*Florida International University*

*School of Computing and Information Sciences*

Software Engineering Focus

Final Deliverable

Project Title: Mult-touch-midAir-and Motion for Virtual and Augmented Reality (TAM) - 3D CodeBlocks

**Team Members:** Fernando de Zayas, Taufiq Islam

**Product Owner(s)**: Dr. Francisco R. Ortega

**Mentor(s)**: Masoud Sadjadi

**Instructor**: Dr. Francisco R. Ortega, 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 simple 3D block based programming. Block based programming is integrated using blueprints from Unreal Engine 4. The 3D block based programming is designed to facilitate the education of students with no background in Computer Science in the field of basic programming concepts.*

**Table of Contents**

**Introduction** …………………………………………………………………………………………………………………………….. 5

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

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

**User Stories**

Implemented User Stories ………………………………………….………………………………………………………..7

Pending User Stories …………………………………………………………………...………………………………..….. 17

**Project Plan**

Hardware and Software Resources ……………………………………………………...……………………….… 18

Sprints Plan ……………………………………………………………..………………………………………………………. 19

*Sprint 1*  …………………………………………………………………………………………………………………………... 19

*Sprint 2*  …………………………………………………………………………………………………………………………... 20

*Sprint 3*  …………………………………………………………………………………………………………………………... 21

*Sprint 4*  …………………………………………………………………………………………………………………………... 22

*Sprint 5*  …………………………………………………………………………………………………………………………... 23

*Sprint 6*  …………………………………………………………………………………………………………………………... 24

**System Design**

Architectural Patterns ………………………………………………………………………………………………….. 25

System and Subsystem Decomposition …………………………………………………………………………….… 25

Deployment Diagram …………………………………………………………………………………………………….…... 26

Design Patterns ………………………………………………………………………………………………………….….... 27

**System Validation**  …………………………………………………………………………………………………………………….28

**Glossary**  ………………………………………………………………………………………………………………………………….31

**Appendix**  ………………………………………………………………………………………………………………………………….32

Appendix A - UML Diagrams ………………………………………………………………………………………………. 32

*Static UML Diagrams*  ……………………………………………………………………………………………………….32

*Dynamic UML Diagrams*  …………………………………………………………………………………………………..32

Appendix B - User Interface Design …………………………………………………………………………….…... 36

Appendix C - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents …………………………………………………………………….…………… 40

**References** ……………………………………………………………………………………………………………………...………...42

# **Introduction**

The field of Computer Science is flourishing at an unprecedented rate, with incremental growth every year. Among other fields, Computer Science has become critical additions for the fields of education and business; however, there are still many students who are reluctant to pursue a career in Computer Science due to their initial impression of programming and core concepts being too difficult to grasp. This project aims to integrate the concept of Virtual Reality and Computer Science by means through an HTC-Vive device, with the goal being to spur the interest of students who plan to pursue a career in Computer Science and have them not be deterred by a learning curve.

## 

## **Current System**

Although there was ongoing research on Virtual Reality using Unreal Engine 4 and the HTC-Vive, there were no existing systems that were integrated and functional. Everything covered beyond this point is the initial programming integration of the research.

## **Purpose of New System**

The purpose of this system is to enable new students pursuing education in Computer Science with a simpler entry into programming, with basic concepts implemented in a simple visual format. VR CodeBlocks presents loop and variable manipulation in relation to basic games, such as Pong or a Dancing Stickman. Currently, students new to these concepts are forced into learning them with a pure text based IDE or a visual language such as Visual Basic. Although these methods have proven to work so far, more efficient methods may be present and should be tested.

# **User Stories**

The following section provides the detailed user stories that were implemented in this iteration of the Mult-touch-midAir-and Motion for Virtual and Augmented Reality (TAM) - 3D CodeBlocks 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 Name: Unreal Engine 4 Tutorials**

* As a developer, I want to complete Unreal Engine tutorials so I can begin the development of the 3D iteration of Code Blocks.

Acceptance Criteria

* Complete the tutorial on<https://www.raywenderlich.com/151018/unreal-engine-4-tutorial-beginners> to create a an actor and blueprint.
* Complete the tutorial series on<https://docs.unrealengine.com/latest/INT/Videos/PLZlv_N0_O1gZalvQWYs8sc7RP_-8eSr3i/r4tltrLLVuQ/> to grasp the basics of in game menus, actions, and communicating between blueprints.

**Use Case**

3D\_CodeBlocks\_UC\_001

* Name: Pick up an item
* Actor: Playtester
* Preconditions: Playtester has the completed project from the tutorial at <https://docs.unrealengine.com/latest/INT/Videos/PLZlv_N0_O1gZalvQWYs8sc7RP_-8eSr3i/r4tltrLLVuQ/> saved on their computer, along with a version of Unreal Engine between 4.14 and 4.15 installed on their computer.
* Description <Flow of events>: Playtester starts the game, moves towards an item with the arrow keys, playtester presses the “E” key when next to the item.

**Use Case Diagram**

Figure 1, Appendix A

**Unit Test**

3D\_CodeBlocks\_UT\_001

* Description/Summary of Test: Pick up an orange, open the inventory to eat it, and see the status change.
* Pre-condition: Playtester has the completed project from the tutorial at <https://docs.unrealengine.com/latest/INT/Videos/PLZlv_N0_O1gZalvQWYs8sc7RP_-8eSr3i/r4tltrLLVuQ/> saved on their computer, along with a version of Unreal Engine between 4.14 and 4.15 installed on their computer.
* Expected Results: Orange appears in inventory, can be eaten, and affects status properly.
* Actual Result: Orange appeared in the inventory, was eaten, and changed the status.
* Status (Fail/Pass): Pass

**Visual User Guide**

Appendix B, Figure 5

Appendix B, Figure 6

**User Story Name: Create Variable Code Blocks**

* As a developer, I want to create variable blocks that can hold data binded to a program or game which will reflect any changes onto the program or game itself, so that I can work on more complicated blocks that do more than just store information.

Acceptance Criteria

* Create blocks that hold data, such as text.
* Create a program or game in Unreal Engine that can bind its internal variables to these blocks.
* Test the binding to make sure the logic behind the binding and events is correct.

### **Related Tasks:**

* Allow for the block information to be accessed across levels
* GameMode settings research for individual level logic to work in concert
* SaveGame object research and implementation

**Use Case**

3D\_CodeBlocks\_UC\_002

* Name: Change to Pong Level
* Actor: Playtester
* Preconditions: Playtester has the current build of the project saved on their computer, along with a version of Unreal Engine between 4.14 and 4.15 installed on their computer.
* Description <Flow of events>: Playtester starts the game, moves towards the door labelled Pong using the arrow keys, touches the door with the character, and changes the game level to the Pong Menu.

**Use Case Diagram**

Figure 2, Appendix A

**Unit Test**

3D\_CodeBlocks\_UT\_002

* Purpose: Verify that the player character spawns and accepts user input after the P1\_Main level
* Pre-condition: CodeBlocksHTCVive project is installed on a computer with Unreal Engine 4.15
* Input:
  + User opens the Game Settings menu
  + User selects pong as the game to play
  + User clicks Start Game
  + User clicks 1P Pong
  + User presses the up and down arrow keys
* Expected Results: The right side player controlled paddle spawns and moves according to arrow key input
* Actual Result: The right side player controlled paddle spawned and moved according to arrow key input
* Status (Fail/Pass): Pass

**Visual User Guide**

Appendix B, Figure 7

Appendix B, Figure 8

**User Story Name: Finish Variable Code Blocks**

* As a developer, I want to externalize the internal variables of a simple game (Pong) to variable Code Blocks so that I can recreate a game and modify it in a simpler format.

Acceptance Criteria

* Be able to modify the variable blocks from a different level.
* Be able to transfer the correct information across levels.

**Use Case**

3D\_CodeBlocks\_UC\_003

* Name: Change Score Limit
* Actor: Playtester
* Preconditions: Playtester has the current build of the project saved on their computer, along with a version of Unreal Engine between 4.14 and 4.15 installed on their computer.
* Description <Flow of events>: Playtester starts the game, presses I, clicks the combo box next to Score Limit and clicks 5.

**Use Case Diagram**

Appendix A, Figure 3

**Unit Test**

3D\_CodeBlocks\_UT\_003

* Description/Summary of Test: Test if the score limit modified in the main level changes the score limit in the Pong level
* Pre-condition: CodeBlocksHTCVive project is installed on a computer with Unreal Engine 4.15
* Expected Results: The score limit is changed to 1 and the game ends after one goal
* Actual Result: The score limit was changed to 1 and the game ended after one goal
* Status (Fail/Pass): Pass

**Visual User Guide**

Appendix B, Figure 9

Appendix B, Figure 10

**User Story Name: Implement Loop Blocks**

* As a user, I want to be able to streamline the commands (which can be read from variable blocks) when they have to be repeated several times so that I can focus on higher level logic.

Acceptance Criteria

* Create a simple Dancing Stickman game to show images based on which command(content of variable block) is read
* Create For blocks to loop through commands

**Use Case**

3D\_CodeBlocks\_UC\_004

* Name: Change Level
* Actor: Playtester
* Preconditions: Playtester has the current build of the project saved on their computer, along with a version of Unreal Engine between 4.14 and 4.15 installed on their computer.
* Description <Flow of events>: Playtester starts the game, presses I, clicks the combo box next to Game, clicks Dancing Stickman, then clicks Start Game

**Sequence Diagram**

Appendix A, Figure 4

**Unit Test**

3D\_CodeBlocks\_UT\_004

* Purpose:Verify that the level change between the MainArea and DancingStickman levels works in both directions
* Pre-condition: CodeBlocksHTCVive project is installed on a computer with Unreal Engine 4.15
* Input:
  + User opens the Game Settings menu
  + User selects Dancing Stickman as the game to play
  + User clicks Start Game
  + User clicks Return
* Expected Results: User spawns in the correct location with the correct camera in the MainArea and DancingStickman levels
* Actual Result: User spawned in the correct location with the correct camera in the MainArea and DancingStickman levels
* Status (Fail/Pass): Pass

**Visual User Guide**

Appendix B, Figure 11

Appendix B, Figure 12

## 

**User Story Name: Finish Loop Blocks**

### **Description:**

## As a user, I want to have the option to repeat several commands so that I do not have to manually add those commands in situations that require a lot of repetition.

## 

### **Acceptance Criteria:**

* Finish the implementation of the Dancing Stickman game
* Finish the implementation of the For loop blocks.

## **Unit Test**

3D\_CodeBlocks\_UT\_005

* Description/Summary of Test: Confirm that the print statements return the same strings as those selected in the combo boxes.
* Pre-condition: CodeBlocksHTCVive project is installed on a computer with Unreal Engine 4.15, User has navigated to the DancingStickman level, set Loop combo box to Command 1-3, Duration combo box to 2, Command 2 combo box to Move Down, Command 3 combo box to Move Right
* Expected Results: Print statements in the following manner:
* Move Up, Move Down, Move Right, Move Up, Move Down, Move Right
* Actual Result: Print statements were in the order:
* Move Up, Move Down, Move Right, Move Up, Move Down, Move Right
* Status (Fail/Pass): Pass

**User Story Name: Unreal Engine 4 Interface**

* As a researcher, I want to study the Unreal Engine interface and learn more about the platform so I can understand how it works and can continue further research on the networking aspects of UE4.

Acceptance Criteria

* Completing Unreal Engine beginner tutorials.
* Opening a standard blueprint in UE4 and testing different functions on it.
* Locating videos and tutorials on the multiplayer and networking aspects of UE4.

**User Story Name: Networking in Unreal Engine 4**

* As a researcher, I want to look into how Unreal Engine 4 handles networking for its’ multiplayer functionality and how I can implement that for our subgroup’s HTC Vive project.

Acceptance Criteria

* View tutorials and articles on networking in UE4.
* Implement classes in UE4 that give standard blueprints multiplayer functionality.

**User Story Name: Networking and Multiplayer Functionality**

* As a developer, I want to research and implement a client/server model in UE4 to allow networking between two separate devices.

Acceptance Criteria

* Research about UE4’s core client/server technology for networking.
* Implement whichever classes / assets needed to get the client/server model running.
* Test a standard UE4 blueprint for multiplayer functionality.

**Use Case**

3D\_CodeBlocks\_UC\_005

* Name: Host a server
* Actor: User
* Preconditions: User has the current build of the networking project saved on their computer, along with a version of Unreal Engine between 4.14 and 4.15 installed on their computer.
* Description <Flow of events>: User opens the main menu, scrolls the mouse to hover over the ‘host game’ button, and clicks on the button.

**User Story Name: Cross-platform Multiplayer Functionality**

* As a developer, I want to modify the existing multiplayer functionality to include cross-platform functionality, for example (PC to Vive).

Acceptance Criteria

* Research about UE4’s multiplayer functionality when it comes to devices that are not PC’s.
* Have a running prototype blueprint that can run cross-platform multiplayer.

**Use Case**

3D\_CodeBlocks\_UC\_006

* Name: Join a hosted server
* Actor: User
* Preconditions: User has the current build of the networking project saved on their computer, along with a version of Unreal Engine between 4.14 and 4.15 installed on their computer.
* Description <Flow of events>: User opens the main menu, scrolls the mouse to hover over the ‘join game’ button, and clicks on the button.

## 

## **Pending User Stories**

**User Story Name: Unreal Engine 4 Networking in C++**

* As a researcher, I want to discover the advantages and disadvantages of shifting the current networking functionality from UE4 event graphing to C++ programming.

Acceptance Criteria

* View online resources about UE4’s networking aspects using C++ programming.
* Compare and contrast that to the already in place networking event graphs.

**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:

* Laptop: Windows 10 Pro 64 bit, Intel(R) Core(™) i7-6820HK CPU @ 2.7GHz 2.71 GHz , 16GB RAM, NVIDIA GeForce GTX 980M Graphics Card
* Laptop: Windows 7 Home premium 64 bit, Intel (R) Core(™) i7-3610QM CPU @ 2.30GHz 2.30GHz**,** 6GB RAM, NVIDIA GeForce GTX 660M
* Desktop: Windows 7 Home Premium 64 bit, Intel ® Core(™) i5-3570K CPU @ 3.40 GHz, 8.00 GB RAM, AMD R9 290 Series Graphics Card
* Software: [Epic Games Launcher](https://www.unrealengine.com/download)
* Software: Unreal Engine 4.15.1
  + Note, Epic Games Launcher must be installed first as of April 18, 2017. Unreal Engine can be installed in the Unreal Engine tab within the Epic Games Launcher.
* Software: [Visual Studio 2015](https://www.visualstudio.com/thank-you-downloading-visual-studio/?sku=Community&rel=15)
  + Note, include the Universal Windows Platform Development and Desktop Development with C++ options during the installation.
* HTC-Vive Device



**Sprints Plan**

### ***Sprint 1***

**Sprint 1 Planning Meeting Minutes:**

Attendees: Fernando de Zayas, Taufiq Islam

Date: 02/17/17

Start time: 10:30PM

End time: 11:00PM

After discussion, the velocity of the team was estimated to be normal.

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

# #1181 (HTC Vive) Unreal Engine 4 Tutorials

* #1210 (HTC Vive) Unreal Engine 4 Interface

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

* Fernando de Zayas
* User Story: Working on user story #1181 for Sprint 2 as well, since I was not able to finish it last sprint.
* Taufiq Islam
* User Story: Working user story #1210 for this Sprint, researching on UE4 and the platform as a whole.

***Sprint 2***

**Sprint 2 Planning Meeting Minutes:**

Attendees: Fernando de Zayas, Taufiq Islam

Date: 02/24/17

Start time: 10:30PM

End time: 11:00PM

After discussion, the velocity of the team was estimated to be normal.

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

# [/#1207] (HTC Vive) Create Variable Code Blocks

[#1211] (HTC Vive) Networking in Unreal Engine 4

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

* Fernando de Zayas
* User Story: Working on user story #1181
* Taufiq Islam
* User Story: Working on user story #1211 for this Sprint, researching about networking and multiplayer functionality in UE4.

### ***Sprint 3***

**Sprint 3 Planning Meeting Minutes:**

Attendees: Fernando de Zayas, Taufiq Islam

Date: 03/03/17

Start time: 10:30PM

End time: 11:00PM

After discussion, the velocity of the team was estimated to be normal.

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

# [/#1207] (HTC Vive) Create Variable Code Blocks

[#1212] (HTC Vive) Networking and Multiplayer Functionality

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

* Fernando de Zayas
* User Story: Working on user story #1207 for Sprint 3
* Taufiq Islam
* User Story: Working on user story #1212 for this Sprint, and currently in the process of researching how to set up the client/server model to be able to test it in UE4 with at least two devices.

***Sprint 4***

**Sprint 4 Planning Meeting Minutes:**

Attendees: Fernando de Zayas, Taufiq Islam

Date: 03/10/17

Start time: 10:30PM

End time: 11:00PM

After discussion, the velocity of the team was estimated to be normal.

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

# [/#1305] (HTC Vive) Finish Variable Code Blocks

[#1212] (HTC Vive) Networking and Multiplayer Functionality

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

* Fernando de Zayas
* Sick and currently unable to work
* Taufiq Islam
* Continued work on user story [#1212] and implemented a client/server networking model on one of the standard UE4 provided blueprints.

### ***Sprint 5***

**Sprint 5 Planning Meeting Minutes:**

Attendees: Fernando de Zayas, Taufiq Islam

Date: 03/24/17

Start time: 10:30PM

End time: 11:00PM

After discussion, the velocity of the team was estimated to be normal.

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

# [#1309](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/mult_touch_midair_and_motion_f/cards/1309) (HTC Vive) Implement Loop Blocks

# [#1294](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/mult_touch_midair_and_motion_f/cards/1294) (HTC Vive) Cross-platform Multiplayer Functionality

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

* Fernando de Zayas
* Sick and unable to work
* Taufiq Islam
* Beginning research user story [#1294] on how to update the existing client/server networking to extend cross-platform functionality.

### ***Sprint 6***

**Sprint 6 Planning Meeting Minutes:**

Attendees: Fernando de Zayas, Taufiq Islam

Date: 03/31/17

Start time: 10:30PM

End time: 11:00PM

After discussion, the velocity of the team was estimated to be normal.

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

# [/#1310] (HTC Vive) Finish Loop Blocks

# [[#1306](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/mult_touch_midair_and_motion_f/cards/1306)] (HTC Vive) UE4 Networking using C++

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

* Fernando de Zayas

# [/#1310] (HTC Vive) Finish Loop Blocks

* Taufiq Islam

# Continued research and implementation on user story [#1294], along with research on user story [#1306].

### 

# **System Design**

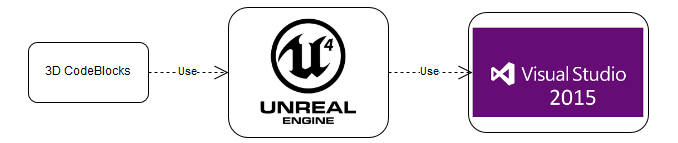
This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

## **Architectural Patterns**

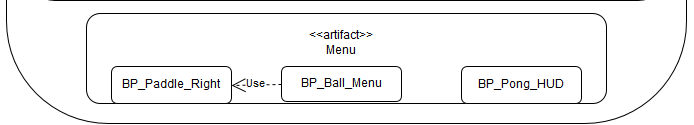
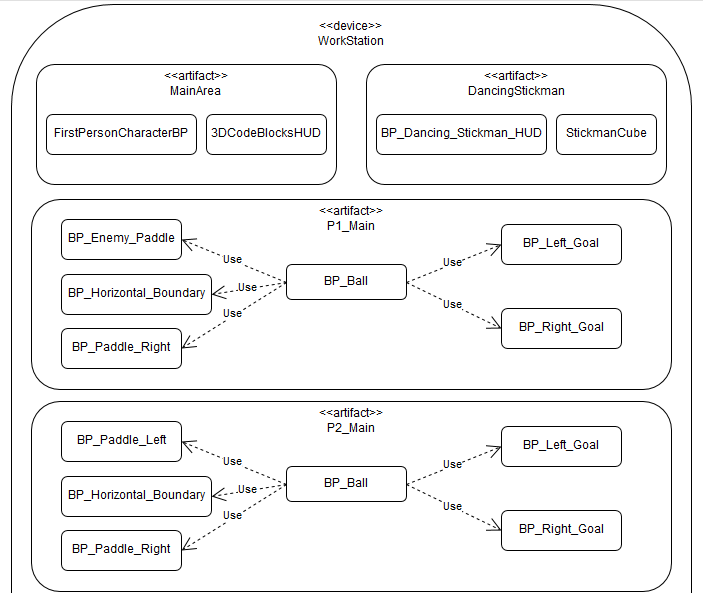
3D CodeBlocks is built around the Event-Driven Architecture, where the sensing, creation of, and response to events is the main focus. This is a required architecture due to programming with blueprints in Unreal Engine 4, as the blueprints themselves are event driven.

## 

## **System and Subsystem Decomposition**



**Deployment Diagram**



## **Design Patterns**

3D CodeBlocks uses the following Design patterns:

* Lazy Initialization : Before a level is needed, it is not loaded.
* Singleton : HUD blueprints for each level are Singletons and referenced as such.
* Marker: The custom GameMode classes have no functionality but define metadata such as which class is the default player character
* Null Object: Every blueprint that requires an object or reference will not allow compilation until one of the correct type is provided.

# 

# **System Validation**

**Use Cases**

3D\_CodeBlocks\_UC\_001

* Name: Pick up an item
* Actor: Playtester
* Preconditions: Playtester has the completed project from the tutorial at <https://docs.unrealengine.com/latest/INT/Videos/PLZlv_N0_O1gZalvQWYs8sc7RP_-8eSr3i/r4tltrLLVuQ/> saved on their computer, along with a version of Unreal Engine between 4.14 and 4.15 installed on their computer.
* Description <Flow of events>: Playtester starts the game, moves towards an item with the arrow keys, playtester presses the “E” key when next to the item.

3D\_CodeBlocks\_UC\_002

* Name: Change to Pong Level
* Actor: Playtester
* Preconditions: Playtester has the current build of the project saved on their computer, along with a version of Unreal Engine between 4.14 and 4.15 installed on their computer.
* Description <Flow of events>: Playtester starts the game, moves towards the door labelled Pong using the arrow keys, touches the door with the character, and changes the game level to the Pong Menu.

3D\_CodeBlocks\_UC\_003

* Name: Change Score Limit
* Actor: Playtester
* Preconditions: Playtester has the current build of the project saved on their computer, along with a version of Unreal Engine between 4.14 and 4.15 installed on their computer.
* Description <Flow of events>: Playtester starts the game, presses I, clicks the combo box next to Score Limit and clicks 5.

3D\_CodeBlocks\_UC\_004

* Name: Change Level
* Actor: Playtester
* Preconditions: Playtester has the current build of the project saved on their computer, along with a version of Unreal Engine between 4.14 and 4.15 installed on their computer.
* Description <Flow of events>: Playtester starts the game, presses I, clicks the combo box next to Game, clicks Dancing Stickman, then clicks Start Game

3D\_CodeBlocks\_UC\_005

* Name: Host a server
* Actor: User
* Preconditions: User has the current build of the networking project saved on their computer, along with a version of Unreal Engine between 4.14 and 4.15 installed on their computer.
* Description <Flow of events>: User opens the main menu, scrolls the mouse to hover over the ‘host game’ button, and clicks on the button.

3D\_CodeBlocks\_UC\_006

* Name: Join a hosted server
* Actor: User
* Preconditions: User has the current build of the networking project saved on their computer, along with a version of Unreal Engine between 4.14 and 4.15 installed on their computer.
* Description <Flow of events>: User opens the main menu, scrolls the mouse to hover over the ‘join game’ button, and clicks on the button.

**Unit Tests**

3D\_CodeBlocks\_UT\_001

* Description/Summary of Test: Pick up an orange, open the inventory to eat it, and see the status change.
* Pre-condition: Playtester has the completed project from the tutorial at <https://docs.unrealengine.com/latest/INT/Videos/PLZlv_N0_O1gZalvQWYs8sc7RP_-8eSr3i/r4tltrLLVuQ/> saved on their computer, along with a version of Unreal Engine between 4.14 and 4.15 installed on their computer.
* Expected Results: Orange appears in inventory, can be eaten, and affects status properly.
* Actual Result: Orange appeared in the inventory, was eaten, and changed the status.
* Status (Fail/Pass): Pass

3D\_CodeBlocks\_UT\_002

* Purpose: Verify that the player character spawns and accepts user input after the P1\_Main level
* Pre-condition: CodeBlocksHTCVive project is installed on a computer with Unreal Engine 4.15
* Input:
  + User opens the Game Settings menu
  + User selects pong as the game to play
  + User clicks Start Game
  + User clicks 1P Pong
  + User presses the up and down arrow keys
* Expected Results: The right side player controlled paddle spawns and moves according to arrow key input
* Actual Result: The right side player controlled paddle spawned and moved according to arrow key input
* Status (Fail/Pass): Pass

3D\_CodeBlocks\_UT\_003

* Description/Summary of Test: Test if the score limit modified in the main level changes the score limit in the Pong level
* Pre-condition: CodeBlocksHTCVive project is installed on a computer with Unreal Engine 4.15
* Expected Results: The score limit is changed to 1 and the game ends after one goal
* Actual Result: The score limit was changed to 1 and the game ended after one goal
* Status (Fail/Pass): Pass

3D\_CodeBlocks\_UT\_004

* Purpose:Verify that the level change between the MainArea and DancingStickman levels works in both directions
* Pre-condition: CodeBlocksHTCVive project is installed on a computer with Unreal Engine 4.15
* Input:
  + - User opens the Game Settings menu
    - User selects Dancing Stickman as the game to play
    - User clicks Start Game
    - User clicks Return
* Expected Results: User spawns in the correct location with the correct camera in the MainArea and DancingStickman levels
* Actual Result: User spawned in the correct location with the correct camera in the MainArea and DancingStickman levels
* Status (Fail/Pass): Pass

**Glossary**

HUD - Head Up Display, a visual way to display relevant information to the user, such as the status in a game or the flight information in a plane.

# **Appendix**

## **Appendix A - UML Diagrams**

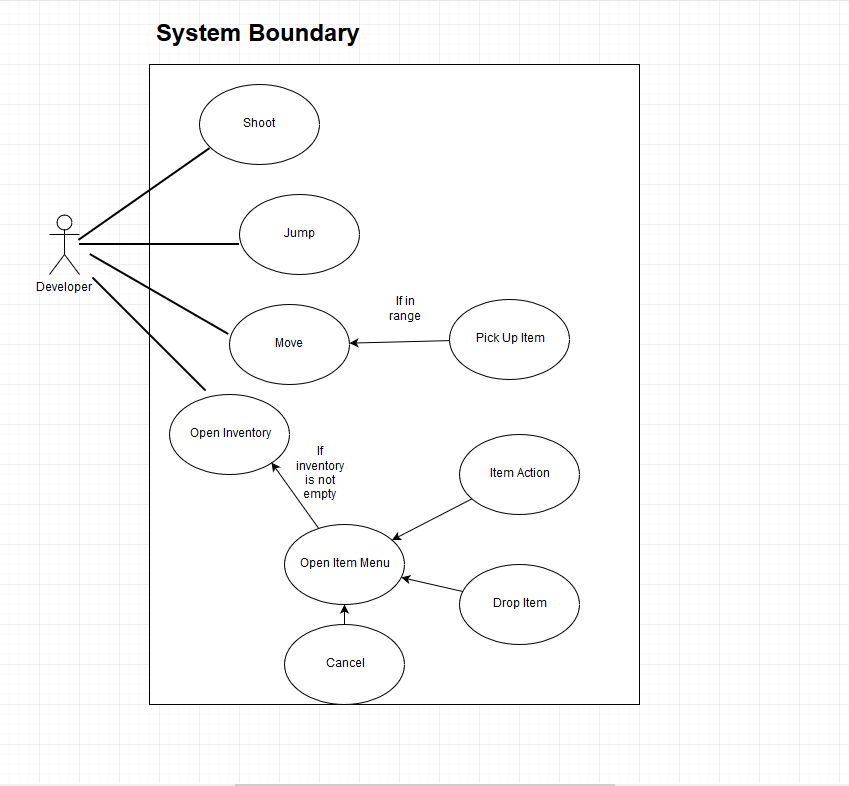


Figure 1

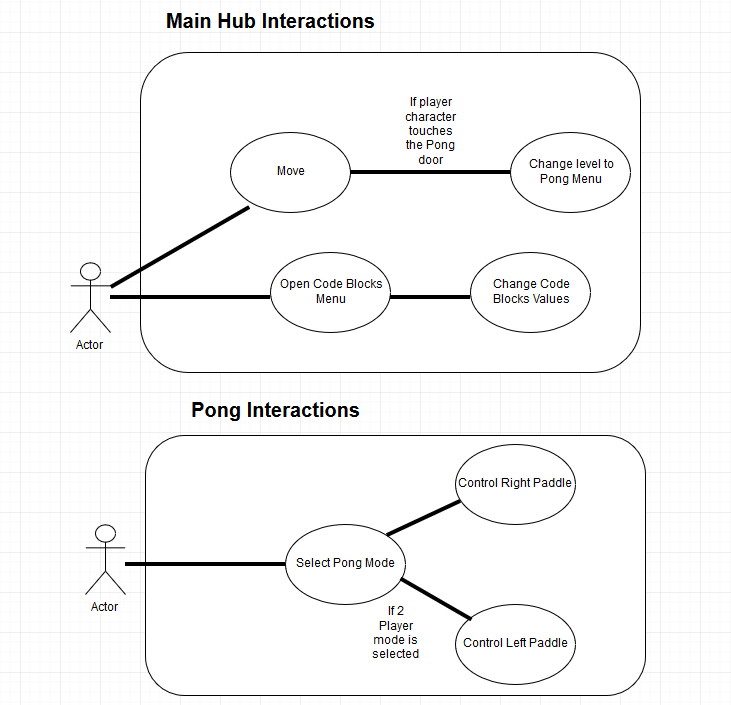


Figure 2

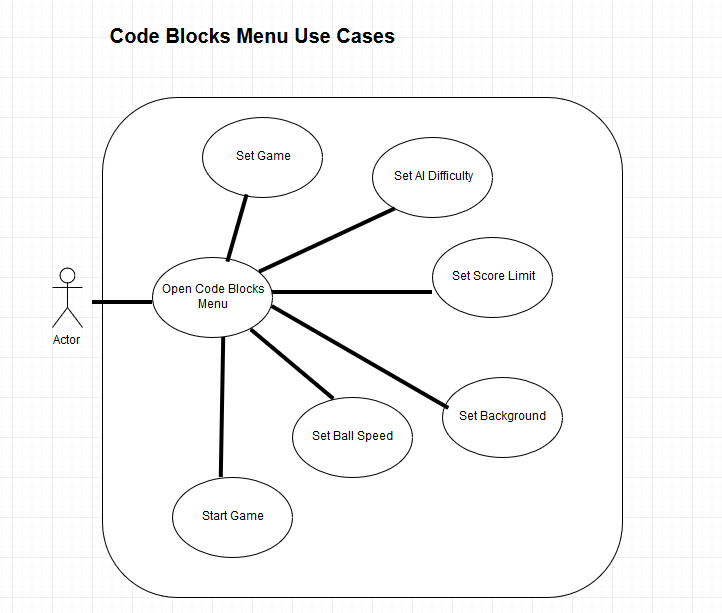


Figure 3

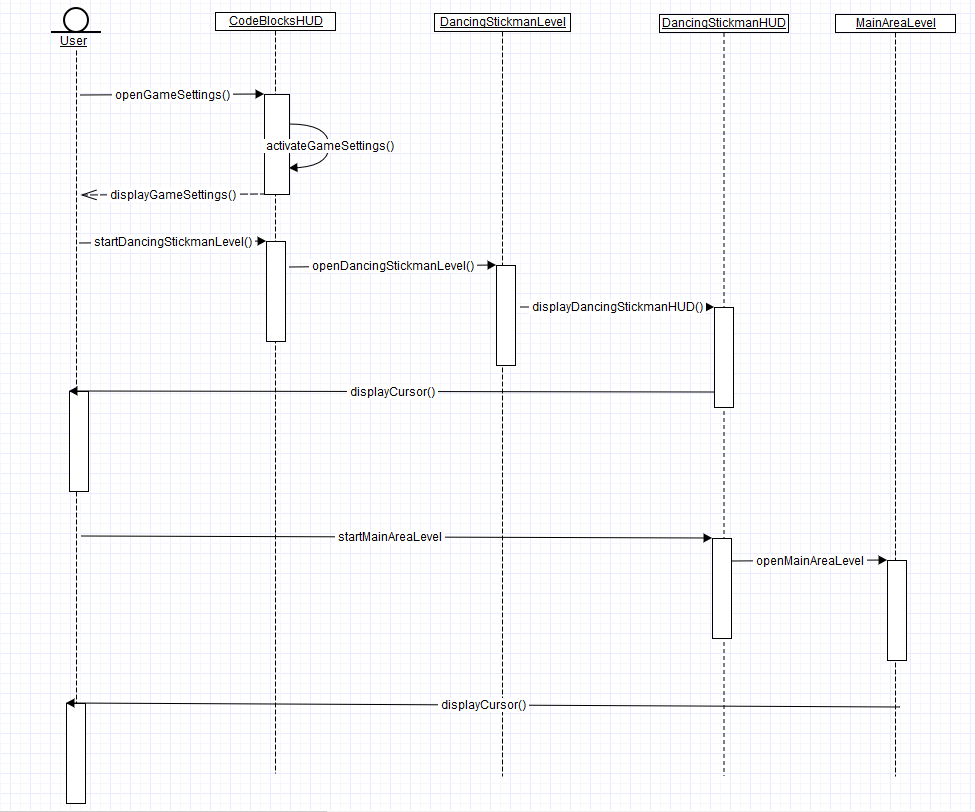


Figure 4

## 

## 

## 

## 

## 

## 

## **Appendix B - User Interface Design**



Figure 5



Figure 6



Figure 7

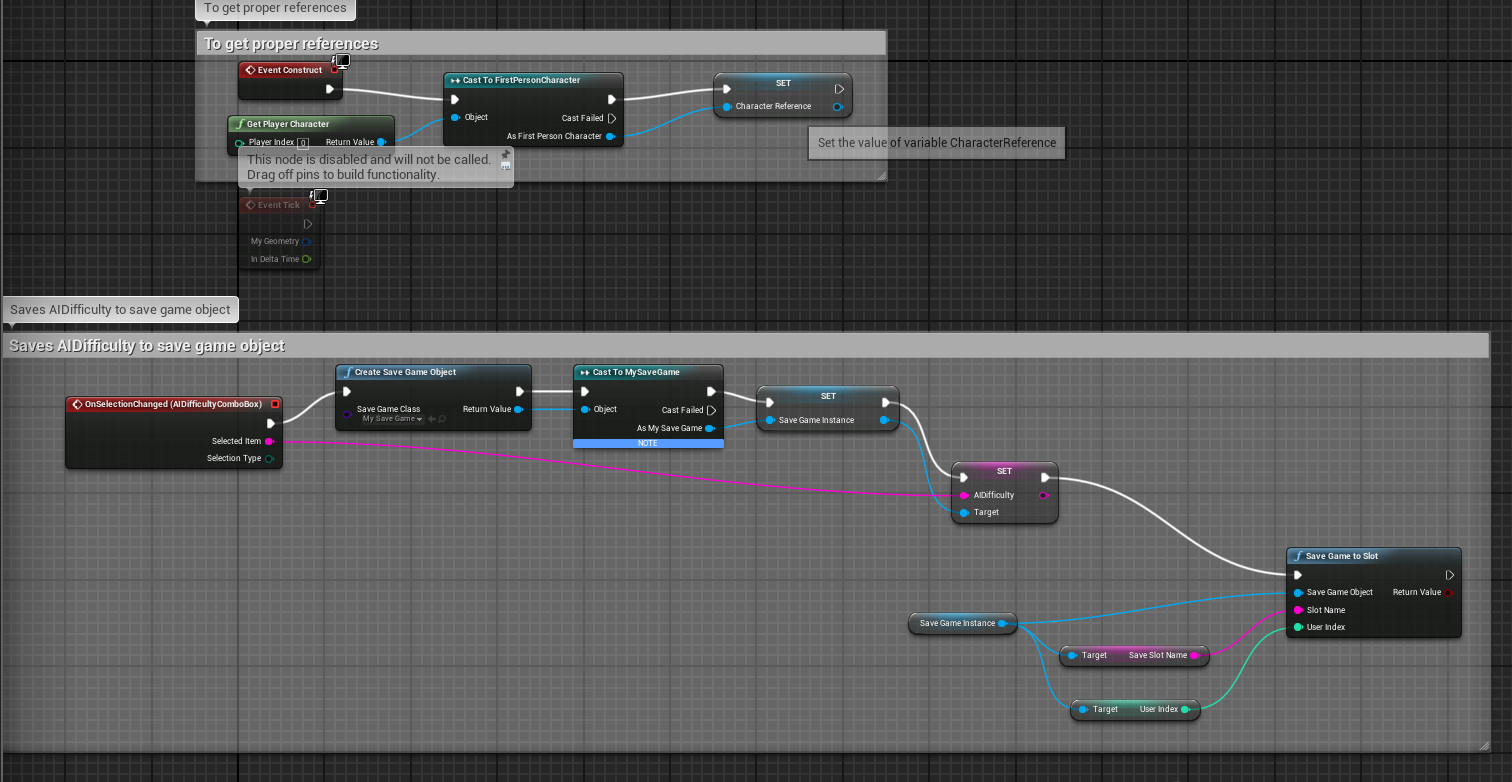


Figure 8

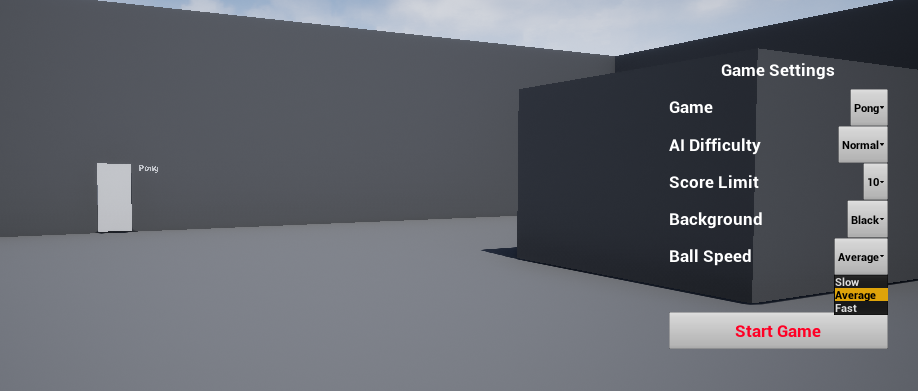


Figure 9

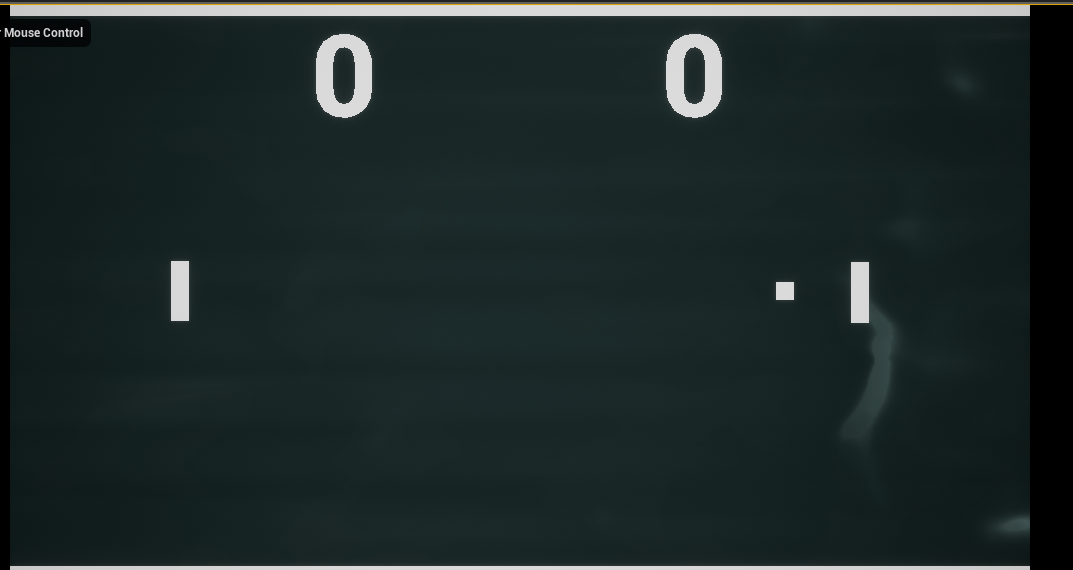


Figure 10

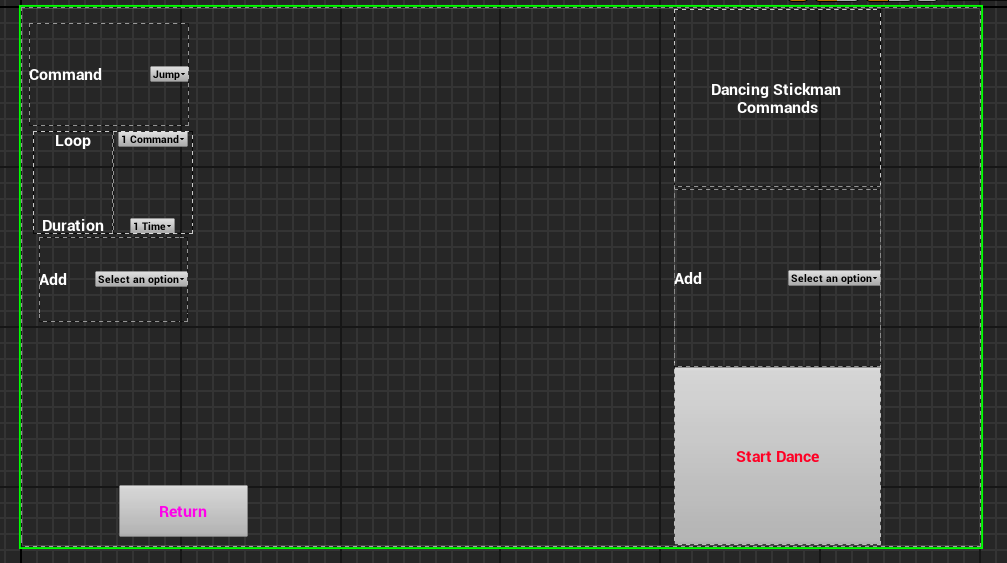


Figure 11

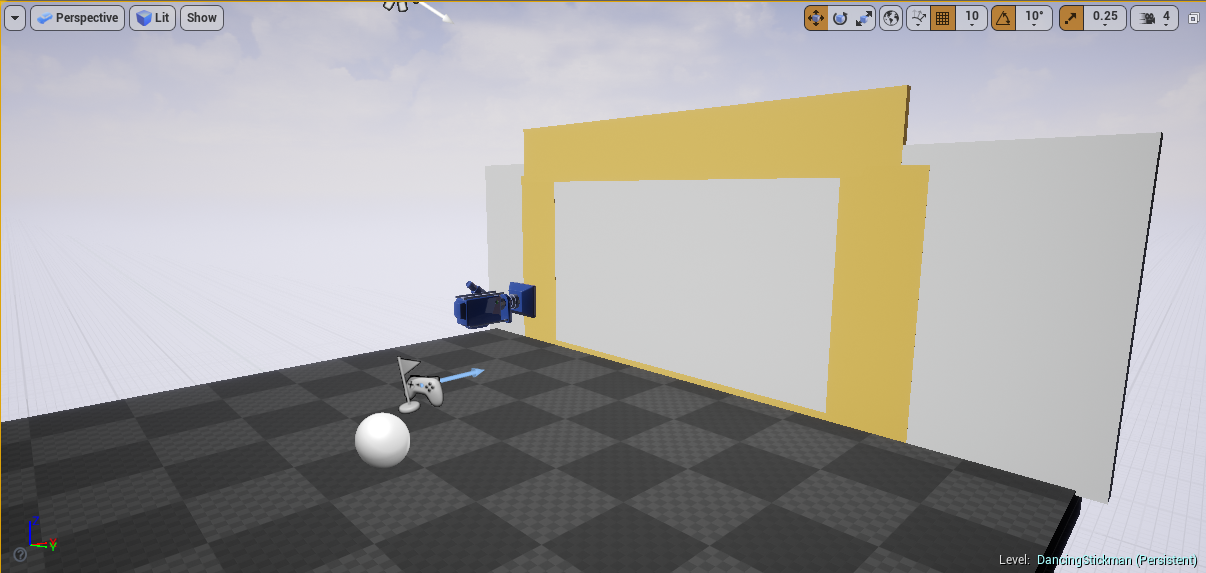


Figure 12

**Appendix C - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents**

**Installation/Maintenance**

Installing 3D CodeBlocks requires opening the CodeBlocksHTCVive.uproject file by browsing to its location through Unreal Engine 4’s option of opening an existing project.

If a version of Unreal Engine 4 later than 4.15.1 is installed, clone the project first and then open the clone to check if project can still compile.

**User Manual**

After installation, open the project through Unreal Engine 4.

Play from the selected viewport.

To open the Game Settings menu, press the “G” key on the keyboard.

Use the mouse to select the game to open.

If Pong was selected, more options appear to change the values in the game. (Appendix B, Figure 9)

Click Start Game after choosing a game (and changing the values, if applicable).

In Pong:

Click 1P Pong for a one player Pong game against an AI.

Click 2P Pong for a two player game on the same computer.

Mouse over Controls to see the controls for Player 1 and Player 2

Click Return to go back to the MainArea level.

In 1P Pong or 2P Pong:

Press the “R” key on your keyboard to restart the current match.

Press the “Q” key on your keyboard to go back to the previous menu.

In DancingStickman:

Set the values of Commands 1-4 to any movement of your preference.

Set the value of Loop to which commands you want to loop through.

Set the value of Duration to how many times you want to loop through the commands.

Click Start Dance to show the text of the commands in the top left, looped in order.

Click Return to go back to the MainArea level.

**Shortcomings**

The VR use of the project requires changing the player spawn location to the current camera location, but this breaks the spawning for the characters when they spawn at their default locations.

The Dancing Stickman level only has a stickman cube that does not move, as the cube cannot be referenced properly from the HUD.

The lack of time, if there was more time allotted, more features could have been implemented and tested.

**Wishlist**

A second HTC-Vive device could have also benefitted feature testing, especially with regard to cross-platform testing.

**References**

[1] “UMG UI Inventory” *Unreal Engine 4 Documentation* Web. 08 Feb 2017.

<<https://docs.unrealengine.com/latest/INT/Videos/PLZlv_N0_O1gZalvQWYs8sc7RP_-8eSr3i/index.html>>.

[2] “How To Make Pong in UE4” *Alex Young* Web. 24 Feb 2017.

<<https://www.youtube.com/playlist?list=PLSfh6YsA0Lhu8r5xVSp5pnMziDkdt-vrW>>