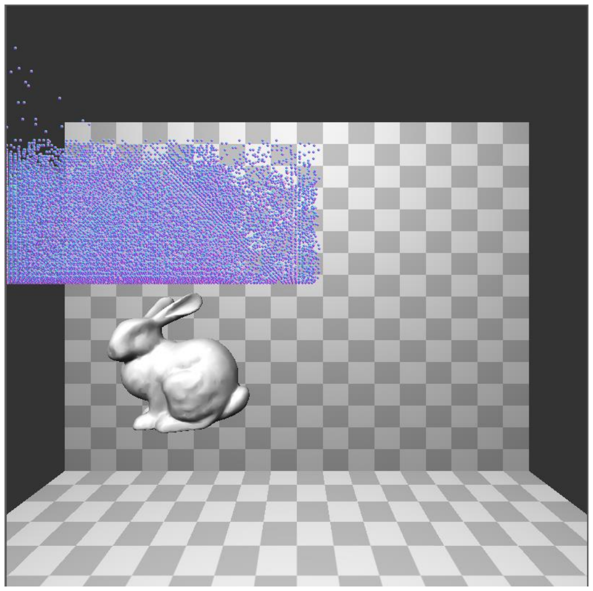
Final Project Progress Report

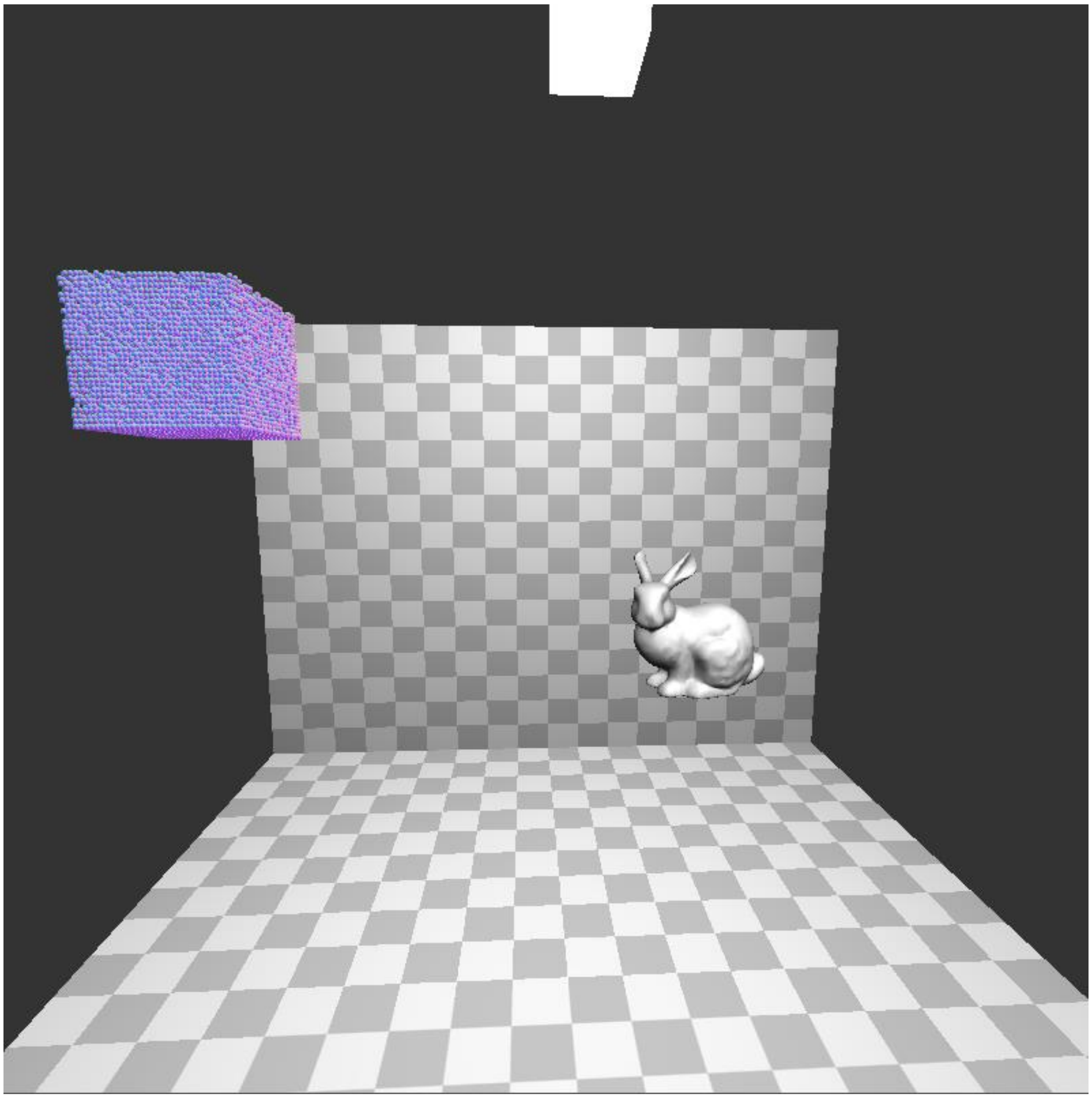
CS 6610-001 Spring 2019

Position Based Fluids

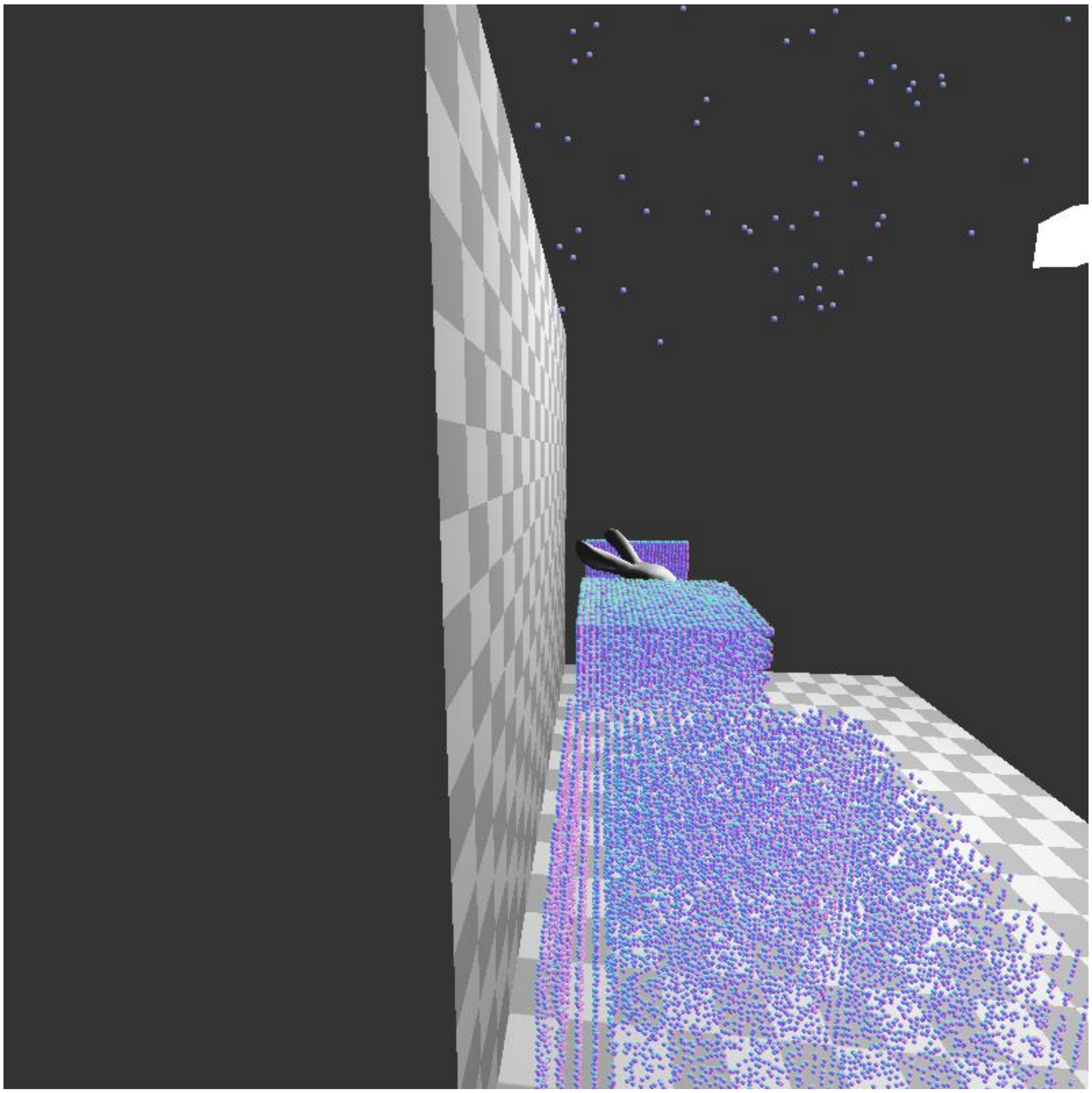
Ultimately in my final implementation I wasn't fully able to implement the fluid based solver described by Miles Macklin. Given a few more days, I think I could have gotten pretty close, but it's a little advanced for-time GL programmer. Along the way I implemented several portions of the algorithm which led me to be able to produce a simple particle simulation. I'm particularly proud of this because I was able to explore more than a few side implementations to try and get the simulation to look as good as possible. For example, I was able to implement a working compute shader to handle the large volume of computation needed to manage 50k particles. There is very little literature out there that explains how to retrieve the compute shader data from the GPU once it has been written to the buffer (1 article, in fact). I was also able to implement the same simulation using instanced arrays. I experimented with laying out the particles in the shapes of different objects, like teapots and bunnies, and noticed that there are a limited number of vertices in these objects. This led me to writing code to tessellate these objects and pass the data back to the host via a transform feedback buffer from the Geometry Shader, etc. My code handles up to 50k particles smoothly on an NVIDIA Quadro K1000M GPU (not super high end) and if I had more time, I would like to have explored combining the instanced arrays with the compute shader on tessellated objects. I think this would make for a nice effect to simulate explosions. Each particle was simulated as sphere objects when instancing was used, but when I went to the compute shader I wasn't able to figure out how to combine compute shaders and instanced arrays so I pivoted and adjusted the glPointSize to make each particle more visible and then applied the sphere-normal texture to each point sprite. It was a thing of beauty. See below for just a couple screenshots:



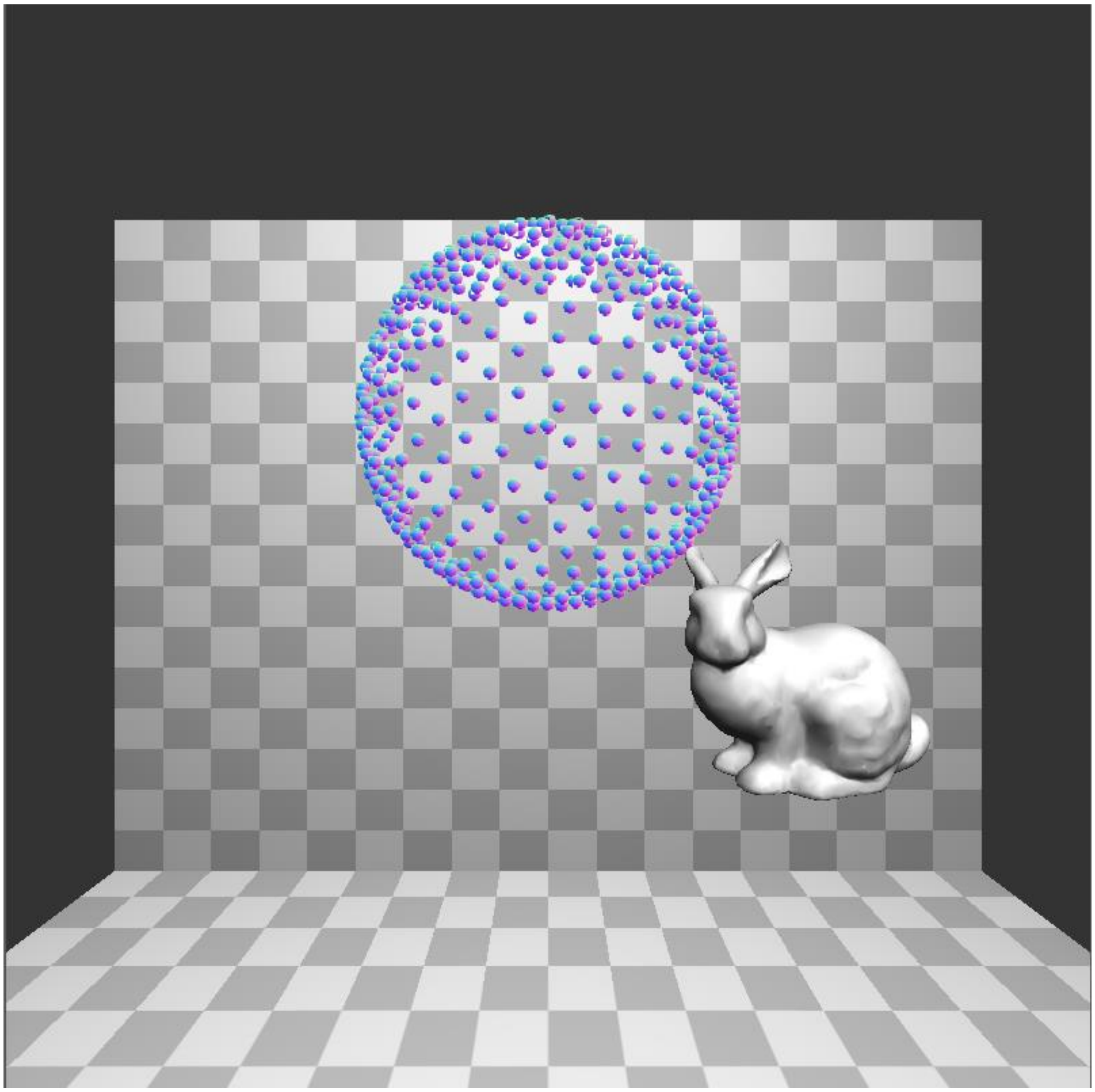
The "Storm Cloud" (50k+ particles about to rain down

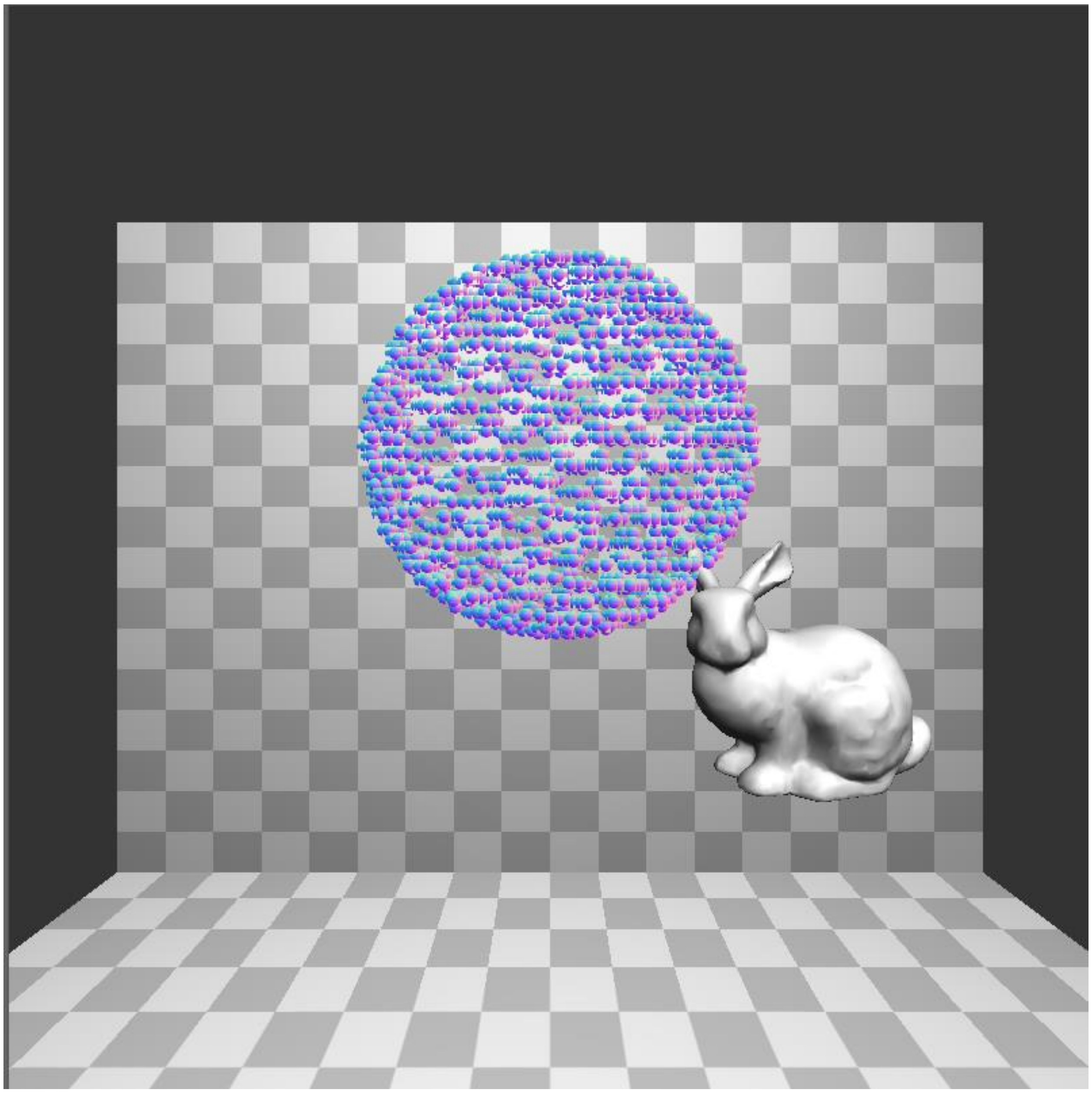


Packaged pixels



A lateral view


The number of particles is easily adjustible



As before, the project relies on the below. shaderCompiler.h is included with the project.

```
#include <GL/glew.h>
#include <GL/freeglut.h>
#include <iostream>
#include <cyCore.h>
#include <cyPoint.h>
#include <cyMatrix.h>
#include <cyTriMesh.h>
#include <cyGL.h>
#include <lodepng.h>
#include <shaderCompiler.h>
```