A WebGL VR Implementation of an Obstacle Avoider Game

Anirudh Patel
and David LaFehr
Department of Electrical Engineering
Stanford University
Stanford, CA 94305

I. Game Description

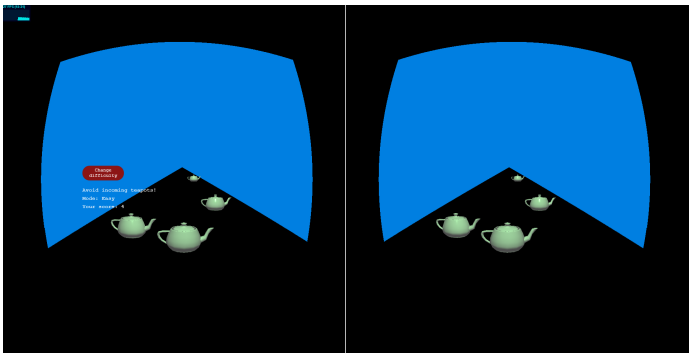

Fig. 1. The stereoscopic rendering of our game.

For our project, we created an interactive virtual environment in which the user plays a game. In the game, the user controls a teapot that sits on a road that seemingly extends from just below the user to the far distance. The user can move the teapot left and right for a total of three positions.

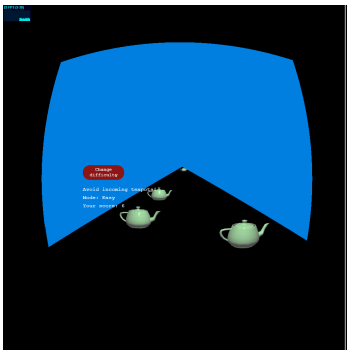

Fig. 2. The teapot that the player controls is in the right lane.

Other teapots appear at the far end of the road and travel along the road, eventually reaching the user. If there is contact between any of these teapots and the user's teapot, the game ends.

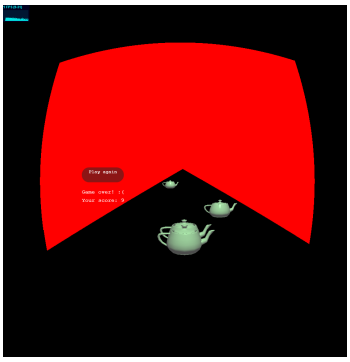

Fig. 3. The standard "Game Over" screen.

When a teapot reaches a position closer to the user than the user's teapot, the user's teapot is considered to have avoided this teapot, which subsequently disappears. The user is awarded a point upon collision avoidance. The goal of the game is to score as many points as possible.

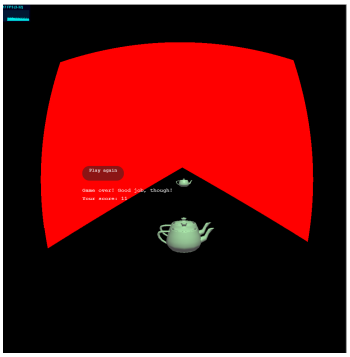

Fig. 4. A special "Game Over" screen for when the player achieves more than ten points.

If the user finds the game too easy at first, the user may increase the difficulty level. When the difficulty level increases, the teapots move more quickly toward the user. In the advanced difficulty mode, while it is more challenging to avoid the incoming teapots and thus easier to be involved in a collision, the user can score more points in a given time period.

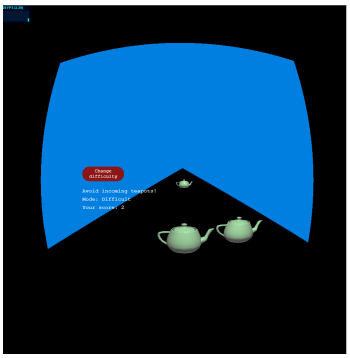

Fig. 5. When on difficult, the text next to "Mode: " changes to "Difficult."

II. Sensing

Both the gyroscope and the accelerometer were used in this application. Angular velocity measurements were used to detect head roll. On the Arduino side, the code from Homework 5 to generate flatland roll was executed to stream z-axis angular position data. In the JavaScript code, each datum was added to a running sum. When the sum exceeded $\pm$ 20 degrees, the user's head was adjudged to be far enough to the left/right that the user intended to move a teapot to the left/right. The x position of the teapot was updated accordingly, and a one-second timer was set to ensure that there would not be an instantaneous consecutive move in the same direction. The assumption here was that the user would move their head back to its natural position within one second, and this assumption was deemed valid through testing and at the public showcase.

Acceleration data were used to detect head pitch. In the Arduino code, the acceleration measurements in the z-direction were streamed. On the JavaScript side, when the acceleration dipped below -5 $\mathrm{m/s^2}$, the user's head was adjudged to be far enough forward that a change in difficulty was being requested. The difficulty level was toggled, and a one-second timer was set to ensure that there would not be a multitude of toggles while the user returned their head to its natural position. The same assumption as above applies here, and it was deemed valid for the same reasons. Note that an acceleration of -5 m/s^2 generally was not due to an actual acceleration of the user's head; instead, it was most often a consequence of the accelerometer being pitched to the point that its z axis registered a relatively large component of acceleration due to gravity.

III. Future Work

Going forward, we hope to fully utilize the VR capabilities of the game. We would like to incorporate interactivity with an HTC Vive Lighthouse such that the user could move their own teapot up or down along the road to a position that would better facilitate a lane change. We were also considering using models other than teapots for the project, but decided to focus on the game aspect rather than the aesthetic appeal. Given more time, we would like to use different objects such as a car for the user and put obstacles like trees or pedestrians in the way. Another game mode could be developed in which there are more than three lanes. Further, a problem that we noticed with our random teapot generator is that it sometimes repeatedly spawns teapots in the same lane. However, this is suboptimal, as the player can sometimes remain in the same lane for several seconds before being forced to move. With that in mind, another interesting feature that we would have liked to implement would be a heuristic that makes the teapots spawn in relation to making the player's life uncomfortable. The higher the difficulty, the more challenging we would want to make the heuristic, as opposed to just changing the speed of incoming traffic as we do now.