

# TEAM JAKD

## WIICONTROL

Final Progress Report

4/28/2009

James Garcia, Aaron Bonebright, Kiranbir Sodia, Derek Weitzel

University of Nebraska-Lincoln

### 1. ABSTRACT

The purpose of this project report is to provide feedback on the progress for the project that has been designed by team JAKD, the Virtual Reality (VR) Head Tracking System. Topics of discussion include a brief review of past problems and difficulties, previous concepts and ideas that brought us to where we are today, the feasibility for this new project idea, the overall design progress at this point, future plans and aspirations, a timetable of completion, and a final conclusion with a description of the final product.

### 2. INTRODUCTION

The VR Head Tracking System is an innovative and new approach to an existing technology employed by the Nintendo Wii Videogame Console. It is a completely wireless system that allows the user to not only interact with the three dimensional environment created on the screen (just as a simple video game controller, keyboard, or mouse can do), but it also allows the user to actually be a part of the environment through the use of its wireless head tracking capability. The only wires needed are for displaying the environment on a monitor or projection screen. It has always been a question of whether or not one day a system would enable a person to actually be a part of the three dimensional environment new graphic displays, such as televisions and monitors, are able to beautifully display.

#### 2.1 *Nintendo Wii*

Nowadays, since displays are able to portray an environment so realistically, the question of how they can possibly improve is a hot topic of discussion among researchers and people alike. The Nintendo Wii introduced a new technology that attempts to solve this question by allowing a player to provide hand gestures and movement in space to interact with the game, but it does not solve the question of how they can actually be in the environment themselves. Even though the player is able to interact with the game by

moving around with the Wiimote, it does not matter where in space the player is located because the Nintendo does not know where they are at, only their proximity, nor does the environment react and change when their position changes. This continues to leave the question open for discussion.

## 2.2 CAVE

Towards the end of the 20<sup>th</sup> century, a new technology called the *Cave Automatic Virtual Environment* (CAVE) was introduced from the University of Illinois at Chicago. Specifically, this technology was designed in response to a competition challenge that called for devices submitted to utilize large screen projections to simulate three-dimensional images. It consists of a small custom “room” inside of a larger room, whose walls are actually the projection screens in which the images are projected from the rear using mirrors. The entire “room” is surrounded by a type of non-magnetic stainless steel frame to reduce or eliminate the chances of outside interference. Users enter the CAVE through the one side of the “room” not enclosed by a projection screen. Computers that control the images on-screen will rapidly generate pairs of images that are intended to be seen by only one eye of the user at a time. The user wears a special type of glasses called Stereoscopic LCD shutter glasses that will be synchronized with the refresh rate of the projectors. This is so that the glasses will actually “block out” one eye each iteration of the refresh rate as so the user will only be able to see the correct image at any given time. All of these elements put together at a high projection refresh rate will give the user a three-dimensional feel for the environment around them. The computer system generating the images simply provides the environment for the user, but the user is actually the one controlling the environment. This is due to the fact that the system knows where the user is located within the CAVE and is able to adjust the images along with the user’s movement. Finally, speakers positioned at various spots within the CAVE will enable the system not only to give three-dimensional images to the user, but also three-dimensional audio. A sample image of the CAVE is shown in Figure 2-1.



**Figure 2-1**

A photograph of a person inside of the CAVE.

*Source: Dave Pape, Assistant Professor at the University of Buffalo*

While the idea and implementation of the CAVE is certainly an amazing feat of human innovation, it is in no way a practical means that will allow the general public to have their own CAVE system in their own homes. The CAVE is a highly system that works outstandingly, but it carries a high price tag in which only large corporations or universities are the only sites that are able to contain and demonstrate one. It is a technology that needs to be optimized to where an average person may be able to purchase one and enjoy what it has to offer. To date, there is not a similar technology in which the user is submersed in total 3-D that a person can purchase within a reasonable price limit, nor is there a similar technology in which the overhead is within a reasonable limit. This includes necessary hardware, software, power consumption, and portability.

### *2.3 VR Head Tracking*

The VR Head Tracking System provides a new and innovative approach that attempts to provide a solution to the concept of 3-D immersion. Not only will it attempt to provide the user with an interactive environment, but it also will be an incredible solution that is low in overall cost and overhead, while being incredibly easy to use and maintain. It is the intention of its creators that this system will be widely available to all who want to experience what it has to offer. It is designed to enable the user to actually be a part of the environment displayed on the screen. The system not only reacts when the user provides motion and input, but it also knows where the user is in space and is able to adjust the environment according to their position. This allows the user to perform many different actions and responses to their “surroundings,” such as looking out a window, hiding behind walls, and walking down a hallway.



**Figure 2-2**  
Custom sensor bar designed specifically for the VR Head Tracking System.

The system employs a custom designed sensor bar which contains a rear-facing Wiimote to determine the user's location, as shown in Figure 2-2. Not only does the sensor bar emit infrared, but the LED glasses (with infrared LED replacements) shown in Figure 2-3, are employed. The user wears these glasses, which act as a secondary sensor that enables the system to determine their orientation to the screen. The system uses texturing extensively to reduce the polygon count. In effect, this increases the response capability of the environment, lowering the chances that the environment will be slow and unresponsive, as well as allowing it to run on older hardware. In order to interact with the environment, the user will hold a Wiimote, shown in Figure 2-4, connected via Bluetooth and use the buttons to provide input. The entire system is a completely wireless technology, with the exception of how to connect the appropriate display screen to the computer.



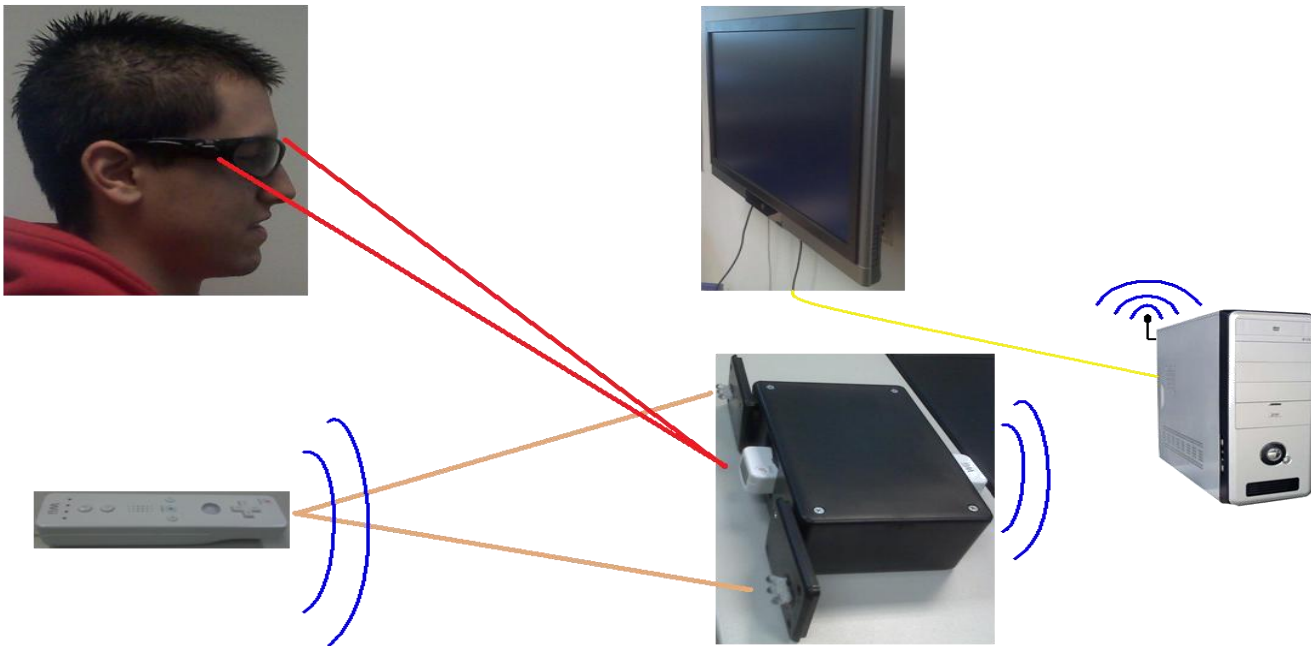
**Figure 2-3**  
Infrared Glasses used by the VR Head Tracking System.



**Figure 2-4**  
An image of a Wiimote, courtesy of *TechDigest* magazine.

### 3. SYSTEM OVERVIEW

An overall system view makes it easier for the user to see how each component works together and provides a better basis for understand how the system works in general. Figure 3-1 shows a full system overview, complete with each hardware component and their relations to each other. Section 2.3 describes the hardware configuration in more detail.



**Figure 3-1**  
Full Hardware System Overview

### 4. EXPERIMENT

It is crucial that the hardware be designed, built, and corrected early so that proper testing of the software can take place. If the hardware does not work correctly, it is not possible to test the software since it heavily relies on working components. Despite the fact that the hardware is a highly important part to the working system, there are not a lot of components present. Once the hardware aspect of the system is in place, this provides grounds to test the software components.

There are several steps to take to ensure the system is complete and error free. Each of these steps have been completed before the deadline to enable a well designed final product to be realized. First, it is important to acquire all of the working hardware early so that, as the system is being designed, the software can be properly tested for correctness.

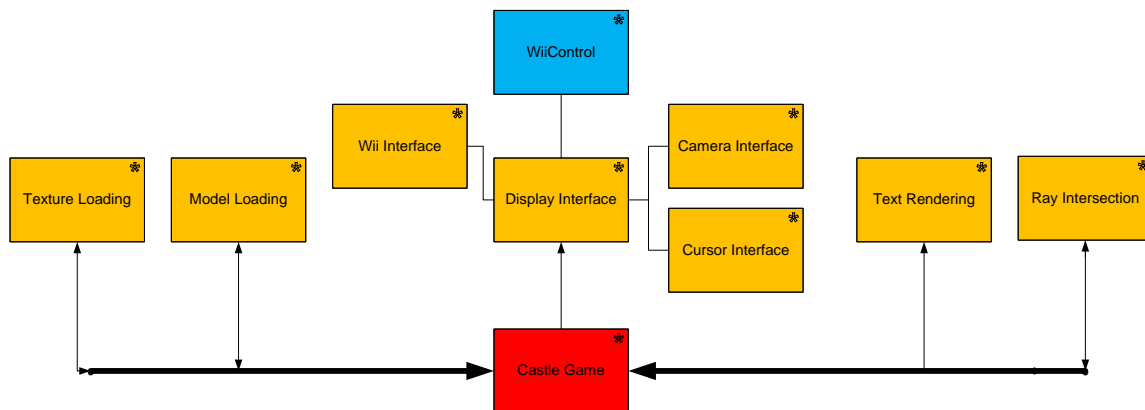
The hardware consists of two Wiimotes (one for head tracking and one for cursor placement), a custom designed sensor bar, and infrared glasses. The sensor bar design closely resembles that of the Wii sensor bar so that it can be compatible with the Wiimotes.

The software contained in the system is concerned with creating the environment, determining player position and response, and enabling the environment to react accordingly. This includes displaying the environment correctly on-screen by rendering it with the motion of the player and their reaction. The software is written in the C++ programming language and is portable to computers with Bluetooth connectivity.

#### 4.1 Hardware

The Wiimotes, sensor bar, infrared glasses, and optional Bluetooth Adapter (if the user's computer is not Bluetooth enabled) consist of the entire hardware spectrum. The Wiimotes are built and tested before they are shipped to customers, so it is assumed that these work correctly. Up to this point, this assumption has been highly reliable as virtually zero adjustments have been made to compensate for any Wiimote error. The user should not have any anticipation that the Wiimotes will fail, other than installing new batteries when the current ones run low. The sensor bar has proven to be a reliable piece of hardware thus far. During experimentation, it responds with very little (if any) delay, and the expectation that it will work correctly from now until the project's completion has held true, proving that it can be dependable if this system were to ever be mass produced and distributed. When testing the first pair of infrared glasses, it was immediately noticed that the user has a very limited motion area. The reason is because the LEDs on each side of those glasses are set too far behind the front of the user's face. Recently, a new pair of glasses has been purchased to provide for a wider viewing angle. Experimenting with this new pair will help us determine whether or not the problem will be solved. This is explained in more detail in the *Results* section. The custom sensor bar contains three LEDs placed on each side that will make their focal points brighter, therefore making it more effective at cursor positioning. During experimentation, the LEDs were not bright enough to produce an infrared field that the Wiimotes can obtain and process without a diffusion glass in front of them, making them very dim at wide angles. A new wireless sensor bar has been purchased that contains a diffusion glass, spreading the infrared light over a greater viewing angle. This new sensor bar has been tested to ensure that the viewing angle is sufficient enough to move around freely in front of the screen. The result of the testing of the new hardware is explained in more detail in the *Results* section.

## 4.2 Software



**Figure 4-1**

Diagram representing the software setup and the class relations to each other.

The software side of the system has been implemented in the C++ programming language. There are several components to the software portion of the head tracking system, such as cursor positioning, head tracking and orientation, Bluetooth connectivity, wireless message parsing/processing, environment/object rendering, and user interaction response. The classes that implement these are described in more detail below. The Wiimote works by aiming its infrared viewing camera towards an infrared light source with two focal points, as provided by using two sets of infrared LEDs on our custom sensor bar shown in Figure 2-2.

WiiControl is designed to have separate, pluggable scenes that can render and respond to events independently. The Wiimotes communicate to the scenes through this pluggable interface. There are several top level utilities that can be used with the scenes, as shown in Figure 4-1, including: Camera Interface, Wii Interface, Display Interface, Text Rendering, Cursor Interface, Model Loading, Texture Loading, and Ray Intersection. Currently, the only scene is the Castle Game.

WiiControl uses the OpenGL rendering API. OpenGL is the standard 3D library on non-Windows operating systems. WiiControl uses as much OpenGL standard utilities as possible, with the exception picking. The problems and solution with picking are discussed in the *Results* section.

## 4.3 Problem Solving

A problem that had been revealed and corrected is that while using the Wiimote to see how accurate the cursor positioning is, the system had difficulty placing the cursor near the edge of the screen. This is because when the cursor, or handheld Wiimote, is



aimed near the edge of the screen, the on-board infrared camera is unable to view both infrared focal points of the sensor bar, thus causing it to be unable to determine where exactly the cursor should be positioned next, as well as which direction it is traveling. Before, this problem could be fixed simply by re-aiming the cursor back towards the center of the screen so that both focal points can be viewed. At this point, however, with the use of a highly mathematical positioning algorithm, the system is able to use the last known position of the “lost” focal point in order to determine where in the environment the cursor should be placed next. In simple terms, the user is now able to move the cursor to the edge of the screen where there is only one focal point in view, and the system has the capability to understand how to handle this obstacle with no error.

Also included in the experimentation and testing is speed of rendering and environment orientation. The question of “How realistic does the environment look when the player moves around?” has been answered by adjusting some speed values to make movement more realistic and accurate according to the natural motion of the person using the system.

#### *4.4 Problem Solving*

It is crucial that the hardware works as close to perfect as possible to reduce any chances of software error. As expected, the Wiimotes are proving to be a reliable piece of hardware that has, up to this point, provided no error.

Upon purchasing the new LED glasses, it was immediately noticeable that the LEDs are set back from the front of the user’s face just as far; however, they are positioned slightly farther apart off the side of the head. This allows the viewing angle to be greater while the user turns their head in either direction. While testing the new glasses versus the old ones, the user is able to turn their head further from center using the new glasses. Because a new sensor bar has been purchased that replaces the old custom sensor bar, we have tested the response of the system using both the old sensor bar and the new one.

The new wireless sensor bar contains a diffusion glass in front of the built-in infrared LEDs. This diffusion glass simply diffuses the light over a greater area, allowing the Wiimote held by the user to see it from more positions around the room rather than limiting the user to stand directly in front of the screen. The combination of wider-angle infrared glasses and diffused light sensor bar will allow the system to become more responsive in terms of use around a wider area inside of a room where the system is located.



When testing the picking (as described in Section 5.2), we noticed a large delay when processing the click. The problem was determined to be the default OpenGL picking routines. OpenGL drops to software mode when operating in the special selection mode used for picking. Software rendering is when the graphics card is bypassed, and the rendering is done entirely on the processor. Software rendering is drastically slower than rendering on the graphics card. Because of this delay, we decided to use a Ray-Polygon intersection algorithm. In this algorithm, a box is placed over each enemy, and each mouse click is translated into a ray pointing from the camera position. Trivial math can be then be used to determine if the ray intersects this box.

Also while testing, we found that on machines that had a lower frame rate, such as a laptop, the infrared reaction time was drastically longer than on larger machines. After much debugging, it was determined this was because too many events were being sent through the Wiimote to be processed by the Wii Interface. To solve this problem, we changed the Wii Interface to process every event that has occurred each frame, rather than one event per frame.

## 5. COST ANALYSIS

In order to obtain an understanding of the feasibility of this project, it is important to compare this technology with previous work. A necessary method of comparison is a cost analysis that compares the cost of a previous design with that of the new technology. This will determine whether or not it is reasonable to continue working on a new design.

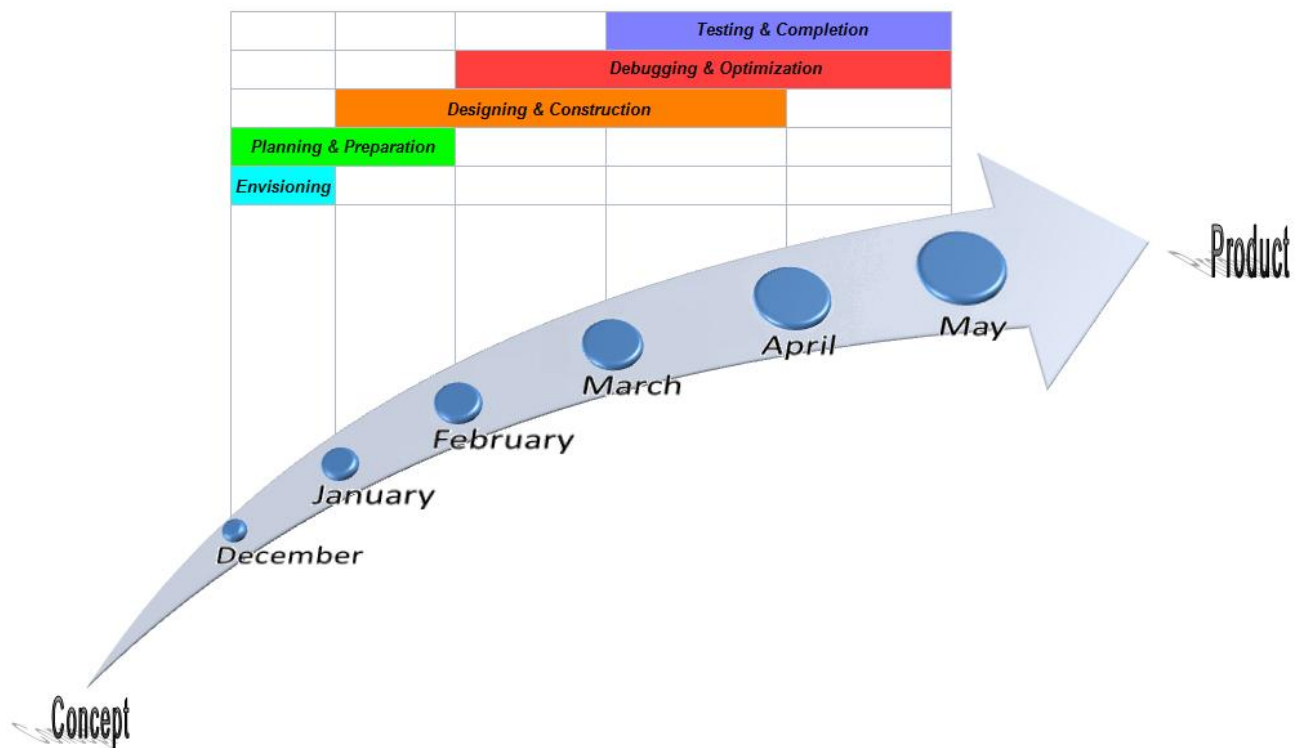
<b>CAVE Cost Analysis</b>			
<i>Component</i>	<i>Each</i>	<i>Quantity</i>	<i>Cost</i>
Rear Projection Screen	\$89.99	6	\$539.94
High-Resolution Projector	\$4,999	6	\$29,994
Stereoscopic LCD Shutter Glasses	\$49.99	1	\$49.99
Desktop PC	~\$900	6 (cluster)	\$5,400.00
Steel Wall Skeleton	?	1 (10' x 10' x 9')	?
<b>TOTAL</b>			<b>\$35,983.93</b>

<b>VR Head Tracker Cost Analysis</b>			
<i>Component</i>	<i>Each</i>	<i>Quantity</i>	<i>Cost</i>
Wiimote	\$39.99	2	\$79.98
Sensor Bar	\$15.99	1	\$15.99
LED Glasses	\$12.79	1	\$12.79
Infrared LEDs	\$0.25	2	\$0.50
<b>TOTAL</b>			<b>\$109.26</b>

According to each of the cost analysis tables, the VR Head Tracking system costs \$10,080.71 less than the CAVE system. In other words, our system is roughly 1% of the cost of the previous CAVE technology, while providing the three-dimensional look and feel that is desired in a system of this type.

## 6. CONCLUSION

The VR Head Tracking System has become an increasingly dependable system that proves to be a feasible piece of work in three-dimensional environment interaction. It immerses the user into an interactive 3-D environment in a new way, removing complex hardware and software needed in more expensive systems, such as CAVE, while providing the same effects. Not only do the experimentation and test results provide a reasonable view of the functionality of the system, but these tests also show some improvements and optimizations (such as more responsive software or better environment rendering) that can be made, as well as a way for Team JAKD to determine whether or not the system can be completed before the deadline as it was being designed. They clearly show that the system works as expected, while other aspects may be added to improve functionality, such as other environments and sound.



**Figure 6-1**  
Design process timeline.

Upon submission of this project, the VR Head Tracking System is working as intended as it was first imagined as a concept. All components have been implemented and are now working together to create an excellent piece of work, as it was supposed to be. A timeline of the design process is shown in Figure 6-1. The hardware consists of two Wiimotes, a customized sensor bar, a pair of infrared glasses, and the ability to portray the environment displayed on-screen as a three-dimensional “arena” that the user can change

their perspective in. In order to show how the technology works, a “castle” type game has been designed that employs the benefits of using this technology. The game consists of the user “standing” on a castle wall defending against automated enemy opponents by using a gun to eliminate them before they reach and destroy the castle wall. In order to view the field in which the enemies are approaching from, the user must move around and “look around” the stone pillars in front of them so that the enemies will be in view. This demonstrates the ability of the system to orient the environment according to the user’s movements and interact with the user when an action is taken. The project was placed on display at the University of Nebraska’s E-Week competition on April 24, 2009 in Othmer Hall. There were many other projects on display, but this one clearly stood out among the rest as its vibrant displays immersed patrons all around with its interactive, easy to use interface. An image of the project on display at E-Week, as well as its creators, are shown in Figure 6-2.



**Figure 6-2**

An image of the VR Head Tracking System on display at E-Week on April 24, 2009. Also shown are its creators, listed below

*From left to right: James Garcia, Aaron Bonebright, and Derek Weitzel.*

From an economical standpoint, the VR Head Tracking System is quickly proving to be a highly feasible solution to 3-D immersion technology with its ease of use, low cost, low hardware overhead, and portability. It is a highly dependable system as shown by its high quality hardware and software. As Team JAKD continues to work towards a more perfect design solution by sifting through bugs and improvements, the technology will only become even more influential to visual enthusiast and normal users alike, and is a huge step towards making this technology more widely available.

## 7. REFERENCES

- [1] 3dstereo (2009). *Wireless Shutter Glasses*. Retrieved March 23, 2009 from <http://www.3dstereo.com/viewmaster/ed-wx.html>
- [2] Best Buy (2009). *Nintendo - Remote Controller for Nintendo Wii*. Retrieved March 23, 2009 from [http://www.bestbuy.com/site/olspage.jsp;jsessionid=3V1LCXNVH0DKJKC4D3HVAFQ?\\_dyncharset=ISO-8859-1&id=pcat17071&type=page&st=wiimote&sc=Global&cp=1&nrp=15&sp=&qp=&list=n&iht=y&usc=All+Categories&ks=960](http://www.bestbuy.com/site/olspage.jsp;jsessionid=3V1LCXNVH0DKJKC4D3HVAFQ?_dyncharset=ISO-8859-1&id=pcat17071&type=page&st=wiimote&sc=Global&cp=1&nrp=15&sp=&qp=&list=n&iht=y&usc=All+Categories&ks=960)
- [3] CAVE (2009). *Cave Automatic Virtual Environment*. Retrieved March 23, 2009 from <http://en.wikipedia.org/wiki/Cave Automatic Virtual Environment>
- [4] Newegg (2009). *EliteSCREENS M100NWV1 100" Manual Projection Screen*. Retrieved March 23, 2009 from <http://www.newegg.com/Product/Product.aspx?Item=N82E16824999107>
- [5] Newegg (2009). *Canon REALiS SX6 1400 x 1050 LCD Projector 3500 ANSI Lumens 1000:1*. Retrieved April 27, 2009 from <http://www.newegg.com/Product/Product.aspx?Item=N82E16824189013>
- [6] Newegg (2009). *NYKO Wireless Sensor Bar Wii*. Retrieved March 23, 2009 from <http://www.newegg.com/Product/Product.aspx?Item=N82E16878112002>
- [7] Nintendo (2009). *Wii at Nintendo*. Retrieved March 23, 2009 from <http://www.nintendo.com/wii>