rahman, "3d modelling and visualization based on the unity game engine@ advantages and challenges," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, pp. 161–166, 2017.

Appendix A

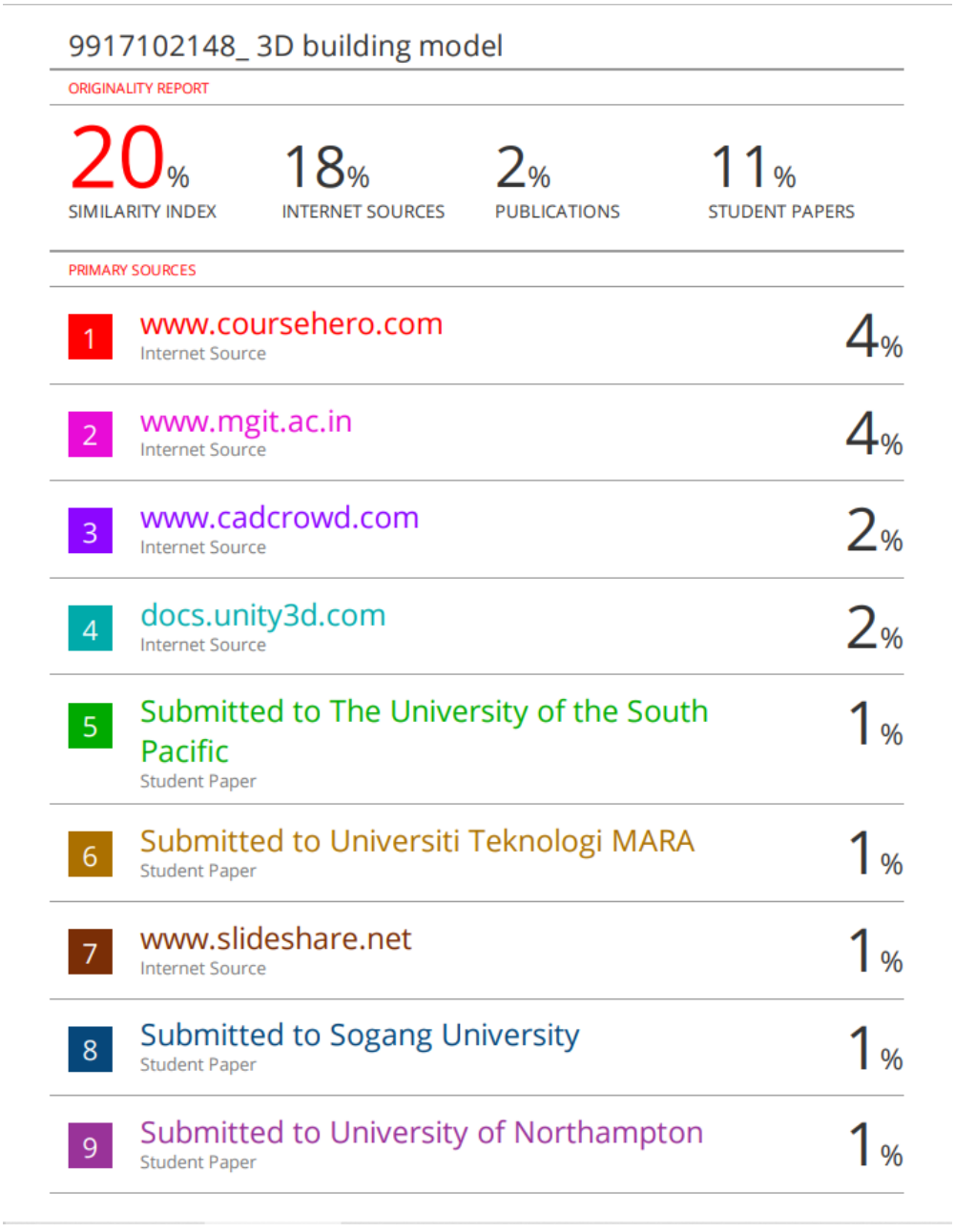
Code

```
using UnityEngine;
public class PlayerMovement : MonoBehaviour
{
public static PlayerMovement Instance { get; private set; }
[SerializeField] float _speed = 5f;
[SerializeField] LayerMask _aimLayerMask;
Animator _animator;
void Awake()
{
Instance = this;
_animator = GetComponent <Animator>();
}
void Update()
{
AimTowardMouse();
// Reading the Input
float horizontal = Input.GetAxis("Horizontal");
float vertical = Input.GetAxis("Vertical");
Vector3 movement = new Vector3(horizontal, 0f, vertical);
// Moving
if (movement.magnitude > 0)
{
movement.Normalize();
movement *= _speed * Time.deltaTime;
transform.Translate(movement, Space.World);
float targetAngle = Mathf.Atan2(movement.x,movement.z) * Mathf.Rad2Deg;
transform.rotation = Quaternion.Euler(0f,targetAngle,0f);
}
// Animating
float velocityZ = Vector3.Dot(movement.normalized, transform.forward);
float velocityX = Vector3.Dot(movement.normalized, transform.right);
_animator.SetFloat("VelocityZ", velocityZ, 0.1f, Time.deltaTime);
_animator.SetFloat("VelocityX", velocityX, 0.1f, Time.deltaTime);
}
void AimTowardMouse()
{
Ray ray=Camera.main.ScreenPointToRay(Input.mousePosition);
if(Physics.Raycast(ray,out var hitInfo,Mathf.Infinity,_aimLayerMask))
```

```
{
var _direction=hitInfo.point-transform.position;
_direction.y=0f;
_direction.Normalize();
transform.forward=_direction;
}
}
}
```

Appendix B

Plagiarism Certificate



9917102148_ 3D building model

ORIGINALITY REPORT

20%	18%	2%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	4%
2	www.mgit.ac.in Internet Source	4%
3	www.cadcrowd.com Internet Source	2%
4	docs.unity3d.com Internet Source	2%
5	Submitted to The University of the South Pacific Student Paper	1%
6	Submitted to Universiti Teknologi MARA Student Paper	1%
7	www.slideshare.net Internet Source	1%
8	Submitted to Sogang University Student Paper	1%
9	Submitted to University of Northampton Student Paper	1%

10	bradley.csail.mit.edu Internet Source	1 %
11	Submitted to Chester College of Higher Education Student Paper	<1 %
12	gitlab.ecs.vuw.ac.nz Internet Source	<1 %
13	id.scribd.com Internet Source	<1 %
14	I. Buyuksalih, S. Bayburt, G. Buyuksalih, A. P. Baskaraca, H. Karim, A. A. Rahman. "3D MODELLING AND VISUALIZATION BASED ON THE UNITY GAME ENGINE – ADVANTAGES AND CHALLENGES", ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2017 Publication	<1 %
15	Submitted to Southampton Solent University Student Paper	<1 %

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		