*Florida International University*

*School of Computing and Information Sciences*

Software Engineering Focus

Final Documentation

TAM: Multi-Touch, Mid-Air, and Motion for Virtual and Augmented Reality – ICAVE – Learning with Virtual Reality

**Team Members:** Michael Quiros, Noel Gonzalez

**Product Owner(s)**: Francisco Ortega

**Mentor(s)**: Francisco Ortega

**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 presents the information necessary to gain a good understanding of the FIU ICAVE, along with an understanding of the process and functionalities that went into the making of the software developed for “I CAVE – Learning with Virtual Reality” through an agile development cycle.*

**Table of Contents**

**Introduction**

Current System ……….…………………………………………………………………………………………………………. 5

Purpose of New System …………………………………………………..………………………………………………….. 5

**User Stories**

Implemented User Stories ………………………………………….………………………………………………………. 6

Pending User Stories …………………………………………………………………...………………………………..….... 13

**Project Plan**

Hardware and Software Resources ……………………………………………………...……………………….…… 14

Sprints Plan ……………………………………………………………..……………………………………………………….. 15

*Sprint 1*  …………………………………………………………………………………………………………………………. 15

*Sprint 2*  …………………………………………………………………………………………………………………………. 16

*Sprint 3*  …………………………………………………………………………………………………………………………. 17

*Sprint 4*  …………………………………………………………………………………………………………………………. 19

*Sprint 5*  …………………………………………………………………………………………………………………………. 20

*Sprint 6*  …………………………………………………………………………………………………………………………. 21

*Sprint 7*  …………………………………………………………………………………………………………………………. 23

**System Design**

Architectural Patterns ……………………………………………………………………………………………………. 24

System and Subsystem Decomposition …………………………………………………………………………….…... 24

Deployment Diagram …………………………………………………………………………………………………….…... 25

Design Patterns ………………………………………………………………………………………………………….…..... 26

**System Validation**  ………………………………………………………………………………………………………………..27

**Glossary**  ………………………………………………………………………………………………………………………………40

**Appendix**

Appendix A - UML Diagrams ……………………………………………………………………………………………….. 42

Appendix B - User Interface Design …………………………………………………………………………….…....... 44

Appendix C - Sprint Review Reports …………………………………………………………………………...……… 46

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist… 49

**References** …………………………………………………………………………………………………………………………... 53

# Introduction

As technology progresses, mankind continues to adapt to our ever expanding repertoire of technology. When the television came along, there was much criticizing as to the impact that it would cause and the type of person raised with television would become. The same could be said with the large wave of popularity of video games and more recently, smart phones. Through this skepticism, all of these new and impactful technologies were able to be harnessed and have been applied far past their original design functions. This project aims to follow the same ideology. *ICAVE - Learning with Virtual Reality* attempts to take the newly growing field of virtual reality and apply it with an educational, yet fun and interactive twist. The ICAVE is a small hexagonal area monitored by motion sensors. Each wall of the hexagon (bar the entrance) is equipped with a screen and projector to allow the user to physically enter into this simulation and interact with objects through use of motion tracked 3D glasses and a motion tracked controller.

## Current System

Currently there is no previous build of this project being developed in FIU. *ICAVE - Learning with Virtual Reality* is the first iteration of this sailing simulation. There are a few other Virtual Reality sailing simulators that have been developed in recent times, but all seem to have taken advantage of the rising popularity of virtual reality headsets such as the HTC Vive or the smart phone enabled Samsung VR Headset. This project instead aims to take advantage of the advanced hardware that the ICAVE provides users to provide a more physical and hands on approach.

## Purpose of New System

Users will be able to commandeer a ship to learn the basics of sailing and get a feel for controlling a ship before ever being near an ocean. These users will be able to either traverse small premade obstacles to collect pick up objects throughout the game or sail the (mostly) open water to get a feel for the movement of the ship on the ocean.

This sailing simulator does not only take sailing into account. The underlying everyday and ever important skills of navigation and wayfinding are put into effect to assist the user in an easy to learn experience. A small sandbox area, large overarching structures and easily identifiable pick up objects will ensure that the user will always know the way to the next pick up object and provides a way back to the main obstacles in case they decide to sail the open water.

# User Stories

The following section provides the detailed user stories that were implemented in this iteration of the ICAVE – Learning with Virtual Reality 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 #1165 - Access to Unity**

**Description:**

As a developer, I must gain access to the correct version of Unity and import the required Plug-in (GetReal3d) to develop software for the ICAVE.

**Acceptance Criteria:**

* Download Unity Version 5.3.2
* Download GetReal3D Plug-in
* Create a Project in Unity
* Import the GetReal3D Plug-in to this Project

**User Story #1167 - Agile/Scrum Development Familiarity**

**Description:**

As a user, I must research Agile/Scrum Development and become familiar with Mingle.

**Acceptance Criteria:**

* Have the available credentials to access Mingle
* View the videos and instructions for Agile/Scrum given by Product Owner

**User Story #1166 - Familiarity with ICAVE**

**Description:**

As a developer, I must gain familiarity with the ICAVE and the associated software.

**Acceptance Criteria:**

* Schedule training session with ICAVE development team.
* Have Unity and the GetReal3D Plug-in readily available.
* Gain access to ICAVE to study software and instructions to start implementing our project.
* Have a familiarity with the project that will be implemented in the ICAVE

**User Story #1287 – ICAVE Familiarity**

**Description:**

I must understand how the ICAVE works in order to design and create a project suitable for it.

**Acceptance Criteria:**

* Must go through the ICAVE training session
* Must know how the ICAVE changes Unity development
* Must have adept Unity skills.

**User Story #1187 - Develop a Small Demo**

**Description:** As a developer I should create a small demo using Unity and GetReal3D to become more familiar with the ICAVE.

**Acceptance Criteria:**

* Must have a good degree of proficiency in Unity.
* Must create an environment which the user can interact with.
* Must have basic features such as UI, controls, & colliders

**User Story #1288 – ICAVE Demo**

**Description:**

Must create a demo project and run it in the ICAVE

**Acceptance Criteria:**

* Project must successfully run in the ICAVE.
* Project must utilize ICAVE functions (controls, camera).
* Project must prepare the team for ICAVE development.

**User Story #1289 – ICAVE Project Design**

**Description:**

This is where the project to be worked on is decided and conceived.

**Acceptance Criteria:**

* Project must adhere to standards set for TAM.
* Project must showcase ICAVE features.
* Project must be doable in the time available in the semester.

**User Story #1296 – Setup Environment**

**Description:**

Setup the basic environment where the player is going to navigate through. This is just the ship itself, the water, lighting effects, and the player.

**Acceptance Criteria:**

* Must function correctly in the ICAVE.
* Must work later when the more dynamic environment is introduced.

**User Story #1168 - Research 3D Object Interaction**

**Description:**

As a developer, I need to research the interactions between 3D objects in Unity and how separate interactions would differ or stay the same when using the ICAVE.

**Acceptance Criteria:**

* Be able to create 3D objects in Unity.
* Be able to interact with the simulated 3D objects.
* Import the interactions to the ICAVE software to test.

**User Story #1297 – Ship Movement**

**Description:**

The ship needs to move in a realistic fashion. The ship controls should not be handled with a simple “WASD” but with a position based rudder to simulate real life ships. The wind speed and direction should also impact the ship’s speed.

**Acceptance Criteria:**

* The ship should move according to wind speed and direction, and the position of the helm.
* The ship should not be too difficult to control.

**User Story #1283 - Animation Creation**

**Description:**

As a developer, I must create and modify animations for different parts of our project.

**Acceptance Criteria:**

* Be familiar with Unity scripts and animators.
* Have knowledge of finite state machines.
* Have a successful animation applied through use of animator.

**User Story #1298 – Ship Enhancements**

**Description:**

The ship should simulate real life floating effects.

**Acceptance Criteria:**

* Floating effect should not disorient the player.
* Floating effect must be somewhat realistic.

**User Story #1284 - Model Manipulation**

**Description:**

As a developer, I must become familiar with software to manipulate the 3D models we will be using as assets for further customization of our goals.

**Acceptance Criteria:**

* Must become familiar with 3D modeling software (In this case Maya).
* Must be able to attach and detach different polygons/objects to the model.

**User Story #1299 – Ship Enhancements 2**

**Description:**

Ship colliders must be added to restrict player movement. The ship should also have some UI elements showing important information such as ship direction.

**Acceptance Criteria:**

* Player should not fall off the ship.
* Player should be able to move freely throughout the ship.
* Player should not get stuck anywhere.
* There must be a basic UI showing the player important information about the wind and the ship.

**User Story #1300 – Ship Enhancements 3**

**Description:**

To better simulate real life ships, the ship should be able to rotate on the Z axis without acting strange.

**Acceptance Criteria:**

* Ship should not appear to be sinking while simulating.

**User Story #1285 - Project Environment Interaction**

**Description:**

As a developer, I must create an interactive environment for our user to be able to interact with.

**Acceptance Criteria:**

* Must have access to assets related to project.
* Must have a familiarity with static objects in Unity.
* Must have a familiarity with colliders in Unity.

**User Story #1286 - Objective Creation**

**Description:**

As a developer, I must create and implement an objective for the user while using the ICAVE.

**Acceptance Criteria:**

* Create randomly generated pick up objects throughout the map.
* Items must be dynamically generated as user picks them up.
* Items must be within the legal bounds of the program and able to be picked up by the user.

**User Story #1302 – ICAVE Control Remapping**

**Description:**

The controls need to be mapped to use the ICAVE.

**Acceptance Criteria:**

* All controls work correctly in the ICAVE.

**User Story #1303 – Collision Detection**

**Description:**

With other objects now present in the game, the game must handle collisions in a semi realistic fashion – meaning the ship should not do anything unrealistic (such as flying or going through objects).

**Acceptance Criteria:**

* The ship must collide and not move further after colliding (including going through other objects).

## Pending User Stories

**User Story #1301 – Ship Controls**

**Description:**

Player should be able to toggle entering the wheel or moving freely.

**Acceptance Criteria:**

* Player should not be able to move while handling the wheel.
* Player should not be able to control the ship while moving freely.
* Player should be able to switch modes at any time by interacting with the helm.

# Project Plan

This section describes the planning that went into the realization of this project. This project incorporated the 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:

**FIU ICAVE –** Contains 4 motion capture cameras, a Logitech F710 Controller with motion capture, 3D glasses with motion capture, a Head node workstation PC, a Render Node workstation PC, and 5 high resolution screens with Projectors

**Unity 5.3.2 –** 3D Editor for developing software in this virtual reality setting. Unity version 5.3.2 must be used for compatibility with the GetReal3D extension.

**GetReal3D –** Virtual Reality plug-in for Unity. Allows easy to use VR tracking and controller data.

**Maya 2017 –** 3D modeling software. Allows editing and customization of 3D model objects used as assets in the Unity projects.

**Visual Studio 2015 –** Microsoft designed IDE. Used in conjunction with C# to write scripts for this particular project.

**MonoDevelop 2017** - IDE native to Unity used to write C# or JS scripts. Used to write C# scripts for this project.

## Sprints Plan

### Sprint 1

**Sprint Planning Meeting Minutes:**

Attendees: Michael Quiros, Noel Gonzalez, Francisco Ortega, Other TAM members

Date: 1/13/2017

Start time: 5:00PM

End time: 6:00PM

After discussion, the velocity of the team was estimated to be:

10 hours per week for VIP Junior students.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

**#1165 Access to Unity**

As a developer, I must gain access to the correct version of Unity and import the required Plug-in (GetReal3d) to develop software for the ICAVE.

**#1167 Agile/Scrum Development Familiarity**

As a user, I must research Agile/Scrum Development and become familiar with Mingle.

The team members indicated their willingness to work on the following user stories:

**#1287 ICAVE Familiarity**

I must understand how the ICAVE works in order to design and create a project suitable for it.

### Sprint 2

**Sprint Planning Meeting Minutes:**

Attendees: Michael Quiros, Noel Gonzalez, Francisco Ortega, Other TAM members

Date: 1/27/2017

Start time: 5:00PM

End time: 6:00PM

After discussion, the velocity of the team was estimated to be:

10 hours per week for VIP Junior students.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

**#1166 Familiarity with ICAVE**

As a developer, I must gain familiarity with the ICAVE and the associated software.

The team members indicated their willingness to work on the following user stories:

**#1289 ICAVE Project Design**

This is where the project to be worked on is decided and conceived.

**#1288 ICAVE Demo**

Must create a demo project and run it in the ICAVE

### Sprint 3

**Sprint Planning Meeting Minutes:**

Attendees: Michael Quiros, Noel Gonzalez, Francisco Ortega, Other TAM members

Date: 1/27/2017

Start time: 5:00PM

End time: 5:30PM

After discussion, the velocity of the team was estimated to be:

10 hours per week for VIP Junior students.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

*None Specified.*

The team members indicated their willingness to work on the following user stories:

**#1168 Research 3D Object Interaction**

As a developer, I need to research the interactions between 3D objects in Unity and how separate interactions would differ or stay the same when using the ICAVE.

**#1187 Develop a small demo**

As a developer I should create a small demo using Unity and GetReal3D to become more familiar with the ICAVE.

**#1289 Setup Environment**

Setup the basic environment where the player is going to navigate through. This is just the ship itself, the water, lighting effects, and the player.

**#1297 Ship Movement**

The ship needs to move in a realistic fashion. The ship controls should not be handled with a simple “WASD” but with a position based rudder to simulate real life ships. The wind speed and direction should also impact the ship’s speed.

### Sprint 4

**Sprint Planning Meeting Minutes:**

Attendees: Michael Quiros, Noel Gonzalez

Date: 3/8/2017

Start time: 11:00AM

End time: 11:30AM

After discussion, the velocity of the team was estimated to be:

10 hours per week for VIP Junior students.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

*None Specified*

The team members indicated their willingness to work on the following user stories:

**#1289 Ship Enhancements**

The ship should simulate real life floating effects.

**#1297 Ship Enhancements 2**

Ship colliders must be added to restrict player movement. The ship should also have some UI elements showing important information such as ship direction.

**#1283 Animation Creation**

As a developer, I must create and modify animations for different parts of our project.

**#1187 3D Model Manipulation**

As a developer, I must become familiar with software to manipulate the 3D models we will be using as assets for further customization of our goals.

### Sprint 5

**Sprint Planning Meeting Minutes:**

Attendees: Michael Quiros, Noel Gonzalez

Date: 3/10/2017

Start time: 6:45PM

End time: 7:00PM

After discussion, the velocity of the team was estimated to be:

10 hours per week for VIP Junior students.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

*None Specified.*

The team members indicated their willingness to work on the following user stories:

**#1300 Ship Enhancements 3**

To better simulate real life ships, the ship should be able to rotate on the Z axis without acting strange.

Other fine tuning of previous user stories such as:

**#1283 Animation Creation**

As a developer, I must create and modify animations for different parts of our project.

**#1284 Model Manipulation**

As a developer, I must become familiar with software to manipulate the 3D models we will be using as assets for further customization of our goals.

### Sprint 6

**Sprint Planning Meeting Minutes:**

Attendees: Michael Quiros, Noel Gonzalez, Francisco Ortega

Date: 3/24/2017

Start time: 3:45PM

End time: 4:00PM

After discussion, the velocity of the team was estimated to be:

10 hours per week for VIP Junior students.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

**#1287 Project Environment Interaction**

As a developer, I must create an interactive environment for our user to be able to interact with.

**#1286 Objective Creation**

As a developer, I must create and implement an objective for the user while using the ICAVE.

The team members indicated their willingness to work on the following user stories:

**#1301 Ship Controls**

To better simulate real life ships, the ship should be able to rotate on the Z axis without acting strange.

**#1302 ICAVE controls**

As a developer, I must create and modify animations for different parts of our project.

**#1284 Collision Detection**

As a developer, I must become familiar with software to manipulate the 3D models we will be using as assets for further customization of our goals.

### Sprint 7

**Sprint Planning Meeting Minutes:**

Attendees: Michael Quiros, Noel Gonzalez

Date: 4/8/2017

Start time: 4:00PM

End time: 4:15PM

After discussion, the velocity of the team was estimated to be:

10 hours per week for VIP Junior students.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

*None Specified.*

The team members indicated their willingness to continue working on the following user stories:

**#1287 Project Environment Interaction**

As a developer, I must create an interactive environment for our user to be able to interact with.

**#1301 Ship Controls**

To better simulate real life ships, the ship should be able to rotate on the Z axis without acting strange.

**#1302 ICAVE controls**

As a developer, I must create and modify animations for different parts of our project.

**#1284 Collision Detection**

As a developer, I must become familiar with software to manipulate the 3D models we will be using as assets for further customization of our goals.

# System Design

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

## Architectural Patterns

Our architectural pattern is a mix of a service oriented architecture with Unity’s component architecture. We follow the Unity standard where everything on the scene is a GameObject first and foremost. Scripts are always attached to GameObjects as “Behaviors”. Finally, specialized object do not form subclasses from GameObject but rather apply their effects via the behaviors mentioned above. Scripts are almost always subclasses of the MonoBehaviour class instead, with some brand new classes occasionally made to tackle very specific functions.

Much like service oriented architecture, however, we have “managers” that individually do services and interact with one another. Our project is still in its infancy stages complexity wise therefore one should take these architectural patterns with a grain of salt as they can easily be changed by future developers. However, this style of architecture is considered standard in the Unity community and should probably be kept the way it is.

## System and Subsystem Decomposition

Following Unity’s component architecture we add all our scripts as components to GameObjects. For example, the PlayerShip script is added to the PlayerShipController GameObject. All the scripts work this way. Some GameObjects are “managers”, but they are also invisible GameObjects in the scene and still use Unity components. The ActiveEnvironment script, for example, attaches itself to the ActiveEnviroment GameObject and manages the changing environment.

The PlayerShip controls everything the player’s ship does.

The ActiveEnvironment modifies how PlayerShip behaves by changing values It also services PlayerShip by calculating things such as the real wind strength, which is a final value used in determining propulsion.

The UIManager gathers data from other objects and displays meaningful information for the player.

Few other scripts such as Rotator are used to have specific GameObjects behave in certain ways independently without the use of more multi object manager scripts.

## Deployment Diagram

OceanView refers to to the Unity Application containing files created by us. OceanView requires GetReal3D 3.2.5 or later in order to run properly. It can be run on Windows, Mac, or Linux even without the ICave but there might be issues with the controls.



## Design Patterns

**Component**

Our application uses the traditional Unity components. Components determine how the one specific object that has the component attached should behave. Generally, these components should not have references to other objects and provide other objects with behavior, except to pass information between them.

**Manager**

Managers are the exception to the component rule and are used to control aspects of the game that do not belong to any one object in particular. Currently used in the Environment and the UI and should be used for Save/Load and Game State in the future.

# System Validation

**User Story #1187 - Develop a Small Demo**

**Purpose:**

* To determine compatibility between Unity projects and the ICAVE and to become more familiar with the ICAVE.

**Precondition:**

* Must have some degree of proficiency with Unity.
* Must have GetReal3D imported into Unity Project.
* Must use GetReal3D prefabricated character controller and camera.
* Must create a demo to test in the ICAVE.

**Test Procedure:**

1. Start the executable of the demo in the ICAVE
2. Move the character around and observe the results of the controls.

**Expected Result:**

1. The character will be movable using the ICAVE controller.
2. The camera will show the extended field of view used in the ICAVE.
3. All features will work correctly in correspondence with a standard Unity Project.

**Actual Result:**

1. The characted was movable using the ICAVE controller.
2. The camera showed across all 5 displays in the ICAVE.
3. The head motion sensor manipulated the camera position correctly.
4. All other features worked as they would in a non-ICAVE setting.

**User Story #1296 – Setup Environment**

**Purpose:**

* To create a shell of the environment that will be used for the duration of this project.

**Precondition:**

* Must have GetReal3D imported into Unity Project.
* Must use GetReal3D prefabricated character controller and camera.
* Must have a small environment created including a ship, water, lighting and the player.

**Test Procedure:**

1. Start the executable of the project in the ICAVE
2. Verify that all aspects of the project behave normally in the ICAVE

**Expected Result:**

1. The ship and player will float on the water.
2. The player camera will follow correctly based on head and player movement,

**Actual Result:**

1. The ship and player floated on the water.
2. The player camera followed correctly using head motion and controller movement.

**User Story #1297 – Ship Movement**

**Purpose:**

* To allow the player to traverse the environment and create the beginning of the ship motion controls.

**Precondition:**

* Must have GetReal3D imported into Unity Project.
* Must use GetReal3D prefabricated character controller and camera.
* Must have a small environment created including a ship, water, lighting and the player.

**Test Procedure:**

1. Start the executable of the project
2. Verify that all aspects of the project behave normally in the ICAVE

**Expected Result:**

1. The ship and player will float on the water.
2. The ship will move in different directions based on wind speed, wind direction, and helm position.

**Actual Result:**

1. The ship and player floated on the water.
2. The ship moved correctly based on wind speed but had a small rotational bug based on the helm position.

**User Story #1283 - Animation Creation**

**Purpose:**

* To allow the player to control the wheel of the boat based on controls and helm position.
* To allow the player to control the cannons of the boat based on controls.

**Precondition:**

* Must have GetReal3D imported into Unity Project.
* Must use GetReal3D prefabricated character controller and camera.
* Must have a small environment created including a ship, water, lighting and the player.
* Must have a boat with separate pieces to allow manipulation of rotation of objects.

**Test Procedure:**

1. Start the executable of the project
2. Verify that wheel and cannons move based on user input.

**Expected Result:**

1. The wheel will rotate based on a rotation translation
2. The cannons will rotate based on a rotation translation

**Actual Result:**

1. The wheel rotated around a separate center point than shown in Unity.
2. The cannons rotated around a separate center point than shown in Unity.
3. The cannons were all attached and thus moving one cannon moved them all.

**User Story #1284 - Model Manipulation**

**Purpose:**

* To manipulate the 3D models of prefabricated objects used in Unity.

**Precondition:**

* Must have Maya installed.
* Must have a prefabricated object to edit.

**Test Procedure:**

1. Start Maya.
2. Import prefabricated object file.
3. Separate all linked objects in need of manipulation.

**Expected Result:**

1. All objects can be separated and loaded back into the original model.

**Actual Result:**

1. Model is stripped of all textures and materials when loaded.
2. Separating an object will separate every polygon in the object file causing hundreds of polygons to be rejoined and materials/textures to be reimplemented.

**User Story #1283 - Animation Creation (Revisited)**

**Purpose:**

* To allow the player to control the wheel of the boat based on controls and helm position.
* ~~To allow the player to control the cannons of the boat based on controls.~~ (Removed as an unnecessary feature of this project)

**Precondition:**

* Must have GetReal3D imported into Unity Project.
* Must use GetReal3D prefabricated character controller and camera.
* Must have a small environment created including a ship, water, lighting and the player.
* Must have a boat with separate pieces to allow manipulation of rotation of objects.

**Test Procedure:**

1. Start the executable of the project
2. Verify that wheel moves based on user input.

**Expected Result:**

1. The wheel will rotate based on a Unity Animator.

**Actual Result:**

1. The wheel rotated correctly using the Animator although with slight hiccups when changing directions.

**User Story #1283 - Animation Creation (Revisited again)**

**Purpose:**

* To allow the player to control the wheel of the boat based on controls and helm position.

**Precondition:**

* Must have GetReal3D imported into Unity Project.
* Must use GetReal3D prefabricated character controller and camera.
* Must have a small environment created including a ship, water, lighting and the player.
* Must have a boat with separate pieces to allow manipulation of rotation of objects.

**Test Procedure:**

1. Start the executable of the project
2. Verify that wheel and cannons move based on user input.

**Expected Result:**

1. The wheel will rotate based on rotation and helm position inside an empty game object with a new center point.

**Actual Result:**

1. The wheel moved correctly based on helm position and rotation transformation.

**User Story #1299 – Ship Enhancements 2**

**Purpose:**

* To allow the player to traverse the environment and create the beginning of the ship motion controls.

**Precondition:**

* Must have GetReal3D imported into Unity Project.
* Must use GetReal3D prefabricated character controller and camera.
* Must have a small environment created including a ship, water, lighting and the player.

**Test Procedure:**

1. Start the executable of the project
2. Move around as the player trying to get stuck or fall off the ship.

**Expected Result:**

1. Player should not fall off the ship.
2. Player should be able to move freely throughout the ship.
3. Player should not get stuck anywhere.
4. There will be a basic UI showing the player important information about the wind and the ship.

**Actual Result:**

1. The player can not fall off the ship while walking around.
2. The player can walk about freely on the ship without fear of falling off.
3. Player does not get stuck anywhere.
4. The UI does not show up when using the ICAVE, but shows properly through a PC.
5. The player will sometimes be randomly flung off the ship when entering the program.

**User Story #1300 – Ship Enhancements 3**

**Purpose:**

* To fix the bug affecting the z-axis rotation as the ship travels

**Precondition:**

* Must have GetReal3D imported into Unity Project.
* Must use GetReal3D prefabricated character controller and camera.
* Must have a small environment created including a ship, water, lighting and the player.

**Test Procedure:**

1. Start the executable of the project
2. Travel for an extended period of time; enough to notice a change in the z-rotation.

**Expected Result:**

1. The unexpected z-rotation will be removed.

**Actual Result:**

1. The unexpected z-rotation was removed.

**User Story #1285 - Project Environment Interaction**

**Purpose:**

* To have the player introduced into an interesting and easily recognizable environment in which they can traverse to test their knowledge of the ship mechanics and their navigation skills.

**Precondition:**

* Must have GetReal3D imported into Unity Project.
* Must use GetReal3D prefabricated character controller and camera.
* Must have a larger environment created including a ship, water, lighting, the player, and interesting obstacles to traverse.

**Test Procedure:**

1. Start the executable of the project.
2. Sail the boat through the environment.

**Expected Result:**

1. The map should include a slight challenge to provide insight into the sailing mechanics.
2. The map should have recognizable structures to give the player points of reference so they can accurately navigate the map.

**Actual Result:**

1. The map included small choke points with piles of rocks and sharp turns to slightly test the sailing of the player.
2. The map had recognizable structures in the forms of large cliffs in different shapes and sizes along with small islands surrounding the rocks.

**User Story #1286 - Objective Creation**

**Purpose:**

* To allow the player to dynamically collect game objects spread throughout the map.
* As one game object is collected another game object will activate at a random point in the map.

**Precondition:**

* Must have GetReal3D imported into Unity Project.
* Must use GetReal3D prefabricated character controller and camera.
* Must have a small environment created including a ship, water, lighting and the player.
* Must have an array of collectable game objects.

**Test Procedure:**

1. Start the executable of the project.
2. Sail the boat to collide with the pick up objects.

**Expected Result:**

1. The pick up object should disappear when collided with.
2. A different pick up object from the array of game objects should appear in a separate part of the map.

**Actual Result:**

1. The pick up object disappeared when collided with.
2. A different pick up object from the array of game objects appeared in a separate part of the map.

**User Story #1286 - Objective Creation (Revisited)**

**Purpose:**

* To allow the player to collect game objects spread throughout the map.
* This user story was revisited to become more in line with the purpose of the project as a whole.
* Randomly activated pick-ups is counter productive to the navigating and wayfinding nature of this project.
* A trail of game objects gives a more direct objective and an innate sense of direction the player should be traveling through, while still allowing exploration to collect game objects out of the breadcrumb order.

**Precondition:**

* Must have GetReal3D imported into Unity Project.
* Must use GetReal3D prefabricated character controller and camera.
* Must have a small environment created including a ship, water, lighting and the player.
* Must have an array of collectable game objects.

**Test Procedure:**

1. Start the executable of the project.
2. Sail the boat to collide with the pick up objects.

**Expected Result:**

1. All pick up objects should be visible initially.
2. The pick up object should disappear when collided with.

**Actual Result:**

1. All pick up objects were visible.
2. The pick up object disappeared when collided with.

**User Story #1303 – Collision Detection**

**Purpose:**

* To ensure that the player is stopped when colliding with a piece of terrain.

**Precondition:**

* Must have GetReal3D imported into Unity Project.
* Must use GetReal3D prefabricated character controller and camera.
* Must have a small environment created including a ship, water, lighting and the player.
* The terrain objects created must have colliders and have the “On Trigger” option selected.

**Test Procedure:**

1. Start the executable of the project.
2. Sail the ship to collide with the terrain objects.

**Expected Result:**

1. The ship will collide with the terrain and come to a halt.

**Actual Result:**

1. The ship collided with the terrain and came to a halt.

# Glossary

**Agile** - A form of project development in which tasks are divided into smaller “user stories” to be completed in pieces.

**Architectural Pattern -** Describes the fundamental principles used to design and develop the application.

**Collider** - A shape or form attached to an object in Unity that detects if any other object touches it.

**Deployment Diagram -** Diagramused to demonstrate the different aspects of the system in question needed to run the application.

**Design Pattern** - Describes common practices used to design and develop the application.

**GetReal3D** - A plug-in for Unity to assist in developing for motion tracking hardware.

**HTC Vive** - A brand of Virtual Reality headsets

**ICAVE** - A virtual reality system designed to track body motions in a physical space and allow the user almost a full 360 degree view of their surroundings.

**IDE** - An “Integrated Development Environment”. Assists programmers in software development by compiling, debugging, and editing code.

**Maya** - A graphics software. Allows graphic designers to create and edit various 2 and 3D models.

**Mingle** - A web application that assists users in managing an agile development workspace.

**Navigation** - The combination of the mental task of wayfinding and the physical task of traveling.

**Product Owner -** The user or manager in which a product is being developed for in an agile workspace.

**Scrum** - A software development framework formed from a Product Owner, a Scrum Master, and a development team. Used to develop products in a flexible environment with progress recorded and shared with the team through daily “scrum” meetings

**System and Subsystem Decomposition**

**TAM** - The acronym for the group this project falls under: Multi-Touch, Mid-Air, and Motion for Virtual and Augmented Reality

**Unity** - A game engine used to develop software that can be built and exported to many different platforms.

**User Story** - The base of an agile development workspace. Each user story defines a task that will be completed to progress through the project as whole.

**VIP** - Vertically Integrated Projects. This group aims to unite undergraduate education and faculty research in a team-based context through agile/scrum software development

**Virtual Reality -** The concept of entering and manipulating objects in an overlapping virtual space

**Visual Studio** - A Microsoft designed IDE used to assist in Windows based programming languages.

**Wayfinding** - The mental portion of navigation. The ability to discern your location relative to other locations and the approach behind reaching it.

# Appendix

## Appendix A - UML Diagrams



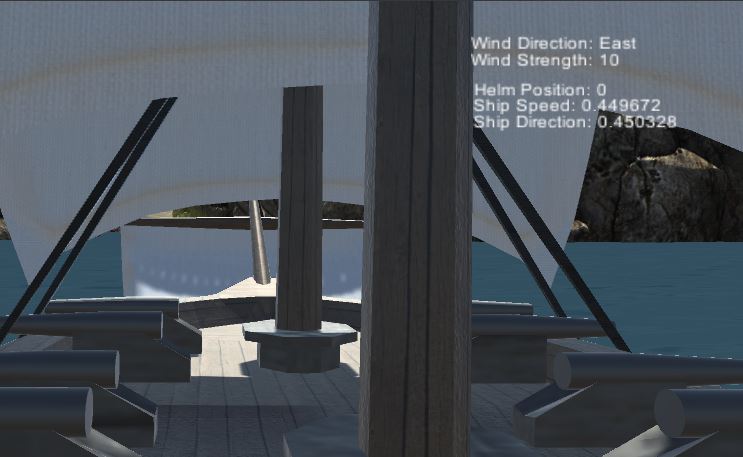
**Ship movement sequence diagram**



## Appendix B - User Interface Design



**The ship’s wheel is the only interactive part of the ship in the current design.The UI elements are very simplistic.**



**Free look is another action the player can do, allowing them to move around the ship.**

## Appendix C - Sprint Review Reports

**Sprint Review 1**

Attendees: Michael, Noel, Francisco

After showing and discussing all implemented user stories, the following user stories were accepted:

* #1165 Access to Unity
* #1167 Agile/Scrum Development Familiarity

The remaining following user stories were pushed back to be continued upon in the next Sprint:

* #1287 ICAVE Familiarity (Appointment was not able to be made to visit ICAVE)

**Sprint Review 2**

Attendees: Michael, Noel, Francisco

After showing and discussing all implemented user stories, the following user stories were accepted:

* #1166 Familiarity with ICAVE
* #1288 ICAVE Demo
* #1289 ICAVE Project Design

The remaining following user stories were pushed back to be continued upon in the next Sprint:

*None*

**Sprint Review 3**

Attendees: Michael, Noel, Francisco

After showing and discussing all implemented user stories, the following user stories were accepted:

* #1168 Research 3D Object Interaction
* #1187 Develop a small Demo
* #1296 Setup environment
* #1297 Ship Movement

The remaining following user stories were pushed back to be continued upon in the next Sprint:

*None*

**Sprint Review 4**

Attendees: Michael, Noel, Francisco

After showing and discussing all implemented user stories, the following user stories were accepted:

* #1283 Animation Creation
* #1284 Model Manipulation
* #1298 Ship Enhancements
* #1299 Ship Enhancements 2

The remaining following user stories were pushed back to be continued upon in the next Sprint:

*None*

1

**Sprint Review 5**

Attendees: Michael, Noel, Francisco

After showing and discussing all implemented user stories, the following user stories were accepted:

* #1283 Animation Creation (Was revisited to improve feature)
* #1286 Objective Creation
* #1300 Ship Enhancements 3

The remaining following user stories were pushed back to be continued upon in the next Sprint:

* #1285 Environment Interaction (Was not complete as this is a very tedious and time consuming task)

**Sprint Review 6**

Attendees: Michael, Noel, Francisco

After showing and discussing all implemented user stories, the following user stories were accepted:

* #1285 Environment Interaction
* #1302 ICAVE Control Remapping
* #1303 Collision Detection

The remaining following user stories were pushed back as a future feature to be implemented:

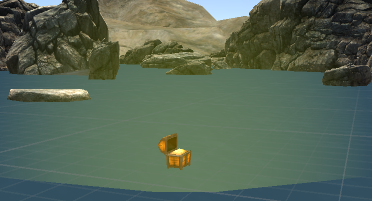
* #1301 Ship Controls (Was not completed)

## Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

**User Manual**

This program is designed to be a sailing simulator. When entering the program, you will begin by spawning on a ship in the center of a ring of rocks. Character controls can be used to move around the ship to get a better view of the surrounding area. Controls are also available to turn the wheel of the ship, affecting the position of the helm and ultimately the direction the ship is travelling.

This project is partially about the freedom of exploration of the test area and the simulation of sailing and partially about the aspect of navigation and wayfinding. The way the map is designed allows both aspects to come forth as there are both open sections of water and a trail to follow and objects to find. The intended trail of objects added is outlined in the figure below. Players can use the ship controls to follow this path and pick up all the collectible objects.



**Figure: The collectible pick up objects throughout the map. They glow and rotate to be able to be spotted by the user from a distance**



**Figure: The top down view of the map with labeled points for the starting area, location of pick up objects, and the intended trail to follow when picking up each object.**

**Installation/Maintenance**

Running and executing this program requires the executable file and DATA folder exported by Unity upon completion and can be found in the final Github folder for this project. Simply running the executable file while the DATA folder is in the same directory will allow the start of this program.

To edit and continue working on this project:

1. Download the project files from the Github folder
2. Install Unity version 5.3.2 <https://unity3d.com/get-unity/download/archive>
3. Install GetReal3D <http://icave.fiu.edu/getReal3D_for_Unity_3.3.0_x64.exe>
4. Start a new 3D project in Unity.
5. Import the project files package into your Unity project.
6. Import the GetReal3D plug in into your Unity project.
7. You may develop from any PC as long as these components are installed. Further testing will require exporting the executable and DATA folder to the ICAVE directories.

**Shortcomings/Wishlist**

There were many constraints on this project that could have been improved upon. Ultimately a lack of time was a large portion of the constraint on this project. As both developers had little to no experience with the Unity 3D game engine, were VIP Juniors (less time to work on projects), and had no experience with the ICAVE, there was a large amount of research to be done and a learning curve to overcome in a short amount of time. Many features came with unintended side effects which required more time to fix to a respectable level.

With more time, there are quite a few modifications/additions that we would have liked to make:

* More maps with larger selection of sailing environments to learn
* Dynamic random pick up positioning for an exploratory experience
* Multiplayer integration with multiple users piloting one or more ships
* More features such as compasses, telescopes, and sail manipulation
* Direct information on navigation and wayfinding within the contents of the game (pop-ups, virtual instructions)
* Use of GetReal3D’s “Wand” tools to mimic hands manipulating objects
* A mini map to assist the player in navigating the maps
* Audio files for the water, sailing, crashing, etc…

# References

Ambler, Scott W. "UML 2 Use Case Diagrams: An Agile Introduction." UML 2 Use Case Diagrams: An Agile Introduction. Ambysoft Inc., n.d. Web. 15 Apr. 2017. <http://agilemodeling.com/artifacts/useCaseDiagram.htm>.

Ambler, Scott W. "UML 2 Sequence Diagrams: An Agile Introduction." UML 2 Sequence Diagrams: An Agile Introduction. Ambysoft Inc., n.d. Web. 15 Apr. 2017. <http://agilemodeling.com/artifacts/sequenceDiagram.htm >.

Ambler, Scott W. "UML 2 Class Diagrams: An Agile Introduction." UML 2 Class Diagrams: An Agile Introduction. Ambysoft Inc., n.d. Web. 15 Apr. 2017. <http://agilemodeling.com/artifacts/classDiagram.htm >.

Ortega, Francisco R, Fatemeh Abyarjoo, Armando Barreto, Naphtali Rishe, and Malek Adjouadi. Interaction Design for 3d User Interfaces: The World of Modern Input Devices for Research, Applications, and Game Development. , 2016. Print.

"Roll-a-ball Tutorial." Unity. Unity, n.d. Web. 23 Feb. 2017. <https://unity3d.com/learn/tutorials/projects/roll-ball-tutorial>.

Serbia, Derrick, and Dhrumel Shah. Code by Cooking 0.0. Rep. Miami: Florida International U, 2016. Print.

Stanney, Kay M. Handbook of Virtual Environments: Design, Implementation, and Applications. Mahwah, N.J: Lawrence Erlbaum Associates, 2002. Internet resource.