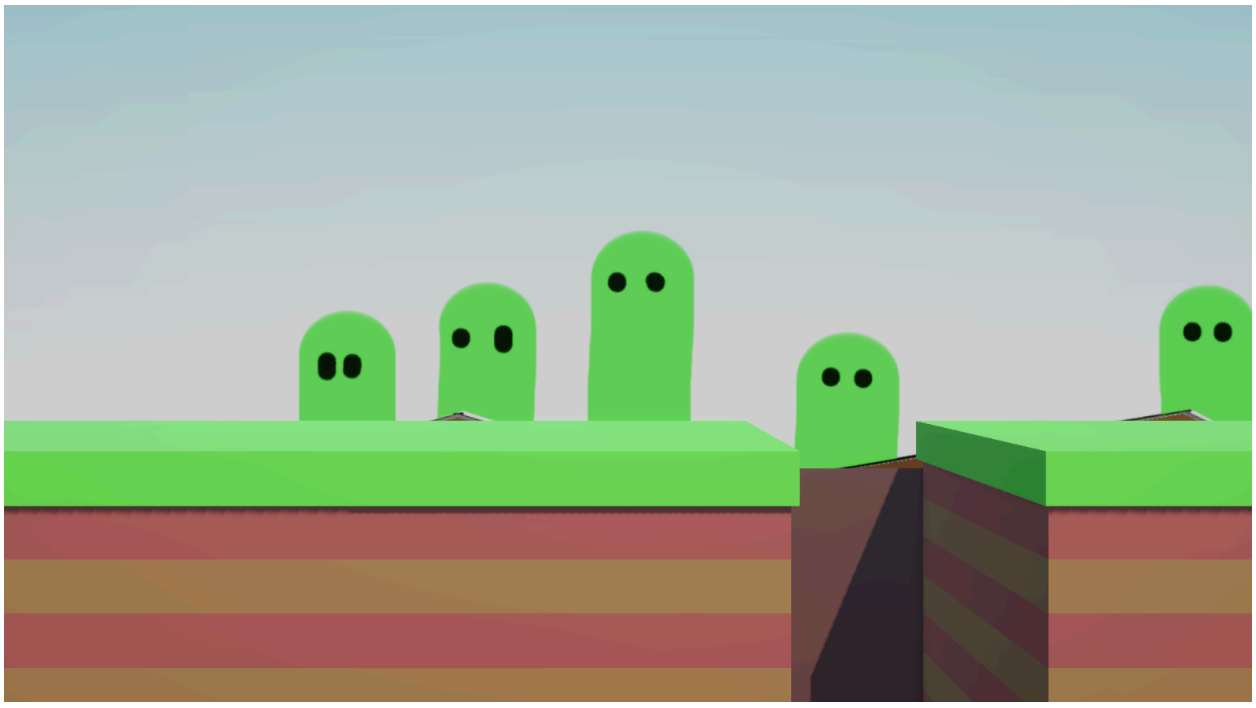
ReadMe

Ethan Schledewitz-Edwards
100908840

Github:
https://github.com/Ethan-Schledewitz-Edwards/CGExam

For some reason, my shaders do not build despite me receiving no warnings or errors. Ensure that you are using the Unity project to check if shaders work. Additionally, I was given permission to share a gif of my game running in engine to prove that they function

Scene
The scene I am attempting to enhance is taken from Super Mario World, in this scene Mario starts on the left hand side of the screen, and there is a pit that he can jump over.



Colour Correction

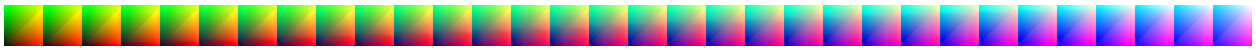
Note: LUTs can be toggled between using L

Colour grading works by replacing a colour value with another, based on the corresponding pixel found in a Look Up Table or LUT.

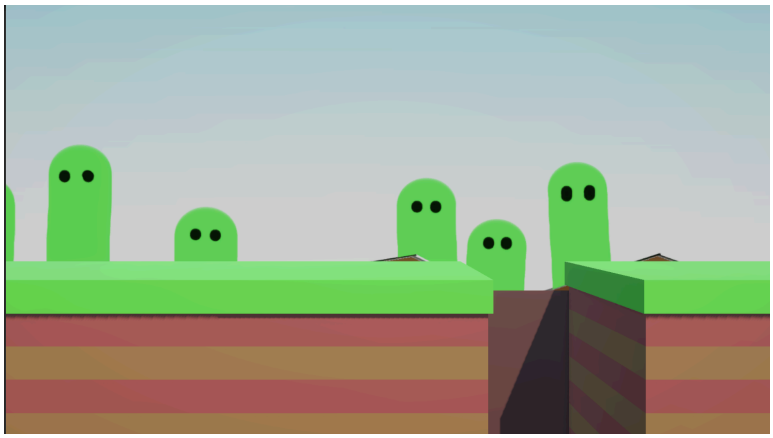
The Mario games often feature bright cheery atmospheres with very saturated colours. The default Unity colour grading does not provide you with this effect. Therefore, I have chosen to enhance the scene by adding a vibrant and saturated LUT that makes the colours pop more.

The output of my LUT shader is produced by taking the image from a separate camera called the “LUTCamera” as a texture. After the texture is rendered, this texture has its colours adjusted by a shader that uses the camera texture alongside a LUT and a contribution scalar. This material is applied to an image that is rendering in my UI which is then displayed on the screen.

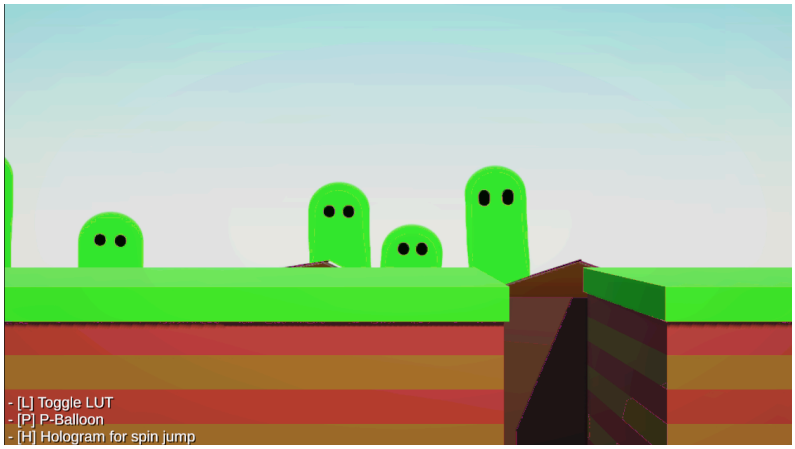
Saturated LUT:



Before:



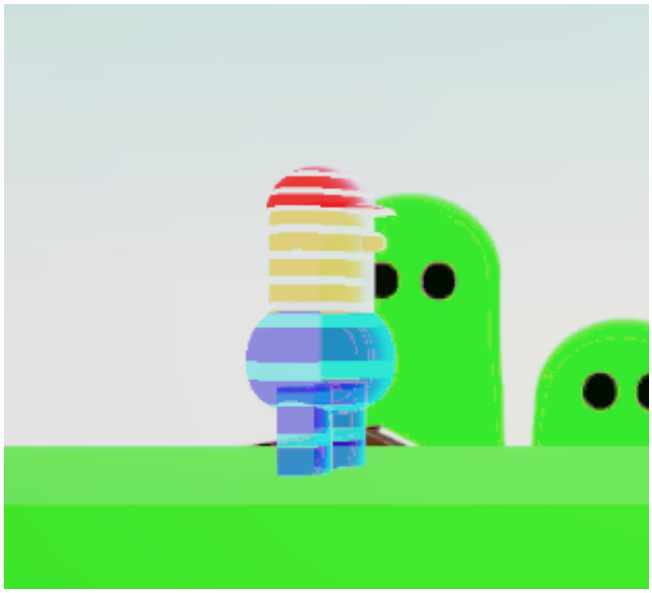
After:



Hologram

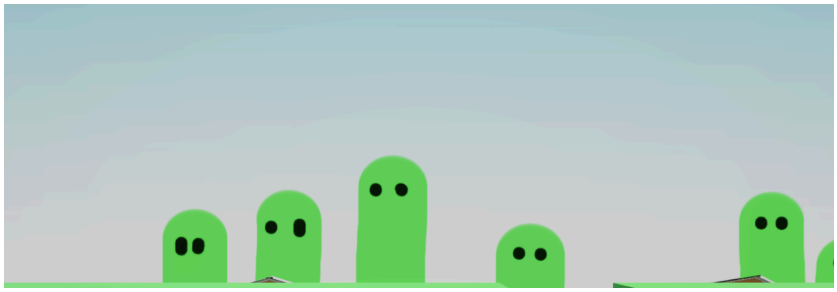
For my implementation of a hologram shader, I intend for it to be applied after mario spin jumps off of an enemy to signify that he is invulnerable. I think that a hologram represents this effect

well because it adds transparency to imply he cannot be hit, but it also provides depth through the rim effect mixed with the fresnel effect.



Properties:
_MainTex: The main texture applied to the model.
_LineColor: The colour of the lines applied to mario.
_FresnelColor: The color of the rim effect.
_RimIntensity: The strength of the rim lighting.
_FresnelPower: The power for the fresnel rim effect.
_LineSpeed: How fast the lines scroll over the mesh.
_LineFrequency: The spacing between the lines.
_Transparency: How transparent the material is.

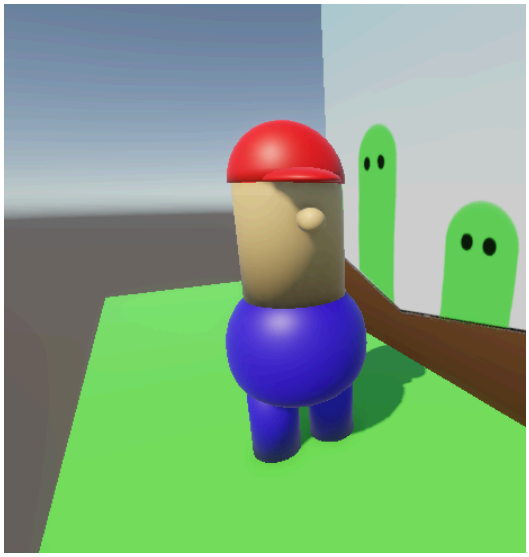
Scrolling Shader
I used a scrolling shader to make a scrolling background for my level. The background only scrolls on the X axis and does not evaluate Sine like the in class example. I removed the Y axis because it was unneeded since I only need the background to move from right to left, and I removed the evaluation of Sine since I don't want the scrolling to slow down and reverse its direction periodically.



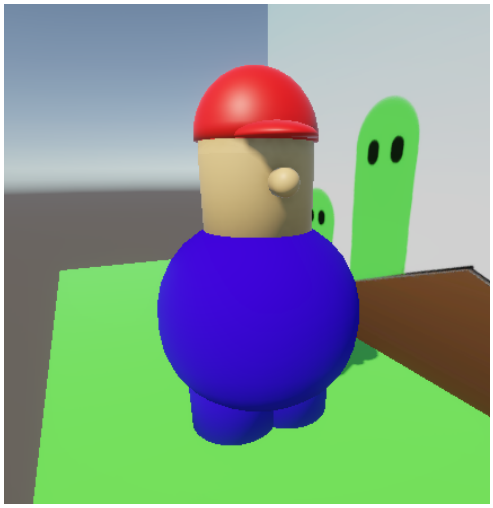
Properties:
MainTex: the main texture being scrolled
ScrollSpeedX: The scrolling speed along the X-axis.

Extrude Shader
For my implementation of an extrude shader, I tied it to the p-balloon power up. In Super Mario World, the player can pick up a power-up called a p-balloon. The p-balloon causes mario to inflate like a balloon and float. This inflation effect can be enhanced with an extrude shader by expanding Marios vertices like a balloon in a vertex shader. To make the effect not sudden, it can be done over a short period of time with a scalar to mimic the effect of a balloon being inflated. Additionally, I implemented Blinn-Phong lighting into my shader to match Unity's default lighting as best as I could.

Before P-Balloon:



After P-Balloon:



Properties:
_MainTex: The main texture used
_SpecularColor: The colour of the specular reflection, in my case this reflection is white.
_Shininess: How reflective the material is.
_Balloon: How much extrusion is applied

Hologram Diagram