Real-Time Water Rendering

Degree programme: BSc in Computer Science | Orientation: Computer Perception and Virtual Reality
Thesis advisor: Prof. Marcus Hudritsch
Expert: Dr. Harald Studer (Optimo Medical AG)

Rendering large bodies of water is still an open problem in computer graphics. On the one hand, because no physical model exists which is able to represent at the same time the motion of deep and shallow ocean waves. On the other hand, because computational time is limited to less than 1/60 seconds in real-time applications. We implement the projected grid level of detail technique into the game engine Pyrogenesis. Ocean waves are synthesized from Phillips' spectrum.

Motivation

Many models trying to describe the shape of ocean water exist, either representing deep ocean water or shallow water. However, the former usually do not consider the interaction between objects and the surface. The latter compute the motion of each water particle, limiting the scaling. Thus, hybrid approaches have to be taken.

Problem statement

The aim of this bachelor thesis is first to find a method which produces realistic ocean waves. By «realistic» the following properties are meant: wave shapes that resemble to deep ocean waves, light effects, foam and white caps. The light effects comprise reflection, refraction and color attenuation according to the water's depth. The second goal is to implement it in an open source project needing either a water system or an overhaul, without causing a noticeable frame drop.

Approach

We choose to implement the projected grid as level of detail technique. It consists of a grid of even spaced vertices in screen space. Those are then projected onto the water plane, allowing a higher mesh resolution towards the viewer than at the horizon. Figure 2 shows a diagram of the projected grid idea. The dotted line represents the grid in screen space. The red line on the ocean surface is the grid after transformation into world space.
Ocean waves are synthesized from Phillips' spectrum, a popular approach taken in industrial productions such as games and movies. Three vector displacement fields are computed at the application's start up, with different initial parameters. They are then summed up and combined with a time scale.
In order to provide better visual accuracy, we implemented light reflection, refraction and color attenuation based on the water's depth.
The elements mentioned above were added into 0 A.D.'s game engine, Pyrogenesis, written in C++. The result can be seen in Figure 1.

Conclusion

Although our implementation runs at a similar frame rate than the original one, there is a noticeable delay at start up when creating the three vector displacement fields. Additionally, our implementation does not take in account water-object interactions.

Samuel Luc Gauthier

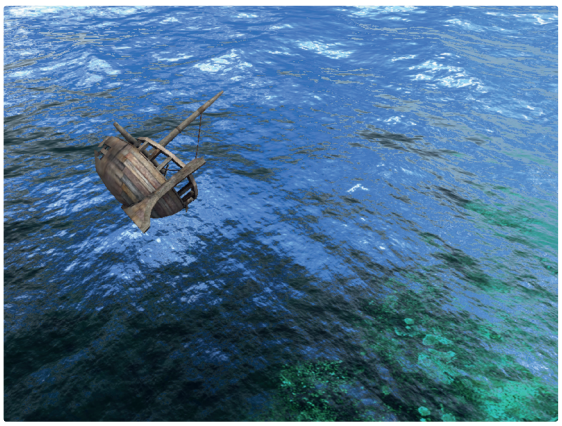

Figure 1: Our water implementation in the game engine Pyrogenesis.

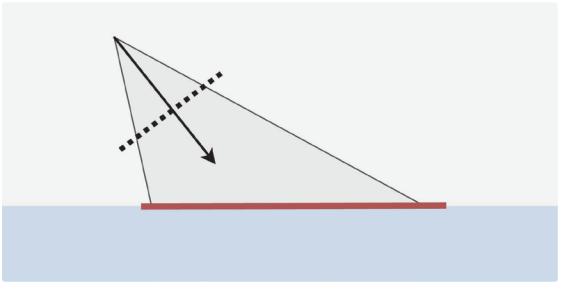

Figure 2: Projected grid diagram.