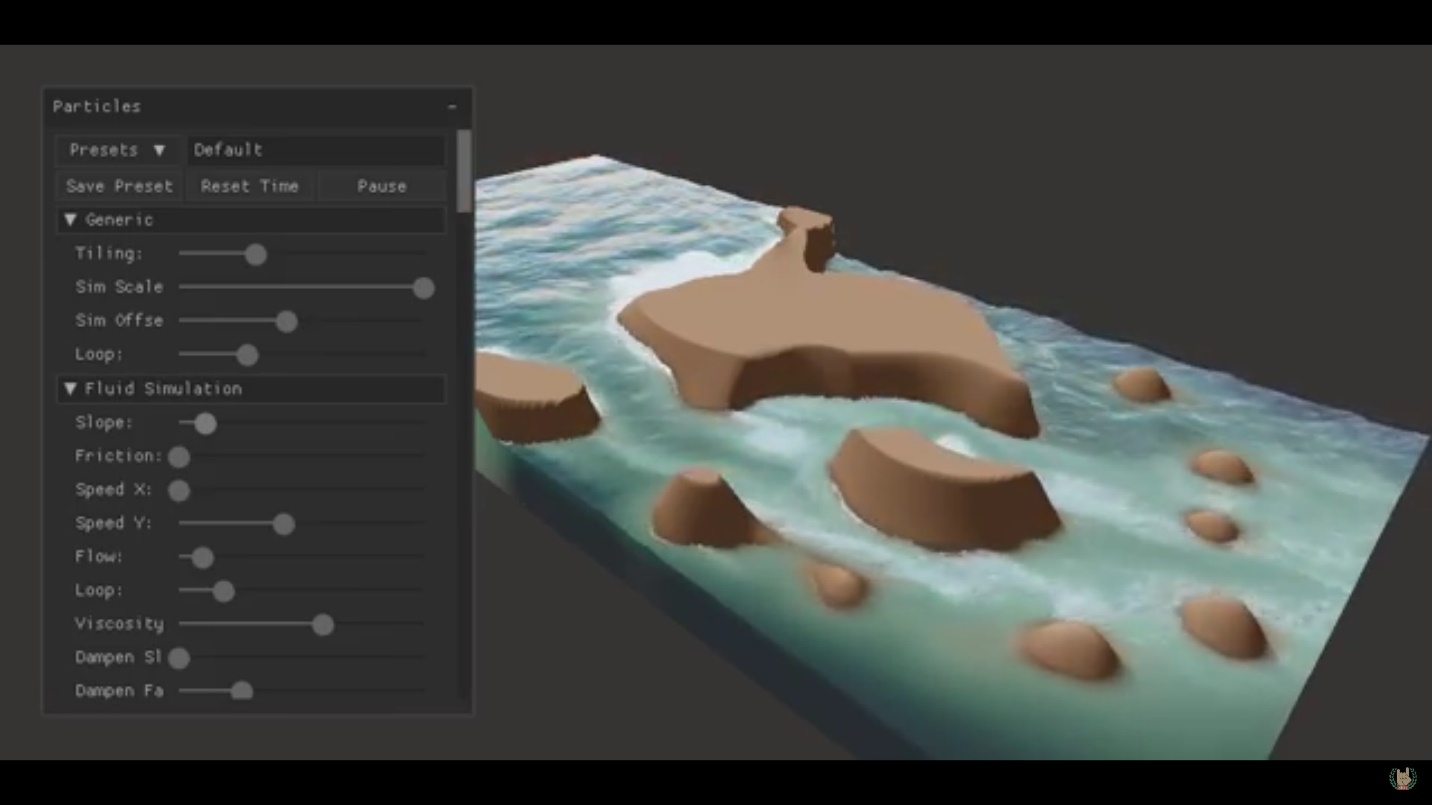
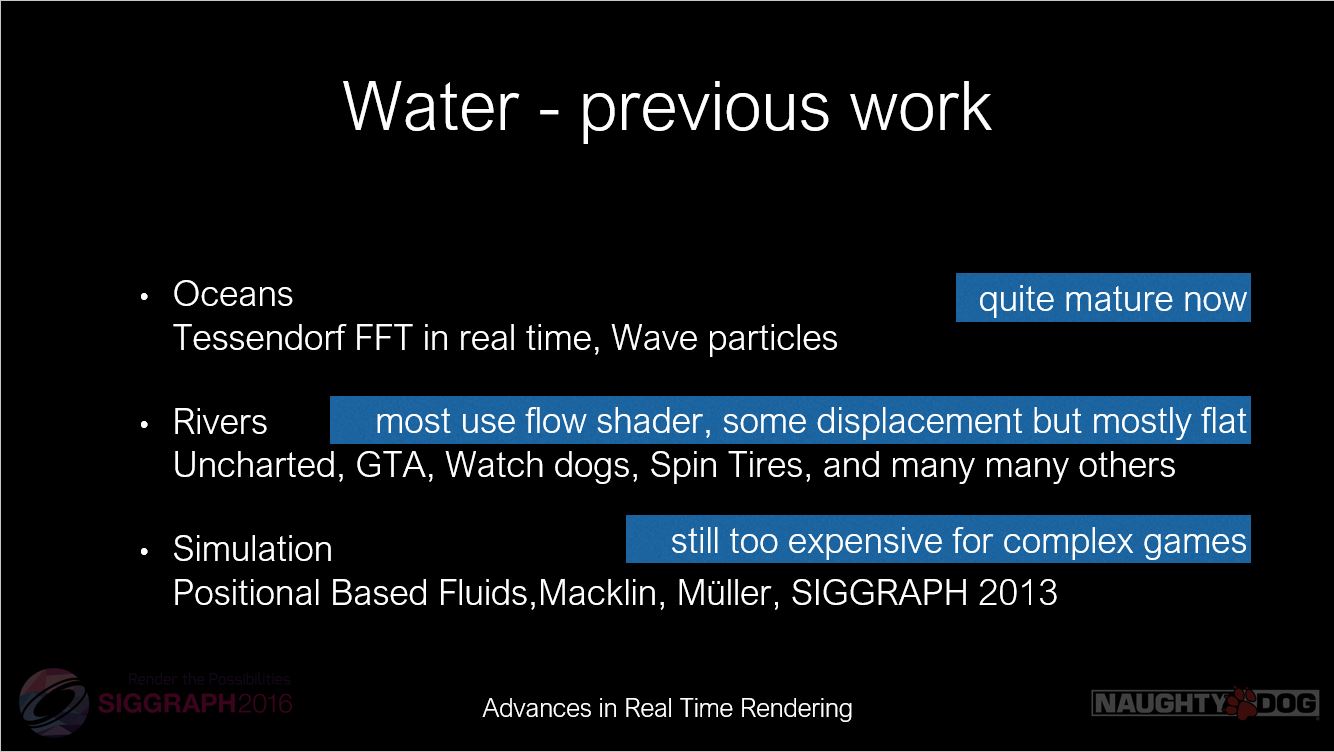
Wave Particles with Interactive Vortex

In game industry water can be divided into two domains: ocean and river. This project focuses on river. Not too long ago, a demo of a real-time river editor was developed by a graphics programmer from Ubisoft. We think what he did is awesome and want to replicate it (as possible as we can).



Before Wave Particles, rivers are rendered using flow maps. Flow maps are generated by fluid simulations or created by artist or even both. Flow maps are used for advection of water properties such as normal and uv. This technique can give the river a deterministic fluid look, but it fails to capture the sense of height compared with the less controlled ocean rendering techniques, which basically stack several noise functions together to generate the waves. Stacking noise functions together only works for ocean because the ocean does not flow. When you want the water both flow and have the sense of height, a new method should be used.



(picture from Rendering rapids in Uncharted 4 by Carlos Gonzalez-Ochoa, Siggraph 2016 Advances in Real-Time Rendering in Games course)

The new method is Wave Particles. Initially, Wave Particles were introduced into the real time water rendering [naughty dog, uncharted 3] to simulate local high frequency wave of oceans. Then it was adopted and adapted to render rivers [naughty dog, uncharted 4] paired with stacking technique.

The significance of the above demo created by Jean-Philippe Grenier from Ubisoft is that it generates flow map in real-time. That is why the user can interactively create rocks to block the river as we can see in the video. When performing advection on height and normal, previous implementations [valve, naughty dog] use noise and blending multiple uv sets to minimize repetition and pulsing caused by advection of a finite sized texture. But in this demo the developer used a new technique called wave profile buffer to solve the repetition and pulsing.

Goals:

1. Simple wave particle rendering.

2. Combine wave particle with flow map.

3. Generate flow map in real-time.

\*4. Identify and solve the problems caused by advection of height, uv, normal.

\* = extra features

Milestones:

Milestone 1:

1. DX12 frame work

* Compute shader
* Tessellation shader
* Basic Interaction
* Basic UI

1. Basic wave particle rendering

* Single particle

Milestone 2:

1. Advanced wave particle rendering

* Multiple particles
* Particle boundary interaction
* User interaction (create bump or dent)

1. Combine flow map with wave particle

* Find some interesting flow maps
* Use flow map to advect wave particle

1. Improve rendering method

* Foam
* Subsurface scattering

Milestone 3:

1. Generate flow map in real-time

* Fluid simulation
* Create blockers (rocks and etc.)
* Advect properties using simulation result

Final:

1. Finish unfinished work
2. Identify and solve the problems caused by advection of height, uv, normal

References:

<https://80.lv/articles/river-editor-water-simulation-in-real-time/>

<http://www.cemyuksel.com/research/waveparticles/waveparticles_sketch_slides.pdf>

<http://www.cemyuksel.com/research/waveparticles/cem_yuksel_dissertation.pdf>

<http://advances.realtimerendering.com/s2016/s16_ramy_final.pptx>

<https://prideout.net/blog/old/blog/index.html@p=58.html>

<https://developer.nvidia.com/gpugems/GPUGems/gpugems_ch38.html>

<http://pub.ist.ac.at/group_wojtan/projects/2018_Jeschke_WaterSurfaceWavelets/WaterSurfaceWavelets.pdf>

<https://www.gdcvault.com/play/1015309/Water-Technology-of>

<http://advances.realtimerendering.com/s2010/Vlachos-Waterflow(SIGGRAPH%202010%20Advanced%20RealTime%20Rendering%20Course).pdf>