Final Year Project REPORT

Project Title

UNIVERSITY GUIDE

Submitted by

MUHAMMAD HAMZA 13178

ASGHAR ALI KHAN 12954

TABRAIZ SHAH 13078

MUHAMMAD MOIZ 13162

Supervisor

Mr. FAHAD NAJEEB

Coordinator

DR. ARIJ MEHMOOD HUSSAIN

Table of Contents

3.1 OVERVIEW

The University Guide is a revolutionary mobile app crafted to simplify the experience of navigating your university campus. Its standout feature is the 3D Map, which provides an interactive and visually engaging way to find your classes and other essential locations on campus. This advanced mapping tool ensures you can easily locate any place you need to visit, making your campus experience smoother and more efficient.

Adding to its convenience, the app includes a friendly, animated character that serves as your personal guide. This character offers step-by-step directions, making it feel like you have a knowledgeable companion walking with you. Whether you're a new student trying to find your way around or a visitor exploring the campus, this feature ensures you never get lost.

One of the most innovative aspects of the University Guide is its virtual tour functionality. This feature allows you to explore the entire university from the comfort of your room. Whether you're preparing for your first visit or simply want to familiarize yourself with the campus layout, the virtual tour provides a comprehensive overview, helping you feel more confident and prepared.

Overall, the University Guide app combines cutting-edge technology with user-friendly design to create an indispensable tool for anyone navigating a university campus.

3.2 LIST OF REQUIREMENTS

FUNCTIONAL REQUIREMENT

Campus Navigation: Interactive 3D maps providing navigation within the university premises. Location-based services to guide users to specific buildings, classrooms, offices, etc.

Explore Access: Access to detailed information about campus facilities, such as libraries, cafeterias, labs, etc.

Information Access: A general information about the university will be provided on reception.

Interactive 3D Map: The app should feature an interactive 3D map of the university campus, allowing users to explore and navigate visually.

NON-FUNCTIONAL REQUIREMENT

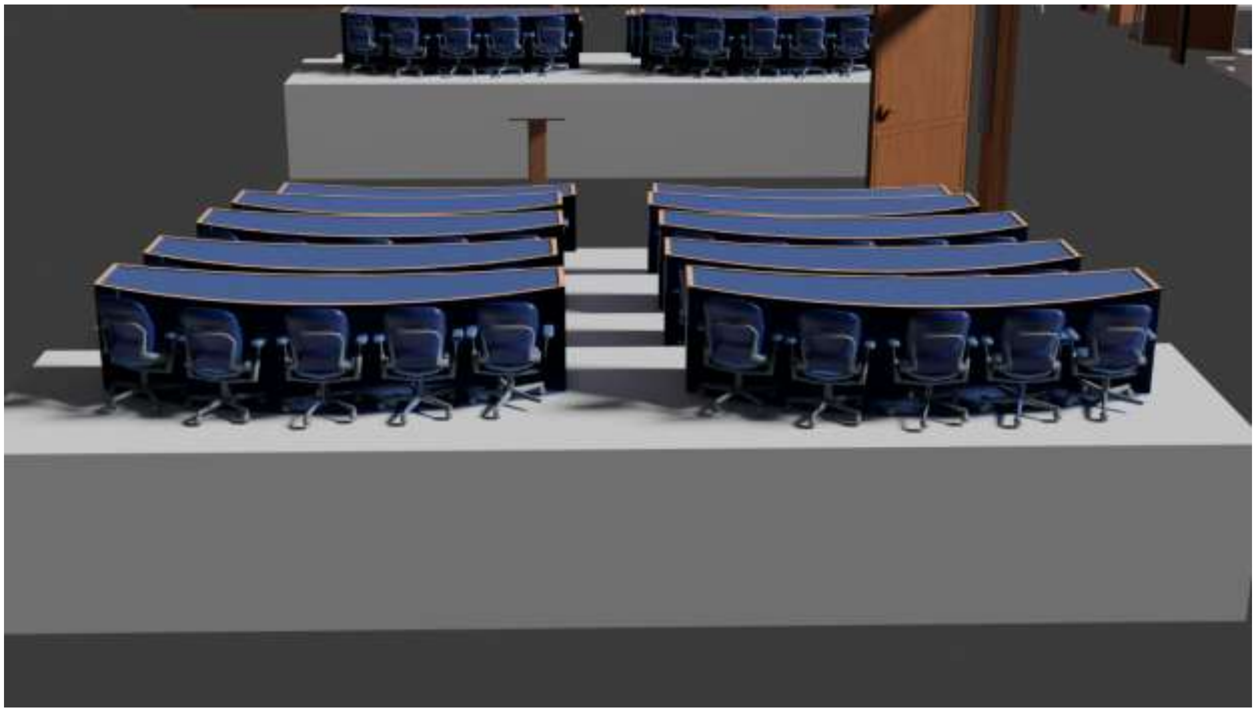
Performance: Smooth navigation within the 3D map.

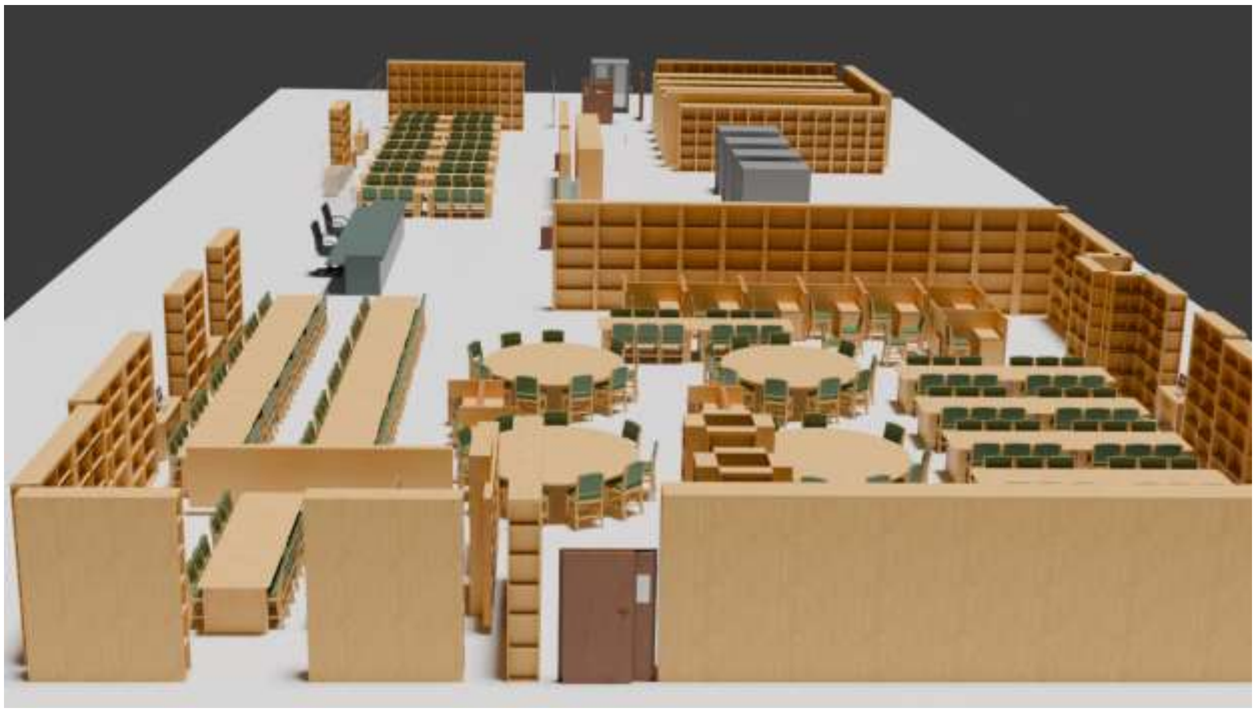
Usability: Easy user interface and user experience (UI/UX) for easy navigation.

Clarity: Visual elements should be clear and providing a pleasant user experience.

Scalability: We will design this app in such a way that everyone can use it easily.

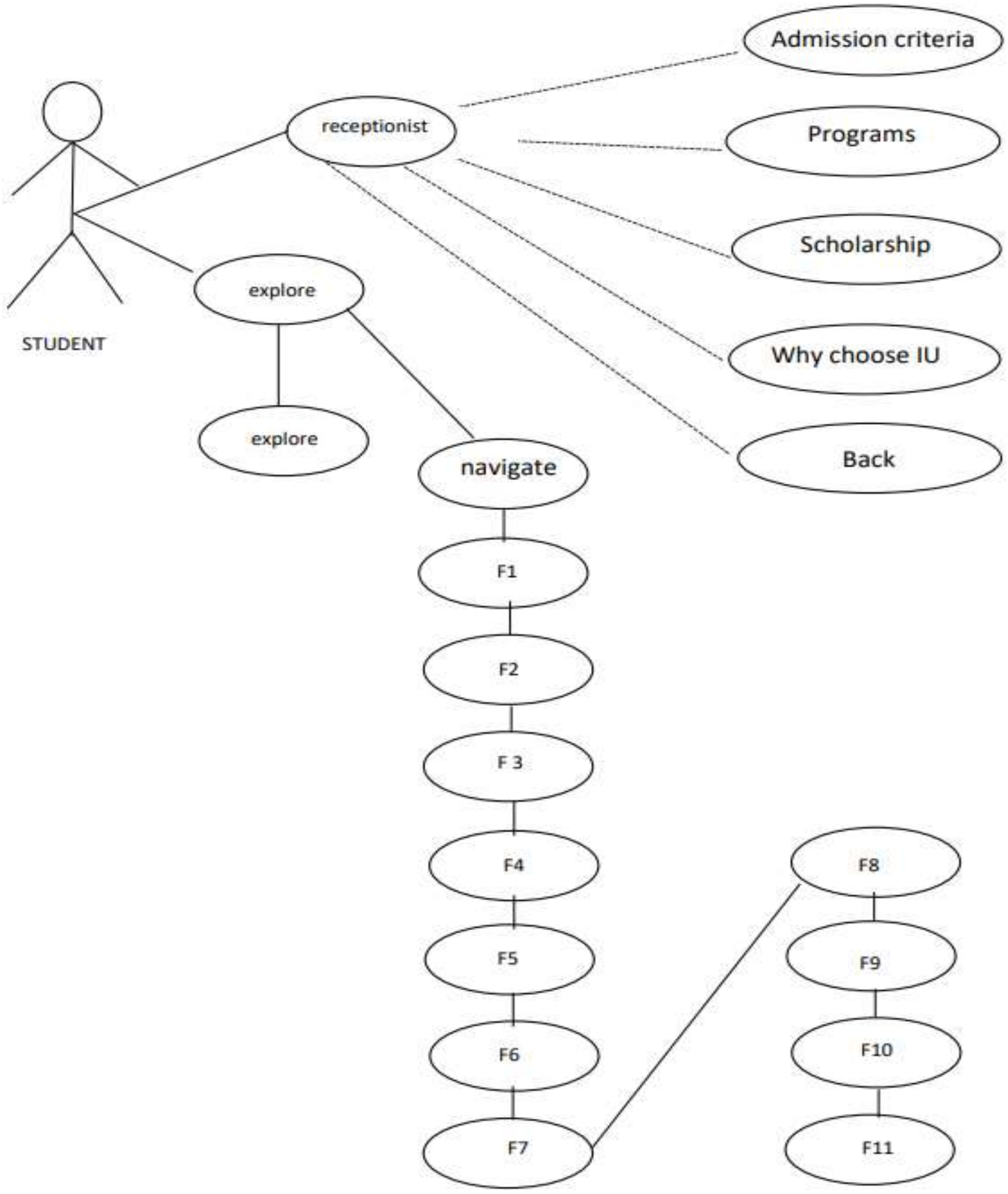
3.3 SCREENSHOTS OF PROJECT





3.4 USE CASE AND DFD DIAGRAM

USE CASE:



USE CASE NARRATIVE:

Entities and Actors:

1. Student (Actor):

Represented by a stick figure. This is the main user interacting with the system.

2. Receptionist (Entity/Process):

Represented by an oval, this is the primary interface that the student interacts with for various tasks.

Processes and Interactions:

1. From Student to Receptionist:

There is a direct line connecting the Student to the Receptionist, indicating that the Student initiates interaction with the Receptionist.

2. Receptionist Options:

The Receptionist provides several options or actions related to university information:

Admission Criteria: There is a dashed arrow from the Receptionist leading to an oval labeled "Admission Criteria," suggesting that the receptionist provides information on admission requirements.

Program: Another dashed arrow points to an oval labeled "Program," indicating that the receptionist can provide details about the university's programs.

Scholarship: Similar to the other options, a dashed arrow points to an oval labeled "Scholarship," representing the availability of scholarship information.

Why Choose IU: A dashed arrow points to an oval labeled "Why Choose IU," suggesting the receptionist can explain the advantages or unique aspects of the institution.

Back: There's a line or connection to a "Back" option, likely allowing the user to return to a previous menu or state.

3. **Exploration and Navigation:**

The Student is linked to an option labeled "Explore," which further connects to a process labeled "Navigate."

- Under "Navigate," there are options for navigating different "Floors" of the university:

- Floor 1

- Floor 2

- Floor 3

- Floor 4

- Floor 5

- Floor 6

- Floor 7

- Floor 8

- Floor 9

-Floor 10

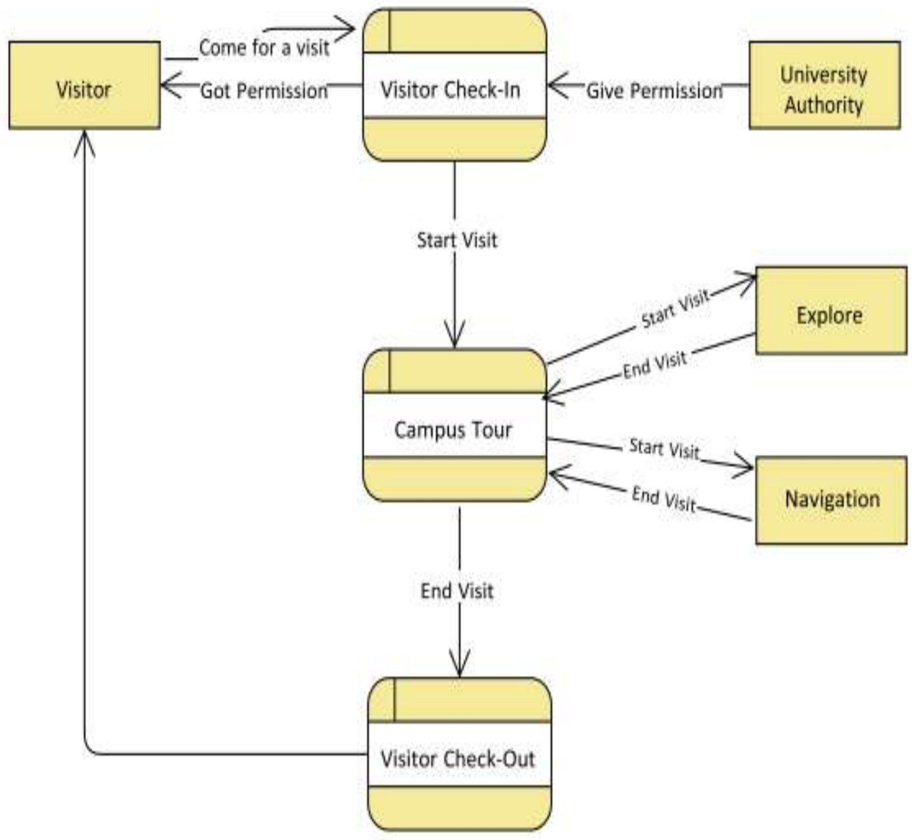
-Floor 11

Summary:

This use case diagram outlines a scenario where a student interacts with a receptionist (possibly a digital entity) to access information about admission criteria, programs, scholarships, and reasons to choose the university. Additionally, the student has the option to explore the campus virtually or physically by navigating through different floors.

The use case diagram is intended for outlining user interactions within a university setting, focusing on providing relevant information and facilitating campus navigation.

<u>DFD Diagram:</u>


Visitor
Come for a visit
Got Permission
Visitor Check-In
Give Permission
University Authority
Start Visit
Start Visit
Explore
End Visit
Campus Tour
Start Visit
Navigation
End Visit
End Visit
Visitor Check-Out

DFD Narrative:

1) Visitor:

This entity initiates the process. Visitors check in to start their visit.

2) University Authority:

They provide permission to the visitors. Once permission is granted, the visitor can proceed with their activities.

3) Processes:

Visitor Check-In:

This process is where visitors register their presence at the university.

Campus Tour:

After checking in, visitors can start a tour.

Navigation:

A guided tour using Navigation.

Free Navigation:

Visitors can choose to explore the campus on their own.

Visitor Check-Out:

The final step, where visitors check out, marking the end of their visit.

4) Data Flows:

"Come for a visit" represents the initial interaction between the visitor and the university.

"Give Permission" and "Got Permission" show the communication between the University Authority and the visitor regarding access.

"Start Visit" and "End Visit" indicate the start and end points of the visitor's journey.

This DFD provides a high-level overview of how visitors are managed during their visit to the university, including check-in, navigation options, and check-out.

3.5 User Guide: How to Download and Open Blender Files from GitHub

1. Install Blender:

- Download Blender from the official website: [Blender Download] (https://www.blender.org/download/).

- Install Blender on your PC following the on-screen instructions.

2. Download Blender Files from GitHub:

- Visit the GitHub repository: [3D University Floor Plans] (https://github.com/asghar123-tech/3Dmodel).

- Click on the file for the floor you want to download.

- Click on the "Download" button, then select "Raw" to download the file directly to your computer.

3. Open the Blender File:

- Launch Blender on your PC.

- Go to `File > Open` and navigate to the downloaded file.

- Select the file and click `Open` to view and edit the 3D model.

User Guide: Step-by-Step Guide to Adding Player Movement in Unity

1. Merge All Floors in Blender:

1. Open Blender and load each of the 11 floors.

2. Merge the Floors:

   - Select the first floor and use `File > Append` to import the other floors.

   - Position each floor correctly to create a complete model of the building.

3. Save the Merged File:

   - Save the merged floors as a single .blend file.

2. Import the Merged Blender File into Unity:

1. Create a New Unity Project:

   - Open Unity and create a new 3D project.

2. Import the Blender Model:

   - Drag and drop the merged .blend file into the Unity project's `Assets` folder.

   - Unity will automatically convert the .blend file into a Unity-compatible 3D model.

3. Set Up the Scene:

   - Drag the imported model into the scene to position it as needed.

3. Import and Set Up the Player:

1. Import a Player Asset:

   - Use a pre-built player controller from the Unity Asset Store or create your own.

   - Drag the player prefab into your scene and position it at the starting point.

2. Add the Movement Script:

   - Download the movement script from [GitHub repository] (https://github.com/asghar123-tech/3Dmodel).

   - Drag and drop the script into the `Assets` folder in Unity.

   - Select the player object in the scene and attach the script to the player by dragging it onto the player in the `Inspector` panel.

4. Configure the Script in Unity:

1. Set Up the Script Parameters:

   - If the script has customizable parameters (like speed, jump height, etc.), configure them in the `Inspector`.

2. Test the Movement:

   - Press `Play` in Unity to test the player movement. Adjust the script or player settings if needed.

5. Save and Build the Project:

1. Save Your Scene:

   - Make sure to save the Unity scene.

2. Build the Project:

   - Go to `File > Build Settings`, select your platform, and click `Build` to create the executable file.

Script Placement in Unity

- Where to Place the Script: The movement script should be attached to the player object. In Unity:

  1. Select the player in the `Hierarchy`.

  2. In the `Inspector`, click `Add Component` and either drag the script into the player's component list or search for the script by name.

User Guide: Step-by-Step Guide to Add Player

In Unity, there isn't a built-in player character, but Unity provides several ways to import or create one. Here's how you can get a player character for your university guide project:

Option 1: Using Unity's Standard Assets (Recommended for Beginners)

Unity used to provide a Standard Assets package that included a pre-built First-Person Controller (FPC) and Third-Person Controller (TPC). However, this package is no longer included by default and is deprecated, but you can still find it online or use similar assets from the Unity Asset Store.

Step-by-Step Guide for Using a Pre-Built Player from Unity Asset Store

1. Open Unity and Your Project:

   - Launch Unity and open the project where your university guide is being developed.

2. Import a Player from Unity Asset Store:

- Go to `Window > Asset Store` or visit the [Unity Asset Store](https://assetstore.unity.com/) in your browser.

- Search for "First Person Controller" or "Third Person Controller."

- Choose a free or paid asset that suits your project (e.g., "Starter Assets - First Person Character Controller").

- Click `Add to My Assets` and then `Open in Unity`.

- In Unity, click `Import` to bring the selected player asset into your project.

3. Set Up the Player:

- After importing, navigate to the `Assets` folder where the player prefab is stored.

- Drag and drop the player prefab into your scene's `Hierarchy`, placing it where you want the player to start.

- Adjust the player's position and rotation as necessary to fit the starting point of your scene.

4. Add a Player Movement Script (If Needed):

- If your player asset doesn't already have a movement script, attach your custom movement script by selecting the player in the `Hierarchy` and dragging the script into the `Inspector`.

5. Test the Player:

- Press the `Play` button in Unity to test the player's movement within your scene.

- If adjustments are needed (e.g., speed, jump height), modify the script parameters or player settings.

Option 2: Using a Custom Player Character

If you prefer to create or use a custom player character (e.g., a unique 3D model):

Step-by-Step Guide for Importing a Custom Player Character

1. Obtain a 3D Model:

- You can either design a character in Blender or download a model from online resources like [Mixamo] (https://www.mixamo.com/) or the Unity Asset Store.

2. Import the 3D Model into Unity:

- Place the 3D model file (e.g., .fbx, .obj) into the `Assets` folder in Unity.
- Unity will automatically import and convert the model for use in your scene.

3. Add a Character Controller:

- Create an empty Game Object in Unity (`Game Object > Create Empty`) and name it "Player."
- Drag your 3D model into the "Player" Game Object in the `Hierarchy` to make it a child object.
- With the "Player" Game Object selected, click `Add Component` in the `Inspector` and add a `Character Controller`.

4. Add a Movement Script:

- If you have a custom movement script, attach it to the "Player" Game Object by dragging it into the `Inspector`.
- Configure any script parameters (like speed) as needed.

5. Set Up Animations (Optional):

- If your character has animations (e.g., walking, running), create an `Animator Controller` in Unity and assign the animations.
- Attach the Animator component to the "Player" Game Object and link it with your Animator Controller.

6. Test the Character:

- Press `Play` to test the character’s movement and interaction within the scene.
- Adjust settings as needed for a smooth experience.

Conclusion

- Using Pre-Built Assets: If you're new to Unity, starting with pre-built player assets from the Unity Asset Store is easier and faster.

- Using Custom Characters: For more customization, you can import your own 3D model and set it up manually.

User Guide: Step-by-Step Guide to Adding Navigation in Unity

Adding navigation in Unity, such as allowing a player to move from one place to another with guided paths or waypoints, involves using Unity's NavMesh system. This system allows you to create a navigable environment where a character or object can move along predefined paths.

Here's a step-by-step guide to setting up navigation in your Unity project:

1. Set Up Your Scene

1. Prepare the Environment:

   - Make sure your scene contains all the necessary models, such as floors, walls, and obstacles.

   - Ensure all models have proper colliders, especially if you imported them from Blender.

2. Create a Navigation Area:

   - Select the floor and any other surfaces you want to be walkable.

   - Go to the `Inspector` and ensure that the "Navigation Static" checkbox is enabled. This will mark the objects as static for navigation purposes.

2. Bake the NavMesh

1. Open the Navigation Window:

   - Go to `Window > Navigation` to open the Navigation window.

2. Bake the NavMesh:

   - In the Navigation window, switch to the `Bake` tab.

   - Adjust the settings under `Agent Radius`, `Agent Height`, `Max Slope`, and `Step Height` according to the dimensions of your player character.

- Click the `Bake` button. Unity will generate a NavMesh on all marked surfaces, highlighting the walkable areas in blue.

3. Add a NavMesh Agent to Your Player

1. Select the Player:

- In the `Hierarchy`, select your player character or object that will be moving along the NavMesh.

2. Add a NavMesh Agent Component:

- In the `Inspector`, click `Add Component` and search for `NavMesh Agent`.

- Add the `NavMesh Agent` component. This will enable your player to navigate along the baked NavMesh.

- Configure the NavMesh Agent settings, such as speed, acceleration, and stopping distance, to suit your game.

4. Create Waypoints or Targets

1. Create Target Points:

- In the `Hierarchy`, right-click and create empty Game Objects (`Create Empty`). These will serve as waypoints or destination points.

- Position these waypoints in your scene where you want the player to navigate.

- Name them appropriately, such as "Waypoint1," "Waypoint2," etc.

2. Optional - Visualize Waypoints:

- If you want to see the waypoints in the scene view, you can add a simple visual, such as a small 3D object (like a sphere) or a gizmo.

3. Attach the Script to the Player:

- Drag the script onto your player character in the `Inspector`.

- Assign the waypoints you created in the scene to the `waypoints` array in the script component.

6. Test the Navigation

1. Test in Play Mode:

- Press `Play` in Unity to test the navigation system.

- Your player should move towards the waypoints sequentially or according to the script logic.

2. Adjust and Fine-Tune:

- If the player isn't navigating as expected, check the NavMesh, the NavMesh Agent settings, and the script logic.

- You can tweak the parameters in the NavMesh Agent and the baking settings to improve navigation accuracy.

Additional Tips

- Dynamic Obstacles: If you have moving obstacles, you can add a `NavMeshObstacle` component to them, which will dynamically adjust the NavMesh to account for these objects.

- Complex Path finding: For more advanced path finding, consider using A Path finding Project, which offers more control and optimization options.

GITHUB URL:

https://github.com/asghar123-tech/3Dmodel