Victoria Lima
vmlima

Part D: Discuss a Visual Effect

https://www.youtube.com/watch?v=clqndE8eapI

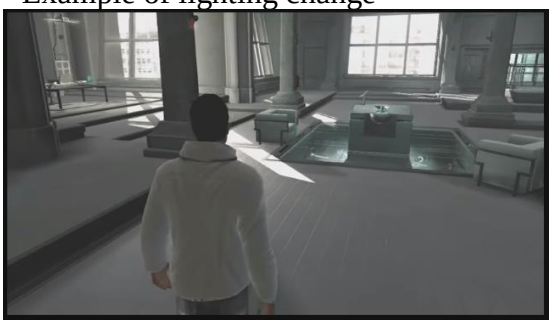
The effect I chose is the Eagle Vision ability effect, specifically from the first Assassin's Creed game. Assassin's Creed 1 is the game that got me interested in video games and I remember when it first came out, I thought the effect was really cool. While the newer games in the series greatly improved on the appearance and functionality of the ability, I think the first game has an interesting combination of effects. When the Eagle Vision ability is activated, the camera shifts from third person perspective into first person and the world becomes grayscale, then it highlights different NPCs. Targets, NPCs central the current objective are highlighted in gold, allies are highlighted in light blue, enemies are highlighted red and hiding spots are outlined. I think to achieve this portion of the effect, two shaders, or two passes would be needed. One shader would be created to turn the view grayscale, and another would be made to outline and highlight NPCs according to some information like a tag. This outline/highlight might be similar to the outline shader shown in class.

While the previous effects are the man portion of the Eagle Vision ability, the two things I find most interesting are how the effect interacts with lights and how it interacts with movement. To start with, the effect turns light sources, like windows, solid white. They become extremely contrasted with the surrounding environment, but the hard shadows they are casting become extremely soft.[1] The shader might change the texture on the windows into an emissive white texture. In doing so, it might block out the hard-shadow casting directional light and replace it with a softer light emitting like a point light from the window. The other effect I found interesting is the nauseating motion blur applied to the camera when the player looks around with the ability activated.[2] This part of the effect seems to be creating copies of the last X number of frames and overlying them with a smaller alpha value on top of the current frame.

I was not able to find a site that showed how to make a shader true to every aspect of the Eagle Vision effect. However, in the previous two paragraphs, I broke up the Eagle Vision effect into 4 parts: a grayscale effect, an outline/highlight effect, a lighting effect, and a motion blur effect. The grayscale shader seems fairly straightforward: create a custom shader and change the albedo value to o.Albedo = (c.r + c.g + c.b)/3. However, as I was searching, I noticed that the "greyscale" in this effect has a slight blue-green tint to it while the shaders I was finding stayed true to black and white, so this effect is closer to a monochromatic blue color scheme. I think this can be achieved just by increasing the blue value slightly. For the highlight effect, the closest

match is the shader shown in a previous class, however that shader only creates the outlines of an object. A shader tutorial created by The Lazy Dev[3] includes the highlight, however this effect only works when an object is behind a wall. Combining both would most likely achieve close to the same effect as the one in the game. I couldn't find any tutorials on the lighting part of the effect. It's probably my fault I didn't find anything because I don't know what the effect is called and I'm not sure what it is even doing. I realized that this effect is similar to the "Don't Clear" flag on cameras used in Unity. I didn't find a tutorial however I did find a post[4] that shows how one might use the "Don't Clear" flag to your advantage and it created a similar effect to this one. The post points out that a mixture of the flag and a blur shader could be used to achieve the desired effect.

[1] Example of lighting change

 vs 

[2] Example of motion blur

[3] http://thelazydev.net/58_/shading-toolbox-2-sillhouette-or-eagle-vision/

[4] https://stackoverflow.com/questions/38531818/usage-of-dont-clear-in-clear-flags-property-of-camera