26-9-2024

GAD178.3 Project 3 2D Asset Design: Creature Collection

SAE University College

Recommended Citation

Wun, N. (2024). *GAD178.3 Project 3 2D Asset Design: Creature Collection* [Student Exemplar, SAE University College]. Creo.

26-9-2024

2D Asset Design: Creature Collection

SAE University College

Recommended Citation

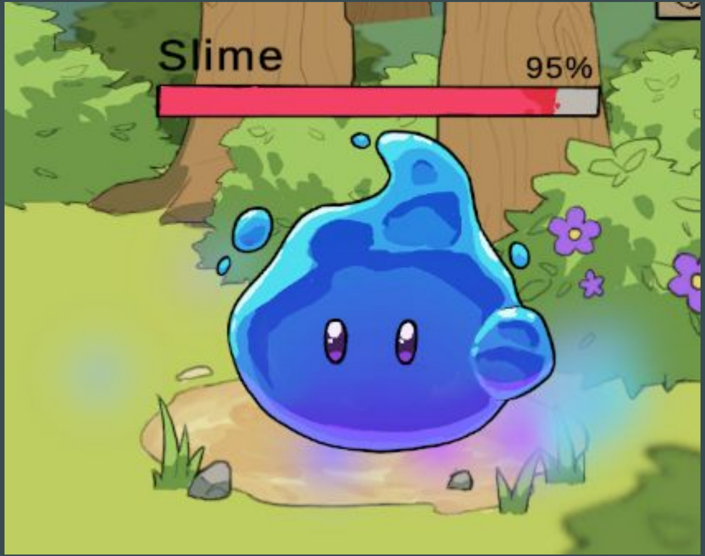
SAE University College "2D Asset Design: Creature Collection" (2024). *Exemplars*. 168.
https://creo.sae.edu.au/exemplars/168

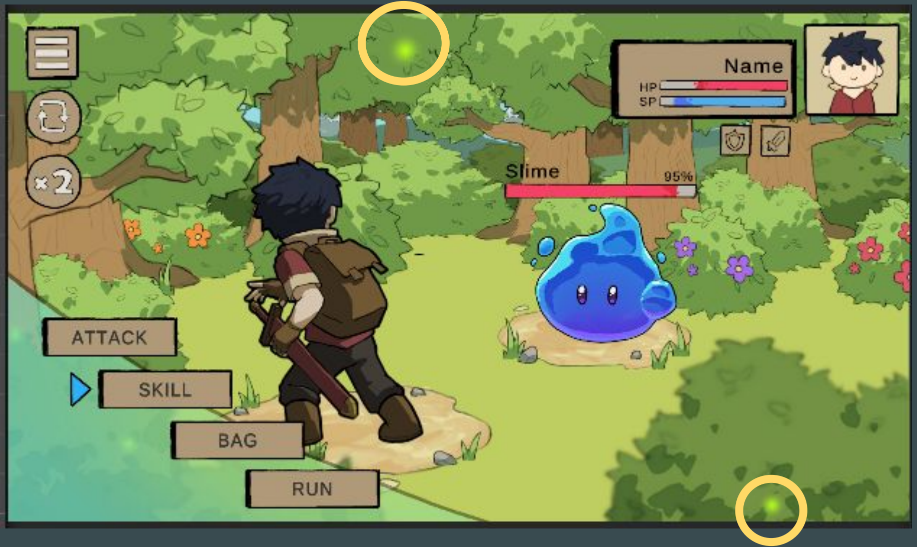
Project Document
GAD178
Project 3

Particle Effects in Game Engine:

The particle effects I have included in my main scene are a glowing mist effect for my slime and some random glowing particles emitted by the environment.

Both of these particle effects uses the default ParticleSystem material.

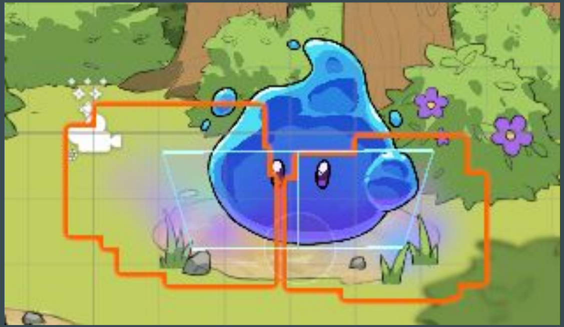


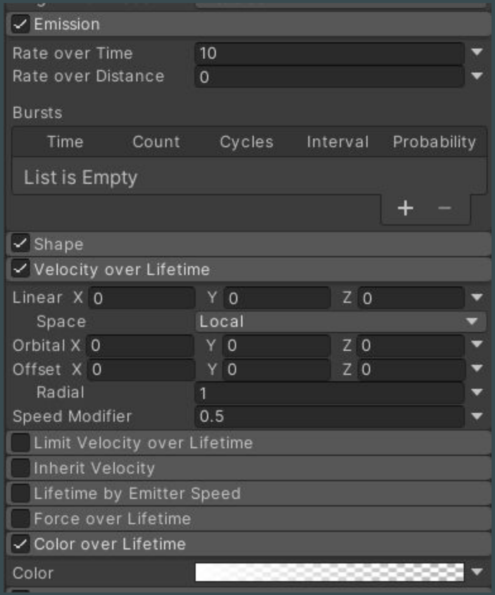


Particle Effects in Game Engine:

For the mist, I’ve set the colours to start randomly between two different colours. I’ve changed the emission rate, speed and max limit of the particles, and made it so they disappear shortly after being emitted.

I’ve also adjusted the colour and velocity over lifetime to make the particles slow down and fade away the longer the particle is present.





✓ Emission

Rate over Time	10
Rate over Distance	0

Bursts

Time	Count	Cycles	Interval	Probability

List is Empty

+ −

✓ Shape

✓ Velocity over Lifetime

Linear	X 0	Y 0	Z 0
Space	Local		
Orbital	X 0	Y 0	Z 0
Offset	X 0	Y 0	Z 0
Radial	1		
Speed Modifier	0.5		

□ Limit Velocity over Lifetime

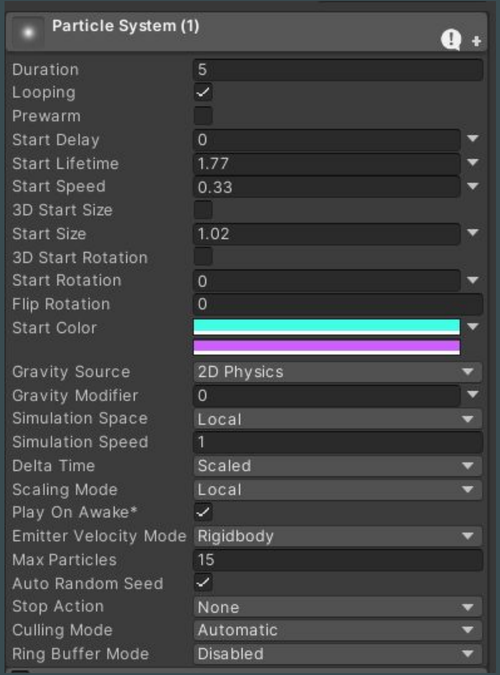
□ Inherit Velocity

□ Lifetime by Emitter Speed

□ Force over Lifetime

✓ Color over Lifetime

Color



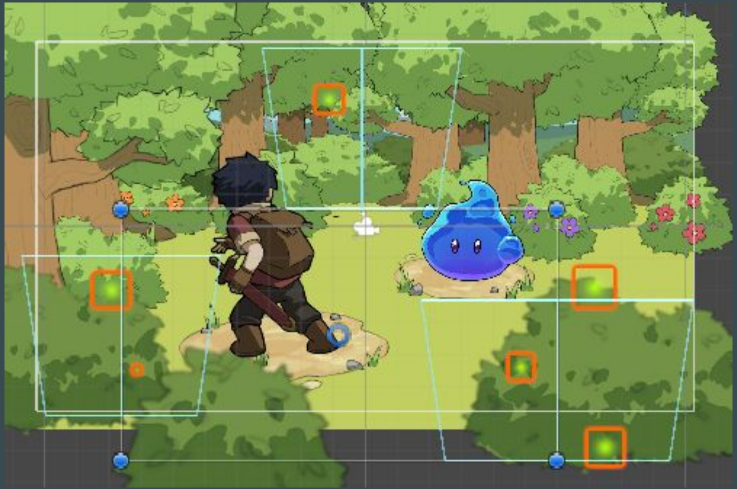
Particle System (1)

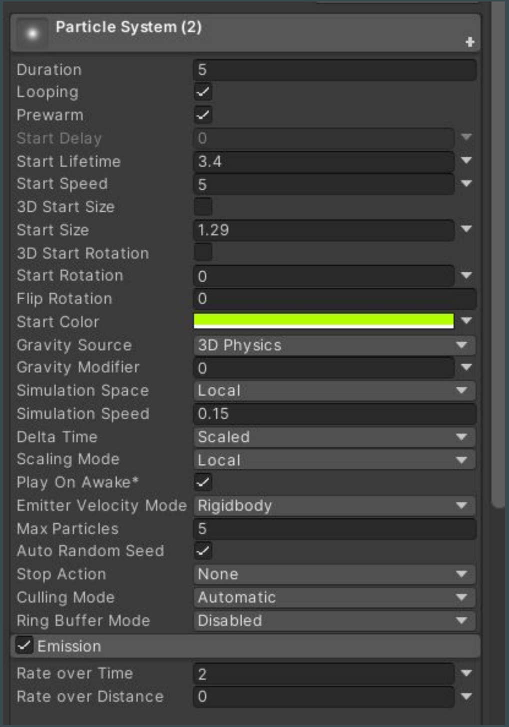
Duration	5
Looping	✓
Prewarm	□
Start Delay	0
Start Lifetime	1.77
Start Speed	0.33
3D Start Size	□
Start Size	1.02
3D Start Rotation	□
Start Rotation	0
Flip Rotation	0
Start Color	🟦 🟪
Gravity Source	2D Physics
Gravity Modifier	0
Simulation Space	Local
Simulation Speed	1
Delta Time	Scaled
Scaling Mode	Local
Play On Awake*	✓
Emitter Velocity Mode	Rigidbody
Max Particles	15
Auto Random Seed	✓
Stop Action	None
Culling Mode	Automatic
Ring Buffer Mode	Disabled

Particle Effects in Game Engine:

The scene is intended to have mysterious/magical forest vibes, which is why I made light green particles appear. To limit where the particles can appear, I made duplicates of the system and placed them around the scene.

The particle system has a lower emission rate but higher particle lifetime compared to the mist particles. I also made it so the particle sizes will change the longer it stays in the scene.



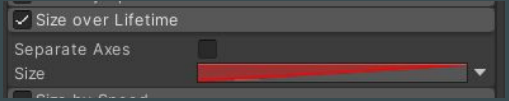


Particle System (2)

Duration	5
Looping	✓
Prewarm	✓
Start Delay	0
Start Lifetime	3.4
Start Speed	5
3D Start Size	☐
Start Size	1.29
3D Start Rotation	☐
Start Rotation	0
Flip Rotation	0
Start Color	🟩
Gravity Source	3D Physics
Gravity Modifier	0
Simulation Space	Local
Simulation Speed	0.15
Delta Time	Scaled
Scaling Mode	Local
Play On Awake*	✓
Emitter Velocity Mode	Rigidbody
Max Particles	5
Auto Random Seed	✓
Stop Action	None
Culling Mode	Automatic
Ring Buffer Mode	Disabled

✓ Emission

Rate over Time	2
Rate over Distance	0



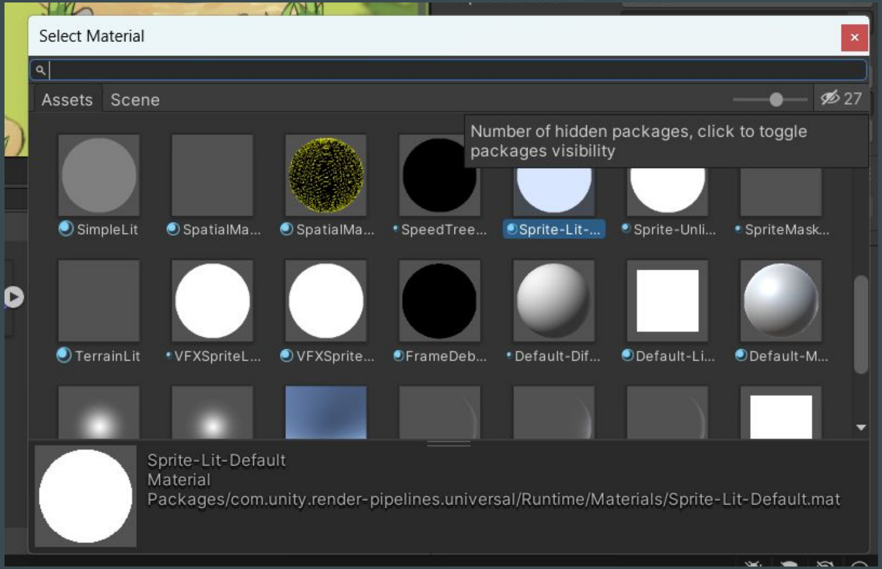
✓ Size over Lifetime

Separate Axes	☐
Size	🟥

Lighting in Game Engine:

Initially, the 2D lights wasn't working on my scene but I was eventually able to set it up properly after some guidance and tips from my instructor and peers.

The most helpful thing I've heard from them was swapping my 2D asset materials to Sprite-Lit-Default. This material is hidden from the material list by default for some reason, which caused me some confusion. Fortunately this is resolved when clicking on the eye button on the top right of the material screen to unhide all the packages, including the one I needed.

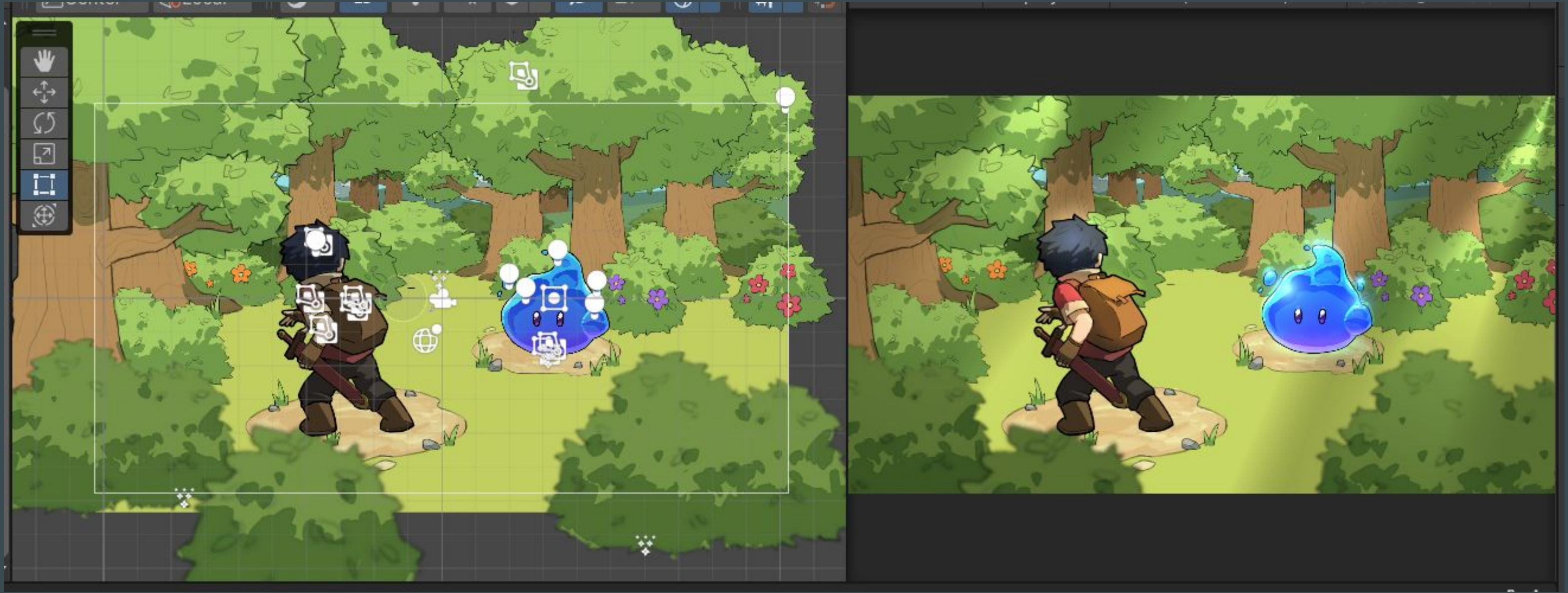


Number of hidden packages, click to toggle packages visibility

Sprite-Lit-Default
Material
Packages/com.unity.render-pipelines.universal/Runtime/Materials/Sprite-Lit-Default.mat

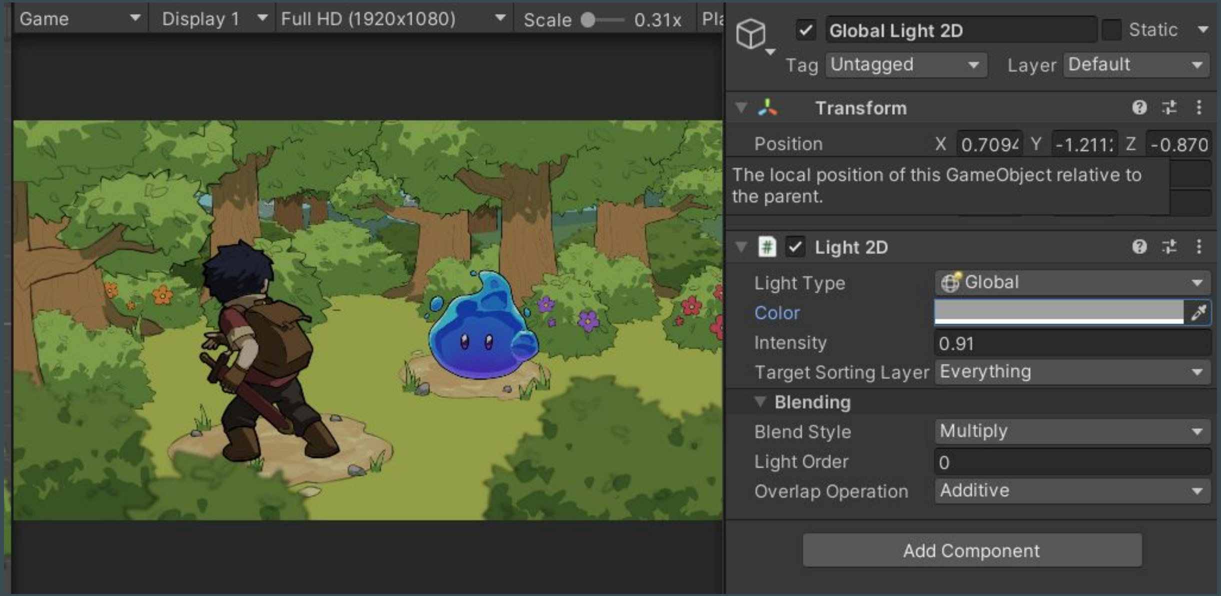
Lighting in Game Engine:

To make the scene look better, I used lighting to set the mood of the scene, made my slime glow and to give the player character some more colour. I am aware that an actual game scene shouldn't have so many lights used for character "shading", if any, but I figured it would be fine for the project since most of the assets in the scene won't be moving.



Lighting in Game Engine:

To start, I used a global light and changed the colour to a grey to make the scene look darker overall.



☑ Global Light 2D □ Static

Tag Untagged Layer Default

Transform

Position X 0.7094 Y -1.211 Z -0.870

The local position of this GameObject relative to the parent.

☑ Light 2D

Light Type: Global

Color

Intensity: 0.91

Target Sorting Layer: Everything

Blending

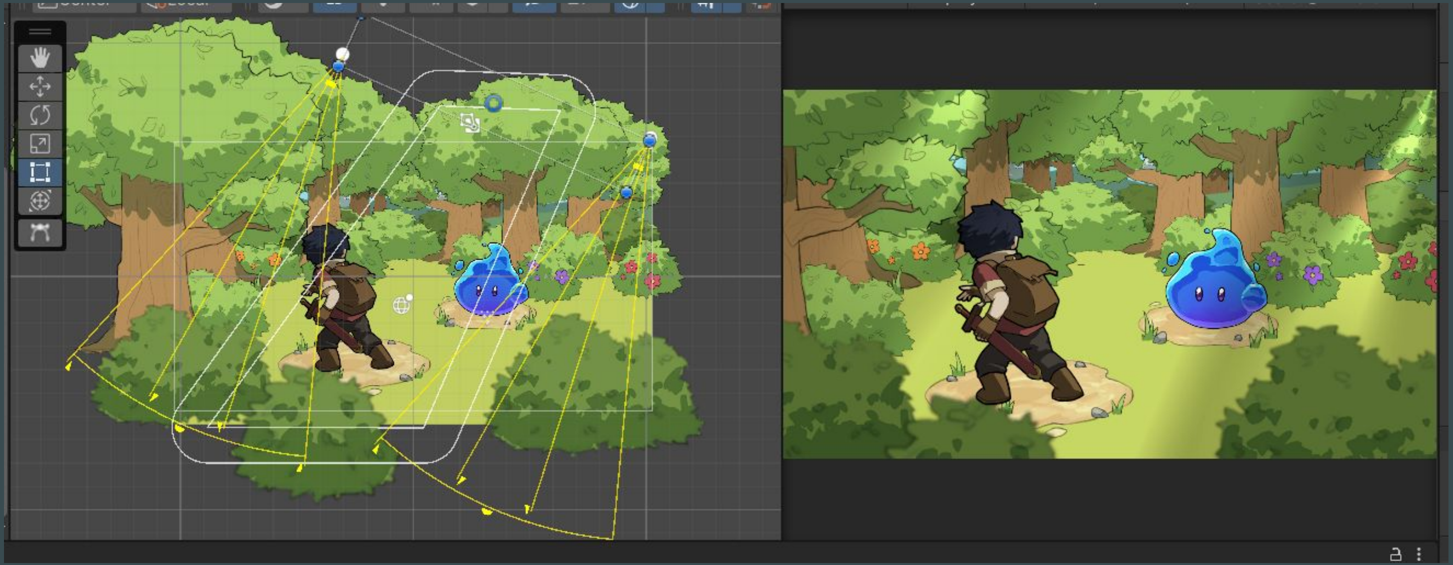
Blend Style: Multiply

Light Order: 0

Overlap Operation: Additive

Lighting in Game Engine:

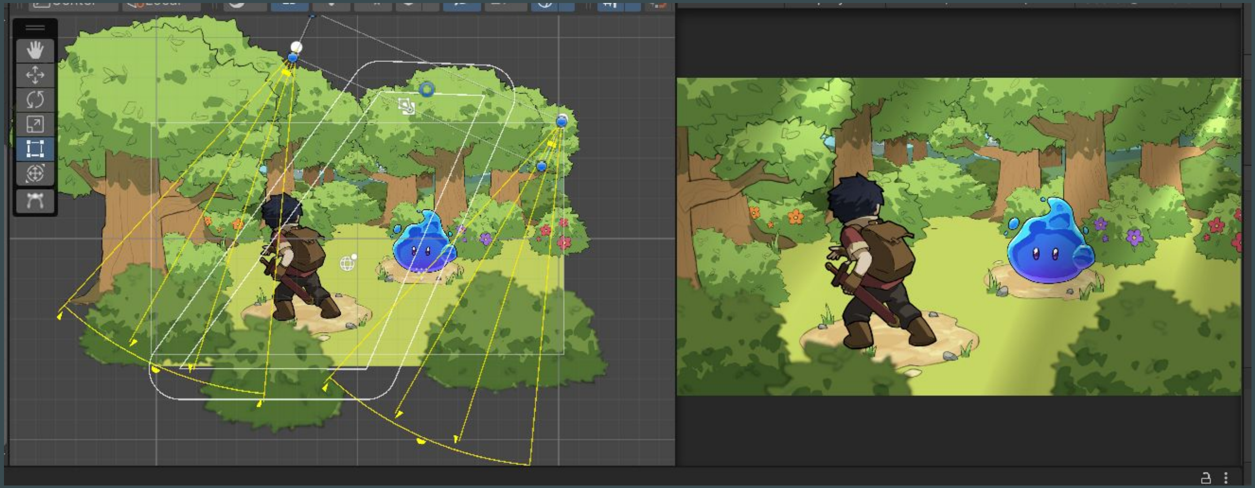
After that, I added two spot lights and a freeform light to simulate light coming from above through the gaps in the trees.

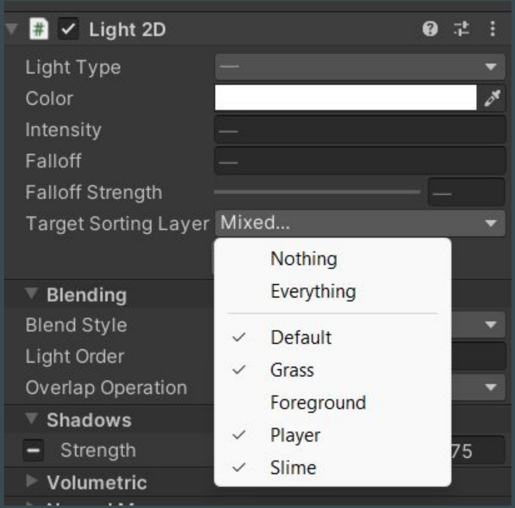


Lighting in Game Engine:

In order to prevent the lights from affecting the bushes I have placed at the very front of the scene, I had to create a new sorting layer named Foreground for the bushes and exclude it from the lighting target sorting layers.

Changing the Z position of the lights doesn't seem to work for what I wanted to do so my solution was to specify the layers the lights I added can affect instead.





☑ Light 2D

Light Type —

Color

Intensity —

Falloff —

Falloff Strength —

Target Sorting Layer Mixed...

- Nothing
- Everything
- ✓ Default
- ✓ Grass
- Foreground
- ✓ Player
- ✓ Slime

Blending

Blend Style

Light Order

Overlap Operation

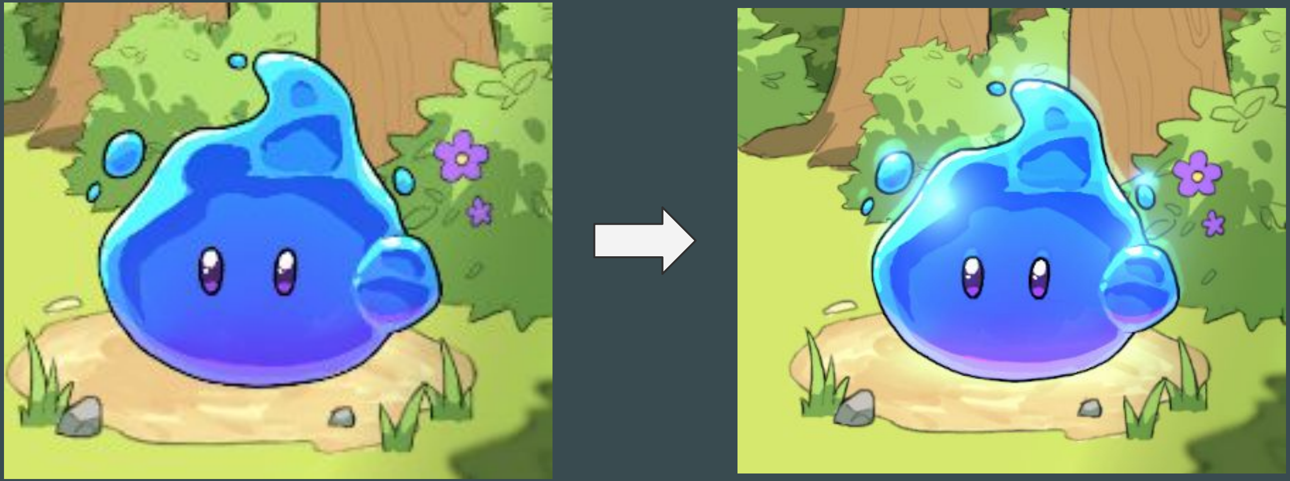
Shadows

Strength 75

Volumetric

Lighting in Game Engine:

I wanted my slime to shine a lot more and give out light slightly. This is why I have used a total of 8 lights on my slime: 5 spot lights, 2 freeform lights and one sprite light.

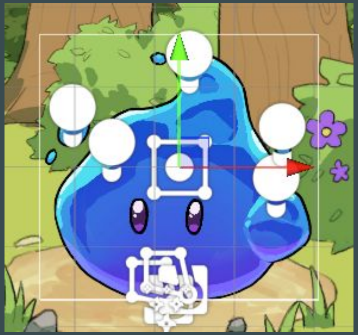


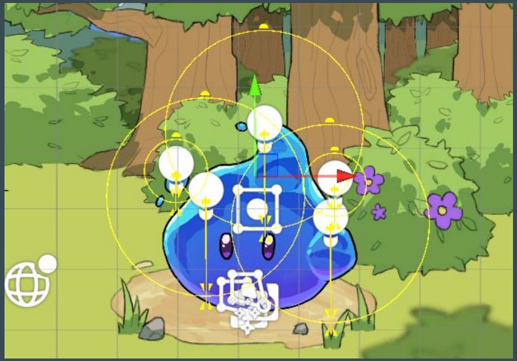
Lighting in Game Engine:

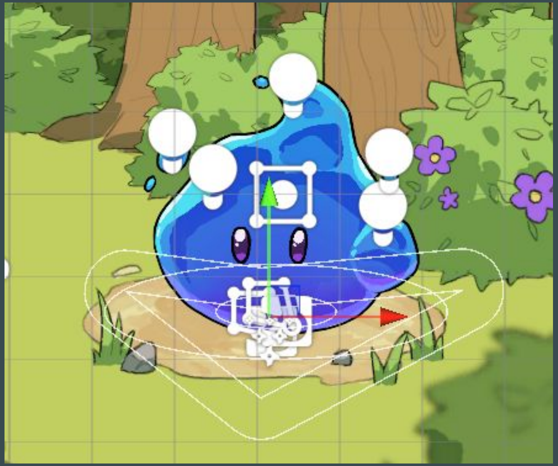
The spot lights are mainly used to make the slime's liquid body shine even more.

The two freeform lights are used for reflections/rim lighting purposes, one to brighten the bottom of the slime and the other to light up a small radius of the ground it's standing on.

The sprite light is mainly added for fun to add a glowing outline around the slime using its own shape.

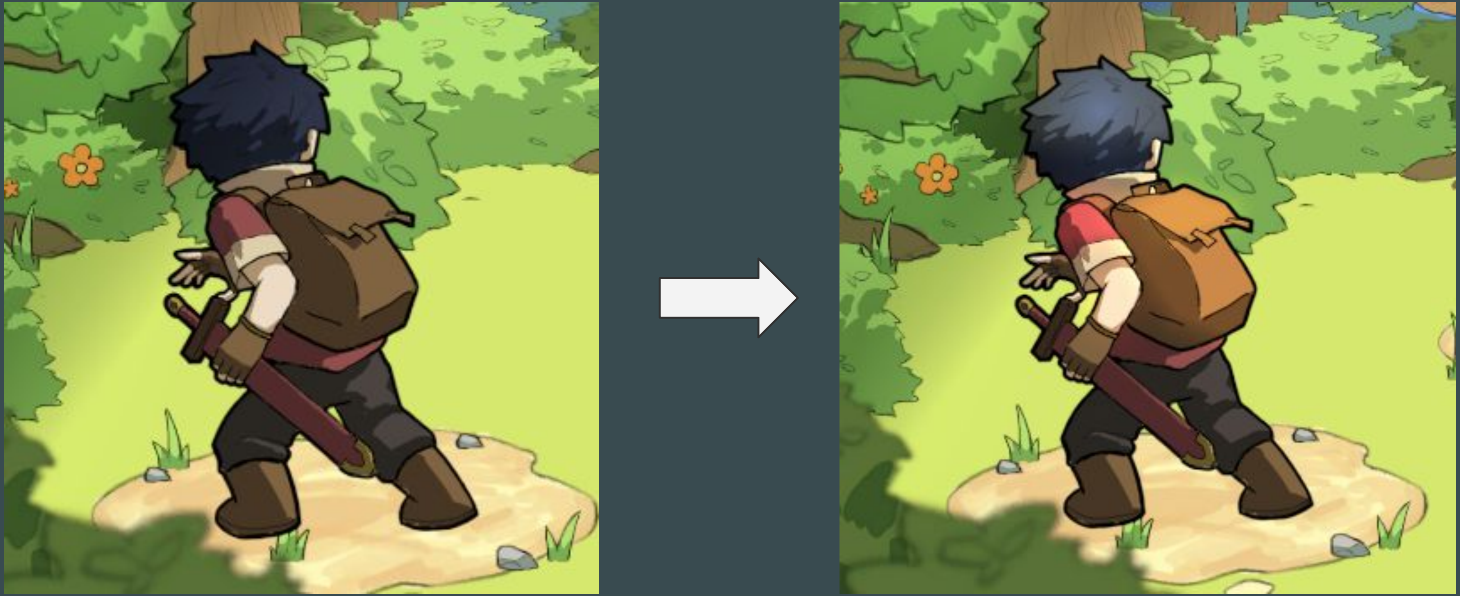






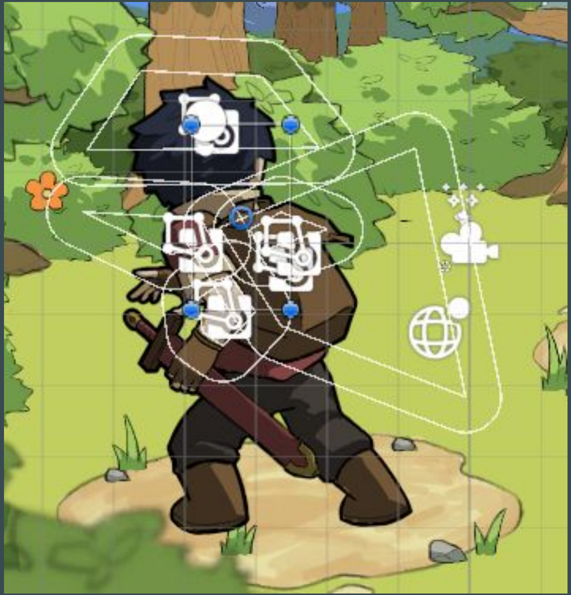
Lighting in Game Engine:

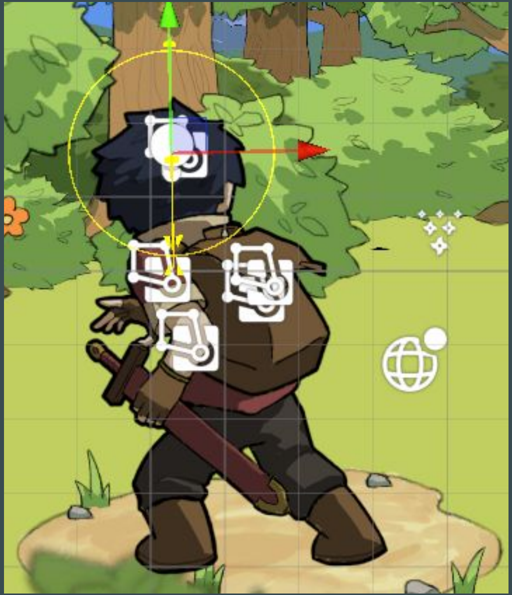
The player’s sprites feel a bit too dark and dull to me, which is why I decided to use lighting to fix it instead of remaking and reimporting the sprite. To do this, 6 freeform lights and 1 spot light is used.



Lighting in Game Engine:

Since the player is an animated sprite, I used mostly freeform lights to cover the rough areas that needs to be affected by the light. I made a new sorting layer called Player and set the lights to target only the new Player sorting layer, preventing the glow of the lights from affecting other parts of the background.

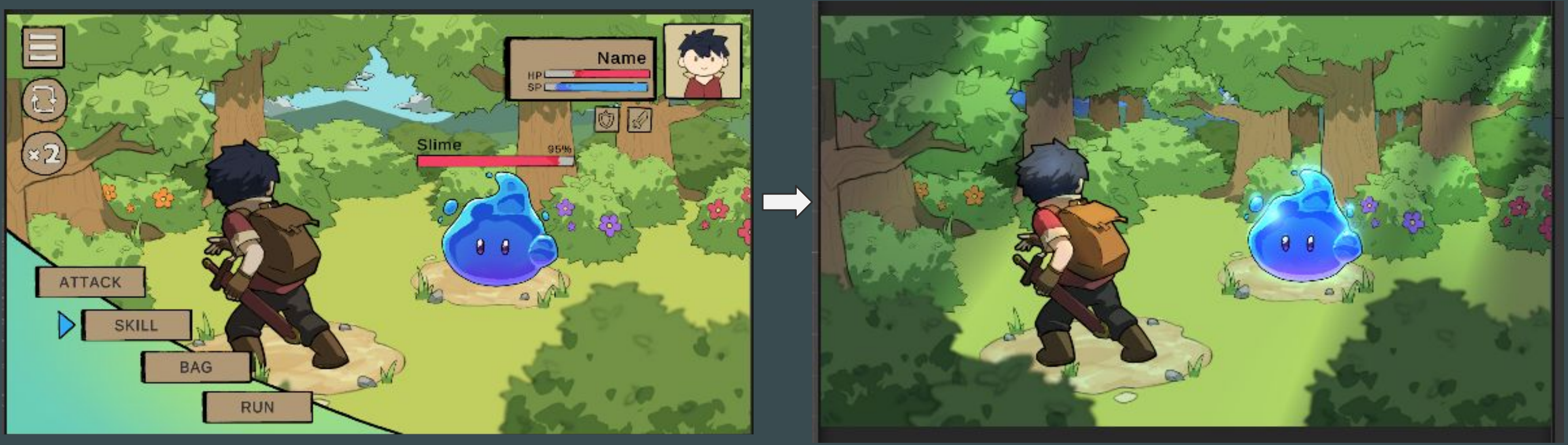




Lighting in Game Engine:

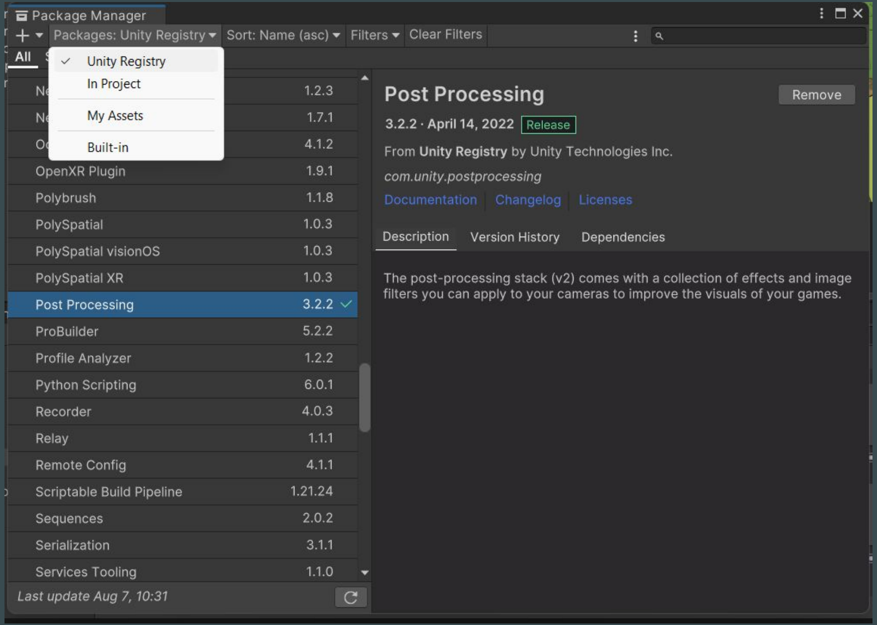
After receiving more feedback on how to make the scene fit my intentions, further adjustments were made to the environmental and global lighting. This made the scene look more like nighttime and helps the lights shining down stand out a bit more.

People also suggested filling the background with more trees, which helps with building an open space in the woods. I also darkened the colours of the newly added trees to enhance the sense of depth.



Post Processing in Game Engine:

I initially had problems with getting post processing to work but the issue is solved after downloading the post processing package into the Unity Registry. Watching the video by Willy Dev (2023) provided on campus online also really helped with understanding what the post processing effects can do.



Ne	1.2.3
Ne	1.7.1
Oc	4.1.2
OpenXR Plugin	1.9.1
Polybrush	1.1.8
PolySpatial	1.0.3
PolySpatial visionOS	1.0.3
PolySpatial XR	1.0.3
Post Processing	3.2.2 ✓
ProBuilder	5.2.2
Profile Analyzer	1.2.2
Python Scripting	6.0.1
Recorder	4.0.3
Relay	1.1.1
Remote Config	4.1.1
Scriptable Build Pipeline	1.21.24
Sequences	2.0.2
Serialization	3.1.1
Services Tooling	1.1.0

Last update Aug 7, 10:31

Post Processing

3.2.2 · April 14, 2022 Release

From **Unity Registry** by Unity Technologies Inc.

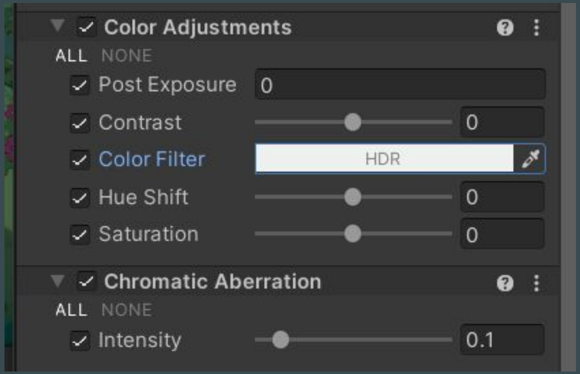
com.unity.postprocessing

Documentation Changelog Licenses

The post-processing stack (v2) comes with a collection of effects and image filters you can apply to your cameras to improve the visuals of your games.

Post Processing in Game Engine:

In my main scene, the post processing effects I used are: colour adjustments, chromatic aberration, bloom and vignette.



✓ Color Adjustments

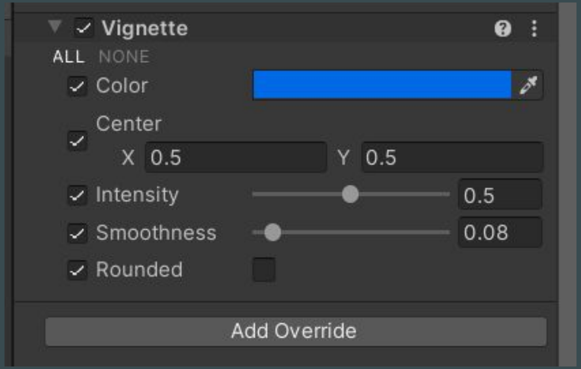
ALL NONE

- ✓ Post Exposure 0
- ✓ Contrast 0
- ✓ Color Filter HDR
- ✓ Hue Shift 0
- ✓ Saturation 0

✓ Chromatic Aberration

ALL NONE

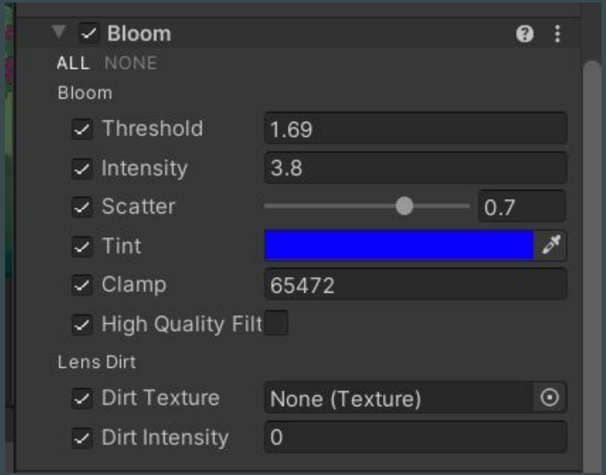
- ✓ Intensity 0.1



✓ Vignette

ALL NONE

- ✓ Color
- ✓ Center X 0.5 Y 0.5
- ✓ Intensity 0.5
- ✓ Smoothness 0.08
- ✓ Rounded □



✓ Bloom

ALL NONE

Bloom

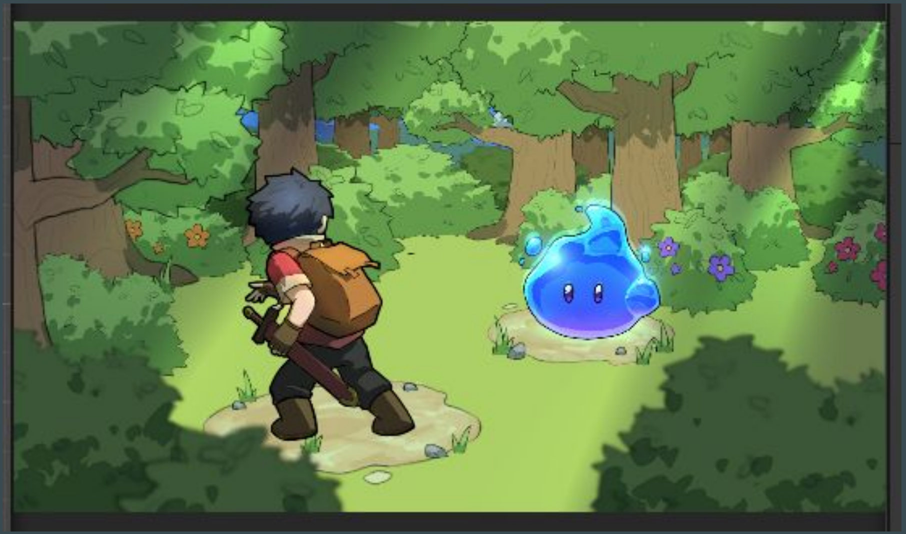
- ✓ Threshold 1.69
- ✓ Intensity 3.8
- ✓ Scatter 0.7
- ✓ Tint
- ✓ Clamp 65472
- ✓ High Quality Filt □

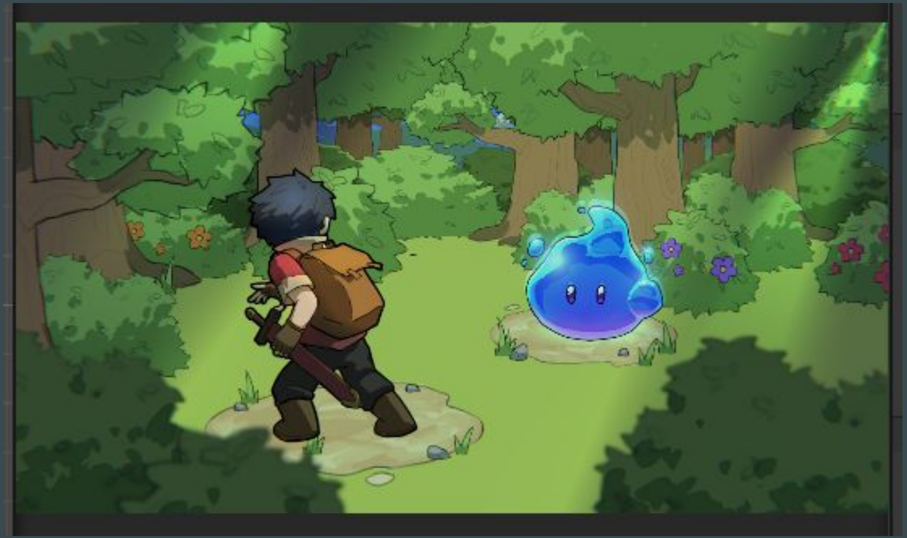
Lens Dirt

- ✓ Dirt Texture None (Texture)
- ✓ Dirt Intensity 0

Post Processing in Game Engine:

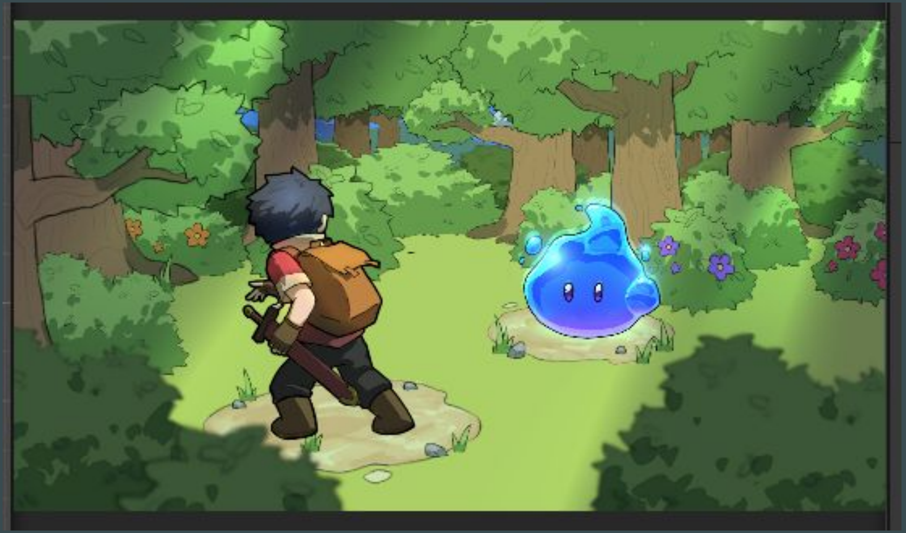
I used colour adjustments to further darken the scene slightly, making it look more like a night scene. I also applied a very slight chromatic aberration effect to increase the tension a little, just enough to change the edges of the scene but not enough for it to start affecting the colours.

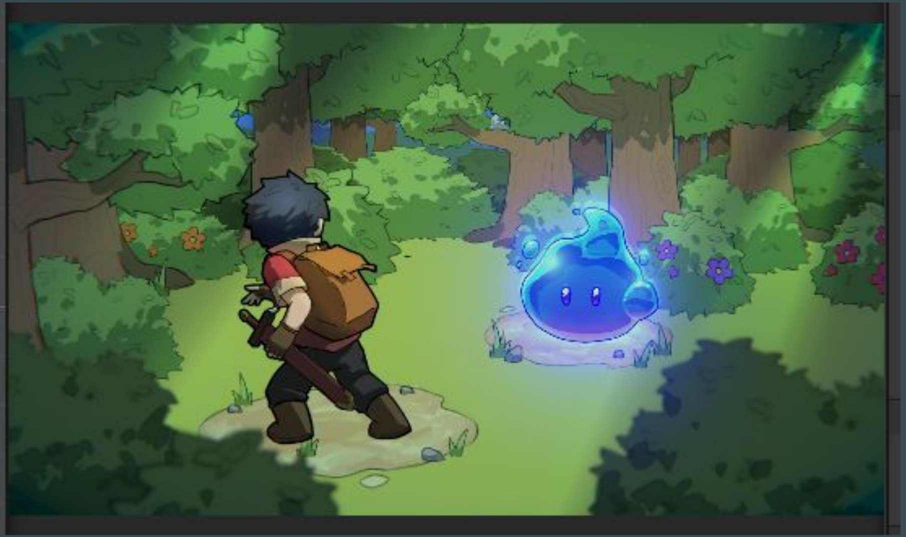




Post Processing in Game Engine:

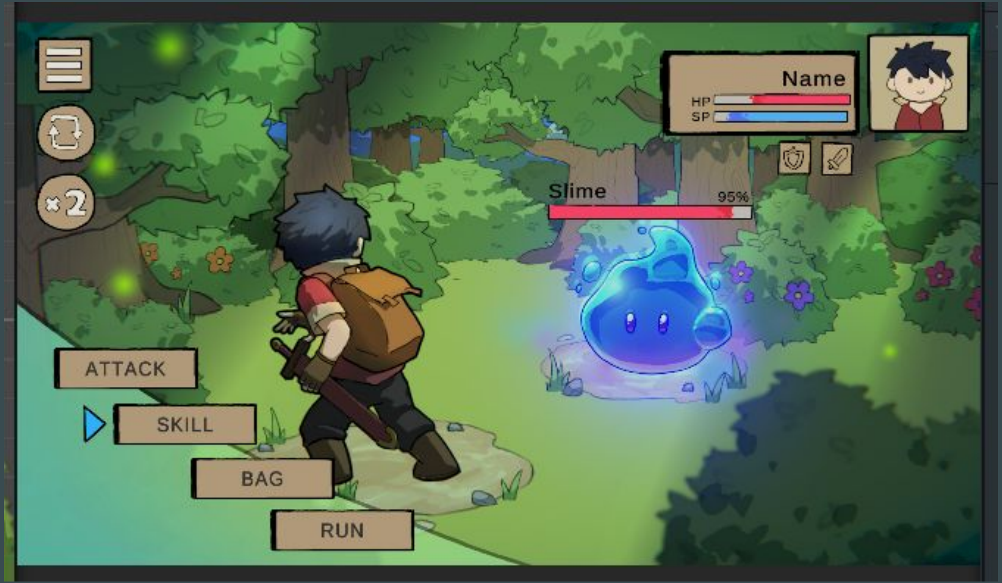
Bloom is used here to make the slime glow even further, making it seem like it’s eluding a blue aura. A vignette is then applied to make the edges of the screen darker, helping the fight in the middle to pop out more.





Scene 1 Final Results:

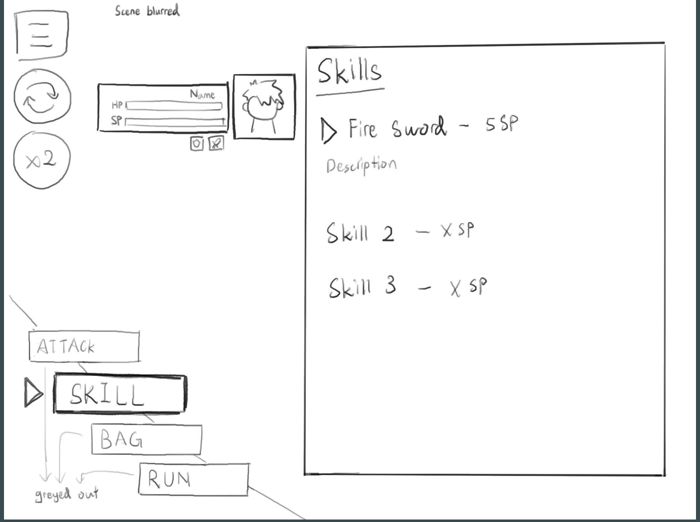
Here is what the first scene looks like with the lighting, particle effects and post processing effects applied to it.

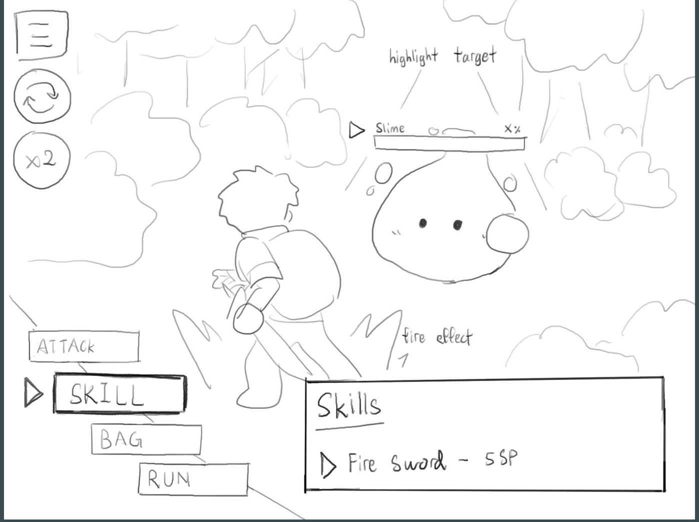


Reapplying the process:

Since I’m making a battle scene, it’ll be tough to make up new scenes without making new assets. In the end, I’ve decided to make 3 other scenes involving different UI placements.

Scene 2 and 3 are both directly linked to the first scene. Scene 2 features the skill option on the menu being selected, bringing up a skill selection menu. Scene 3 is when the player has selected a skill to use and is now given the option to select the enemy target to use the skill on.

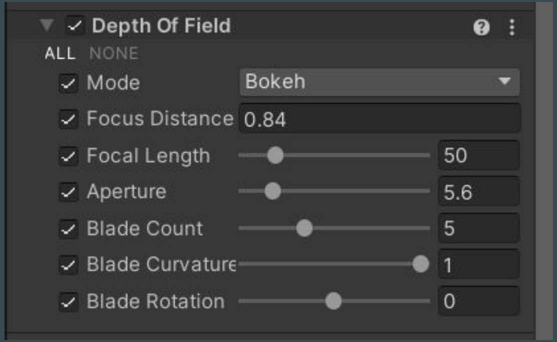




Reapplying the process:

Here is how Scene 2 looked in the game engine. The menu panel has been created using the menu 9 slice asset that I already have and the player details have been moved to accommodate the new skill menu.

The background of the scene has been blurred using the Depth of Field post processing effect in Bokeh mode. Even though the depth part of the effect doesn't work properly in 2D, the effect is still usable in my case as I wanted everything to be blurred out.



☑ Depth Of Field

ALL NONE

☑ Mode: Bokeh

☑ Focus Distance: 0.84

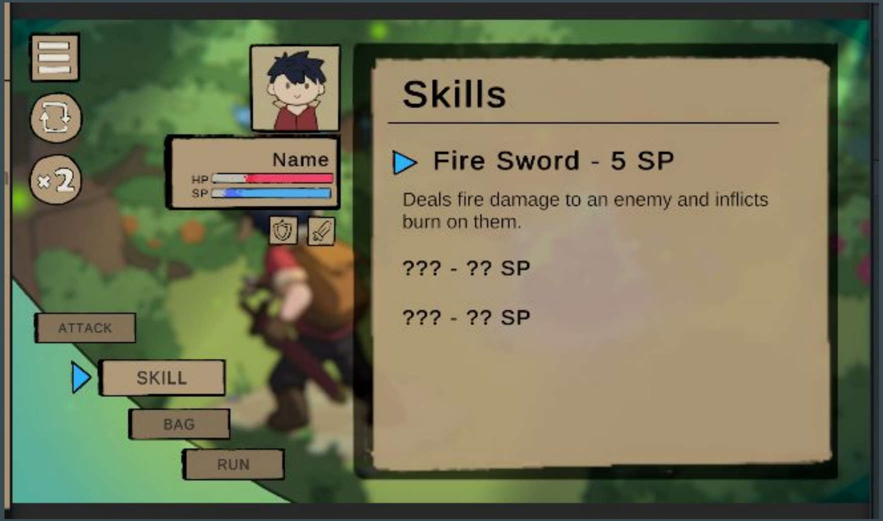
☑ Focal Length: 50

☑ Aperture: 5.6

☑ Blade Count: 5

☑ Blade Curvature: 1

☑ Blade Rotation: 0



Skills

Fire Sword - 5 SP

Deals fire damage to an enemy and inflicts burn on them.

??? - ?? SP

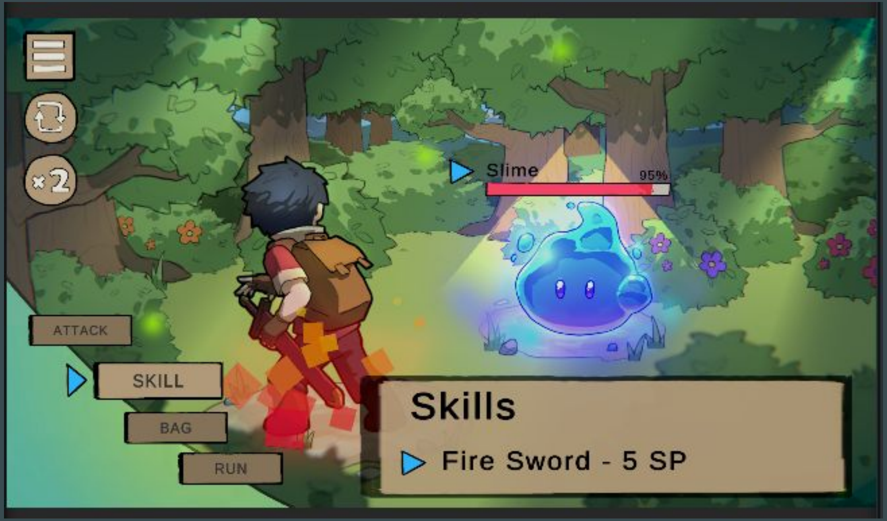
??? - ?? SP

Reapplying the process:

Here is how Scene 3 looked in the game engine. The skill menu has been resized and the Depth of Field effect is now removed, allowing the battle to be seen clearly again.

I’ve added a particle effect at the player’s feet releasing particles that starts out red and ends with yellow before fading away, imitating a fiery hot feel. I’ve also added some red lights at the player’s feet to make the temperature seem higher. To make things different, I changed the particle effect material, resulting in squares instead of the default soft circles.

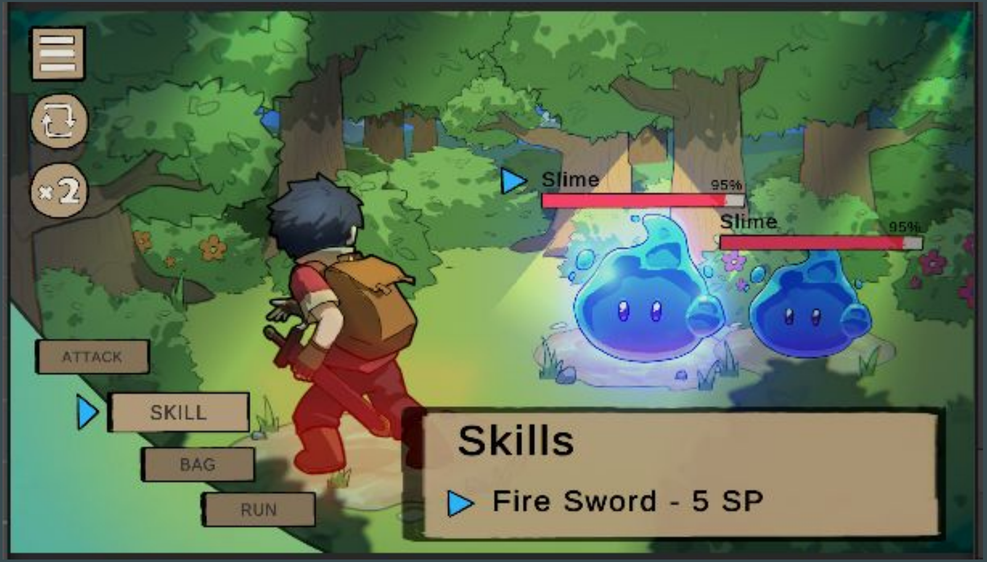
A spot light has been added on top of the slime to show that its being targeted by the player. I also added the blue pointer beside the slime’s name to make the fact that it's the target clearer.



Reapplying the process:

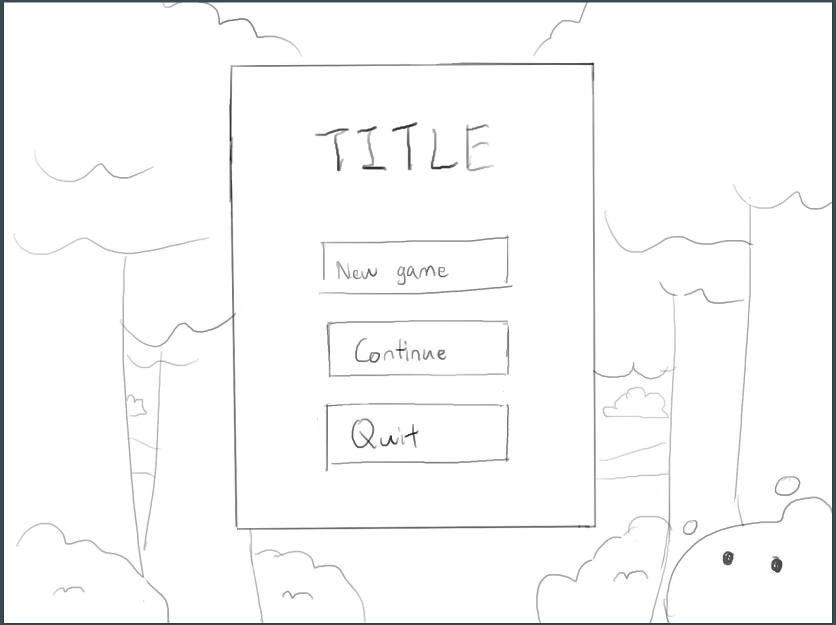
After considering the feedback I’ve received, I added another slime to the scene to show how it would look like if there are multiple enemies present in the field.

The enemies who aren’t being targeted in the scene will be slightly darkened along with their HP bar, making the current target(s) stand out more.



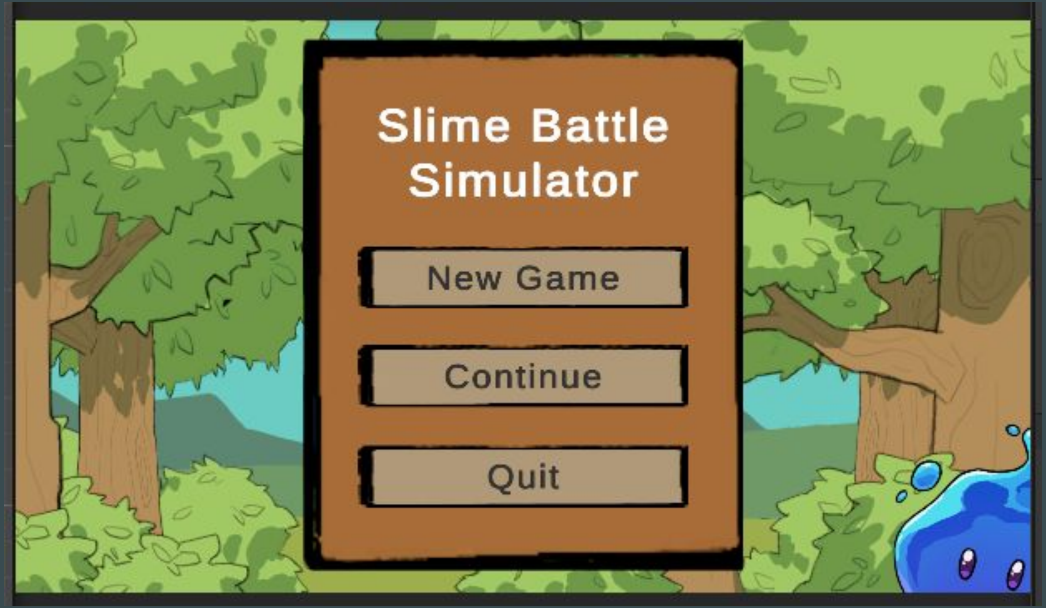
Reapplying the process:

Scene 4 will be a main menu scene featuring a new background and some basic main menu UI. The main idea for the background is having clear, open plains behind a few trees. The trees will be darker than the plains but has some lighting on the top, making it feel like the player is just entering the dark woods.



Reapplying the process:

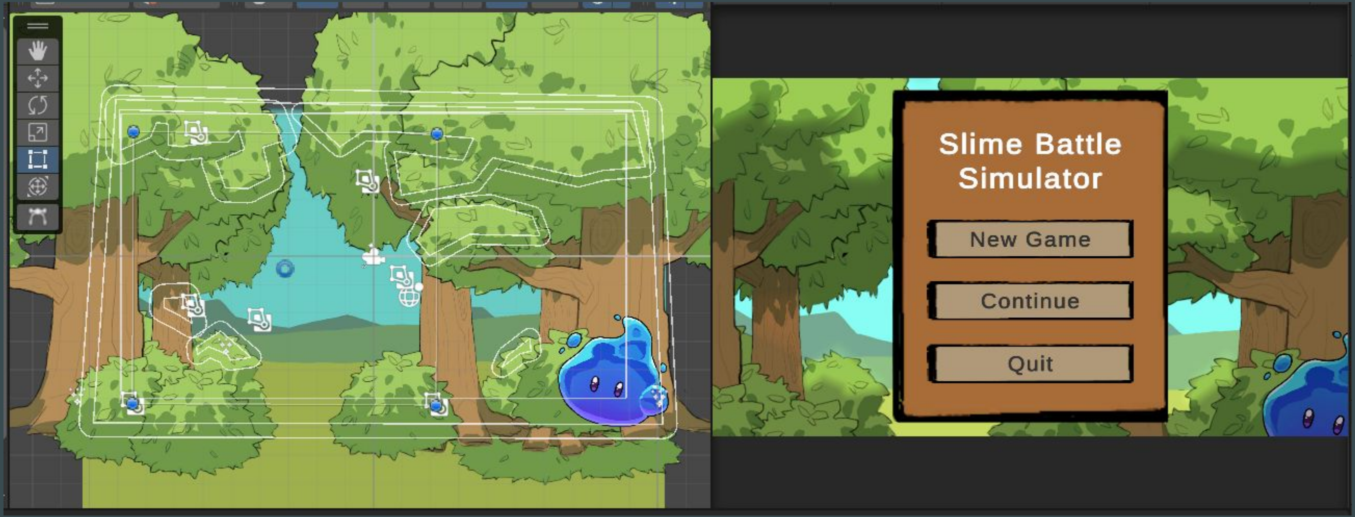
Using the assets that I already have, I was able to quickly put together the scene and make a main menu UI.



Reapplying the process:

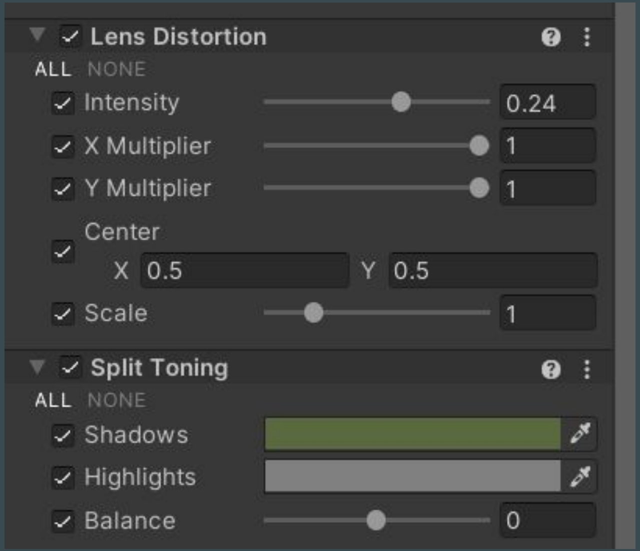
For the lighting in this scene, I tried playing around a bit more with the freeform lights.

I first added a global light to make everything slightly dark. Then I made a freeform light covering the entire scene but set it to only affect the ground and the sky at the back, making it feel like the trees are giving a lot of shade. After that, I added freeform lights on top of the trees and some bushes to give them a sense of depth. While this can easily be done with regular lights on normal maps on the 2D assets, I am unable to make any for my scene and had to improvise.



Reapplying the process:

3 different post processing effects are being used in this main menu scene: Lens Distortion, Split Toning and a bit of Bloom.



☑ **Lens Distortion**

ALL NONE

☑ Intensity 0.24

☑ X Multiplier 1

☑ Y Multiplier 1

☑ Center X 0.5 Y 0.5

☑ Scale 1

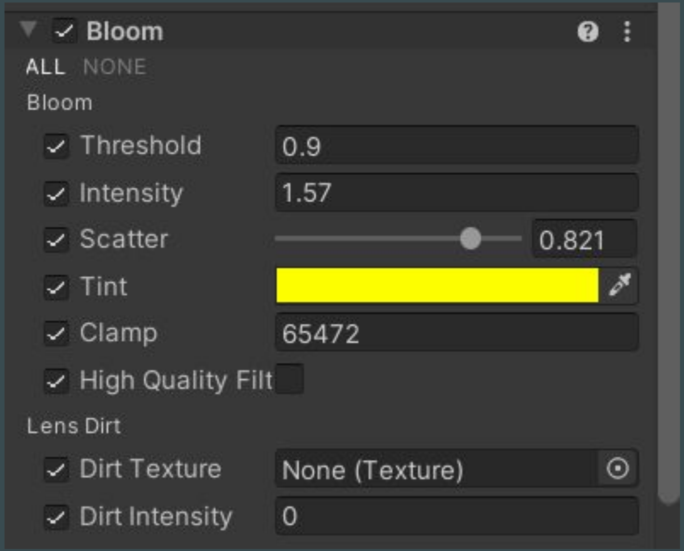
☑ **Split Toning**

ALL NONE

☑ Shadows 🟩

☑ Highlights ⬜

☑ Balance 0



☑ **Bloom**

ALL NONE

Bloom

☑ Threshold 0.9

☑ Intensity 1.57

☑ Scatter 0.821

☑ Tint 🟨

☑ Clamp 65472

☑ High Quality Filt □

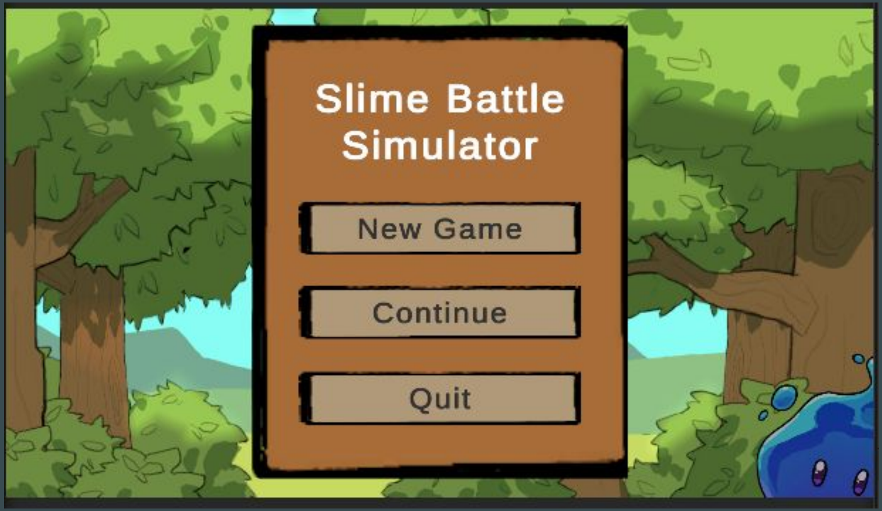
Lens Dirt

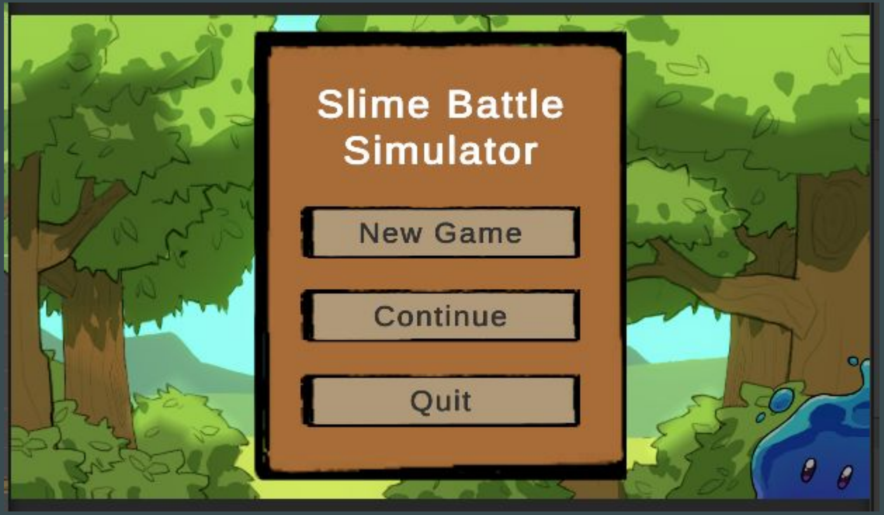
☑ Dirt Texture None (Texture)

☑ Dirt Intensity 0

Reapplying the process:

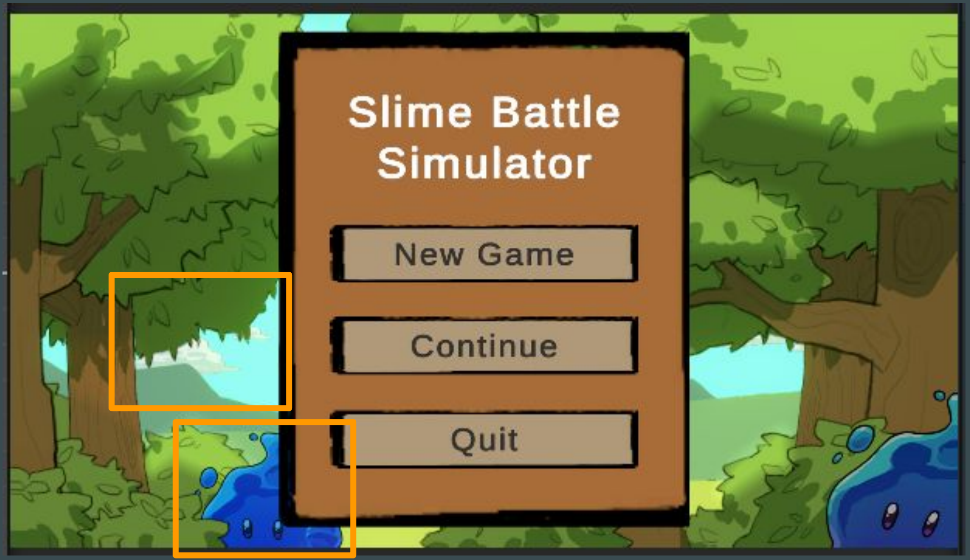
The split toning made the scene look a bit greener while the slight bloom applied gave a bit of a glow around the tree’s edges. Meanwhile, the lens distortion gave the trees a bit of a curve, making them look less straight.





Reapplying the process:

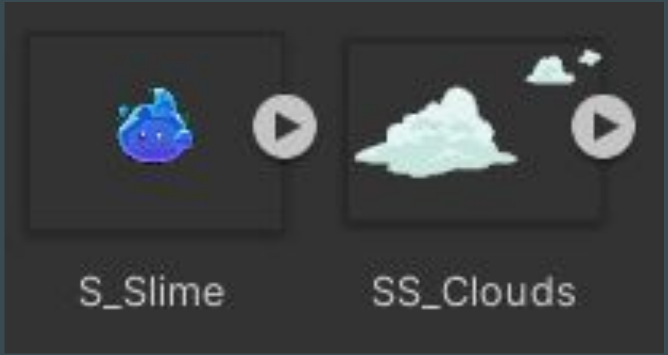
In this scene, I decided to use the particle system to simulate moving objects rather than making an animation for them. There are two different particle systems added, one for the slimes and one for the clouds. The idea is to have slimes moving across the screen beneath the trees and clouds being blown by the wind in the sky at the back of the scene.

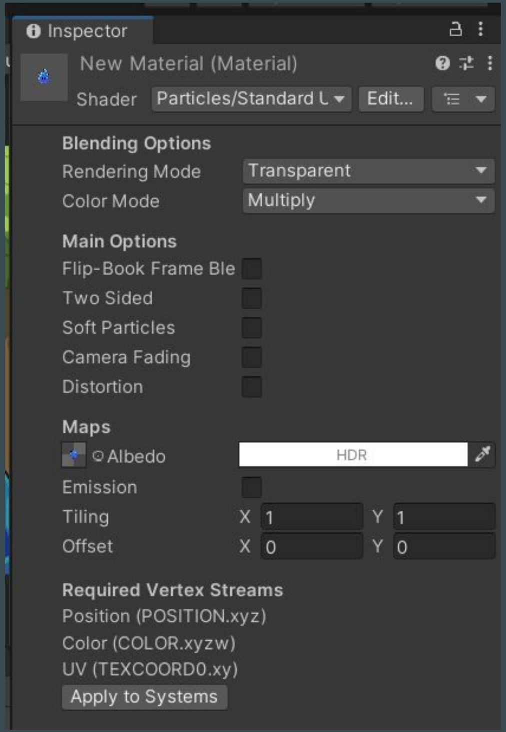


Reapplying the process:

In order to use the slime as a particle, I imported an image of it alone and made a material out of it using the transparent rendering mode.

I have also imported a sprite atlas of the clouds which I will be using for my other particle system.





New Material (Material)

Shader Particles/Standard U Edit...

Blending Options

Rendering Mode: Transparent

Color Mode: Multiply

Main Options

Flip-Book Frame Ble □

Two Sided □

Soft Particles □

Camera Fading □

Distortion □

Maps

Albedo: HDR

Emission □

Tiling: X 1 Y 1

Offset: X 0 Y 0

Required Vertex Streams

Position (POSITION.xyz)

Color (COLOR.xyzw)

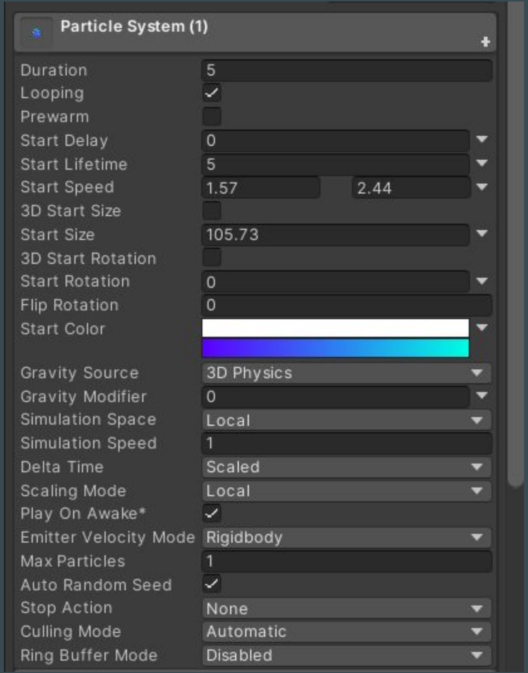
UV (TEXCOORD0.xy)

Apply to Systems

Reapplying the process:

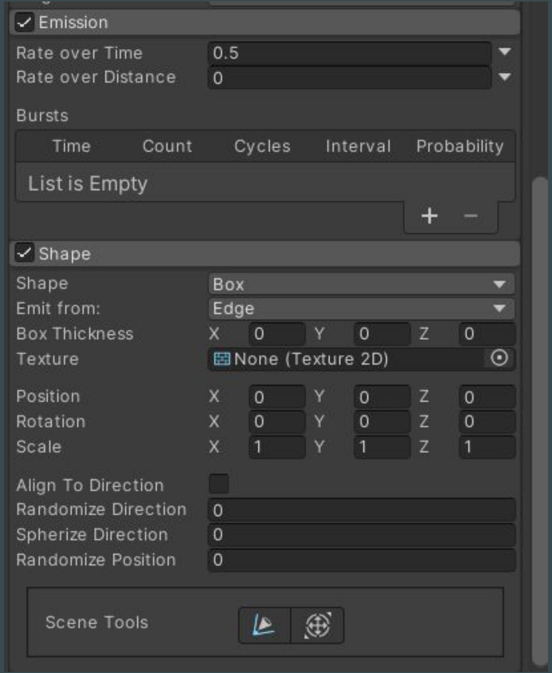
Using the slime material, I was able to make a box shaped system to emit slimes in a single direction. I gave the system two different start speeds to make fast and slow moving slimes, plus different starting colours to make each slime passing by a bit more unique.

Things like the emission rate and the max particles allowed at once were adjusted to ensure the screen doesn't get filled with too many slimes.



Particle System (1)

Duration	5
Looping	✓
Prewarm	□
Start Delay	0
Start Lifetime	5
Start Speed	1.57 2.44
3D Start Size	□
Start Size	105.73
3D Start Rotation	□
Start Rotation	0
Flip Rotation	0
Start Color	
Gravity Source	3D Physics
Gravity Modifier	0
Simulation Space	Local
Simulation Speed	1
Delta Time	Scaled
Scaling Mode	Local
Play On Awake*	✓
Emitter Velocity Mode	Rigidbody
Max Particles	1
Auto Random Seed	✓
Stop Action	None
Culling Mode	Automatic
Ring Buffer Mode	Disabled



✓ Emission

Rate over Time	0.5
Rate over Distance	0

Bursts

Time	Count	Cycles	Interval	Probability
List is Empty				

✓ Shape

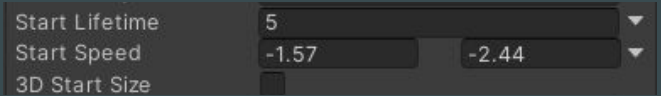
Shape	Box
Emit from:	Edge
Box Thickness	X 0 Y 0 Z 0
Texture	None (Texture 2D)
Position	X 0 Y 0 Z 0
Rotation	X 0 Y 0 Z 0
Scale	X 1 Y 1 Z 1
Align To Direction	□
Randomize Direction	0
Spherize Direction	0
Randomize Position	0

Scene Tools

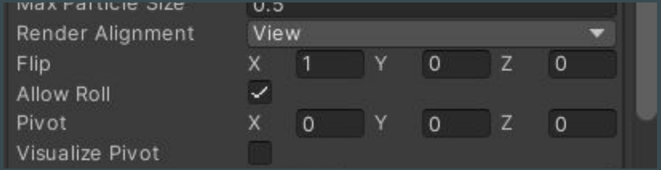
Reapplying the process:

The slime emitter particle system was then placed at one edge of the screen so slimes can spawn off screen, move then disappear on the other side. After finishing all my adjustments, I duplicated the system and placed it on the other end to simulate slimes travelling in the other direction.

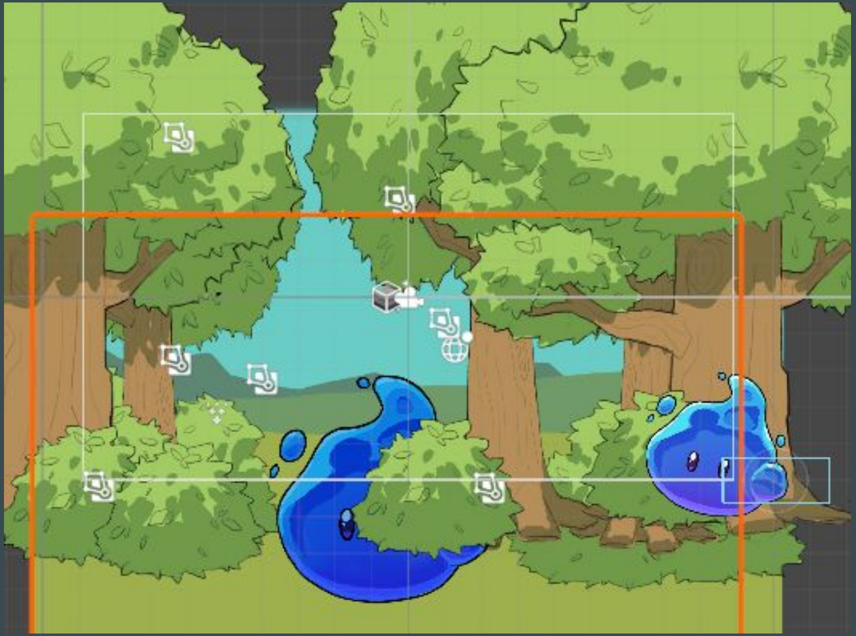
Playing around with the speed led me to discover that having a negative start speed listed would cause the particles to travel in the opposite direction, allowing me to take advantage of that here. I also had to flip the sprites so they’re facing the right direction, which is done by setting flip X to 1 under the Renderer section in the inspector.



```
Start Lifetime      5
Start Speed         -1.57        -2.44
3D Start Size       □
```

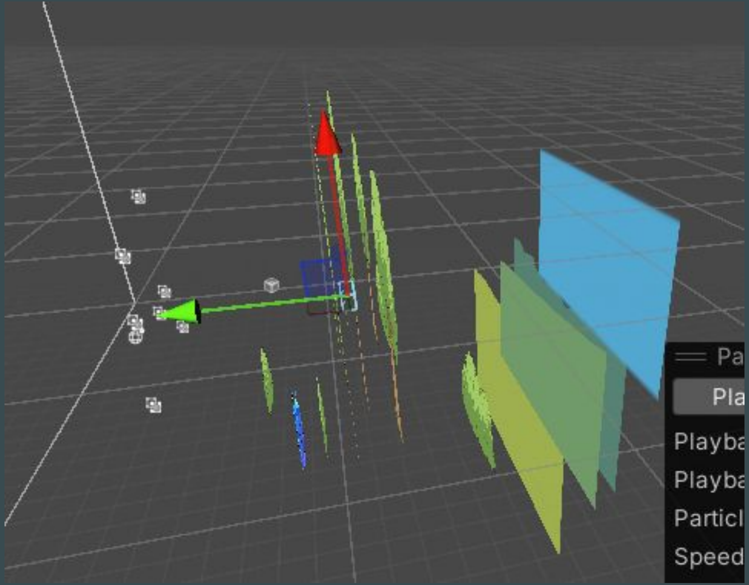


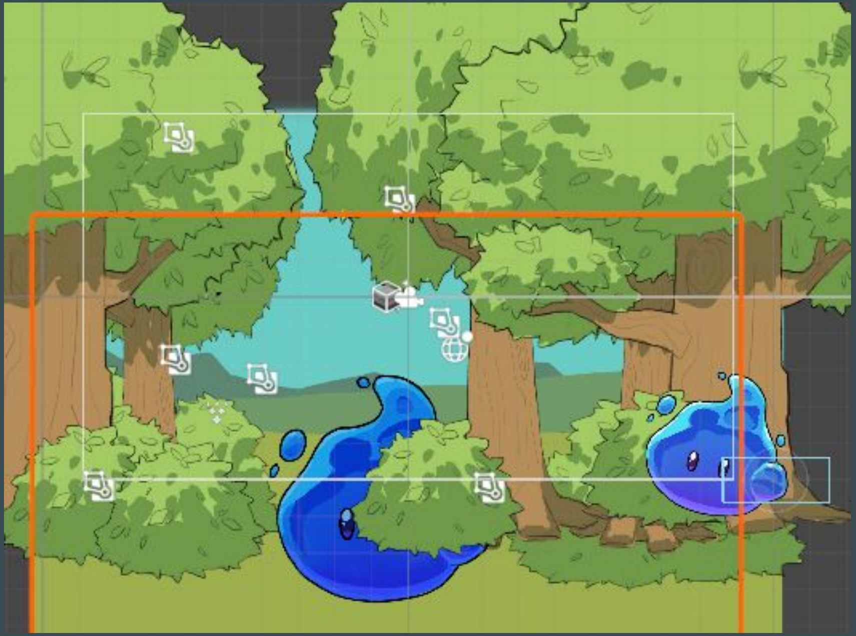
```
Max Particle Size   0.5
Render Alignment    View
Flip                X  1    Y  0    Z  0
Allow Roll          ☑
Pivot               X  0    Y  0    Z  0
Visualize Pivot     □
```



Reapplying the process:

I adjusted the Z positioning for the particle systems so the slimes can be seen moving in between the layers of bushes and trees rather than having the slimes spawn on top or behind all the plants in the scene.

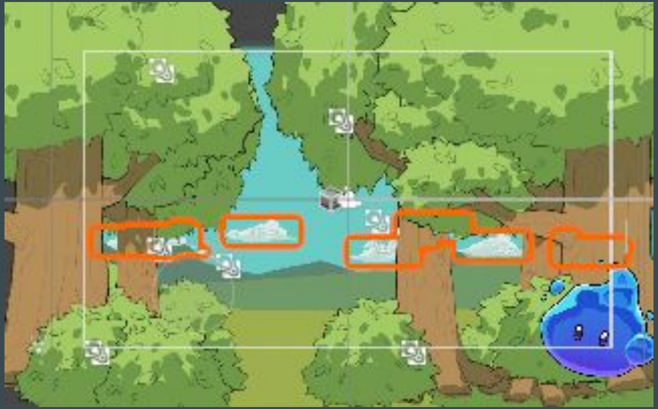


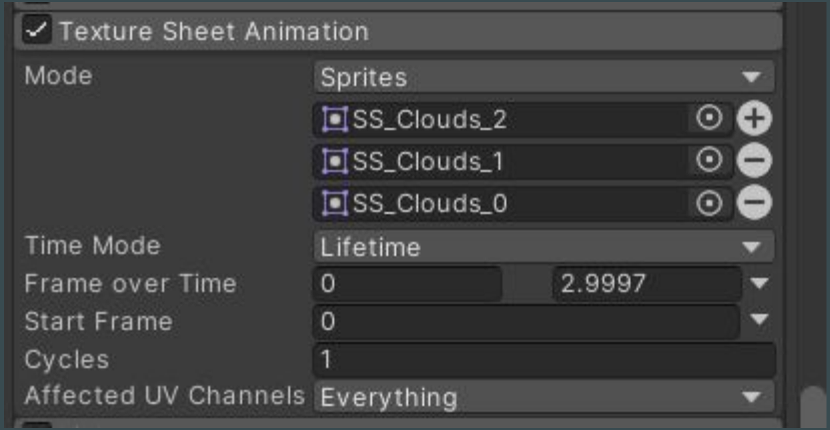


Reapplying the process:

The clouds system is similar to the slime system in the way that they both emit a custom sprite, however I had to do things a bit differently since I wanted the system to emit all 3 of my cloud variants. Using a Youtube video by SpeedTutor (2021) as a reference, I was able to get the system working like I wanted it to.

I made a sprite atlas of the 3 cloud variants, sliced them up and added them into the particle system under the texture sheet animation section. Since the time mode is set to lifetime, the sprites won't be changing. With the frame over time set on a random number between two constants, the system is able to randomly emit a cloud variant from the list.





☑ Texture Sheet Animation

Mode: Sprites
- SS_Clouds_2
- SS_Clouds_1
- SS_Clouds_0

Time Mode: Lifetime

Frame over Time: 0 2.9997

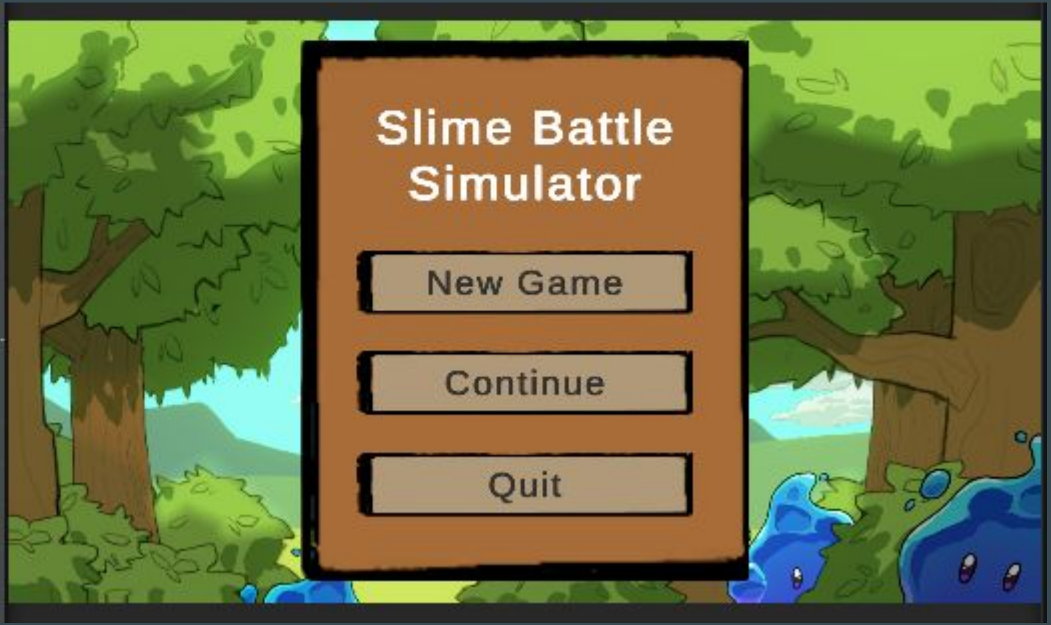
Start Frame: 0

Cycles: 1

Affected UV Channels: Everything

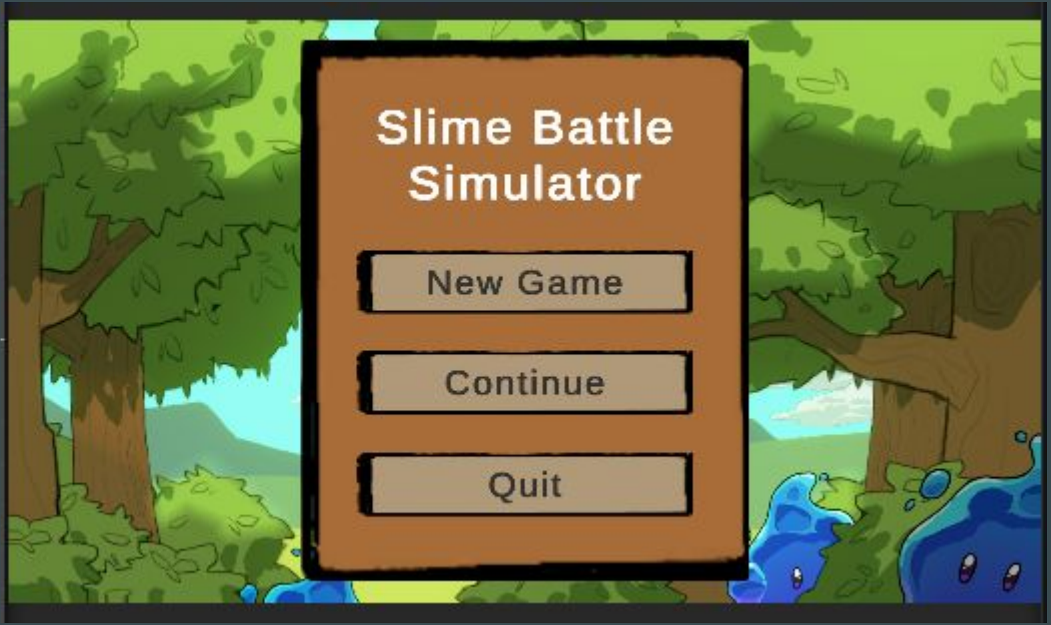
Reapplying the process:

Here is how Scene 4 looks in the end with the lighting, particle effects and post processing effects applied.



Reapplying the process:

The feedback I’ve received for this scene is that the clouds and slimes are moving too fast, which may be a distraction from the main menu. To fix this, I lowered the speed of the moving clouds and slimes by a lot.

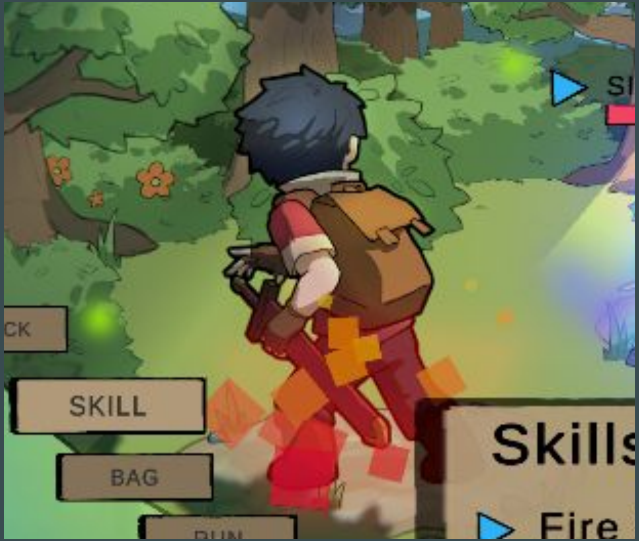


Feedback:

I’ve received some feedback on how to make the particle effects in my scene look better, more specifically on the flame effect and the green glowing particles.

One person suggested that I could try to make the green glowing particles enlarge and fade out randomly as they move to symbolize some movement on the Z axis. Another suggested that I add a more custom shape to the fire particle system to make the flames look more authentic.

While these are both great ideas, I am unfortunately unable to implement them due to time constraints but will keep them in mind for future projects.

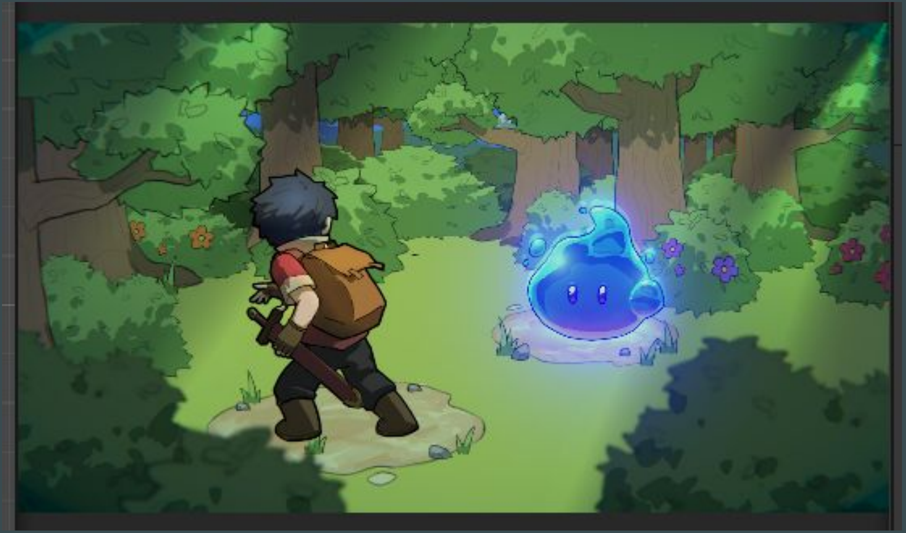


Post Reflection Thoughts:

I really liked how my scenes came out in the end but thinking back now, it probably wasn’t the best idea to use post processing colour adjustments to darken the scene as that also darkened the lights slightly.

I wanted to try making it so the slime particles in my main menu could leave a trail of particles behind as it moves across the screen but was unable to make it work in the end.

In future projects, I’ll try to research about the things I wanted to do in advance so there’s room for experimentation.



Reference list:

SpeedTutor. (2021, Dec 18). *Multiple Sprites With One Particle System (Unity Tutorial)* [Video]. Youtube. https://youtu.be/nv58IclhfmA?feature=shared

Willy Dev. (2023, Jun 27). *Post-Processing BASICS in Unity* [Video]. Youtube.

https://youtu.be/0n5UK2igodQ?feature=shared