CS 184: Computer Graphics and Imaging, Summer 2020

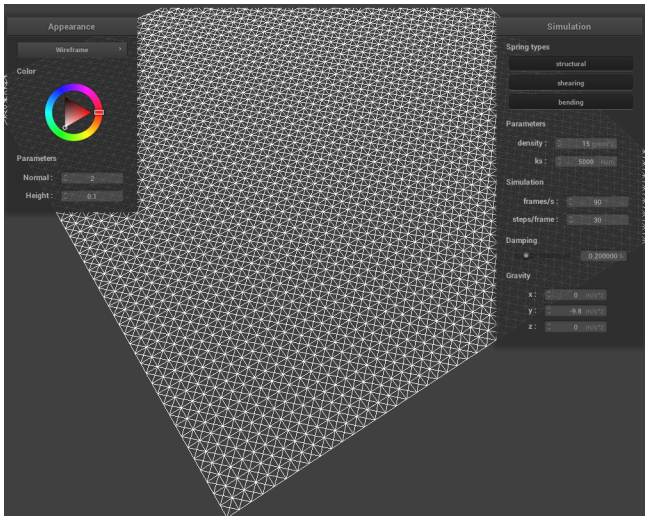
Project 4: Cloth Simulator

Billy Chau, CS184-amztc34283

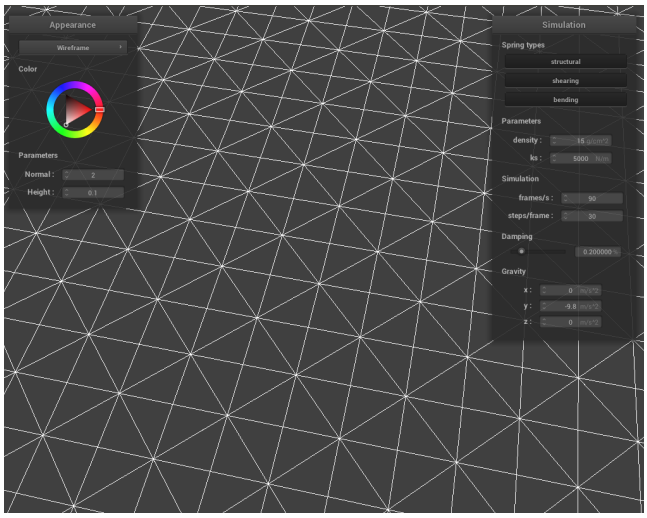
Overview

Give a high-level overview of what you implemented in this project. Think about what you've built as a whole. Share your thoughts on what interesting things you've learned from completing the project.

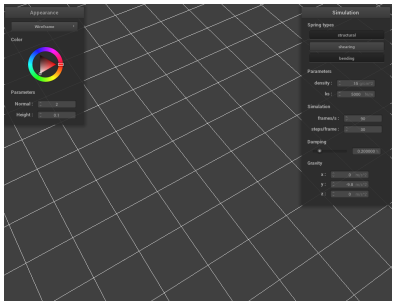
Part I: Masses and springs



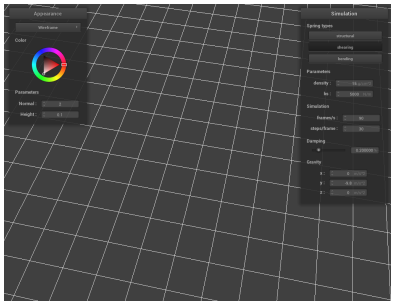
scene/pinned2.json



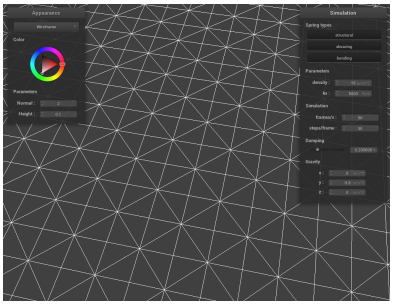
scene/pinned2.json



scene/pinned2.json without any shearing constraint



scene/pinned2.json with only shearing constraint



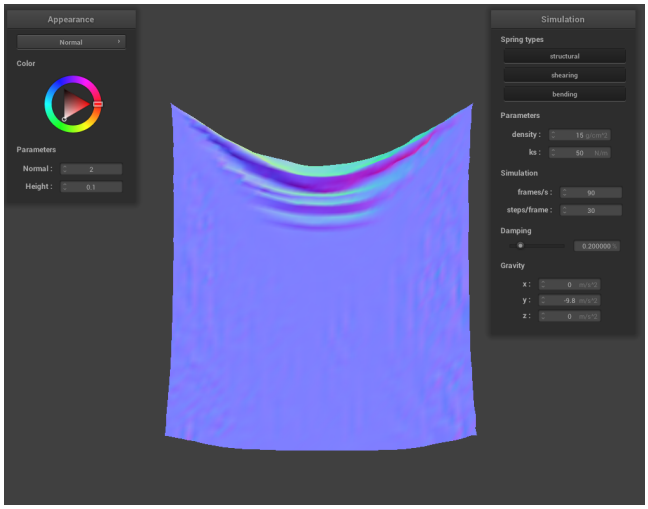
scene/pinned2.json with all constraints

As expected, the addition of the first two images forms the last image.

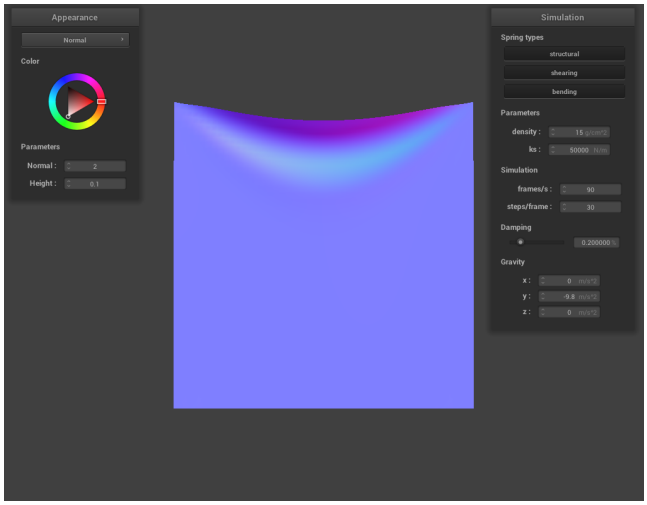
Part II: Simulation via numerical integration

By increasing the spring constant ks, the cloth appears to be less springy or more tight-fit whereas the cloth appears to be more springy or more loose-fit when the spring constant decreases. In other words, the cloth with lower ks takes a longer time to stabilize and vice versa. By increasing the density, the cloth appears to be heavier and takes a longer time to stabilize itself and vice versa. One of the reasons is that the heavier cloth has more gravitational force exerts on it, thus requiring a longer time to burn out the energy by friction, heat, etc. By increasing the damping constant, the cloth appears to fall down slower because there are more opposing forces

decelerating the cloth. On the other hand, decreasing the damping constant can decrease the time required to fall down, and an extreme case will be using damping = 0, where the cloth will swing around for a very long time.

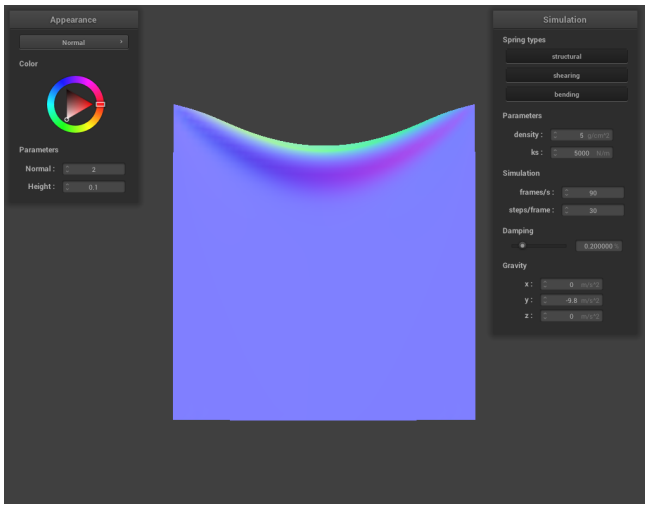


scene/pinned2.json with ks = 50

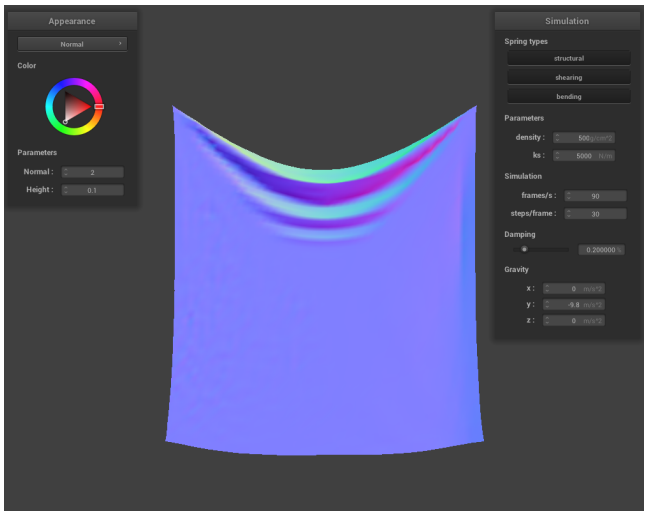


scene/pinned2.json with ks = 50000

As expected, the cloth with smaller ks has a lot of vibrations because of its springy nature while the cloth with bigger ks is stabilized.

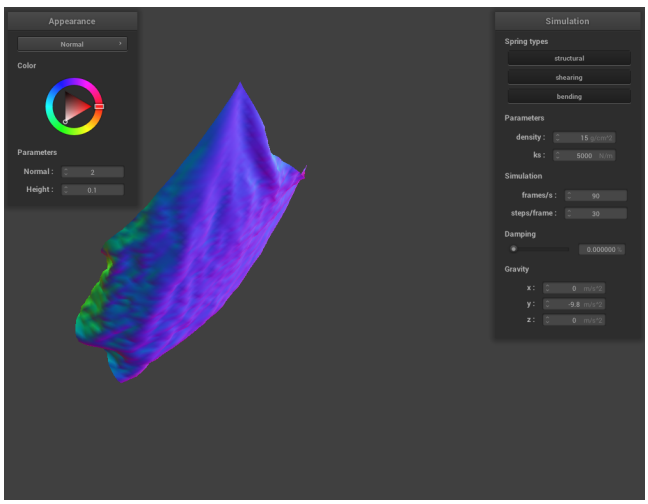


scene/pinned2.json with density = 5

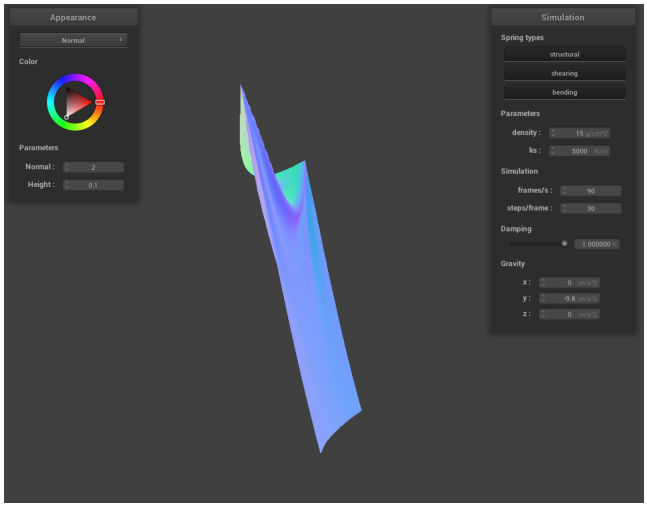


scene/pinned2.json with density = 500

The cloth with lower density does not have as much pulling force as the other. We can see that the pulling force is higher with higher density cloth because the cloth is more flat or stretched-out in general except at the top part. In addition, we can see the vibrations in the cloth with higher density. Interestingly, we can see that both density and spring constant have big impact on stability because they both have impacts on the forces acting on the point masses.

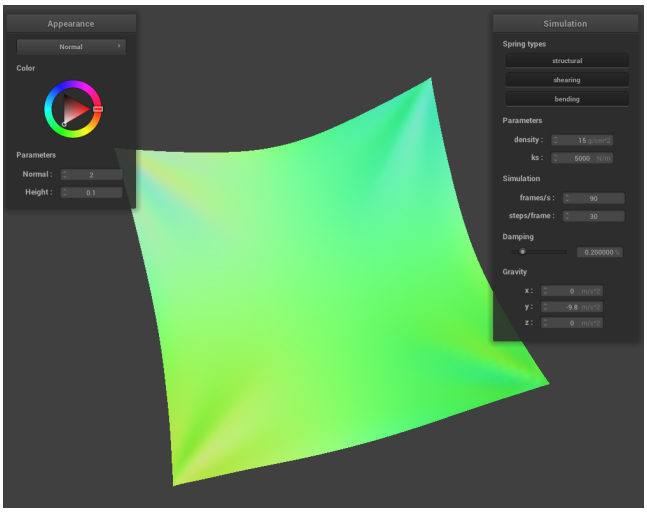


scene/pinned2.json with damp = 0%



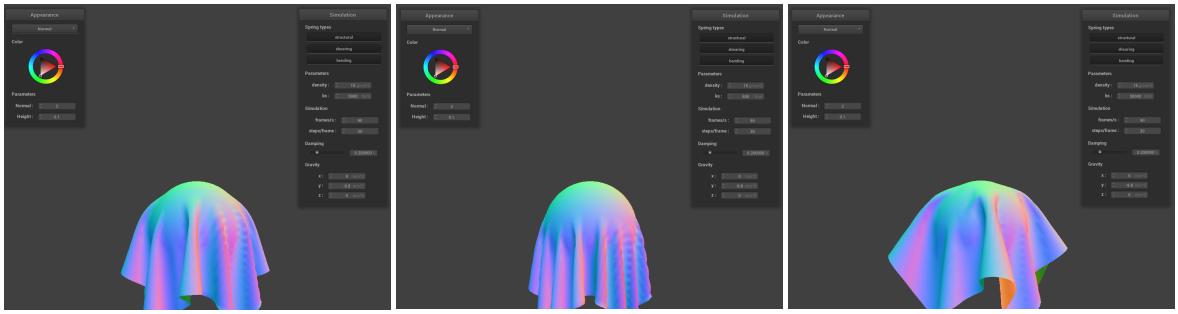
scene/pinned2.json with damp = 1%

The cloth with lower damping constant swings wildly because there are not much friction burning off the energy. On the other hand, the cloth with higher damping constant flows down evenly from top to bottom because a lof of energy is dissipated via friction, heat, etc; therefore, the acceleration is significantly decreased.



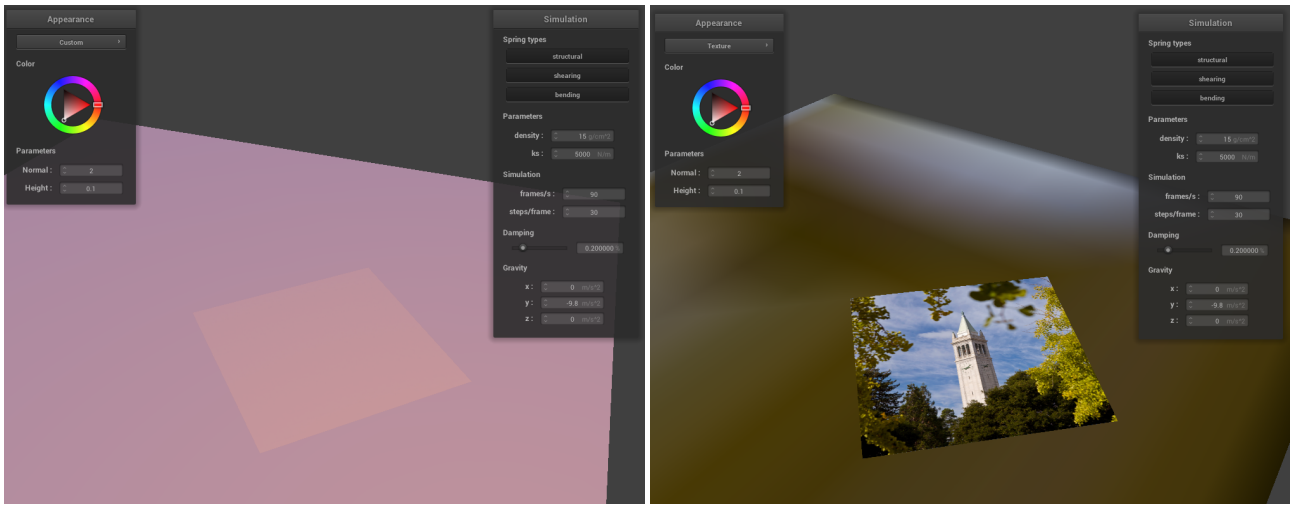
Final state

Part III: Handling collisions with other objects



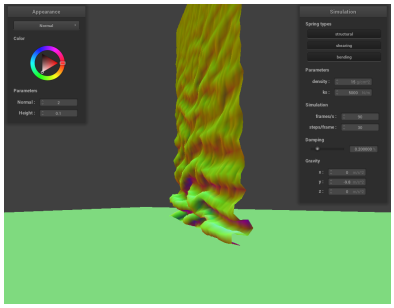
scene/sphere.json with ks = 5000 scene/sphere.json with ks = 500 scene/sphere.json with ks = 50000

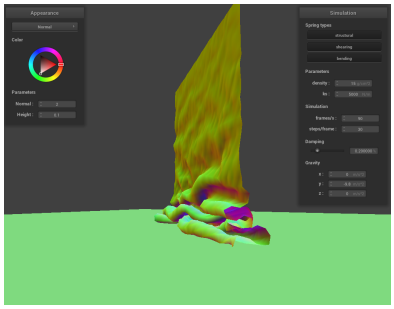
With increased spring constant, the cloth appears to be very resistive to bending because the cloth in the last image does not bend as much as the cloth in the middle (which is bended like a dumpling). The flexibility of the cloth in the first image is in between them.

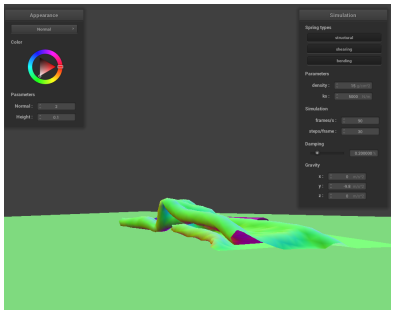


scene/plane.json with cold-warm shading scene/plane.json with campanile

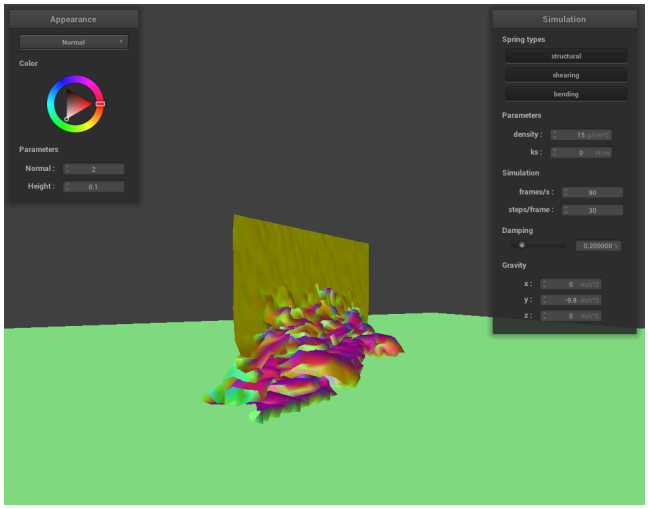
Part IV: Handling self-collisions

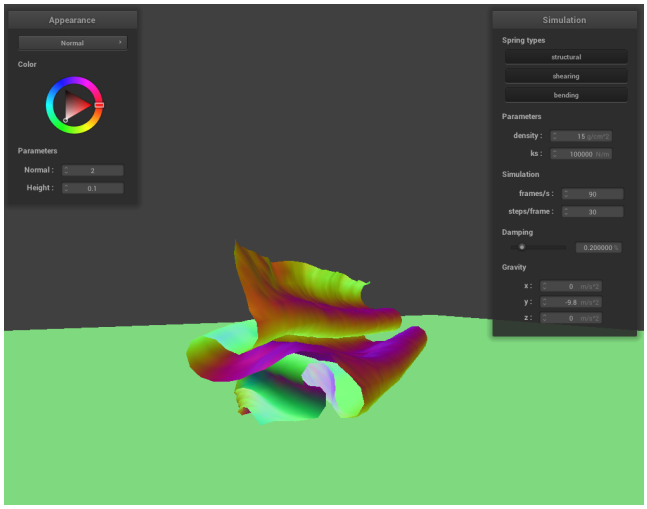

scene/selfCollision.json first stage


scene/selfCollision.json second stage

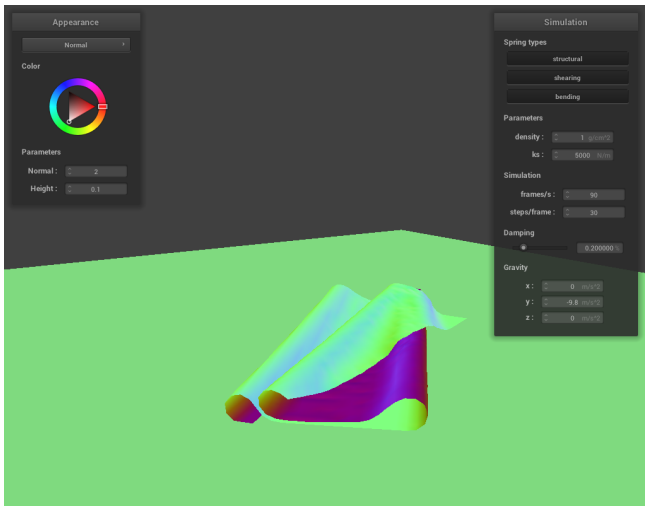

scene/selfCollision.json third stage

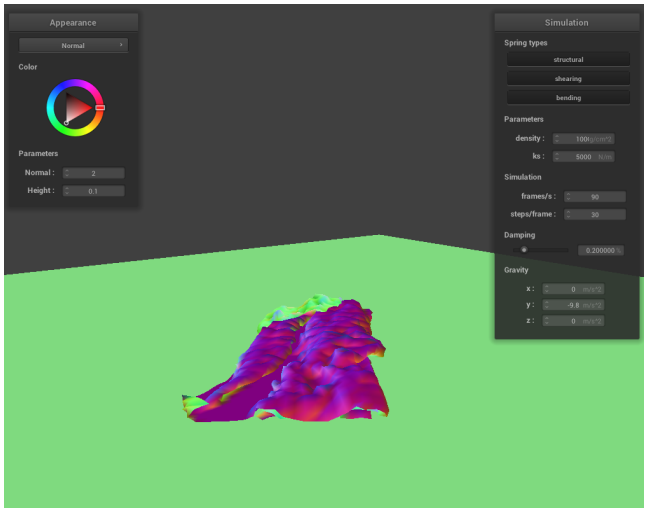
As expected, the point masses on the cloth adjust their positions when the cloth collides.


scene/selfCollision.json with spring constant = 0


scene/selfCollision.json with spring constant = 100000

When the spring constant is zero, the cloth appears to be like water in which does not have as much structures as a normal cloth whereas it appears to be like a thick paper falling when the spring constant is high.


scene/selfCollision.json with density = 1


scene/selfCollision.json with density = 1000

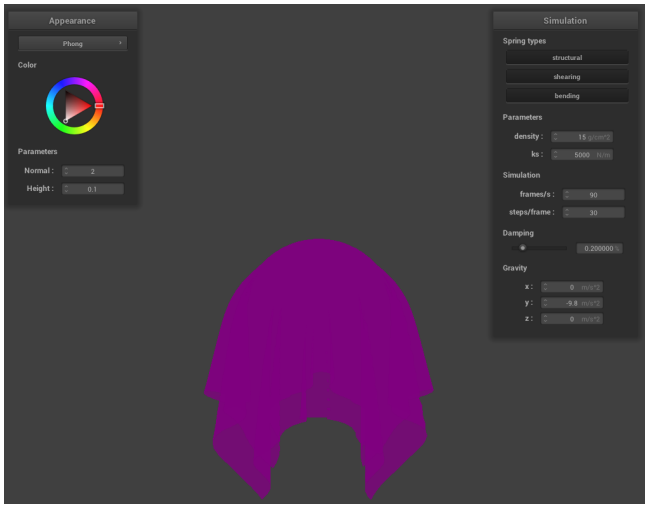
When the density is low, the cloth appears to be like a thick paper falling whereas it appears to be like water in which does not have as much structures as a normal cloth when the spring constant is high. This trade-off is exactly the same discussed in part 2 in which the force is the result of both spring constant and density.

Part V: Shaders

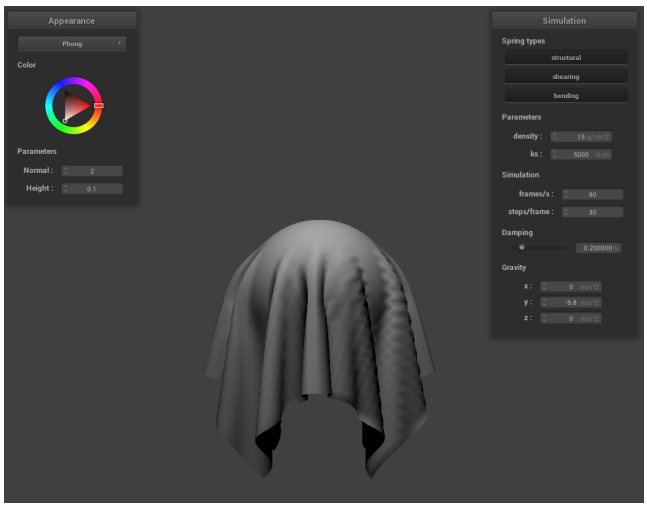
Shader program is a program that could run massively parallel directly on the GPU. The built-in functions of the shader programs are usually limited by the micro-architecture of the GPU. Normally, it is used for two purposes: processing vertex and fragment. Vertex shader takes in the information of the vertex such as position, normal vector, tangent vector, and the uv coordinate as inputs and outputs the same information in model space and screen space; therefore, the info passed in is based on the object space. Fragment

shader on the other hand takes in the output of the vertex shader and outputs a final color for that vertex/fragment; therefore, most of the work is done at the fragment shader.

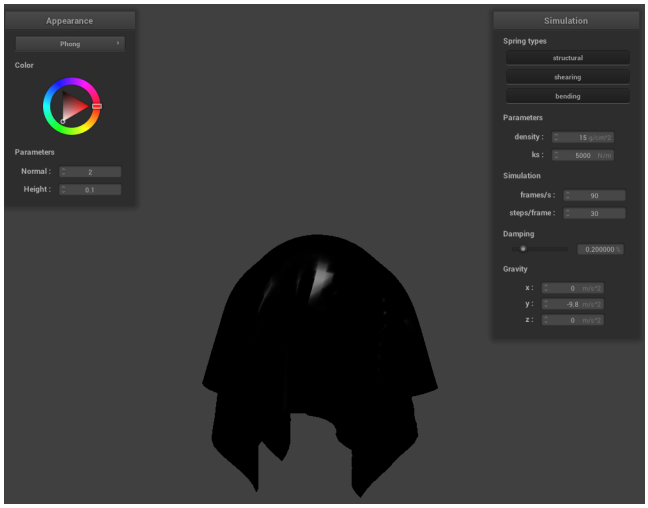
Blinn-Phong shading combines three different perspectives of lighting together. It combines ambient effect, diffuse effect, and specular effect together. First, ambient effect does not depend on anything, so we can think of it as the base color when there is light or not. Second, diffuse effect depends on the light and normal directions, and it decays according to 1/r^2. Third, specular effect depends on the light, normal, and viewing directions.



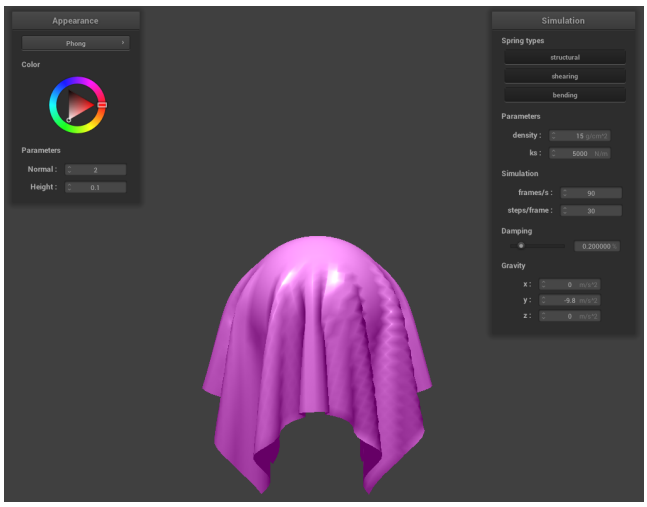
scene/sphere.json with ambient



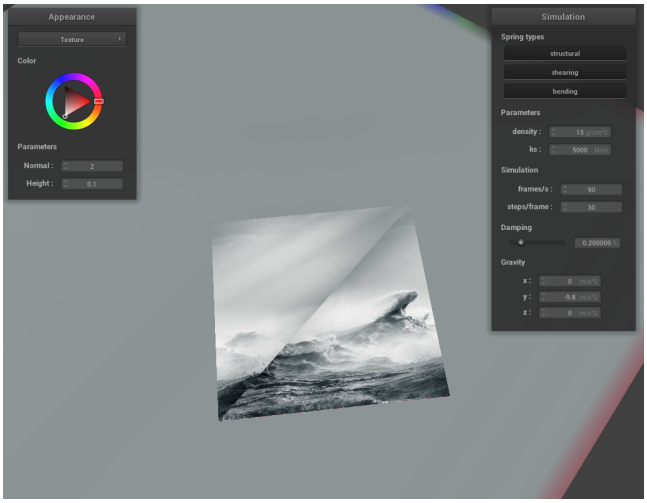
scene/sphere.json with diffuse



scene/sphere.json with specular



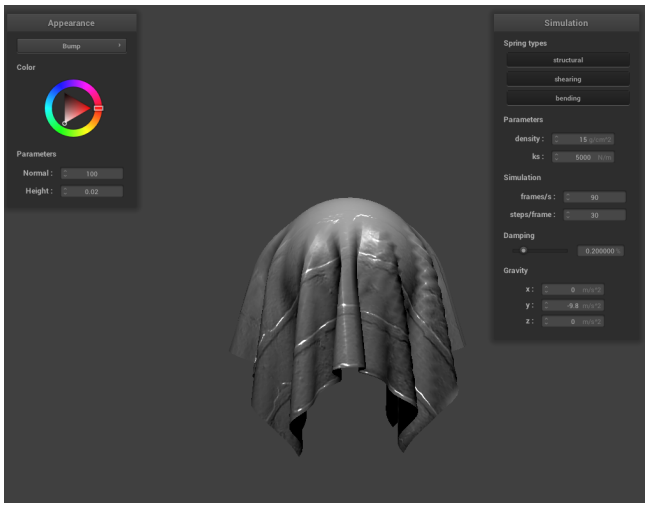
scene/sphere.json with all



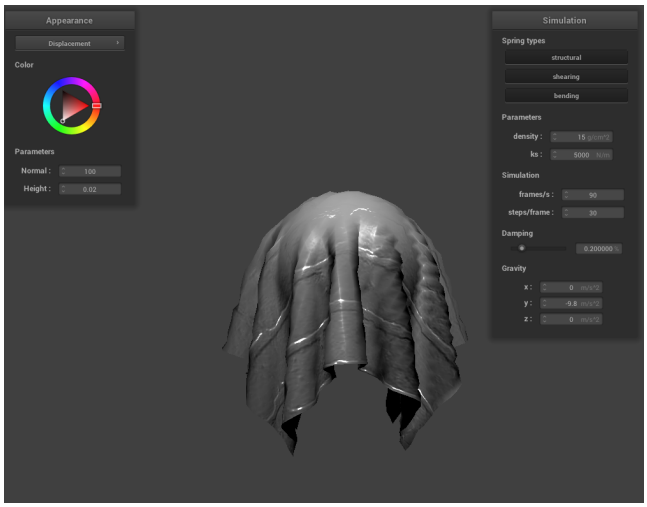
Custom texture with poseidon rising with the bad perspective transform

Bump Mapping artificially creates texture that appears to be more 3d by looking at the height map of the texture. The height is then used to calculate the normal vector in object space. The normal vector is then transformed to the model space by multiplying the tangent-bitangent-normal matrix which defines the coordinates of the object space in the model space. On the other hand,

displacement mapping additionally modifies the position of the vertices according to the normal vector change calculated from bump mapping.

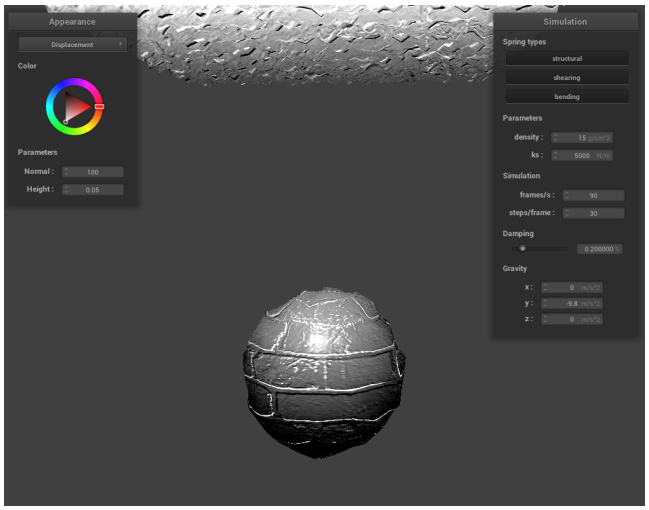


scene/sphere.json with bump

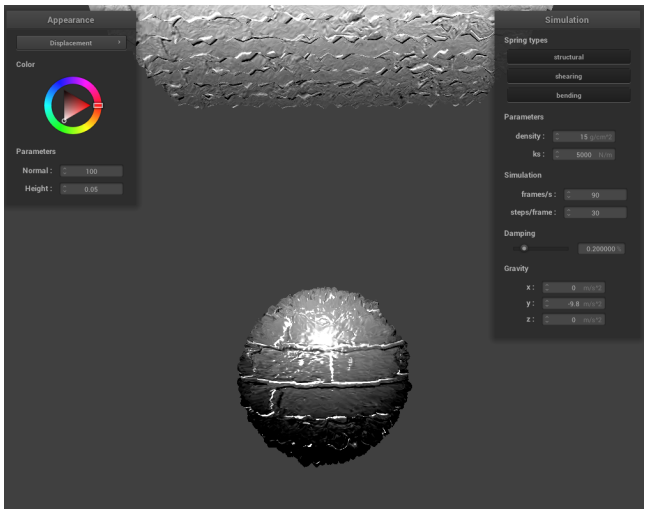


scene/sphere.json with displacement

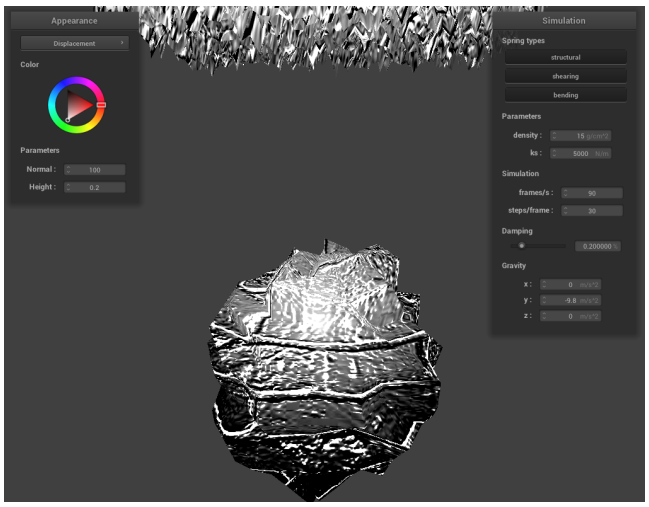
By increasing the coarseness of the sphere, there are more spiky shapes. Although it is not obvious with low bump height as depicted below, it becomes significant when the bump height is high.



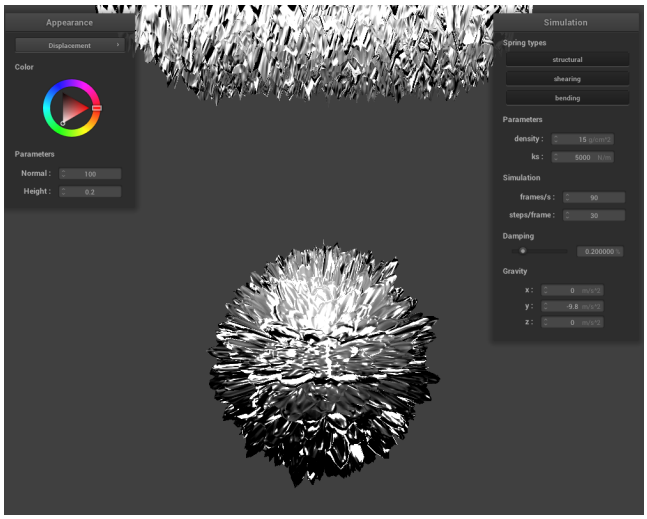
scene/sphere.json with displacement and low bump height and low coarseness



scene/sphere.json with displacement and low bump height and high coarseness

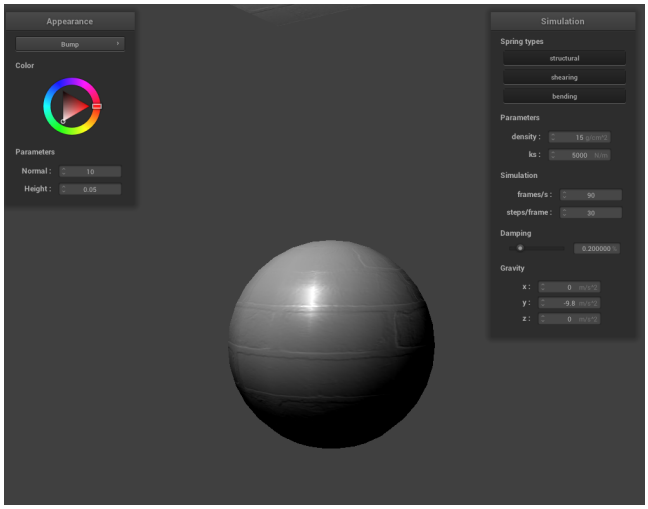


scene/sphere.json with displacement and high bump height and low coarseness

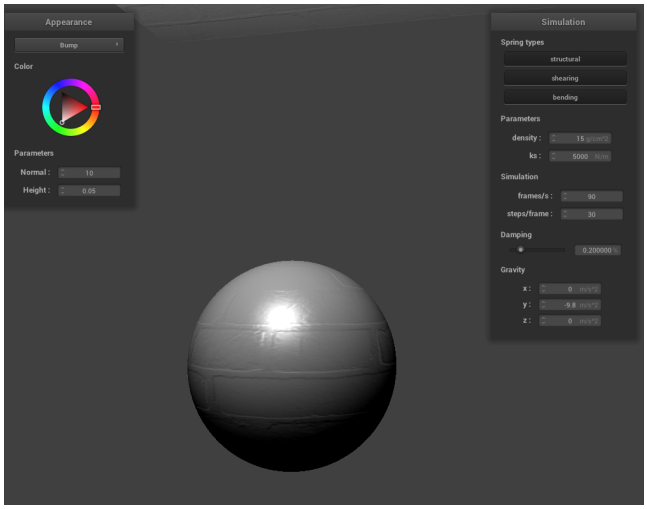


scene/sphere.json with displacement and high bump height and high coarseness

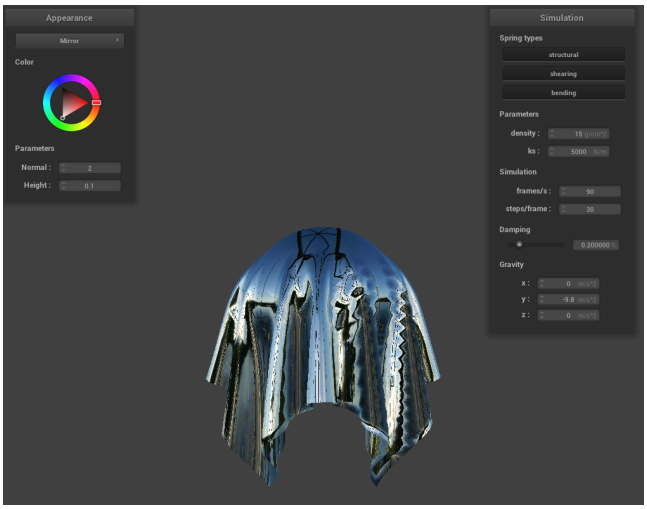
By increasing the coarseness of the sphere, the specular reflection becomes more spherical in comparison to a non-spherical shaped specular reflection in sphere with lower coarseness.



scene/sphere.json with bump and low coarseness



scene/sphere.json with bump and high coarseness



scene/sphere.json with reflection

In the extra credit portion, I implemented the cold-to-warm shading by using the following equation:

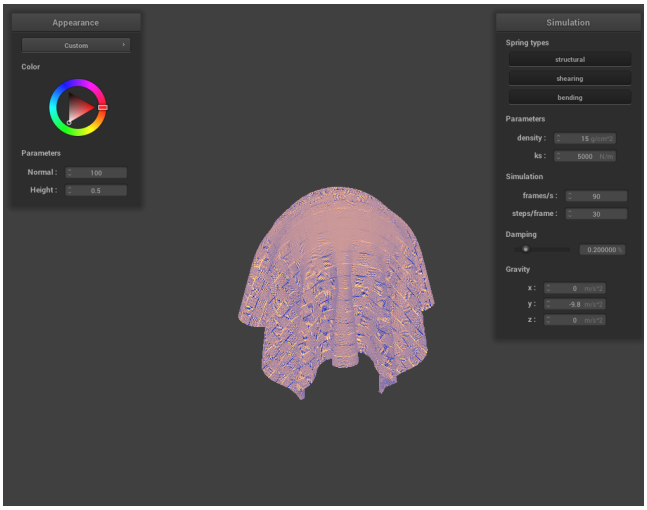
$$c = k_w c_w + (1 - k_w) c_c$$

$$k_w = \frac{1 + n \cdot l}{2}$$

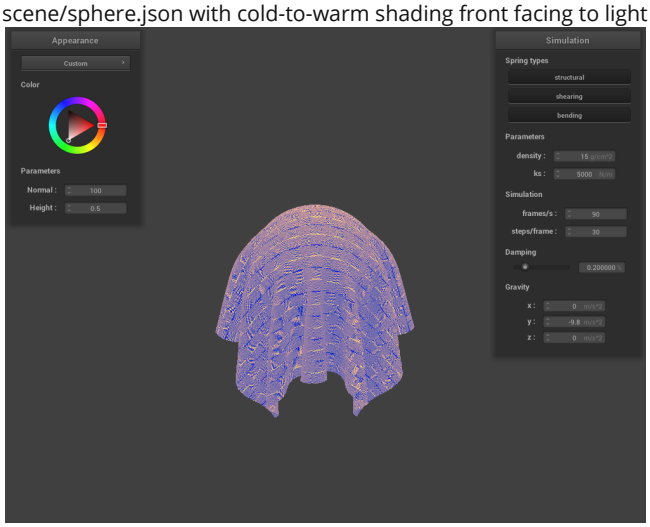
Value chosen:

$$c_c = (0.4, 0.4, 0.7)$$

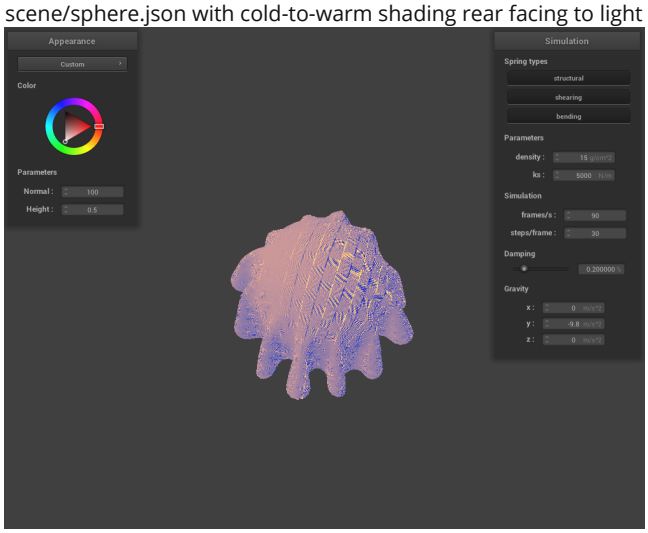
$$c_w = (0.8, 0.6, 0.6)$$



scene/sphere.json with cold-to-warm shading front facing to light



scene/sphere.json with cold-to-warm shading rear facing to light



scene/sphere.json with cold-to-warm shading top-down view

As expected, the idea of cold-to-warm shading is to give more warm color to position where its normal is facing more to the light while giving cold color to position where its normal is not facing to the light as much.