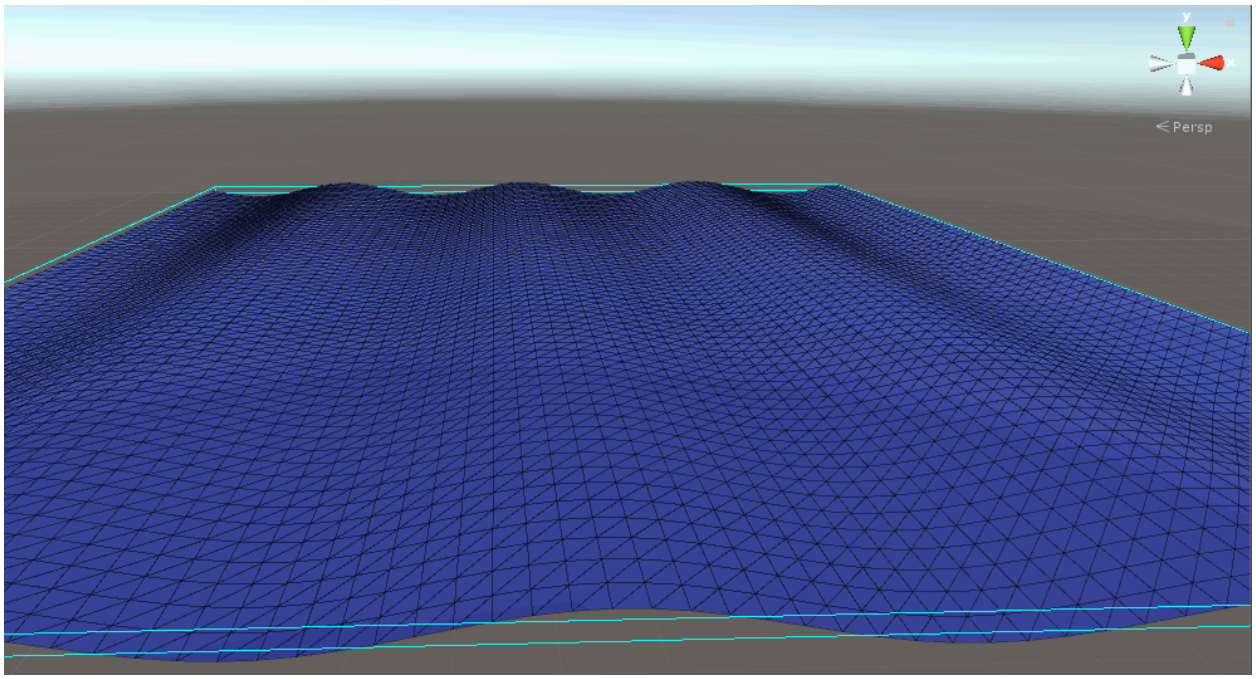
Ægir

A scalable large body mesh deformation system.



Created By:

Jacob Pigg

Game Programming

Last Updated: 04/14/2020

Technical Field

This project pertains to the technical fields of programming, technical art, and rendering.

Background Information

Sea of Thieves(Rare, 2018) is often considered to have some of the most beautiful water in a video game. The complexity and personality of the water in that game was a huge inspiration for coming up with the idea of Ægir. When looking on the Unity Asset Store for water assets, you will often find that nothing both looks dynamic and is also performant, that gave me the goal of Ægir, to create a performant and dynamic water system that could be used in any game. Some other major influences on my interest in graphics programming, which ultimately led to me choosing to pursue the Ægir idea, are Jean-Philippe Grenier's real-time and reactive river editor (Grenier, 2018) and the reactive water shaders created by Matt Wilde (Wilde, 2018) for Campo Santo's upcoming game "In the Valley of Gods" (Campo Santo, TBD). These impressive systems all inspired me to learn how to do something similar, but make it performant and available to everyday developers like myself.

Prior Art (legal term)

With water being a major factor for so many games, there are obviously a ton of water-related products on the Unity Asset Store. I specifically chose not to search other asset stores, such as the Unreal Engine Marketplace, for anything other than inspiration, since I do not see these products as direct competition with them being made for a separate engine than what I plan to release on. So, here I will present four of the most popular water assets currently on the Asset Store. Each of these assets seem to focus on a specific need when creating water in a project, and have grown to be the most popular in each of their own respective focuses on water mechanics. The first product is R.A.M - River Auto Material (NatureManufacture, 2017), this asset uses flow maps and shaders to determine the orientation and slope of a mesh to create water that gives the illusion of flowing. Using only shaders, this asset creates a flat looking water that also tends to be unreactive to the environment. The next assert is Cascade (EarthShaping, 2018), which focuses more on allowing water to have verticality, such as waterfalls. The asset claims to use multi-thread and a similar multi-tile approach to Ægir, however, it states that this is used for terrain carving, whereas Ægir utilizes the GPU and multi-tile approach for performant mesh deformation on the water itself preventing it from looking "flat", like most water assets currently on the store. The third asset, AQUAS (Dogmatic,2016) creates very realistic and great looking water but only offers a series of flat shaders which prevent the water from appearing to naturally and organically flow. The final asset, Suimono (Tanuki Digital, 2012), achieves a visual result similar to Ægir, with the water having the capability of offering large scale movement and wave

height, however, is not nearly as performant as Ægir since it clearly states that it is not compatible with mobile or VR, most likely meaning this system is done using the CPU for mesh movement, or that the shader does not have the ability to “scale” to hardware requirements. Ægir will have the ability for users to enable/disable components and tweak properties to fit their performance needs.

Project Description

Ægir is a scalable, reactive, customizable, and performant large body water deformation system built in Unity. The delivered result of this project will include the mesh deformation system consisting of multiple layers of Gerstner (trochoidal) waves, a shader with subsurface scattering, caustics for below the surface of the water, a GPU accelerated particle system using compute shaders, and depth/object aware fluid movement. Volumetric rendering, a flow painter system, screen-space and signed distance field particle collisions, and occlusion volumes are all planned as secondary goals which will be released after the given timespan of this project has elapsed. Ægir will be an easy to use customizable asset released on the Unity Asset Store for developers to drag and drop into their scenes and tweak in any way necessary to achieve the results they require for their games and environments. The audience of this project is Unity developers looking to add high quality and reactive water simulations into their projects.

Innovation Claim

Ægir offers a completely scalable water system with performance by default through utilizing the GPU for computations and simulation. Ægir also offers large scale, physically reactive, mesh deformation to make the water really come alive in a scene.

Usage Scenario

An alternate usage for Ægir could really be any form of large scale mesh deformation. While the main goal of the project is focused on water systems, the core trochoidal vertex motion combined with a responsive shader and GPU compute shader particles systems allow for a number of uses. Such as mud or snow that reacts to characters moving through it, terrain destruction, or if using just the rapid multi-tile mesh generation, then procedural terrain generation.

The primary demographic for users of Ægir would be game developers or users of Unity looking to implement a performant yet high-quality water system. The demographic is difficult to narrow down since it can be used in a vast number of applications and with the built-in customizability allow for nearly any art style.

Evaluation Criteria

The following questions will be used to evaluate the project and identify its level of success:

1. Performance
    a. Does the system run well on a wide range of devices?
    b. Does the system perform acceptable well in the editor relative to how it does in a build?
    c. Can large scale environments be generated quickly using Ægir?
    d. Does Ægir take minimal CPU and GPU times relative to a user's own mechanics and scripts?
2. Customizability
    a. Does Ægir allow for easy and intuitive customization?
    b. Are the features and options presented to the user relevant to the customization trying to be made?
    c. Can Ægir be adapted to the specific art style of the user?
    d. Can performance options be tweaked by the user?
    e. Is it easy to get started with Ægir? (minimal learning curve)
3. Appearance
    a. Does Ægir offer a water simulation resembling that of realistic waves?
    b. Do the shader and particle systems respond appropriately to the situation or environment?
    c. Can wave and water movements appear different based on world space location?
    d. Is the default example scene a good representation of all of the features included in Ægir?

Objectives and Tasks Associated with the Project

Goal - To develop a scalable and performant mesh deformation, shader, and particle system that combine to create a seamless water effect for use by Unity developers looking to step up the quality of their projects.

1. **Objectives -** Build a performant mesh deformation system with scalability in mind. **-COMPLETED**
    a. **Activity 1:** Build a system for creating the individual mesh tiles **-COMPLETED**
    b. **Activity 2:** Build a tile spawner that creates each individual tile in its appropriate place in world space **-COMPLETED**
    c. **Activity 3:** Implement an algorithm that moves the vertices in a trochoidal motion that reflects realistic wave movements **-COMPLETED**
2. **Objectives -** Create reactive water shaders
    a. **Activity 1:** Create a shader that is aware of its orientation in the world **-COMPLETED**
    b. **Activity 2:** Add the ability for the shader to react to collisions or objects intersecting its mesh - Targeted **April 2020**

    c. **Activity 3:** Add the ability for the shader to react and collide with itself - Targeted **May2020**
    d. **Activity 4:** Create textures for the shader to give it a more realistic look **-COMPLETED**
    e. **Activity 5:** Implement caustics and subsurface scattering into the shader for more realistic lighting - Targeted **April/May 2020**
3. **Objectives -** Implement particle systems for selling the effects
    a. **Activity 1:** Create a particle system that acts appropriately to collision velocities and orientation - Targeted **May 2020**
    b. **Activity 2:** Optimize particle system for lower-end devices - Targeted **June 2020**
    c. **Activity 3:** Make particles collide with objects in the world - Secondary Objective Targeted - **August 2020**
4. **Objectives -** Design the system from a customization standpoint
    a. **Activity 1:** Create plenty of options for generating the meshes **-COMPLETED**
    b. **Activity 2:** Allow for customization of wave properties and flow - Properties **COMPLETED,** Flow is a Secondary Objective Targeted **June 2020**
    c. **Activity 3:** Allow a user to specify different "intensity" areas with different wave sizes and water appearance (dark water and large waves for storms or calm aqua waters for a lagoon) - Targeted **July 2020**
    d. **Activity 4:** Develop multiple quality and performance options for a user to easily access - Targeted **July 2020**

Description of Design Prototype

The initial design prototype featured mesh generator, tile generator, and tile motion scripts, which allowed a user to generate a grid of any specified size, number of tiles, and resolution. The motion is done using the tile motion script which at this point uses a vertex's position in world space to determine where it should be on the y-axis at a given point in time evaluated on a sine wave. The wave height and wave width properties on the tile motion component allow a user to specify the shape of the sine wave by determining what each value should be multiplied or divided by. The current version of the project is almost entirely different from the initial design prototype and features a much better looking water surface as well as GPU driven performance making it exponentially faster on meshes of roughly the same size.

Evaluation Plan

Ægir will be evaluated by users of Unity to determine how easy and customizable the project is. The project will be open to evaluation by fellow students and peers. The applicable evaluation questions will be asked or presented to those testing out the project as a way to get user feedback and determine the completion percentage of Ægir at that time. For non-applicable questions,

such as does Ægir run well on a variety of devices and platforms, data from testers and self-conducted field test results will be compiled to determine an answer.

Project Completion Assessment

The current project sits at a semi-completed/Minimum Viable Product state with many new ideas planned for future development. Currently, the mesh generation system has been rewritten to be more efficient, a custom editor window has been created for customization; allowing users to create an Ægir mesh object of any size, with any number of tiles, containing between 2 and 255 vertices per axis on each tile. Currently there are a few variants of the Ægir shader, each testing out different mechanics. One shader implements multi-layered trochoidal waves, another has transparency, screen-space refraction using grab passes, and depth fog, and the third tests out directional flow from a texture. A fourth shader which unifies everything is currently in the works and will be completed within the next week or so of this writing. I have also completed a re-working of the UI window to allow for ease of use and customization. The next step is to begin working on implementing the reactivity portion of the shaders, which will allow a user to pass in physical properties of collisions to generate an appropriate wave or ripple response.

Appendices

Appendix A: References -

Campo Santo. (tbd). In the Valley of Gods - A New Video Game from Campo Santo. Retrieved from https://inthevalleyofgods.com/

Dogmatic. (2016, January 8). AQUAS Water/River Set. Retrieved January 20, 2019, from https://assetstore.unity.com/packages/tools/particles-effects/aquas-water-river-set-52103

EarthShaping. (2018, February 16). Cascade - River, Lake, Waterfall and more. Retrieved January 20, 2019, from https://assetstore.unity.com/packages/tools/terrain/cascade-river-lake-waterfall-and-more-106072

Finch, M. (2004). Chapter 1. Effective Water Simulation from Physical Models. Retrieved from https://developer.nvidia.com/gpugems/gpugems/part-i-natural-effects/chapter-1-effective-water-simulation-physical-models

Grenier, J. (2018, October 18). Realtime River Editor. Retrieved January 20, 2019, from http://jpgrenier.org/riverrenderer.html

NatureManufacture. (2017, October 12). R.A.M - River Auto Material. Retrieved from https://assetstore.unity.com/packages/tools/terrain/r-a-m-river-auto-material-101205

Rare. (2018, March 20). Sea of thieves [Computer software]. Retrieved from https://www.seaofthieves.com/

Svensson, J. (2018, December 4). Water/Ocean Shader. Retrieved from https://medium.com/dotcrossdot/water-ocean-shader-9173e0977f98

Tanuki Digital. (2012, September 17). SUIMONO Water System. Retrieved January 20, 2019, from https://assetstore.unity.com/packages/vfx/shaders/suimono-water-system-4387

Tardif, A. (n.d.). Water Walkthrough. Retrieved from http://alextardif.com/Water.html

Wilde, M. (2018, March 16). From the Blog. Retrieved January 20, 2019, from http://blog.camposanto.com/post/171934927979/hi-im-matt-wilde-an-old-man-from-the-north-of