

Virtual Reality Games and Physical Therapy

Mike Ridgeway¹

Advisors: Jacob Rosen², Ji Ma²

¹San Jose State University, ²University of California Santa Cruz
mridgeway24@yahoo.com

Abstract— Previously, Virtual Reality Games were implemented with an Upper Limb Exoskeleton (Bionics Lab at UC Santa Cruz) for Stroke patients to perform Physical Therapy. The previously developed Virtual Reality games were based on Microsoft Robotics Studio with the C# programming language. In this project, the previous Flower Game along with the new Whack-a-Ball game are implemented with the CHAI-3D platform. The functionality of the new Flower and Whack-a-Ball games has been programmed with C++ using the CHAI-3D libraries.

Index Terms—Computer Haptics and Active Interfaces 3D (CHAI-3D)

I. INTRODUCTION

Robotics can be used for a wide variety of things including as assistive devices for physical therapy. At the University of California Santa Cruz Bionics Lab, a dual Upper-Limb exoskeleton was previously used with virtual reality games for stroke patients to perform physical therapy. This leads to the summer research project which involved converting the original virtual reality games that were programmed using Microsoft's Robotics Studio and the C# programming language to another platform. The reason for switching to a new platform was due to the original programs running into issues along with the ease of implementation. The goal was to transfer the original eight games using the CHAI-3D libraries which are for creating virtual reality environments with haptic interaction. The CHAI-3D libraries made implementing the games easier and possible on multiple operating systems. The eight games were distributed among six undergraduate students and the focus of this paper is on the Flower game, which was the original game to be transferred from C# to C++ using the CHAI-3D libraries. A new game was also developed called Whack-a-Ball.

II. DESCRIPTION OF THE VIRTUAL REALITY GAMES

A. The Flower Game

The Flower Game consisted of nine stages, with five balls to touch per stage. The balls were connected by lines. The balls were distributed using 45 degree gaps between placements of line of which the ball was on while following the outline of a sphere as in Figure 1. The center ball would

be placed at the origin of the sphere and have two balls on either side of the ball located at the origin. The user would touch the balls and hit a reset ball to get the next stage or push the equal sign “=” on the keyboard to skip to the next stage of the flower. The score would then be displayed using the local score and overall score. The local score referred to the number of balls touched during that stage with a maximum of five balls. The overall score referred to the amount of balls touched in general, which was a sum of the various local scores. Figure 2 shows the actual Flower Game to be played.

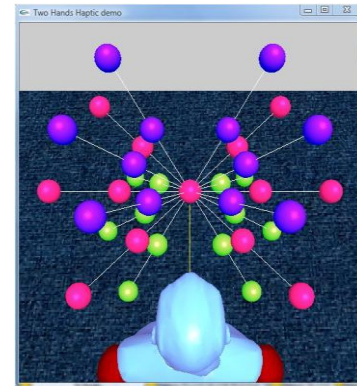


Figure 1. A screen shot of the ball positions in the Flower Game showing the placement of the balls following the outline of a sphere. Not the actual Flower Game to be played.

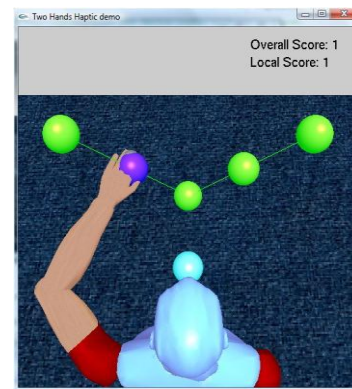


Figure 2. A screen shot of the actual Flower Game showing the placement of the balls for one stage with the

user touching a ball and the overall and local scores displayed in the top right of the figure.

B. The Whack-a-Ball Game

The Whack-a-Ball Game was based on the Whack-a-Mole concept where the user has to touch something that appears semi-randomly. In this case the first screen showed the user all the possible ball positions and then once the user touched the ball and then touched the reset sphere or hit the equal sign “=” on the keyboard, the new position would be displayed for the user to touch. The ball positions were determined using the rand() function call. The ball position could also end up in the same position multiple times. Time could be added as another variable for the user to compete with. If the user did not touch the ball within X amount of seconds then the next position would appear and the user would not get a point added to their score. The overall and local scores were displayed at the top right of the window along with the time. The time would reset to zero at the start of a new position and then increment from there.

Two versions of the Whack-a-Ball Game were made. One had two paths of balls, one for each arm (Figure 3), and the other had one path of balls for both arms (Figure 4).

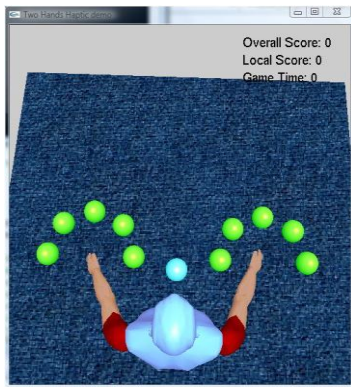


Figure 3. A screen shot the Whack-a-Ball game with two paths of balls, one for each arm.

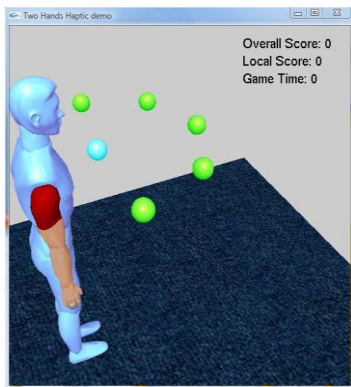


Figure 4. A screen shot the Whack-a-Ball game with one path of balls for both arms.

III. TOOLS AND IMPLEMENTATION

A. Microsoft Visual Studio 2008 C++ Express

Used to program and debug the virtual reality games. The CHAI-3D download came with pre-built projects that could be opened in this version of Microsoft Visual Studio along with the 2003 and 2005 version.

B. Phantom Omni Haptic Device

The Phantom Omni Haptic device has 6 degrees of freedom whereas the dual Upper-Limb exoskeleton has 7 degrees of freedom. The Phantom Omni was used to test the properties of the objects, such as the balls, and the functionality of the games. The Phantom omni was a tool to test that objects were behaving as programmed. The Phantom Omni can be thought of a three-dimensional mouse that provides the user feedback through the user held pen.

C. CHAI-3D Libraries

CHAI-3D was used because it is open source, has embedded haptic interaction, and supported on Linux, Windows, and MAC operating systems. The libraries provided allowed the user to select or design objects with various features and appearances. The user could import object files to display objects designed using other software. Objects could have textures applied from bitmap image files. CHAI-3D also has built in classes that allow users to connect to haptic devices.

IV. CONCLUSION

The functionality of the Flower and Whack-a-Ball games has been programmed. The games were tested using the Phantom Omni Haptic Device but have yet to be tested using the dual Upper-Limb exoskeleton. The goal is to eventually test the games using the dual Upper-Limb exoskeleton. The communication protocol between the exoskeleton and games has yet to be developed which would allow the two to interact. There are features for the games that can be added to produce an easier to use more enjoyable game. Refining needs to be done with user friendliness, data collection, and object features. User friendliness is referring to the ease with which the features of the game can be changed and the ease of use for the person playing the game. User friendliness can include the ease with which parameters are adjusted like the position of the balls in the Whack-a-Ball game and the location of the origin ball in the Flower Game. The original games collected data from user interaction with the games and that has yet to be implemented with the new games. Object features include changing the way users feel objects and how the objects are displayed.

V. FUTURE WORK

Develop a Whack-a-Ball game using deformable objects to control how a user interacts with the objects and potentially control the amount of force a user is required to apply to an object in order to move it. A potential start to this version of the Whack-a-Ball game is displayed in Figure 5.

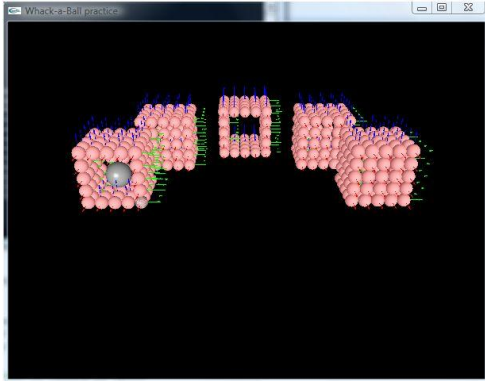


Figure 5. A screen of the Whack-a-Ball game using deformable objects to create cubes around the center ball to be touched.

ACKNOWLEDGEMENT

Professor Rosen for providing the interesting project and lab equipment. Ji Ma for setting up the template for the games and all the help with debugging. The SURF-IT program for providing the opportunity and experience. All the students working on the games for their help.

REFERENCES

- [1] Conti, F., Barbagli, F., Morris, D., Sewell, C., & Grange, S. (2009). CHAI 3D (Version 2.0.0) [Software]. Available from <http://www.chai3d.org/>