

Real-Time Rendering of Ocean Water

Aaron Hornby
10176084

- **GOAL:**

I would like to attempt the implementation of real-time rendering of a simulation of ocean water/waves. The results should provide a well-documented open-source repository that can be studied by other students or others with interest in this topic. The project will primarily benefit hobbyist game developers who wish for an easier entry point into a highly popular feature of many games.

- **CHALLENGE:**

This project will be a challenge since it incorporates aspects of rendering, modeling and animation. The rendering component will of course be the main focus, and could be used as a way to minimize the modeling effort (e.g. bump/normal/displacement mapping). The animation will probably be a very simple looping mechanism.

Traditionally, water is rendered as a texture-mapped quad. This doesn't look very realistic and isn't very extensible to many modern use cases. The referenced approaches (included below), vary from using more realistic techniques for rendering planar water to actually perturbing the surface with displacement maps.

- **APPROACH:**

I can use an iterative approach to building up the project, through consulting my references. I can start by attempting the simpler planar method, with the surface being static (no animation). Once I have achieved a consistent level of realism, I can then introduce a looping animation. I can then test how well the simulation scales and its impact on the rendering time. Furthermore, I can play around with the displacement mapping method. Finally, optimization will come into play at some point yet to be determined.

- **METHODOLOGY (needs more work):**

The task list is as follows (builds upon what is outlined by [Johanson 2004])...

1. Generate plane with appropriate spatial data structure
2. Discretize the perturbation of the surface (e.g. use height maps)
3. Render the plane as a perfect mirror and add a skybox and sun with parameters to control colours to be used to test water

4. Apply the rendering/shading techniques in one of the papers
5. Add a simple animation
6. Attempt to scale 1 patch of water into a large area (probably optimizing with LOD techniques)
7. Improve rendering by adding in effects such as fog, foam or refracted objects/fish underwater

- **METRICS:**

Fortunately, this topic has been researched a ton, and thus I should have ample reference material (images, video, simulations, other applications, etc.) to compare my results to.

My application should meet at least the following criteria:

1. Able to run at >30fps (preferably 60fps) on the Linux graphics lab computers.
2. Provide a simple UI to adjust parameters (such as sun position/angle, surface “bumpiness”, animation speed, etc.)
3. Suitably realistic for real-time applications, with an ability to scale up or down the simulation.

- **SUMMARY:**

This project will serve as a great learning experience for myself in implementing more advanced rendering techniques. Of particular use will be gaining experience in discovering the challenges with optimizing for real-time applications, while maintaining visual fidelity. I will also develop a better ability to work with researched work and hopefully build on their efforts.

- **REFERENCES:**

<https://fileadmin.cs.lth.se/graphics/theses/projects/projgrid/projgrid-hq.pdf>

<http://www.xtong.info/rtwave.pdf>

<https://www.hindawi.com/journals/mpe/2016/9034649/>