User’s Manual

CSCI 711 Final project

Noya Cai, Tsingtao Zhang

# Program Description

This ocean simulation program renders an ocean wave in real-time and interactively\*, with the feature of whitted ray tracing, procedurally generated waves and a fast caustics effect.

The user can click on the ocean surface to create the ripple effect.

The whitted ray tracing is developed throughout the checkpoints in CSCI 711.

The procedurally generated waves are constructed by a series of [Gerstner waves](https://www.tandfonline.com/doi/pdf/10.2991/jnmp.2008.15.S2.7#:~:text=Gerstner%E2%80%99s%20wave%20is%20given%20by%20emb,x%20%3D%20a%20%2B%20sin%20m%28a).

The fast caustics effect is created in real time and inspired by [Evan Wallace](https://madebyevan.com/)’s [Rendering Realtime Caustics in WebGL](https://medium.com/@evanwallace/rendering-realtime-caustics-in-webgl-2a99a29a0b2c).

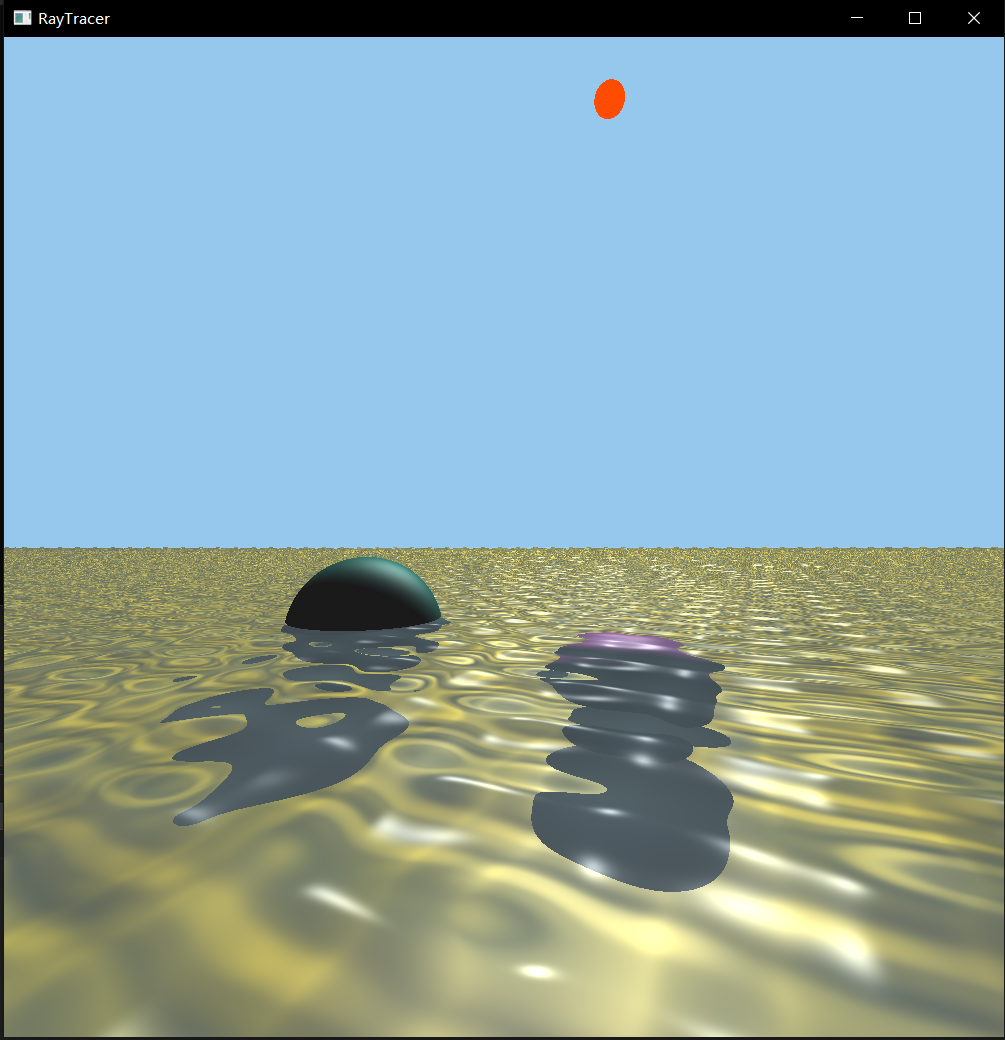
\*The interactive part is done in the course IGME 740 as a final project.

# Input

No need for any input, extract the file and run the RayTracer.sln. However, do read the Command Sequence before you compile and run the program.

# Normal Output

Once you run the program successfully, you can see the ocean simulation played in real time, and you can click on the ocean to create a ripple, observe the waves and caustics effect. We also implemented the rotation of the sun and moon, which affects the environment lighting.  
Below is a screen shot of the program.



[CSCI711Final.mp4](https://drive.google.com/file/d/1L65ZH91Dx2d1JTCu2CHZgv1UXK4tcnPv/view?usp=drive_link)

# Exception Reports

If your machine is taking too long (over 10 mins) compiling the shaders, and you see nothing but black/white screen, consider your CPU and RAM.

This program requires about 22 GB of RAM, and 8 minutes to compile in AMD Ryzen 9 5900HS with Radeon Graphics, resulting in 15 FPS.

However, in Intel 12th Gen i7-12650H, the program reaches over 60 FPS.

If the command line shows “Too many instructions”, refer Command Sequence to check if you’re using integrated graphics.

# Program Limitations

As you can see from above, this program trades memory for speed. So we need quite a lot of RAM (over 22 GB) and a decent CPU to run. The program itself takes up 1.15 GB once extracted.

Again, for reference, this runs at 15 FPS in AMD Ryzen 9 5900HS with Radeon Graphics.

However, in Intel 12th Gen i7-12650H, the program reaches over 60 FPS.

The program runs bug free on the Windows 64 platform, but due to limited time, we couldn’t find another platform to test.

# Command Sequence

First, as we’ve tested in several machines, current mainstream standalone graphics cards cannot compile our massive shader. So sadly we can only use the integrated graphics card in GPU. If you’re lucky using Nvidia, here’s a guide to switch into the integrated graphics card.

1. Open Nvidia Control Panel, find “3D Settings”, and select “Manage 3D settings”.
2. Change the “preferred graphics processor” into “Integrated graphics”.

Once that’s done, you can extract the zip file, in the RayTracer folder, open RayTracer.sln, make sure in debug mode and x64, compile and run.

The shader this program uses is in CSCI711FinalCode/RayTracer/RayTracer/src/shaders, Water.vert and Water.frag respectively as vertex shader and fragment shader.

# Author Information

Noya Cai, [qc6338@g.rit.edu](mailto:qc6338@g.rit.edu)

Tsingtao Zhang, [qz3017@g.rit.edu](mailto:qz3017@g.rit.edu)