|  |
| --- |
| Pandemonium Lab notes |
| Large rock formation in rocky landscape under blue sky |

|  |  |
| --- | --- |
| 12/07/2020 | By Group 4 |

This document is an overview of how the game “Pandemonium” was created and the resources used to make it.

# Resources used

## Unity Asset Store

### Blue Cartoon GUI Skin:

### <https://assetstore.unity.com/packages/2d/gui/blue-cartoon-gui-skin-19535>

Used for Health and Exp Bars

### Bubble Font (Free Version):

### <https://assetstore.unity.com/packages/2d/fonts/bubble-font-free-version-24987>

Custom Font for all text

### Cartoon Temple Building Kit Lite: <https://assetstore.unity.com/packages/3d/environments/dungeons/cartoon-temple-building-kit-lite-110397>

Used only on portals but was intended for level design.

### Dragon the Soul Eater and Dragon Boar:

### <https://assetstore.unity.com/packages/3d/characters/creatures/dragon-the-soul-eater-and-dragon-boar-77121>

Used the model and animation for the final boss.

### Effect textures and prefabs:

### <https://assetstore.unity.com/packages/vfx/particles/effect-textures-and-prefabs-109031>

Used for magic hit effects.

### Farland Skies - Cloudy Crown:

### <https://assetstore.unity.com/packages/2d/textures-materials/sky/farland-skies-cloudy-crown-60004>

Skyboxes

### Meshtint Free Polygonal Metalon:

### <https://assetstore.unity.com/packages/3d/characters/creatures/meshtint-free-polygonal-metalon-151383>

Used the model and animation for the A rank Giant bug enemy

### Mini Legion Grunt PBR HP Polyart:

### <https://assetstore.unity.com/packages/3d/characters/humanoids/fantasy/mini-legion-grunt-pbr-hp-polyart-98187>

Used the model and animation for C rank Grunt Enemy.

### Mini Legion Rock Golem PBR HP Polyart:

### <https://assetstore.unity.com/packages/3d/characters/humanoids/fantasy/mini-legion-rock-golem-pbr-hp-polyart-94707>

Used the model and animation for B rank Golem Enemy.

### RPG Monster Duo PBR Polyart:

### <https://assetstore.unity.com/packages/3d/characters/creatures/rpg-monster-duo-pbr-polyart-157762>

Used the model and animation for D rank Slime and Big Slime Enemy.

### RPG Wizard Character + Effect:

### <https://assetstore.unity.com/packages/3d/characters/humanoids/humans/rpg-wizard-character-effect-164613>

Used the model for Player Wizard.

### Simple UI & icons:

### <https://assetstore.unity.com/packages/2d/gui/icons/simple-ui-icons-147101>

Used for all the UI menus and buttons.

## Instructional Sources

### Cinemachine:

### <https://www.youtube.com/watch?v=537B1kJp9YQ>

### Data Flow:

### <https://www.youtube.com/watch?v=WchH-JCwVI8>

### Loading Screen:

### <https://www.youtube.com/watch?v=YMj2qPq9CP8>

### NavMesh Tutorials:

### <https://www.youtube.com/watch?v=CHV1ymlw-P8>

### <https://www.youtube.com/watch?v=blPglabGueM>

### Pause Menu:

### <https://www.youtube.com/watch?v=JivuXdrIHK0>

### Shooting:

### <https://www.youtube.com/watch?v=6eIVxyxoimc>

### <https://www.youtube.com/watch?v=THnivyG0Mvo>

### <https://www.youtube.com/watch?v=Nke5JKPiQTw>

### Third Person Movement:

### <https://www.youtube.com/watch?v=4HpC--2iowE>

### UI Bars:

### <https://www.youtube.com/watch?v=BLfNP4Sc_iA>

### Unity Documentation:

### <https://docs.unity3d.com/ScriptReference/index.html>

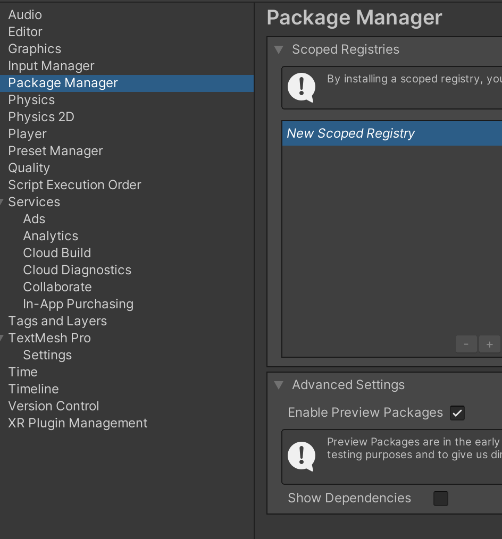
## Other Resources

### NavMesh Components: <https://github.com/Unity-Technologies/NavMeshComponents>

Instructional Summary

# Prepare Project

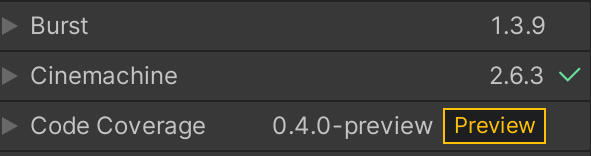
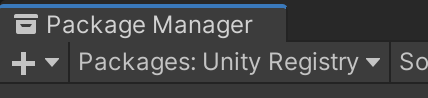
Make a URP project instead of the normal 3D Template. In the project settings Enable Preview Packages.

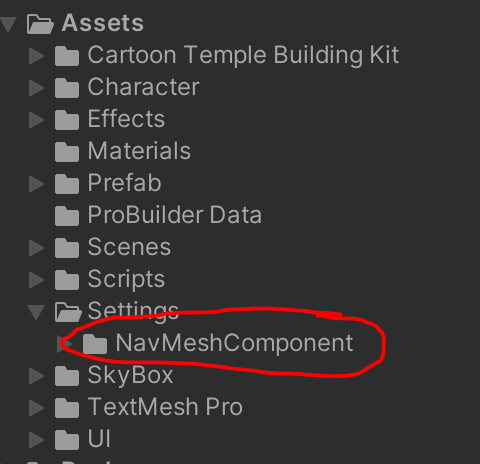


Then install Cinemachine, ProBuilder, and ProGrid from the Unity Registry.

Cinamachine for the camera logic.

ProBuilder for easier level prototyping.

ProGrids for easier alignment when making the level.

Lastly, copy the asset folder of the navmesh component from github to the inside the asset folder of the project

# Animations and Animator

## Animations

For all the movement animation, set to loop.

The other animation like getting hit, attaking, and death should not loop.

## 

## Animator

### Player Animator

Add a Blend Tree for movement, and two states for jumping and death.

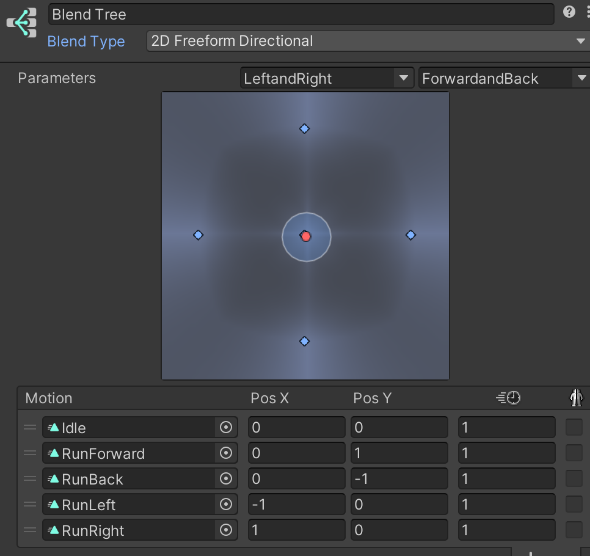
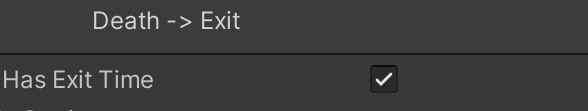
Add 3 parameters, 2 float and 1 bool. These are used for forward and back, left and right, and jumping.

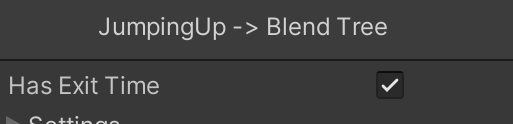
Make the blend tree the default state.

For the blend Tree:

Add 3 transition:

* Blend tree to Jump: set condition using bool parameter set to true.
* Jump to Blend tree: set true to ”has exit time”.
* Death to Exit: set true to “has exit time”.





### 

### 

### 

### Enemy Animator

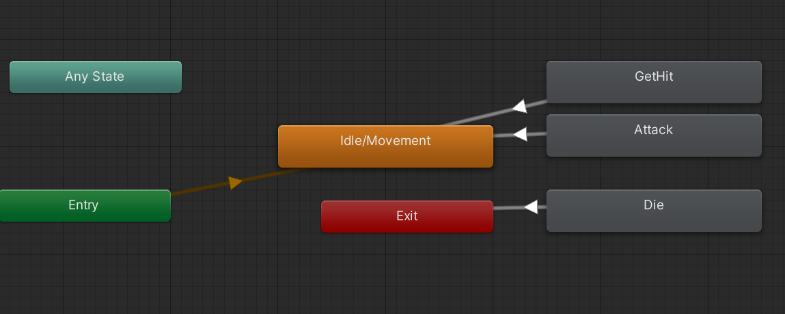
Make a base Animator for all the animator override to be based on.

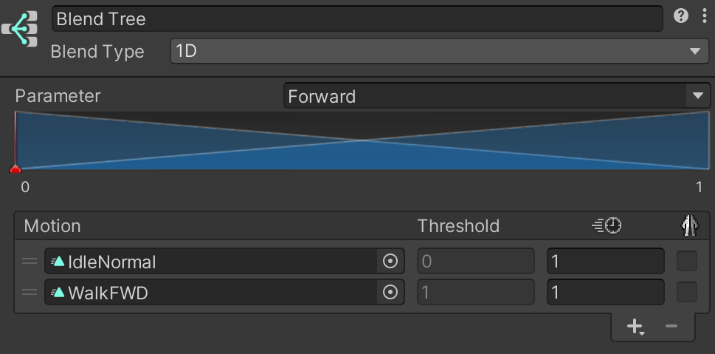
Add 1 float parameter.

Add 1 Blend Tree for movement, and 3 states for getting hit, attaking, and death.

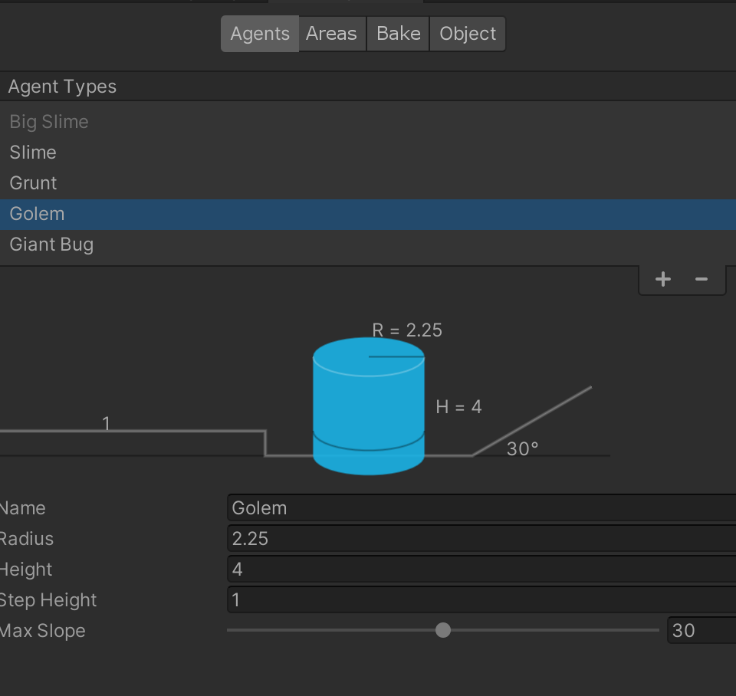
Make the blend tree the default state, add transition from getting hit and attacking to the blend tree, and add a transition from death state to exit.

Make all 3 state transition set true to “has exit time”.

For the blend tree:



Make an animator override controller for each enemy and set the controller to the base animator. With that you can set unique animation for each states like in the base animator without recreating the animator.



NavMesh Agents

Create all the agents needed for all the enemies and customized according to the size of each enemy

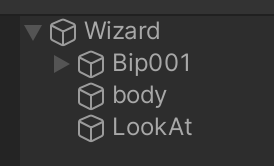
# Prefabs

## Characters

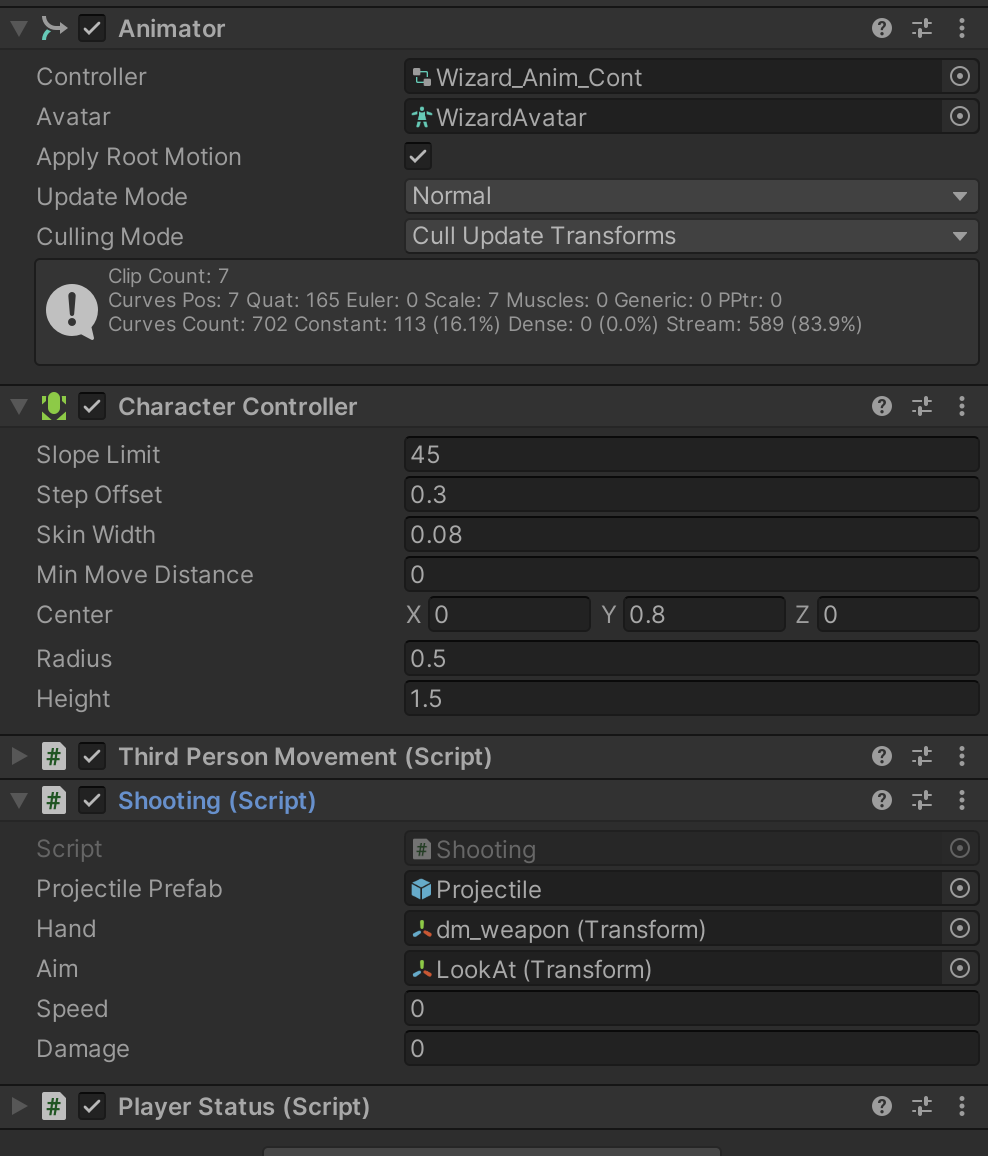
### Wizard

Using the premade wizard prefab from the “PRG Wizard Character + Effect”, add an Empty Object to act as the object that contains the “Mouse aim” script\*. Change the tag to “Player”

Add the following

* Animator; initialized with the player animator.
* Character Controller; customized based on the mesh used.
* Third Person Movement script
* Shooting script; Initialized with the projectile prefab, the transform of where the projectile would come out (rig of the hand), and the aim object in the player’s head.
* Player Status script

\*was needed before but could be removed and transfer the scripts in the player object.

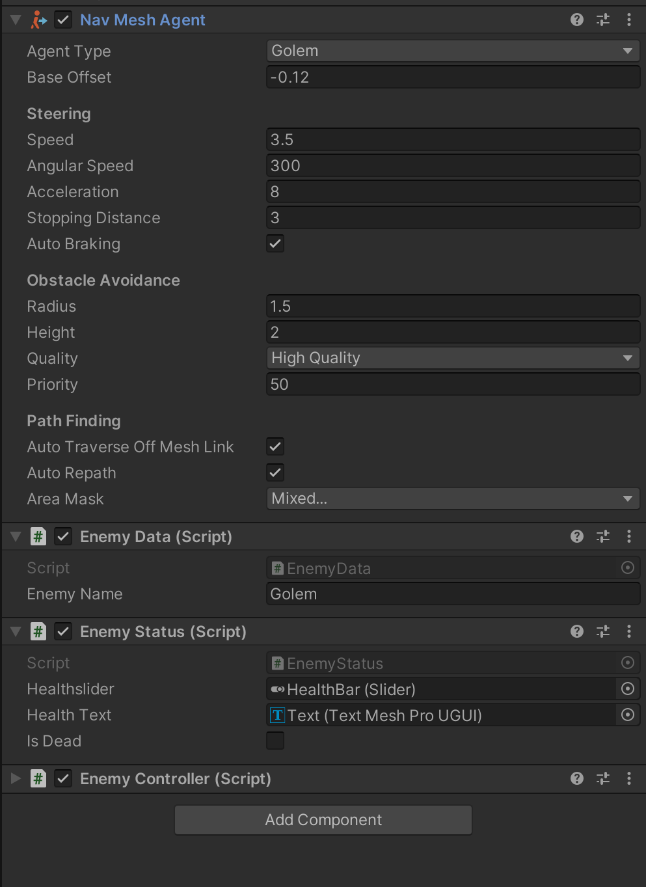


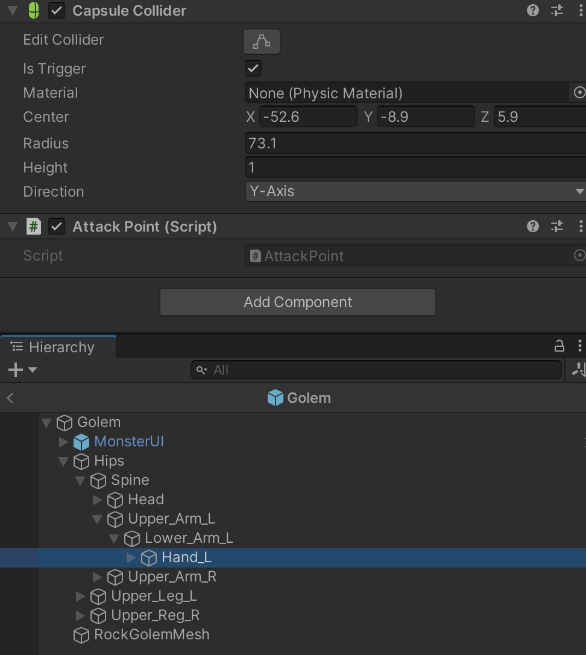
### Enemy

With the same structure as the player, each enemy with have an empty game object with the mesh and rig as a child object. Change the tag to “Enemy”

Add the Enemy Health bar for prefab as a child the position at the right height.

In the rig, add the collider until the whole mesh is covered with collider. At the portion that will be used to attack the player like a fist or a weapon, Add a Trigger collider and the “Attack Point” Script.

In the parent object, add the following:

* Animator: Initialize controller with the animator override controller.
* RigidBody: Freeze both position and rotation in all three axis.
* NavMesh Agent: Initialize the Agent type, Base offset, Steering Angular Speed and Stopping Distance(used for range for attacking), and Obstacle Avoidance Radius and Height (can use tweaked agent type definition).
* Enemy Data script: Initialize name.
* Enemy Status script: Initialize the slider and text using the health bar prefab.
* Enemy Controller script

## 

## Projectiles\*

The projectile is just a sphere with the following:

* Tag changed to “ProjectilePlayer”\*\*
* Mesh Renderer with a Material that has the color and emission.
* Sphere Collider with a Physic Material for how bouncy it is.
* RigidBody
* Light: a point light, initialized with bright the light is and what color.
* Projectile Script: initialized with the effect prefab from the “Effect texture and prefabs” assets

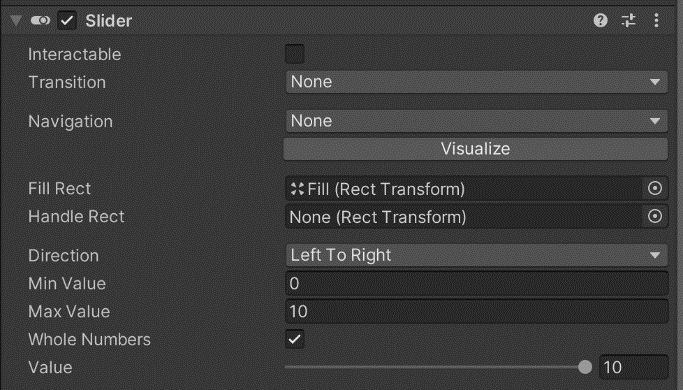
\* the plan was to make it elemental with particles and light trails.

\*\* was not managed to add shooting mechanics for the enemy

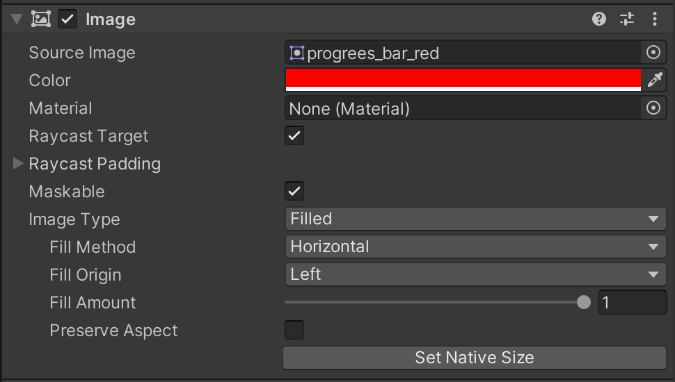
## UI

### Enemy Health Bar

This health bar is used only for the enemy.

This is a canvas element with render mode as World Space and has the Bill board script.

For the child object is a UI Slider with the Interactable set to false, Transition and Navigation as none, deleted handle, direction left to right and whole numbers set to true.

The Background and Fill has a custom sprite from the “Blue Cartoon GUI Skin” asset. The Fill image’s Image type is set to filled, Filled method set to Horizontal, and Fill Origin set to Left.

Lastly, add a TextMeshPro child to the slider with the custom font and the desired looks.

### Loading Screen

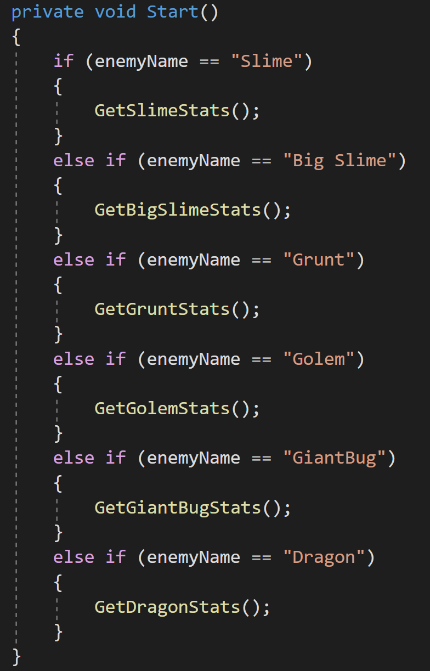
This is a UI Panel with a Slider with the same settings as the Enemy Health Bar Prefab but with different color and sprites.

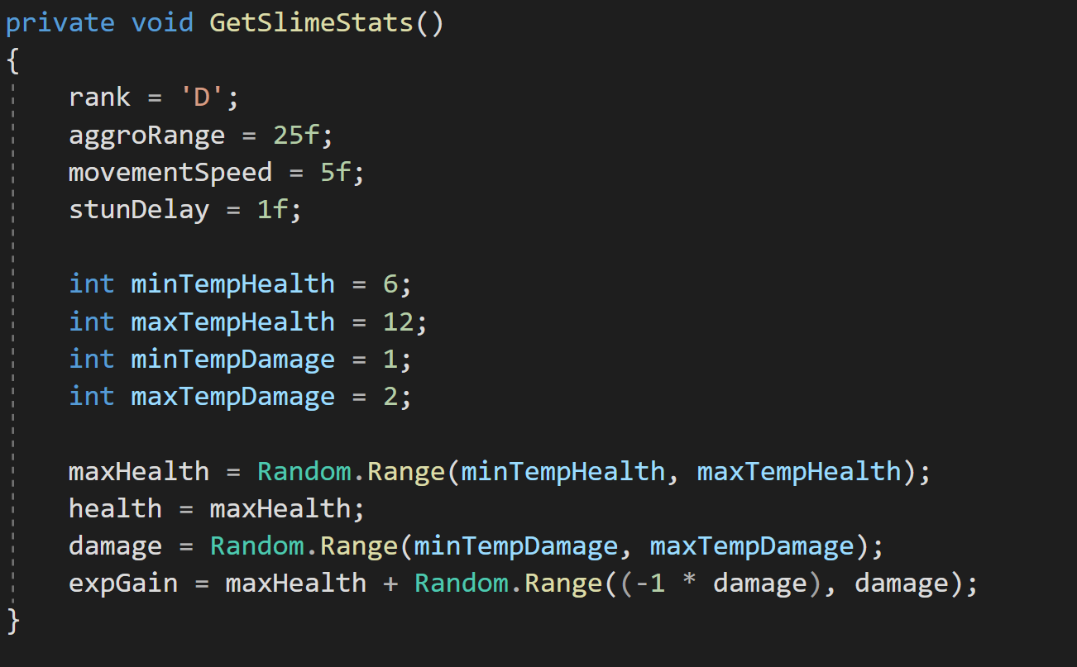
# Data

## Enemy Data Script

For each enemy add an if statement comparing the name initialized in the prefab and a function for the enemy.

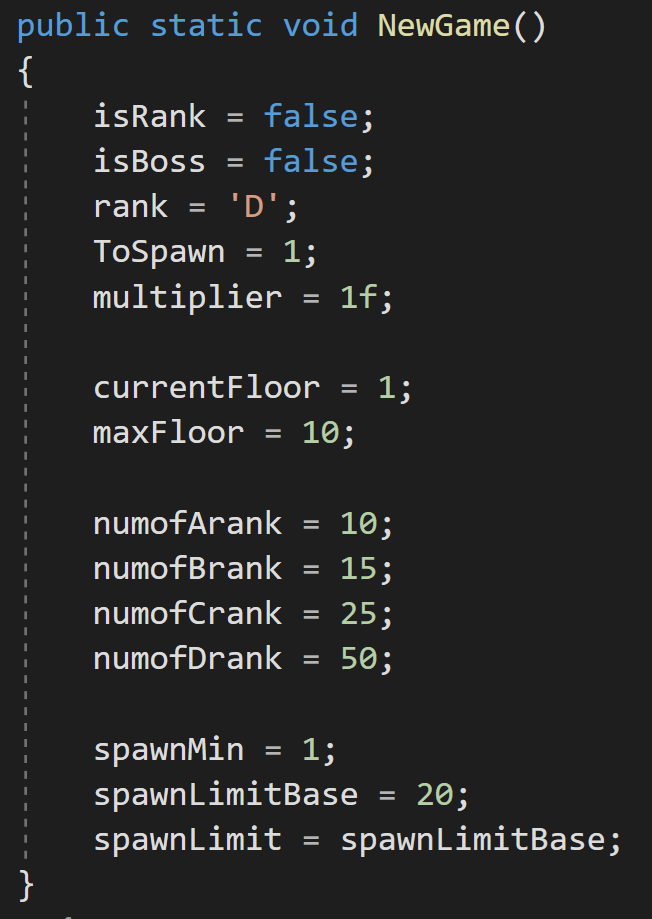
For the function, set the following:

* Rank: what rank the enemy is\*
* aggroRange: how for will the player be from the enemy before they react.
* movementSpeed: the speed of the navmesh Agent
* stunDelay: The interval of how long each getting hit reaction the enemy will have to the player’s attack
* MinTempHealth and MaxTempHealth: Range of values that will determine the enemy’s health
* MinTempDamage and MaxTempDamage: Range of values that will determin the enemy;s damage to the player.



## Player Data Script

A Static class that will keep track of the player data throughout the game. Set the following:

* Level: Starting level
* Intelligence: set the min and max initial intelligence
* Defense: set the min and max initial defense
* Constitution: set the min and max initial constitution
* Dexterity: set the min and max initial dexterity
* skillIncrement: how many skill points per level up
* MaxHealth: base max health and the random modifier
* MaxExp: how many exp until the player level up
* SkillPoints: how many skill point at the start of the game
* MovementSpeed: base speed for the player
* JumpSpeed: jump force of the player (constant)
* ProjectileRange: base lifetime of each projectile
* ProjectileSpeed: base speed of each projectile
* ShootDelay: base interval of each shot

## Tower Data Script

A Static class that will keep track of the level data throughout the game. Set the following:

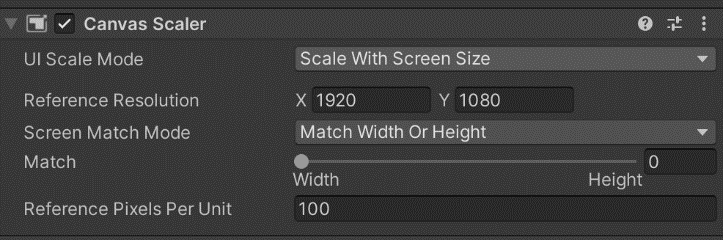
* MaxFloor: number of floors in the game
* Numof#Rank: number of monster for each rank of monsters (the total should be larger than the max floor)
* SpawnLimitbase: that largest possible number of monster to be spawn on each floor.

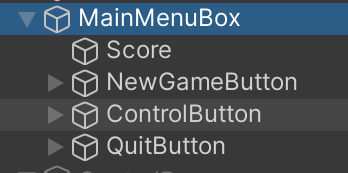
# Scenes

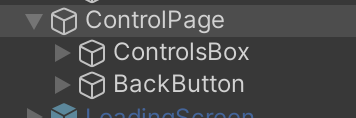
## Main Menu

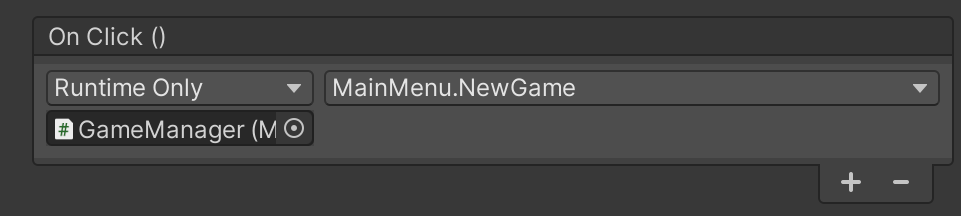
The Main menu scene should have an Events system, an empty game object that contains the scripts(Game Manager) , a Canvas Object which has all the menus, the loading screen prefab, and camera. The light is not needed.

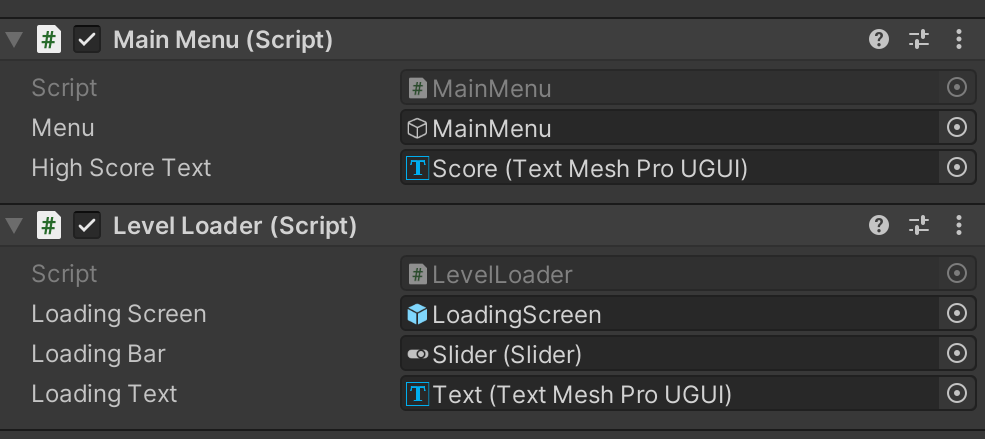
The Canvas object should be initialized with the Canvas Scaler Component set to Scale with Screen Size and the Resolution like 1080p or 4K for the menus to scale for any 16:9 ratio screens.

The Canvas object will have 3 child, A main menu, the control page, and the loading screen prefab.

The Main menu child will have a text that will display the current highscore, and 3 buttons. Each button will have an Onclick event from the game manager game object; MainMenu.Newgame for starting a new game, MainMenu.ControlPage for showing the control page, and MainMenu.QuitButton for quitting the game.

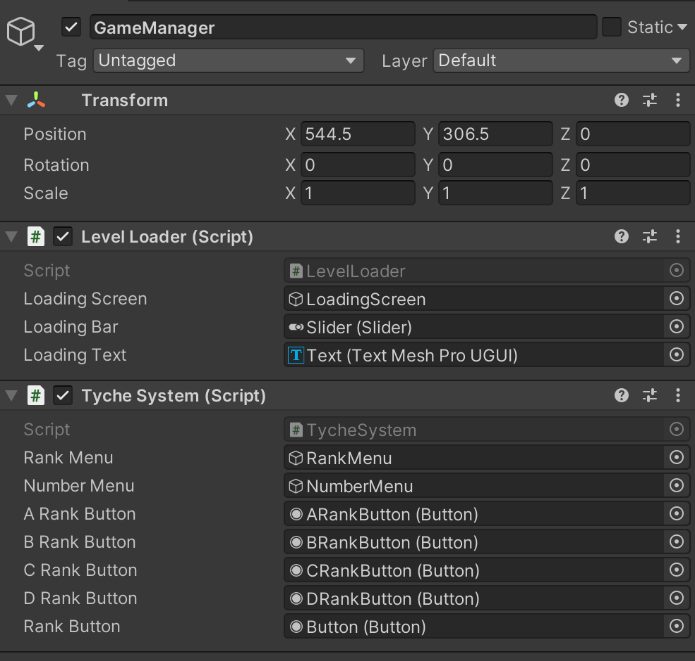
The Control page child will have a TextMeshPro that will be the control of the game and a button with an Onclick Event MainMenu.BackButton.

The Game Manager game object will have the Main Menu script and Level Loader script attached. Them Main Menu script is initialized with the Main Menu Child and the TextMexhPro object for the score. The Level Loader is initialized with the loading screen game object, the slider and the textMeshPro object.

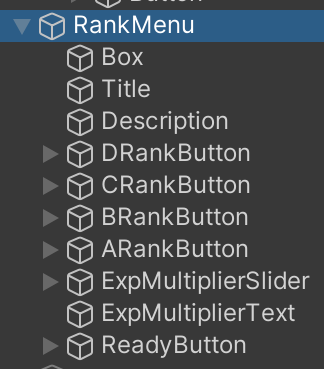


## Tyche System

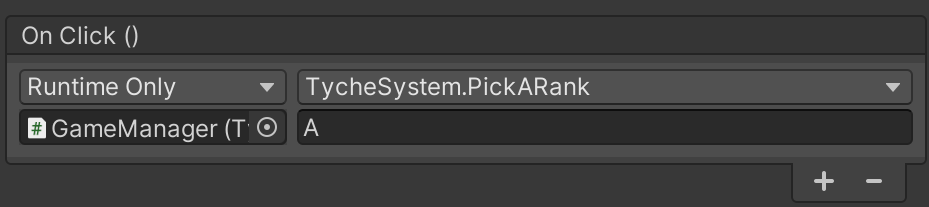
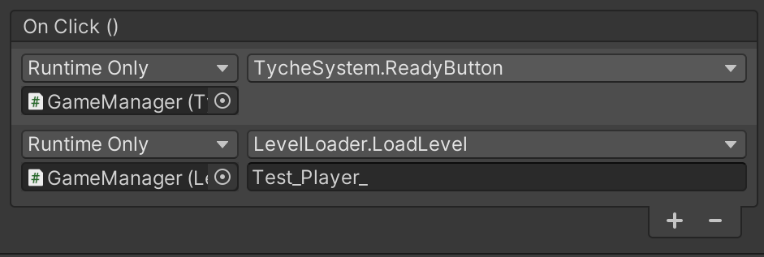
The Tyche System scene should have an Events system, an empty game object that contains the scripts(Game Manager) , a Canvas Object which has all the menus, the loading screen prefab, and camera. The light is not needed.

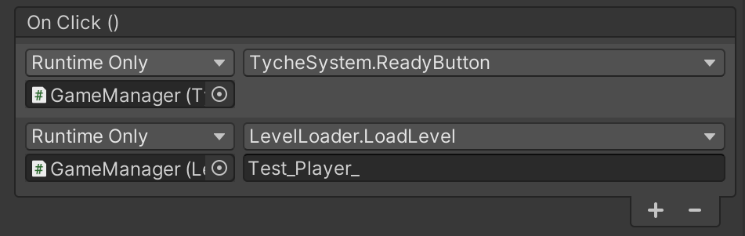
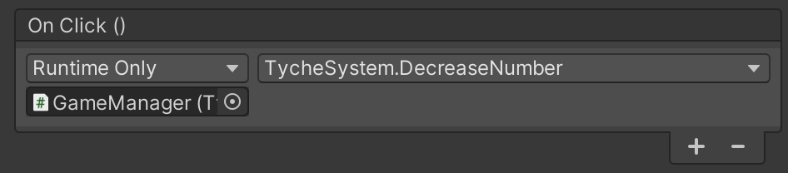
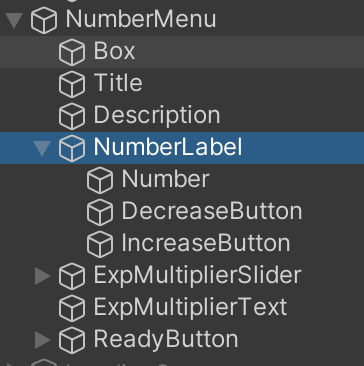
The Canvas object should be initialized with the Canvas Scaler Component set to Scale with Screen Size and the Resolution like 1080p or 4K for the menus to scale for any 16:9 ratio screens.

The Canvas object will have 4 child, A Initial Menu, the Rank Menu, the Numbers Menu, and the loading screen prefab.

The Game Manager object will have The Level Loader Script and Tyche System Script. The level loader script will be initialized with the loading screen game object, its slider, and its TextMeshPro. The Tyche system scrip is initialized with the Rank menu game object, Number Menu game object, The 4 choice buttons in the Rank menu and lastly, the rank button in the initial menu.

The Initial Menu will have Image for the background, TextMeshPro for the description for the choices and 2 buttons. One button has an Onclick event from the game manager TycheSystem.RankButton and the other button will have TycheSystem.NumberButton.

The Rank Menu will have an Image for the background, TextMeshPro for the description, 4 buttons with Onclick event from the game manager TycheSystem.PickARank with the parameter of the letter of the rank, A slider with the same setting as the loading screen slider but with a max value of 4, another TextMeshPro for the multiplier description, and lastly, another button with 2 Onclick event from the game manager TycheSystem.ReadyButton and LevelLoader.LoadLevel with name of the floor Level without the number.

 The Number Menu will have 3 TextMeshPro for menu description, player’s choice, and multiplier description, and 3 buttons for the decreasing the player’s choice, increasing it, and to continue. The Onclick events for the button will be from the game manager TycheSystem.DecreaseNumber, TycheSystem.IncreaseNumber, and 2 for the ready button, TycheSystem.ReadyButton, and LevelLoader.LoadLevel with the name of the floor level without the number as the parameter.

## Floor Levels

Each floor will be their own scene with the same name but has a number at the end. Each scene will have the following:

* An Event System
* Directional Light with the Sun script attached
* A level with all the mesh collider set up
* A list of empty game object that will act a spawn points
* An Empty game object with the spawner script attached, initialized the wizard’s transform component, the number of spawn points and their transform components, and the array of prefabs for each rank.
* An empty game object NavmeshSurface component attached. Attached multiple navemenshSurfave component for the same amount of agent configured and set each agent type.
* 3 Cinemachine Virtual Camera
  + A 3rd person camera with Follow and Look at initialized with the aim object in the player prefab, Body set to 3rd person follow and a wide fov in Lens.
  + A zoomed in camera with Follow and Look at initialized with the aim object in the player prefab, Body set to 3rd person follow, closer to the player’s head, and a narrower fov in Lens.
  + A death camera with Follow initialized with the aim object in the player prefab, Body and Aim set to Do nothing, and pointing down
* The Portal using a door prefab in the “Cartoon Temple Building Kit Lite” with a trigger box collider in the entry way. Also, Exit Level script attached initialized with the game manager game object.
* The Wizard Prefab with the Player Status script initialized with Health bar slider, exp slider, heath bar text, exp bar text, level indicator text, and score text. The aim object’s Mouse Aim script is initialized with 4 cameras and the 2 reticle images.
* A Canvas Object with Canvas Scalar set to Scale with Screen Size and Resolution of 16:9 ratio. Also have the following children:
  + 2 Images for normal Reticle and a reticle for when zoomed in
  + 2 slider for health and exp
  + 5 Textmeshpro for the health bar, exp bar, level indicator, score, and floor indicator
  + Loading screen prefab
  + Pause Menu with 3 buttons with the Onclick events from the game manager (1)PauseMenu.ResumeGame, (2) PauseMenu.OpenControlsPage, and (3) PauseMenu.BackToMainMenu and LevelLoader.LoadLevel with the parameter of the name of the main menu scene.
  + Controls menu with a Text for the instruction and a button with an Onclick event from the game manager PauseMenu.BackButton.
  + A Stats Page with text for the score, text for the player’s current level, text for the unused skill point, 4 text for each stats number, 4 buttions with LevelUpMenu.DecreaseButton event, 4 buttons with LevelUpMenu.IncreaseButton, and a button with LevelUpMenu.FinishedButton.
  + A Game Over page with a text to show the final score and a button with PauseMenu.BackToMainMenu and LevelLoader.LoadLevel with a parameter of the same name of the main menu scene.
* An empty game object with 5 scripts attached.
  + Game Manager Script initialized with the portal game object and the Floor indicator text
  + Pause Menu Script initialized with Pause Menu and Control Menu game objects
  + Level Loader Script initialized with the loading screen prefab, its slider, and its text
  + Game Over Script initialized with game over page game object and the score text.
  + Level Up Menu Script initialized stat page and player game objects, 6 labels for the level, score, and stats number, 2 array of buttons for decrease and increase button and lastly, the finish button.

When the first level scene is finished, this scene can be duplicated with the same amount of floors used. Make sure to iterate the name of the scene.\*

For each floor scene, customize the level game object, spawn points, and the portal location.\*\*

\*Current Tyche Level script deal with only 2 floors alternating.

\*\*Current Game Manager script deals with test level and have a random spawn point.

## Final Level

The final level is just a duplicate of the floor levels with little modification.

There is only 1 navmeshSurface.

In the spawner, there is only one spawn point and the s rank array is the only on initialized.

A duplicate of the game over page but modified to say the winner.

Win Script is added to the game manager initialized with the Win page game object and win page text.

When all the levels are done, go thorough each level and bake the navmeshsurface.