# Project Overview

[Git Hub Document]

This project is for use in an educational assessment and is not permitted for reuse.

Project name: JM Depth Shader

This project is a Unity 5.6.0f3 shader for object greyscale depth visualization.

# Project Analysis

## Project Requirements

Table 1: Outline of user tasks for: JM Depth Shader.

|  |  |  |  |
| --- | --- | --- | --- |
| JM Depth Shader Unity | User Story | Tasks | Time Allocated |
|  | Camera Setup Components | Default Unity components.  Shader Case Script. | 1 min |
|  | Shader Case Script | Setup as storage for the shader script. | 5 min |
|  | Shader Script | Full screen effect.  Turn the image to greyscale,  Detect the Z depth of each pixel,  Each pixel is slightly darker the further from the camera. | 30 min |

## Use Case

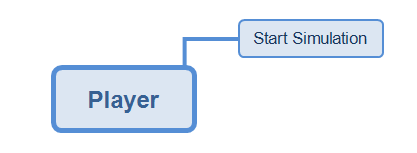
****

Figure 1: Use case flowchart between player and system.

# Shader Design Document

## Architectural Design

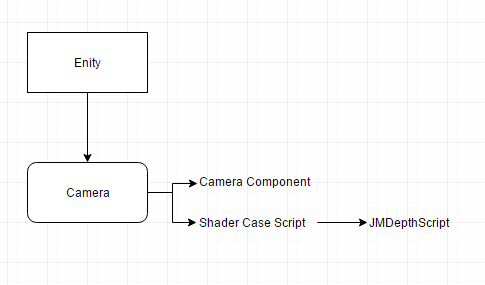


Figure 2: Architectural Design map showing all components and their relations in the game.

## Component Design (Required Only)

#### Unity Camera Component

This component captures and displays the world to the player.

In this project it is used to draw the game world and allow the user to see the shader effect.

#### Shader Case Script

This component acts as the holder of the custom shader that detects depth and

|  |  |  |  |
| --- | --- | --- | --- |
| Entities | Required Components | | |
| gameObjects | Camera Component | Shader Case Script |
| Camera | Yes | Yes |

## Algorithm Design

Scripts in this game are named as follows: ShaderCaseScript and JMDepthScript.

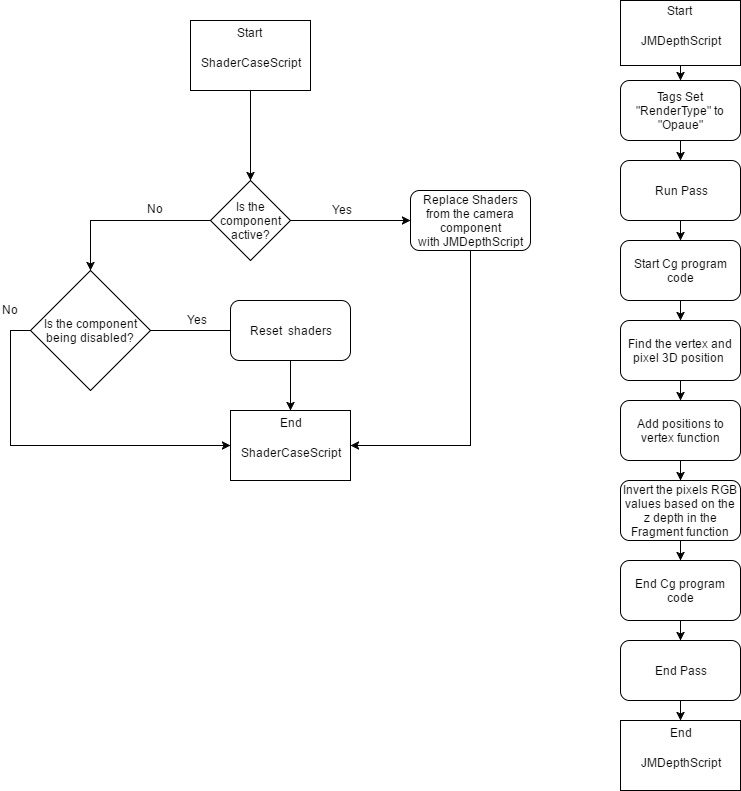


Figure 4: Flowchart of progression.

### Pseudo code for scripts

#### ShaderCaseScript

1. initialize public shader **ReplacementShader**
2. FUNCTION **OnEnable**
3. IF **ReplacementShader** is not empty THEN
4. GET **Camera** component AND apply **ReplacementShader** WITH Tag "**RenderType**"
5. END IF
6. END FUNCTION
7. FUNCTION **OnDisable**
8. GET **Camera** component AND RESET **ReplacementShader**
9. END FUNCTION

#### JMDepthScript

1. TAGS "**RenderType**" = "**Opaque**"
2. PASS
3. CGPROGRAM
4. #pragma vertex vert
5. #pragma fragment frag
6. #include "UnityCG.cginc"
7. STRUCT **appdata**
8. Initialize float4 **vertex** with **POSITION**
9. END STRUCT
10. STRUCT **v2f**
11. Initialize float4 **vertex** with **SV\_POSITION**
12. Initialize float **depth** with **DEPTH**
13. END STRUCT
14. Initialize VERT FUNCTION **v2f** TAKES **v2f** TAKES **appdata**
15. Initialize **v2f** variable **o**
16. Give **o** a vertex **position**
17. Give **o** a depth within the camera view field
18. RETURN **o**
19. END VERT FUNCTION
20. Initialize FRAG FUNCTION TAKES **v2f** TAKES **SV\_Target**
21. Initialize variable float **invert** with 1 minus the depth of **v2f** pixel
22. RETURN fixed4 RGB as **invert** and A as 1
23. END FRAG FUNCTION
24. ENDCG
25. END PASS

# Implementation and Output

To use: add ***ShaderCaseScript*** as a component of the camera game object and add ***JMDepthScript*** as the "Replacement Shader" within the component.

In the example provided the only modifications to preset assets are to the Camera components "Field of View"(now set to "50"), "Clear Flags"(now set to "Depth only") and the "Clipping plains"(now set to "85"). Please fill free to change these settings.

Note: this shader only greyscales objects by design and not Unity's skybox.

#### ShaderCaseScript

The Script acts as a component for the camera object and is a container for the Cg shader script.

##### Code:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

[ExecuteInEditMode] // Allows the program to run in edit mode

public class ShaderCaseScript : MonoBehaviour { // declares that this class inherets from MonoBehaviour

public Shader ReplacementShader; // Creates a placeholder for the Cg shader script as well as a variable name for use

void OnEnable() // When the script is activated, use the replacement shader instead of Unity's default shaders

{

if (ReplacementShader != null) GetComponent<Camera>().SetReplacementShader(ReplacementShader, "RenderType");

}

void OnDisable() // When the script is disabled, reset the shaders

{

GetComponent<Camera>().ResetReplacementShader();

}

}

#### JMDepthScript

This script is the shader script that detects the depth of each on screen pixel as well as changing its colour between white and black based on the pixels depth.

##### Code:

Shader "Hidden/JMDepthScript"

{

SubShader{

Tags{ "RenderType" = "Opaque" } // Give the Tag "RenderType" a type of "Opaque"

Pass{ // Runs a Pass over the program allowing the geometry of a GameObject to be rendered once

CGPROGRAM // The start of the ative Cg program

#pragma vertex vert // The program includes a Vertex shader function

#pragma fragment frag // The program includes a Fragment shader function

#include "UnityCG.cginc" // The program includes the Unity shader Libraries

struct appdata { // A function that contains a position as a float4

float4 vertex : POSITION;

};

struct v2f { // A function that has a pixels position and depth

float4 vertex : SV\_POSITION;

float depth : DEPTH;

};

v2f vert(appdata\_base v) { // A Vertex function that has the position, normal and one texture coordinate for a vertex and referred as "v"

v2f o; // Pulls information from v2f and is referred as "o"

o.vertex = UnityObjectToClipPos(v.vertex); // The "position" variable in the v2f function is now a "v" position in screen space

o.depth = -mul(UNITY\_MATRIX\_MV, v.vertex).z \*\_ProjectionParams.w; // The "depth" variable in the v2f function is now a "v" position based on its Z position in screen space

return o;

}

fixed4 frag(v2f i) : SV\_Target{ // A Fragament function that has the v2f properties and referred as "i" as well as the current pixel colour

float invert = 1 - i.depth; // Create an inverted float from the v2f depth

return fixed4(invert, invert, invert, 1); // Invert the RBG values and set the A to "1"

}

ENDCG // The end of the ative Cg program

}

}

}