Angelo Indre

Professor Xiao

Computer Graphics

4/19/2022

Barista Simulator Final Report

Barista Simulator is a WebGL-based interactive game where the user plays as an employee at a coffee shop selling coffee and croissants. The style of the game is 2D RPG. All the drawings and interactable objects on the screen are made from simple 1x1x0 white squares, colored, stretched, shrank, and translated to appropriate positions on the canvas. Composing tons of these modified templates together, I constructed complex shapes like my personal favorite, a croissant.

The current version of the game has 20 customers implemented as a set of lists, orderTaken[], served[], cOrder[], cXPos[], cYPos[] and for each of these lists, every index represents a customer. Each order is randomized with Math.random() and there is only 1 path for customers to follow which I defined through “< >” comparisons to their (X, y) coordinates. Customers walk to the counter and stay there until orderTaken[customerIndex] is set to true. They then give the player their order and move over to the far-left counter to wait for their order. Once the player serves all 20 customers, they have beaten the game in its current form.

The player in this game can move in 2 dimensions, controls bound to the keys WASD. Their motion is limited to the area enclosed by the counter where they work. When the player approaches the front counter, they attempt to take the order of a customer if there is one there and the player’s last order has been filled. They can go to the croissant or coffee to pick one up. Go to the trash to throw away what you are holding and go to the left counter to serve a customer when they are standing there and you hold the correct order.

If I were to continue working on this project, I’d make some improvements and changes. Right now, everything in my script is implemented functionally. Which got me pretty far in terms of keeping things organized, neat, and easy to work with. Although I feel like my customer implementation is hard to improve and extend. I noted in my code that I’d like to change customers to be implemented as a class with methods and fields, or whatever the JavaScript equivalent of that is. I believe this more object-oriented style would allow for customers to have more variety in their behavior. The end goal with this change would be to allow more customers in the shop, let lines of customers form, let served customers roam about the dining room. I think this would make the game more fun and immersive. Another thing I would change is that right now, all the customer events in the game are dependent on the position and speed of the player. In other words, there’s no way you can fail or do bad at the game because customers will only enter if you are caught up on orders. The game would be more interesting if the customer’s actions were determined by a time or clock variable so that tasks and order filling are faster. I’d like to give output of how you do in the game as well. Assign a price to each coffee shop item, keep track of how many are sold and output your sales, profits, expenses and whatnot at the end of the game. I’ve seen games like this one before that let you use these profits to upgrade your store. I could make it so that coffee and croissants have a cooldown, or slow the player down, and other creative obstacles that can be overcome by using daily profits to but upgrades. I love games that have senses of humor too. One idea I had was to add in a lazy manager character with dialogue about how there are no bathroom breaks due to the counter and store having no visible exits. But text and dialogue on the canvas just seemed too daunting to design with my strategy. A more fun name than “Barista Simulator” would do the game justice as well.

More things I didn’t get to do that I think would make the game nicer would be to resize the canvas to fit the screen, maybe add a menu that contains directions, controls, and some fun options like changing the customer frequency, movement speed, or change the complexity of the orders. Reasons why I didn’t do this are time constraint related. The longest phase of this project was drawing objects from the 1x1x1 stencil. To make characters and a whole interactable menu would have taken lots of time. It definitely gives me a firsthand perspective of why virtual graphics studios are important to make that process faster. If I had a tool that let me color pixels with my mouse and did all the tedious coordinate and size coding for me, that would have allowed me to draw my objects so much faster.

Another thing I thought about is that while my game is relatively small compared to ones that have high system requirements, I’m rendering a 2D game with 3D coordinates. Since the z-value is constant for all of my coordinates in all of my many objects, I’m led to believe that framerate would be faster if I took out the z-axis.

Overall, I am pretty happy with the way my game turned out. Even though I have lots of changes I’d make and ideas that did not make it into this version, I still had lots of fun putting the game together. All the code for this project was written by me. Resources from The WebGL class examples were helpful for formatting and syntax of my files.