Game Development Test #3

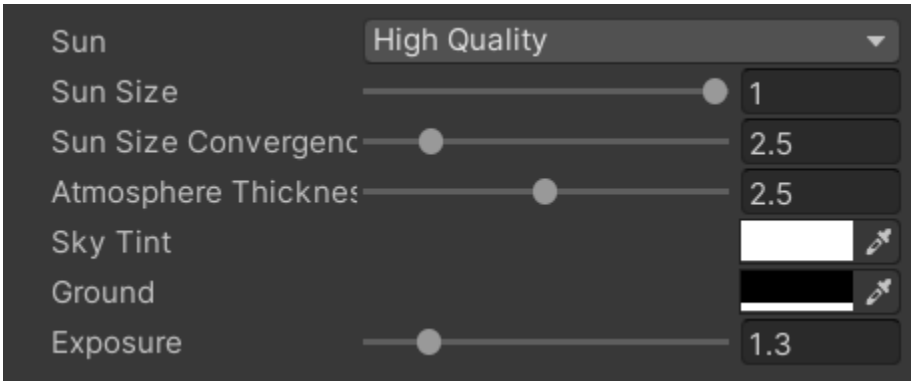
The test can be downloaded form https://github.com/Sgrygorczuk/Test_Three

On this test you will have to recreate this Main Menu Scene and provide functionality to this Game Scene. There is a total of 25 pts to get.

Part 1: Main Menu (19pts)

Part A: Create the Skybox (1pt)

Inside the Skybox folder create a new Material, change Shade to Skybox/Procedural and set it to have these settings:



Once you've created your skybox add it to the Scene so that it will look like this.

Part B: UI (4pt)

Create three buttons, give each of them the text of Play, Settings, and Exit and attach the Button Sprite to them, like so: (1pt)

Connect the MainMenuController to the Play Button onClick section and using the MenuScript select the GoToNextLevel function. (1pt)

Create an Image with the title sprite attached to it and place it on the top center of the Canvas, so it looks like so. (1pt)

In the Font Folder you will find a Font, create a TextMeshPro asset by right clicking on it and creating a TextMeshPro Font Asset. Attach the font to all three buttons such that it looks like so. (1pt)

Part C: Audio (3pt)

You will find two game objects in your Hierarchy one called Music and the other ButtonSFX, each of them has a Audio Source.

For the Music attach the “MenuMusic” Audio Clip to it and make sure it's set to Play on Awake and Loop. (1pt)

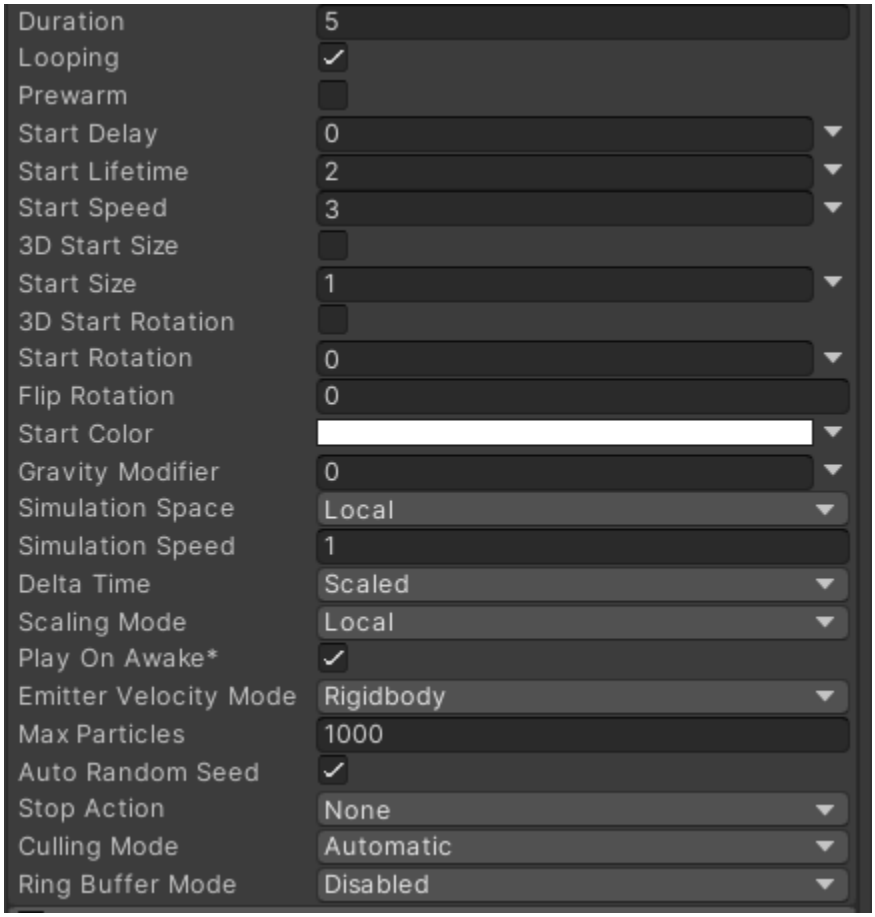
For the ButtonSFX attach the “ButtonClick” Audio Clip to it and make sure it’s set NOT to Play on Awake and NOT Loop. (1pt)

Create an Audio Mixer in the Audio Mixer Folder and create two Group one called Music and one called SFX. Connect the corresponding Groups to the Music and ButtonSFX outputs. (1pt)

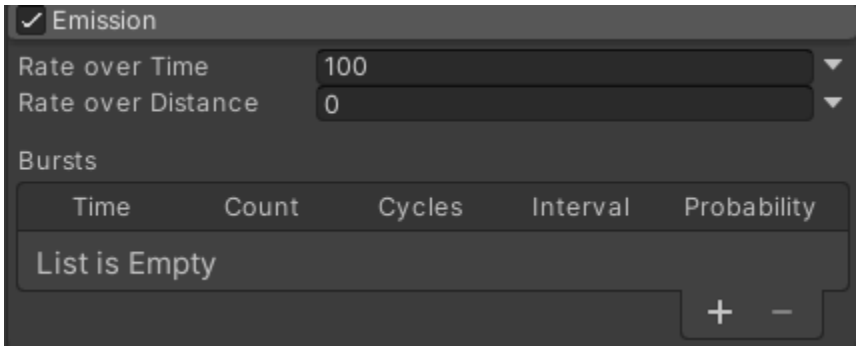
Part D: Particle System (6pt)

Create a Particle System in your hierarchy and make sure it's underneath the camera, not the canvas.

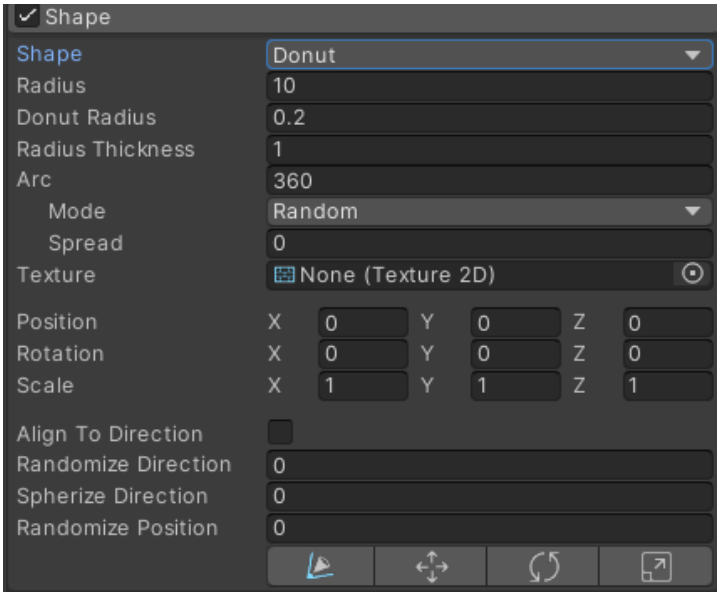
In the general Particle Setting make sure that Start LifeTime is 2 and the Start Speed is 3. (1pt)



Set the emission settings to make 100 particle (1pt)

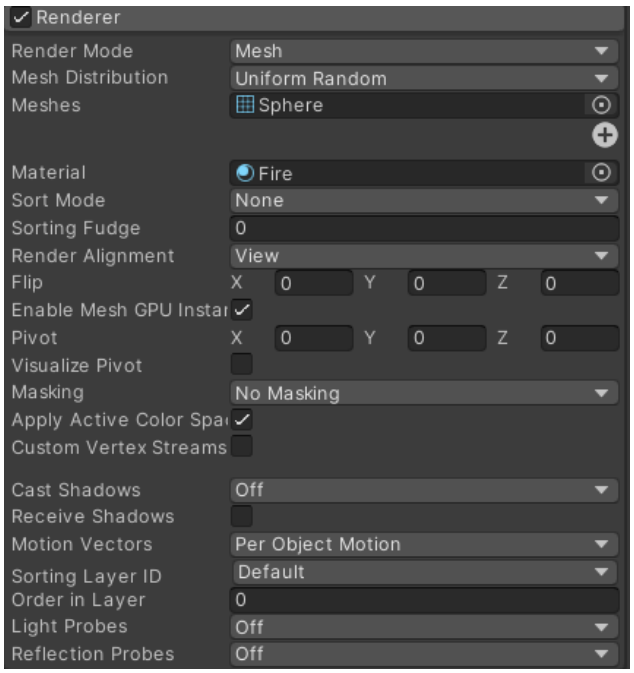


Set the shape settings to use the Donut shape, with Radius of 10 and doughnut Radius of 0.2 (1pt)



Toggle Size over Lifetime and set it to have a logarithmic download slope. (1pt)

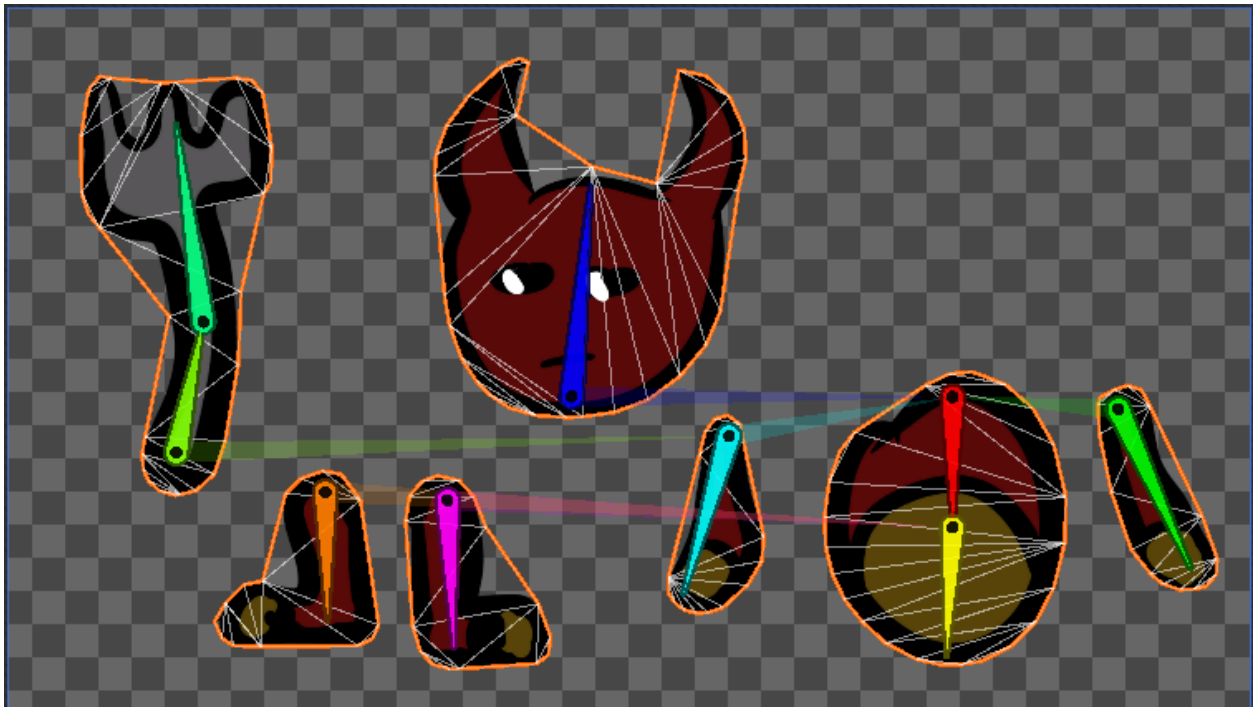
In the Renderer change the Render Mode to Mesh, change the Meshes to Spheres and set the Material to the Fire. (2pts)



You’re particle system should be making something like this:

Part E: Animation (5pt)

In your Sprite Folder you will find a Demon Sprite, we will create an animation for him.
Open up Sprite Editor and go to Skinning Editor, start adding bones from the body, connect the head and arms to the top body bone, and connect the legs to the bottom body bone. Connect the left arm to the pitchfork. Generate the weights Auto Geometry and your Character should looks like this (2pt)



Place the Demon Sprite in the Camera's view, add the Sprite Skin Component, generate the bones and make the demon stand like so (1pt)

Animate the demon so that he raise his hand and pitchfork and has the head grow and the goes back to his original stance like so (2pt)

Part 2: Game Screen (6pts)

Part A: Cookie (2pt)

In your sprite folder you will find a sprite that's called Cookie, make sure that its Texture Type is set to Cookie and the Alpha Source is set to From Gray Scale. (1pt)

Connect the Cookie to the Spotlight such that it looks like this (1pt)

Part B: Script (4pt)

In the start function make sure that the GUI text starts the game with the correct counter and the light button says Turn Off. (1pt)

```
// Start is called before the first frame update
Unity Message | 0 references
void Start()
{
    //Update the counterGUI text to display current count
    //Update the buttonGUI to say "Turn Off"
}
```

In the TurnLightOff function update the buttonGUI to say Turn On and Turn Off at appropriate times and update the counter GUI when the light is turned on. (2pt)

```
public void TurnLightOnOff()
{
    if (lighting.enabled)
    {
        lighting.enabled = false;
        //Update the buttonGUI to say "Turn On"
    }
    else
    {
        lighting.enabled = true;
        counter++;
        lighting.color = new Color(Random.Range(0,0.99f), Rando

        //Update the counterGUI text to display current count
        //Update the buttonGUI to say "Turn Off"
    }
    buttonSFX.Play();
}
```

In the goToNextLevel function use the SceneManager and the nextLevelName variable to go to the MainMenu and then use the buttonSFX to play the button noise when the function is called. (1pt)

```
0 references
public void goToNextLevel()
{
    //Use Scene Manager and the nextLevelName variable to Load into MainMenu
    //Use the buttonSFX to play when this function executes
}
```