Air Hockey

Darian Bhathena

June 2, 2016

AP CS, Spring Term

Dr. Miles

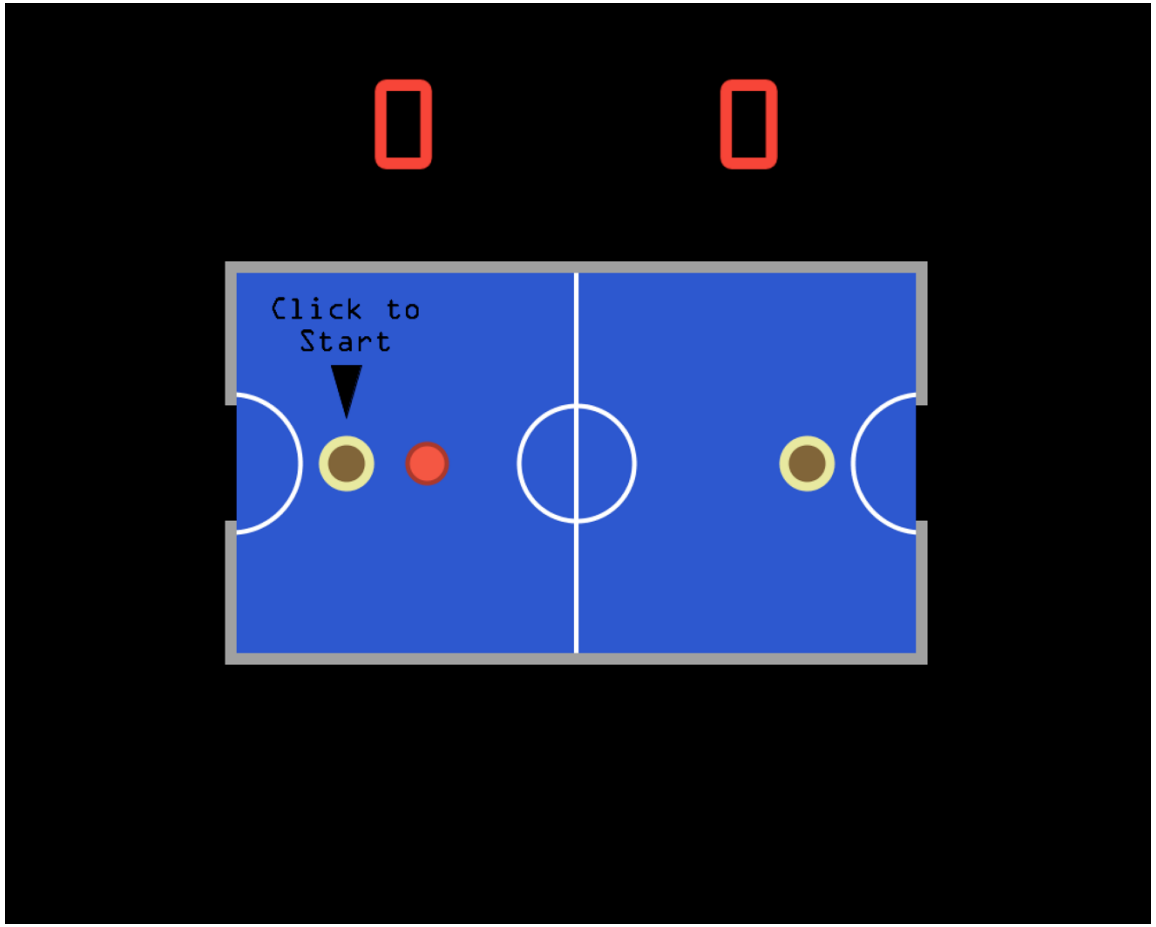
Screenshots:

New Game:

This is the first screen the user encounters when a game is initiated. The buttons are interactive (hover-sensitive)

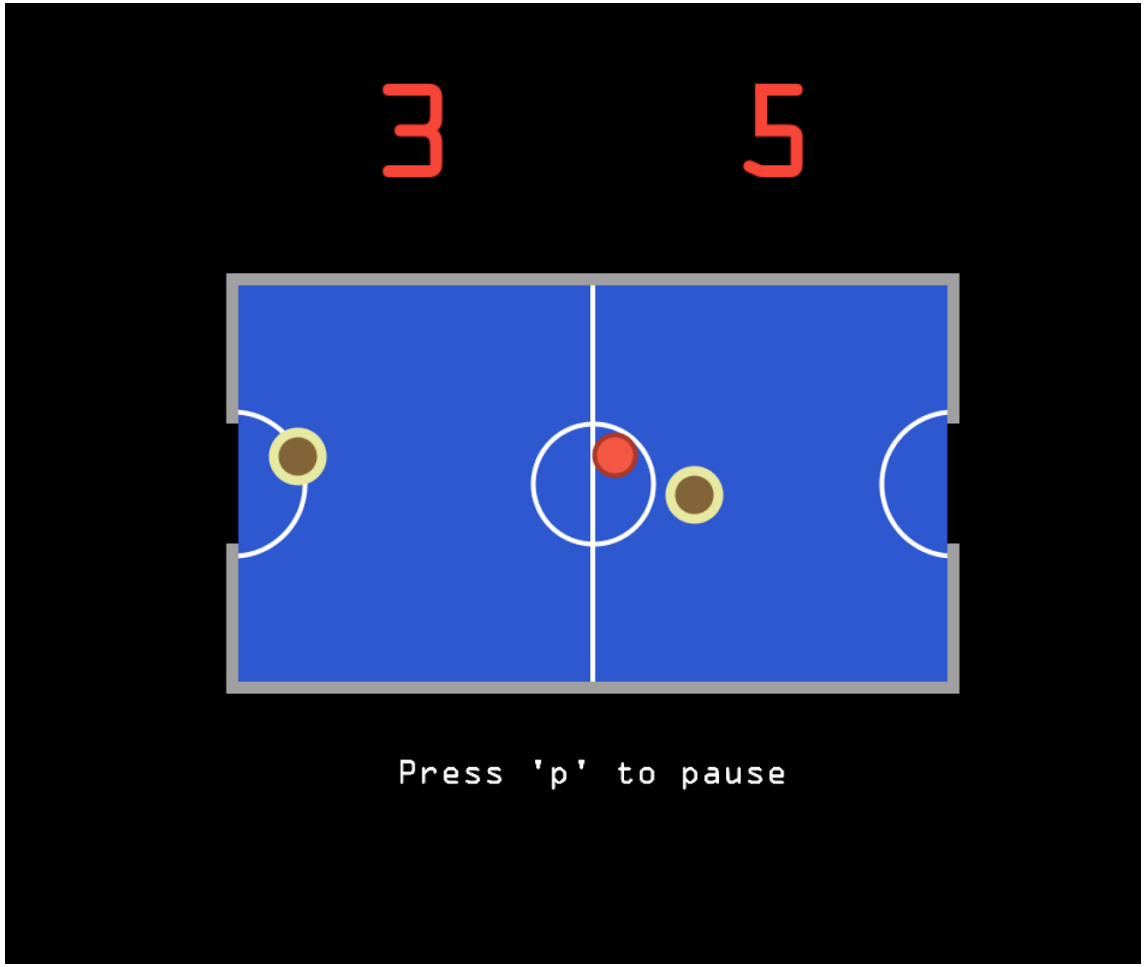
Choose Level:

If the player clicks “New Game” on the previous screen, they will be directed to this screen, where they must choose on what level of difficulty they would like to play. Again, the buttons are hover-sensitive.



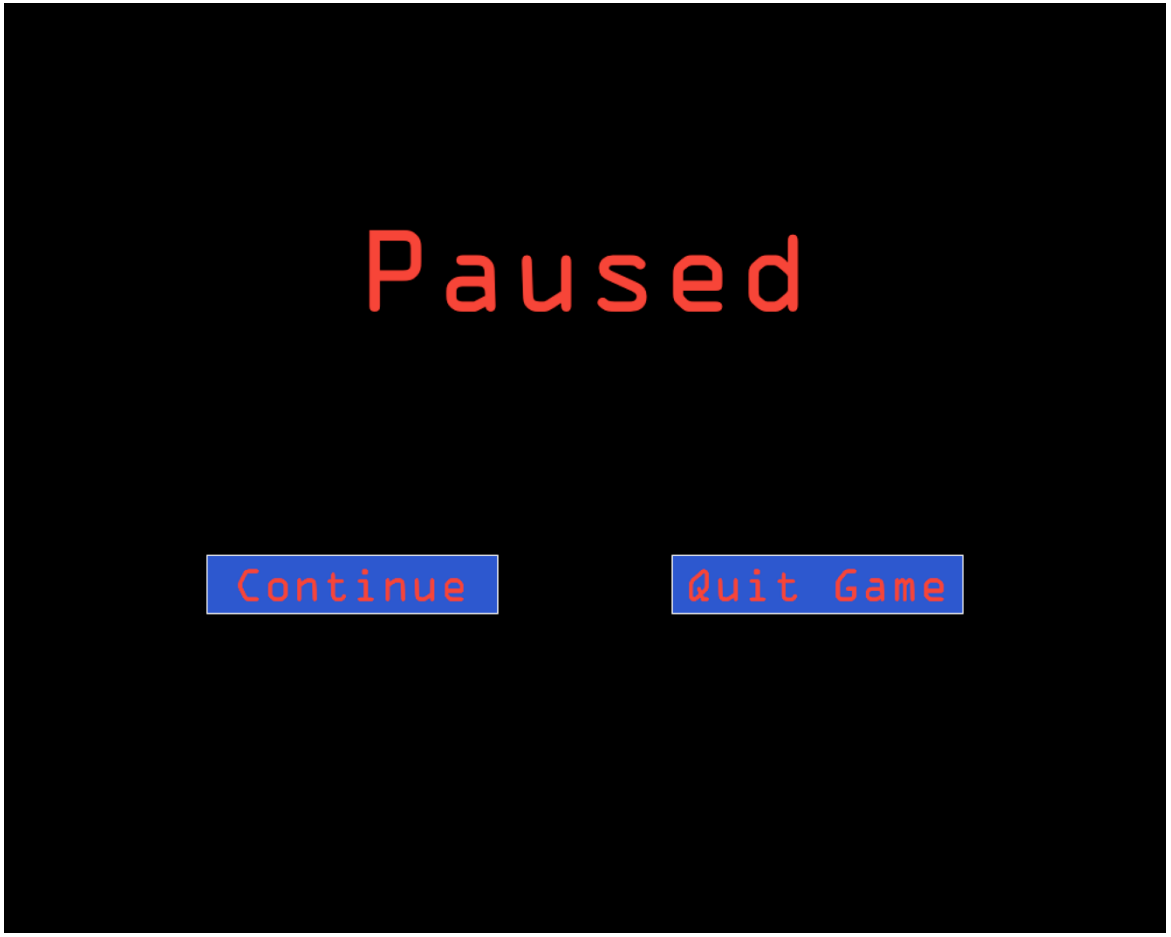
Start Screen:

After choosing a level, the player will be prompted to this screen where they must click on the left side mallet to start the game. The text and arrow are hover-sensitive (turn white when the mouse hovers over the mallet).



Play Game:

The game actually being played.



Paused:

If the user hits 'p' during gameplay, this screen will appear. If "Continue" is selected, gameplay will resume just as it was before the pause. Again, buttons are hover-sensitive.

Play Again:

Depending on which team (player or computer) reaches 7 goals first, the user will encounter one of these screens. "Yes" brings the user back to the "Choose Level" screen, while "No" quits the application. Again, buttons are hover-sensitive.

Conceptual Description:

I was inspired to create this project by the Air Hockey game on my iPhone. It was simple yet elegant, and a good amount of fun, so I decided to make my own. I designed this program using Object Oriented Program, with separate interacting classes for each element of the program (puck, table, and mallet).

I relied heavily on basic tutorials from the Processing website for much of this project. Vectors were particularly useful in terms of the physics of air hockey. I also consulted with Mike Demsher, a classmate of mine, who was creating a billiards simulator, about how to mathematically represent two balls bouncing off each other. He directed me to the website http://www.hoomanr.com/Demos/Elastic2/, from which I took my function for mallet-puck interaction.

Project Reflection:

This project was very fun to create. The table was the first thing I made, and even just seeing the table object displayed was a very proud moment for me - it really did look like an air hockey table! Overall this assignment was enjoyable. I felt a little lost at first without any real guidance on what we could or should do for a final project, but once I discovered what I wanted to do, I appreciated that freedom, as it allowed me to be as creative as I wanted without worrying about restraints.

The aesthetic part of my program - displaying text, buttons, pucks, mallets, tables, etc. - did not really trouble me all too much. I was good at manipulating basic shapes to create more complex ones. The physics of the program, however, was more complicated. Getting the puck to bounce off the mallets was a particularly difficult task - I still have been unable to fix the bug of the spastic mallet and puck when the two are overlapping, despite much online searching and tweaking of code.

Other than this slightly minor yet altogether very frustrating bug, I am pleased with the way this project came out. I particularly like my use of a "mode" variable (initially an int variable but eventually an enumerated type) - I thought it to be an intelligent, elegant way of solving the issue of different game states.

Acknowledgements:

As mentioned earlier, the outside resources used to help develop this project were the Processing website and the website below, which was suggested by Mike Demsher.

Website: http://www.hoomanr.com/Demos/Elastic2/