ICG Assignment 1

By: Aidan Mohammed
ID: 100875307

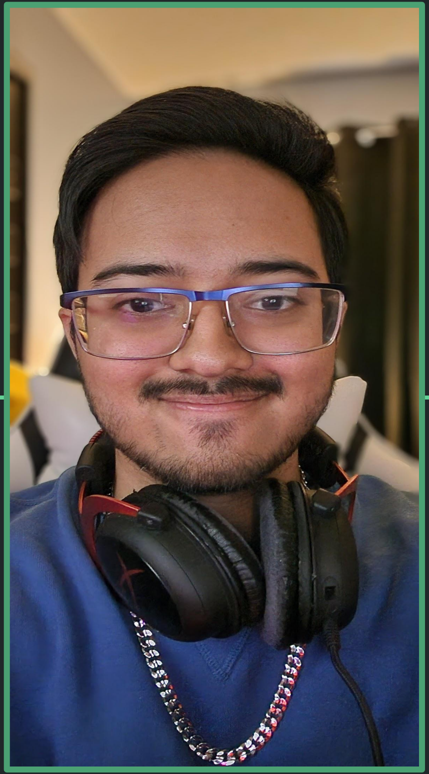


Table of Contents

Game

- Game Pitch
- Game Demo

Implementations / Why I added it

- Color Grading
- Lighting Options
- Shaders
- Other effects
- Final Notes

Game Pitch

- This game is a parkour survival game
- Escape a volcano with rising lava
- You have 3 lives, reach checkpoints, and make it to the top to win

The goal with the shaders is to learn how to create an effective scene of what it's like to be in a volcano.

Wii Sports Resort Volcano
(Reference for the vibe of the game)

Game Demo

Will be shown live in the video of recording this to help explain what the viewer is seeing during the game

Color Grading

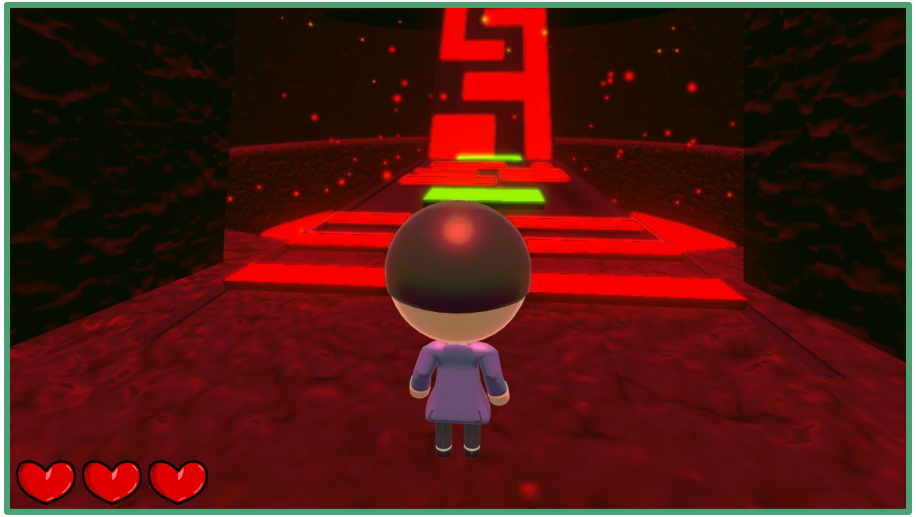
Color Grading

- Color Grading was implemented exactly as shown in class with my own twists for the scene
- After effects was used as I have 3 years editing experience.
- Adjusted the saturation, added a red tint, a deep glow, and increased contrast

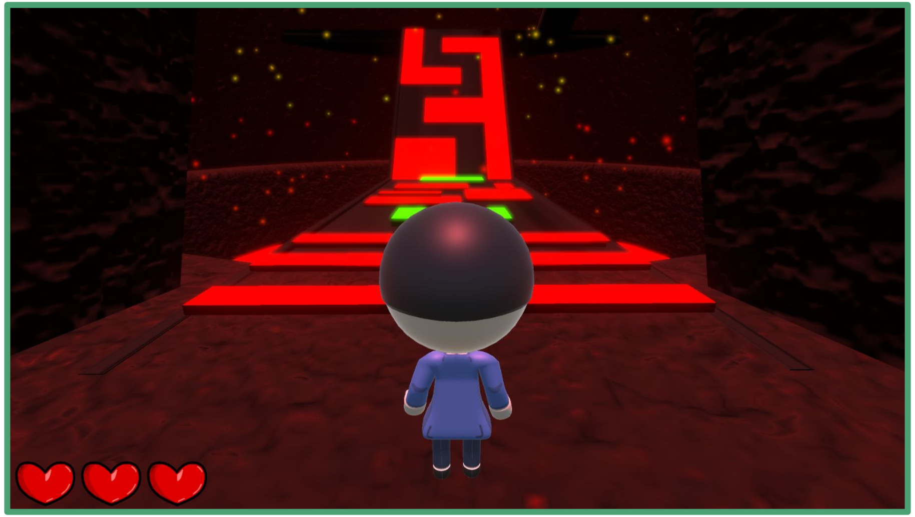
- **Comparison may be hard to notice, but didn’t wanna overdo it (Next Slide)**

Color Grading Difference

With



Without



The Color Grading is subtle but it is there. I added an orange/red tint, increased contrast.
The characters back shows the color grading better in comparison.

Lighting Implementations

Lighting

- There is a total of 9 shaders with 3 of them for each material in my game
- Can be toggled with 1, 2 or 3

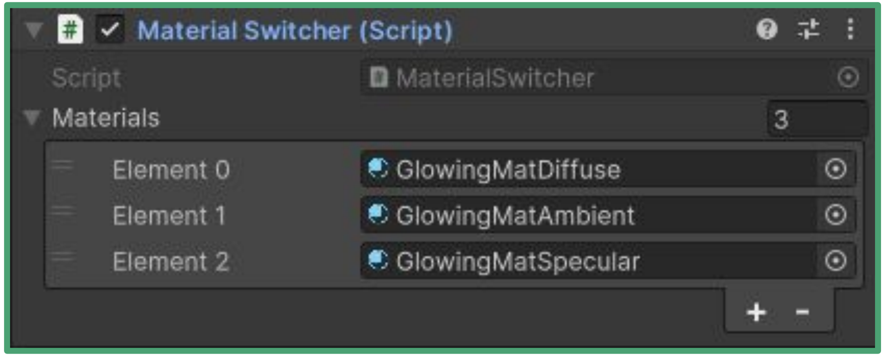
Lighting added:

- Simple Diffuse
- Diffuse Ambient
- Simple Specular

- **In the video, the first light you see is Simple Diffuse, then Ambient, then Specular in that order.**

Lighting

- The toggle is done with a script where you assign 3 materials
- I prefer diffuse lighting for the scene as it makes the colors pop, and the other we're done to showcase other variations



Material Switcher (Script)

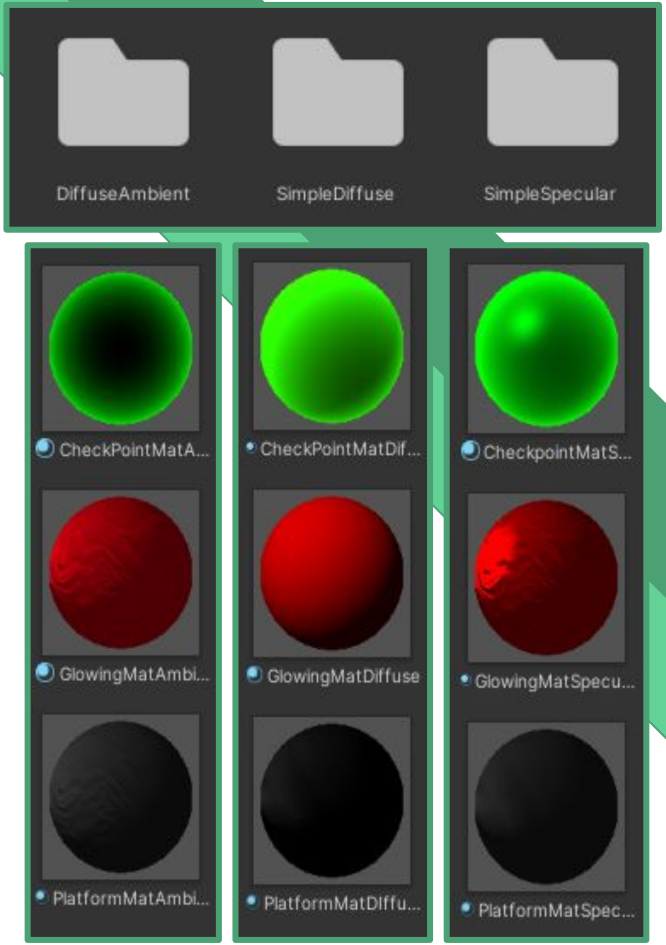
Script: MaterialSwitcher

Materials: 3

Element 0: GlowingMatDiffuse

Element 1: GlowingMatAmbient

Element 2: GlowingMatSpecular



Shaders

Shaders

- There are 3 materials each with their own light model as well as a shader that defines each one

Shaders added:

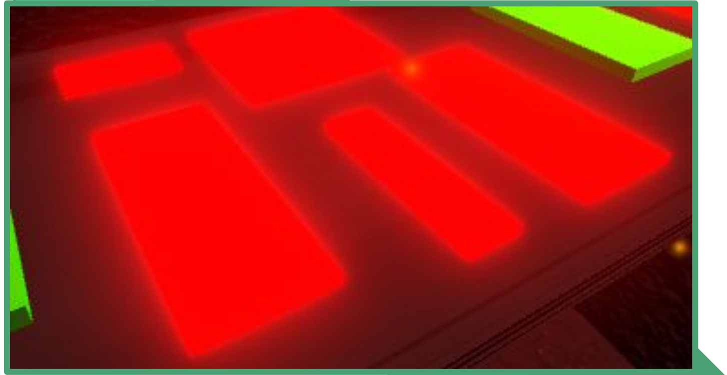
- Lava Shader (**Emissions** + Normal Mapping + Albedo Color)
- Checkpoint Shader (**Rim lighting** + Normal Mapping + Albedo Color)
- Simple Specular (**Bump Mapping** + Albedo Color)

- **Shader Scripts Next Slide**

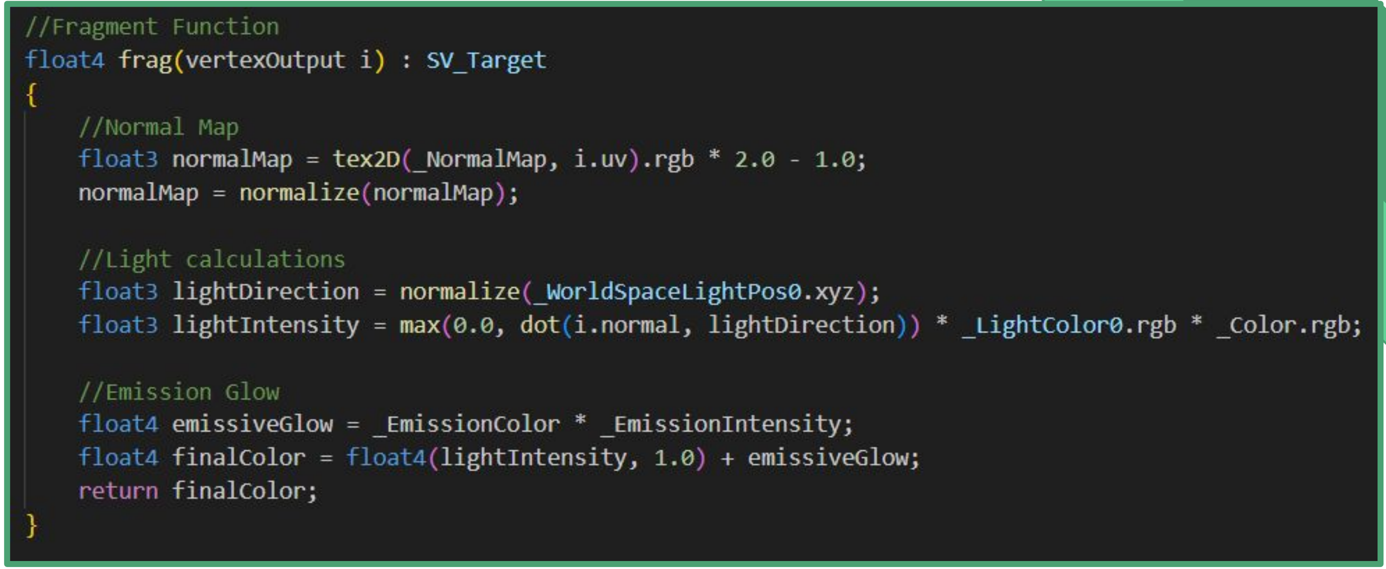
Purpose

- Lava to glow
- Used Emission to admit light from the object with an intensity
- The emission causes the normal map to barely be seen, but it’s very noticeable if you switch to Diffuse Ambient

I also added post processing bloom to expand on the emission to make it have an area glow effect



Lava Shader

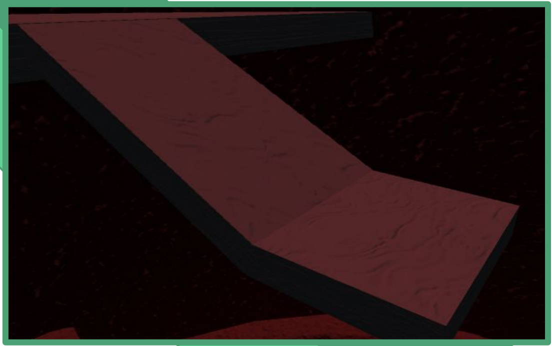


```
//Fragment Function
float4 frag(vertexOutput i) : SV_Target
{
    //Normal Map
    float3 normalMap = tex2D(_NormalMap, i.uv).rgb * 2.0 - 1.0;
    normalMap = normalize(normalMap);

    //Light calculations
    float3 lightDirection = normalize(_WorldSpaceLightPos0.xyz);
    float3 lightIntensity = max(0.0, dot(i.normal, lightDirection)) * _LightColor0.rgb * _Color.rgb;

    //Emission Glow
    float4 emissiveGlow = _EmissionColor * _EmissionIntensity;
    float4 finalColor = float4(lightIntensity, 1.0) + emissiveGlow;
    return finalColor;
}
```

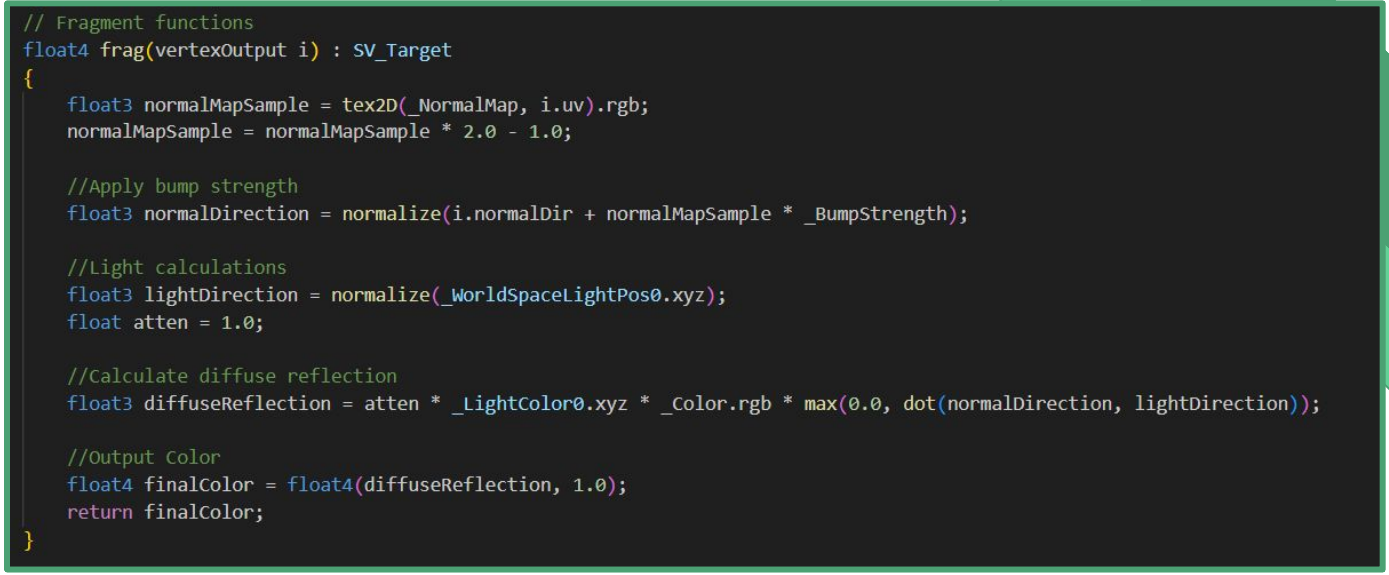
Purpose

- Give it a normal map and adjust the strength of it with a basic albedo color
- Create and object which the player can detect as safe or the floor

Since this is the base platform for my game, I need it to be similar to the volcano.



Platform Shader

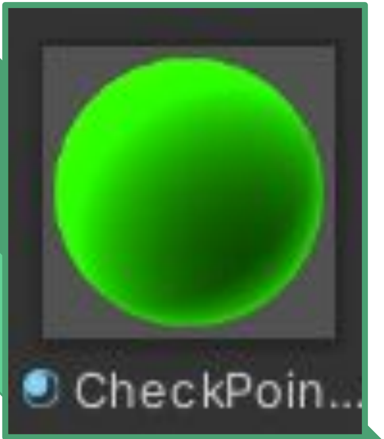


```
// Fragment functions
float4 frag(vertexOutput i) : SV_Target
{
    float3 normalMapSample = tex2D(_NormalMap, i.uv).rgb;
    normalMapSample = normalMapSample * 2.0 - 1.0;

    //Apply bump strength
    float3 normalDirection = normalize(i.normalDir + normalMapSample * _BumpStrength);

    //Light calculations
    float3 lightDirection = normalize(_WorldSpaceLightPos0.xyz);
    float atten = 1.0;

    //Calculate diffuse reflection
    float3 diffuseReflection = atten * _LightColor0.xyz * _Color.rgb * max(0.0, dot(normalDirection, lightDirection));

    //Output Color
    float4 finalColor = float4(diffuseReflection, 1.0);
    return finalColor;
}
```

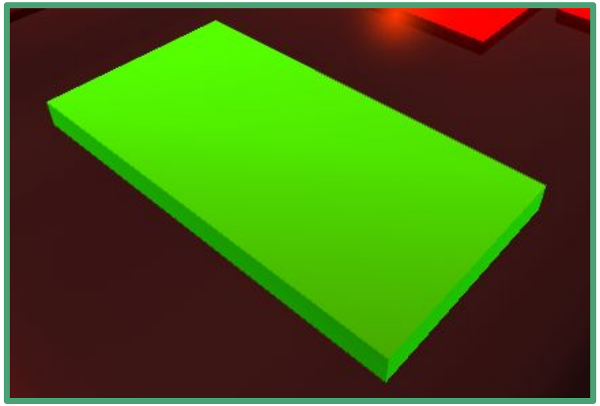
Checkpoint Shader

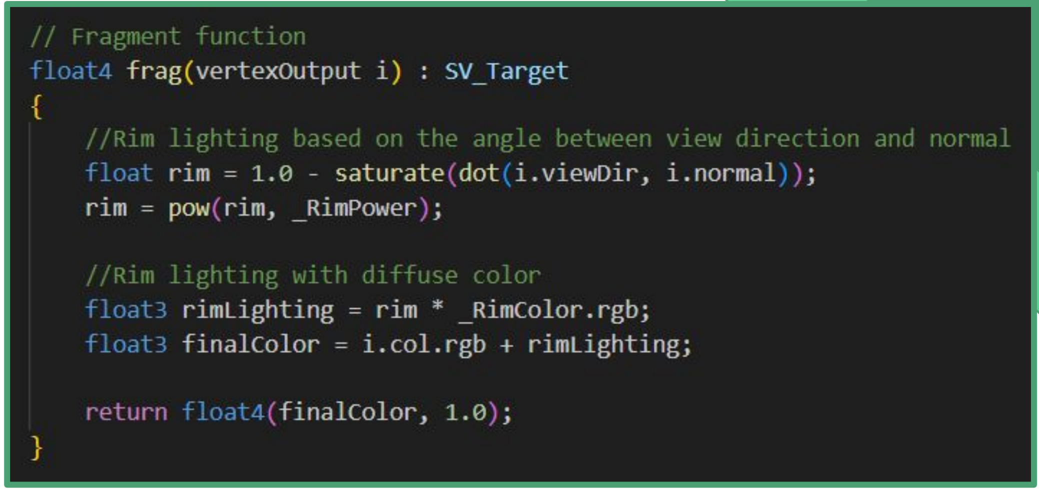
Purpose

- Make a noticeable area for the player to interact with it
- Glowing edge gives the effect that it can be interacted with/is important



Since I didn’t use a round object the item’s Rim Lighting can be hard to see at times





```
// Fragment function
float4 frag(vertexOutput i) : SV_Target
{
    //Rim lighting based on the angle between view direction and normal
    float rim = 1.0 - saturate(dot(i.viewDir, i.normal));
    rim = pow(rim, _RimPower);

    //Rim lighting with diffuse color
    float3 rimLighting = rim * _RimColor.rgb;
    float3 finalColor = i.col.rgb + rimLighting;

    return float4(finalColor, 1.0);
}
```

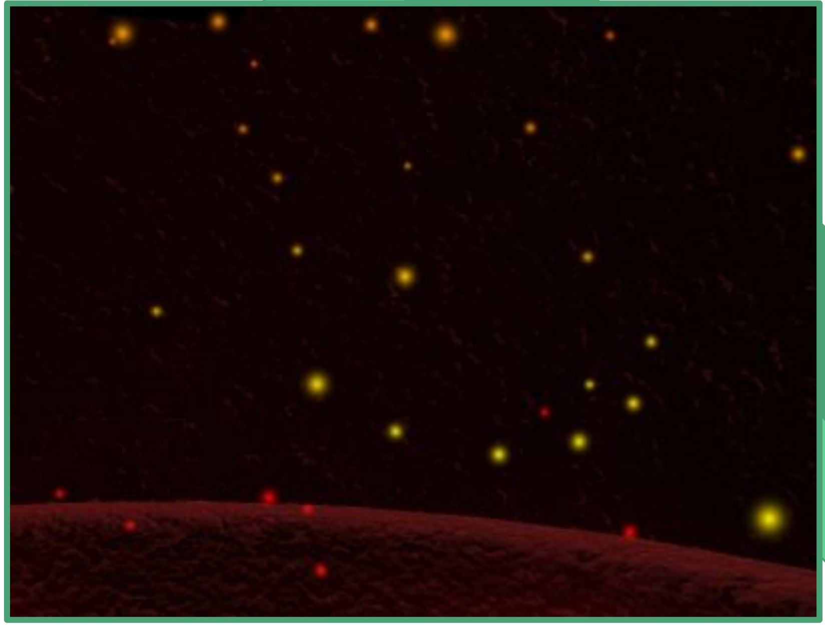
Extra Effects

Extra Effects

Particles

- I added a small particle effect using Unity’s particle system, to visualize floating embers.
- They trail up slowly to the top, have a gradient from red to yellow, and slowly decrease in size

- **Nothing to crazy**



All material that **ISN’T MINE** within this game

- Lava Texture (**https://www.filterforge.com/filters/1316-normal.html**)
- Simple Diffuse, Diffuse Ambient, Simple Specular base scripts **(From lecture)**
- Color Grading base scripts **(From lecture)**
- Generative AI code for checkpoint system

There was NO generative AI used FOR ANY scripts related to ICG. All Shader scripts we’re put together via the assistance of the lecture materials

Thank you for Listening