Augmented Reality Games for
Upper Limb Rehabilitation
First Semester Report
Fall Semester 2013

- Full report -

by
Joshua Wallace
Ryan Nash
Jacob Thornton

Prepared to partially fulfill the requirements for
ECE401

Department of Electrical and Computer Engineering
Colorado State University
Fort Collins, Colorado 80523

Project advisor: ___Sudeep Pasricha___

Approved by: _____Sudeep Pasricha___
ABSTRACT

Numerous people go through upper limb rehabilitation every year due to strokes, accidents or surgery. The goal of our project is to provide a rehabilitation environment that encourages and stimulates the patient in a way clinical rehab currently does not. Our games focus on stimulating three types of motion: horizontal, vertical, and depth. Through the use of simple games we can disguise these motions as the controls to our games. We have developed a number of augmented reality games which are aimed at assisting patients in upper limb rehabilitation. Our games will be developed on both the Leap Motion Controller interface device and will be played on a PC. We have developed our games using HTML5 and JavaScript along with various game development libraries. By making our games web-based the player will have the ability to simply log in to a website from their home computer to access and play the games. This also enables the collection and consolidation of game data for each patient which can then be viewed by the patient’s physical therapist in order to track progress. This system would be useful for many people across the globe.

This project has been a senior design project for two years at Colorado State University. The first version of the project used a Web Cam to implement Object Recognition and user tracking. The second version of the project attempted to do the same thing on a mobile computing platform. We took the successes and failures of these projects into account for our own purposes.

We went through a number of phases to reach our current solution. Initially we started developing with the Microsoft Kinect. This initial research consisted of investigation into available Kinect libraries, as well as supported functionality. We needed to get an idea of the type of games that the Kinect could support. The most useful part of the Kinect was the skeleton class. It allowed the programmer easy access to information about the placement of extremities. This data could then be monitored to catalogue the user performing prescribed events.

The goal of our project, however, was not to monitor the entire user. Instead we were tasked with the rehabilitation of upper limbs. This skeleton provides us with extra, unnecessary features. Fearing these extra features were just something to go wrong, we sought a system that isolated the upper body. The Leap Motion Controller was released over the summer and uses infrared cameras to report high resolution information about the user. Using this device we could monitor the number of fingers extended, whether the user’s hand was balled in a fist or flat, and most importantly the speed and direction of the movement.

Finally we had to keep in mind the state of our users. The Augmented Reality Games system is intended for people suffering from upper limb rehabilitation, and many of these people struggle with simple tasks. With this in mind, we built our games to be played with the user resting his hand on the table, while the Leap Motion Controller was mounted above. This restricted motion to the X and Z dimensions, but benefited users who otherwise could not extend their arm in front of them.

Our efforts have led to a one of a kind setup that allows patients to continue rehab in the comfort of their own home. Unlike rehabilitation done without this technology, progress can be
tracked by storing results on a server. These results can be viewed by the patient and therapist to get an idea of the progress that has been made. The end result of our system is a new way of doing in home rehabilitation that offers many of the benefits of rehab performed on site.

Advancements in technology have only recently allowed for this type of system. Bringing technology into the home has been a major trend in the past few decades, but ways in which this can be done are always evolving. We are developing something that is one of a kind, and as a result have nothing to measure ourselves by. We must instead develop a system to replicate results that would be achieved at a clinic, but in your own home.

We anticipate the bulk of the remaining work to be in refining the system through a series of clinical trials. What is “good enough” for us may not meet the needs of the therapy community. We planned the completion of our project around this and pushed to have a working prototype by the end of the first semester.

Further work includes designing a robust framework that can be applied to all of the games. We need to ensure that our games will work on any monitor size, computers of all performance levels and differing internet speeds. Ensuring our system works on all computers will require extensive documentation and testing on all computers we can get our hands on.

The final aspect to consider is the effectiveness of our work. There is no system in the world like ours, and we hope to confirm over the second phase of development the effectiveness of our system. The clinical trials will serve as a form of customer feedback as we refine our project. Once these refinements are finished, we are confident in a working system that will enhance the quality of life for people suffering from neurological and physical impairments.
# TABLE OF CONTENTS

1. Title
2. Abstract
3. Table of Contents
4. List of Figures and Tables

   6. Introduction
   
   9. Summary of Work
   
   9. Section 1 - Hardware
   
   11. Section 2 - Software Selection
   
   12. Section 3 – The Database
   
   14. Section 4 – The Web Interface
   
   20. Section 5 – The Game Engine
   
   22. Alternatives, Testing and Ethics
   
   24. Conclusions and Future Work
   
   26. References

A-1. Appendix A - Abbreviations

B-1. Appendix B - Budget

C-1. Appendix C – Additional Documents
# LIST OF TABLES AND FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Database Model</td>
<td>13</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Default Layout View</td>
<td>14</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Login View</td>
<td>15</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Register View</td>
<td>15</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Default View</td>
<td>16</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Main Page View</td>
<td>17</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Ribbon</td>
<td>18</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Information Window</td>
<td>18</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Profile View</td>
<td>19</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Progress View</td>
<td>19</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Game Engine</td>
<td>20</td>
</tr>
</tbody>
</table>

| Table 1   | Team Expenditures            | 27   |
| Table 2   | Personal Expenditures        | 27   |
Chapter I - Introduction

Physiatry, commonly known as rehabilitation, dates back to the two major wars of the 20th century. Physiatrists recognized the need for physical stimulation of soldiers before they could return to combat or everyday life. Major Frank Granger was designated the head of this new physiotherapy service, setting up reconstruction units in fifty-three army and general hospitals across the nation. Patients were assigned exercises of varying difficulty based on the progress of their rehabilitation. From these seeds the profession of physical therapy grew into its own field. Specialists emerged to usher the field into a respected status.

Physical therapy grew in the thirties to include elements like hydro and spa therapy. A Society for Physical Therapists was founded in 1938. World War II saw an explosion in the field. The Baruch Committee was founded to explore advancements in the field. The committee focused on education, teaching, research, public relations, rehabilitation, hydrology, occupational therapy and prevention and body mechanics. This council provided funds to universities, backed by Bernard Baruch, to advance the field. The end of the war saw the emergence of the Archives of Physical Medicine and Rehabilitation, the premier journal in the field. During the war Physical Therapists and Rehabilitation Experts began to differ. Physical Therapists (Physiatrists) are people who have graduated medical school and coordinate services for people who are in need of rehabilitation. Rehabilitation specialists administered those treatments, focusing on speech, occupation, corrective and recreational therapy[1].

Millions of Americans are effected by traumatic brain injury (TBI), stroke or cerebral palsy (CP). These impairments result in restricted movement which makes everyday activities difficult. We developed a system with the intent of using augmented reality to rehabilitate patients with neurological and physical impairments.

Our current system has been designed to focus on the physical elements of therapy, especially occupational therapy. Our task from the beginning was to incorporate elements of everyday life into our apparatus, thus the concept of Augmented Reality games. The concept behind rehabilitation has been to help people assimilate into society despite injuries suffered in the workforce, playing sports or medical misfortune like strokes or seizures. With this in mind, we must design games to encourage users to replicate these actions.

These actions can range from the simple, like holding a ball or a cup, to the complex, like completing a puzzle. Our system must cover an entire spectrum of tasks, and prompt users to perform them at varying levels of difficulty. This can be done in a number of ways. The first is to program varying levels of difficulty into the game through time limits, speed of game play and range of motion variables. The second is by changing the object the patient must use. By having the patient use varying size and weight combinations we can increase the difficulty of the motions.
The other end of the problem is tracking the progress of a user and managing this data. Because the therapist is not there in the room with the patient, we need a way of chronicling and measuring the actions of the user. To do this we created a database that tracked users with a login system. This data can then be accessed by the patient and therapist to track the amount of progress that has been made over the course of treatment.

It should be noted that this system is not meant to take the place of a full rehab regimen. When designing this system we must take into consideration the patient's ability to cheat the system. When certain muscles become unavailable to our body it compensates by using muscles not originally intended for the desired movement. Proper rehab should avoid this scenario. We have designed this system to supplement visits to a rehab expert. In office visits are checkpoints to ensure the patient has been performing the exercises correctly.

Our system simply disguises these repetitive actions in the form of simple games. Asking a patient to extend their arm while holding a weight can be disguised as collecting coins. Instructing a patient to perform lateral movements can be masked by having a patient collect water droplets onscreen. In this way we are attempting to inject some fun into the process of rehabilitation.

Throughout the process we attempted to keep in mind the future of the project. Our initial attempts at games have been simplistic. Most of the games focus on collecting objects or repeating patterns in various planes of motion. We don't want our system to be limited to these actions. The homepage has been written to allow for the inclusion of new games down the road. The server has been configured for an adjustable number of users. The games need to be resizable for different window sizes. The release of Windows 8 requires us to design games that will work for both Windows 8 and Windows 7.

We also cannot be guaranteed the processing power of the host computer. The Leap Motion controller requires that the computer has an AMD Phenom II or Intel Core i3, along with 2GB of RAM. We have no guarantees that patients have a host computer that meets these specifications. As a result we must do everything we can to avoid stressing the computers resources. This means simplistic images, limiting the number of particles onscreen and developing efficient code.

Designing a good product requires keeping all of this in mind. To do that we needed an efficient division of labor and good communication to ensure all goals were being met. We used sites like drop box and Google drive to enhance the sharing of information. We had to keep access to these sites restricted to ensure no leakage of information.

Summary of Previous Work
There have been two previous iterations of this work. Both have focused on the integration of real world objects into a computerized system for the in home rehabilitation of patients. The benefits of this type of system are huge. It reduces the strain on a physical therapists schedule, increases the quality of care for low mobility patients and reduces the costs of care.

The first iteration used a webcam with object recognition to track the movement of objects across the plane of vision. The students were able to successfully design a system the allowed the user to use their arm to initiate game play. Armed with the knowledge that the idea had legitimacy to it, our advisor decided to take it mobile.

The second iteration of the project attempted to implement a similar system on mobile devices, such as a tablet. This ended up more difficult than originally thought. Sate of the art mobile systems did not have the processing power required to handle games of this nature. Having learned this, the project had to shift back to platform based computing.

The first few iterations set the stage for our project. We needed a system that could handle the load object recognition demanded and allowed for increased quality of game play. Systems considered had a background in gameplay, specifically gameplay with the user’s body as the controller.
Chapter II – Summary of Work

Section 1) Hardware

In order to fit the needs of upper limb rehabilitation we needed to investigate and select the best hardware that would fit the needs of our application. The selection of our hardware depended on accuracy, size, and ease of use. Our professor, Sudeep Pasricha, was kind enough to narrow down our options to either the Microsoft Kinect or the Leap Motion Controller. During the first few weeks of development we investigated the capabilities of each device and ultimately chose which device would best fit our needs.

Both the Microsoft Kinect and Leap Motion Controller make use of an infrared grid detection scheme to track bodily movement. The devices emit an infrared grid which is then detected by a series of infrared cameras. The data collected by the infrared cameras is sent through an image processing system which defines the locations of various parts of the human body. Although both of these devices make use of similar tracking technologies, they differ in several significant manners.

First, there may be a time when a patient must take the hardware home with them in order to fulfill their physical therapy outside of the facility. In this case, the smallest piece of hardware will be most desirable. The Microsoft Kinect measures roughly 11” wide, 3” deep, and 3” tall. This is considerably larger than the Leap Motion Controller which measures only 3” wide, 1.2 inches deep, and 0.5” tall. Given these measurements it is clear that the Leap Motion Controller is the most portable option.

Second, the ease of setup of each device must be considered. The Microsoft Kinect requires an external DC power supply along with a special USB cable. Along with these requirements, the Microsoft Kinect must be mounted in such a way that it has a view of the entire human body. However, the Leap Motion Controller requires only a single standard USB 3.0 cable to both power and provide communication from the device. It also only needs enough space to view a user’s hand. Again, given these facts the Leap Motion Controller is clearly the best device for our application.

Finally, the accuracy of each device must be considered in order to provide the best feedback for the patients. According to Gelman et al. the Microsoft Kinect has an average resolution of 29 mm\(^2\). However, according to developers, the Leap Motion Controller has an accuracy of \(1/100^{th}\) of a mm, or 10µm of resolution. Given this data it is obvious that the Leap Motion Controller is considerably more accurate than the Microsoft Kinect.

During our decision in choosing the best device for our application we considered several criterions. The first criteria we investigated was in regards to size, which we found that the Leap Motion Controller was significantly smaller than the Microsoft Kinect. The second criterion was in regards to ease of use, which we found the Leap Motion Controller to require less setup and space than the Microsoft Kinect. Our final criterion was accuracy, which we found the Leap Motion Controller to be significantly more accurate than the Microsoft Kinect. Considering all
three of these criterions it is easy to see that the Leap Motion Controller is the best choice for our application.

The next step in our investigation was to choose the best mounting position for the Leap Motion Controller. According to our advisor, the patients that will be making use of our device will need to rest their hands on a surface. This means that the user will be unable to hold their hand above the Leap Motion Controller which was the original use case for the device. In order to combat this we needed to design a simple solution that would allow the user to rest their hand on a table while still being able to be tracked by the Leap Motion Controller.

Mounting the device above a “playing area” proved to be the easiest portion of this task. The most complicated portion of this task was figuring out at what height the Leap Motion Controller needed to be mounted at and which surface would provide the best tracking capabilities. We began by simply taping the leap motion controller to a paint stir stick and taping the stir stick to various objects of varying height. Next, we used various surfaces for the playing field such as printer paper, different colored tables, and different colored gloves. During our investigation we found that with the Leap Motion Controller mounted roughly 12” to 18” above a black desk mat provided the most accurate and consistent performance.

The mount that was created was very rudimentary. It consisted of a vertical piece of wood that was cut to be about 18” tall and a horizontal piece of wood that was cut in such a way that would allow the Leap Motion Controller to be centered above the desk mat. Jacob had a friend design and 3D print a piece of mounting hardware that allowed the Leap Motion Controller to be easily attached to the horizontal beam of the mount and also allowed it to be adjusted in regards to depth in order to center the Leap Motion Controller above the playing area.
Section 2) Software Selection

We wanted our games to be portable between various types of platforms such as PC, Mac, and Linux. In order to meet this requirement the programming language needed to be interpretive. This means that the code does not need to be compiled before it can run, it is simply interpreted at runtime. This cuts down on the need to compile the code numerous times for various different platforms. Given this requirement we narrowed down our programming language options to either Python or Javascript. Our final decision was made when we considered the requirements of each language. In order for us to use Python we would need to ensure that the end user had a compatible version of Python installed on their system. Also, Python would require us to manually integrate the Leap Motion API and any other development libraries. On the other hand, Javascript does not require any special installation or dependency integration on the users end. Its only requirement is that the user has an internet connection and a web browser. Given these facts we chose Javascript as our primary game development language.

In order to communicate with the Leap Motion Controller through a web browser we needed to include the Leap.js library. The Leap.js library provides us with numerous functions to retrieve desired data from the leap motion controller in real time. Once communication was established with the Leap Motion Controller we set our focus on researching the numerous Javascript game development libraries. We restricted our search to libraries that were open source and had decent documentation. After some time we came across the CreateJS game development library suite. CreateJS has numerous libraries that allow us to draw and manipulate images in the playing area, add sounds and background music, and create smooth animations and transitions. Now that our development environment was chosen we were able to continue with other software design choices.

One of the major benefits of making our games web based is that we can host all of them on a central web server. This web server could host a site that would allow patients to log in to play the games, and allow the therapists to track the progress of their patients. In order to make a powerful and easy to maintain website we needed to choose a suitable backend framework. For this task we chose the web2py framework for its abilities to easily maintain a database and deploy new updates and games. Given our choice of framework we could then begin designing the database structure and the website layout.
Section 3) The Database

We wanted our system to be able to keep track of the various patient and therapist information. In order to store this information we created a *person* table in our database. The *person* table contains the following fields: *first_name*, *last_name*, *email*, *username*, and *password*. When creating the *person* table we put in various requirements to ensure that the email provided was valid and unique along with ensuring that the username chosen by each person was unique. This would ensure that a user could not accidentally log into an account of another user who had the same username or email address. Also, whenever a new entry is added to the *person* table a unique ID is generated for this entry. This further ensures uniqueness between entries in the *person* table.

Now that we are able to store information regarding a *person* we needed to be able to identify which *person*’s are patients and which are therapists. For this task we created a *patients* table and a *therapists* table. The *patients* table contains a *patient_id* field which takes the unique ID of a *person* that is to be the *patient*. It also contains a *therapist_id* field that takes the unique ID of a *therapist* who is assigned to the *patient*. The *therapists* table contains a single *therapist* ID field that is the unique ID of the *person* that is to be a *therapist*. In order to retrieve the *therapist* that is assigned to a *patient*, we simply query the *patients* table with the *patients* ID. However, in order to find all *patients* assigned to a *therapist* we must query the *patients* table with the ID of the *therapist*. This is known as a one-to-many relationship, i.e. one *therapist* to many *patients*.

The final task that our system needed to do was track the progress of each patient. To do this, we created a *games_played* table which contains the following fields: *player_id*, *game_title*, *start_time*, *end_time*, and *score*. Whenever a new entry is created in the *games_played* table, the unique ID of the *patient* is stored in the *player_id* field. Whenever a therapist needs to retrieve progress information pertaining to a particular patient, they would simply query the *games_played* table with the unique ID of the *patient*. 
Figure 1: Database model
Section 4) The Web Interface

In order to make our system easy to use we developed a clean looking and intuitive web interface. The web interface needed to be able to allow users to log in, register, launch and play games, review progress, and change account information. We began by creating a default layout view which consisted of nothing but the webpage background and our teams logo.

![Default layout view](image)

**Figure 2: Default layout view**

We then created the login view which consisted of login form and the proper styling for its container. The login view inherited the default view which results in the page below.
Next, we created the user registration view. This view consisted of a form that was dynamically generated depending on the various fields found in the person database table. By dynamically generating the form, we can ensure that the fields available to the user are always up to date. For example, if we were to add a birthday field to the person table, a birthday input would then be automatically rendered in the register form. The register form also inherits from the default view and is depicted below.

Once users were able to register and log in, we began developing the main web page. Our first task was to develop a main page view that would inherit from the default view. The main page
view consisted of the navigation tabs, user welcome message, and the copyright container in the bottom of the page. The main page view is presented below.

![Figure 5: Default View](image)

We needed to present the various games available to the user in an intuitive and clean manner. To do so we developed a sort of grid system that is similar to the ones found in many of the mobile device app markets. Also, the game options needed to be viewable on various sized displays. To meet this requirement we chose to implement a web design method known as “responsive web design”. This means that whenever the user resizes the browser window or views the webpage on different sized displays the webpage will “respond” by dynamically resizing, reordering, or restyling the various elements so that everything will fit in the viewing area and still look clean. For example, on a large display our grid would be three columns wide, but on smaller screens the grid would dynamically reduce to two or even one column wide. Our game options view inherited the main page view and is presented below when the browser window was resized to force a two column grid.
Whenever a user hovers over a game option, a ribbon will appear over the top of the game option. This ribbon contains the title of the game, an information button, and a launch game button. By clicking on the information button, an information window slides down from below the ribbon. In this information window we provide the user with a brief overview of the game such as what the goal of the game is, how it is played, and how points are awarded or revoked. Also, we inform the user of the type of motion that this game focuses on such as left and right, forward and back, up and down, or a combination of all three. This will allow the user to pick a game that will best fit their rehabilitation needs. Below we present the ribbon that appears above a game option and the information window that can be viewed.

**Figure 6: Main page view**
Once our game option interface was created we created the user profile view. This view would allow the user to modify their profile information such as their email address or password. Whenever a patient views their profile they are also presented with their therapists contact information which is not modifiable by the patient. This feature was added because we thought that there might be a time when a patient needed to contact their therapist, but had lost their contact information. When the user selects the profile tab, the person database is queried to retrieve the patients information which is then used to render the patient information form. The patients database is then queried in order to retrieve the unique ID of the patients therapist which is then
used to query the *person* database in order to retrieve the therapists information for the therapist information form. The profile view inherits from the main page view and is presented below.

![Profile view](image)

**Figure 9: Profile view**

The final view that needed to be created was the progress view. This view would present various gameplay information pertaining to the games that the patient has played. Currently this information is presented in a table, but we will be implementing software that would place this information in an interactive plot so that the patient/therapist can better visualize progress over time. When the user selects the progress tab, the *games_played* database table is queried with the unique ID of the *patient*. The data is then formatted and placed in a table for presentation. The progress view inherits from the main page view and is presented below.

![Progress view](image)

**Figure 10: Progress view**
Section 5) Game Engine

![Game Engine Flowchart](image)

**Figure 11: Game Engine**

All of our games are designed with a very similar control flow. Directly from startup it boots into a menu. From this menu the user can set parameters that define the game play. For example he/she can choose the pattern of the gameplay, the sensitivity of the system and the length of game play. The sensitivity determines how responsive the game is to hand movements. A higher sensitivity means that the on screen icon will move further in relation to a hand movement.

From this menu the games begins initialization. This state loads the required images and inserts them into game play. It also initializes arrays that are used for pattern generation. All patterns are currently hard coded into the process. Following this the system moves into game play mode. Game play is initiated in a continuous loop. Every iteration of the loop the location of the cursor is updated. The logic also checks for a collision between the cursor and some active element. If a collision occurs, the score is increased and the element is moved to a new location.
Upon completion of the game, the results need to be stored on the server. The depth of this report has yet to be determined.

Using this method any motion can be stimulated. All of our games figure to work in the X, Z or X and Z dimensions. By defining a uniform control flow we can control the behavior of games while leaving implementation decisions up to the designer. This was done in an effort to encourage creativity without hampering results.
Chapter III – Alternatives, Testing and Ethics

Alternative Approaches

There have been two previous iterations of this project that explored other solutions. The first project done by Jacob Poore and Baris Tevfik explored the use of Web Cams to implement augmented reality games. These games used 3D Graphics and Augmented Reality tracking libraries to achieve in home rehabilitation results. Overall they were successful, but Professor Pasricha wanted to explore mobile solutions\textsuperscript{[4]}. Alex Vlahinos, Gregorio Campuzano, and Nathaniel Olsen explored implementing these games on an iPad. While successful, the iPad lacks the processing power to perform complex object recognition, severely limiting the games that could be implemented.

As a result of these findings, the decision was made to shift back to personal computer based systems. Because webcams had already been explored, this project chose to explore the use of hardware built with gaming in mind. The alternative to the Leap Motion Controller was the Microsoft Kinect. The Microsoft Kinect is used in conjunction with the xBox 360 to play user interactive games like Kinect Sports or Dance Central. Playing games with the Microsoft Kinect on your computer is a much more difficult process. It requires the installation of Microsoft Visual as well as the requisite drivers, a task we deemed too complex for a patient.

We also considered using alternate languages for developing our games. When still considering the Microsoft Connect languages like C++ and C# were considered. The Microsoft Kinect was optimized for these languages, especially C#. Once the decision was made to abandon the Microsoft Kinect it re-opened other programming languages. We did not explore other development routes for the implementation of our database architecture. We went directly to the HTML and web2py framework.

Testing

The extent of our testing so far has been to find feasible solutions to implement into our system framework. We tested our hardware to decide between the Microsoft Kinect and Leap Motion Controller. Once we decided on the Leap Motion Controller we tested the capabilities and accuracy of the Leap Motion Controller. Following this, we needed to test the Leap Motion Controller to ensure the incorporation of real world object would not throw off the sensor. We also needed to test that environmental variables would not throw off the sensor.

Testing to decide between the Leap Motion Controller and Microsoft Kinect consisted of exploring the capabilities of both. We played many of the available development library games from Kinect Development Library and modified the code to get a feel for the ease of programming the device. The Leap Motion Controller is a very new device however and barely had a development community. The testing of this device included playing some of the available apps and programming a few of our own games with the available Leap API.
Testing of the Leap Motion Controllers range was rather simple. We needed to guarantee it provided sufficient area for the user to move their hand. According to the Leap Website the device has around 8 ft$^3$ of detection area. For reasons mentioned earlier the Leap Motion Controller needed to be top mounted. We adjusted the height at which it was mounted and found that 12” above the users hand provided the most consistent data. Furthermore, the user still experienced just over two feet of motion available to them.

Following this discovery we needed to categorize the objects that could be used with the Leap Motion Controller. Extensive testing has not been completed on this, but we know that objects too tall interfere with the Leap’s sensors. Color does not have an impact, but translucency does seem to. Solid objects are better for our purposes. Room lighting has no effect, and neither does the skin color of the patient.

**Ethical Concerns**

Our biggest concern with this project is copyright infringement. We are leveraging many aspects of our project from previous work or examples. We must be careful not to steal anyone’s work, as it would put the University at risk. For example, the three of us are not graphic designers. As a result, the three of us have built the theme of our games using images from the internet. These images must be open source images, and if we cannot find open source images to suit our needs we must construct images of our own.

Furthermore, we must be careful to avoid copying code. Any examples we encounter on the internet need to be created with the intent of teaching, and cannot be copyright material. Again, using this code will put us and the University at risk.
Chapter IV - Conclusion

All features discussed in this report have been implemented and demonstrated to work. The most important feature of our system is its adaptability. New games can be developed efficiently to isolate new motions, incorporate new rehabilitation techniques and improve the quality of the system. Our database has the capability to handle varying numbers of user and the data demands associated with each user.

The Leap recognizes motion in three dimensions, giving us the capability to build a variety of games. Games can be built to isolate motion in any combination of these directions. For our purposes we focused on stimulating motion in the X and Z dimension. We reasoned that most user would struggle holding their hand in midair, and gave them the benefit of a table to rest their hands on. This eliminates up and down motion but increases the feasibility of our system.

Having proved the functionality of our system, the remaining work lies in streamlining it. In no particular order, the games need to be standardized to meet the following requirements. All of the games need a game menu to configure the games to a user's needs. All games too to be adjustable to the user's windows size. Upon adjustment of the window the game needs to respond accordingly.

The database coordinates information passing between the therapist and his patients. Therapists can control the games available to a patient, the required game time and the difficulty. Furthermore, patient progression data is only available to the patient’s therapist and the patient himself. The database can have any number of patient to therapist groupings and therapist sub groups. This organization is an effort to keep patient confidentiality between the patient and their therapist.

Our main plans for next semester revolve around clinical testing. Possible design decisions such as the therapist/user interface as well as slight gameplay modifications will be made based on feedback from these sessions. Dr. Sudeep Pasricha has informed us that he will take care of setting up the appointment and has instructed our group to have our games ready for testing by the end of January. It is possible that we will be required to create another game in the future to expand our collection of games.

Improvements to current games include adding more encouragement, graphical enhancements, and progressive difficulty. Depending on feedback from therapists, one possibility is to have our games automatically get harder mid-game or have the difficulty increase from game to game on startup if the user is doing well. If this is not something that is not desirable we can set up the server so that the therapist and possibly even the user will be able to set the difficulty of each game manually before gameplay starts. Future graphical enhancements include such things as adding elements to the background or making our menu screens more aesthetically pleasing.

Another major challenge our group faces next semester is implementing a way to quantify the results of each game and standardizing the way the results are presented. Although each game potentially emphasizes different elements of rehab we want to be able to be able to present the results for the different games in a way that is easy to understand and is useful for the therapist to
track results over time. The main results that can be taken from our games are accuracy and time spent. To do this we must somewhat reconstruct our games so that it is not random and that different patterns are available. Although we can still keep randomly generated patterns for leisurely purposes we must implement patterns into each game so that results will be comparable over time and previous iterations. One thing that is still to be implemented is to take the data that we find and represent it graphically.

One major task that we will be assigned for next semester is to further investigate which objects are best suited for the Leap Motion Controller. We are to build a collection of objects to be held by the participant in clinical testing and in project demonstrations. These objects will consist of a variety of shapes, sizes, and weights. This helps simulate holding everyday objects and enhances the benefit of the rehab. In addition to building and collecting a set of useable objects we are to construct a list of objects that do not work with the Leap Motion Controller and our current project set-up. Each game must be tested individually as to insure that each object can be used with each game. Once we have a set of objects and connected them to each game we are to implement a graphical indication on the selection screen that shows the user which objects are useable with each game.

A possible hurdle for our team to overcome is being able to gather CPU information so we can calibrate each game on the specifications provided. For example, if we can detect the computer has low grade specs we could possibly activate a different version of the game with less intensive graphics. This is important because of the fact we our game is so portable and it set up so that it can possibly be played from any computer we want it to work if the patient does not have access to high end computing. Finding these specs can be a challenge because our current research has found that currently there is no way to find the information desired using basic HTML and JavaScript that our games are constructed in. We are hopeful that future Leap Motion upgrades will be able to present CPU specs or that we can find other alternatives. One possible alternative is having the user manually select configurations or to implement different software that can give us the info, possibly by asking for user permission and using a system call.
REFERENCES


APPENDICES

- Appendix A

Augmented Reality - AR
Cerebral Palsy - CP
Graphic User Interface – GUI
Traumatic Brain Injury - TBI

- Appendix B

The cost of our project has been minimal. Using open source software and Academic Software downloads available through sites like Dreamspark we have been able to save most of our money. The other costs incurred by our project have come out of team members pockets. Below is a summary of available funds and purchases made with them. Currently our only future plans are for a large display device to increase the visibility of our project. We envision a setup with a viewing device moved back from the user, the laptop off to the side (its only purpose is processing now) and only the Leap Motion Controller in the immediate area.

<table>
<thead>
<tr>
<th>Cost</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beginning of the Project Total</td>
<td>$374.73</td>
</tr>
<tr>
<td>Josh Leap Motion (10/03)</td>
<td>-$90</td>
</tr>
<tr>
<td>Updated Total</td>
<td>$284.73</td>
</tr>
</tbody>
</table>

**Table 1: Team Expenditures**

Costs incurred to individual team members include two Leap Motion Controllers and parts for the demo mount. These Leap Motion Controllers were purchased with the intent of keeping them once the project had concluded or been handed off. As such, it was deemed improper to use projects funds for them.

<table>
<thead>
<tr>
<th>Cost</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ryan’s Leap Motion</td>
<td>-$90.00</td>
</tr>
<tr>
<td>Jake’s Leap Motion</td>
<td>-$90.00</td>
</tr>
<tr>
<td>Jake Mount Hardware</td>
<td>-$10.00</td>
</tr>
</tbody>
</table>

**Table 2: Personal Expenditures**

The only aspect of our project that has any effect on shipping manufacturing costs is the Leap Motion Controller mount. While it is not necessary to play any of the games, it assists people with an impairment of the upper body by allowing them to rest their arm on a table. The solution
we currently have is crude and meant to demonstrate the concept. We envision future development costs to create a better solution.

This is not our area of expertise however, and we have not realistic estimates of the cost for labor, shop, mechanical hours and parts. We can tell you our current solution costs under ten dollars and takes around half an hour to assemble. Shipping our current solution would cost $13 under a USPS flat rate shipping system\[^3\]. There are no heat sensitivity concerns to ship our current mount.

- Appendix C

Our project was difficult to categorize as a breakdown of individual tasks. As a result we created a very general description of milestones we hoped to hit. In hindsight we wish we would have come up with a better way to illustrate the progress we had made. This outline would be very good from a corporate perspective, it is difficult to track microscopic success from a teap perspective.

Revised September 30th

**Phase 1** – Investigate capabilities of Hardware - Delivery date: September 20\(^{th}\)

Leap Motion Controller – Jacob Thornton / Ryan Nash:

Investigate the various positions that the leap motion can be placed in and how well the device can track the user. We will test the Leap Motion by placing it above, below, and to the sides of the viewing area. We will also investigate how accurate the device is at tracking a user’s movement. We will also investigate the type of data that is being gathered by the Leap Motion Controller and how to best use this data to create games for physical rehabilitation exercises.

Kinect – Joshua Wallace / Ryan Nash:

Investigate how accurately the device can track user movement, skeletal matching, speech integration, and object manipulation. Research previous projects that have used the Kinect as a human interface device on a PC and apply what we learn to our programs.

**Phase 2** – Simple Game Development – Delivery date: November 30\(^{th}\) – All team members

Simple games will be developed on both pieces of hardware in order to gain knowledge of how each piece of hardware works and how to best write software for each.

**Jacob Thornton**

Jacob Thornton will focus on the Leap Motion Controller. Games that will be created by Jacob will include brick breaker, follow the leader, and car racing. These games will focus on fine grain movement of the arms and hands in the X, Y, and Z axis. Jacob will be writing software in Python, Javascript, and Java.

October 11\(^{th}\) – Have rough design of graphics for games created
October 25th – Have general structure of games implemented and at least one game ready for testing.
November 6th – Have at least two games ready for testing. One of which has already been revised.
November 20th – Have all three games completed and in testing. Revisions applied to at least two games.
November 30th – All three games completed and revised.

**Joshua Wallace**

Joshua Wallace will focus on the Kinect. Games that will be created by Joshua will include pop the balloon and Simon says. These games will focus on a broader range of movement as opposed to the leap motion controller. Movement will be primarily on the X and Y axis. Joshua will be writing software in C# and possibly C++ if the need arises.

October 11th – detailed outline and layout of Simon says and balloon game including list of features, objectives and movements.
October 25th – Basic functionality including movement tracking and general game structure.

**Ryan Nash**

Ryan Nash will work on both the Kinect and Leap Motion Controller. Ryan will work primarily on the leap, but assist Josh where applicable. Ryan will focus on games target toward simulating real world actions taken for granted by most people. Ryan will be programming software for the Kinect in C# and software for the Leap Motion Controller in Javascript.

October 2nd – Meet with Prof. Pasricha to brainstorm ides for action recreation games.
October 11th – Execution plan delivered.
October 25th – Game engine implemented.
November 6th – Games finalized and reviewed by team.
November 13th – Critiques implemented and shown to Prof. Pasricha.
November 30th – Investigate new possibilities for recreation in leap and determine course of action.

**Phase 3 – Refinement – Delivery Date: December 19th**

All team members

After some simple games have been created we will re-work them in order to better suit the ultimate goal of the game. This may include enhancing graphics, enhancing response time, scoring, web access, menus, voice recognition, and sounds. New revisions may be thought of as we progress through the project and gather results and feedback from clinical trials.

**Semester Two**
Phase 4 – Clinical Trials – Delivery Date: Pending Meeting With Sudeep – All team members

Games that have been developed during phase 2 will be tested with patients. Feedback will be gathered and will be used to further improve the capabilities and functionalities of the games.

Course of action with be determined by outcome of these trials.

ACKNOWLEDGMENTS

We would like to thank Sudeep Pasricha for his assistance in the development of our project. His support has been invaluable in developing the Augmented Reality Games system.