Sean Higgins

November 2, 2017

CS 562 – Adv. Real-Time Graphics

Prof. Herron

CS 562 Adv. Real-Time Graphics – Proj. 2 Report

Overview:

As an addition to the previous project, I’ve extended and added upon the basic, unfiltered shadow algorithm that I’d implemented in previous semesters. I’ve implemented exponential shadow mapping over top of it, allowing for smooth-edged shadows, instead of the harsh, all-or-nothing edges from before.

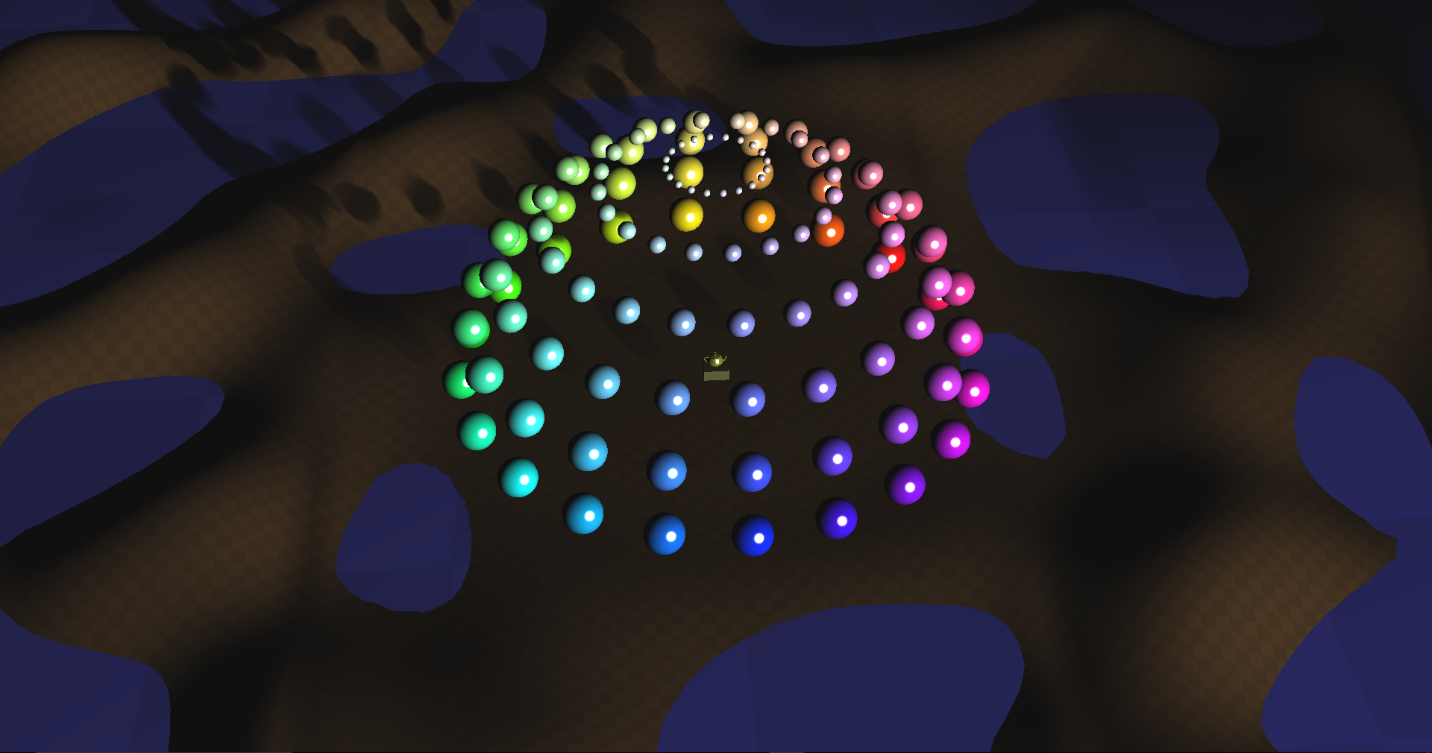
Exponential Map:

The use of storing an exponential value into our shadow map instead of the pure depth is twofold. Firstly, an exponential curve lets us approximate the actual binary shadow decision. In the original shadow check, a pixel was either purely in shadow or purely lit, with no gradient in-between. An exponential curve, however, can smoothly curve from the same ranges—0 for shadowed, 1 for lit, assuming the various depths involved in the calculations have been normalized—and also inherently curves smoothly down from one value to the other. The other major benefit of an exponential curve is that we can also add in constants to scale up and down how smooth or sharp the shadow edges can be.

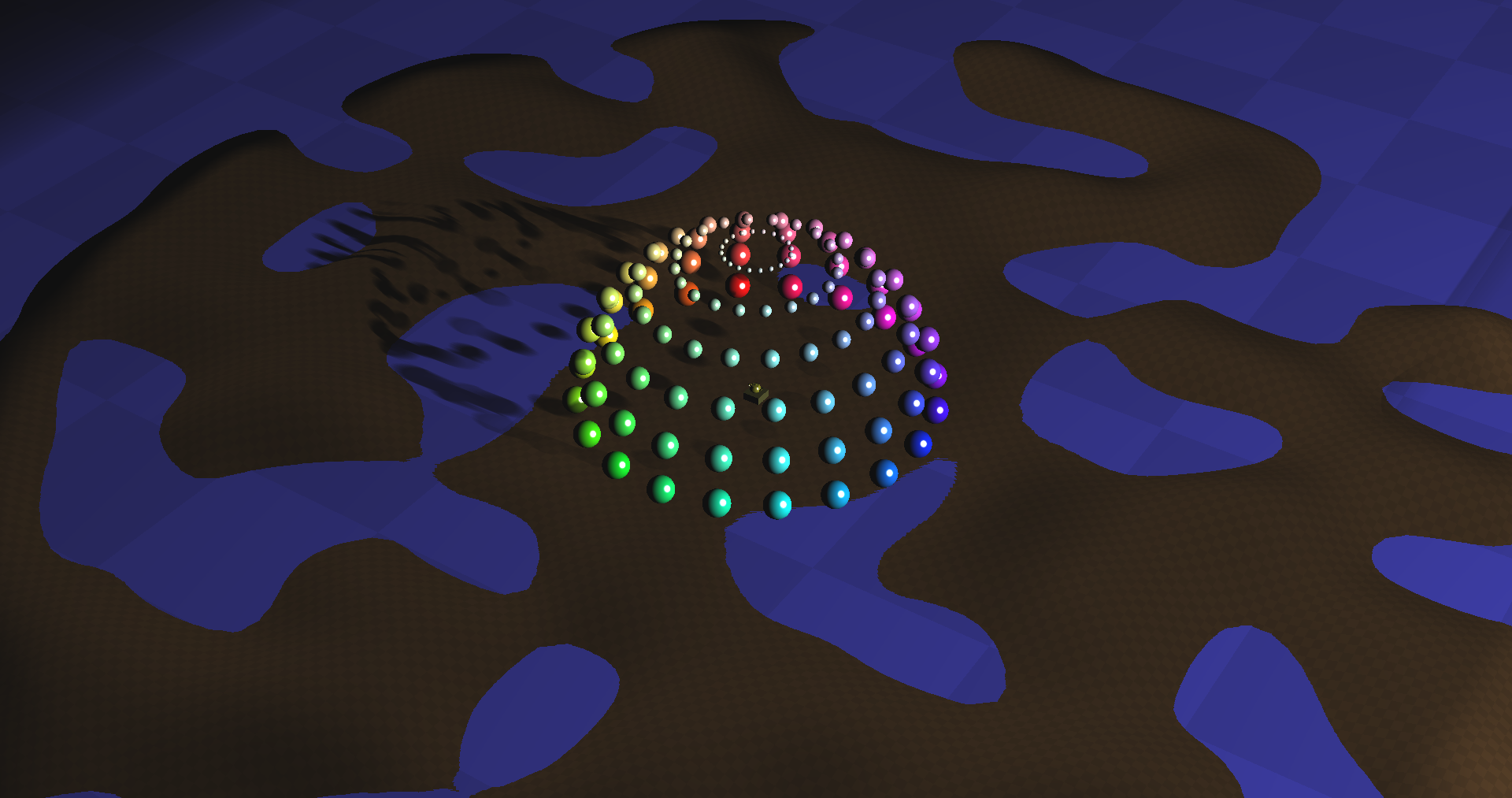
Gaussian Blur:

The second major part of the exponential shadow map is the addition of a Gaussian blur filter to the stored, initial depth values. This smooths out the shadow edges even more, and it is very efficient and simple to do in parallel with the help of a compute shader—even if they can be a bit tricky to set up properly. We generate and normalize the Gaussian weights during the initialization phase, then pass and use them in the compute shader, compressing a O(n^2) algorithm into O(2n) via two passes over the weights.

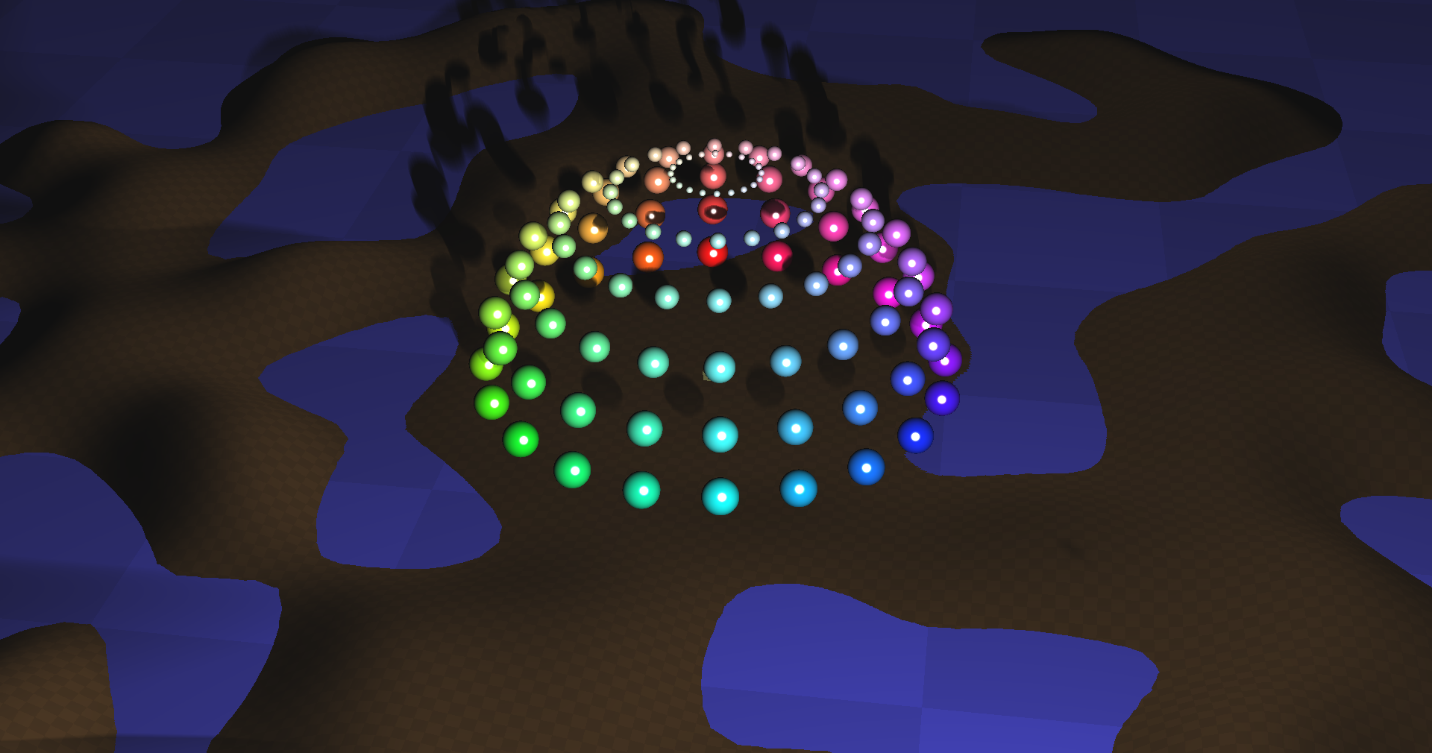
Blurred Final Outcome – C = 40



Blurred Final Outcome – C = 60



Blurred Final Outcome – C = 80



Depth Map Log – Blurred



Depth Map Log – Unblurred

