Water Simulation from Physical Models

For this project I did a physical simulation of water in OpenGL 3.3 with C++. To calculate how the water appears you need to calculate two things, the geometric undulation of the wave and the wave in the ripples of the normal map. To calculate the geometric undulation of the wave I summed a series of Gerstner waves. I found four to be a good number. The BiNormal, Tangent and, Normal can be calculated by finding the partial derivatives. For the texture wave something similar is done, however I instead used a sum of 15 sin waves, also we only care about the height map, so we need only calculate the normal and the u value of the texture is all we care about.

I created the mesh itself in Blender by just using the grid primitive and adding more vertices. Also in Blender I added the uv coordinates by unwrapping from perspective and then created a material with the normal map as a texture and applied it to the mesh. I then exported the mesh as .obj file.

For lighting I just used Blinn-Phong and an environment map.

How to use the program:

When you startup the program you will have to look down with the mouse to see the wave itself. You can use WASD to move the camera around and the scroll wheel to zoom in and out. I've created some keybindings so that you can change the wave up. These are displayed in the console window when you are running the program. They allow you to do things such as change the median amplitude, wavelength, light direction, wave direction, etc. Another neat thing to note is that if you press spacebar the wave will change to a wireframe. This is really good for seeing the geometric undulation of the wave and I think it`s pretty cool to look at. Sometimes the values will change really quickly when you press a button or pressing the spacebar will change the wave to a wireframe and back to a fill without staying on wireframe. Eventually it will work though, I think this is because I use a mechanical keyboard so the keys are actuated for a bit longer.

Dependencies:

My project is dependent on GLEW, SOIL, assimp, GLFW, and GLM. The libraries can be found in the Libs folder in the main folder and the includes in the Include folder in the main folder. So if it doesn't compile you can just change to include and library directories to wherever that is on your machine.

Sources:

Most of my work is based off of this article:
https://developer.nvidia.com/gpugems/GPUGems/gpugems_ch01.html

Most of the boilerplate code and stuff such as the mesh, model, camera, and shader class are based off of the tutorials at http://learnopengl.com/. I did these tutorials quite a while ago so I can't remember what exactly is from tutorial itself and what I have changed.

The normal map is from here:
http://www.digitalrune.com/Portals/0/Blog/Files/11/210/Windows-Live-Writer-Water-Rendering---Implementation_E983-clip_image004_2.png

The skybox is from: http://www.custommapmakers.org/skyboxes.php
The envmap_miramar folder has a readme file which has the author information.

If I've left anything out please contact me.