# Natural User Interface and Virtual Reality Integration in Video Games

Last Revised: February 8, 2015

Alexandre Wimmers adwimmers@ucdavis.edu
Mario Yepez myepez@ucdavis.edu
Timothy Tong tktong@ucdavis.edu
Valerie Gadjali vgadjali@ucdavis.edu

## 1 Background

### 1.1 Natural User Interface

Natural user interface is the ability to use an application without the need for any controller. In this case, the players own body or parts of the body becomes the controller. The concept of natural user interface first appeared in 1990, but the first popular usage for video games appeared in 2010 with the release of the Microsoft Kinect. Traditional keyboards and mice can only interact with a monitor or any visual interface in a two-dimensional sense. However, with a natural user interface it becomes possible to interact in three dimensions.

### 1.2 Virtual Reality

Virtual reality is an immersive experience within a computer-simulated environment. First conceptualized in 1930, but only recently has it become a popular with the Oculus Rift Development Kit 1. It provides the user a complete immersion and a new perspective into the virtual world. Whereas a monitor restricts a user to only view in two dimensions, virtual reality allows the user to see the world in three dimensions.

### 1.3 Unity Game Engine

Unity is the game engine of choice for this project. Developed by Unity Technologies, Unity includes both a game engine and an integrated development environment. There are a variety of factors that led to the adoption of Unity, such as portability. Games built on Unity are able to run on multiple platforms; this allows for game development regardless of operating system. A host of plug-ins and libraries are also available for Unity such as the A* Pathfinding Project and Photon, which will prove to be useful in development of this project. The most notable plug-in and library is the Oculus Rift and Leap Motion compatibility. Other game engines require workarounds while Unity makes integration simple.

## 2 Goal

Take an existing video game genre and develop a video game integrating both natural user interface and virtual reality.

## 3 Game Genres

### 3.1 Real-Time Strategy (RTS)

As a subcategory of strategy video games, real-time strategy is a type of war game that requires you to manage resources and fight with an opponent in real time. Unlike turn-based games, RTS games require quick reaction and thinking to succeed. The term real-time strategy first appeared in 1982 in a game called Cytron Masters by Dani Bunten Berry.

#### 3.1.1 Pros

- True three dimensional real-time strategy.
- ”Cool” factor for in-person perspective with combat.
- High customization.

#### 3.1.2 Cons

- Steep learning curve.
- Reduced precision in controls due to technology.
- In-person perspective has no objective gain in competitive gameplay, but rather impedes player. Zooming out to bird’s eye view while aligning to one of three axis will provide a much more effective perspective which defeats the purpose of certain technologies in this game.

### 3.2 Castle Defense

The original version of Castle Defense was a Flash-based browser video game called “Defend Your Castle”. In this game, there were a series of different enemies who would walk to the castle and attack it, but all of them were in a stick figure art style. Players were to prevent enemies from destroying the castle, by using a variety of methods, such as drag and dropping the enemies or flinging them away. In 2008, a milder version of this game was released on the Wii and included a multiplayer mode. The weapons that were used were daily household items and the use of blood was removed. In 2009, an iOS version was release and implemented the finger-touch method for controls rather than a mouse. This game genre has received completely mixed reviews; some critics and users enjoyed the game play, but disliked the graphics, while many thought the vice versa.

#### 3.2.1 Pros

- Easy to pickup and play.
- Requires less precision for hand gestures.

#### 3.2.2 Cons

- Near zero gameplay customization.
- Standard gameplay is simpler in complexity compared to real-time strategy.

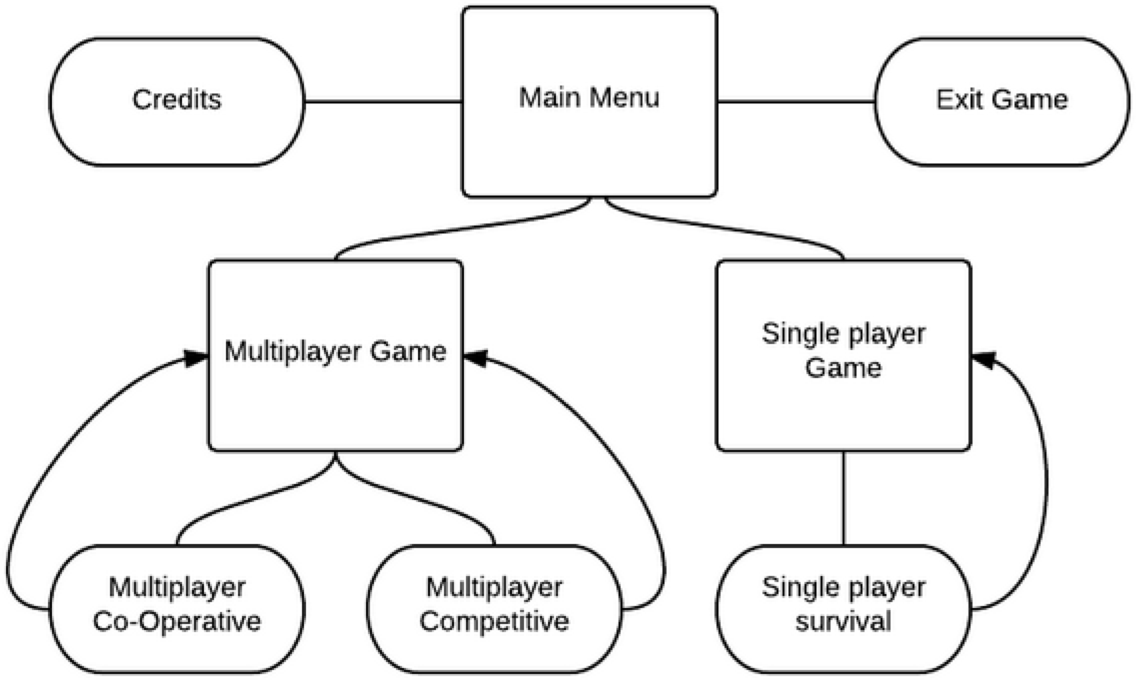
# 4 Implementation

## 4.1 Genre of Choice

Castle Defense.

After weighing the Pros and Cons of both game genres the team opted for the Castle Defense game. Castle Defense targets a wider audience of players and this is mainly due to the fact that there is a steeper learning curve for a Real Time Strategy game. Another critical reason was the fact that, for the real-time strategy game, the standard birds-eye view was more effective than the virtual reality in-person view. People of all ages will be able to pick up and play the game easily and start having fun.

## 4.2 Game Flow



One of the core principles for designing the game was to allow the user to jump right into the action and start playing without the hassle of navigating through a series of menus. Menus break immersion if done incorrectly, which is why we have decided to minimize menus as much as possible. As can be seen from the flow-chart above, the user does not need to fiddle with options to get into the game.

## 4.3 Victory Conditions

Victory conditions will vary depending on the type of game mode that the player(s) select.

For the Single Player Survival and Multiplayer Cooperative mode, the user(s) must survive wave after wave of enemies that attack their castle. If the player(s) manage(s) to defend the castle against a set number of waves then they will win the game and be sent to a Victory! screen. However if the player fails to do so they will be sent to a "Game Over!" screen.

For the Multiplayer Competitive mode, two players will compete with each other to see who is the dominant one. Player A will be be defending the castle as in the Single Player Survival mode. Player B will take control of the waves that are attacking the castle. Player B will be responsible for spawning the waves and sending them through paths that lead to the castle. If Player A successfully fends off 3 waves Player A will be declared the victor. On the other hand, if Player A fails to do so then Player B will win the game.