Caustics

* Waves in water result in the refracted and reflected light bending in different angles and producing patterns on the surfaces under and near water.
* The patterns are created by regions where more light rays intersect.
* This is called caustics.
* Caustics can also occur when light passes through other materials as well such as curved glass.

How are they simulated

* Rasterization works but from one point of view (nvidia video)
* Ray tracing can be done for real-time but needs some short cuts
* Commonly an animated texture in videogames
  + Does not preserve realistic looking shadows
* Splatting (nvidia video)
* Most common: approximate intersections of rays on surface
* Vertex connection and merging (chaos blog)
  + Very slow
  + Over samples rays
  + Combines photon mapping and BDPT
* Corona (chaos blog)
  + Reverse path tracing and phton mapping
  + Cheaper than VCM

Math behind how the code simulates them

* Path/ray tracing
  + Follow rays sent out from pixels and compute pixel color based on what ray hits and reflects and refracts from
* When rays go through water, they refract at different angles (snell’s law)
  + When water surface is disturbed, more rays focus on some spots for lighter spot and some areas are not very focused darker spots

Notes:

* Code uses r, g, b to access components of vectors instead of x, y, z
* Look at environment shader for caustic info
  + Shaded color = underwater color \* computed light intensity
  + Light intensity is computed from caustic information calculated in [different file]
* Shadow mapping used for shadow computations
* Raycater is used to determine where mouse hits
* Caustic is calculated and stored as a texture
* Sides of bowl emulate reflection by using the environment as its texture.
* Fresnel equation used to shade water (and also glass)
* Uses render target to make textures
  + <https://webglfundamentals.org/webgl/lessons/webgl-render-to-texture.html>

Things I changed

-adding bowl

-use LatheGeometry to create a ‘bowl’ looking geometry

let points = [new THREE.Vector2(0,0), new THREE.Vector2(-1,-0.5),

              new THREE.Vector2(-2,3.5), new THREE.Vector2(-1.9,5),

              new THREE.Vector2(-1.5,5.5), new THREE.Vector2(-1.8,5.5),

              new THREE.Vector2(-1.8,6), new THREE.Vector2(-2,7)];

const bowlGeometry = new THREE.LatheGeometry(points);

bowlGeometry.computeVertexNormals();

bowlGeometry.scale(.25,.25,.25);

bowlGeometry.translate(0,-1,0);

bowlGeometry.rotateX(Math.PI/2)

Sources:

<https://en.wikipedia.org/wiki/Caustic_(optics)>

<https://www.chaos.com/blog/what-are-caustics-and-how-to-render-them-the-right-way>

<https://developer.nvidia.com/blog/working-with-ray-traced-water-caustics-in-dxr/>

<https://medium.com/@evanwallace/rendering-realtime-caustics-in-webgl-2a99a29a0b2c>

<https://medium.com/@martinRenou/real-time-rendering-of-water-caustics-59cda1d74aa>

house map from

<http://www.flamingpear.com/flexify-for-cube-maps.html>

nvidia building skybox from

<https://web.engr.oregonstate.edu/~mjb/glman/Examples/CubeMap/?C=S;O=A>

marble texture from

<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.istockphoto.com%2Fphotos%2Fblack-marble&psig=AOvVaw2A2WeAxePN9bcrB4pGMAxM&ust=1670638189323000&source=images&cd=vfe&ved=0CBAQjhxqFwoTCIirvLa66_sCFQAAAAAdAAAAABAF>