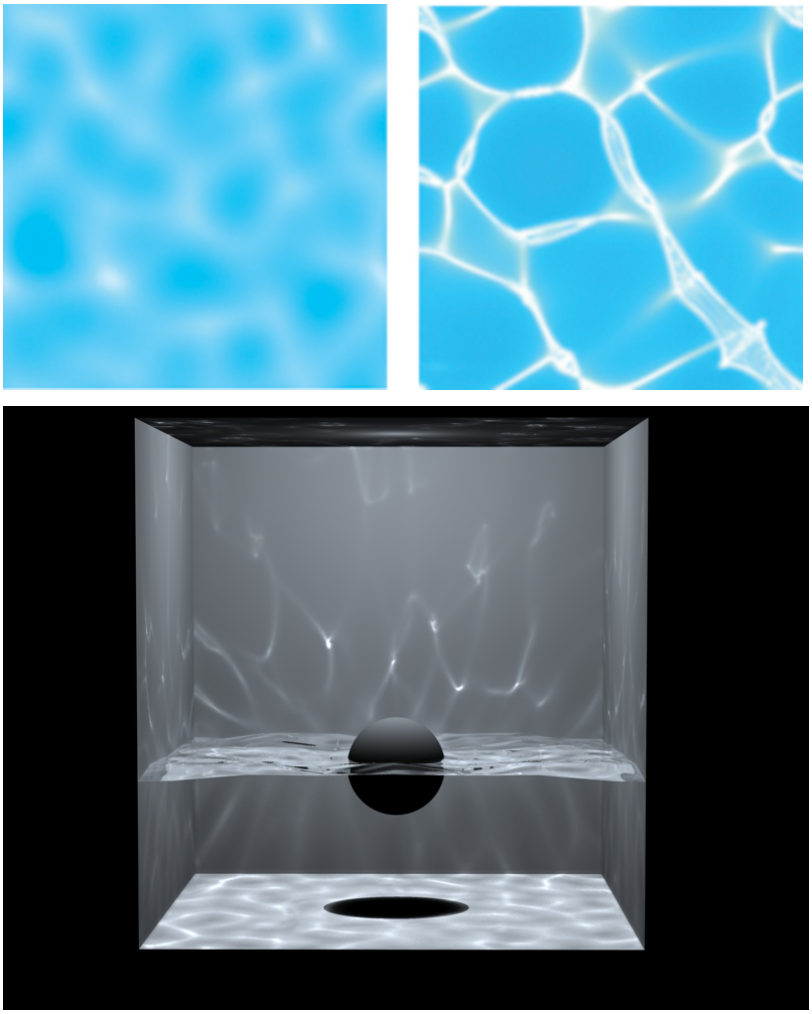
C. Begin Planning Your Final Project



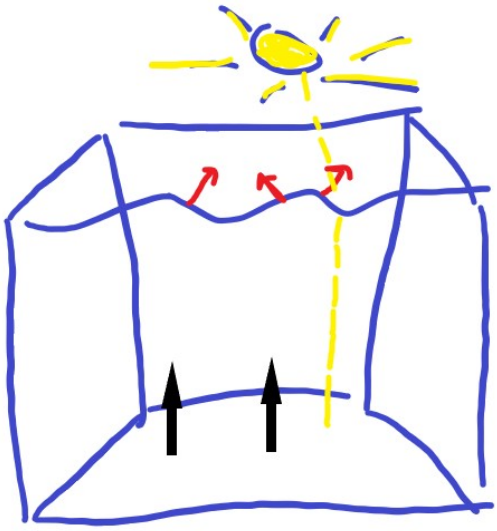
For my topic I chose to look into caustics, mainly, because I really want to create a water shader. Before looking up caustics I didn't have any idea how a person might go about making cool caustic effects.

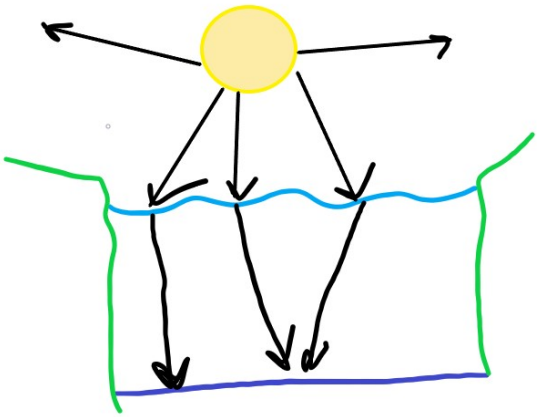
My first guess for how to go about doing this was to simply create a texture and blend it with an objects base texture. This actually wasn't too bad of an idea. I read from different places that this could work as a cheap way of adding the effect. However, depending on how it is implemented it seems to be the easiest to notice as being fake. So this might work in situations where a person will pay little to no attention to water.

To get real-time caustics I realized that the caustic effect would have to reflect the movement in water. I learned from readings and from class that a regular person has a difficult time noticing a physically correct caustic effect. So you could get a variety of real looking caustic effects without having to calculate light 100% correctly. This means that the best ways to approach caustics is by attempting to get a simple approximation about how light should be reacting. This lead to to an NVIDIA page that gave a good explanation for how to get cheap caustic effects.

I learned that the way you approach calculating caustics is by checking to see how light reacts to coming in contact with water. The brute force force method of doing this is by shooting out several light rays from a light source, this is forward ray tracing. You check to see if the rays come in contact with water or bounce off of other objects. This is very costly because only a fraction of the rays will actually make contact with water. This means that a lot of calculations are wasted and don't affect the end result. The second way to go about simulating caustics is to do backward ray tracing. This involves starting from the object and creating rays that finish at the light source. This way you know for sure that the rays hit the water, however, it is still costly to determine whether they actually hit the light source.

The approach used in the NVIDIA guide tries to cut down on a lot of the calculations by making assumptions before hand. This will create caustics that look real to people however, are completely wrong in a physical sense. The first assumption is that the light source is directly above the object. The second assumption is that the object is lit by rays that come directly above the point of interest. So the way you calculate caustics is by shooting a vertical ray for each vertex in the mesh. You compute the normals for the water above the mesh and use Snell's Law to get the adjusted angle of the ray. This ray shoots up into the air. You then have to determine whether this ray would have originated from the light source directly overhead.





The above sketches are my best attempt at trying to illustrate how I believe that you achieve caustics.

Part2)

I have formed a group with Trenten Kaufeldt and Robert Gaines.