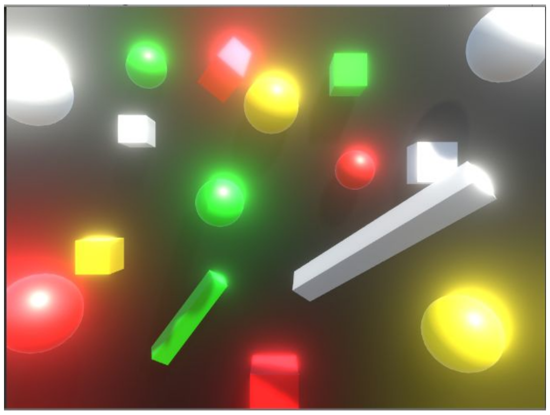
Leona Craig

Kelsey Fink

CS5963

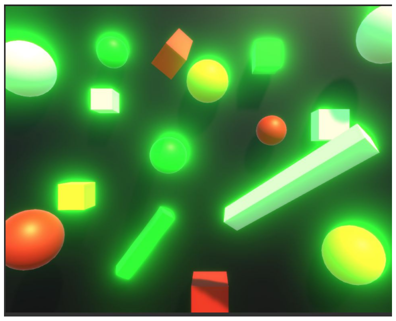
Assignment 3 Report

We began by placing random shapes in our scene in front of a black plane. We used our custom shader to replace the default material when we blit. We then downsampled the displayed data and progressively upsampled back to the original resolution depending on how many iterations we made in the downsampling process. We originally began with bilinear sampling and then improved the result by upgrading to box sampling to increase the sampling size and create a smoother effect. The final step was to use additive blending to include the original texture in with new texture. We then selectively picked which pixels to include in our bloom effect through our shader, specifically excluding dark pixels. We then added a soft threshold slider to create a smoother transition within the effect and created an intensity slider to fade in and out of the bloom effect.

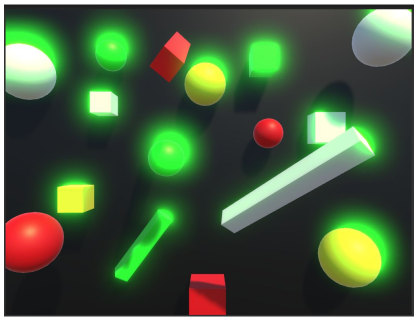


For the modification, we decided to add a tint to the bloom effect. We recalled that earlier in the tutorial, they added a red filter to every object in the scene by multiplying the text2D return object by the red values. We did not want the effect to change every object in the scene like this code did, so we tried placing the same concept in other passes. This still was not creating the effect we desired, so we tried placing it in the SampleBox function that's called in all of the passes. This was the solution to our desired effect, which was to create a tinted glow. We then wanted to make the color a neon-green shade so it would give the objects a traditional, radioactive glow appearance. The result was exactly what we wanted, however it was interesting because the modification did not affect the red material objects to the same degree as the others. This modification could also work really well for christmas lights. Below are a few screenshots to show the modification at different intensities.

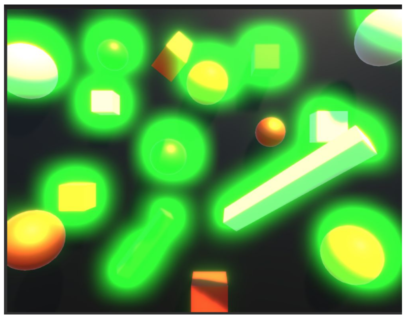
Increasing all of the sliders



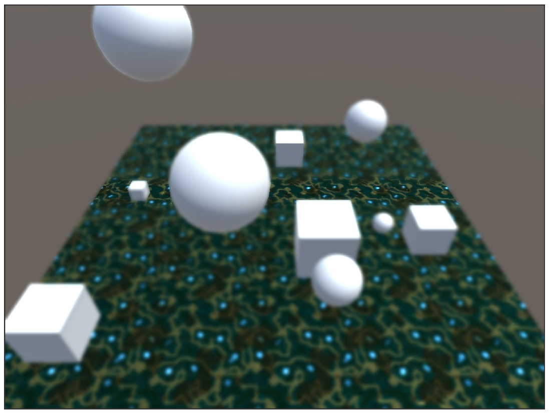
Just increasing the intensity a little



Lowering the threshold



For the second tutorial, we began by creating a scene with some objects and then copied the bloom script and shader to the new project and edited out the bloom specific parts. Then, we created a circle of confusion to measure how out of focus the projection of a point is by using depth, focus distance, and focus range in the circle of confusion equation. To implement the Bokeh effect, we used the approach where each fragment accumulates colors from all texels that may influence it which involves many texture samples. At first this created a square bokeh and we wanted it to be more circular with a disc shape and disregarding samples that have too strong of an effect. To create the blur, we blit to a downsized resolution and then blit again to bring it back to the original resolution and as we downsample and upsample the bokeh effect, we also do it to the circle of confusion to combine the two. We then blend the parts that are in full resolution with the parts that are in half resolution to maintain the focus.



To begin the Low Level FFR portion (Part 2.2), we set up the toggling by importing the bloom shader and script over into this project as a placeholder. We then added the bloom script as a component to the camera, just like in Depth of Field but we set the default to disabled. Then, we made a new script that controls the toggling. The script first sets the bloom script to disabled and then on update, the script checks if a key press has happened. If the key press happened, it checks which script is currently active and swaps them.

After this, we found that we still did not have enough experience working with shaders to be able to figure out how to manipulate the depth of field code to make the blur appear on screen in the way that we needed it to. We worked on this part alone for over a week but in the end were not able to make progress in moving the blur effect to a certain place on screen. We sought help on the discussion board, we went to TA hours, we emailed the professor, and still weren't able to figure out exactly how shader syntax works. From this, we were able to get some general ideas of how it could work though based off how other people described accomplishing it. In the script, take the screen width and height to be given to the shader. From the shader, use i.pos() to check the position of each pixel that is being rendered. We would then compare this pixel's position to the knowledge we have about the screen width and height to determine if it is in the section that we would want to blur and if so, how much. Based on this classification, we would then create 3 passes, one for each level of resolution, and blit those passes depending which part of the screen it’s affecting.

Since we weren’t able to get this working though, we did try a few back up plans. The first being trying to create a UI panel and blur that. First we tried just dragging the given depth of field shader directly onto the panel. This did not work. Then we tried googling how to blur a UI panel in Unity. This was more complicated than anticipated. Everyone’s solution online was to once again just write a shader, which brought us right back to the original problem.

Then we briefly tried to see if we could attach the depth of field shader to other types of game objects, particularly, a cube. This did not work. We then tried making a custom material with the depth of field shader to attach to the cube. This also did not work.

After this, we tried adding a second camera into the scene to see if we could project the blur from one camera and view it from the other camera. In a shocking turn of events, this too did not work.

Overall, we really enjoyed the majority of the assignment and getting to see the power of shaders, however, we felt that the tutorials were too advanced to be able to learn the basics of post processing effects. We feel like we put all of our effort that we possibly could have into trying to teach ourselves a skill that is out of our experience. Despite our struggles with Low Level FFR, we accomplished the majority of the assignment and still learned a lot from the part that we could not completely finish.