Water Simulation using Smoothed Particle Hydrodynamics

Group Members:

* Zack Mayeda (cs184-bg)
* Tyler Brabham (cs184-ej)

Project Description:

We simulate water using the method of Smoothed Particle Hydrodynamics. This works by taking physical quantities, such as density, at some finite number of particles and using kernels to smooth out the values to the surrounding space. Thus, with these kernels we can get the value of any physical quantity at any point in space. Then, we can numerically integrate the Navier-Stokes equation by calculating for each particle the necessary quantities, such as pressure gradient, viscosity, gravity, and surface tension, and applying the changes in position and velocity to the particles.

After running the simulation, we reconstruct the surface created by the particles in two ways. One is by using Marching Cubes to generate a triangulated surface in OpenGL. This surface is created by finding points in space which have density above threshold and calling those points part of the fluid. The other method is by calling the ray tracer we created for Assignment 2 and using ray marching on the particle density field. With added reflection and refraction in the ray tracer, we get a cool looking surface.

For this ray traced image we reused our ray tracer from Assignment 2 and implemented a new class extended from the Shape superclass to represent the particle density field. From the SPH program we output the particles to an .obj file format. The ray tracer parses the file and initiates the tracing process. We call an intersection routine on the particle field using ray marching to determine if and when the ray encounters a point near a threshold density. To determine the density and normal vector at a point we reuse the kernel functions from the fluid simulation. In addition we reuse the neighbor structure to accelerate the processing of the particles within the ray tracer.

The above image was created by performing the marching cubes algorithm on the density field created by the particles and the kernels. The algorithm breaking up the container of the fluid into a grid and considering the density values at each of the eight vertices. Where the vertex densities are above a certain threshold we have different cases to consider for how to draw the triangles. We implemented the marching squares version ourselves for 2D simulation (which requires considering 16 different cases when drawing triangles) but used code found online to perform marching cubes (which requires 256 cases).

The above image shows the particles in the scene drawn as spheres. There are approximately 2000 particles in the scene. Below the image with the reconstructed surface was created from the same frame in the dam break simulation as these particles.

Idea behind SPH: