CS 680

Computer Graphics

Air Hockey User and Technical Manual

Author:
Joshua Gleason Marvin Smith

November 23, 2011

Air Hockey Overview

Opinions

Time Spent

Extra Credit

High Scores Window accessible from either the menu or by pressing '8' key

User Manual

Introduction

This manual will introduce you to the basic functions of the AirHockey game. Figure 1 shows the basic layout of the game.

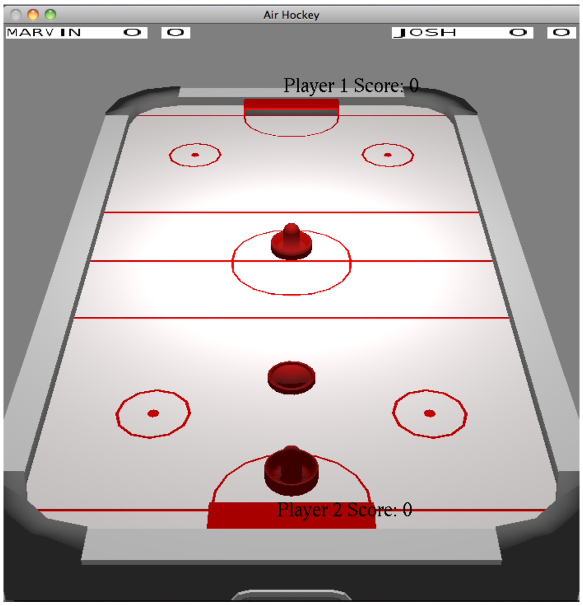
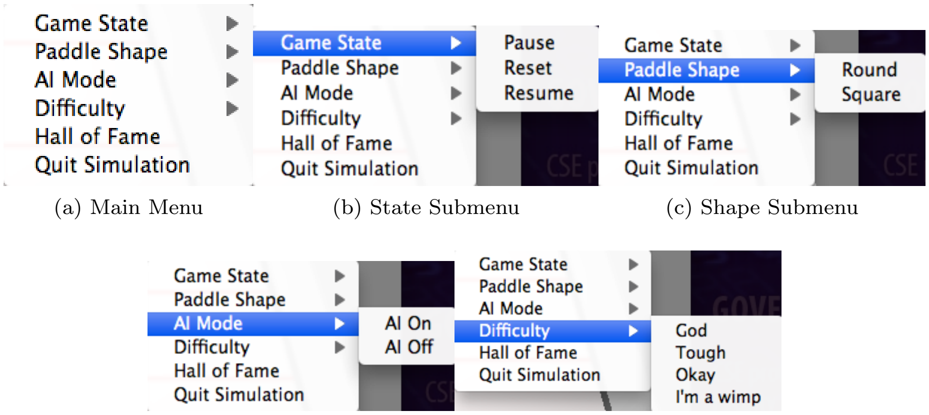


Figure 1: Basic Layout of Air Hockey

Throughout gameplay, the user will have access to game state information through the Heads Up Display (HUD). This is the bar at the top of the screen. The player names may be set in the data/options.cfg file under the PLAYER#_NAME options. The first number to the right of the names is the current score for that particular game. The second number is the number of player wins in the sequence. If the player wins enough games, the console will ask the user for their tag at the end to add you in the Hall of Fame. This is described in more detail below. Figure 1 shows the HUD.

User Menu

There are many different features in Air Hockey. To get to all of them, you will need either the keyboard or the menu. To access the menu, click on the screen with the right-mouse button. Figure 2 shows the basic layout of the menu.



(a) Main Menu (b) State Submenu (c) Shape Submenu

(d) AI Submenu (e) Difficulty Submenu

Figure 2: Basic Menu Layout of Air Hockey

Menu Options

Game State Affects the gameplay action.

- **Pause Game** Halts game progress until resumed.
- **Reset Game** Clears the player score and starts the game over. Will keep player wins however.
- **Resume Game** Will un-pause the game, allowing the game to step.

Paddle Shape Affects the shape of the paddle.

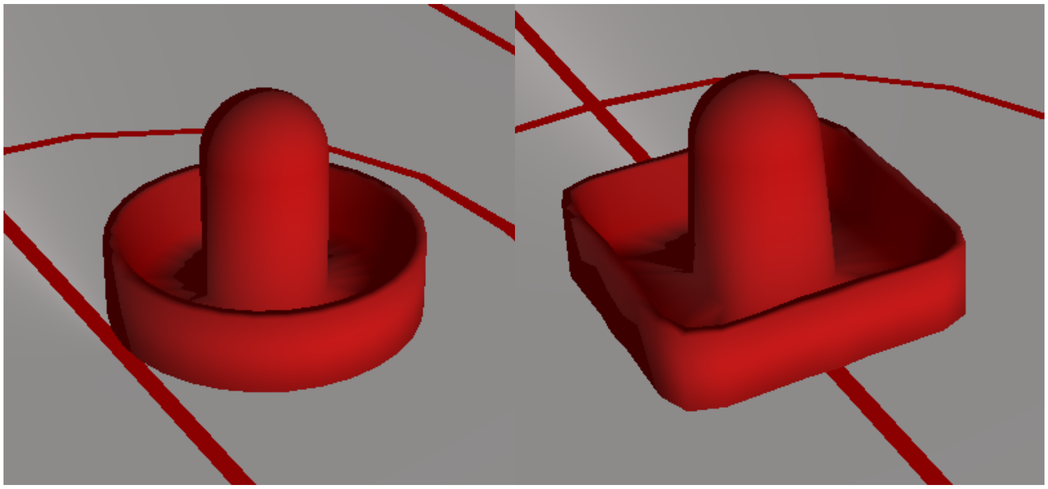
- **Round** Round Paddle. See Figure 3.
- **Square** Square Paddle. See FIgure 3.

AI Mode Toggles between single and double player modes.

Difficulty Determines the AI Skill Level

- **I'm a wimp** Don't bother, you already won.
- **Okay** Default. The AI shows up to play.
- **Tough** You had better come ready for a fight.
- **God** If you can get the puck past the AI, you are better than me.

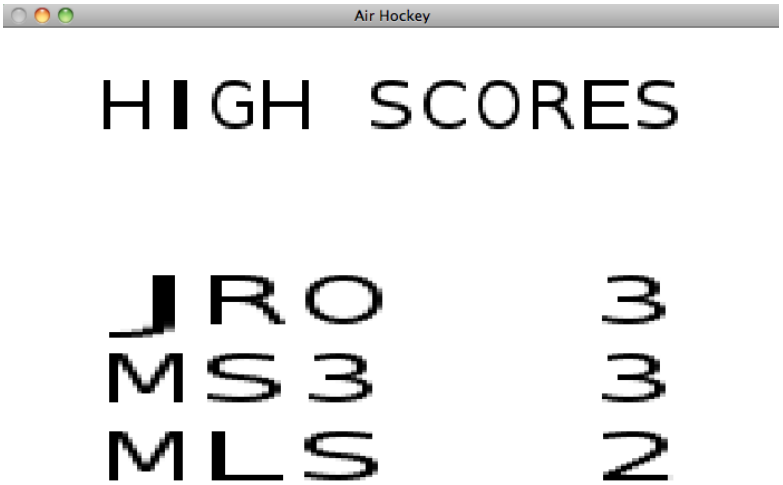
Hall of Fame Show the top performers.



(a) Round Paddle

(b) Square Paddle

Figure 3: Paddle Shapes



```
HIGH SCORES

JRO      3
MS3      3
MLS      2
```

Figure 4: Hall of Fame Screen (Shows number of wins)

Quit Simulation Exit the program.

Keyboard Map

Key	**Function**		**Key**	**Function**
8	Show High Score		w	Move Camera Forward
			s	Move Camera Backward
←	Player 2 move paddle left		a	Shift Camera Left
→	Player 2 move paddle right		d	Shift Camera Right
↑	Player 2 move paddle up		q	Shift Camera Upwards
↓	Player 2 move paddle down		e	Shift Camera Downwards
			i	Rotate Camera Eye Upwards
m	Translate Light +Z axis		k	Rotate Camera Eye Downwards
n	Translate Light -Z axis		j	Rotate Camera Eye Left
.	Translate Light +X axis		l	Rotate Camera Eye Right
,	Translate Light -X axis			

Table 1: Keyboard Map of Air Hockey

Camera

The camera in our Air Hockey allows for complete motion and will enable the user to have any view of the game that they wish. The camera controls are outlined in Figure 1. Figure 5 shows various views of the camera.

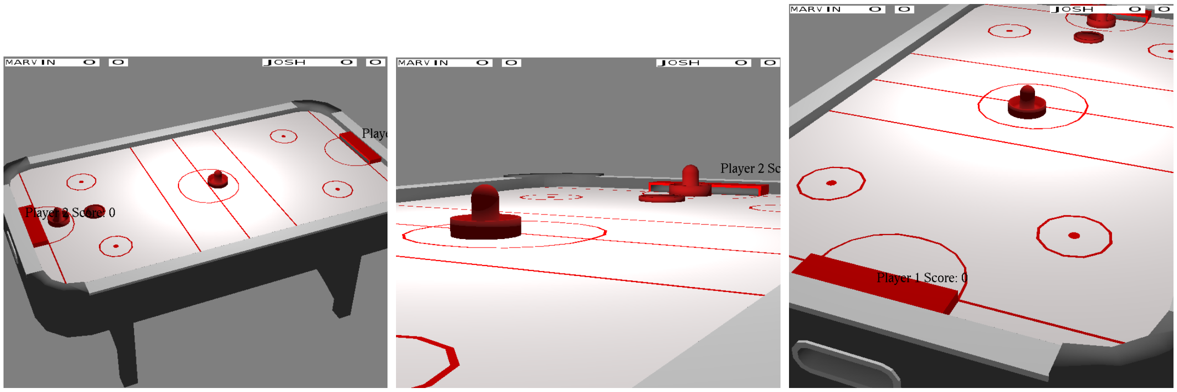
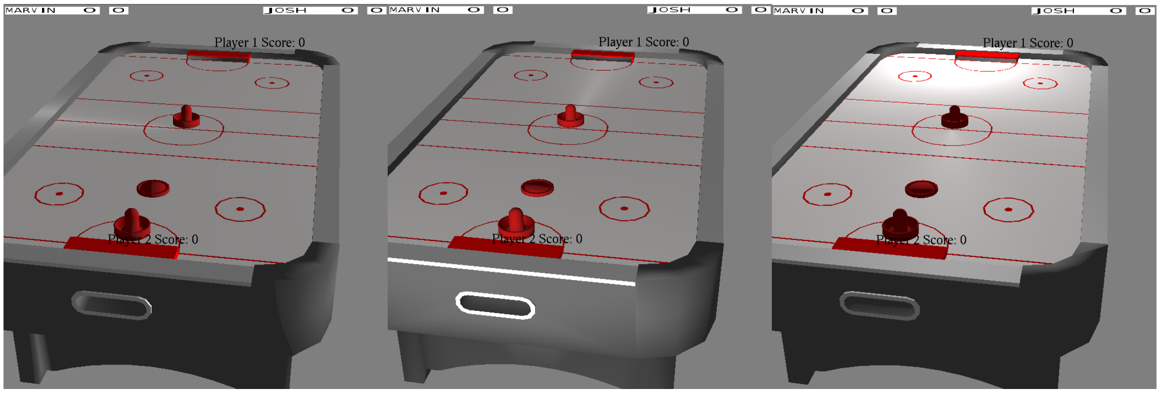


Figure 5: Camera Views Possible

Light Model

The light model in Air Hockey allows for X and Z axis *translation*. This will allow the user to see effects of light and shadows. Figure 6 shows various light positions of the same scene.



(a) Light Source Right of Camera (b) Light Source Behind Camera (c) Light Source Away from Camera

Figure 6: Light Model Examples

Technical Manual

Design Decisions

This project required some new and very interesting design decisions with respect to the last project.

Model Loader We incorporated a model loader from Kixor to load and display models. We had to significantly modify it, however it works and we were able to create a generic object loader which will work for every type of model we use.

Text Engine We built our own tool for loading text into our opengl program. We load an image with every character and then use indeces to fix the locations of each letter. Then, given a string, we create a list of texture locations and vertex locations which will allow OpenGL to render the text. This required a thorough understanding of textures and was a great learning experience.

Physics Engine We used the Bullet physics engine to handle the collision detection and response. This took an enormous amount of time, however it creates a very good physics environment. Our paddles respond accordingly to any shape and respond well to friction and ”bounciness”.

Deficiencies

This project meets all requirements and does not have any major known bugs.

Areas for Improvement

We had issues getting arbitrary shaped paddles to work, which was annoying. This is something we would like to fix so that given any object, we can use it as a paddle. Another area for improvement is our font engine. We used only capital letters as well as the numbers. I would like to pick a larger image to have higher resolution fonts as well as cover all ASCII characters.