ERDC/ITL TN-22-1
September 2022

Ship Simulator of the Future in Virtual Reality

by Kelly Ervin, Karl Smink, Bryan Vu, and Jonathan Boone

INTRODUCTION: The Army's modernization priorities include the development of augmented reality and virtual reality (AR/VR) simulations for enabling the regiment and increasing soldier readiness. The use of AR/VR technology at the U.S. Army Engineer Research and Development Center (ERDC) is also growing in the realm of military and civil works program missions. The ERDC Coastal and Hydraulics Laboratory (CHL) has developed a ship simulator to evaluate bay channels across the world; however, the current simulator has little to no physical realism in nearshore coastal regions (Figure 1). Thus, the ERDC team is researching opportunities to advance ship simulation to deliver the Ship Simulator of the Future (SSoF). The SSoF will be equipped with a VR mode and will more accurately resolve nearshore wave phenomena by ingesting precalculated output from a Boussinesq-type wave model. This initial prototype of the SSoF application is intended for research and development purposes; however, the technologies employed will be applicable to other disciplines and project scopes, including the Synthetic Training Environment (STE) and ship and coastal structure design in future versions.

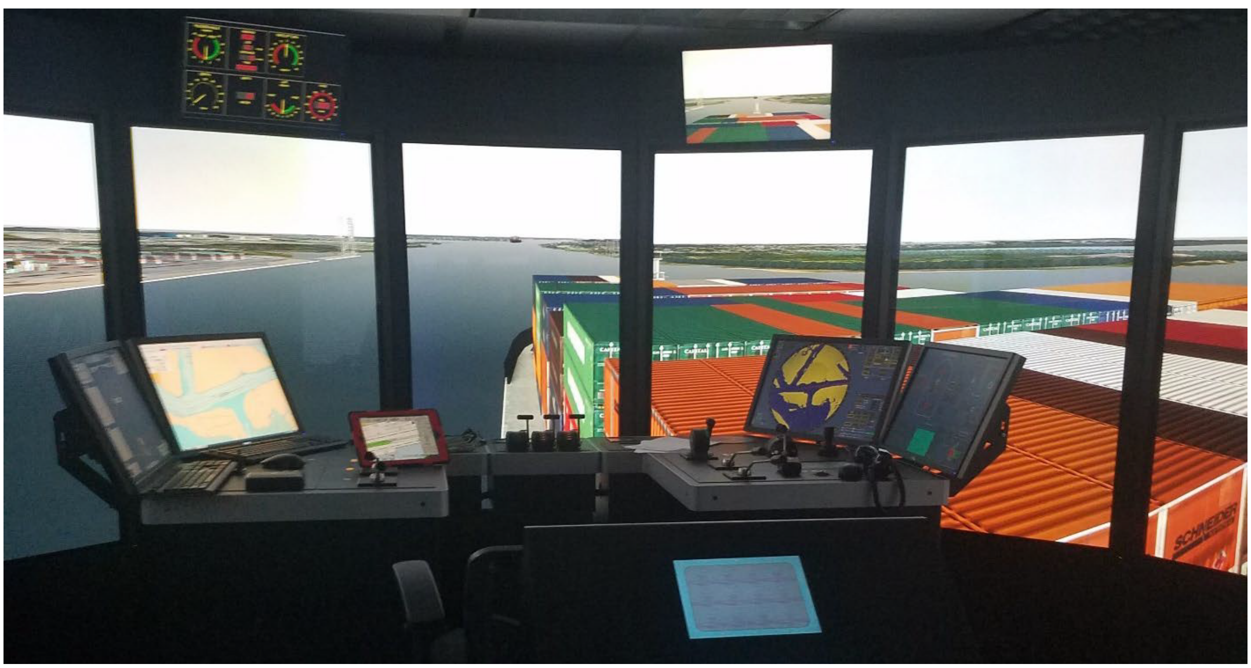

Figure 1. Current Ship Simulator located at ERDC CHL.

US Army Corps of Engineers®

BACKGROUND: To transition the current ship simulator capabilities to a VR platform, a solution must be an integration of numerical solutions for near-shore ship simulation response and proper visualization of the response of the simulated watercraft given the resulting parameters. The research team found that a custom VR software solution could be created using Unreal Engine 4 (UE 4) by expanding its capabilities through plugins. To promote cohesive VR development across ERDC, this engine was selected. "Unreal Engine is a complete suite of development tools for anyone working with real-time technology. It gives creators across industries the freedom and control to deliver cutting-edge entertainment, compelling visualizations, and immersive virtual worlds" (Epic Games, "Unreal Engine," n.d.). Blueprints, a visual scripting language native to UE 4, is the standard programming technique within the platform, but more fine tuning can be achieved with programming in C++. Additionally, the research team also explored the incorporation of ERDC's real physics data integration within a VR simulation using the output from near-shore wave computational models. FUNWAVE-Total Variation Diminishing (TVD), STWAVE, and Celeris Base are a few applications considered to fulfill this role, and a proof of concept was produced by ingesting wave frequency output from FUNWAVE-TVD into UE's data table feature.

Several plugins were found that show potential for VR and general UE development. As defined by the Unreal Engine documentation, "plugins are collections of code and data that developers can easily enable or disable within the Editor on a per-project basis. Plugins can add runtime gameplay functionality, modify built-in Engine features (or add new ones), create new file types, and extend the capabilities of the Editor with new menus, tool bar commands, and sub-modes. Many existing UE 4 subsystems were designed to be extensible using plugins" (Epic Games, "Plugins," n.d.). Plugins are important because they can save developers time, they are inexpensive, and some are free. There are two main types of plugins: game and engine plugins. Game plugins are project-specific, while engine plugins are available to all projects. Additionally, UE 4 comes with several built-in plugins that can be quickly toggled in the "Plugin" window. The team searched for plugins that would generate Boussinesq waves, but it was determined that such a plugin did not yet exist.

PLUGIN EVALUATION OVERVIEW: Water plugins are available in the Unreal Engine Marketplace and Git, but they were not purchased at this time. As development of the SSoF continues, the team will work to integrate existing ship simulator software into UE 4. When necessary, the team will develop new custom plugins to minimize conflicts and streamline the overall workflow for this and future projects.

The VR Expansion Plugin is a Massachusetts Institute of Technology (MIT) licensed and open-source standard plugin used for VR development at the ERDC Information Technology Laboratory (ITL). The plugin offers "a series of modules and tools with an emphasis on multiplayer and networking, locomotion, gripping, and solving hard problems in a flexible manner" (Statzer, n.d.). Several features have been incorporated for this project, such as locomotion and gripping.

The UE4GitPlugin improves upon UE 4's source control using Git: a popular and open-source Distributed Version Control System (DVCS). Git is a critical tool for large software development projects by allowing collaboration on different networks and managing tracking changes to project files. DVCS also provides a digital backup of project assets and enables multiple versions to be swapped and compared quickly. ERDC's public GitLab was used to host projects, and Source Tree and Git Bash were used for source control operations (Rombauts, n.d.).

Cesium for Unreal (Figure 2) is a free and open-source (under the Apache 2.0 license) plugin. It provides 3D geospatial terrain visualization using photogrammetry and optimized spatially indexed 3D-tiled map features. The plugin is still in early development and was therefore found to be unsuited for this project's use case. Cesium for Unreal is highly accurate and has a complete rendering of the entire globe. This makes it very memory intensive, which can slow down performance. The plugin has a lot of potential that is expected to grow through the partnership with Epic Games. According to Cesium, an on-premises solution exists that will be necessary to conform to the Army's cyber security requirements, unless a Government Cloud instance is setup (Cesium GS, n.d.).

Figure 2. City buildings generated in Cesium for Unreal. (Cesium GS, reprinted with permission).

UE 4.26 shows some impressive water rendering features that could prove to be useful in the development of hydrological simulations. At the moment, this feature only natively supports Gerstner waves: an approximation for deep water, offshore waves. Although the visualization is advanced, the water features do not include near-shore wave physics which could be achieved with Boussinesq-type wave modeling.

The UE 4.26 water features (Figure 3) come with three different presets: rivers, lakes, and oceans. These presets are built on top of UE's landscape tools, meaning their shape and size can be manipulated and defined by various means. UE 4 will automatically detect and merge separate, overlapping water meshes to provide a seamless transition between them. For performance, UE 4 automatically adjusts the Level of Detail (LOD) of water meshes based on an exposed "LOD scale" parameter in their respective Water Mesh Actor. Depending on the proximity to the viewed objects, geometry and textures are rendered with different fidelities in order to increase performance. Higher, more precise versions are displayed when the viewer is close to the objects. Lower, less precise versions are displaced the farther away the viewer gets, eventually being culled altogether (Epic Games, "Water," n.d.).

UE 4.26's base water features are defined in a single layer water material. Here, several rendering techniques are implemented to provide more realistic water visuals such as analytical caustics, beach foam, scattering and absorption, and automatically enabling and blending of post-processing materials. Additionally, the material provides height mapping, fluid simulation and wave generation, all of which have been abstracted within their own individual functions. The water material is opaque but appears translucent as water is rendered on a different layer than the landscape underneath it. Because the material is not actually translucent as far as the lighting engine is concerned, no calculations regarding refraction or pass-through are done. To get the visual effect, the pixels on that layer are rendered to the screen with an alpha value lower than 1. As far as the Engine is concerned, those that material is entirely opaque. This allows UE 4.26 to emulate translucency while greatly saving on performance (Unreal Engine 2020).

Figure 3. Unreal Engine 4.26 water demo scene in the SSoF VR.

COMPUTATIONAL MODEL OVERVIEW: The team researched near-shore wave computational modeling applications and how they could be incorporated into UE for more accurate ship simulation. The first application examined was FUNWAVE-TVD (Figure 4), a fully nonlinear, shallow-to-intermediate water phase-resolving, Boussinesq numerical wave model. It provides high-fidelity simulations for many coastal processes including nearshore waves, currents, wave breaking with run-up and overtopping, harbor resonance, infragravity waves, and vessel-generated waves. Many of its capabilities are only possible due to its ability to resolve the phases between different superpositioned wave frequencies. This high degree of accuracy, however, comes at the cost of the application being unable to run in real-time (Shi, n.d.).

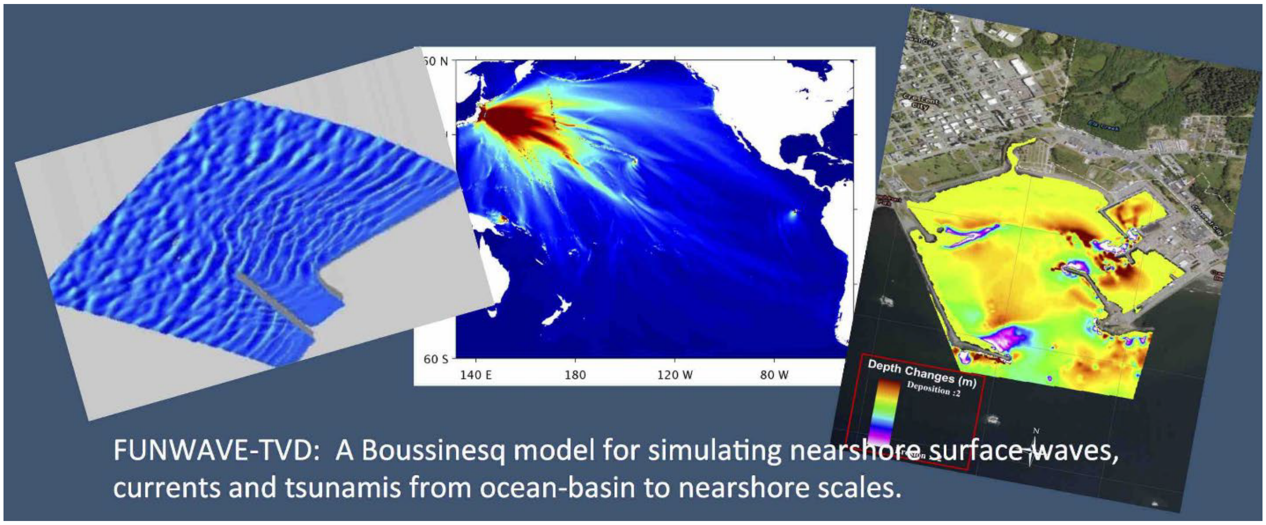

Figure 4. Boussinesq-TVD Example Visualizations of near-shore waves (Shi, n.d., reprinted with permission).

Dr. Patrick Lynett at the University of Southern California developed a coastal wave simulation software called Celeris Base (Figure 5). It provides a realistic model for nearshore waves by solving the extended Boussinesq equations using a hybrid finite volume-finite difference method. Celeris Base is the second iteration overhauling the original Celeris Advent. This major change replaced the original application's custom visualization system by generating inputs using Unity and C#. Those inputs are then sent over to a Direct3D (D3D) graphics processing unit (GPU) shader written in high-level shader language (HLSL). This vastly improves Celeris's extensibility and scalability. The visualization and simulation is handled with D3D, which allows for faster than real-time run times. Additionally, Celeris Base comes with some useful features, including a virtual reality view, real-time gauge plotters, and 360° video capture (Tavakkol and Lynett 2020).

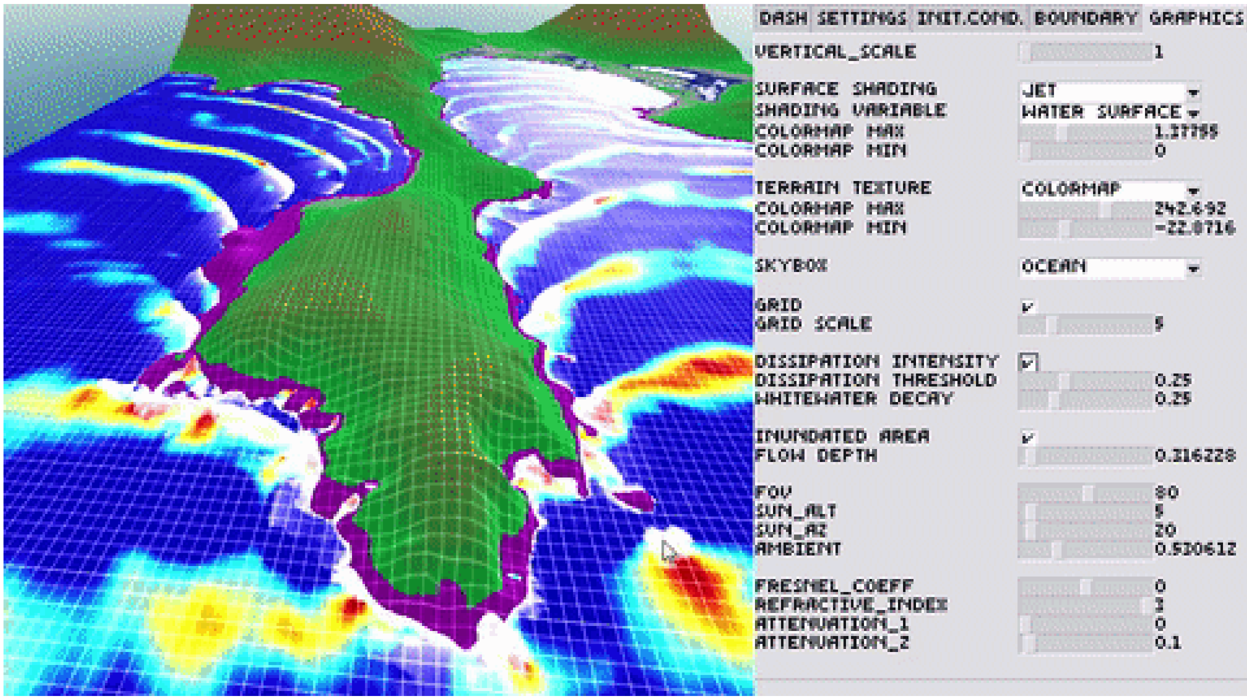

Figure 5. Celeris Base simulation of near-shore wave breaking and refraction (Photo courtesy of Sasan Tavakkol and Patrick Lynett).

Celeris Base is built on the extended Boussinesq equations, which is weakly dispersive and weakly nonlinear, whereas FUNWAVE-TVD is weakly dispersive and fully nonlinear. Celeris Base has a left-boundary wavemaker, while FUNWAVE-TVD has both a left-boundary and internal wavemaker, allowing for more flexibility and better capturing nonlinearity. And unlike FUNWAVE-TVD, Celeris Base does not present any viscosity type wave breaking simulations. It is for these reasons, along with its higher fidelity, that FUNWAVE-TVD was chosen to serve as our underlying nearshore-to-intermediate surface wave physics engine.

METHODS AND APPROACH: At the current stage, the research team was able to create a drivable ship in an immersive VR water environment and start the process of ingesting FUNWAVE-TVD output data into the UE software platform, through the use of data tables. Data tables (Figure 6) are gameplay elements that UE offers to store related data. They can be accessed in either C++ or Blueprints and allow UE to input and output data from comma separated values (CSV) and JavaScript Object Notation (JSON) files. For this project, precomputed output from FUNWAVE-TVD was used to populate a data table. The FUNWAVE-TVD output was received in a text file format that first had to be imported into a Microsoft Excel spreadsheet before being ingestible by UE's data tables. The data received included bathymetry and time series of wave frequencies (Epic Games, "Data Driven Gameplay," n.d.). Researchers are currently experimenting with how that data will be represented in the final implementation.

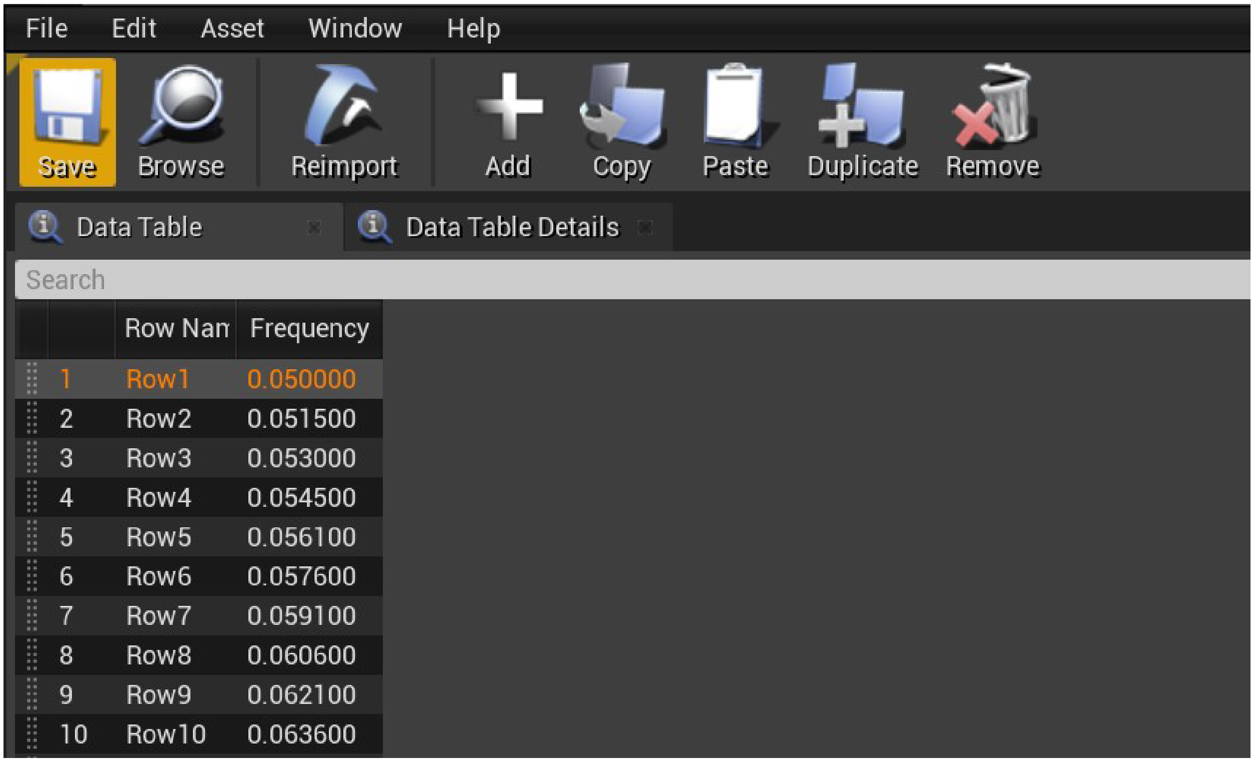

Figure 6. UE data table with FUNWAVE-TVD wave frequencies.

SUMMARY AND CONCLUSIONS: In step with Army and ERDC priorities, the Ship Simulator of the Future is under development. A major upgrade from the existing ship simulator is including accurate near-shore wave model physics. Unreal Engine is selected to promote cohesive VR development across ERDC labs and several plugins were evaluated for their use in VR ship simulation. Plugins can be used by UE content creators to speed up development such as VR Expansion Plugin and UE's water features. UE4GitPlugin offers source control which is vital for managing code projects. Other plugins like Cesium for Unreal were in too early a stage to be useful for the project, but could serve valuable in the future. Celeris wave modeling in Unity and FUNWAVE-TVD were considered for near-shore wave modeling, but FUNWAVE-TVD was selected due to more accuracy and flexibility. A limitation is lack of real-time output, therefore the wave model is calculated a priori and ingested into the Unreal Engine using data tables. Future work includes hardware upgrades to support realistic rendering.

Coupling the Ship Simulator of the Future with high-fidelity numerical modeling within a VR environment will provide a means to enhance ship survivability in operational deployments. This will be a generational leap forward in the simulation of ship motion by innovatively linking ship simulation software, and high-fidelity numerical wave models in a VR environment for operationally relevant scenarios. The research framework will both accelerate development and facilitate simulation, planning, and rehearsal of multi-domain operations by ensuring a seamless integration of sea- and land-based modeling and simulation tools to enable physics-based real-time accuracy and run-time efficiency.

FUTURE WORK: The project is still in early development and is scheduled to end in FY24. The team will continue work to improve upon the SSoF by coupling data ingested from FUNWAVE-TVD to provide realistic simulation based on wave physics. Further investigation into UE's data importation methods is necessary to fully understand the capabilities of incorporating valuable ERDC data into VR simulations. With continued advancements in physical realism and visualization, like those created by reality capture, the team envisions that the technologies employed in the SSoF could prove valuable for the Army and in designing ships and coastal structures.

The Rugged Edge Computing Node (RECON) mobile High Performance Computer (HPC) could possibly offer the computational ability required to calculate high-fidelity wave models. FUNWAVE-TVD currently must be run on the bigger Onyx HPC system at ERDC in order to calculate a hydrodynamic wave model simulation (Figure 7). Advancements in computational and graphical processing will enhance the realistic rendering of highly immersive simulations, and as technology continues to grow in this area, the RECON system may provide a mobile solution.

Figure 7. ERDC's Onyx HPC and RECON Mobile HPC.

ADDITIONAL INFORMATION: This technical note was prepared by Mr. Kelly Ervin, research computer scientist (kelly.b.ervin@erdc.dren.mil), Mr. Karl Smink, research computer scientist, Mr. Bryan Vu, research hydraulics engineer, and Mr. Jonathan Boone, research civil engineer, U.S. Army Engineer Research and Development Center, Information Technology Laboratory. This technical note should be cited as follows:

> Ervin, Kelly, Karl Smink, Bryan Vu, and Jonathan Boone. 2021. *Ship Simulator of the Future in Virtual Reality*. ERDC/ITL TN-22-1. Vicksburg, MS: U.S. Army Engineer Research and Development Center.

REFERENCES

Cesium GS. n.d. "Cesium for Unreal." Platform. Accessed May 2021. https://cesium.com/platform/cesium-for-unreal/.

Epic Games. n.d. "Data Driven Gameplay Elements." Making Interactive Experiences, Unreal Engine 4 Documentation. Accessed May 2021. https://docs.unrealengine.com/4.26/en-US/InteractiveExperiences/DataDriven/.

Epic Games. n.d. "Plugins." Setting Up Your Production Pipeline, Unreal Engine 4 Documentation. Accessed May 2021. https://docs.unrealengine.com/enUS/ProductionPipelines/Plugins/index.html.

Epic Games. n.d. "Unreal Engine." Accessed May 2021. https://www.unrealengine.com/en-US/unreal.

Epic Games. n.d. "Water." Building Virtual Worlds, Unreal Engine 4 Documentation. Accessed May 2021. https://docs.unrealengine.com/en-US/BuildingWorlds/Water/index.html.

Rombauts, Sébastien. n.d. "UE4GitPlugin." Accessed May. https://github.com/SRombauts/UE4GitPlugin.

Shi, Fengyan. n.d. "FUNWAVE-TVD." FUNWAVE-TVD Documentation. Accessed May fengyanshi.github.io/build/html/index.html.

Statzer, Joshua. n.d. "VR Expansion Plugin—A Virtual Reality Tool Kit." Accessed May https://vreue4.com.

Tavakkol, Sasan and Patrick Lynett. 2020. Celeris Base: An Interactive and Immersive Boussinesq-type Nearshore Wave Simulation Software. *Computer Physics Communications* 248, (March): 106966. https://doi.org/10.1016/j.cpc.2019.106966.

Unreal Engine. 2020. "Diving Into the Depths of Water as a Feature in Unreal Engine 4.26." YouTube Video, 35:12. December 4, 2020. https://www.youtube.com/watch?v=d5Ft7UaYayM&t=946s.