

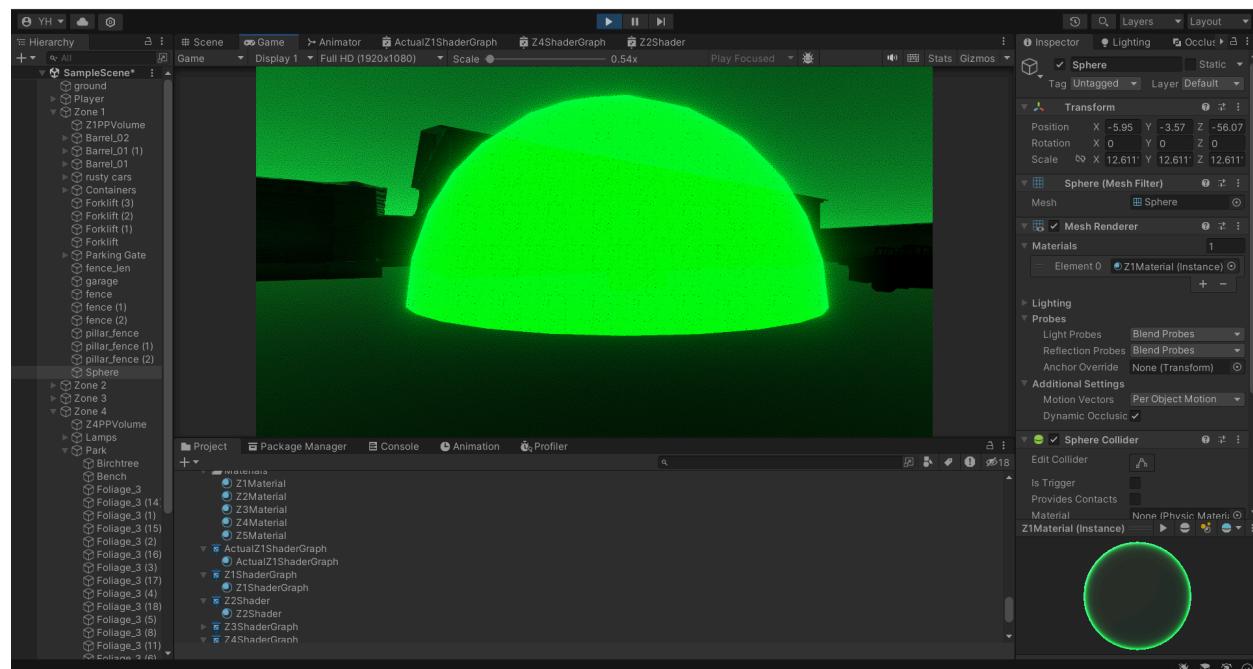
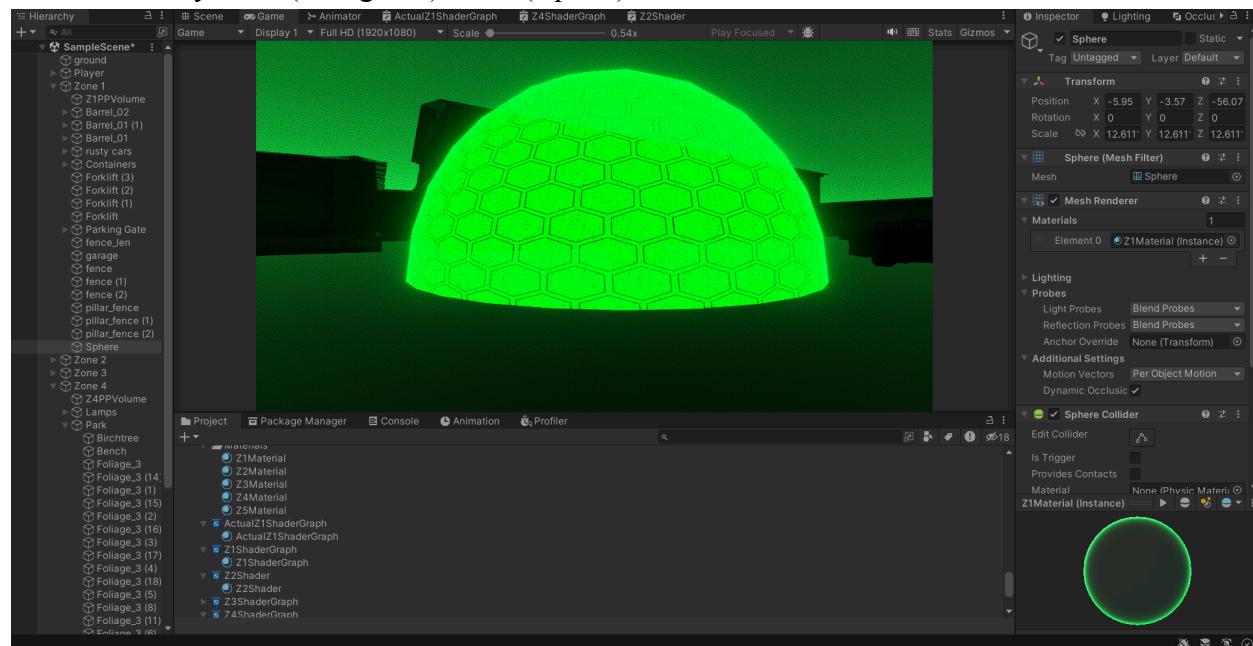
<b>Shaders</b>	<b>2</b>
Zone 1 Shader	2
Zone 2 Shader	5
Zone 3 Shader	8
Zone 4 Shader	10
Zone 5 Shader	12
<b>Particle Systems</b>	<b>14</b>
Zone 1 Particle System	14
Zone 2 Particle System	16
Experiment	17
Zone 3 Particle System	21
Zone 4 Particle System	23
Zone 5 Particle System	28
<b>Scanned Model</b>	<b>30</b>

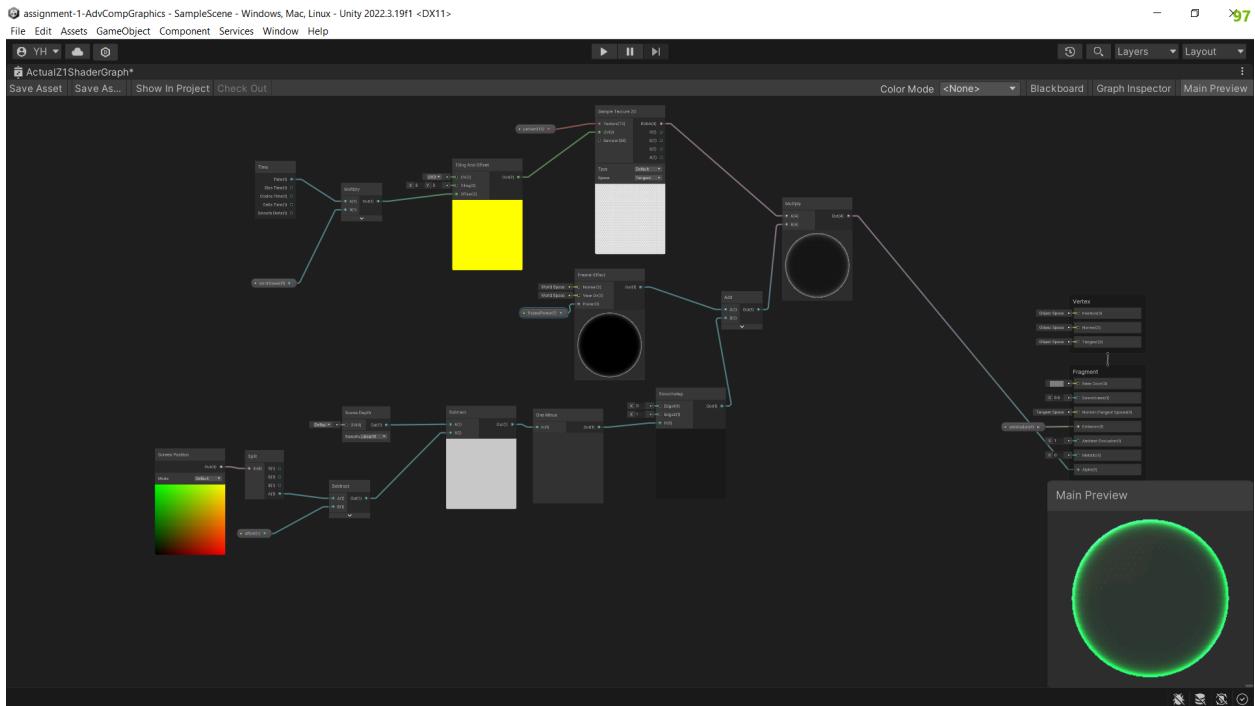
# Shaders

This section covers the 5 different shaders applied in each zone

## Zone 1 Shader

This shader is simulating a force field effect. It has a scrolling effect and the tiling on the shader is controlled by the X(hexagonal) and Z(square) buttons.





- **Graph Description**

- **Variables:**

- **Offset**

- Float value
      - Set to 0.6
      - Helps in controlling overlapping with the object the shader is applied on

- **Emmission**

- HDR Color
      - Set to (0,191,7) with 4.2 Intensity
      - Handles the color of the shader

- **Fresnel Power**

- Float
      - Set to 5
      - Handles the intensity of the fresnel effect(glow)

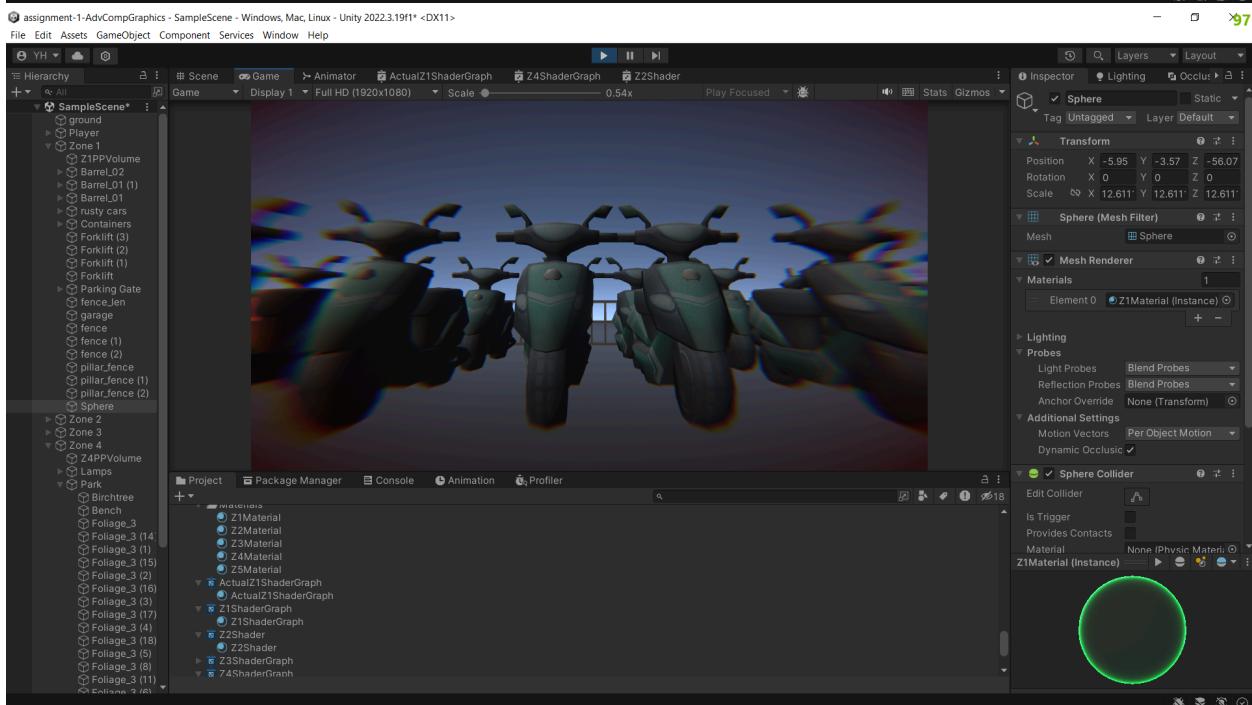
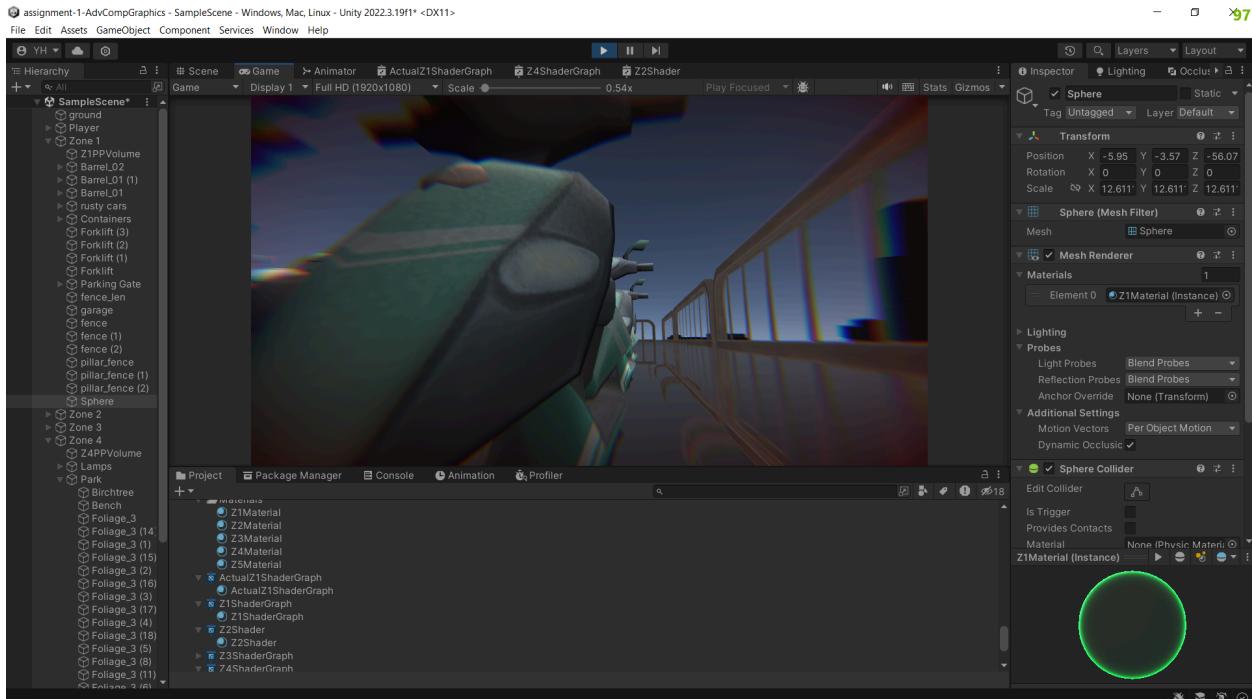
- **Pattern**

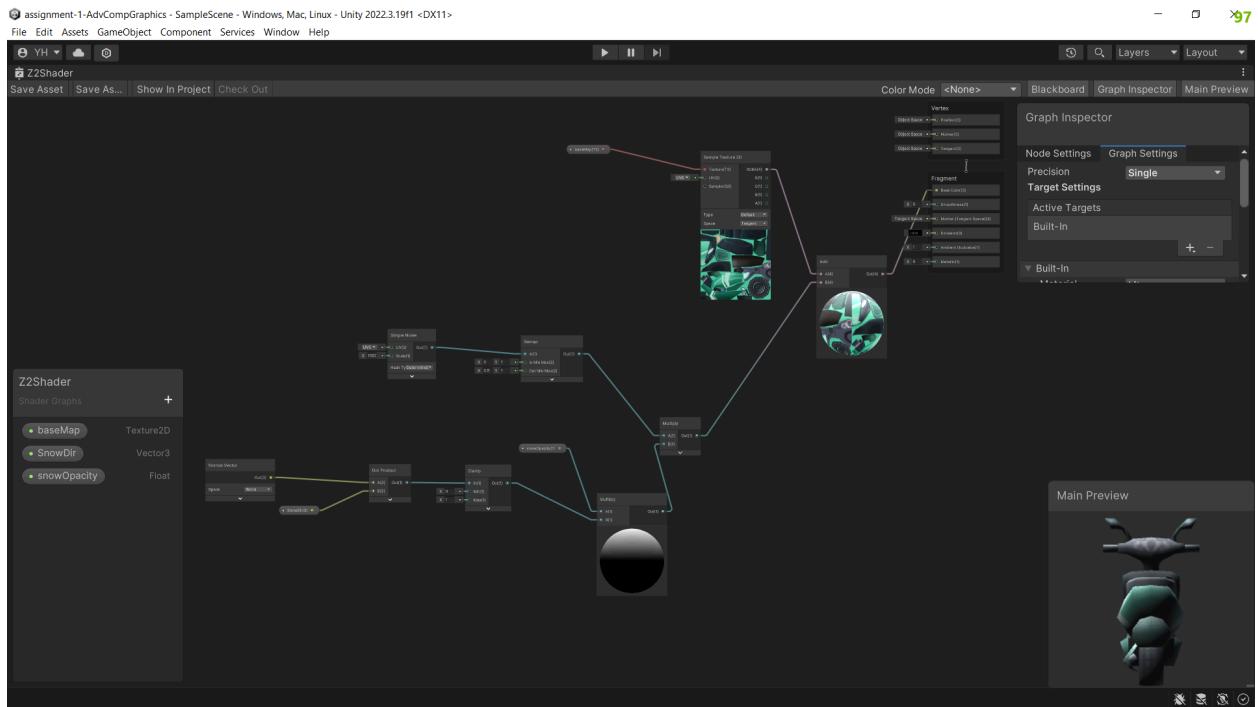
- Texture2D
      - Set to hexagon by default
      - Handles the pattern on the object, changed by X and Z keys

- Scroll Speed
  - Float
  - Set to 0.1
  - Handles the speed of the scroll effect on the pattern of the object
- Node Description:
  - Overlap Detection:
    - This is used to apply a glow effect to the edges of the object while taking into consideration the intersection of the object with other objects
    - In this section of the graph, we:
      - Retrieve the scene position
      - Split it into RGBA Values
      - Subtract the offset from our Alpha value
      - Subtract the scene depth from the output of the previous step
      - Invert the previous output using the one minus node and pass it to a step node
      - Add the output of the step node to a Fresnel Node that takes fresnalPower as the power input
      - Multiply this final output with the output from the texture application section to get the final output of the shader and connect it to the Alpha Channel of the Fragment Node
  - Texture Application:
    - This section applies the scrolling effect as well as the texture of the object
    - In this section of the graph, we:
      - Get a time node and multiply the time output with the scrollSpeed
      - Take the previous output and connect it to the offset input of a Tiling and Offset node with tiling set to (3, 3)
      - Take the previous output and connect it to a sample texture 2D node's UV input with our pattern variable as the Texture input
      - Multiply the previous output with the output of the overlap detection section and connect it to the Alpha Channel of the Fragment Node

# Zone 2 Shader

This shader applies a type of low-poly snow effect on an object





- Graph Description:

- Variables:

- SnowDir:

- Vector 3
      - (0,1,0)
      - Handles direction at which snow “falls” (Y-axis by default)

- SnowOpacity:

- Float
      - 1
      - Handles the visibility of the snow effect

- BaseMap:

- Texture2D
      - Set to the base texture map of the object you want
      - Holds the base texture map of the object

- Node Description:

- The output of the addition the following section is passed to the base color in the Fragment Node

- Texture Application:

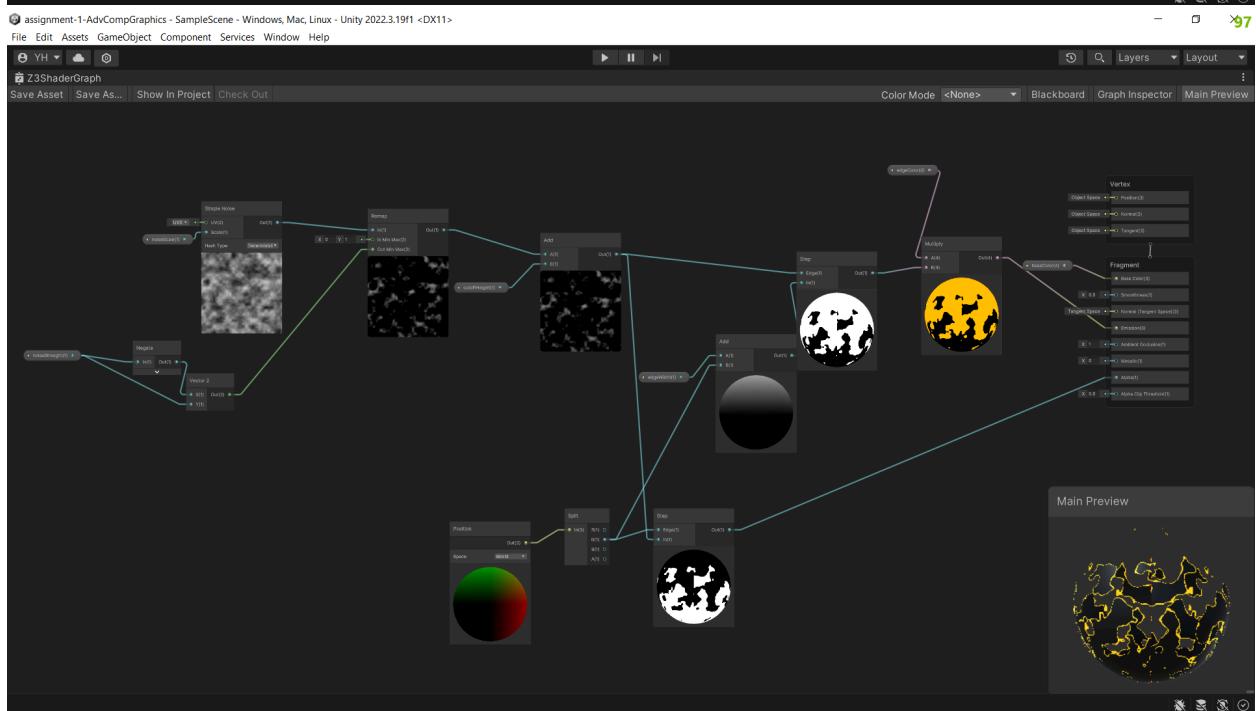
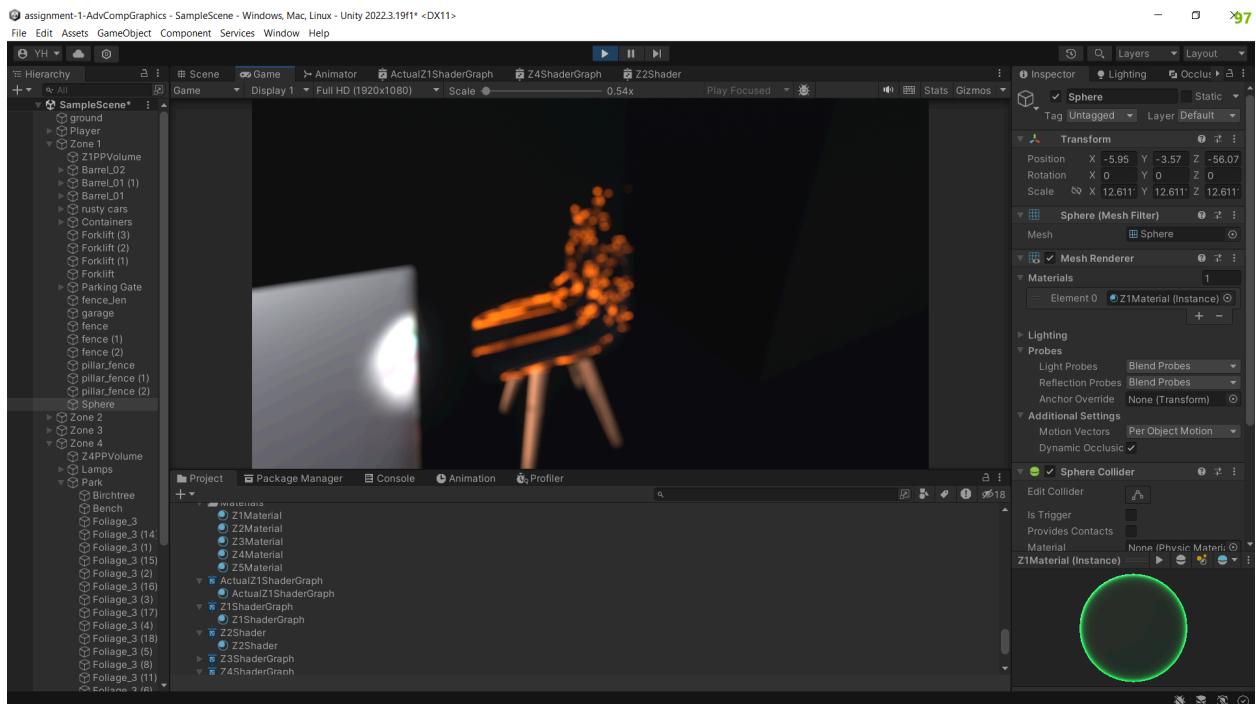
- This section preserves the original texture of the object.
      - In this section, we:
        - Pass the base map variable to a sample texture 2D node.
        - Add the RGBA value to the output of the effect application section

- Effect Application:

- This section applies snow to the object
- In this section, we:
  - Get dot product of the normals with the snow dir variable
  - Clamp it to be between 0 and 1
  - Multiply the value by the snow opacity
  - Multiply the value by a simple noise node with scale 1000 and remapped with Out Min Max of (0.5, 1)
  - The final value is added to the texture application output

# Zone 3 Shader

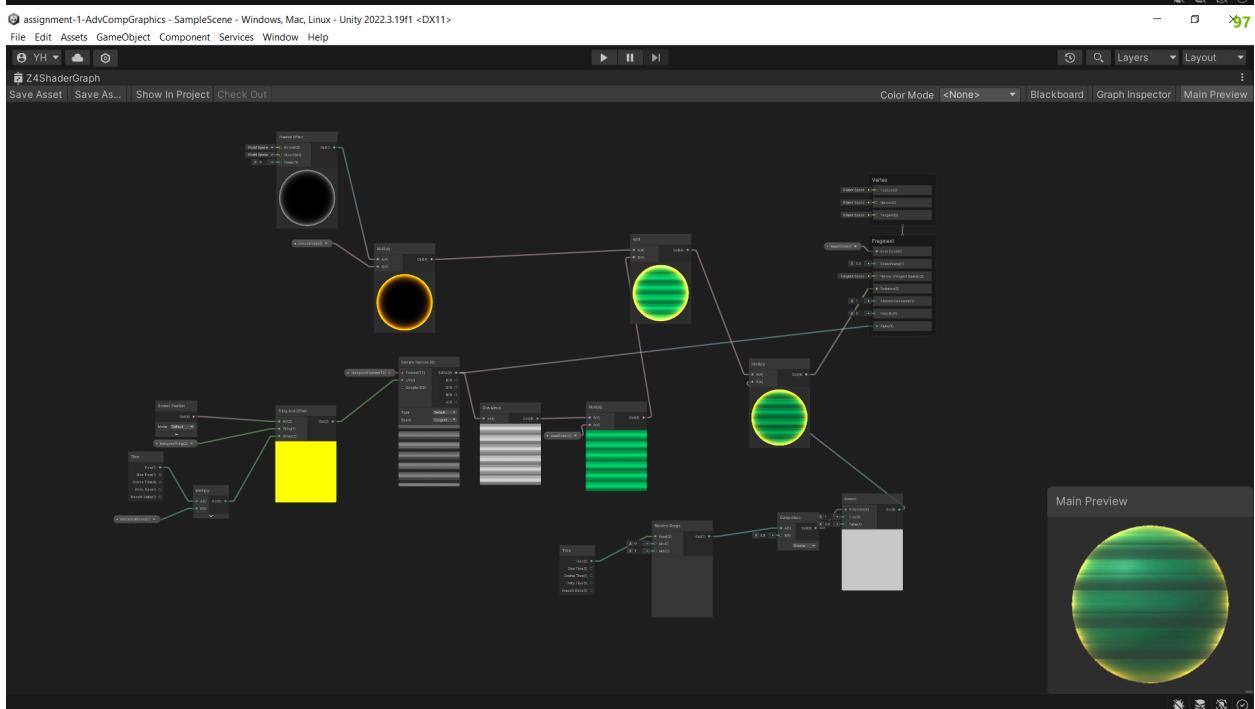
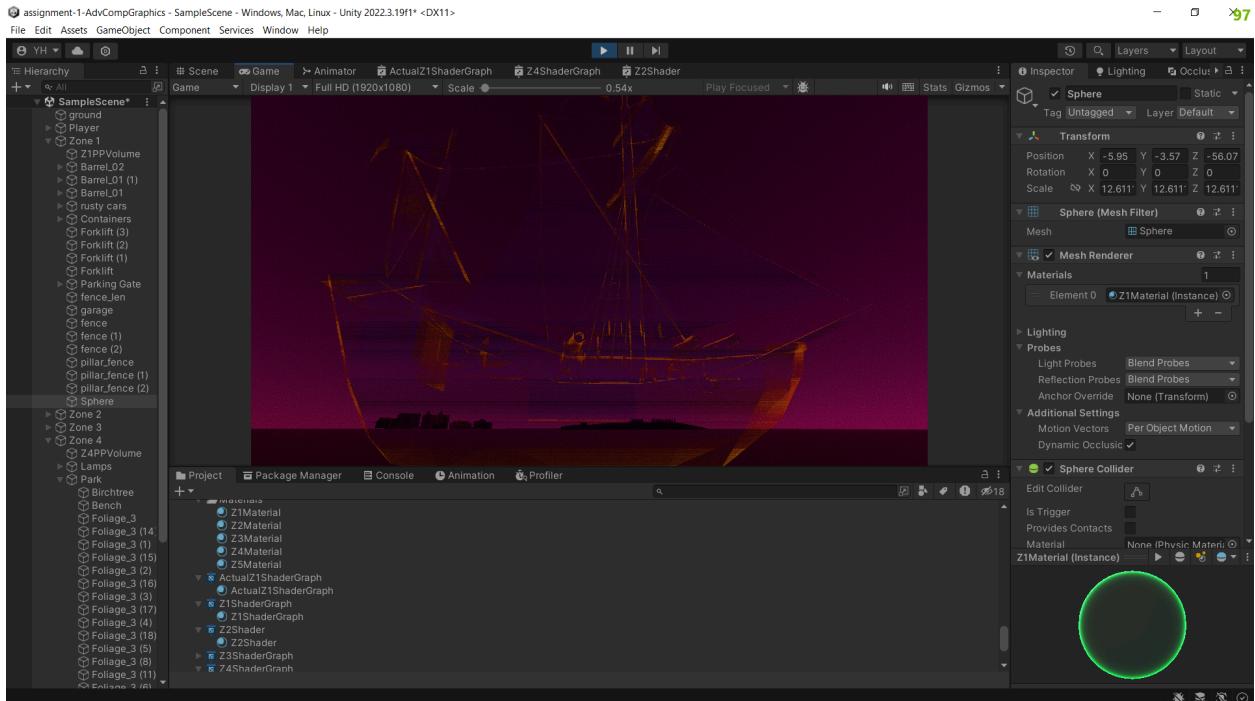
This shader adds a dissolving effect to an object



- Graph Description:
  - The most significant nodes for this graph are the noise and step nodes.
    - For the noise:
      - It controls the specific pattern in which the object dissolves
    - For the step:
      - It controls the dissolution of the object itself
  - When these nodes are arranged and funneled into the shape shown in the image, the result is a dissolving effect

# Zone 4 Shader

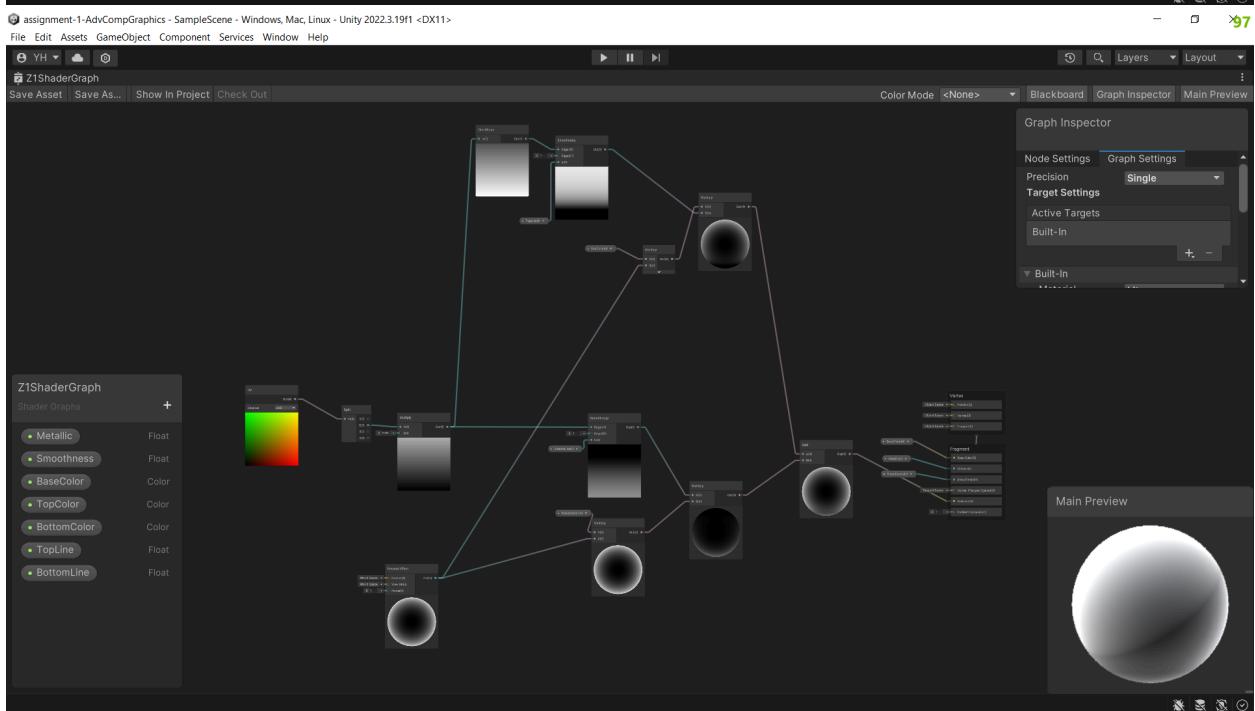
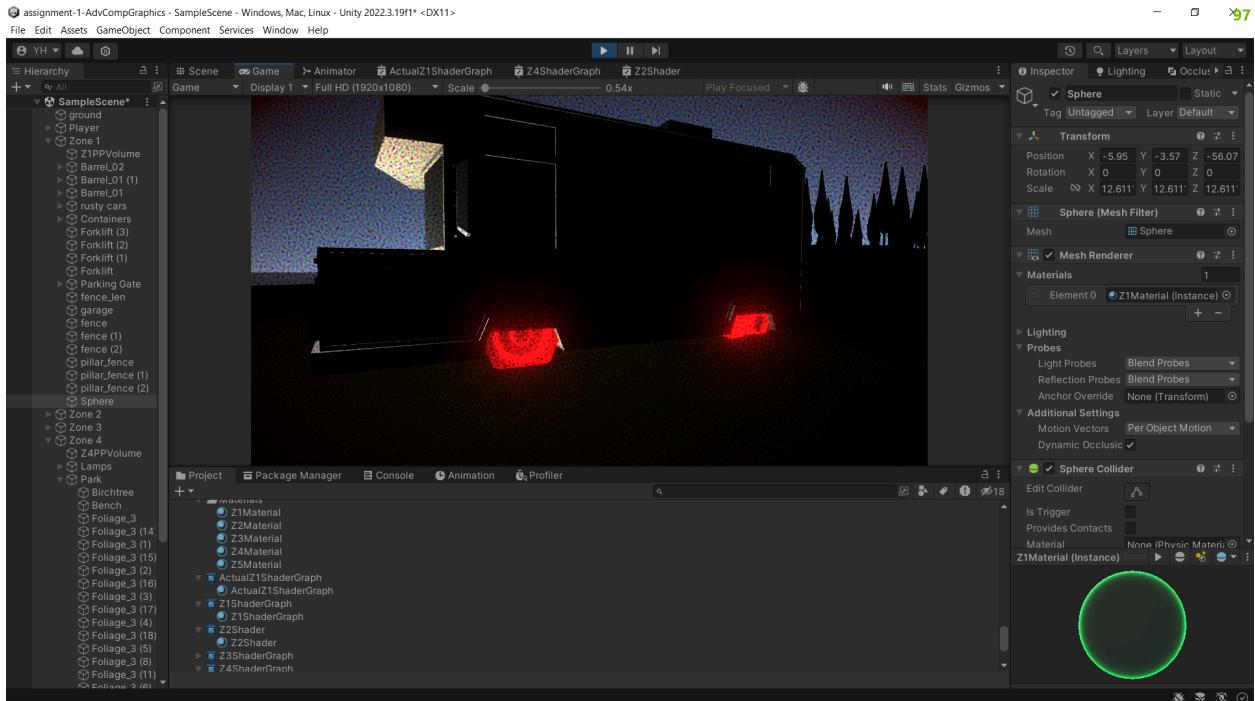
This shader applies a holographic effect to the object



- Graph Description:
  - Significant nodes:
    - Random Range:
      - Causes the effect to flicker according to the inputs in the image
    - Tiling and Offset
      - Tiles the texture and applies a scroll effect on the texture
    - Fresnel Effect
      - Applies glow to the edges of the effect
    - Time
      - Used multiple times to control both the random range and the scroll speed of the effect

# Zone 5 Shader

This effect applies a gradient glow to an object.



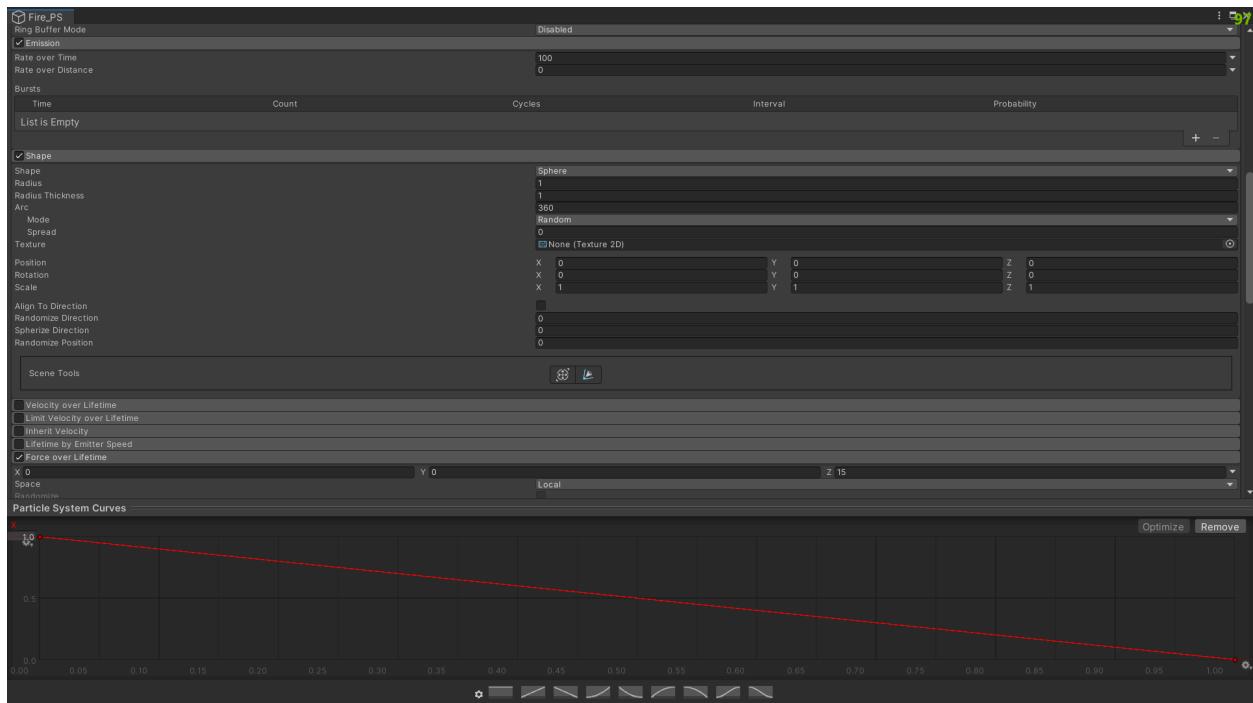
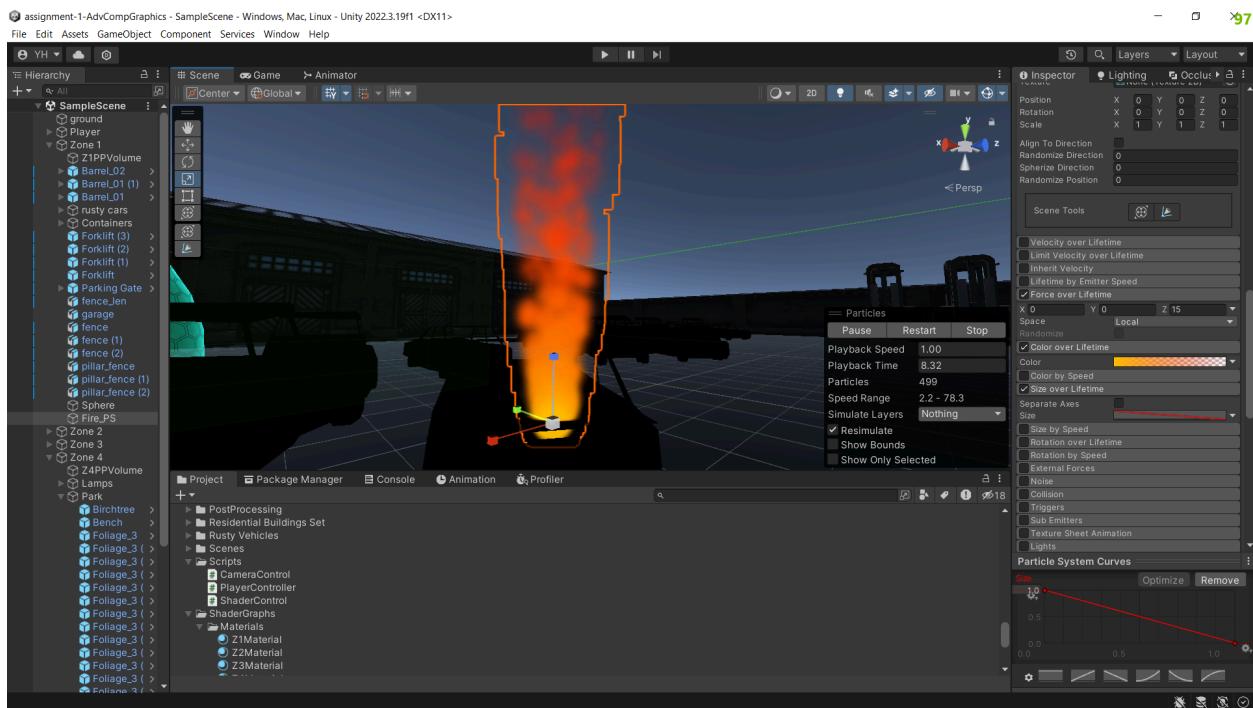
- Graph Description:
  - Significant Nodes:
    - Fresnel Effect:
      - Applies glow to the effect
    - UV:
      - Gets colors of the object
    - SmoothStep
      - Applies gradient to effect

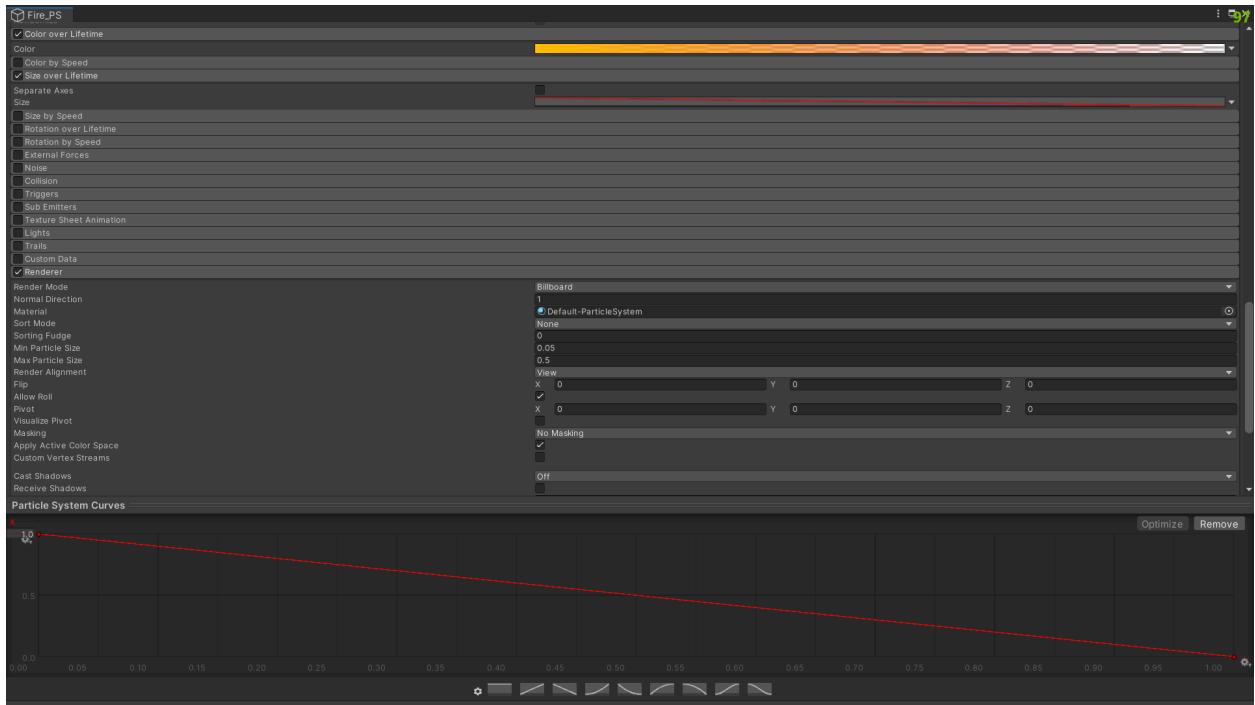
# Particle Systems

This section covers the particle systems in the different zones

## Zone 1 Particle System

This particle system simulates fire.



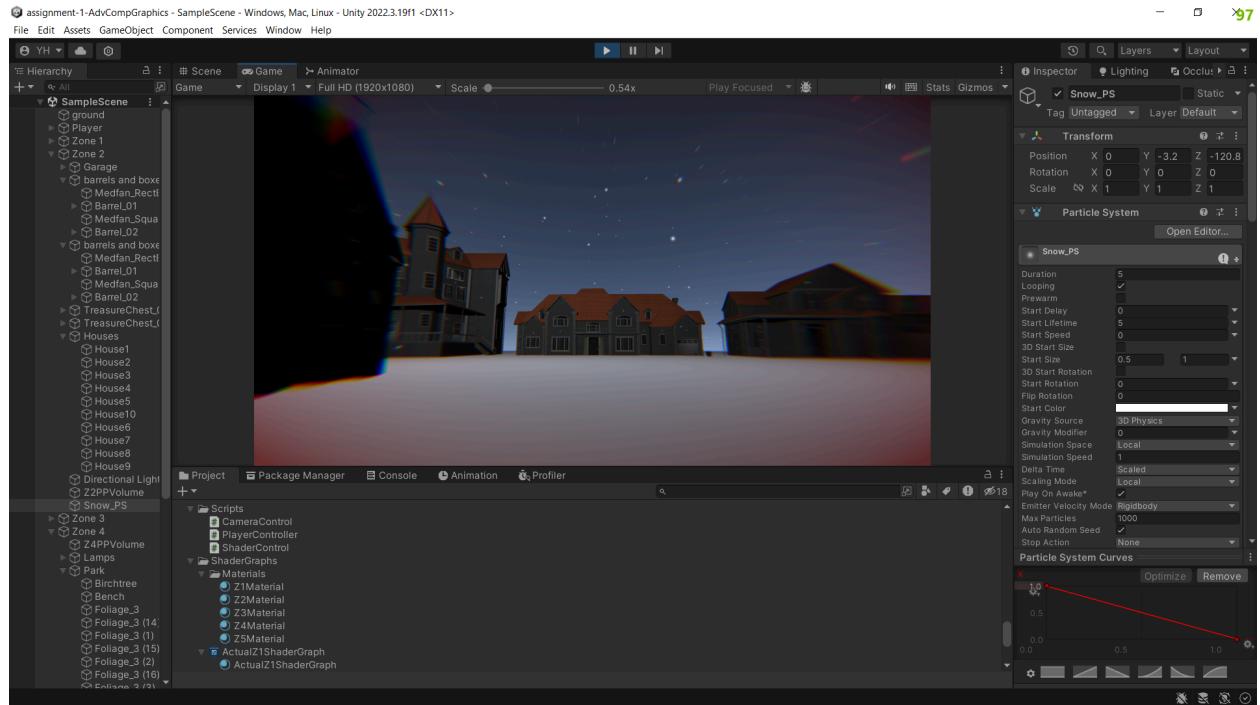


The previous images include the particle system as well as the changes applied to the parameters. Some of the parameters are for added effect but the main ones are:

- Color Over Lifetime:
  - Fading gradient from Yellow to Red
- Shape:
  - Sphere
- Emission:
  - Rate Over Time = 100

# Zone 2 Particle System

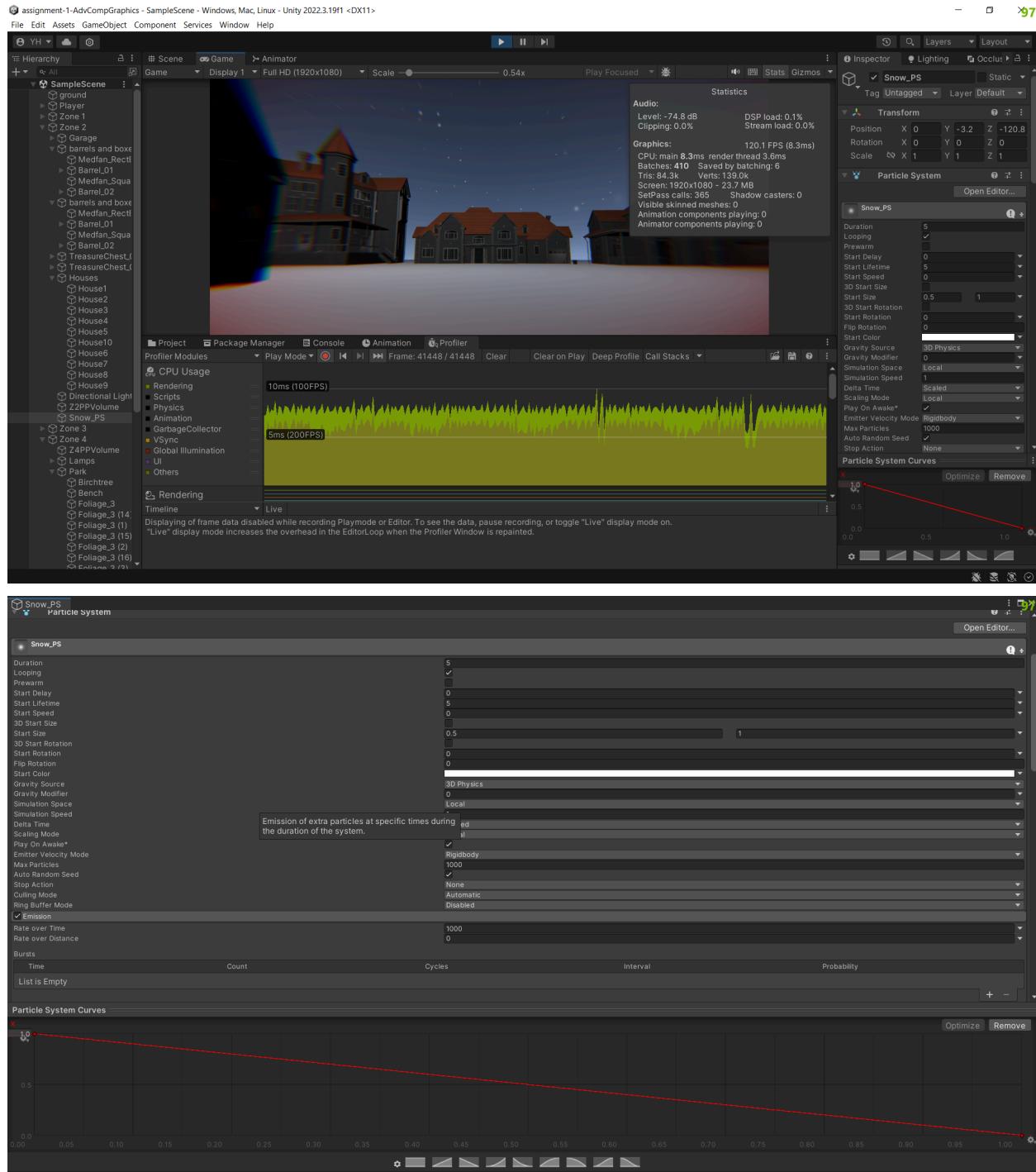
This particle system simulates a snow effect

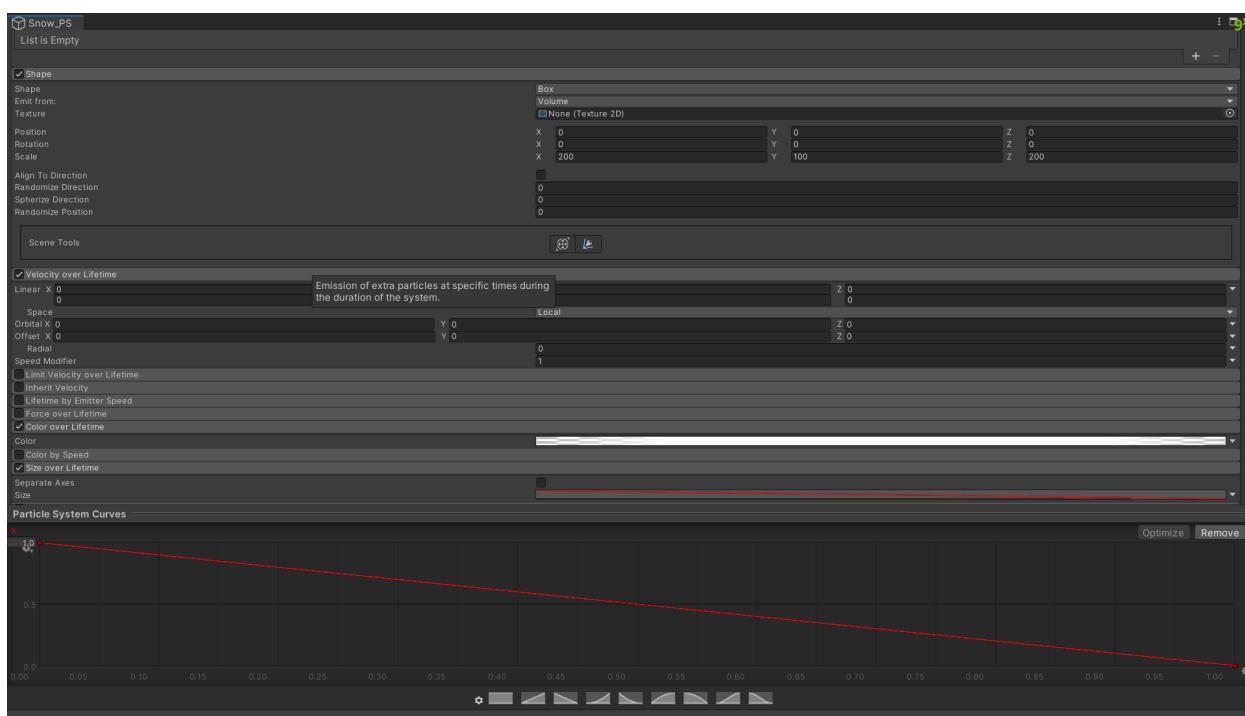
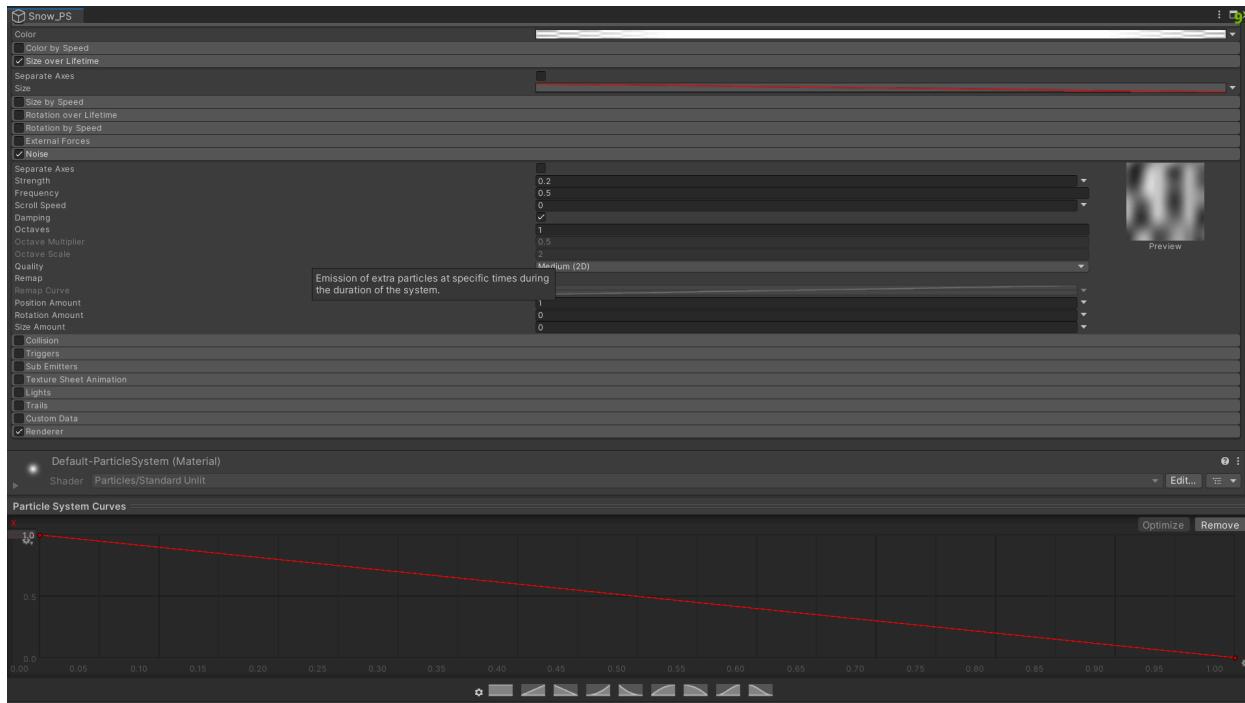


# Experiment

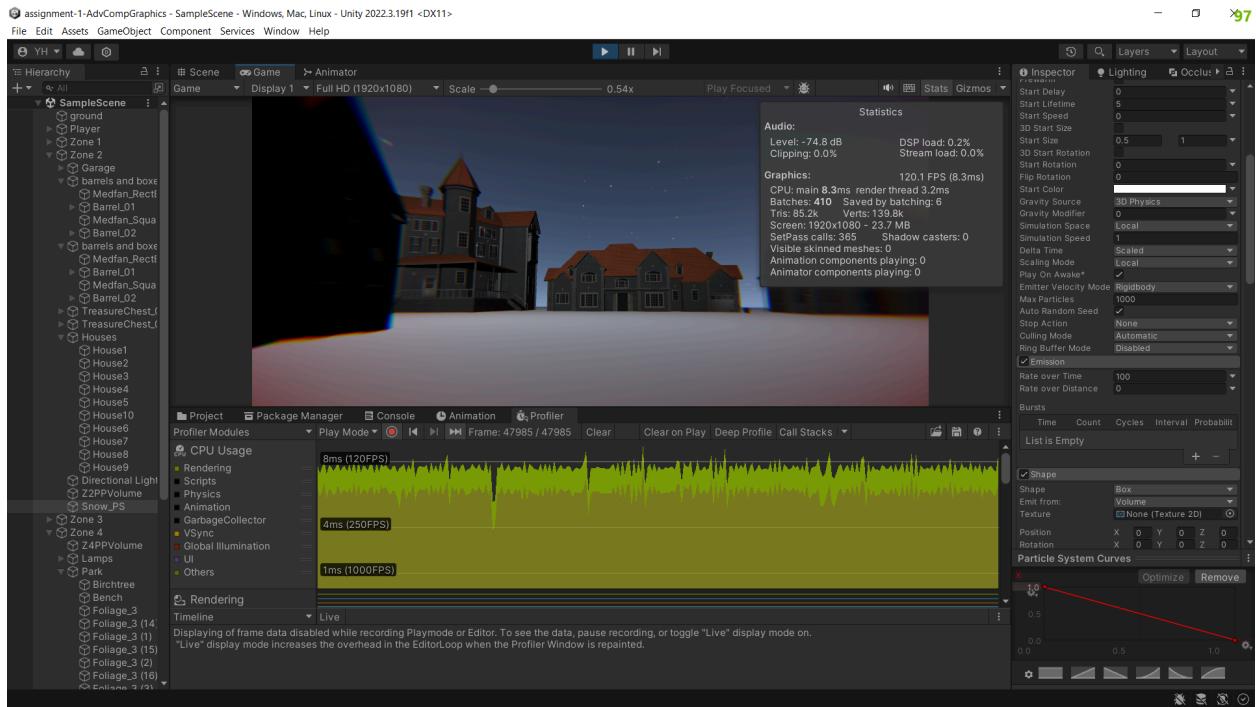
Here, we test parameter tuning as a form of optimization for particle systems

## Baseline Metrics and Initial Settings:

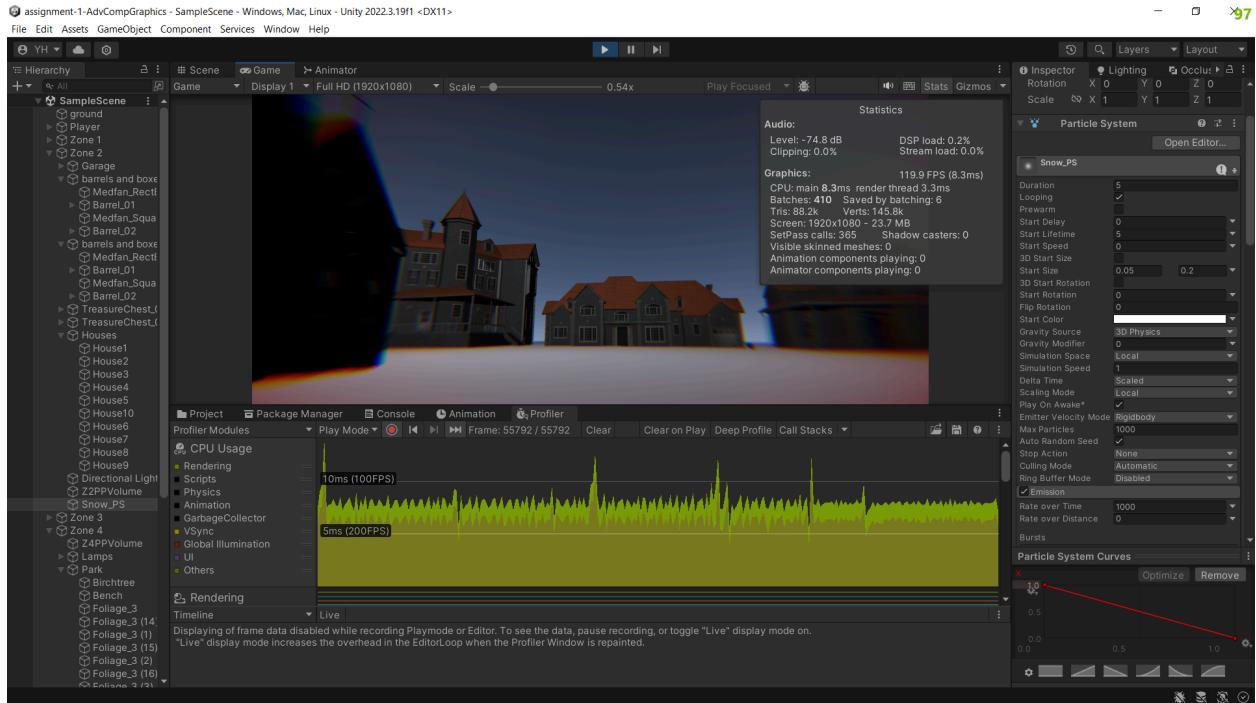




## Performance after setting Emission Rate Over Time to 100 instead of 1000:



## Performance after changing particle size to be smaller(0.05-0.2 instead of 0.5-1.0)

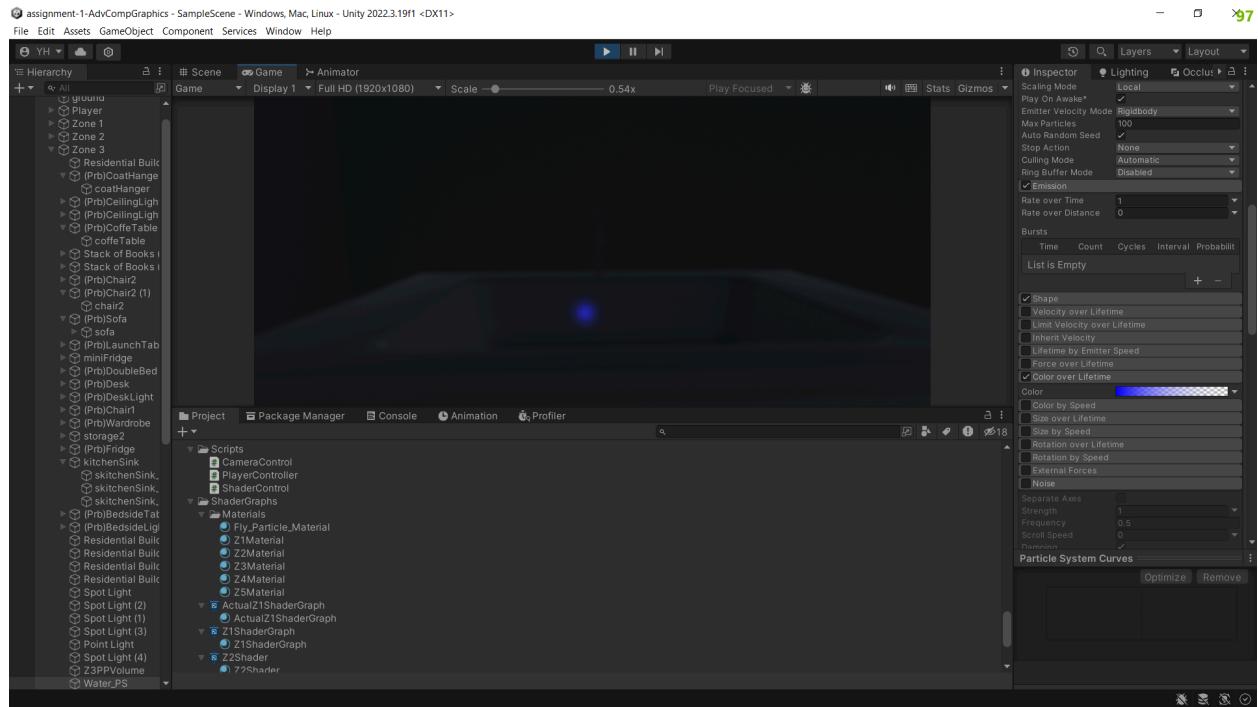


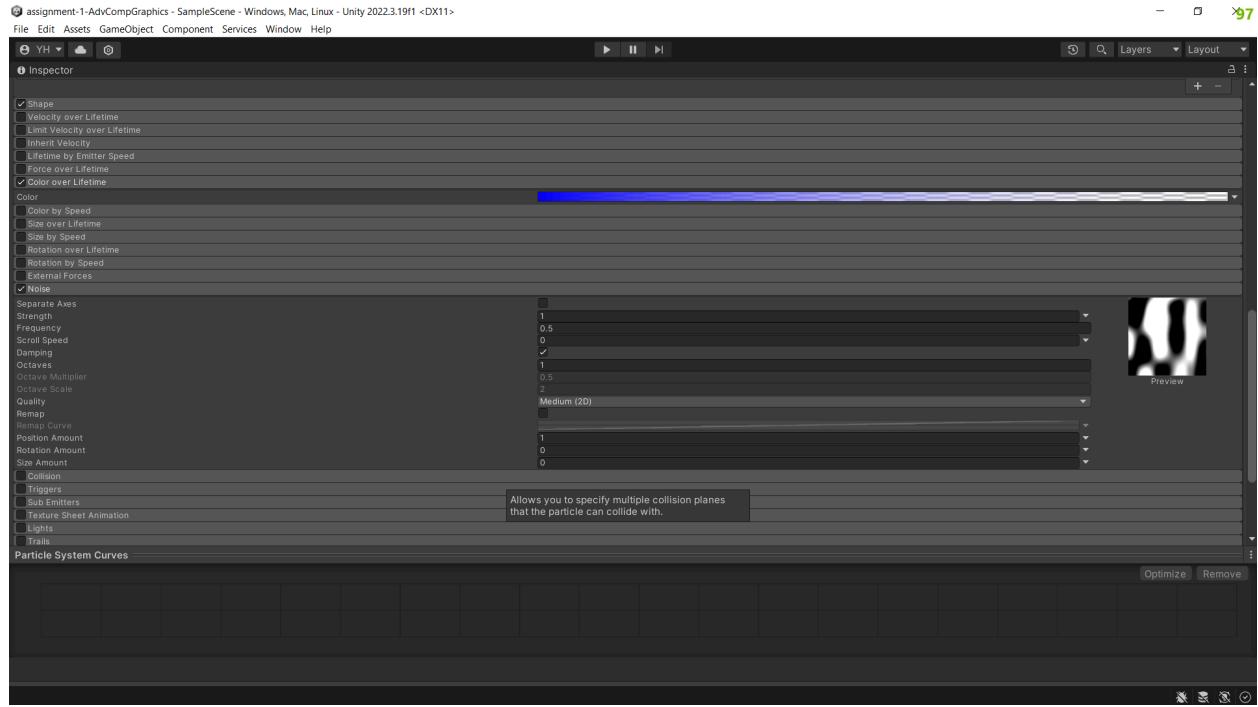
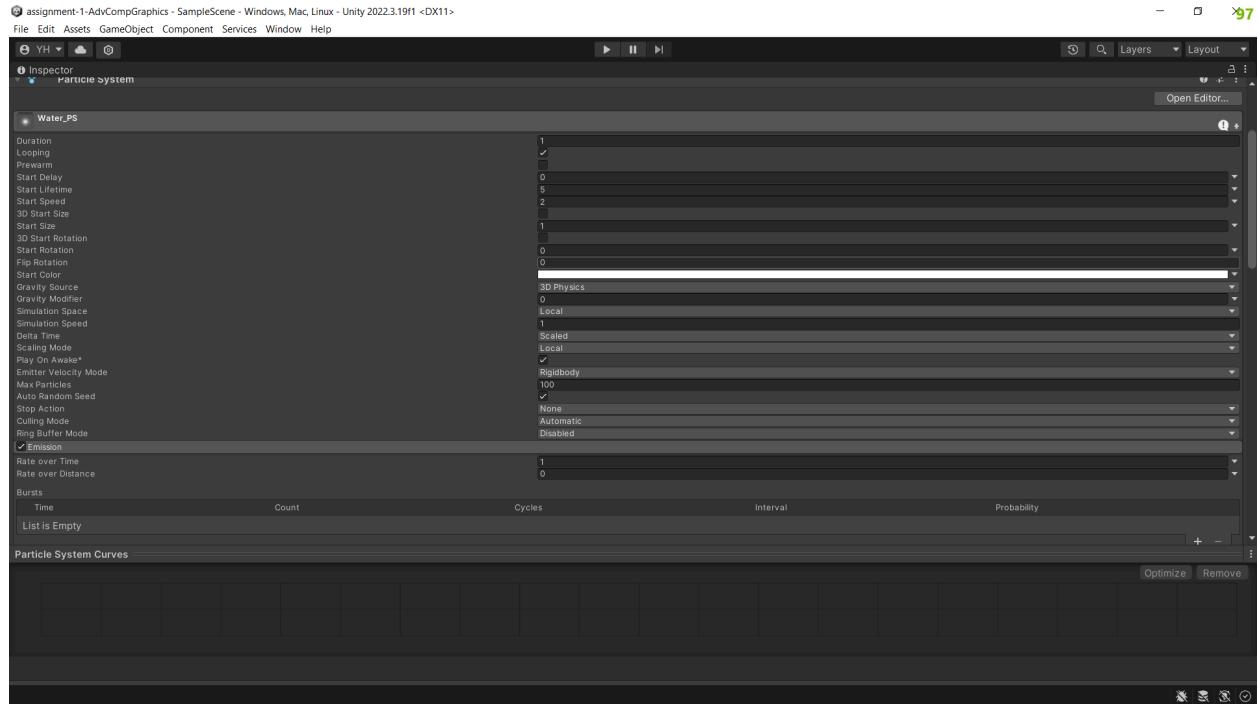
Conclusion:

- The best-performing and most stable technique is lowering the amount of particles being emitted per burst
- This, however, may not be applicable if the scene requires a certain rate of particles to be on screen at a time

# Zone 3 Particle System

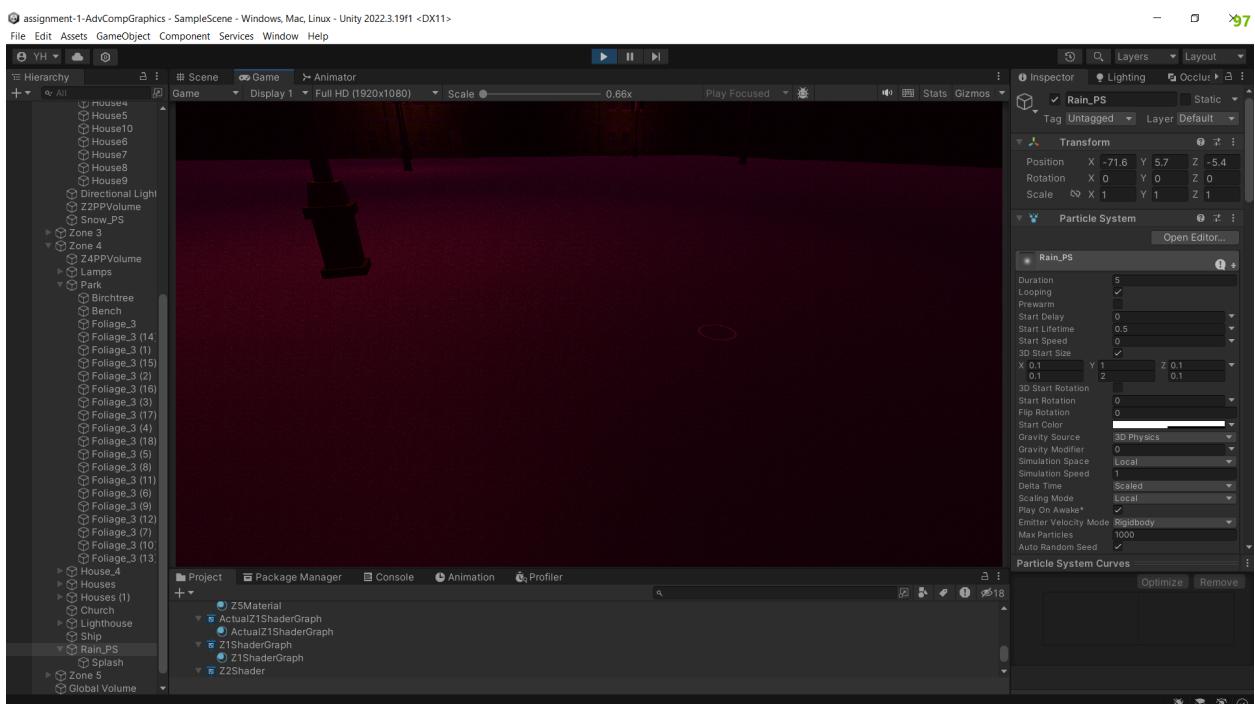
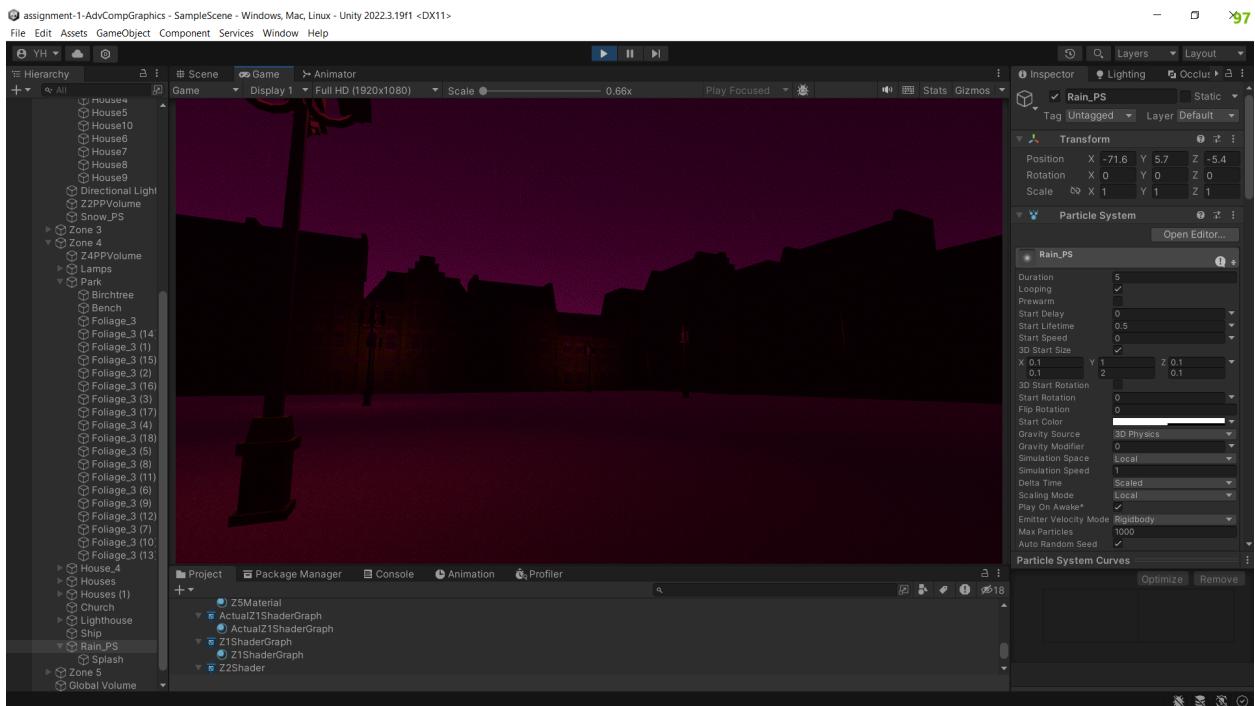
This particle system simulates a leaky faucet



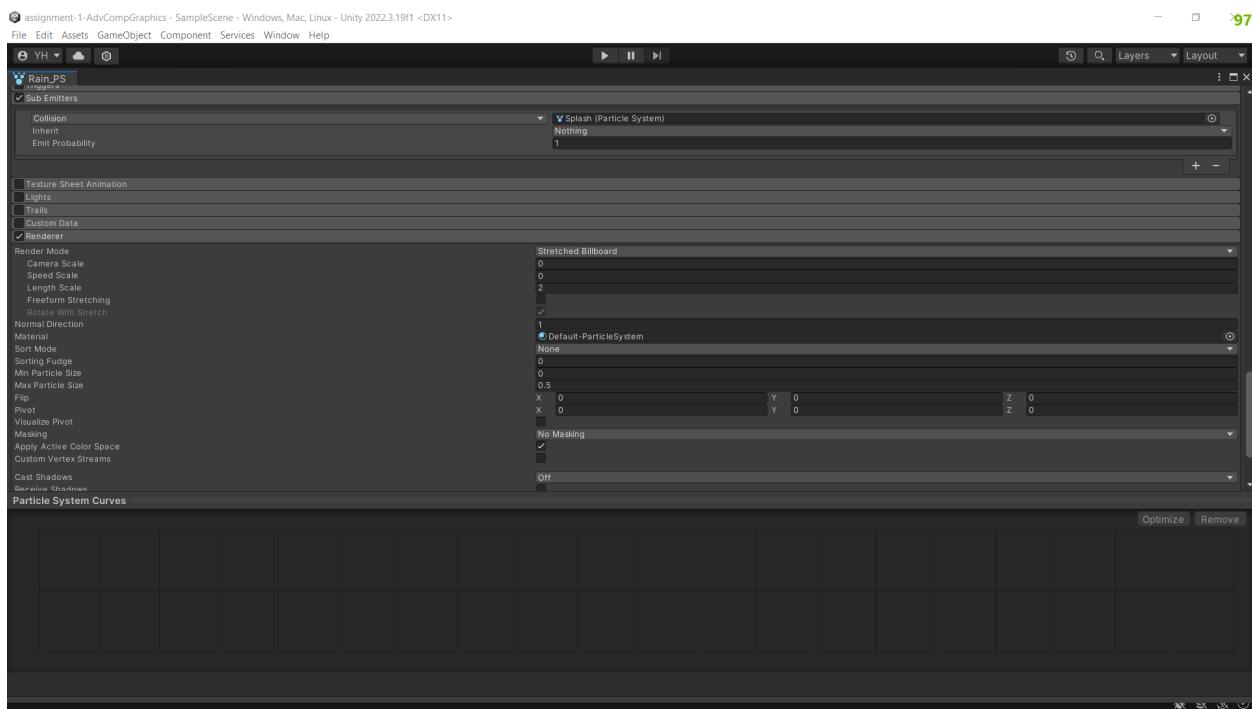
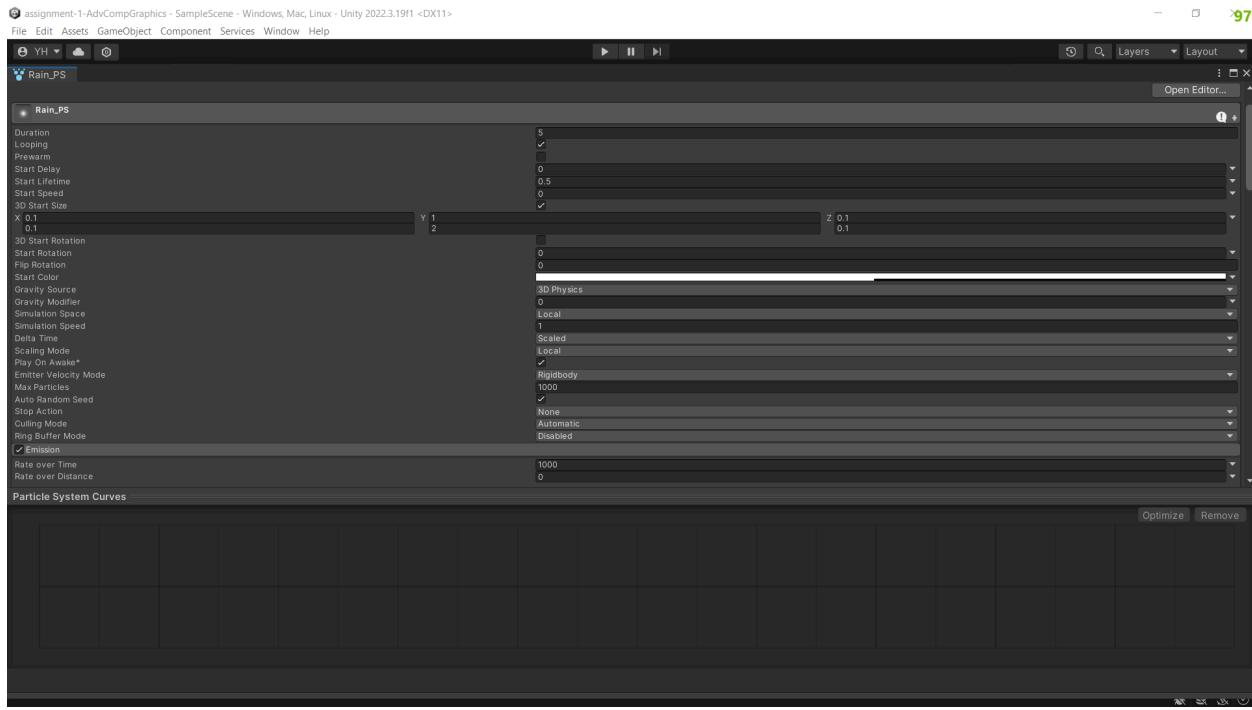


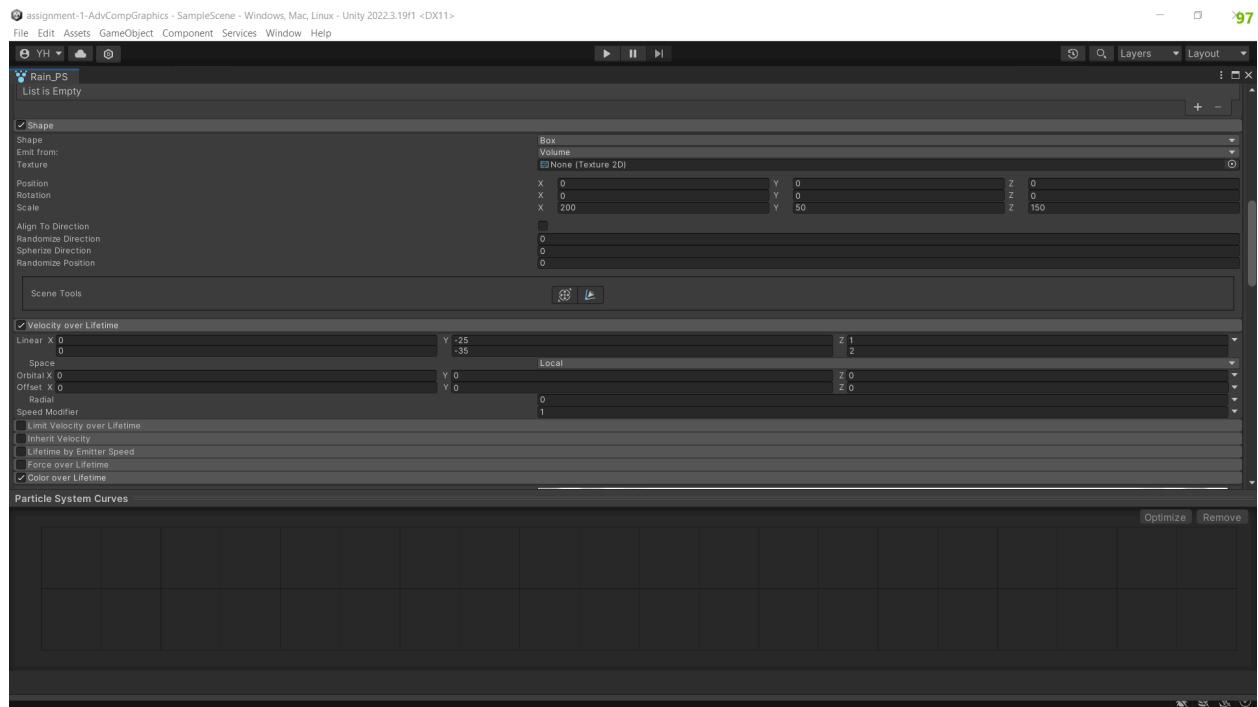
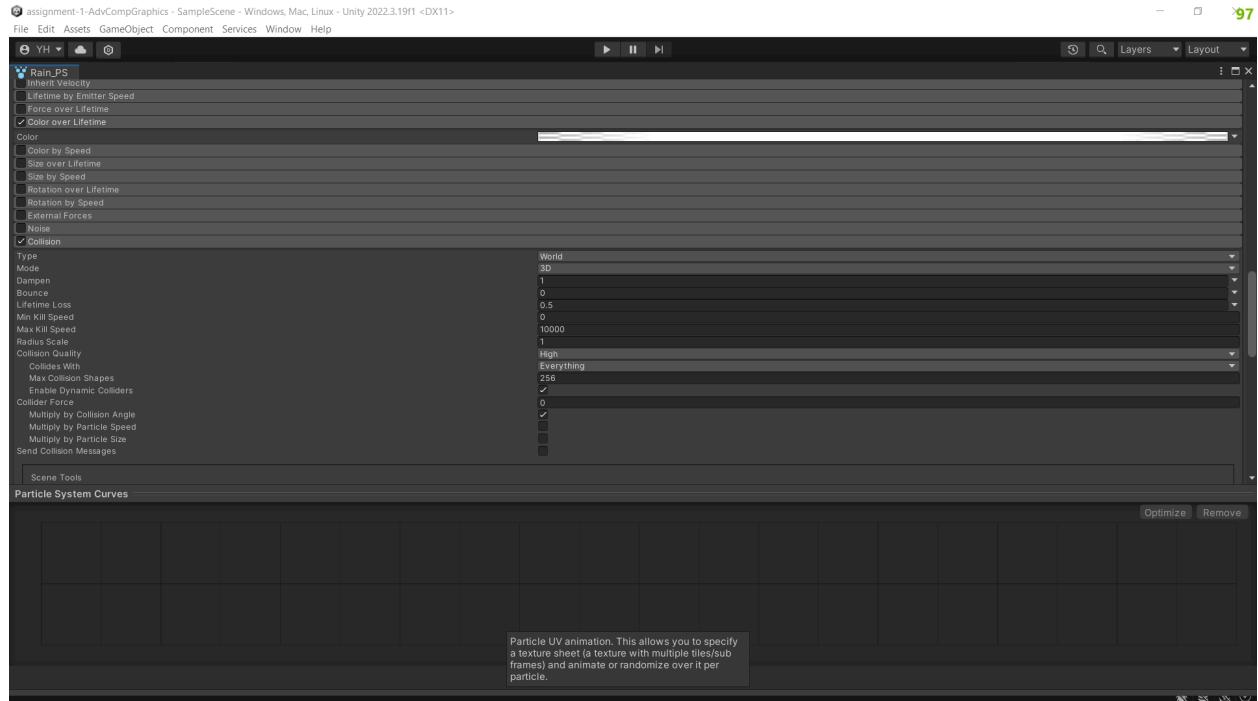
# Zone 4 Particle System

This particle system simulates rain with a sub-emitter for splashes

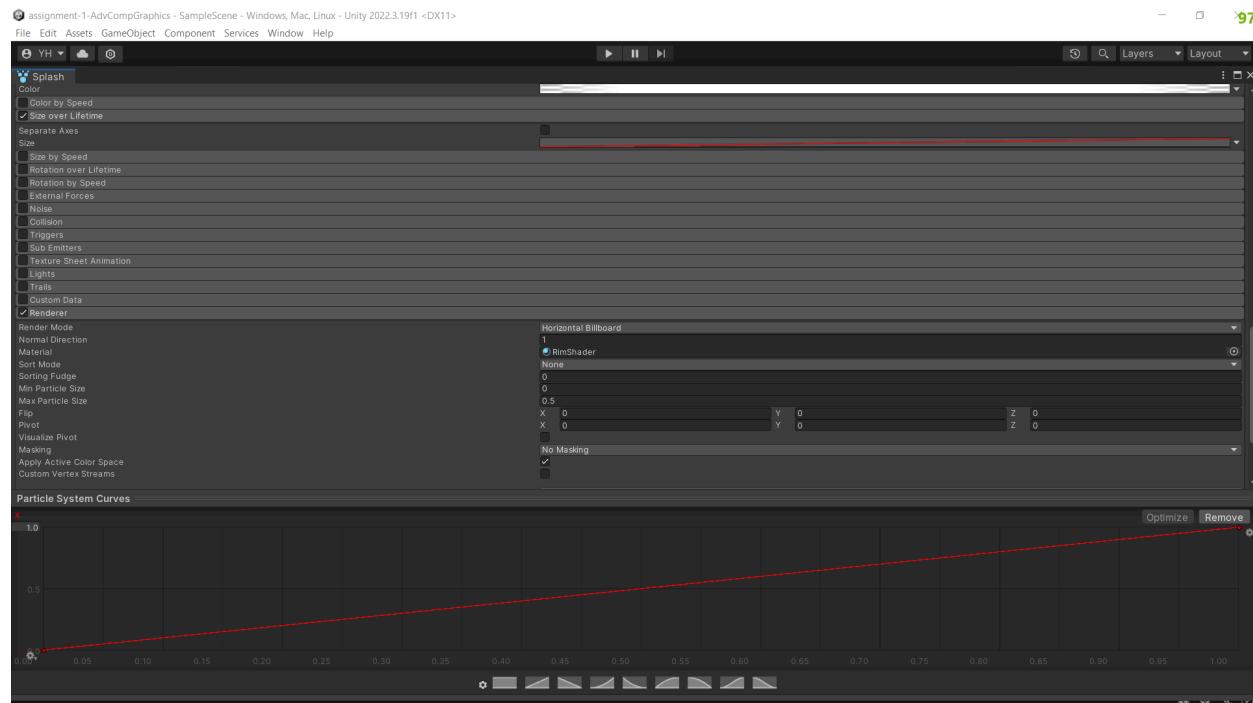
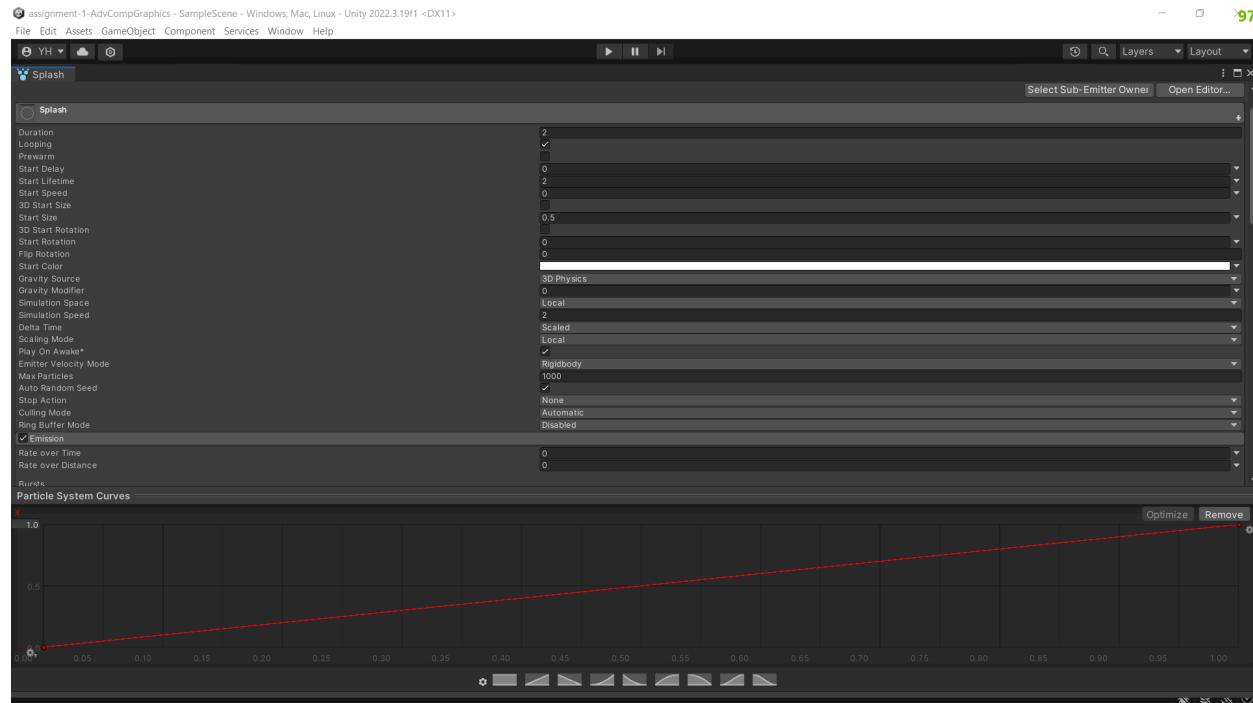


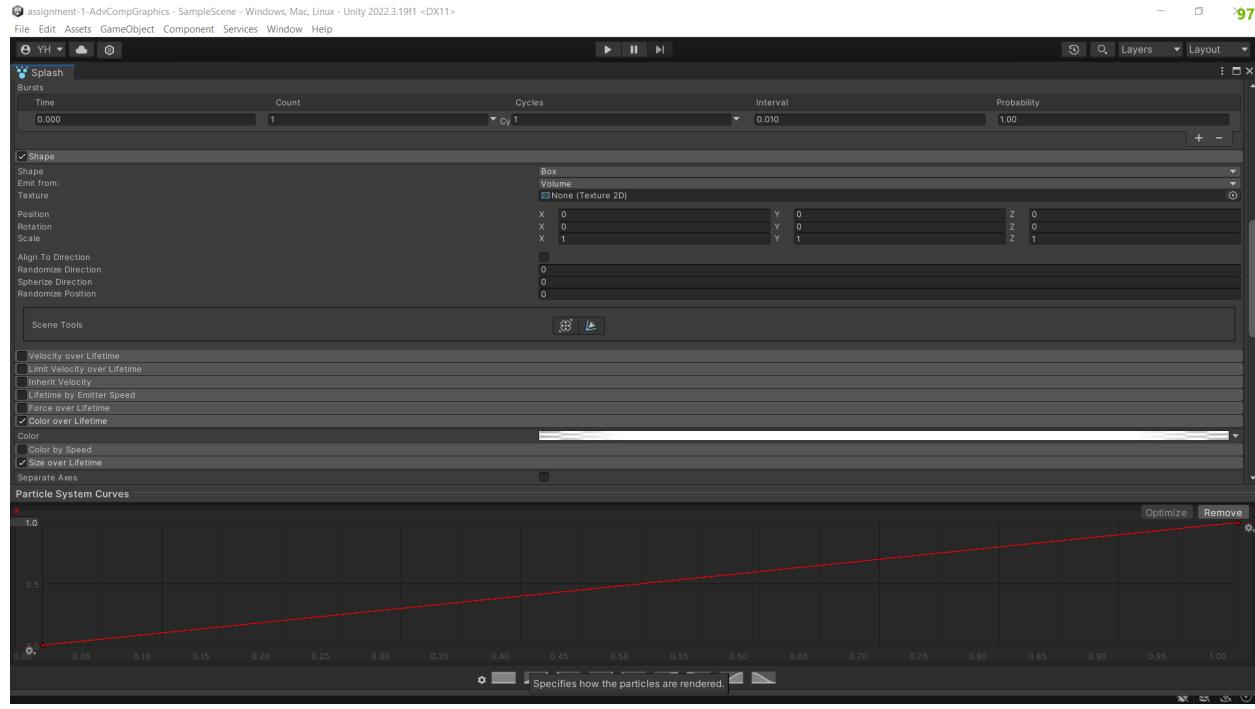
## Settings for Main Particle System





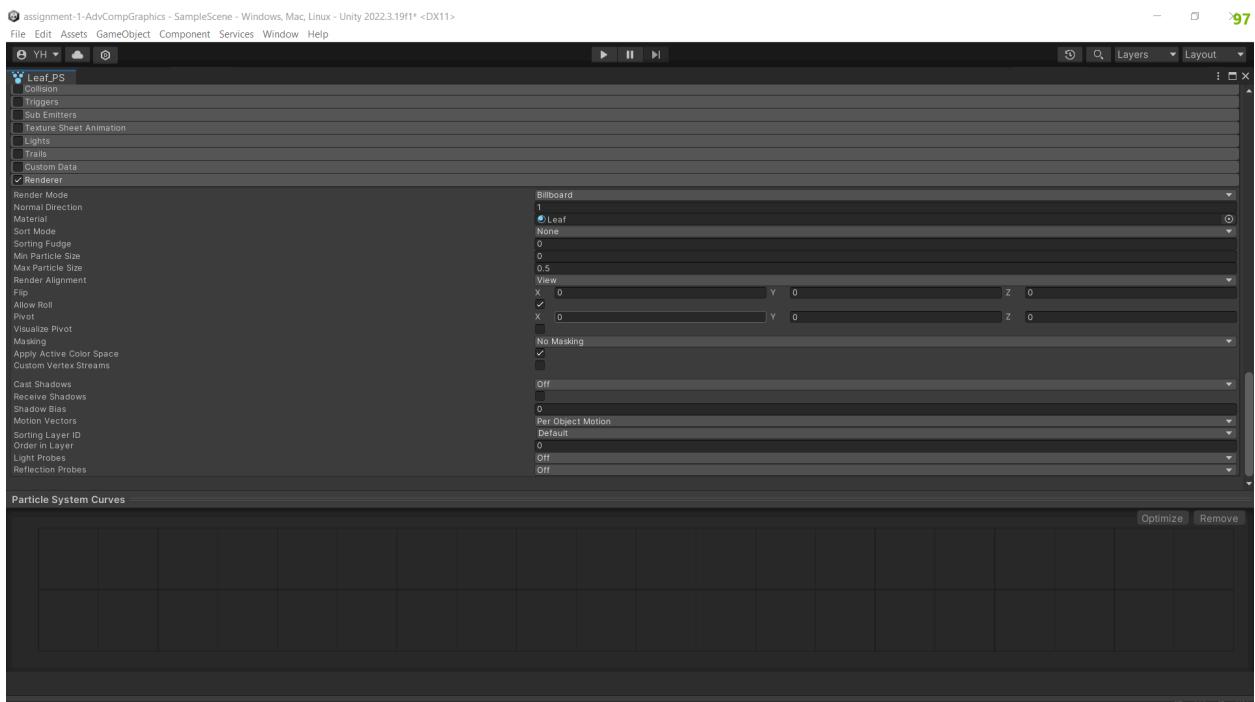
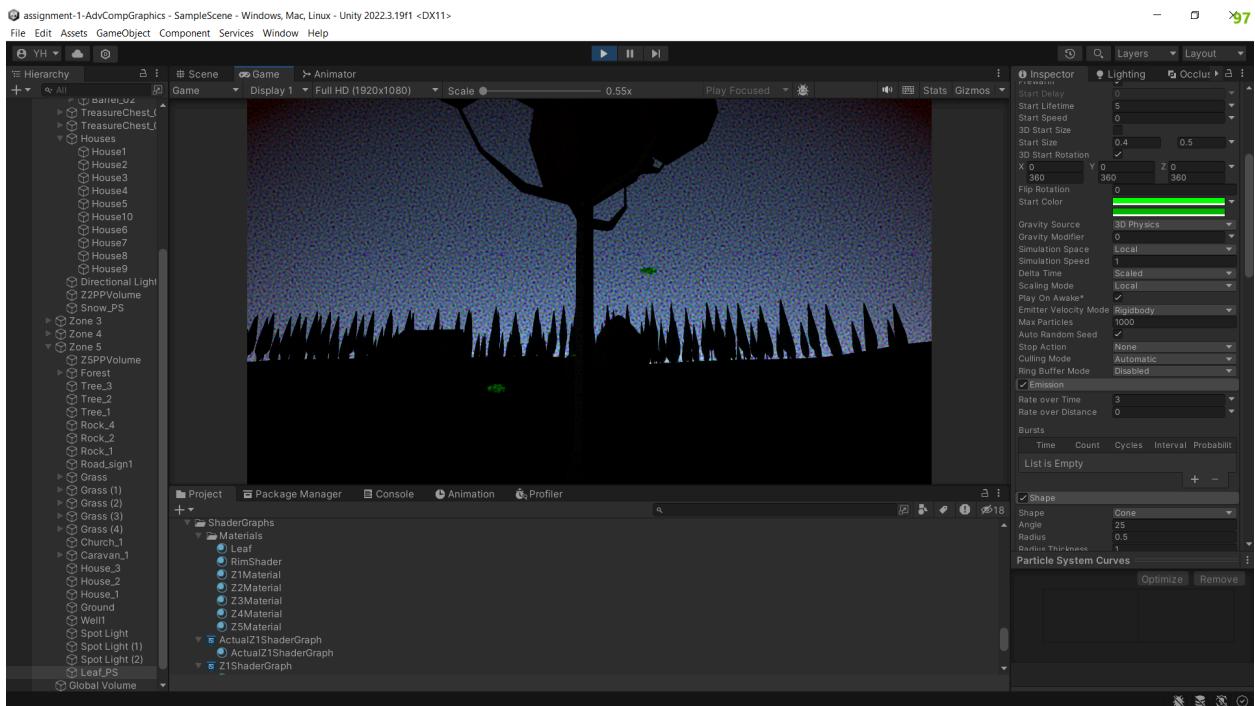
## Settings for Sub-emitter Particle System:

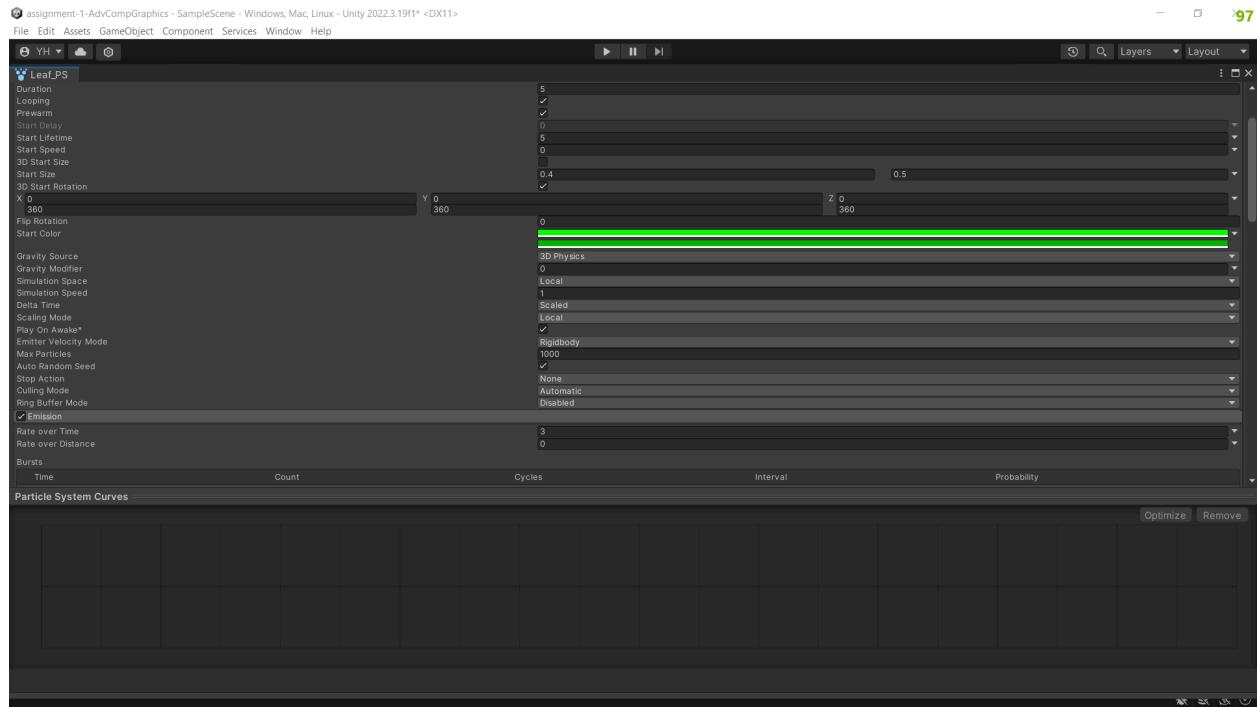
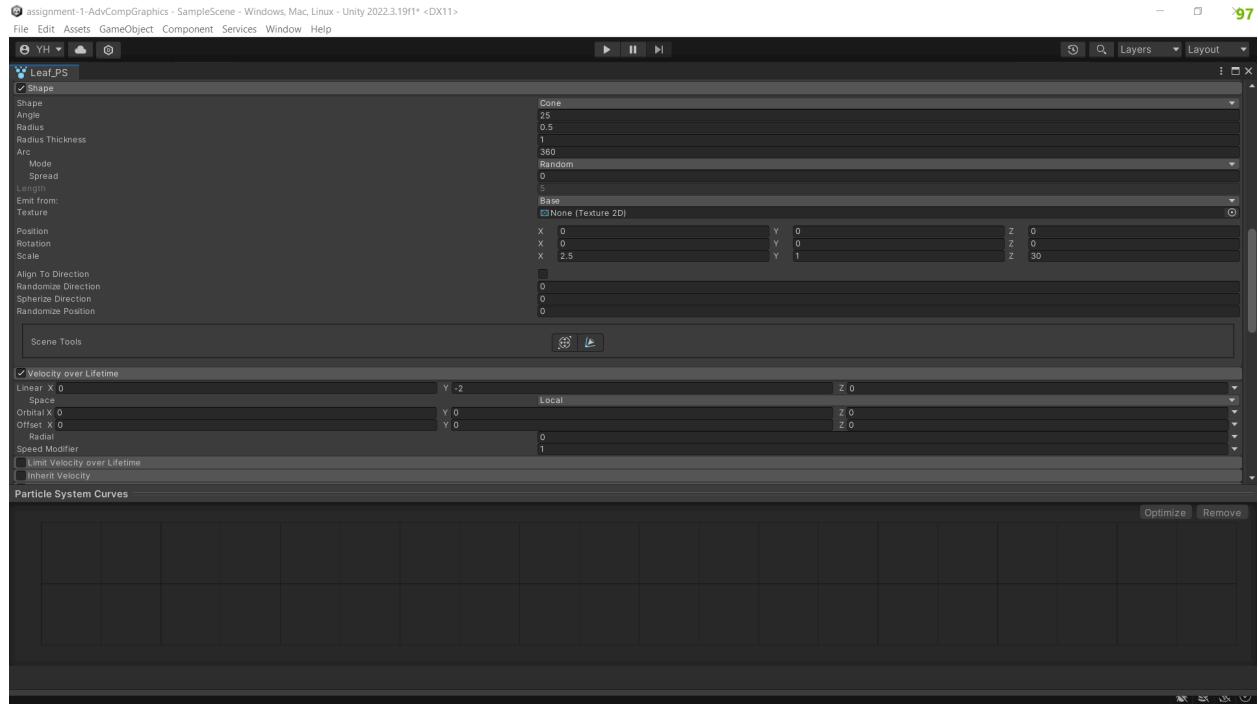




# Zone 5 Particle System

This particle system simulates leaves falling from a tree.





# Scanned Model

It's my wallet.

