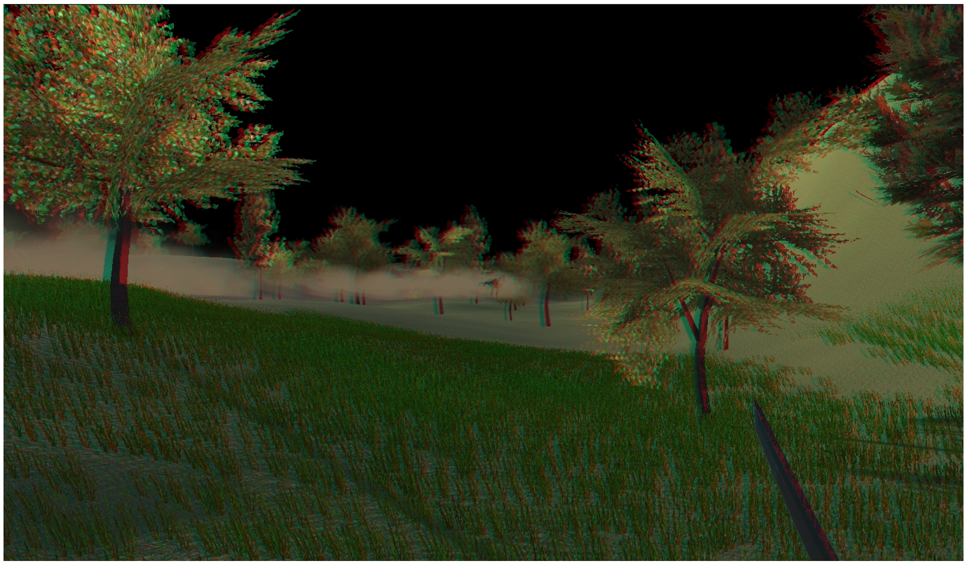
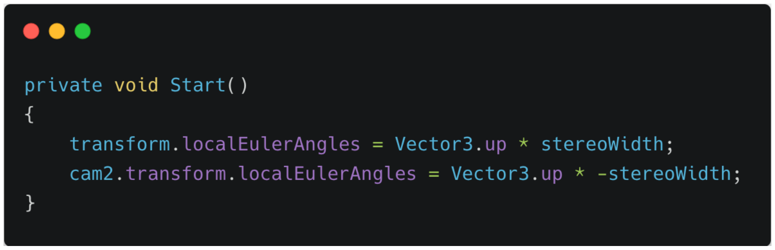
Assignment 3 Report 3 — Spencer Churchill

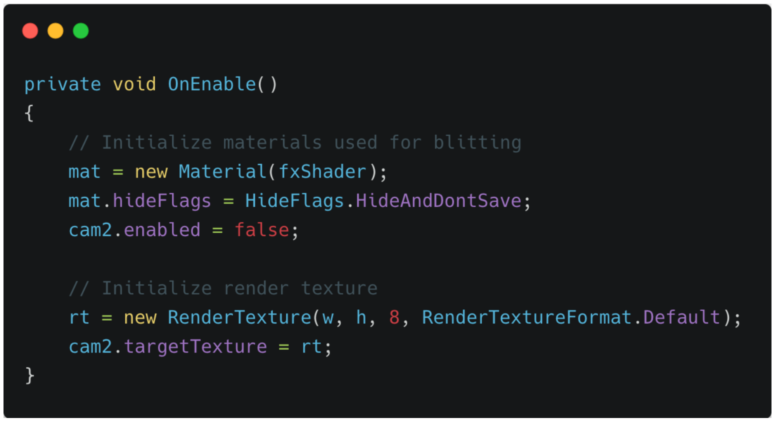
- Anaglyph 3D Rendering
  - 

- Paragraph:
  - I implemented anaglyph rendering with the help of Michael Skec's YouTube video and Daniel Ilett's image effects series. This rendering is performed in a series of basic steps: first, a second camera is instantiated to create a parallax effect; second, the main and secondary cameras are slightly skewed to emulate two eyes; and third, a red and blue texture is applied on the main and secondary cameras respectively. These steps are shown in the code snippets below. The cameras are created with Unity, the Start() method adjusts the cameras' angles, the OnEnable() method applies the red and blue anaglyph shader to the cameras, and the OnRender() method pushes these changes to the main camera.

Assignment 3 Report 3 — Spencer Churchill

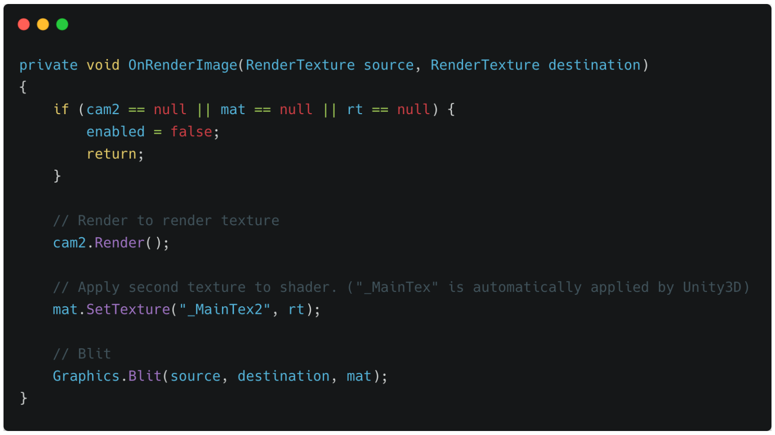
- Code Snippets:
    - 

```
private void Start()
{
    transform.localEulerAngles = Vector3.up * stereoWidth;
    cam2.transform.localEulerAngles = Vector3.up * -stereoWidth;
}
```

    - 

```
private void OnEnable()
{
    // Initialize materials used for blitting
    mat = new Material(fxShader);
    mat.hideFlags = HideFlags.HideAndDontSave;
    cam2.enabled = false;

    // Initialize render texture
    rt = new RenderTexture(w, h, 8, RenderTextureFormat.Default);
    cam2.targetTexture = rt;
}
```

    - 

```
private void OnRenderImage(RenderTexture source, RenderTexture destination)
{
    if (cam2 == null || mat == null || rt == null) {
        enabled = false;
        return;
    }

    // Render to render texture
    cam2.Render();

    // Apply second texture to shader. ("_MainTex" is automatically applied by Unity3D)
    mat.SetTexture("_MainTex2", rt);

    // Blit
    Graphics.Blit(source, destination, mat);
}
```

- Resources:
    - [Unity3D] Writing A 3-D Anaglyph Effect...
    - Ultra Effects | Part 4 - The Next Dimension Fully-featured 3D with glasses!