Claw Craziness - Technical Write Up

CSCI 4100U

Justin Trudeau Victim Support Group

https://github.com/CSCI4100U/course-project-justintrudeauvictimsupportgroup

Project Overview

The following project was created with the idea of mixing IoT structures with mobile devices to allow for remote control of electrical systems. We decided to apply this concept in the form of a game that would allow the user to remotely play an arcade claw machine. To achieve this, our group mixed both hardware and software development, as well as a series of other concepts to bring the idea to light.

We started by evaluating the control mechanism of the machine, which gave us an understanding of how the machine was controlled. After evaluating the controls, we began to test electrical connections in an attempt to design a basic control system. After some testing, we decided to use a transistor setup hooked to the GPIO of a Raspberry Pi. This would allow us to host the web server on a device that was directly wired into the controls.

After some wiring and testing to ensure the integrity of the machine, we began writing a web server in node.js that would allow us to issue remote controls to the machine. Initially, we thought that we could use web requests to change the control state of the machine, however, this turned out to be quite laggy and inefficient, so we had to do some more research. We came to the conclusion that real time web technology would be necessary, so we opted to use WebSockets for our controls. After building a debugging site just to test the backend, we worked out the kinks, but we faced another issue. Hosting our own backend within the machine posed a quite difficult problem. That problem was that the server was being hosted on a virtualized local area network that prevented any and all outbound traffic from entering the network regardless of local firewall rules. This meant that our group would have to find a way to bypass the firewall of our network to allow clients access to the machine. Luckily, one of our group members had experience with networks and issues like this, and we were able to use local network tunneling to expose the web server. This meant that we were free to control the machine from anywhere and that our backend was functioning.

For the UI we decided to make a main menu to allow for easy navigation to each page, we put the title of our app “Claw City” at the top of the page with the ZenDots font from google to make it standout. We chose a background image of a city scape to go with our title and to make the app look nicer than having just a white background, we got the background image from: https://wallpapercave.com/w/wp4694506. We added a button in the main menu for each page the user could navigate to and gave them a distinctive color and icon to make them distinguishable. For the play page we made a youtube video player to show the livestream of the claw machine at the top and underneath we made 5 buttons to control the claw machine layed out in a standard controller layout to make it intuitive to use. At the bottom of the page we made a container that displays where you are in the queue of people waiting to play so it was easy to keep track of without being in the way. For the play history page we decided to make it a more simple design as it is only showing the raw statistics of the users play history. We stretched the data table to fit the screen so it looked nicer and added 2 buttons to the app bar. The left most button brings up a date picker that filters the data so we chose a sorting styled button to make it clear but added a tooltip just incase as well as displaying a snackbar that tells the user the filter

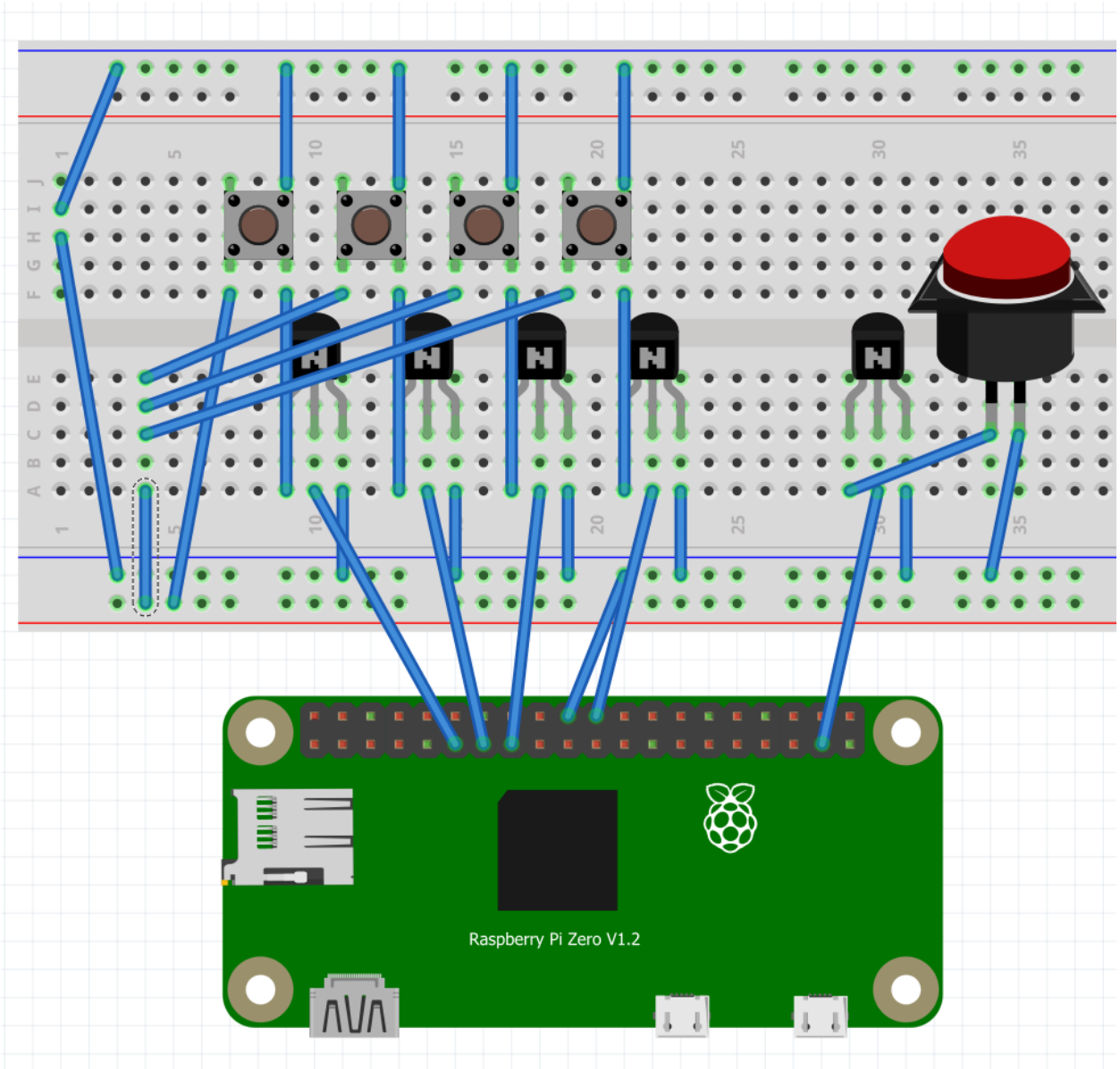
was cleared. The right most button clears that filter so we chose a clear format type button for the same reason but also added a tooltip. For the claw machine location page we show a map of where the claw machine is located so the user can get an idea of how far it is and to show that it's not simply one of hundreds in a warehouse somewhere, along with a picture of the claw machine itself so users can get a better idea of what it looks like than just from the stream which mainly shows just the inside of the machine.

Contributions

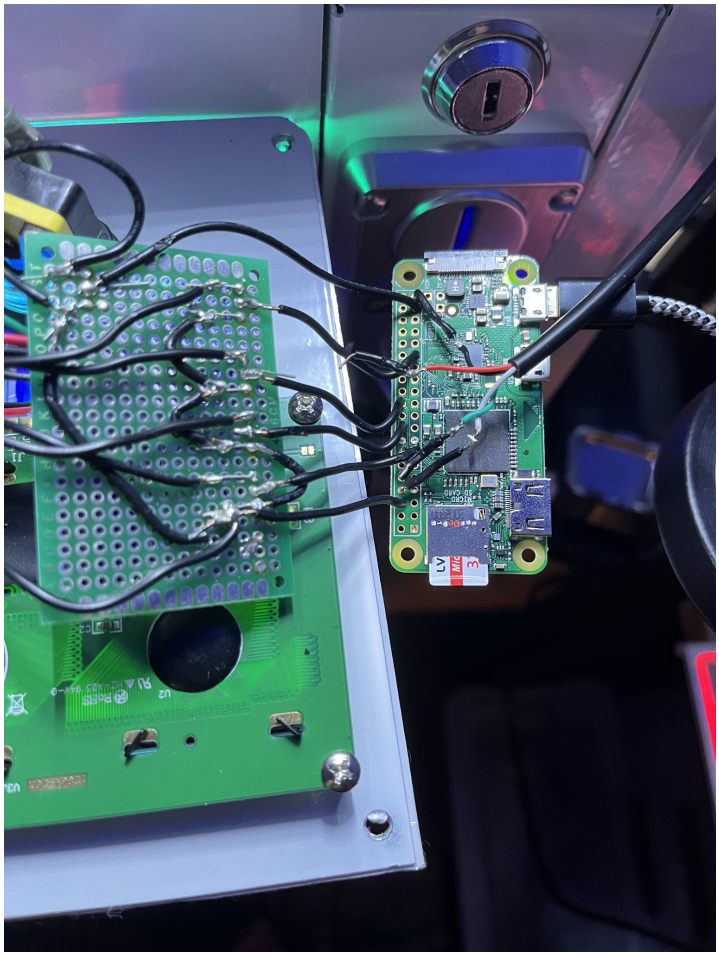
Michael Magahey https://www.youtube.com/watch?v=d0H9fBVdUrc	SSL Networking, WebSocket backend, electronics automation, networking and tunneling, Websocket flutter integration (frontend), Dynamic UI components (Play page), Technical Writeup, Phone Analytics, Notifications
Keith Middleton https://youtu.be/5XPcj6-zGuA	UI, Stream Functionality, Dialogue and Picker, Multiple Screens, Local Storage, Data Table, Snackbar, Technical Writeup (UI overview, User Guide)
Haaris Ahmed https://www.youtube.com/watch?v=pjcvV6EPc20	Map Ui, Location Tracking.
Andrew Clegg https://drive.google.com/file/d/1q0WIoxgBBHNkUlNlyu5j-qgk9SdZ-Xrn/view?usp=share_link	Charts, Multiple screens, attempted cloud storage
Ethan Randle (Not sent since 11:30pm)	

Circuit Design

Our base design uses



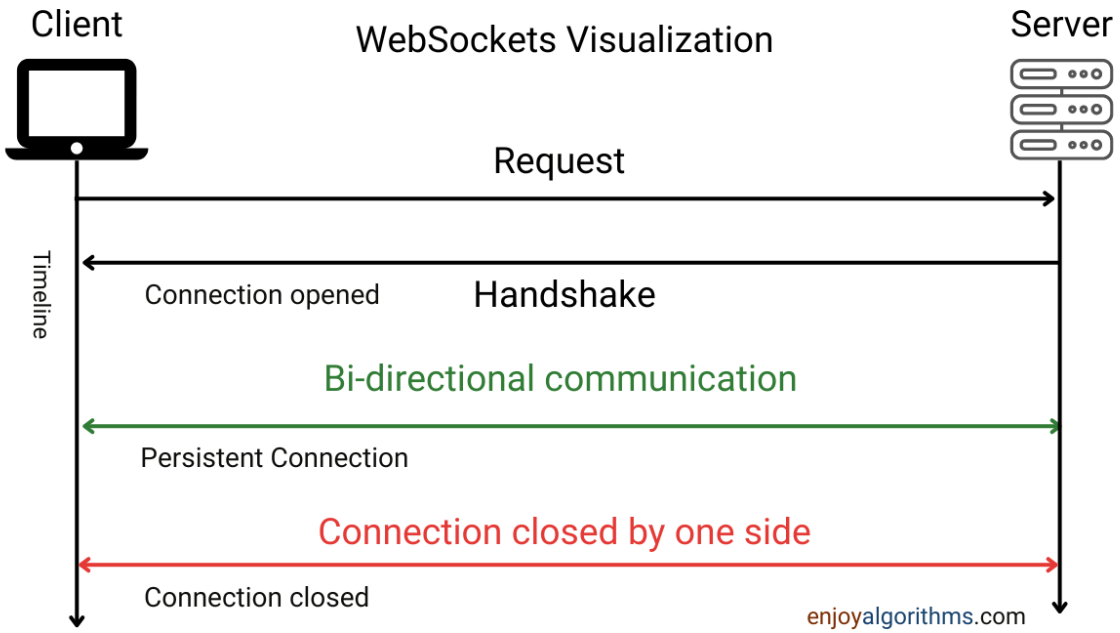
Transistor’s hooked into Raspberry Pi Zero Webserver



Code Design

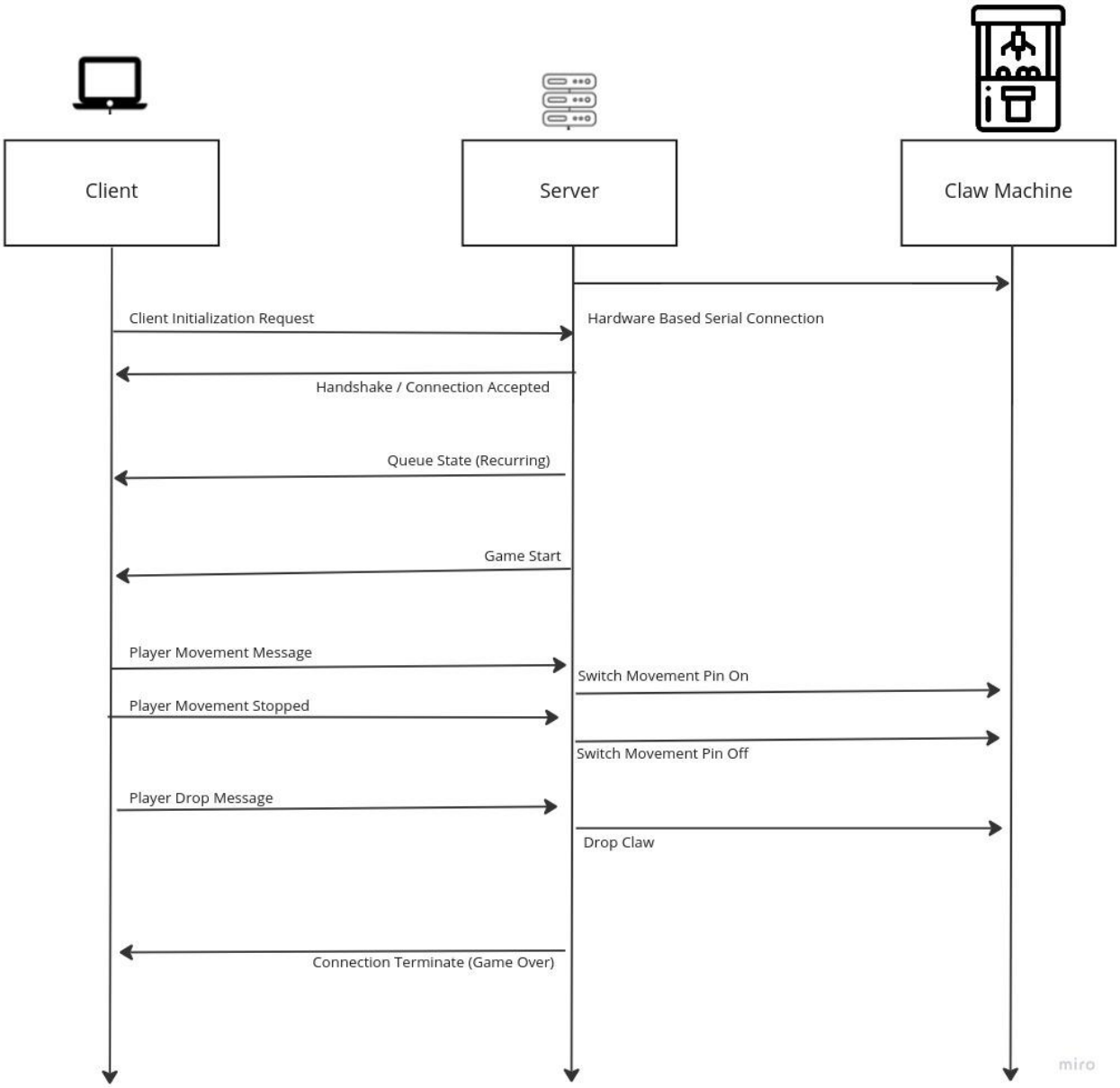
Basic Communication Design

When dealing with a system such as ours, real time connectivity between client and server is the most difficult issue. Traditional internet based software designs rely on the client to make a request of the server and the server to make a response. The response is then processed and loaded into the consuming service (browser, app, etc.). For our design, such a method would not be sufficient, as we need quick constant communication with the server. This lead us down the path of using WebSockets to communicate with the server in a more consistent format.



As per the diagram above, WebSocket functionality gives us the opportunity to have direct, bi-directional communications between our app and the server at all times, allowing for fast constant updates on our environment, as well as speedy usable control of the remote device.

Overall Communication Design



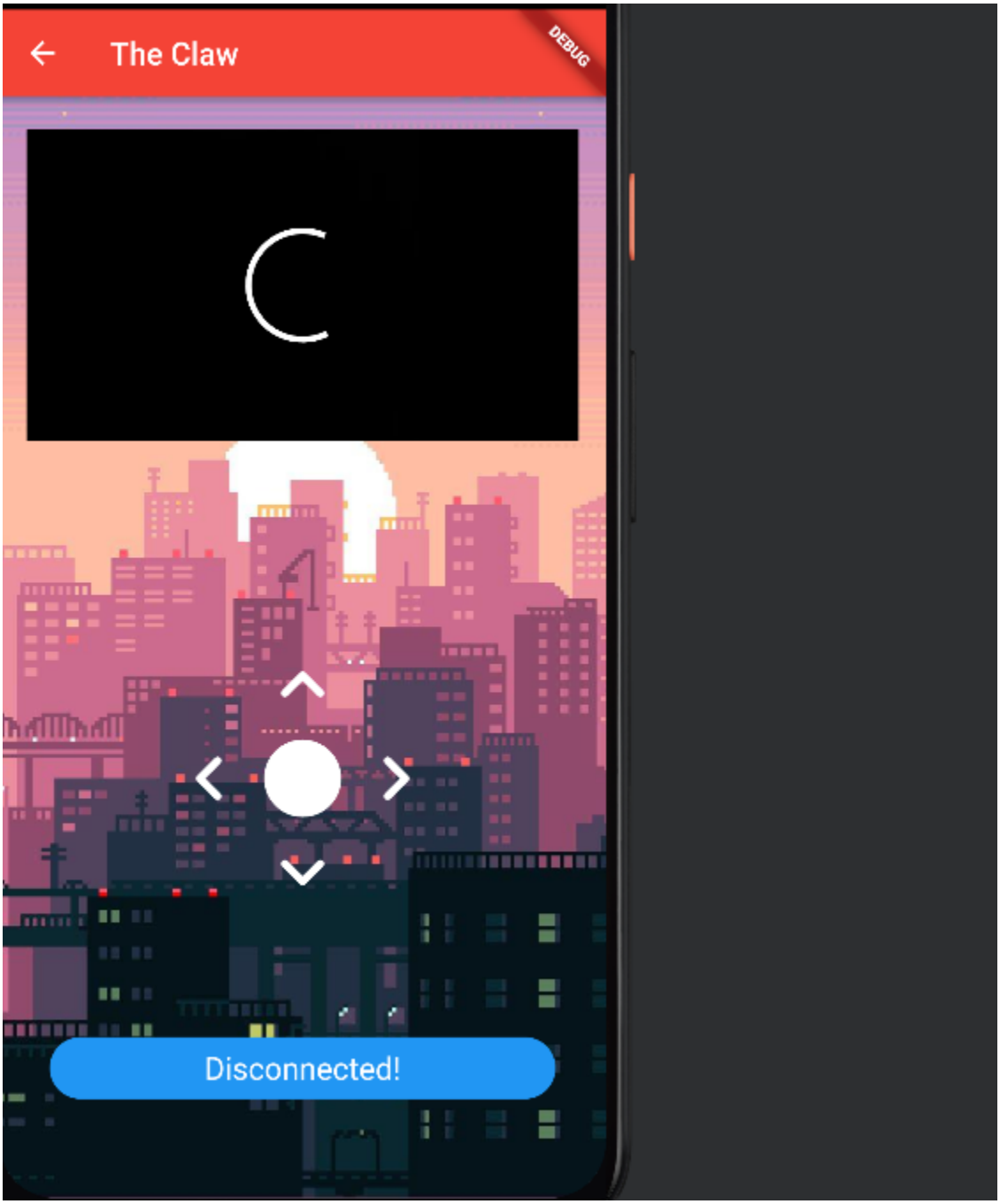
Due to the nature of our hardware, as well as the design of events handling in flutter, we opted to design our backend with a set and clear structure for handling events. This prevents the need of the socket to constantly be spamming the server, and rather, allows for control updates to only be communicated at the moment of change. This allows for much more fluid movement of the machine, as the server is not constantly handling messages. In our design, we also built the solution with scalability in mind. This manifested as a queue waiting system for control to be shifted from client to client. The instant that a client ends their game, their controls are locked SERVER SIDE to prevent any interference with others games. Other design changes that resulted of this scalability issue were a robust disconnect handling service that ensures no hung games as well as dynamic queue resizing, should a customer decide to leave their place in line. Not to worry though, we also designed our system with the customer in mind. Customers may

enter the queue, then minimize our app as they work on other priorities. Once the customer is getting close to their turn, they will receive push notifications informing them of their queue status and even when they are ready to play.

User Guide

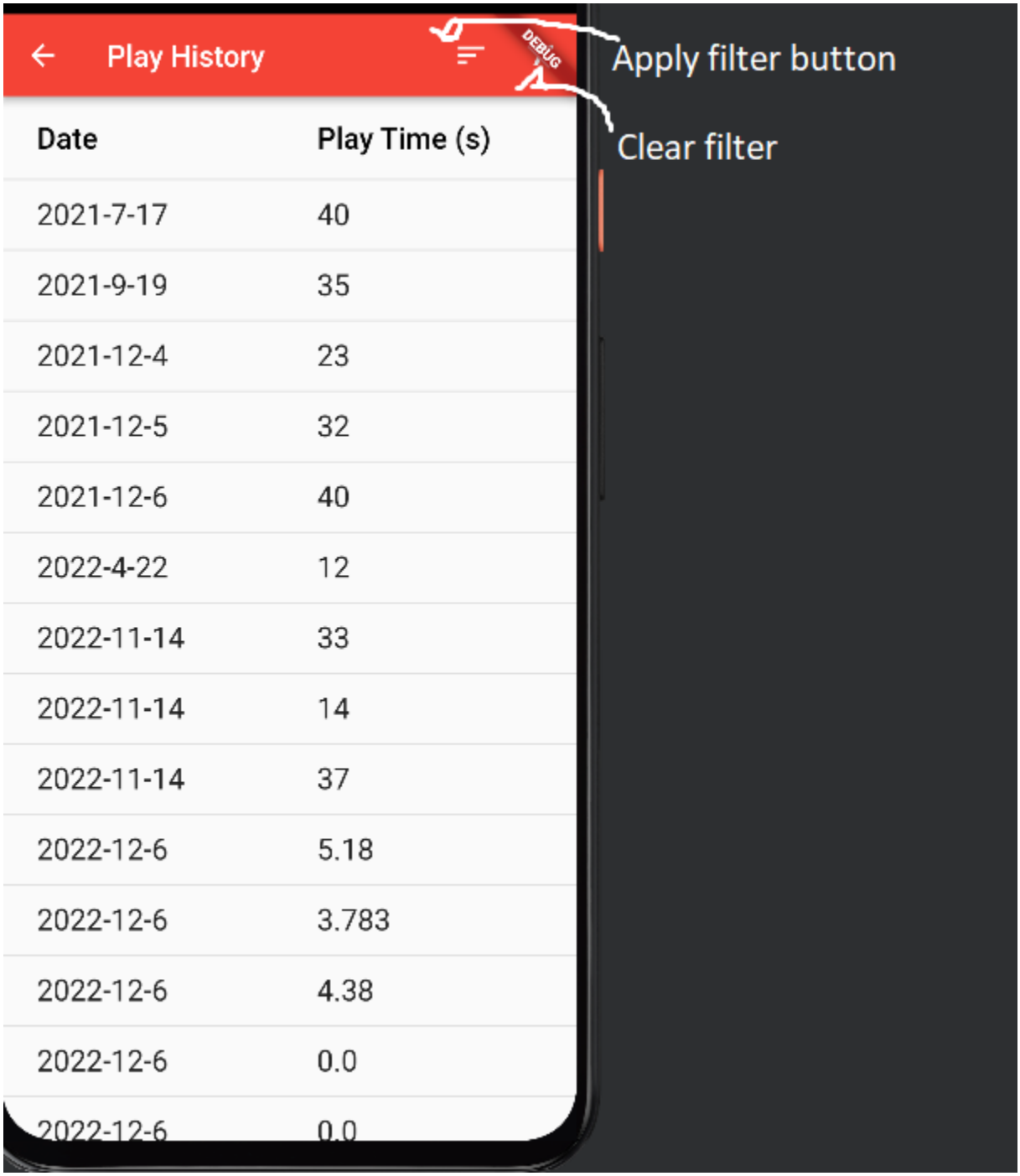
When the app is opened you will be presented with the main menu, the menu has 4 buttons the one labeled play will bring you to the page where the claw machine is controlled from, the play history button will bring you to a page that shows you your play history, the

globals stats button brings you to a page where you can choose between two charts, the CM location button will bring you to a page that shows you on a map where the physical claw machine is located.



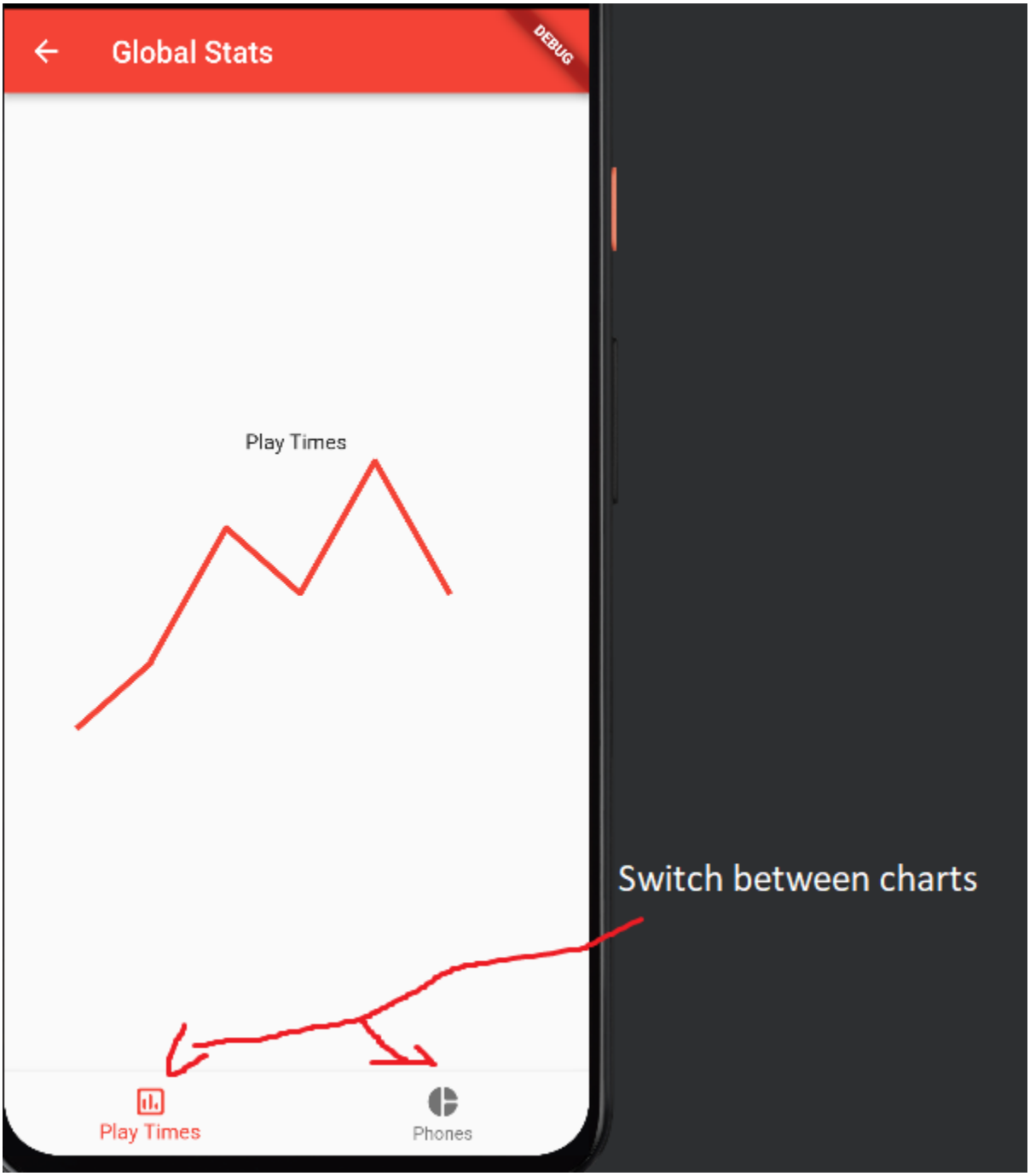
When you go to the page the claw machine is controlled in you will be added to the cue of people waiting to use the claw machine, and your position in this cue will be displayed at the bottom of the screen. At the top of the screen you will see the live stream of the claw machine so you can see where the claw and prizes are. Beneath the stream is the controls for the claw, the up arrow moves the claw towards the front of the machine, the down arrow moves the claw towards

the back of the machine, the left arrow moves the claw towards the left of the machine, the right arrow moves the claw towards the right of the machine, and the circle button will drop the claw at its current position. Pressing the drop button will also bring up the option to play again, if you select yes you will be put back into the cue to play again, if you select no you will be returned to the main menu it will also add a play history to the data table.

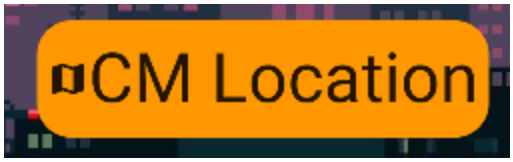


When you go to the play history page you will see the date of every time you have played as well as how long you played for in a scrollable data table. In the app bar you will see two buttons the

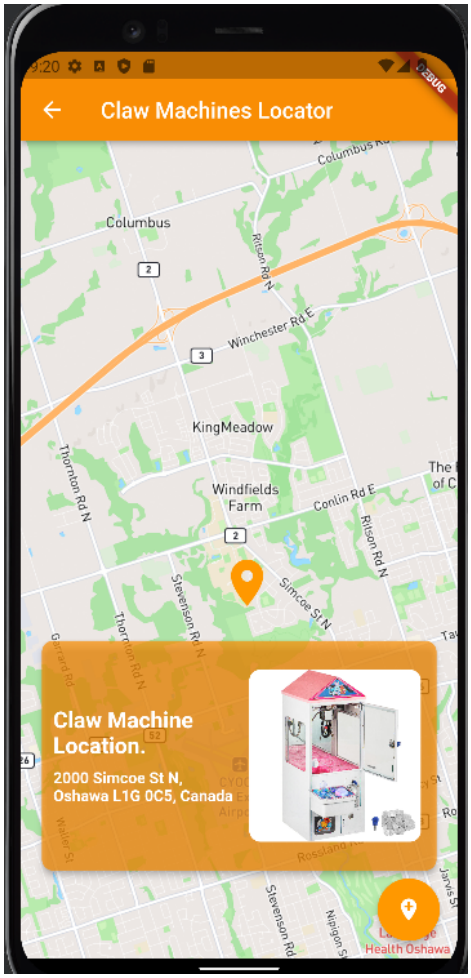
left button will let you pick a date to filter the data table to so you can easily find all of your plays from a certain date, it also has a tooltip incase you forget, the rightmost button will clear that filter so that you can once again see all of your plays, this too has a tooltip.



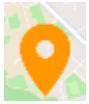
When you go to the global stats page the first chart shows the play times of other users as a line graph, and the second chart shows a pie chart of the different phone types used.



Now we shall explore the CM/Claw machine location option, here we have an option to view a map that will allow us to see where our claw machine is located.

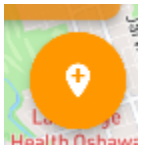


This is the entire Map UI, using Mapbox as a base, we were able to display not just the location of our claw machine, but the location of the user as well.

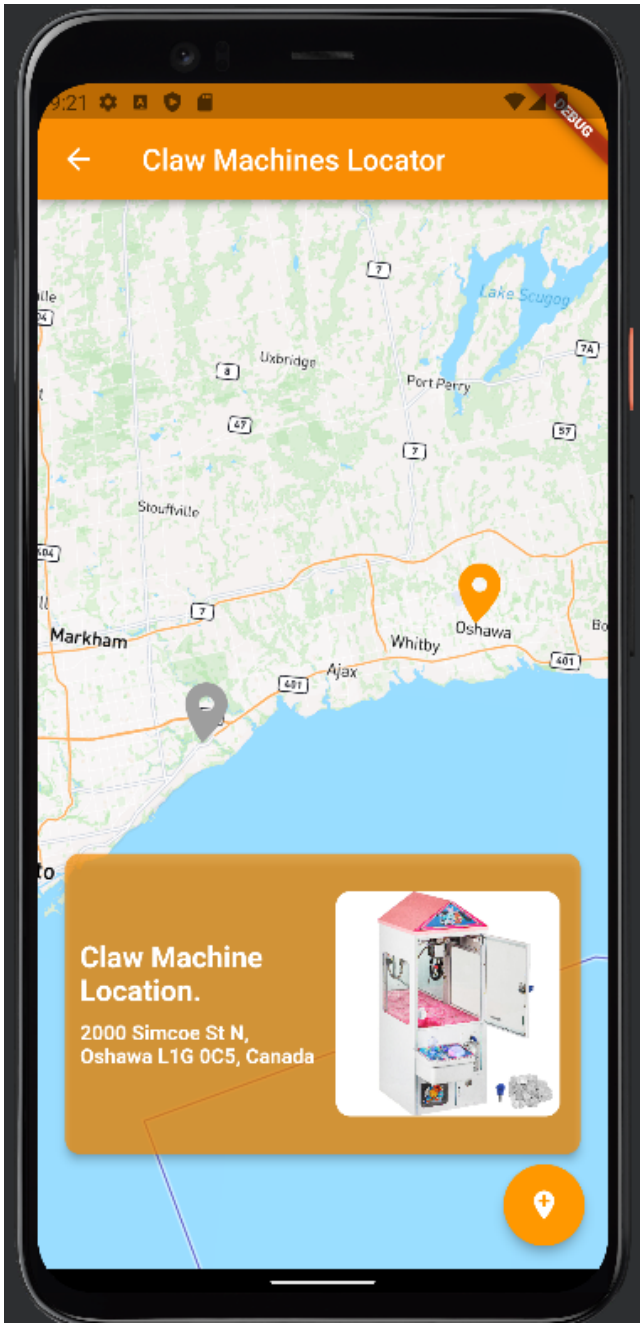


This is the marker for our map that we are using to display the location of the user on the map.

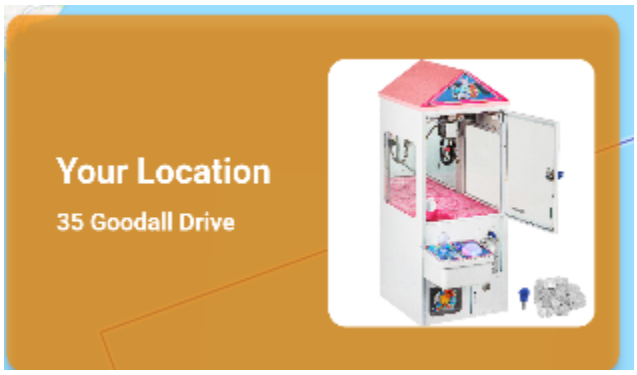
This panel provides a little bit of information about where we are located.



This is the option to add a marker of our current location so that we are able to get a brief overview of how far away the user is from the claw machine.



Clicking on these icons will allow us to swap from location to location.



Clicking those icons will switch the information panels too, switching from the user's location, to the claw machine's location.

Project Requirements

Uncompleted

Feature	**Delivery**
Dialogs and Pickers	The dialog appears after the user presses the drop button and asks if they wish to play again which will add them back into the queue if they do or return them to the main menu if they say no. The picker is a date picker in the play history that filters the play history to the specified date.
Multiple Screens	There is a main menu, a menu where the claw machine is controlled from, a page with the users play history, a page that shows charts, and a page that shows the claw machines position.
Snack Bar	Change the title bar to show who is playing
Notifications	Push notification on queue change
Local Storage	SQFLITE keeps track of when and for how long users play.
Cloud Storage	Keep record of players, locations and phones

HTTP Requests	Grabbing location of webserver
Data Tables	A data table displays when and for how long users play.
Charts	Show pretty statistics
Maps	Show current GPS and claw location
Geolocation	Show player on map relative to claw
Geocoding	Calculate distance from claw to player
Internationalization	Russian