CS 184: Computer Graphics and Imaging, Spring 2019

Project 4: Cloth Simulator

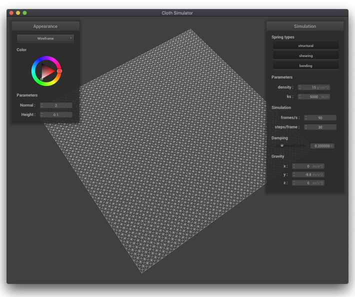
Gefen Kohavi, CS184-aag

Overview

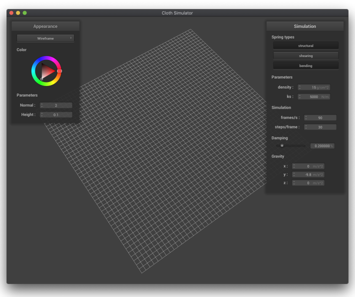
In this project, I implemented a cloth simulator as well as a OpenGL shader. I learned that to make a cloth simulation, we need to represent it as a grid of springs. I learned about some of the challenges with cloths such as position constraints and self-collisions. I also learned about how shaders are implemented using OpenGL. Shaders suprisingly have quite a bit of power, all while being super fast.

Part 1: Masses and springs

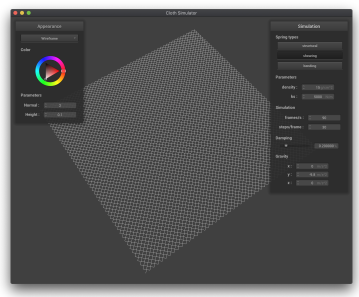
This part entailed simply constructing the wireframe and initializing the springs.



All springs

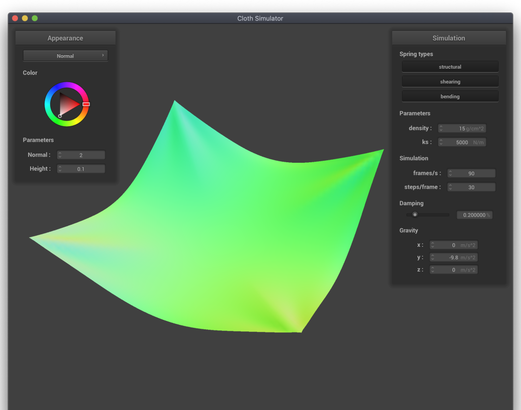


No shear springs

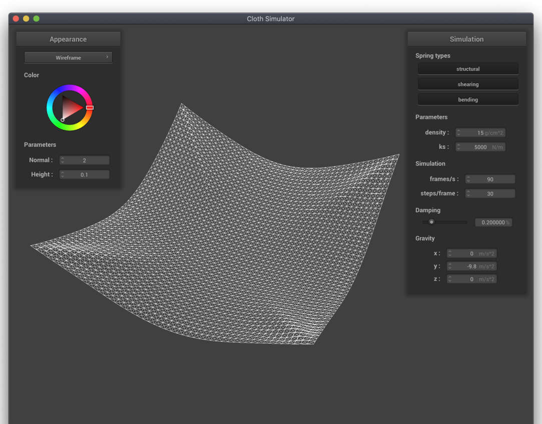


Only shear springs

Part 2: Simulation via numerical integration

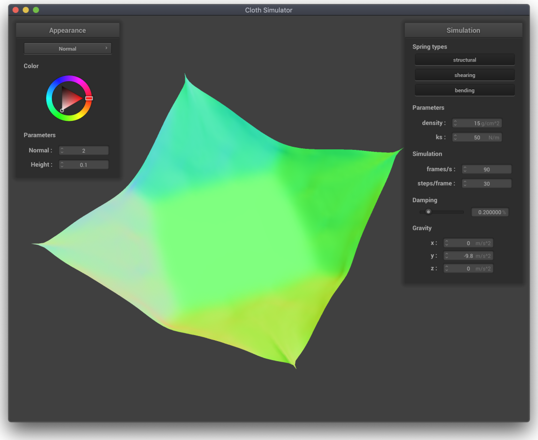


Shaded default pinned 4

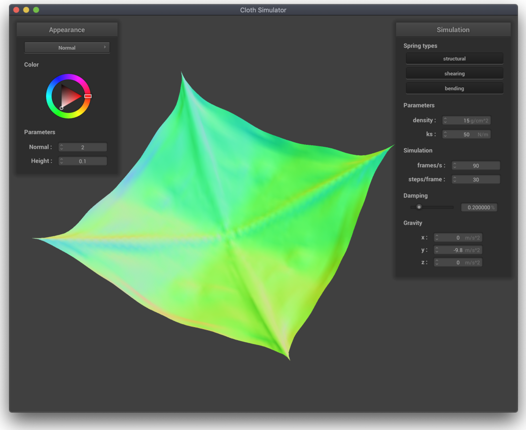


Wireframe default pinned 4

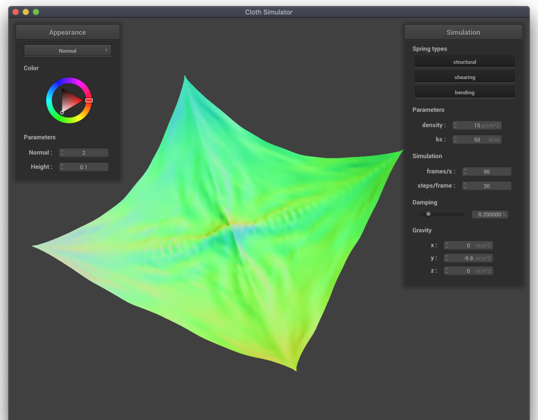
Changing the spring constant K_s in the Hooke's Law equation is equivalent to changing how much restorative force is generated given some displacement from the resting length. A high K_s will have a high restorative force and cause the mesh to act more rigid and "springy". A low K_s will have low restorative forces and cause the mesh to act less springy and more like water.



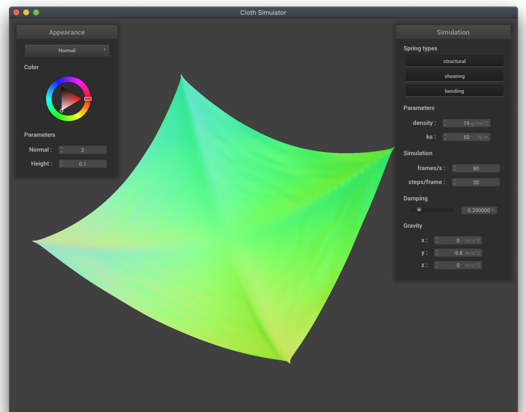
Ks = 50 img1



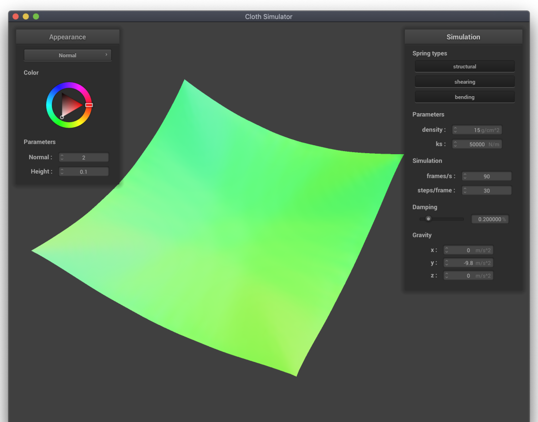
Ks = 50 img2



Ks = 50 img3

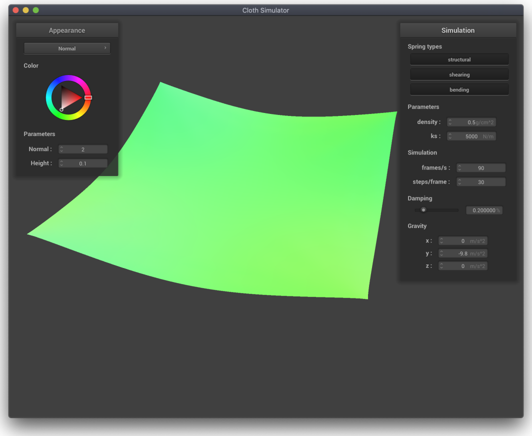


Ks = 50 img4

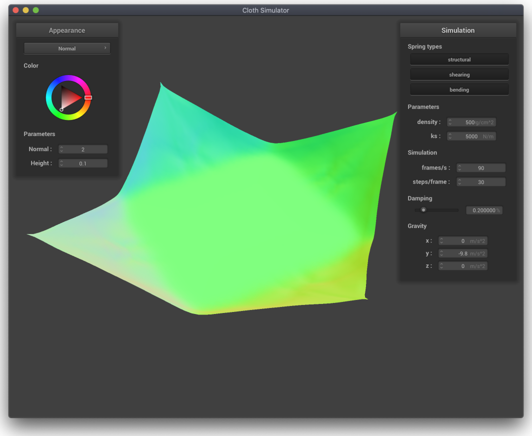


Ks = 50,000 resting

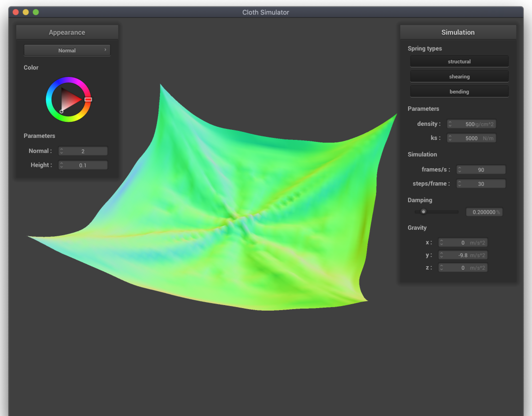
Density affects how much each point mass weighs. A higher density is quite effective at counteracting the spring forces since it takes a longer time to speed up a heavier mass with the same force. A lower density will quickly be affected by the spring forces and reach the resting state quickly.



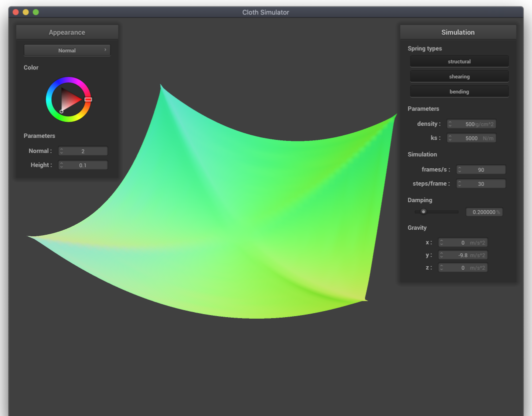
Density = .5



Density = 5000 img1

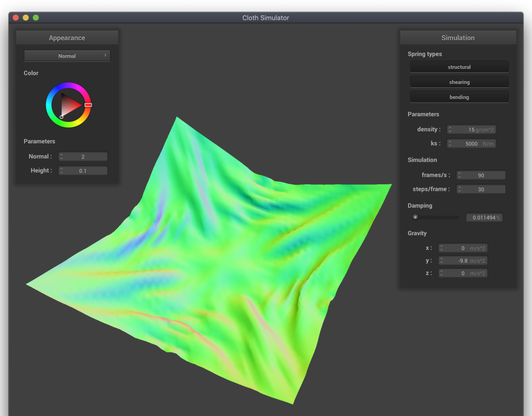


Density = 5000 img2

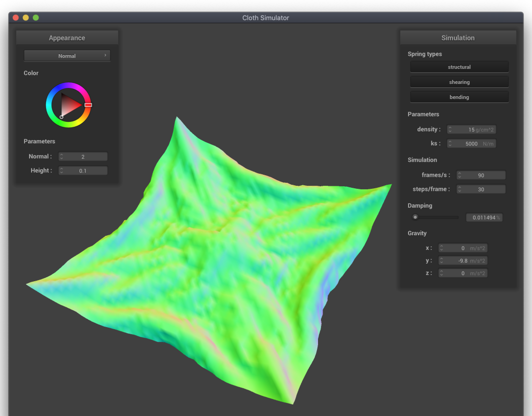


Density = 5000 img3

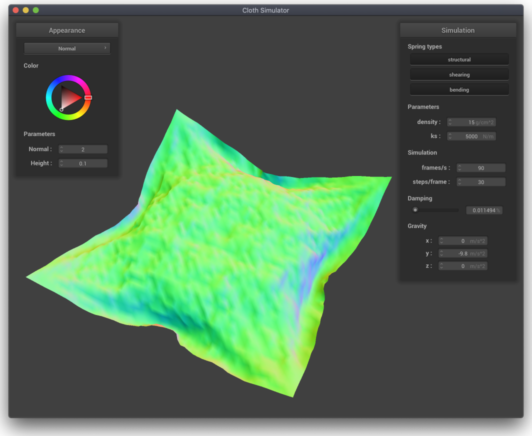
Damping is used in the Verlet integration step and it is used to remove energy from the springs. Ideally the springs should move less and less over time after falling. By using damping to scale each new poistion change, the springs will occillate less and less over time. With high damping, the occillations of the springs quickly go away and we reach the resting state of the spring quickly. With low damping, the springs expand and contract, keeping the energy of the previous movements.



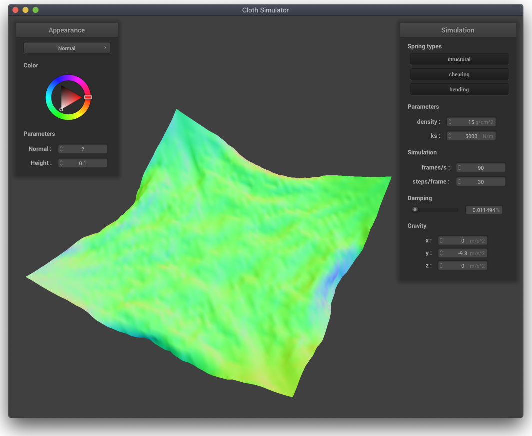
Damping ~0, img1



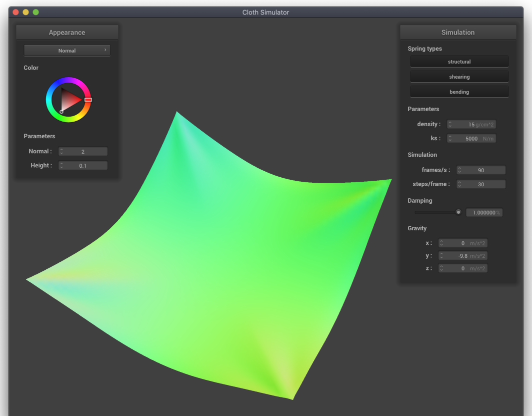
Damping ~0, img2



Damping ~0, img3

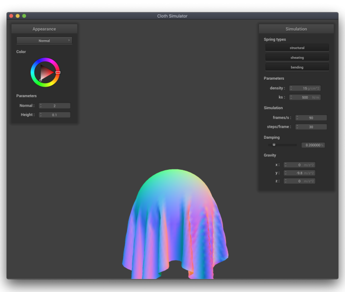


Damping ~0, img4

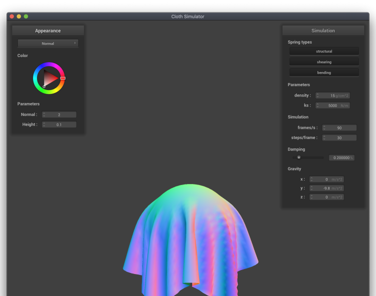


Damping 1, img1

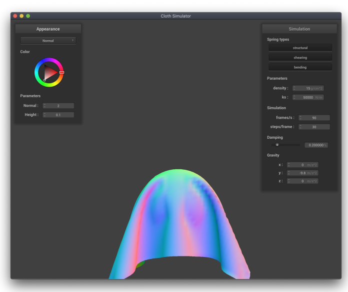
Part 3: Handling collisions with other objects



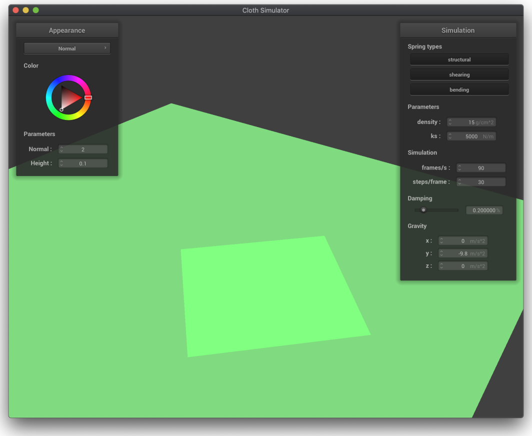
Ks = 500



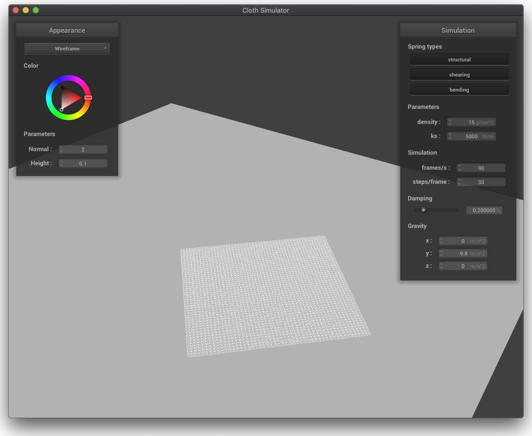
Ks = 5k (default)



Ks = 50k

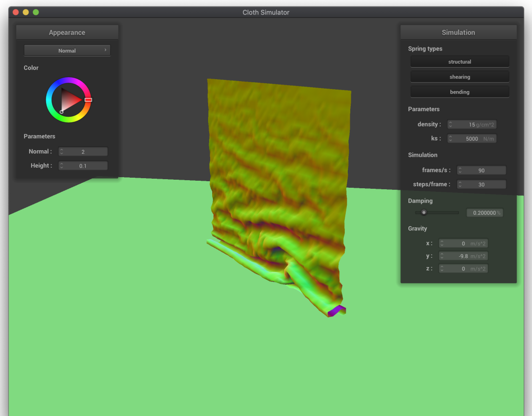


Cloth on plane

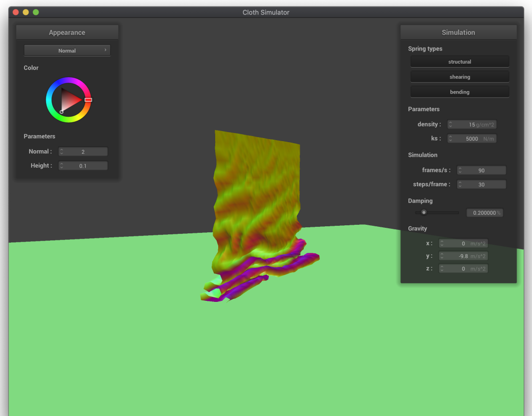


Cloth on plane

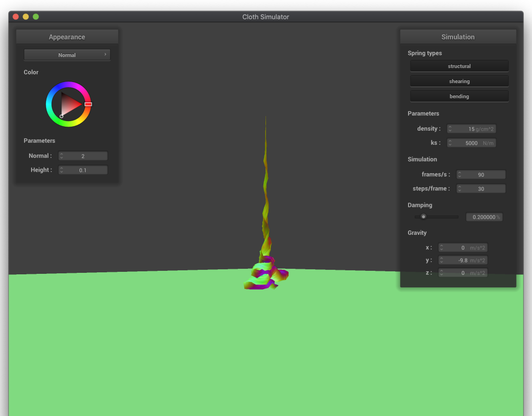
Part 4: Handling self-collisions



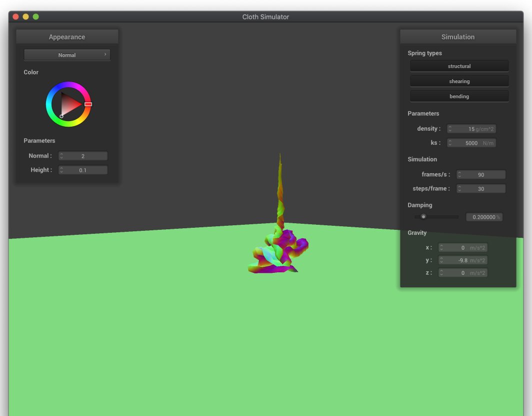
t = 1



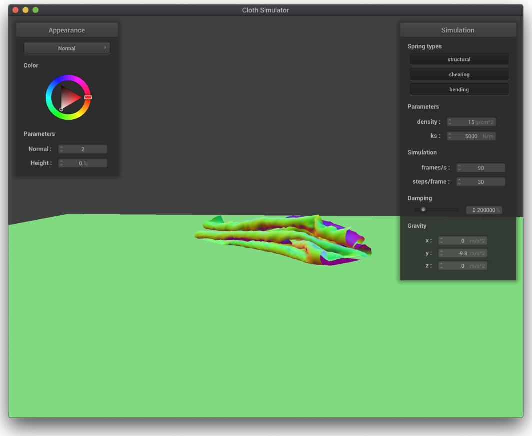
t = 2



t = 3

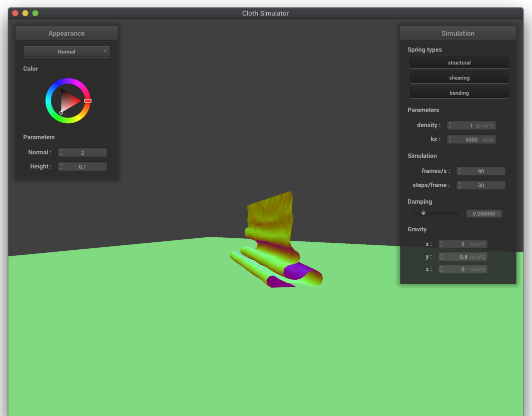


t = 4

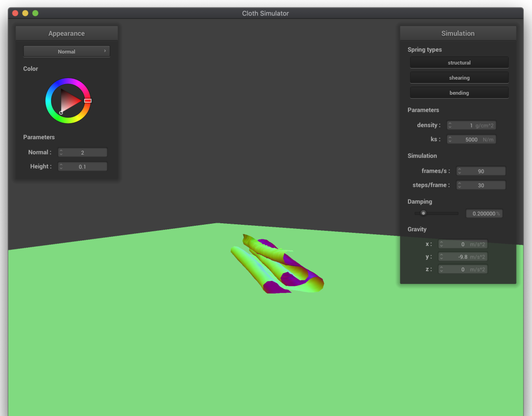


t = 5

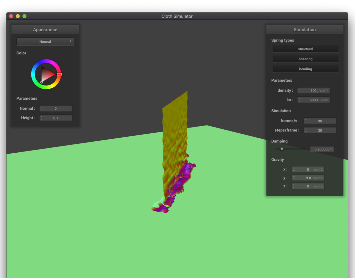
Low density alllows the self-collision code to affect the trajectory of pointmasses significantely and results in a nice, smooth decent of the cloth onto the plane. On the other hand, with great density, we see that self-collision just prevents self collion and doesn't change pointmass trajectory too much. As a result, the dense cloth clumps up since once any self collision is fixed, the same point will move in almost the same direction and self-collide in the next time step.



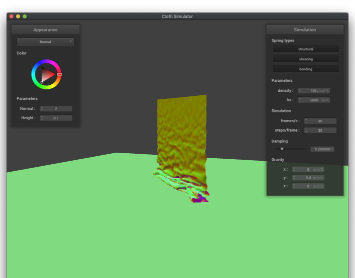
Low density 1, d = 1



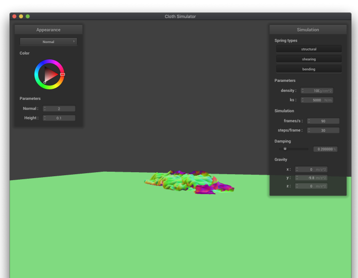
Low density 2, d = 1



High density 1, d = 100

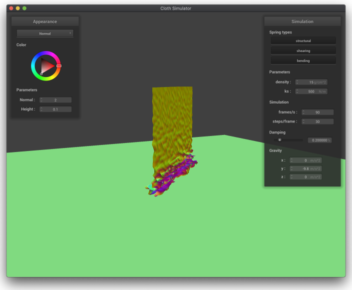


High density 2, d = 100

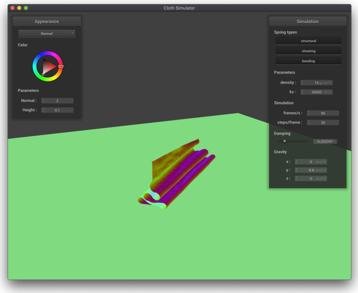


High density 3, d = 100

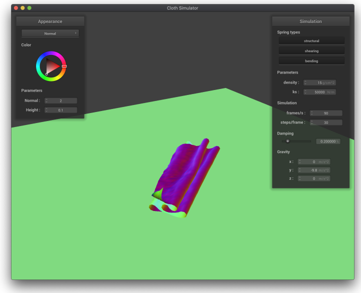
K_s acts very similarly to density. With a high K_s, we see that the cloth falls and has a smooth shape throughout the fall. This makes sense since a high restorative force, prevents any one self collision from affecting the overall cloth mesh too much. With a low K_s, we see that the result is extreemely noisy and this makes sense as the resorative force does not smooth the self-collision of points across the mesh.



Low Ks, Ks = 500



High Ks, Ks = 50k

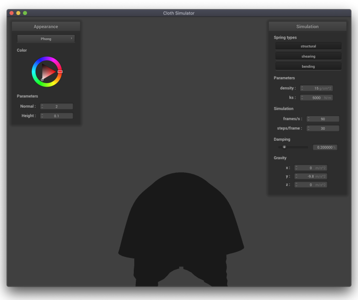


High Ks, Ks = 50k

Part 5: Shaders

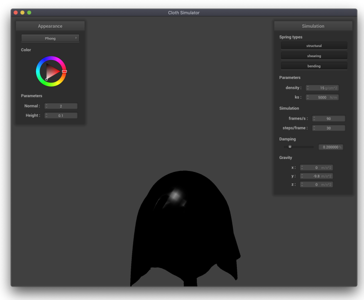
Shaders are a way to render a scene without using ray-tracing. In fact, shaders are similar to project 1 where most of the work being done is rasterization. As a result, this method can be run extreemely fast. We can emulate raytracing effects with shaders. Given a mesh, we know information about an object's normals and position. We can then use attributes like textures or object properties to create cool looking results.

The Blinn-Phong Shading Model consists of three parts. First, we have ambient lighting. This is off the assumption that there is some baseline level of lighting in the enviroment that lights the entire object equally. Next, the object is diffusely shaded to represent a lambertian object. This gives a sense of directional lighting and shadows. Finally, specular shading is added to show light highlights and shinyness. Each of these three terms are combined with a weighted average. Specular shading in particular, can act more diffuse or more like a mirror by chaning the exponent value when taking the dot product between the normal vector and the half vector.

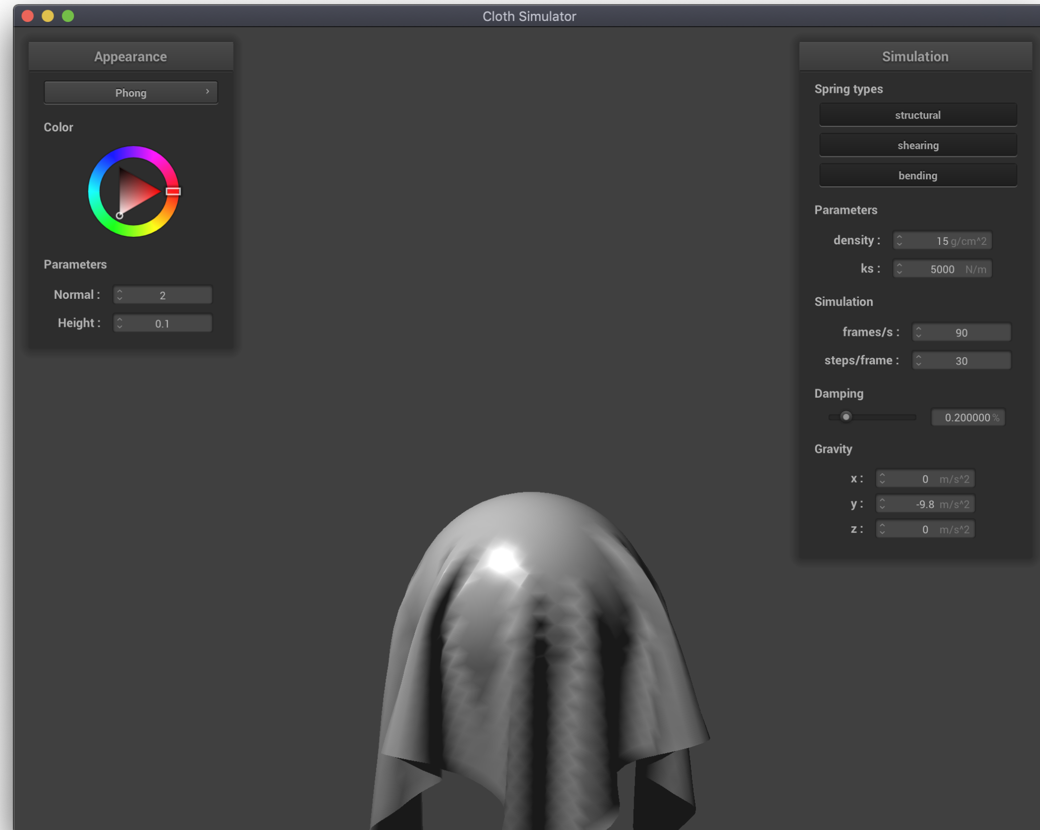


Ambient

Diffuse

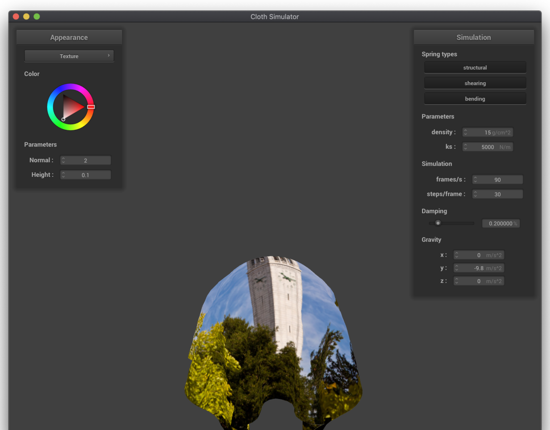


Specular

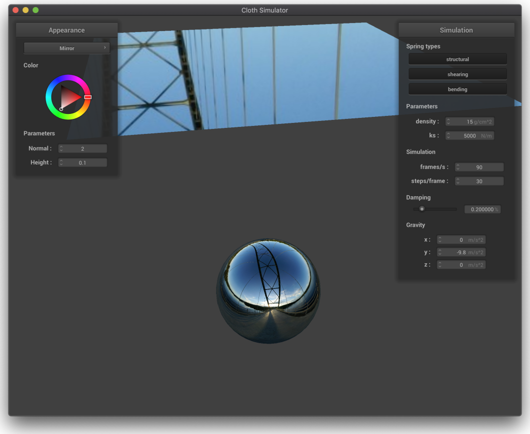


Blinn Phong

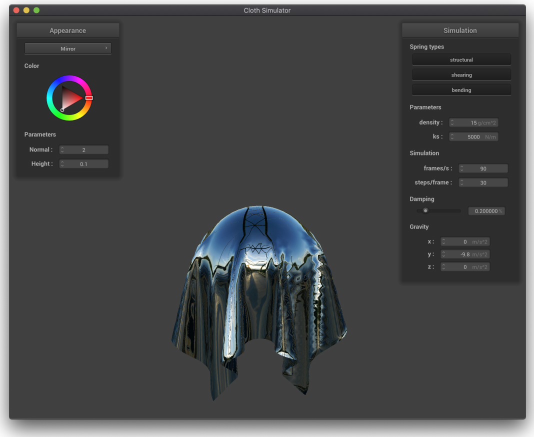
My custom texture, paper wad



Campanile texture

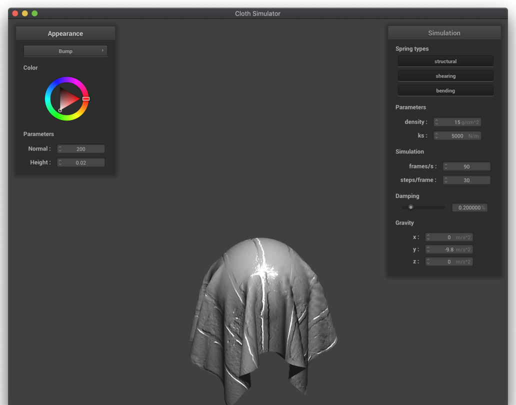


Mirror cloth

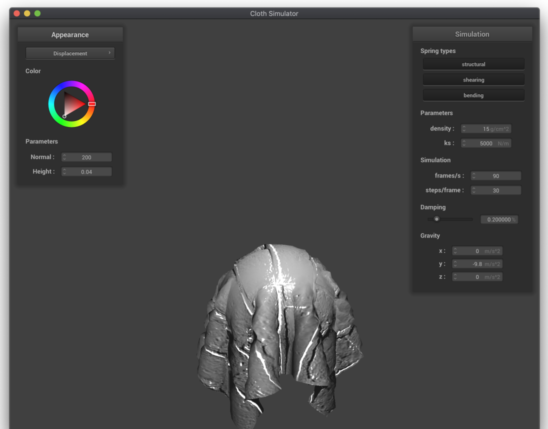


Mirror sphere

For bump and displacement mapping, I used texture 3.

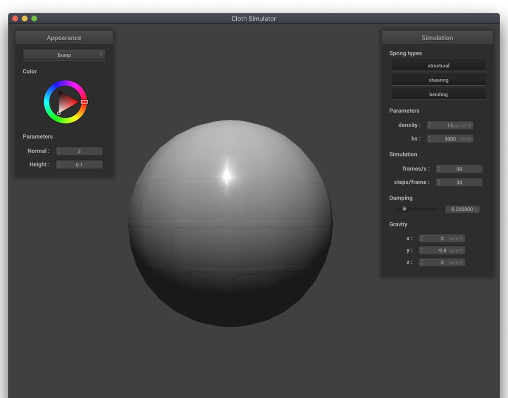


Bump mapping

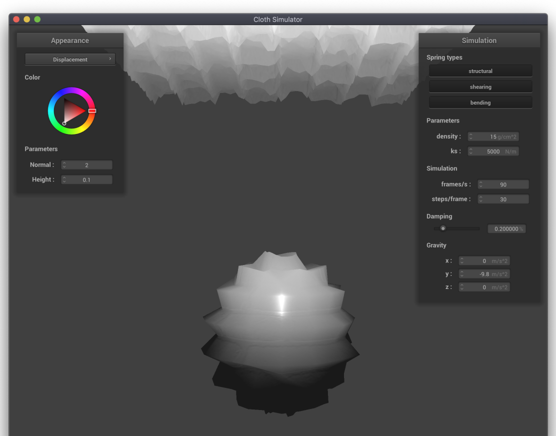


Displacement mapping

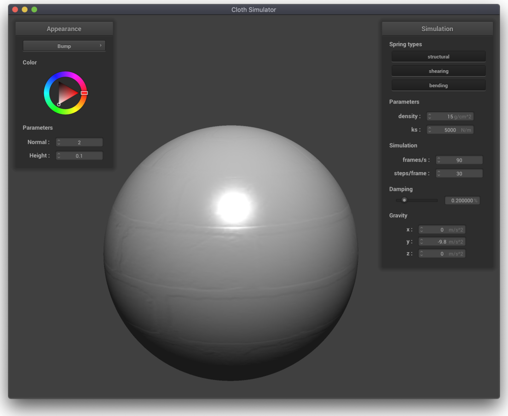
Bump mapping gives a good view of the underlying texture, but it still looks flat. Displacement mapping on the other hand looks like a significant improvment in making the mesh look bumpy. Especially around the cornners, the cloth looks thicker, almost as if the cloth is acutally made of the texture I used.



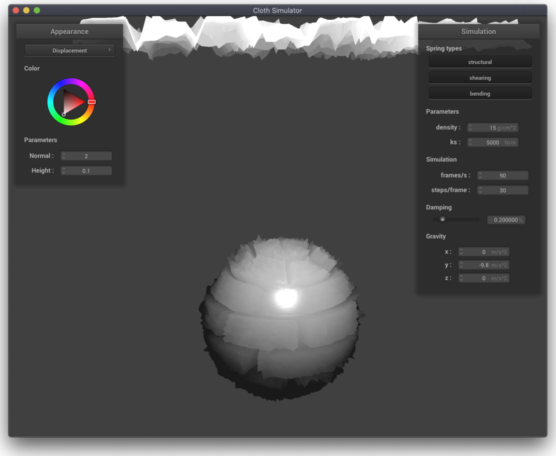
Bump -o 16 -a 16



Displacement -o 16 -a 16



Bump -o 128 -a 128



Displacement -o 128 -a 128

The difference between the two is the level of bumpieness in the meshes. -o 16 -a 16 has very coarse bumps which represent a rough approximation of the texture. With -o 128 -a 128, the mesh approximates the texture better but it is noisier. Therefore, to have a good mesh scene, it is good to have a balance between fine-tunned approximation and little noise (coarseness).