HADLEY, JAMES

Teesside University

Project Diary

Contents

[Milestone 1: Game Assets 2](#_Toc34601574)

[Milestone 2: Procedural Generation 2](#_Toc34601575)

[2.1 Terrain Generation 2](#_Toc34601576)

[2.2 Structure Generation 4](#_Toc34601577)

[Milestone 3: Player 5](#_Toc34601578)

[Milestone 4 AI 5](#_Toc34601579)

[Milestone 5 Pickups 5](#_Toc34601580)

[Milestone 6 UI 5](#_Toc34601581)

[Milestone 7 Polish 5](#_Toc34601582)

[Milestone 8 Submission 5](#_Toc34601583)

# Milestone 1: Game Assets

All relevant game assets were found on the Unity Asset store on Tuesday 4th February. Below is a table of contents for the Assets used within the project, referencing the author, link to the asset and how I intend on using the asset.

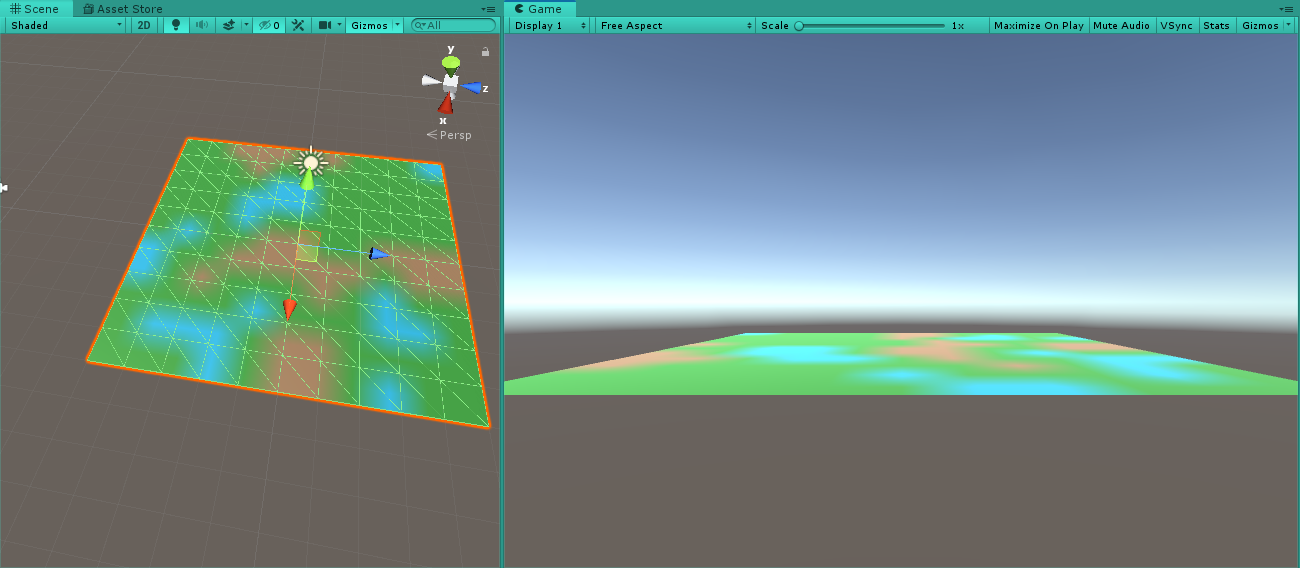
|  |  |  |  |
| --- | --- | --- | --- |
| Asset | Author | Link | Use |
| Prototyping Pack (Free) | DigitalKonstrukt | <https://assetstore.unity.com/packages/3d/prototyping-pack-free-94277> | Structures |
| Modular self-stand fence | Aleksey Kozhemyakin | <https://assetstore.unity.com/packages/3d/props/modular-self-stand-fence-105862> | Walls / cover |
| 18 High Resolution Wall Textures | A dog's life software | <https://assetstore.unity.com/packages/2d/textures-materials/brick/18-high-resolution-wall-textures-12567> | Texture structures |
| Seamless Textures (Realistic) | 2kPixel Studios | <https://assetstore.unity.com/packages/2d/textures-materials/seamless-textures-realistic-105177> | Texture terrain |
| Modern Weapons Pack | 7XF Design | <https://assetstore.unity.com/packages/3d/props/guns/modern-weapons-pack-14233> | Weapons to be used by both the player an AI |
| Low Poly Soldiers Demo | Polygon Blacksmith | <https://assetstore.unity.com/packages/3d/characters/low-poly-soldiers-demo-73611> | Character models for both the player and enemy AI |

# Milestone 2: Procedural Generation

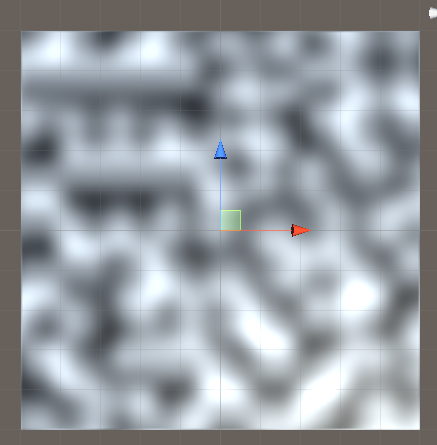
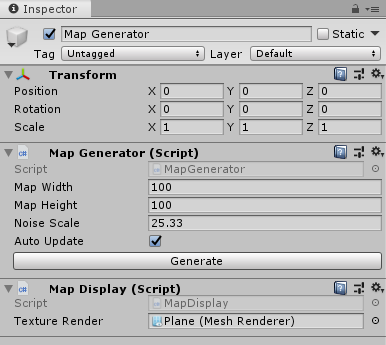
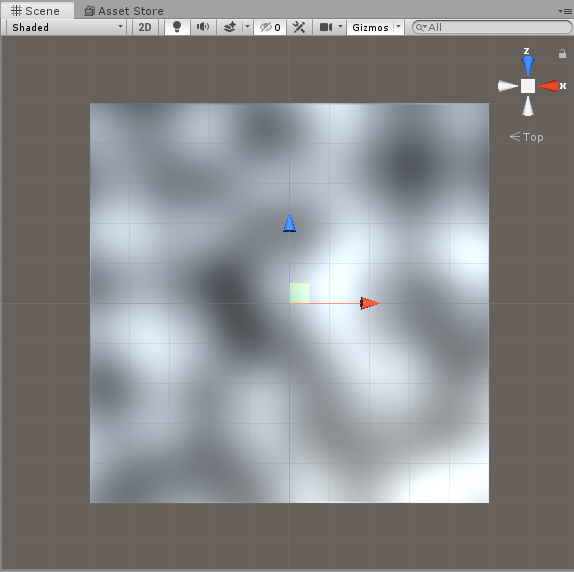
## 2.1 Terrain Generation

Terrain level generation started Thursday 13th February, with the current progress shown below: The screenshots below show basic tile generation, where the tile is textured with 3 different colours, which represent water (blue), grass (green), mountain (brown).

The current progress shown below is created through the use of two C-Sharp scripts and a 3D game object plane. The terrain generation for the game will be made of multiple of the plane objects to create an overall level. Once this stage of procedural generation is complete the next stage is to work on the build structures to be randomly created and placed around the generated terrain.

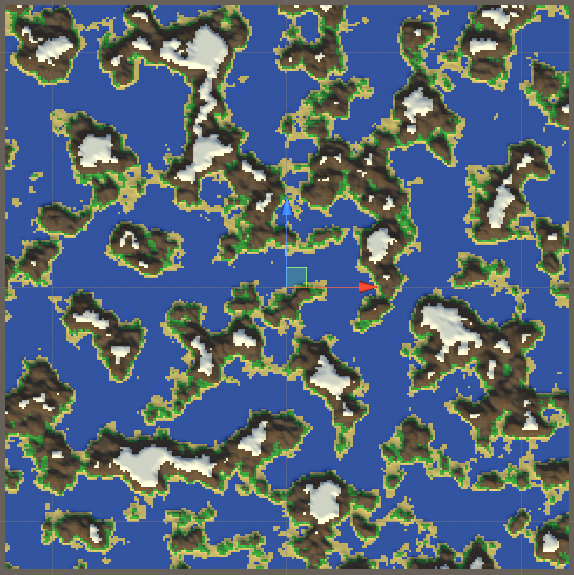
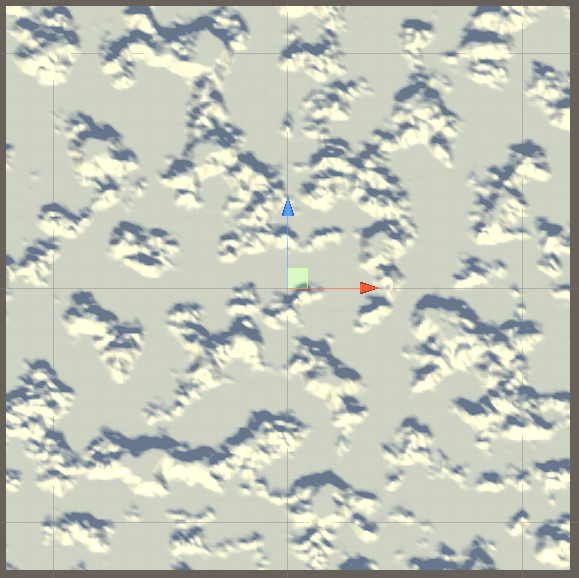


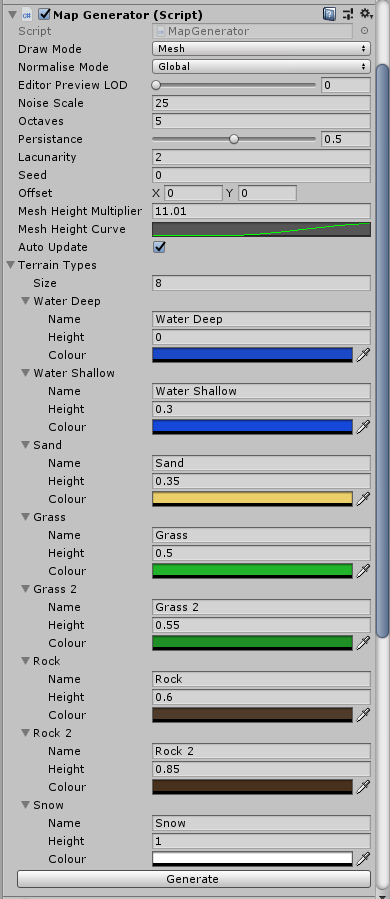
Friday 21st February, for the map generation I choose to change the direction I was working in and work directly use perlin noise and creating a noise map, with doing this it I had to start over with my current development of my map. However, ever though this is minor setback I feel this new approach with give the product an overall better finish then the original path shown in the screenshots above.



Above is perlin noise being applied to a plane, this can be change automatically in the editor shown in the screenshot above be pressing the generate button in the inspector using the map generator game object. The screenshot on the right is a newly generated map with a lowered noise scale in comparison to the screenshot on the left.

Monday 2nd March, the terrain was created and textured as shown below through the using the map generator script. The screenshot of the left shows the terrain generated with different height levels, this is done through a height map and scaled more accurately with the use of animation curve. The screenshot to the right shows the terrain textured. This is done in a separate script called texture generated which is then called in the map generator script to texture the terrain when the generate button in the inspector is pressed. The animation curve is used to mesh height according so the terrain sustains a set height between one and zero. The height curve as curve in place at around 0.4 this is done so that terrain such as water and sand remain flat whereas terrains types above 0.4 will be more elevate such as grass, rock and snow. The values for the different height are shown in the screenshot below showing the different types, with their determined heights and colour.





Wednesday 4th March, once the level of detail was applied to the terrain in the form of different height levels and textures the next step was to create endless terrain. For this I created a script named endless terrain, where the chunks of the terrain are only spawned in once visible to the player. Then once they are no longer visible to the player the chunks are then hidden from the player, also with the script I applied level of detail. This is based on how far away the chunks are from the player, meaning that the closer the terrain chunk is to the player the higher level of detail that chunk will have whereas the further away the chunk is from the player the level of detail will then be lower. A viewer object is then used in the scene to represent this and when the viewer is moved around the screen the map generator will highlight which terrain chunk is being viewed. Below is a short video demonstrating how the endless terrain system works within the scene. As shown in the video, the viewer is moved around the screen and new terrain chunks are shown, this is also highlighted in the hierarchy as “Terrain Chunk” becomes a darker colour font when the chunk is in the viewer otherwise the terrain chunk is greyed out as it cannot be viewed.



Friday 6th March, collisions we applied to the terrain, this was done by applying the mesh collider component to the script and applying a bool function use for collider, where in the inspector the tick box can be tick to allow collision to occur at a set level of detail. To test this I had to add a RigidBodyFPSController to the scene. The short video below shows that the first person character can walk around the generated terrain and collide with the world created. The game can also be paused using the RigidBodyFPSController, this is done by pressing the escape key and is shown in the few seconds of the video.



## 2.2 Structure Generation

# Milestone 3: Player

Due 19th March

# Milestone 4 AI

Due 9th April

# Milestone 5 Pickups

Due 16th April

# Milestone 6 UI

Due 30th April

# Milestone 7 Polish

Due 7th May

# Milestone 8 Submission

Due 8th May