

Level Protocol Layout Document (Unity – Terrain-Based Workflow)

Version: 1.0

Purpose:

To establish a consistent, repeatable pipeline for designing and implementing levels using Unity's Terrain system.

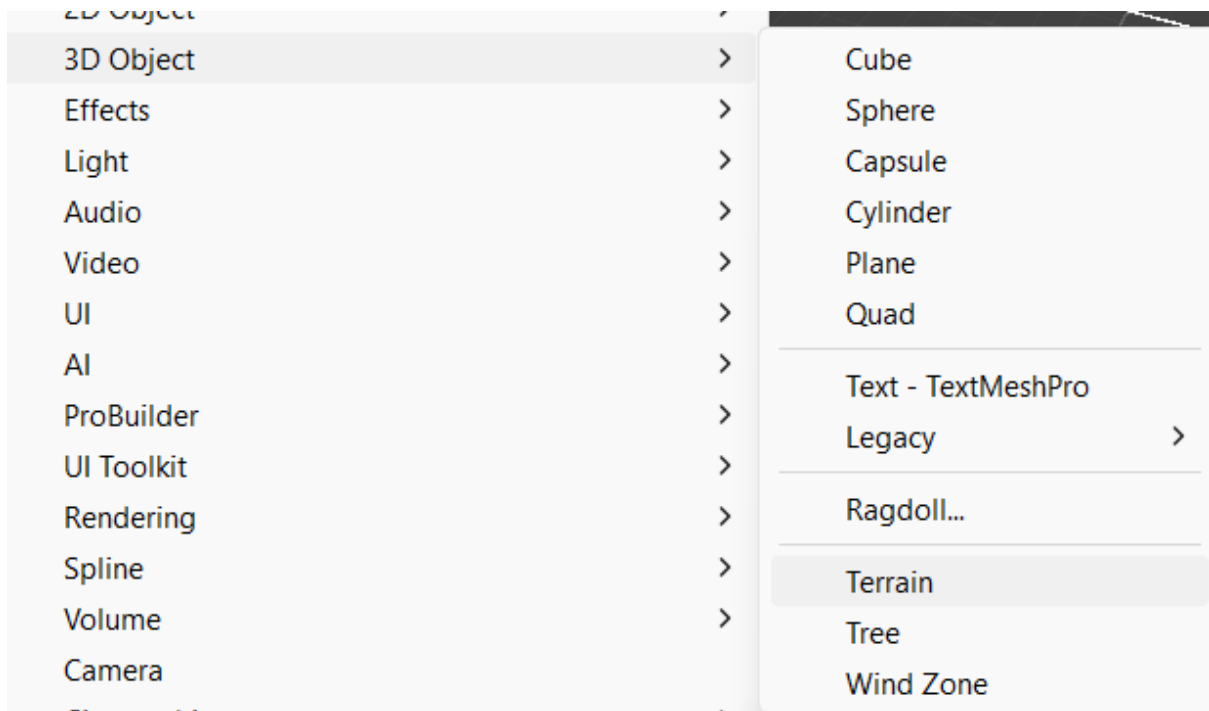
1. Terrain Layout & Blockout (Designer)

1.1 Objective

Create the foundational shape of the level using Unity's Terrain tools. Focus on gameplay flow, scale, and navigation paths—not detail.

1.2 Steps

- Create a new Terrain object in the scene.



- Level Mesh resolution.

- Small Island specs:

Mesh Resolution (On Terrain Data)	
Terrain Width	50
Terrain Length	50
Terrain Height	600
Detail Resolution Per Patch	32
Detail Resolution	1024

- Big Island specs

Mesh Resolution (On Terrain Data)	
Terrain Width	100
Terrain Length	100
Terrain Height	600
Detail Resolution Per Patch	32
Detail Resolution	1024

- Sculpt primary landforms:
 - Main paths
 - Playable plateaus
- Add major landmark shapes (“macro silhouette”):
 - Player Spawner
 - Rivers/lakes
 - Large hills and slope

1.3 Key Considerations

- Check for camera angle before creating the terrain
- Test scale frequently by entering Play Mode.

2. Placing Functionality & Gameplay Systems (Designer)

2.1 Objective

Add all major gameplay components before artistic polish.

2.2 Steps

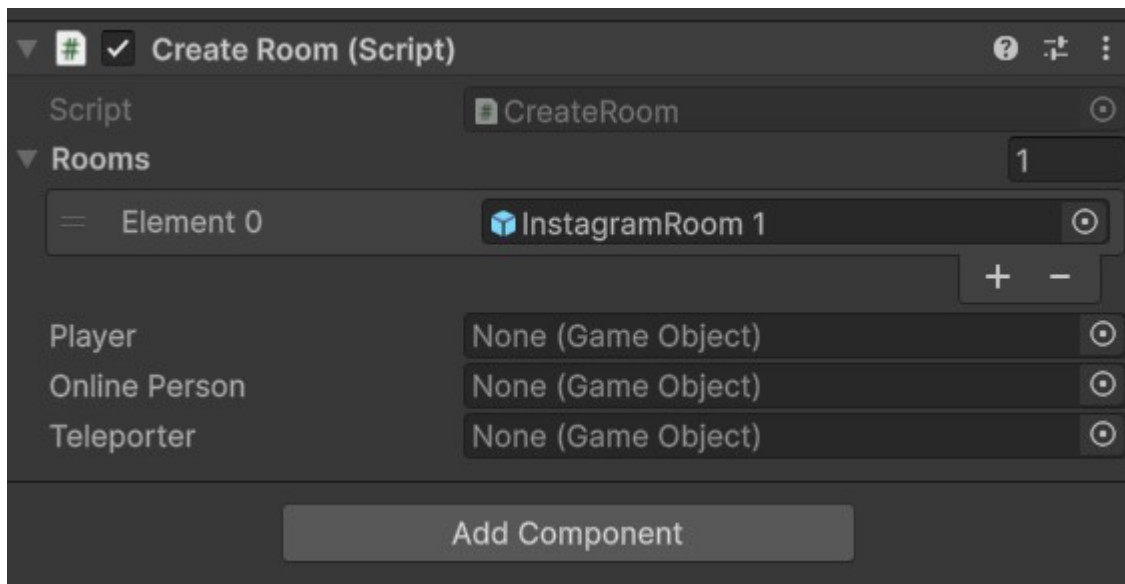
- Place **spawn points** (player, enemies, NPCs).
- Place **border colliders** and define **zones** (Unwalkable zone and Playable zone).



- Confirm all gameplay elements are reachable using current terrain blockout.

2.3 Key Considerations

- Check the border regularly with the enemies if there are any bugs then report it to the engineers
- Test core gameplay loops early.



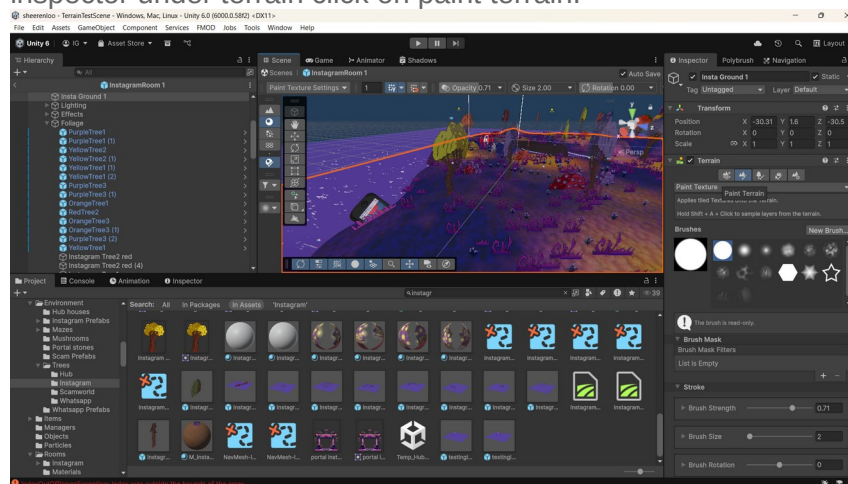
3. Terrain Painting & Detail Pass (Artist)

3.1 Objective

Refine the terrain visually and physically by painting textures, trees, and foliage.

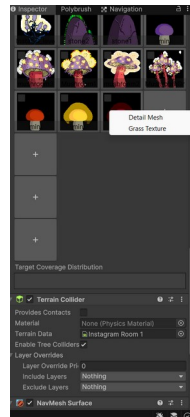
3.2 Steps

- Apply terrain texture layers like, Base soil/grass painted on, Rock textures for cliffs or Sand/dirt paths. Here's a small guide on how to add new textures for the terrain:
 - First select the terrain you want to paint ground textures on. Then in the inspector under terrain click on paint terrain.

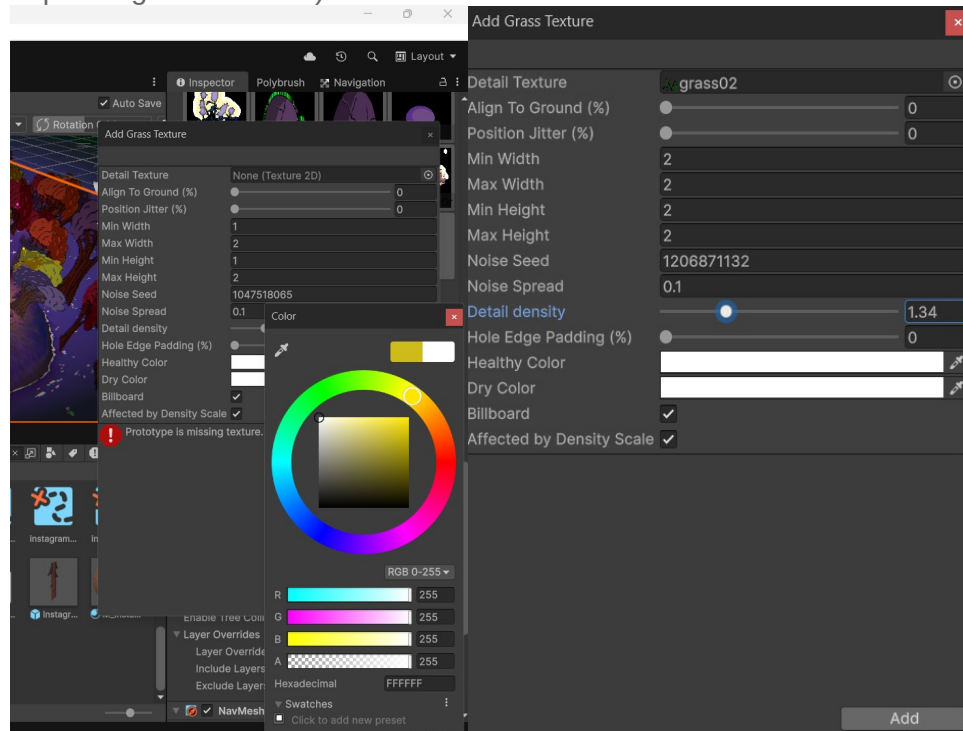


- Then scroll down to layers, click on add layer and select the texture you want to add

convenient.

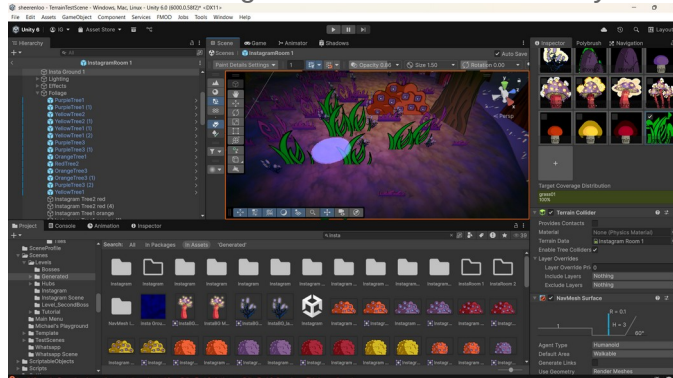


- Now with this menu open add the needed asset to detail texture. Put the healthy color and dry color to white and play a bit with the settings like detail density and min and max height and stuff(all these settings kind of change depending on the asset) after this is done click on add

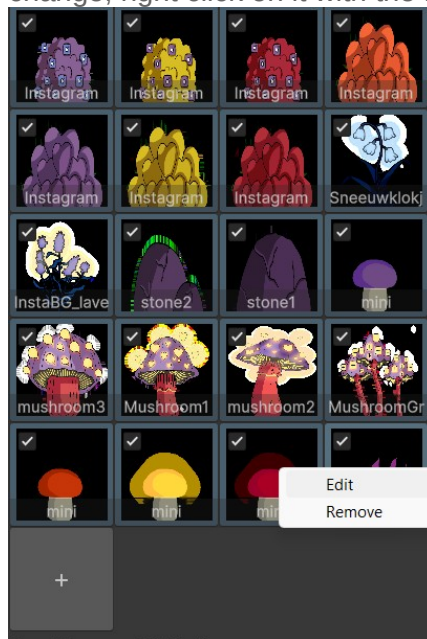


- You can now place the asset(or assets) by selecting the added brushes you need in the inspector with the terrain still selected and then you can just paint

the details on the ground in the scene where you want to place them.



- Sometimes you may still need to change the placement density or max/min height. If this is the case you can do that by going to the asset you want to change, right click on it with the mouse and then edit

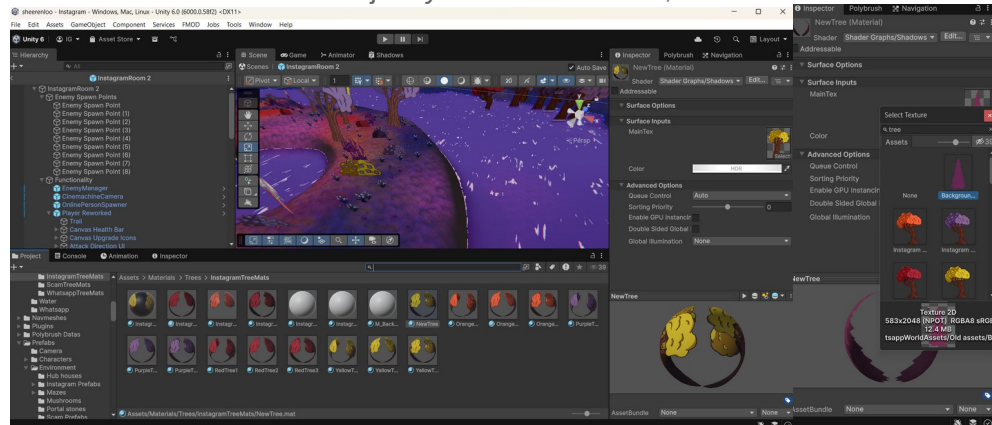


- Then you also have to add trees and bigger objects, but we added them in a different way using prefabs, because that way they become invisible when walking in front of these type of assets. Here's a step by step guide to add these type of assets:

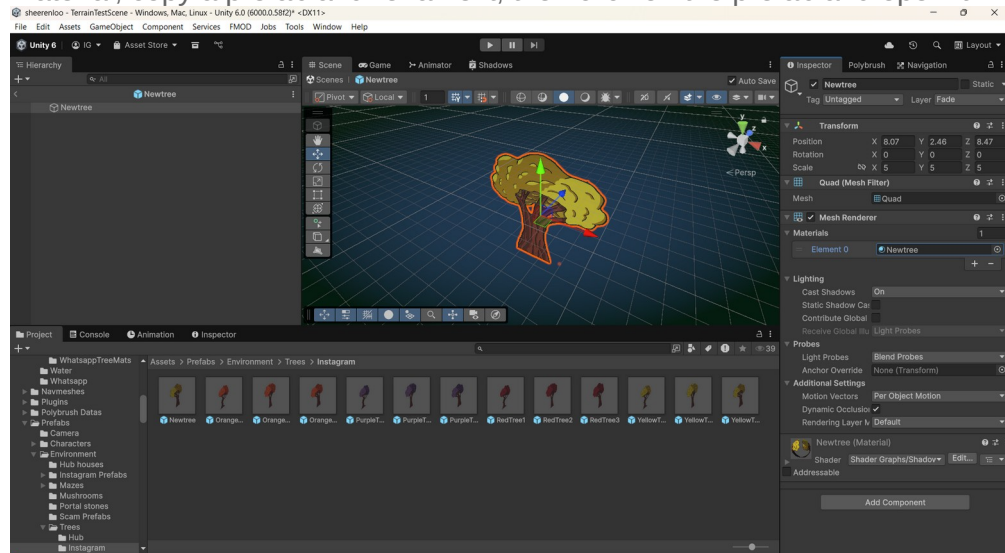
- You first have to go to assets > materials > trees > (correct section depending on the world), then select a material, copy it and rename it.



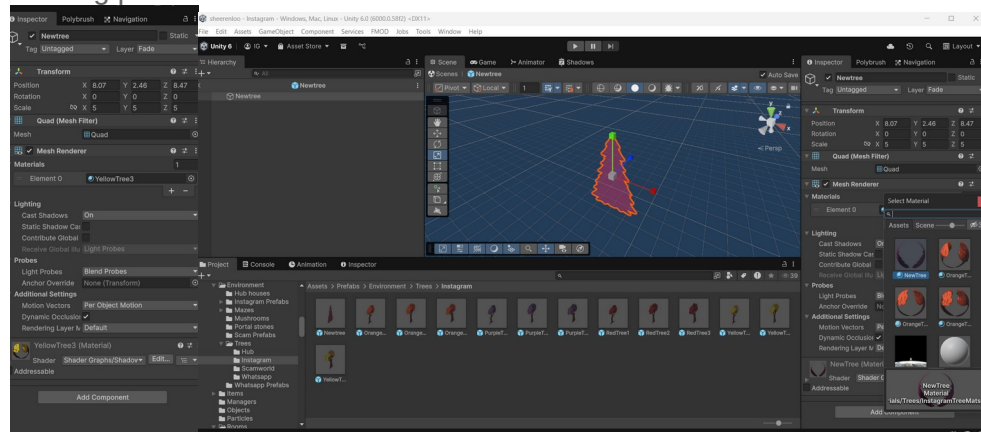
- Then with the new material selected under surface inputs in the inspector you can see mainTex and an asset. In that section click on select and replace that asset with whatever new object you want to create, like a different tree.



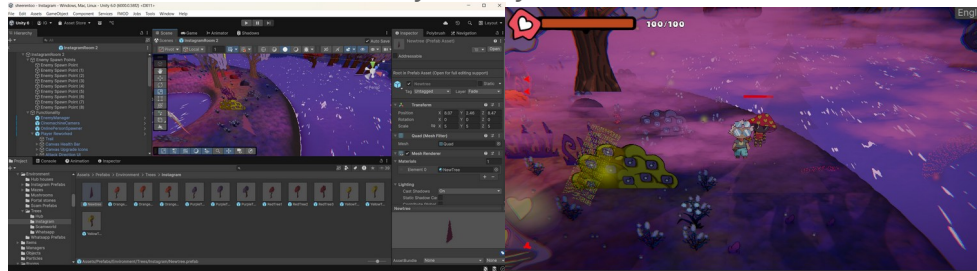
- Then go to assets>prefabs>environment>trees(or other prefab you want to make)>(correct section depending on the world) and just like with the material, copy a prefab and rename it, then click on the prefab and open it.



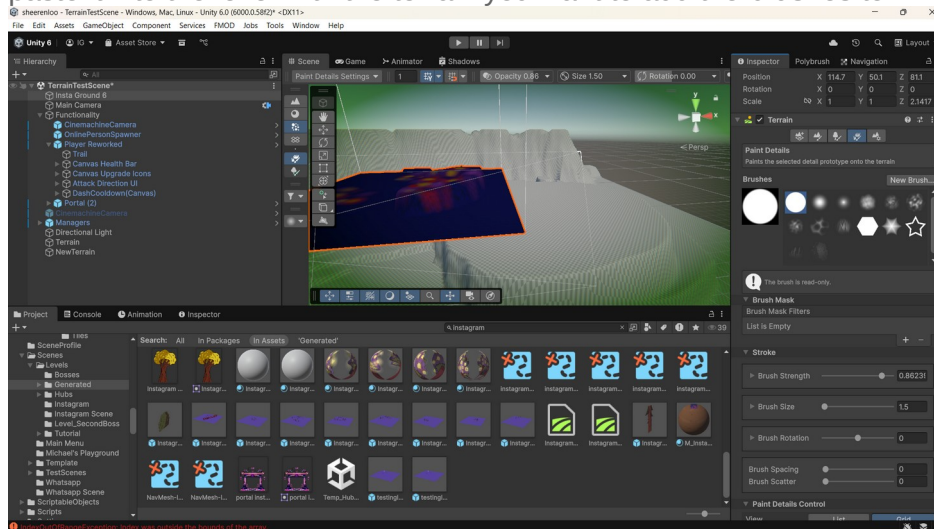
- Then with your new prefab selected look in the inspector and under materials and Element 0 in the mesh renderer click on the dot next to the old material and replace the old material with your new one. This should change the existing prefab.



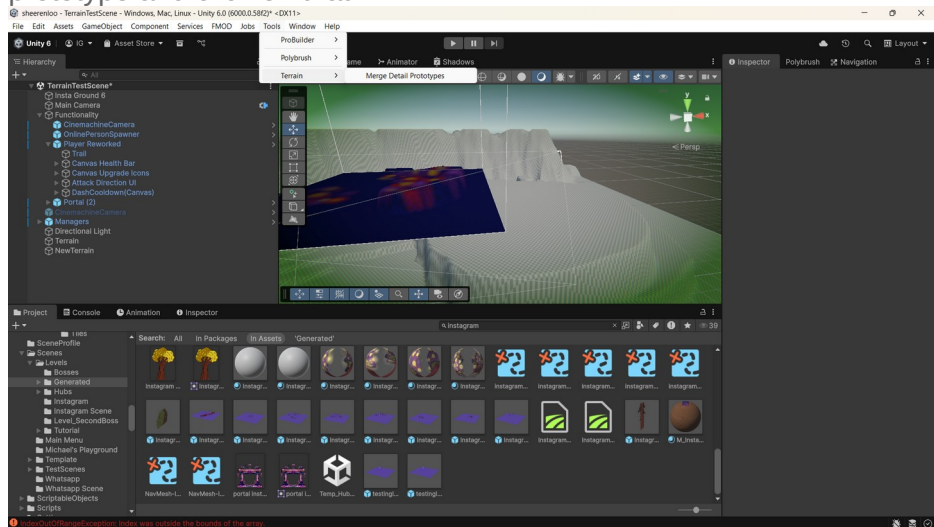
- With that done you can go back to your level and just drag your new prefab into the scene and the invisibility when you walk in front of it should work.



- Sometimes you may want to copy asset brushes from one level to another to not have to add them all again. Here is a small guide to do that:
 - First copy a terrain from a level with all the right brushes already on there and paste it into the level with the terrain you want to add the brushes to.

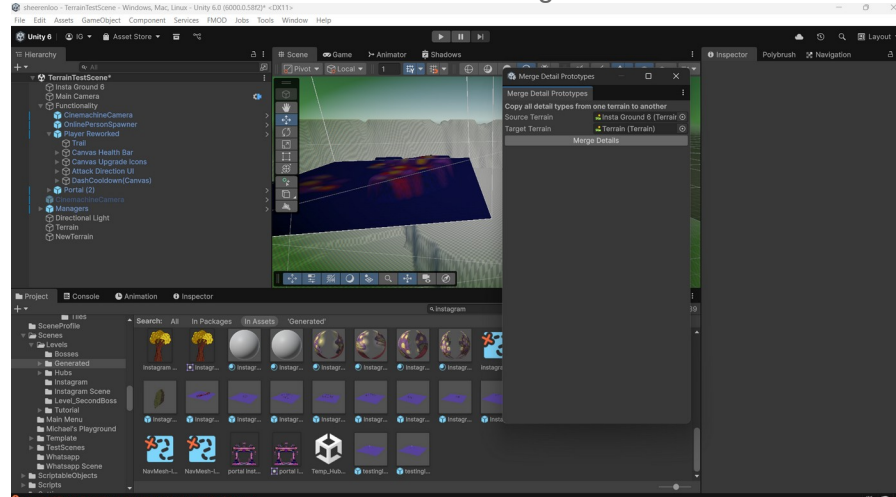


- Then look at the top of your screen and go to tools> Terrain > Merge detail prototype and click on that

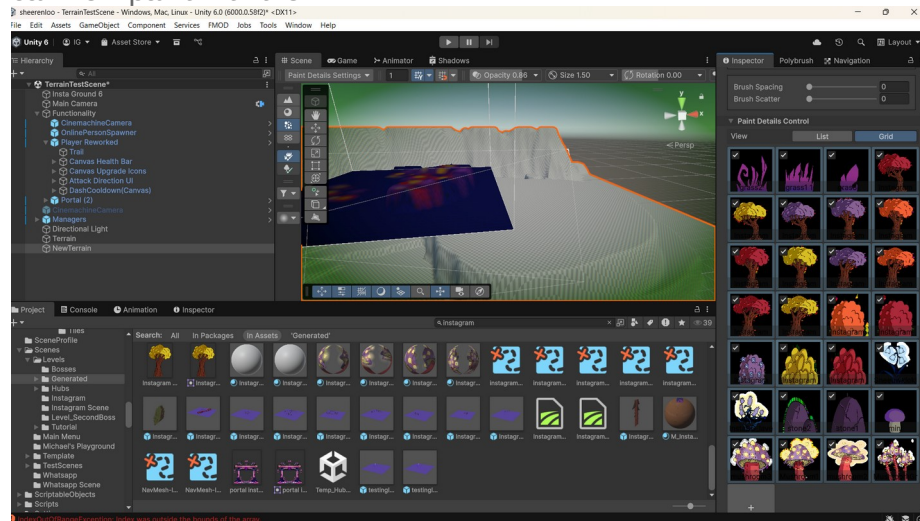


- Then put the terrain from the old level with the brushes already on there in the source terrain and put the terrain you want to add the brushes to in target

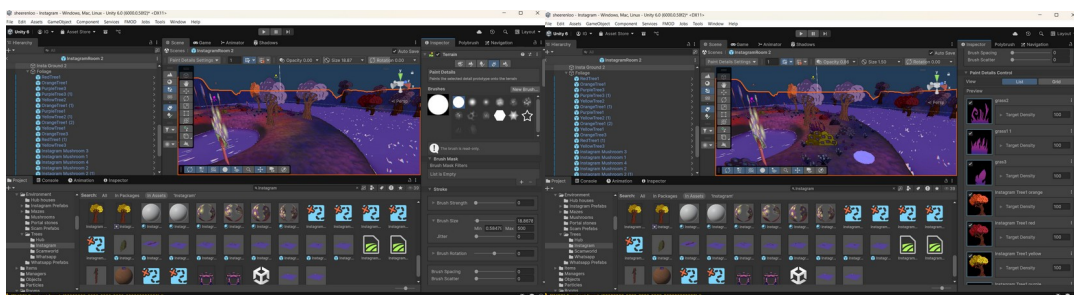
terrain and then click a few times on merge details.



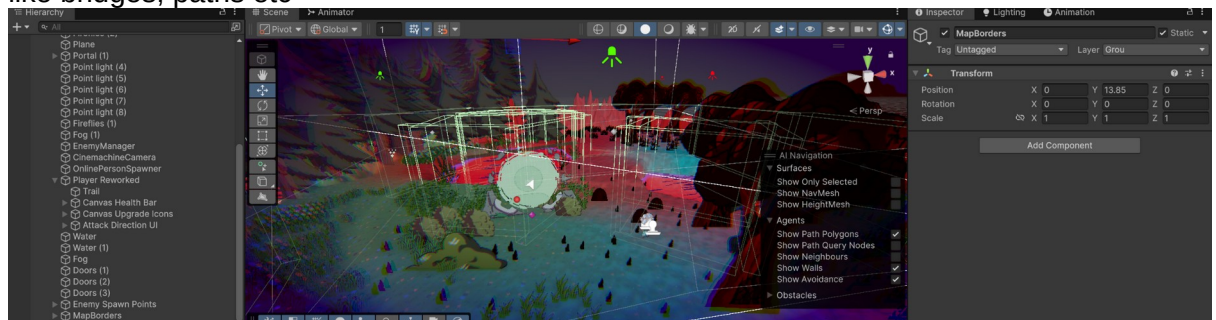
- This should have added the necessary brushes to your new terrain and you can now paint with them.



- Now you have all the brushes and prefabs. first paint the assets into the levels by selecting the terrain itself in the level hierarchy, going to paint details, selecting all the needed brushes there and just paint on the ground in the scene itself and it should automatically appear on the ground. (you can also unpaint details you do not want in a specific location by holding ctrl) and place all the prefabs into the scenes as well in the way you want to place them until the level looks exactly the way you want it to look



- Next make smooth paths by adding walls for npc and player movement and add stuff like bridges, paths etc

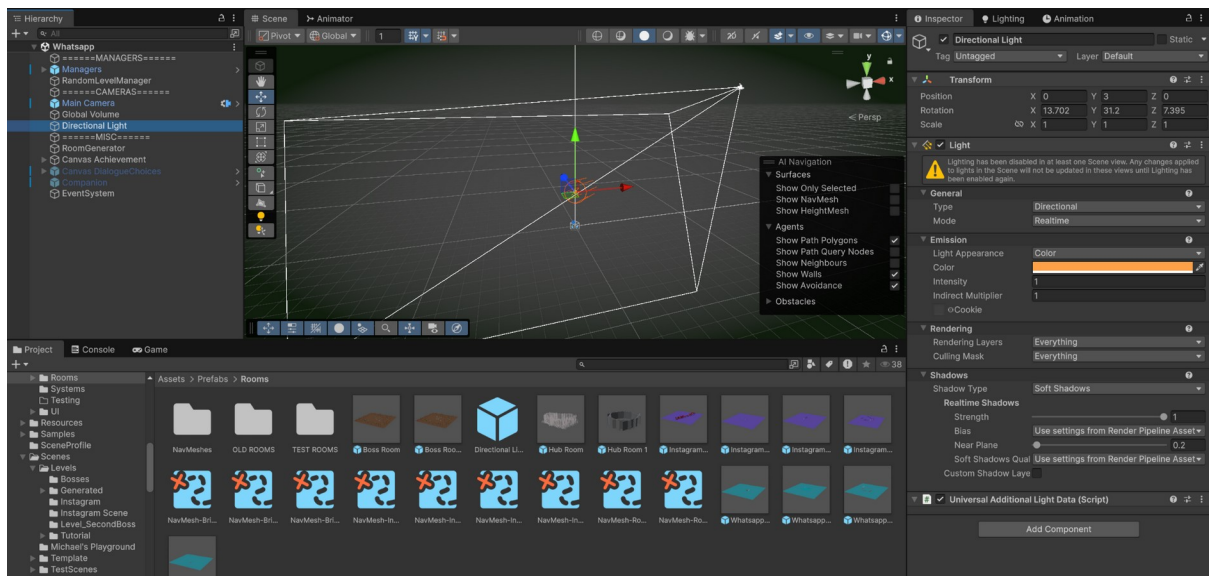


- And don't forget to Improve environmental storytelling using terrain variation.

3.3 Key Considerations

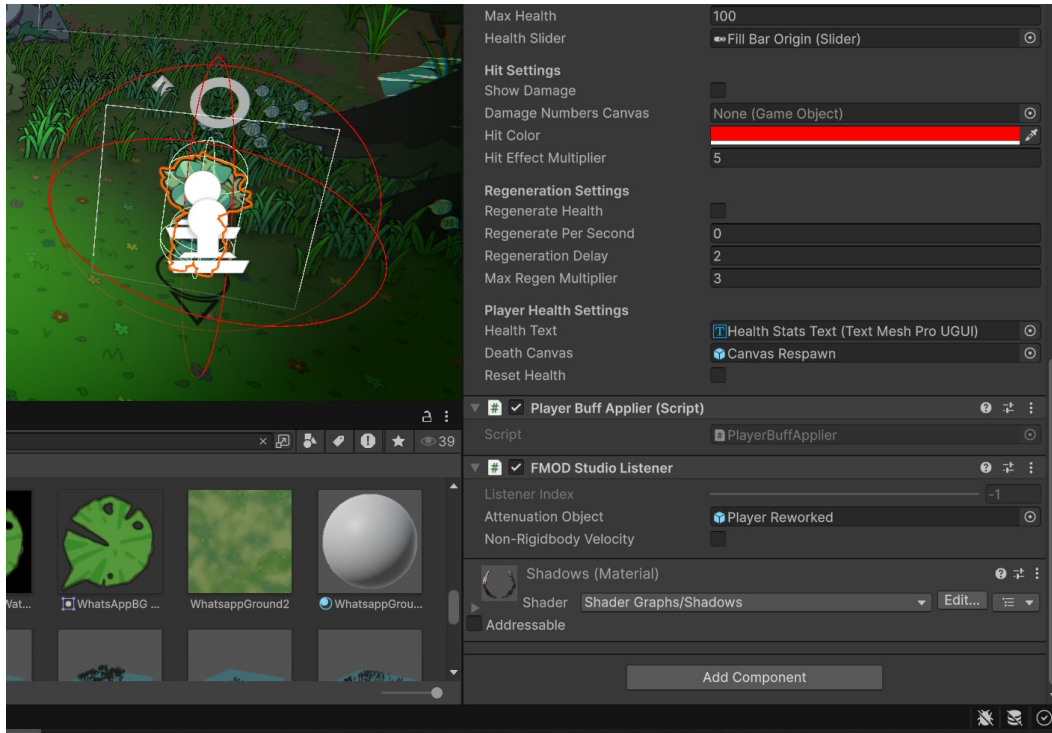
- Maintain performance: limit dense foliage.
- Keep gameplay paths readable using visual language (lighter paths, clearer edges).
- **In case of deadline requirement: Minimum the ground must have texture, tree placement around the map and plane with water shader under the terrain.**

4. Lighting, Atmosphere & Post-Processing (Artist)

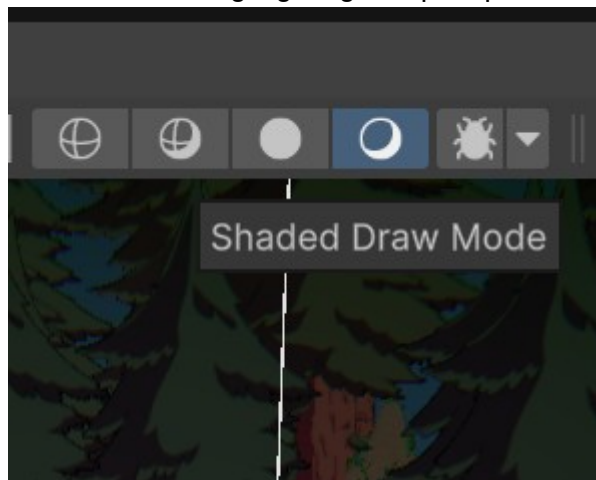


- In order to convey the atmosphere of the levels, you need to use multiple light sources across your level. First of all, the most important is the directional light, acting as the sun of your level. Modify it accordingly, by editing the color, strength and rotation of the sun. **WARNING** - in some cases, the sun will **not** create shadows on your assets as much as you move it. Be sure to add a shadow material on your

desired assets

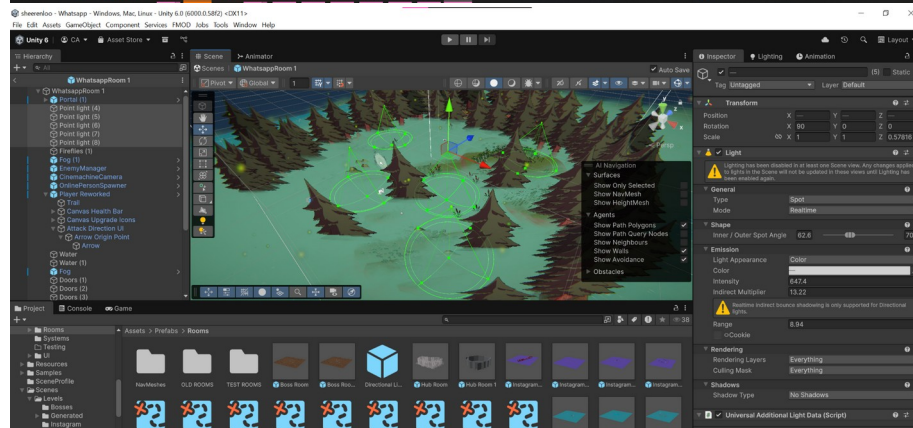
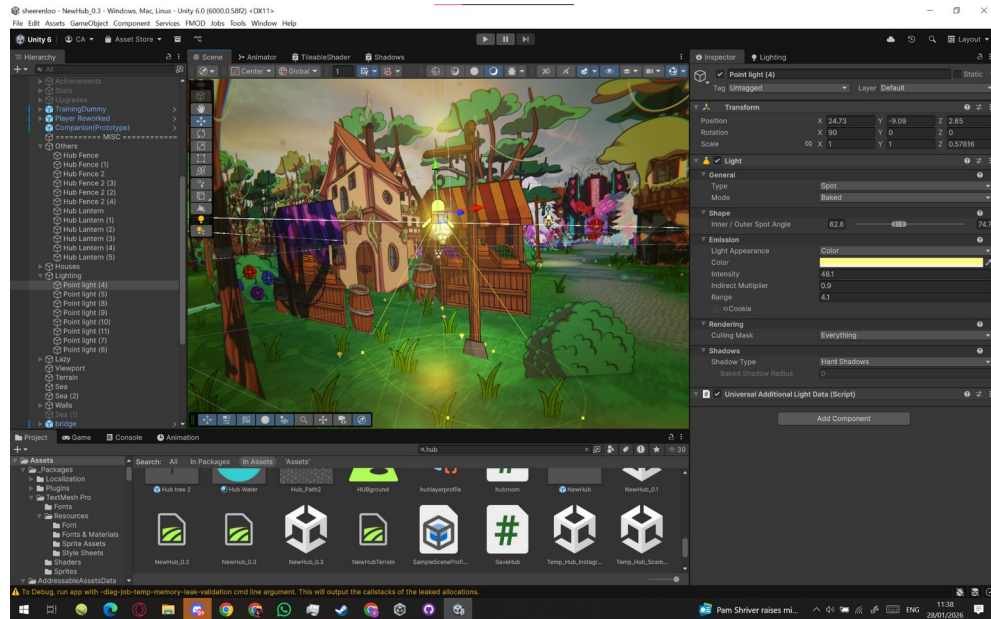


- EXTRA NOTE: **ALWAYS** work in Shaded Draw mode when adding any effect, such as fog, lighting and post processing effects, YOU WILL NOT SEE

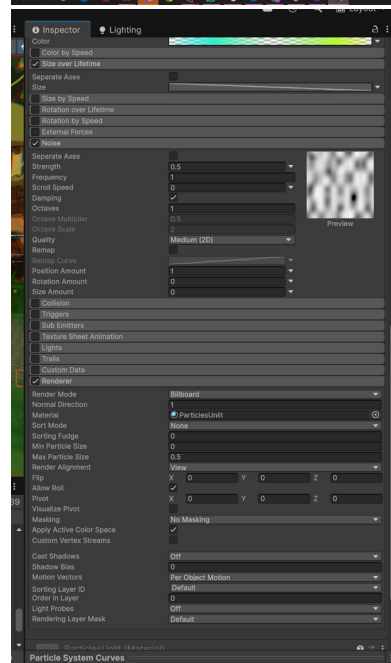
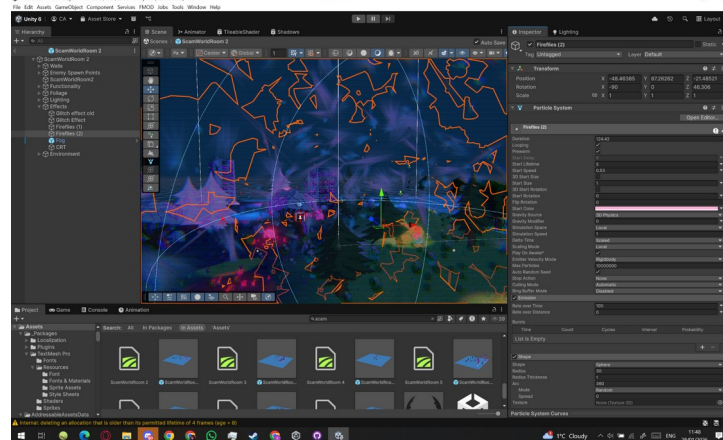
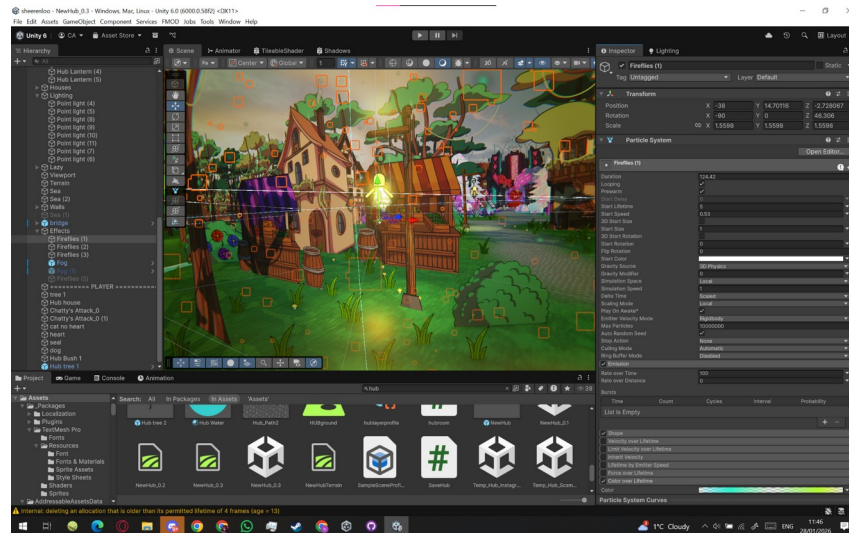


- Spot lights are also very important for your level, especially for foggy or darker environments, use them to enhance the colors of the ground or make the level feel

more magical also. They can also have functional uses, such as for lanterns

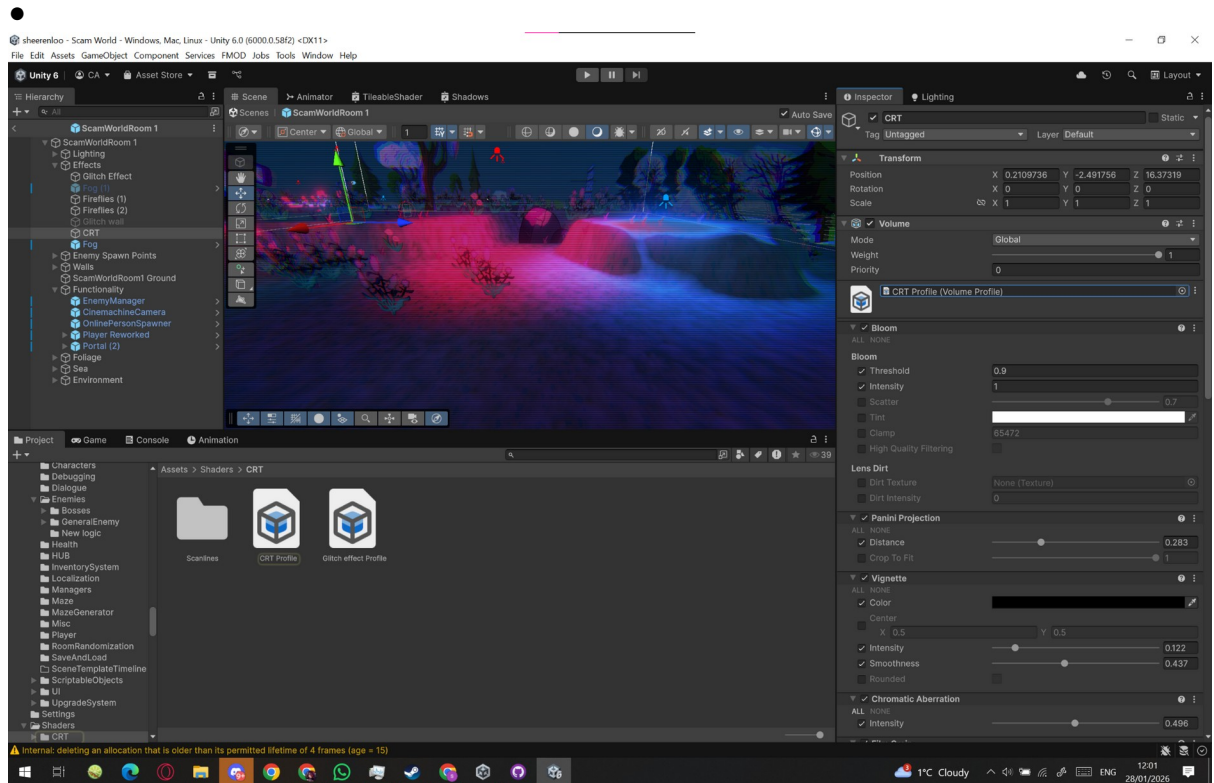


- Particle effects will be your best friend when creating fireflies, rain, light beams etc. I suggest seeking fit tutorials for each different particle system you wish to make, as I have made the ones in the game by experimenting (aside from the fireflies)



- Fog is very helpful when wanting to create a better forest or swamp feel for your level. I have utilised a particle system once again, and used a pre-made texture from the old team. After that, I just changed it between levels after copying it across

added it on the scene itself. Sometimes making a new object per prefab for things like these saves you the headache of trying to fix weird bugs and issues



4.1 Objective

Set the mood, clarify visibility, and finalize the level's feel.

4.2 Steps

- Configure **sun direction** and **light color** (day, sunset, night).
- Adjust skybox or create a custom one.
- Add **volumetric fog**, if needed, for mood and depth.
- Place **reflection probes** for accurate lighting.
- Add **post-processing** (Bloom, AO, Color Grading, Depth of Field).
- Review nighttime or indoor-outdoor transitions.

4.3 Key Considerations

- Make sure lighting supports gameplay visibility.
- Optimize shadows to avoid performance drops.
- Keep light intensity consistent with player goals (e.g., brighter near objectives).

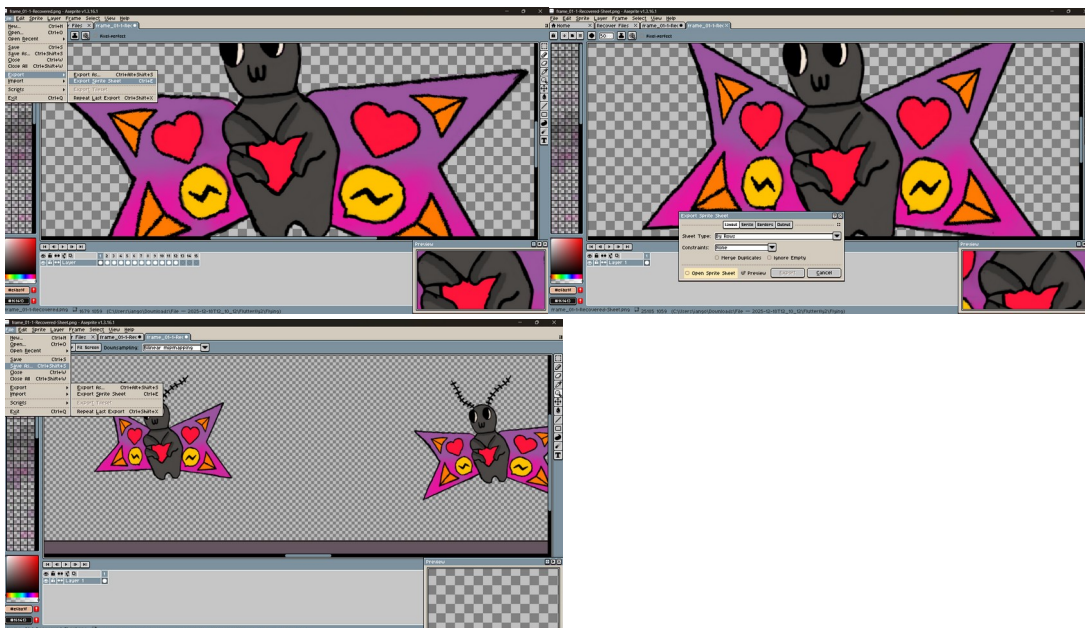
5. Implementing animations into unity (Artist)

5.1 Objective

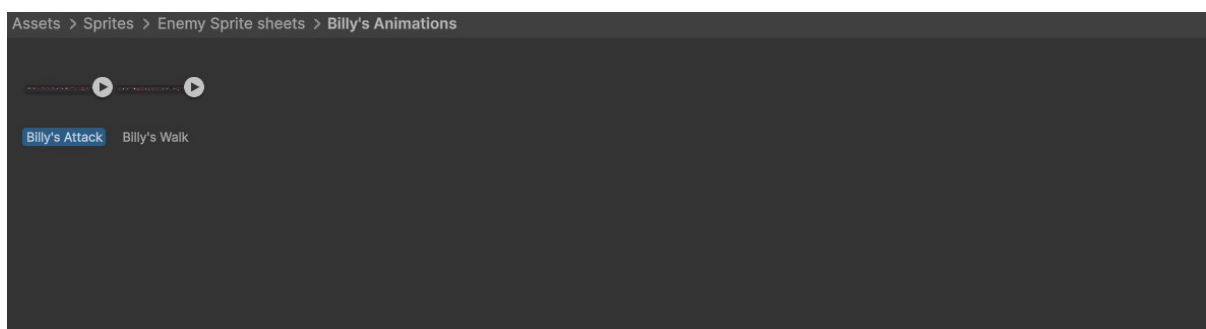
Putting animations into unity so that they can work where needed in unity.

5.2 Steps

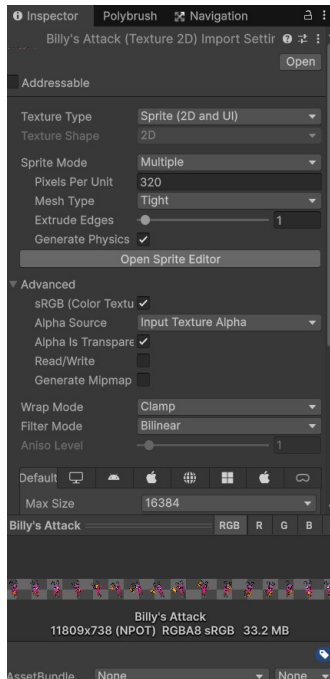
- Put all sprites in a program that can make sprite sheets, like Asesprite
- To make a sprite sheet in Asesprite, put all the frames in the program, go to file, export and the export sprite sheet. Then make sure you click on open sprite sheet and export. You should now have a sprite sheet and the last thing to do is to go to file, save as and then save it as a png.



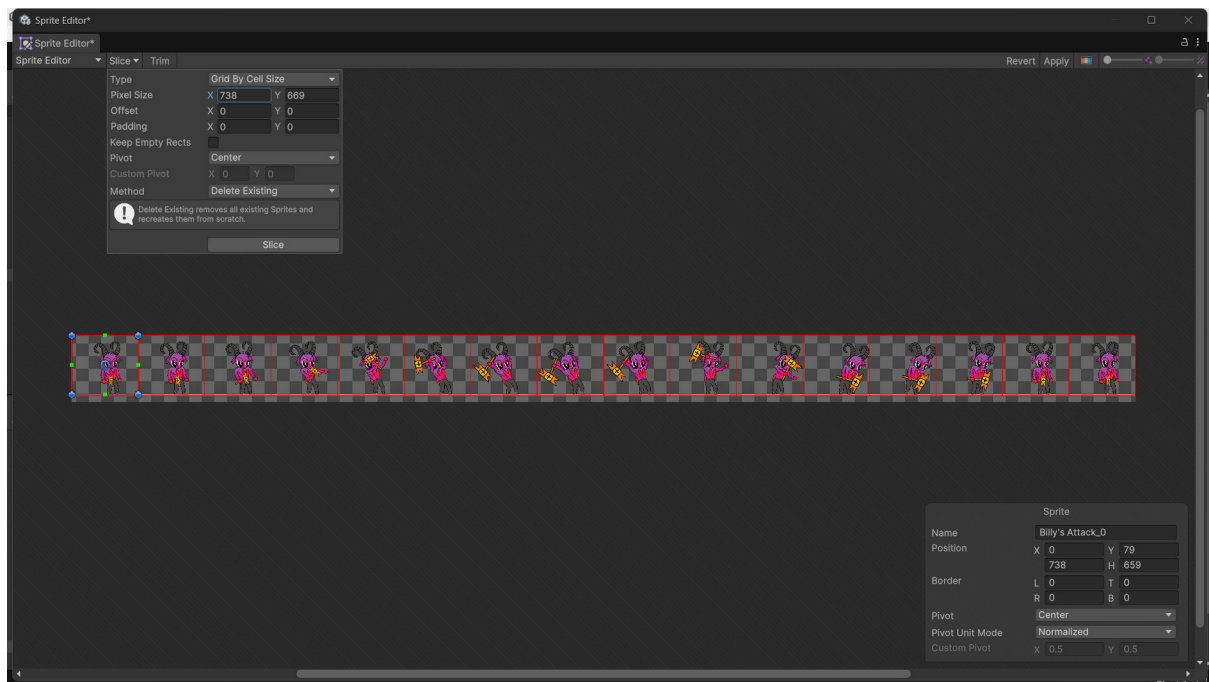
- When you have the sprite sheet import it into unity into the right folder. Which for an instagram enemy for example would be: Assets > Sprites > Enemy Sprite sheets > *enemy name*'s animations. To import the animation you can just drag the file into unity from your file explorer.



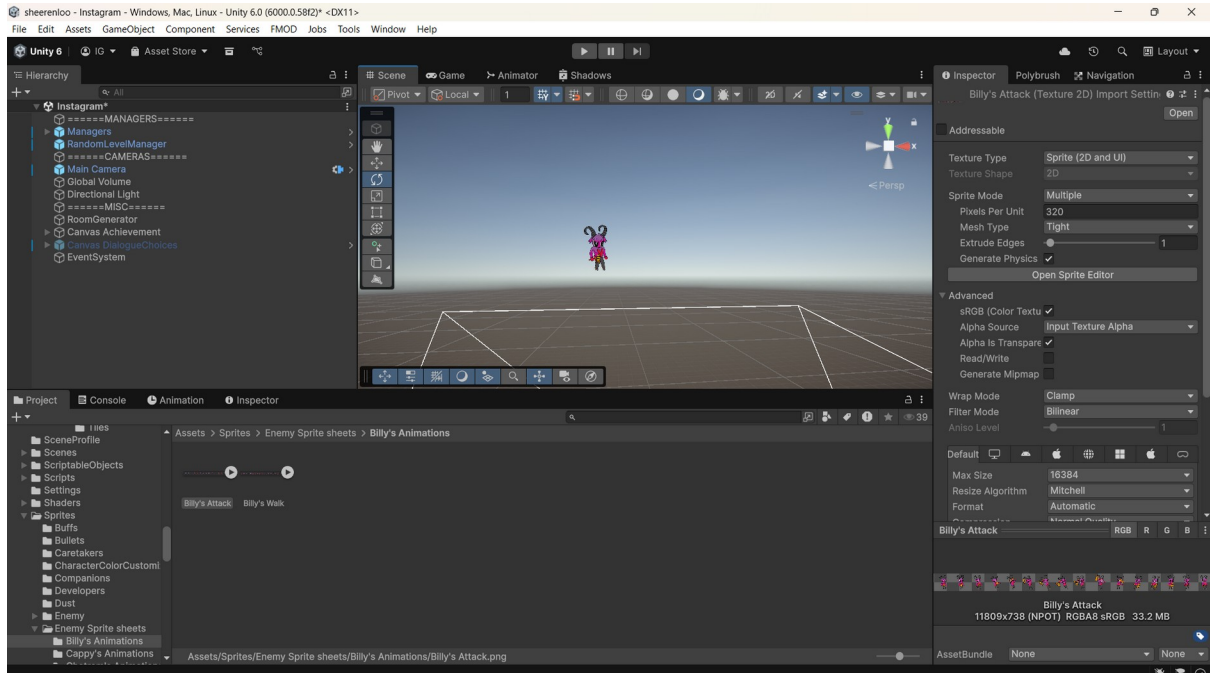
- Then click on the sprite in the project window(at the bottom of your screen. And in the inspector click on open sprite editor.



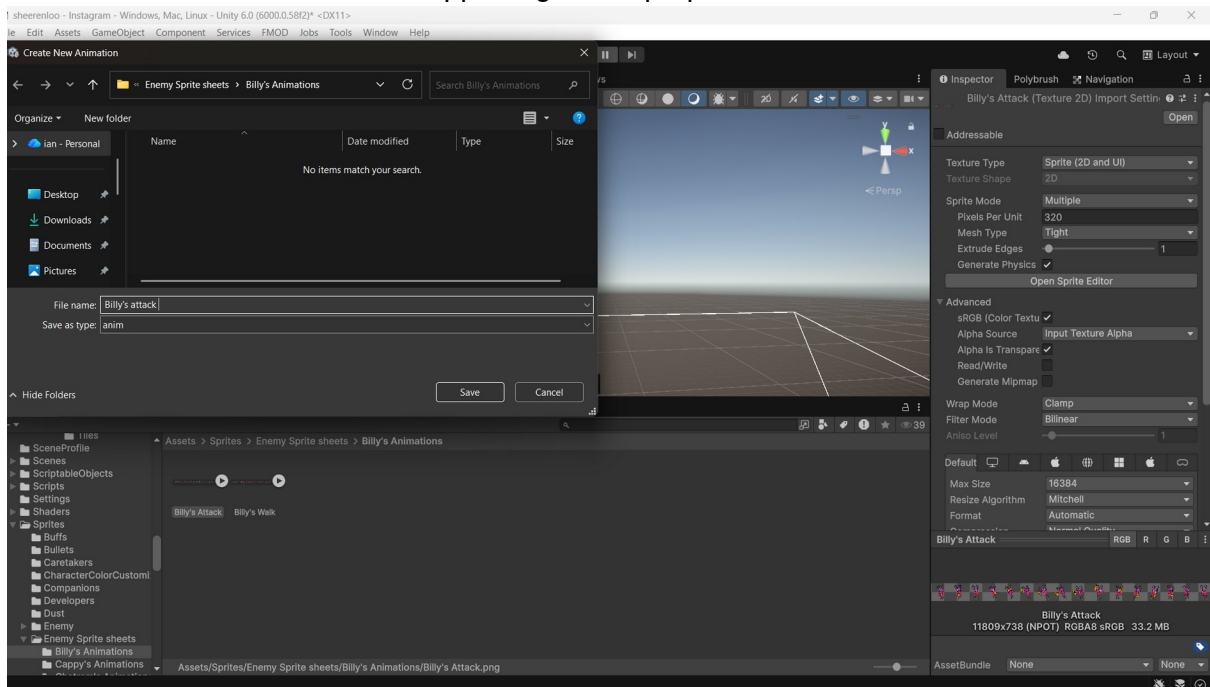
- In the sprite editor go to slice, then change the type to grid by cell and play with the pixel size until you can cut it right and then press slice. Then click apply and then you can close this window.



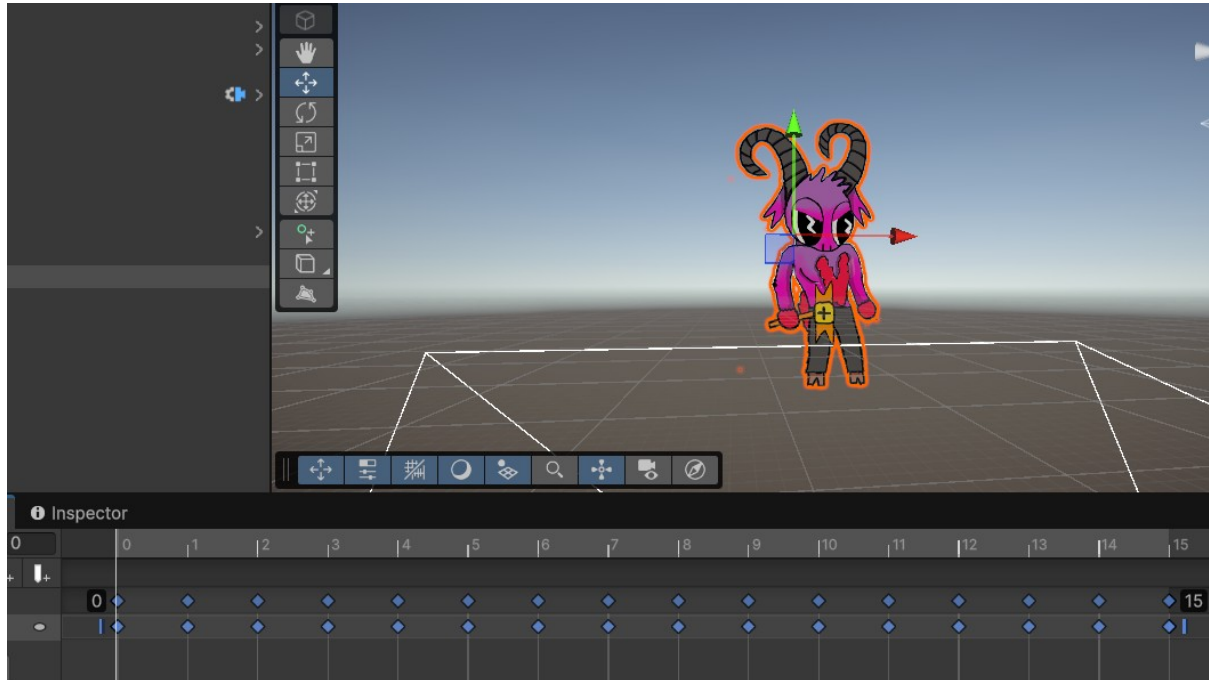
- After this is done select the sprite sheet and just drag it into the scene.



- Then a save window will appear, give it a proper name and save the animation.



- After that is done just click on the created sprite in the scene itself and go to animation and see if the animation plays at the right speed. If not, select all the frames and just drag them more apart by clicking the last frame and then literally dragging it.



5.3 Key Considerations

- Make sure the sprite image is big enough, otherwise it might not slice at all in unity.

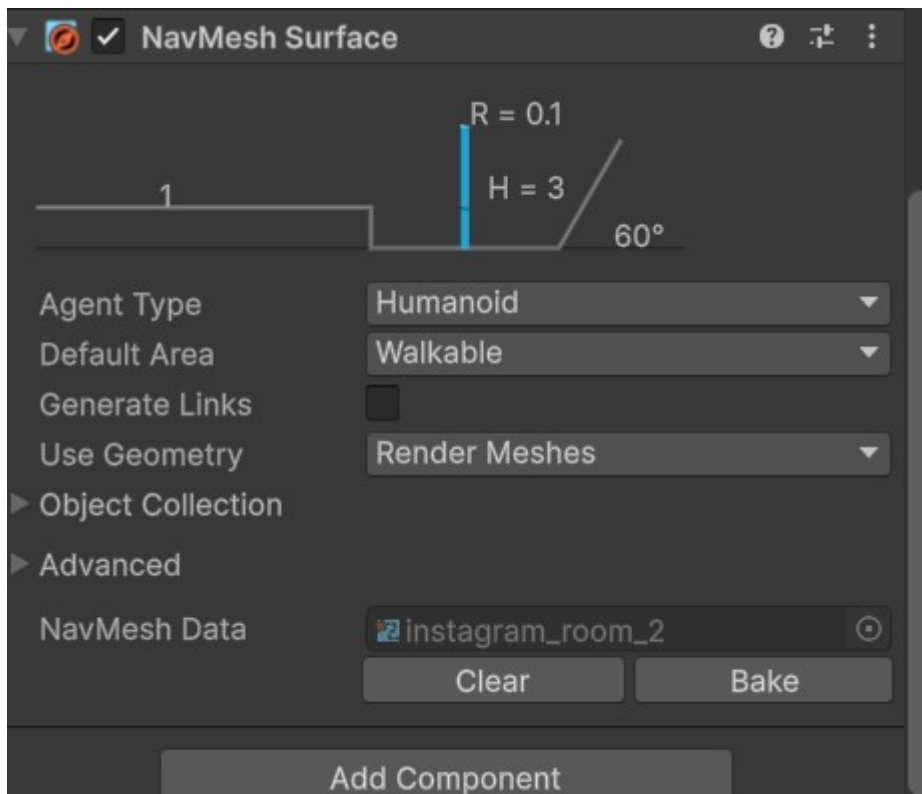
6. Navigation Mesh (NavMesh) & Testing (Designer)

6.1 Objective

Bake AI navigation and verify they can traverse the level correctly.

6.2 Steps

- Add **NavMesh Surface** component



- Bake NavMesh and verify coverage by pulling it into the scenes
- Test AI movement:
 - Pathfinding
 - Chasing
 - Patrols
 - Obstacle avoidance
- Ensure player and AI gameplay loops work as expected.
- Fix any terrain bumps, steep slopes, or inaccessible zones.

6.3 Key Considerations

- AI must not get stuck near rocks or trees.
- **Before baking the NavMesh turn off all object in the prefab since it can interfere with the Navmesh**
- Areas requiring precision traversal may need custom **Off Mesh Links**.
- Re-bake after layout changes.

Final Verification Checklist

Terrain

- Landforms readable
- No sharp terrain edges unless designed
- Clear player paths
- No uncrossable terrain without dash

Gameplay

- All interactables functional
- No unreachable regions
- Smooth pathing for player and AI

Visual Polish

- Texture blends naturally
- Foliage density reasonable
- Lighting readable and atmospheric

Performance

- Acceptable frame rate
- Efficient shadows and foliage
- Clean NavMesh without unnecessary detail