

Florida International University
Knight Foundation School of Computing and Information Sciences

Final Deliverable

Project Title: Advanced Computer Graphics

Team Members: Saul Abreu Almonte, Carlos Cruz, Laura Lou, Mario Cerra, Marco Pena

Product Owner: Carlos Cruz

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 *Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

This document provides an overview of our Advanced Computer Graphics project—an interactive, real-time simulation of a dynamic ocean, integrated with day/night transitions and realistic and dynamic weather effects. Motivated by the limitations of traditional wave models, we implemented a Fast Fourier Transform–based ocean renderer alongside a custom GLSL pipeline for clouds, and rain,, all driven by a user-adjustable ImGui interface. Developed in two-week sprints with a 6 sprint timeline, we used C++, OpenGL, GLFW, GLM, and Assimp, in order for our modular architecture to deliver high performance and flexibility. The document also outlines our testing strategy, system architecture, and includes UML diagrams and sprint retrospectives to illustrate our iterative, collaborative process.

Introduction.....	5
Current System.....	5
Purpose of New System.....	5
User Stories.....	7
Implemented User Stories.....	7
Pending User Stories.....	8
Project Plan.....	9
Hardware and Software Resources.....	9
Sprints Plan.....	10
Sprint 1: Planning and Set up.....	10
Sprint 2: System Design and Architecture.....	10
Sprint 3: Integration and Testing.....	10
Sprint 4: Integration and Testing.....	11
Sprint 5: Integration,Testing and Refinement.....	11
Sprint 6: Documentation and Finalization.....	11
System Design.....	13
Architectural Patterns.....	13
System and Subsystem Decomposition.....	14
Design Patterns.....	15
System Validation.....	16
Glossary.....	17
Appendix.....	19
Appendix A - Sprint Review Reports.....	19

Introduction

One of the major challenges in developing an ocean simulation is striking the right balance between visual realism and performance. Traditional methods—such as Gerstner Waves—can approximate simple wave motion but struggle to capture the full complexity of natural ocean states. Moreover, as these techniques grow more sophisticated, their computational cost often causes frame-rate instability in real-time rendering.

To address this, our project adopts a Fast Fourier Transform–based approach. By decomposing complex wave spectra into frequency components and then efficiently reconstructing spatial wave heights on the GPU, we achieve both high-fidelity visuals and consistent performance.

We also extend this realism beyond the water itself by integrating a dynamic day/night cycle and fully procedural atmospheric effects—volumetric clouds, and particle-based rain. Smooth transitions in lighting and weather not only enhance immersion but ensure that our ocean environment remains cohesive under any sky conditions.

Current System

Traditional ocean simulation methods struggle to balance realism and performance in graphics applications. Gerstner Waves often fail to capture realistic ocean states at high fidelity while maintaining stable frame rates. These methods are often computationally expensive, which leads to performance issues in real-time applications. Additionally, integrating atmospheric effects such as weather systems and day/night cycles with ocean simulations presents significant technical challenges that many current solutions don't address effectively.

Purpose of New System

Unlike other ocean simulation implementations that sacrifice either visual quality or performance, our system applies Fast Fourier Transform (FFT) techniques to efficiently compute wave dynamics, enabling complex and realistic ocean states while maintaining stable frame rates. Our ocean simulation provides mathematically accurate wave generation using FFT for realistic ocean movement. It also includes Physically Based Rendering for authentic water

appearance through proper light interactions. As well as, seamless integration with dynamic weather events and day/night cycles. Our simulation also has an intuitive user interface that allows real-time parameter adjustments and optimized performance that can run even on a laptop.

USER STORIES

The following section contains user stories that were implemented during this iteration of the "Advanced Computer Graphics" project. These user stories served as the basis for the implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

User Story 1: As a developer, I want to review and practice C++ concepts so that I can apply them in creating the project.

User Story 2: As a developer, I want to brainstorm on a project idea that is feasible to create for this Capstone 2 project.

User Story 3: As a developer, I want to install OpenGL and create a repository on Github so that I have an environment to work on the project.

User Story 4: As a developer, I want to complete the "Getting Started" chapter for OpenGL so that I have a foundational understanding of graphics programming.

User Story 5: As a developer, I want to create the basic ocean main function() to initialize the ocean simulation, so I can build a better foundation to build and test the program.

User Story 6: As a developer, I want to create dynamic weather events such as clouds and rain, so the environment feels more engaging and alive.

User Story 7: As a developer, I want to add a day and night cycle in order to cast dynamic lighting to the environment.

User Story 8: As a developer, I want to research what features can be added to the project.

User Story 9: As a developer, I want to have an ocean render prototype ready to test our current progress.

User Story 10: As a developer, I want to research how to implement Fourier Transform waves in order to create complex and dynamic ocean behavior.

User Story 11: As a developer, I want to add Tessellation shaders to improve the performance of the ocean and improve the look with real-time vertex subdivision.

User Story 12: As a developer, I want to optimize and revise day and night cycles in order to cast realistic dynamic lighting.

User Story 13: As a developer, I want to have a prototype ocean movement with Fourier Transform Waves and physically based rendering.

User Story 14: As a developer, I want to keep revising the dynamic weather events such as clouds and rain, by going over the current prototype.

User Story 15: As a developer, I want to debug and fully implement the day and night cycle so that the current prototype reflects realistic sun transitions without errors.

User Story 16: As a developer, I want to add an interface so users can interact real-time with the simulation.

User Story 17: As a developer, I want to add buoyancy so objects interact with the ocean simulation.

User Story 18: As a developer, I want to finish setting up an interface so users can interact real-time with the simulation.

User Story 19: As a developer, I want to add all the features together into a final version.

Pending User Stories

None.

PROJECT PLAN

This section describes the planning that went into the realization of this project. This project incorporated agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

- OpenGL
- C++
- Github
- Visual Studios
- ImGUI
- High end GPU NVIDIA RTX 3060
- 6-core GPU
- RAM 16GB DDR4
- VRAM 6-8 GB

Sprints Plan

Sprint 1: Planning and Set up

- **Objective:** Plan and set up a project environment.
- **Tasks:**
 - Review C++ concepts; brainstorm project idea
 - Install OpenGL, GLFW, GLAD, GLM, Assimp
 - Create GitHub repo & CMake build
 - Complete “Getting Started” tutorials on OpenGL
- **Deliverables:**
 - Project repository with initial setup

Sprint 2: System Design and Architecture

- **Objective:** Create system design and architecture diagrams.
- **Tasks:**
 - Research FFT-based waves & weather features
 - Define modular architecture
 - Draw UML & package diagrams
 - Design shader & resource-loader interface
- **Deliverables:**
 - Architecture docs (UML, package diagrams)
 - API interfaces & module specs

Sprint 3: Integration and Testing

- **Objective:** Begin Integrating features.
- **Tasks:**
 - Implement basic main ocean class
 - Integrate FFT spectrum init/evolution
 - Build day/night interpolation (lighting, sky gradients, starfield)
 - Begin development of volumetric clouds and rain

- **Deliverables:**
 - Smooth rain and day/night cycle demo
 - Prototype ocean simulation

Sprint 4: Integration and Testing

- **Objective:** Continue integrating features and refining.
- **Tasks:**
 - Add tessellation shaders for LOD
 - Render animated ocean mesh
 - Debug lighting transitions
- **Deliverables:**
 - Prototype ocean simulation with FFT & tessellation
 - Mesh displacement & normal maps
 - Volumetric clouds, & rain.

Sprint 5: Integration, Testing, and Refinement

- **Objective:** Finalize the ocean simulation project and prepare for a demo.
- **Tasks:**
 - Integrate ImGui panel for live tweaking
 - Implement buoyancy for objects on waves
 - Merge ocean, day/night, weather, and UI modules
- **Deliverables:**
 - Fully integrated system with UI & buoyancy

Sprint 6: Documentation and Finalization

- **Objective:** Finalize the project and prepare documentation.
- **Tasks:**
 - Write system documentation
 - Draft test cases and results

- **Deliverables:**
 - Complete project documentation
 - Final executables

System Design

This section explains the key design decisions and architectural patterns that shaped our real-time simulation engine. We begin with an overview of our modular architecture, then decompose the system into its major subsystems, present a deployment view, and finally describe the design patterns we employed.

Architectural Patterns

The system employs a modular architecture to separate concerns and enhance maintainability. The architecture includes:

- **Modular Engine Architecture**

We separate concerns into distinct modules—Asset Management, Simulation, Rendering, UI, and Input/Camera—to simplify development, testing, and future extensions.

- **Rendering Pipeline**

1. **Simulation Pass** (compute shaders for FFT ocean + weather fields)
2. **Geometry Pass** (tessellation and mesh generation)
3. **Lighting & Skybox Pass** (day/night, starfield, sky gradients)
4. **Post-Processing/UI Pass** (ImGui overlay, tone mapping)

- **Resource Manager**

A single ResourceManager handles all shader, texture, and mesh loading—caching assets to avoid redundant I/O and ensure consistent access throughout the engine.

System and Subsystem Decomposition

The system can be decomposed into the following subsystems:

Asset Management:

- fileFinder: Resolves file paths at runtime
- ResourceManager: Loads and caches shaders, cubemaps, and models (Assimp)

Simulation:

- OceanSimulator: FFT-based wave spectrum evolution and inverse FFT to generate displacement/normal maps
- DayNightCycle: Interpolates time-of-day, sky gradients, and sun/moon positions
- WeatherSystem: Generates volumetric clouds (ray-marching/noise textures), and rain particles.

Rendering:

- ShaderManager: Compiles and links GLSL programs
- MainRenderer: Orchestrates the render passes in the correct order each frame

UI & Input:

- ImGui Interface: Exposes real-time controls for wave amplitude, cloud density, lighting, etc.
- CameraController: Processes keyboard/mouse input to update a free-fly camera

Design Patterns

The following design patterns are used to support modularity, performance, and real time interaction in the ocean simulation:

- **Model-View-Controller (MVC)**
 - Model: Handles the core simulation data, including the Fast Fourier Transform (FFT) wave calculations, ocean state parameters, and weather condition variables
 - View: Renders the ocean surface, atmospheric effects, and UI components through OpenGL shaders and Dear ImGui interface
 - Controller: Manages user inputs and interactions with the simulation parameters, orchestrating changes between the model and view
- **Observer Pattern**
 - Implemented to ensure real-time updates between interrelated components:
 - Weather system notifies the ocean simulation when conditions change.
 - Day/night cycle broadcasts lighting updates to all rendering components.
 - User interface parameter adjustments trigger immediate visual feedback.
- **Factory Method**
 - Used to dynamically create and manage various environmental elements:
 - Weather effect generators (clouds and rain) based on selected conditions.
 - Ocean surface components with different parameter configurations.
 - Shader programs tailored to different rendering requirements.

System Validation

System validation for our ocean simulation project was conducted through a structured, multi-stage process to ensure both functionality and performance stability:

- **Unit Testing:**
Individual modules such as the Fast Fourier Transform (FFT) implementation, day/night cycle, and atmospheric weather effects were tested independently to verify correctness and rendering fidelity.
- **Integration Testing:**
Components were integrated progressively—starting with ocean simulation and extending to lighting and weather systems. Testing focused on ensuring smooth transitions, proper shader interaction, and coherent behavior across modules (e.g., how lighting reflects off dynamically generated waves).
- **Performance Testing:**
The simulation was evaluated for frame rate stability under different environmental conditions (e.g., rain, high wave activity). Optimizations were made to maintain real-time responsiveness on mid-range GPUs.
- **User Acceptance Testing (UAT):**
The interface built using ImGui allowed testers to adjust simulation parameters live. Feedback was gathered from users evaluating realism, immersion, and system responsiveness, especially during transitions like sunrise or storm onset.
- **Bug Tracking and Resolution:**
All identified issues were logged and tracked using GitHub. Common bugs included GPU texture artifacts, light misalignment during time-of-day transitions, and FFT instability under edge case inputs. Fixes were prioritized based on impact on visual fidelity and performance.

GLOSSARY

- **Fast Fourier Transform (FFT):** A mathematical algorithm used to convert signals between time/space and frequency domains. In this ocean simulation, FFT is used to efficiently calculate realistic wave patterns and movement across the ocean surface.
- **Physically Based Rendering (PBR):** A rendering approach that models light in a way that simulates real-world physics. In this project, PBR techniques enable realistic water appearance through accurate simulation of light interactions, including reflections, refractions, and subsurface scattering.
- **OpenGL:** An open-source graphics rendering API used to create 2D and 3D applications. This project uses OpenGL to implement the ocean simulation and rendering pipeline.
- **Tessellation:** A technique in computer graphics that subdivides geometry into smaller pieces at render time. In this ocean simulation, tessellation shaders improve performance and visual quality by dynamically adjusting the detail level of the ocean surface.
- **Dear ImGui (ImGui):** A lightweight graphical user interface library designed for applications that need simple, immediate-mode GUI elements. In this project, ImGui provides the interface for adjusting simulation parameters in real-time.
- **Vertex Shader:** A programmable shader stage in the rendering pipeline that processes individual vertices. In this ocean simulation, vertex shaders handle the positioning and displacement of ocean surface points.
- **Fragment Shader:** A programmable shader stage that processes individual fragments (potential pixels) before they appear on screen. In this project, fragment shaders calculate the final color of water based on lighting, reflections, and other visual properties.
- **Compute Shader:** A shader stage that performs general-purpose computations on the GPU. In this ocean simulation, compute shaders are used for efficient FFT calculations and wave processing.
- **Normal Map:** A texture that stores surface normal directions rather than colors, used to add detail to surfaces without increasing geometry complexity. In this project, normal maps create the appearance of small-scale wave details on the ocean surface.
- **Buoyancy:** The upward force exerted by a fluid on an immersed object. In this simulation, buoyancy is implemented to create realistic interactions between objects and the ocean surface.

- **Day/Night Cycle:** A simulation of the passage of time from day to night, including changes in lighting, colors, and shadows. In this project, the day/night cycle affects how the ocean appears under different lighting conditions.
- **Render Pipeline:** The sequence of stages through which scene data passes to create the final rendered image. The ocean simulation's render pipeline includes specialized stages for water rendering, atmospheric effects, and post-processing.

APPENDIX

Appendix A - Sprint Review Reports

Sprint 1 Review Meeting Minutes

Attendees: Saul Abreu Almonte, Carlos Cruz, Laura Lou, Mario Cerra, and Marco Pena

Start time: 1/24/25 6:30 PM

End time: 1/24/25 7:30 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story 1: As a developer, I want to review and practice C++ concepts so that I can apply them in creating the project.
 - Hours: 2
- User Story 2: As a developer, I want to brainstorm on a project idea that is feasible to create for this Capstone 2 project.
 - Hours: 2
- User Story 3: As a developer, I want to install OpenGL and create a repository on Github so that I have an environment to work on the project.
 - Hours: 1
- User Story 4: As a developer, I want to complete the “Getting Started” chapter for OpenGL so that I have a foundational understanding of graphics programming.
 - Hours: 3

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting:

- User Story 5: As a developer, I want to create the basic ocean main function() to initialize the ocean simulation, so I can build a better foundation to build and test the program.
 - Hours: 2

Sprint 2 Review Meeting Minutes

Attendees: Saul Abreu Almonte, Carlos Cruz, Laura Lou, Mario Cerra, and Marco Pena

Start time: 2/07/25 4:00 PM

End time: 2/07/25 5:15 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story 1: As a developer, I want to create the basic ocean main function() to initialize the ocean simulation, so I can build a better foundation to build and test the program.
 - Hours: 3
- User Story 5: As a developer, I want to research what features can be added to the project
 - Hours:3

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting:

- User Story 2: As a developer, I want to create dynamic weather events such as clouds and rain, so the environment feels more engaging and alive.
 - Hours: 3
- User Story 3: As a developer, I want to add a day and night cycle in order to cast dynamic lighting to the environment.
 - Hours: 3
- User Story 4: As a developer, I want to have an ocean render prototype
 - Hours:3

Sprint 3 Review Meeting Minutes

Attendees: Saul Abreu Almonte, Carlos Cruz, Laura Lou, Mario Cerra, and Marco Pena

Start time: 2/21/25 7:05 PM

End time: 2/21/25 8:05 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story 3: As a developer, I want to have an ocean render prototype ready to test our current progress.
 - Hours: 3

- User Story 5: As a developer, I want to add Tessellation shaders to improve the performance of the ocean and improve the look with real-time vertex subdivision
Hours: 3

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting:

- User Story 1: As a developer, I want to create dynamic weather events such as clouds and rain, so the environment feels more engaging and alive.
Hours: 3
- User Story 2: As a developer, I want to add a day and night cycle in order to cast dynamic lighting to the environment.
Hours: 3
- User Story 4: As a developer, I want to research how to implement Fourier Transform waves in order to create complex and dynamic ocean behavior.
Hours: 3

Sprint 4 Review Meeting Minutes

Attendees: Saul Abreu Almonte, Carlos Cruz, Laura Lou, Mario Cerra, and Marco Pena

Start time: 3/16/25 5:00 PM

End time: 3/16/25 6:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story 1: As a developer, I want to create dynamic weather events such as clouds and rain, so the environment feels more engaging and alive.
 - Hours: 3
- User Story 2: As a developer, I want to add a day and night cycle in order to cast dynamic lighting to the environment.
 - Hours: 3
- User Story 4: As a developer, I want to implement Fourier Transform waves in order to create complex and dynamic ocean behavior.
 - Hours: 3
- User Story 5: As a developer, I want to have a prototype ocean movement with Fourier Transform Waves and physically based rendering
 - Hours: 3

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting:

- User Story 3: As a developer, I want to have an ocean render prototype ready to test our current progress.
 - Hours: 3

Sprint 5 Review Meeting Minutes

Attendees: Saul Abreu Almonte, Carlos Cruz, Laura Lou, Mario Cerra, and Marco Pena

Start time: 3/28/25 5:00 PM

End time: 3/28/25 6:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story 3: As a developer, I want to have an ocean render prototype ready to test our current progress.
 - Hours: 3

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting:

- User Story 1: As a developer, I want to keep revising the dynamic weather events such as clouds and rain, by going over the current prototype.
 - Hours: 3
- User Story 2: As a developer, I want to debug and fully implement the day and night cycle so that the current prototype reflects realistic sun transitions without errors.
 - Hours: 3
- User Story 4: As a developer, I want to add an interface so users can interact real-time with the simulation.
 - Hours: 3
- User Story 5: As a developer, I want to add buoyancy so objects interact with the ocean simulation
 - Hours: 3

Sprint 6 Review Meeting Minutes

Attendees: Saul Abreu Almonte, Carlos Cruz, Laura Lou, Mario Cerra, and Marco Pena

Start time: 4/12/25 7:00 PM

End time: 4/12/25 8:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story 1: As a developer, I want to keep revising the dynamic weather events such as clouds and rain, by going over the current prototype.
 - Hours: 3
- User Story 2: As a developer, I want to have the day and night cycle completed so that this feature is completed for the project.
 - Hours: 3
- User Story 3: As a developer, I want to finish setting up an interface so users can interact real-time with the simulation.
 - Hours: 3
- User Story 4: As a developer, I want to add buoyancy so objects interact with the ocean simulation
 - Hours: 3
- User Story 5: As a developer, I want to add all the features together into a final version.
 - Hours: 3

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting:

- N/A