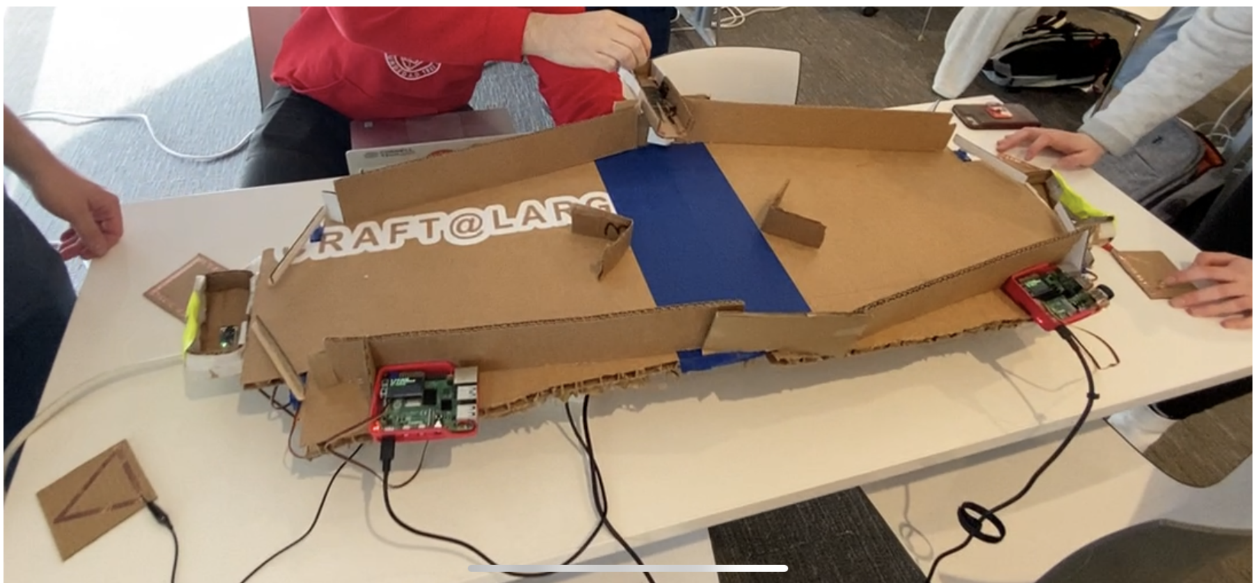
Two Player Pinball

Alex Kluver; abk98 - Matthew Ardizzone; ma759

How it works:

The design utilizes two Raspberry Pi's and four servo motors to implement a two person pinball experience, where each player has touch sensors to control the movement of the pinball paddles and prevent the ball from entering their goal. To slow down the gameplay a bit and increase interactivity, a single obstacle has been placed on each side of the board, making it more difficult for a user to hit the ball directly into the opposing player's goal. This pinball game is complete with a working scoreboard and pinball dispenser.

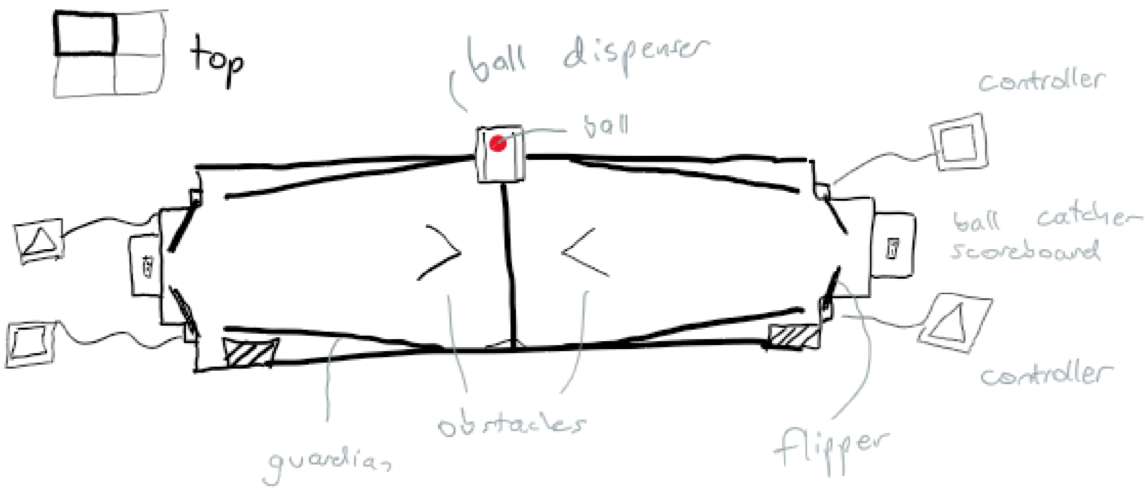
Final Implementation:

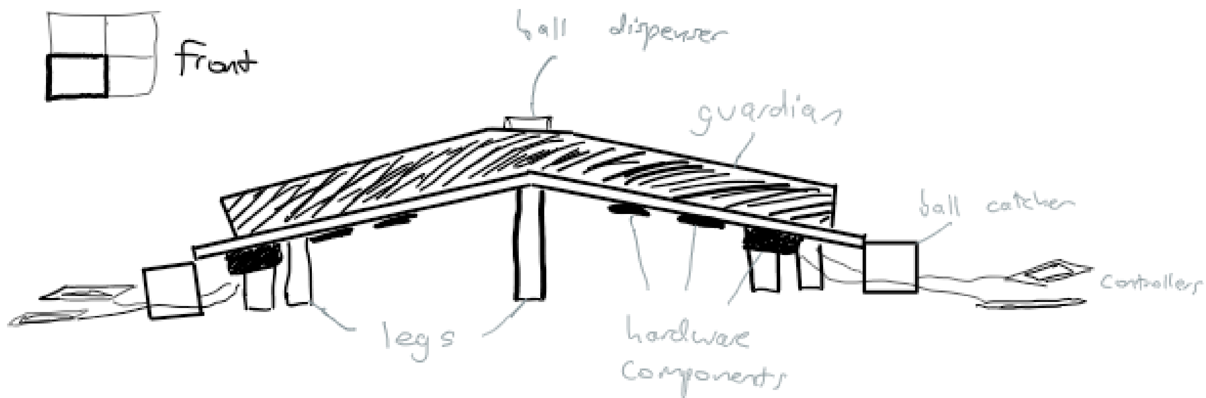


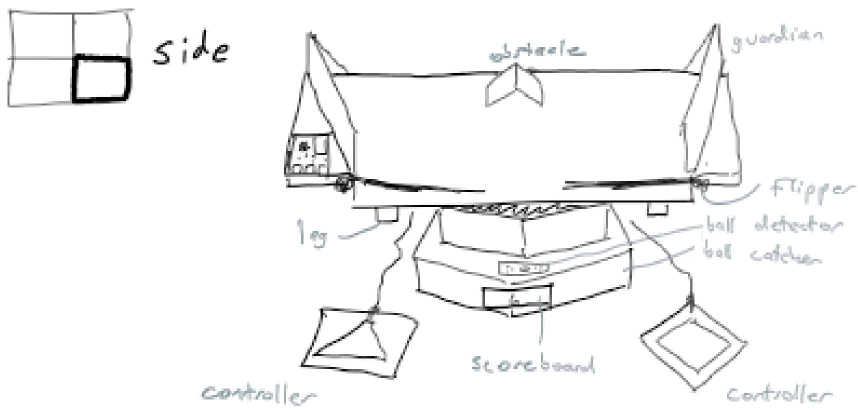
Video of Gameplay:

https://www.youtube.com/watch?v=XUWK7NXjNsw

<u>Designs Documentation</u>:







<u>Sensors and technical parts:</u>
2 Raspberry Pis
2 servo controllers with battery packs
2 capacitive touch sensors
2 proximity sensors
2 Mini OLED screens
4 servos
4 alligator clip wires
56 cm of conductive tape
0.4278 m^2 of cardboard

<u>Other miscellaneous parts used:</u>
8 inch wide masking tape
1 inch wide masking tape
Popsicles sticks

<u>Design Choices and purpose of each part:</u>

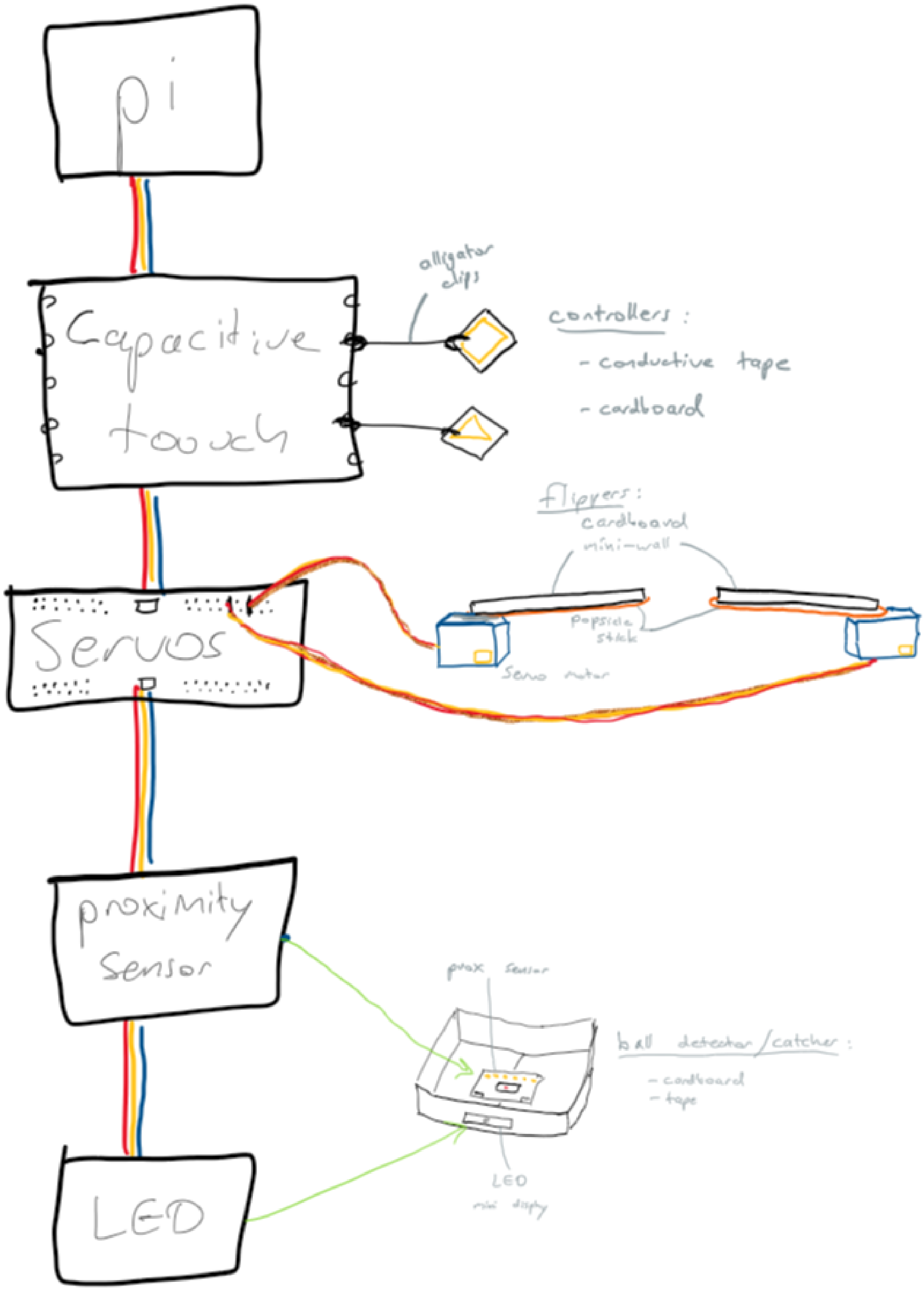
- Two Raspberry Pi's:
    - We decided to use two Raspberry Pi's as opposed to one since the were restricted on the distance our sensors were able to reach with the short qwic connectors that we were given with the rest of our parts.
    - Each Pi uses exactly the same code. The code controls the servos, proximity sensors, the capacitive touch, and the mini OLED screens
- Servos:

- We used 4 servos (two on each pi) to control the pinball paddle movement
- The servos only have two states in the program. Those states control the up and down position of the paddles. Since the servo positions are constantly being set in a while-loop, this gives it the illusion of having an angular speed based on how long the player touches their controller and allows for the player to make smaller hits by tapping the controller for a shorter period of time.
- As the Pi is not capable of powering multiple servos, we were also required to use the servo controller provided to us with our parts. This servo controller also made it easier to control multiple servos using a single program.

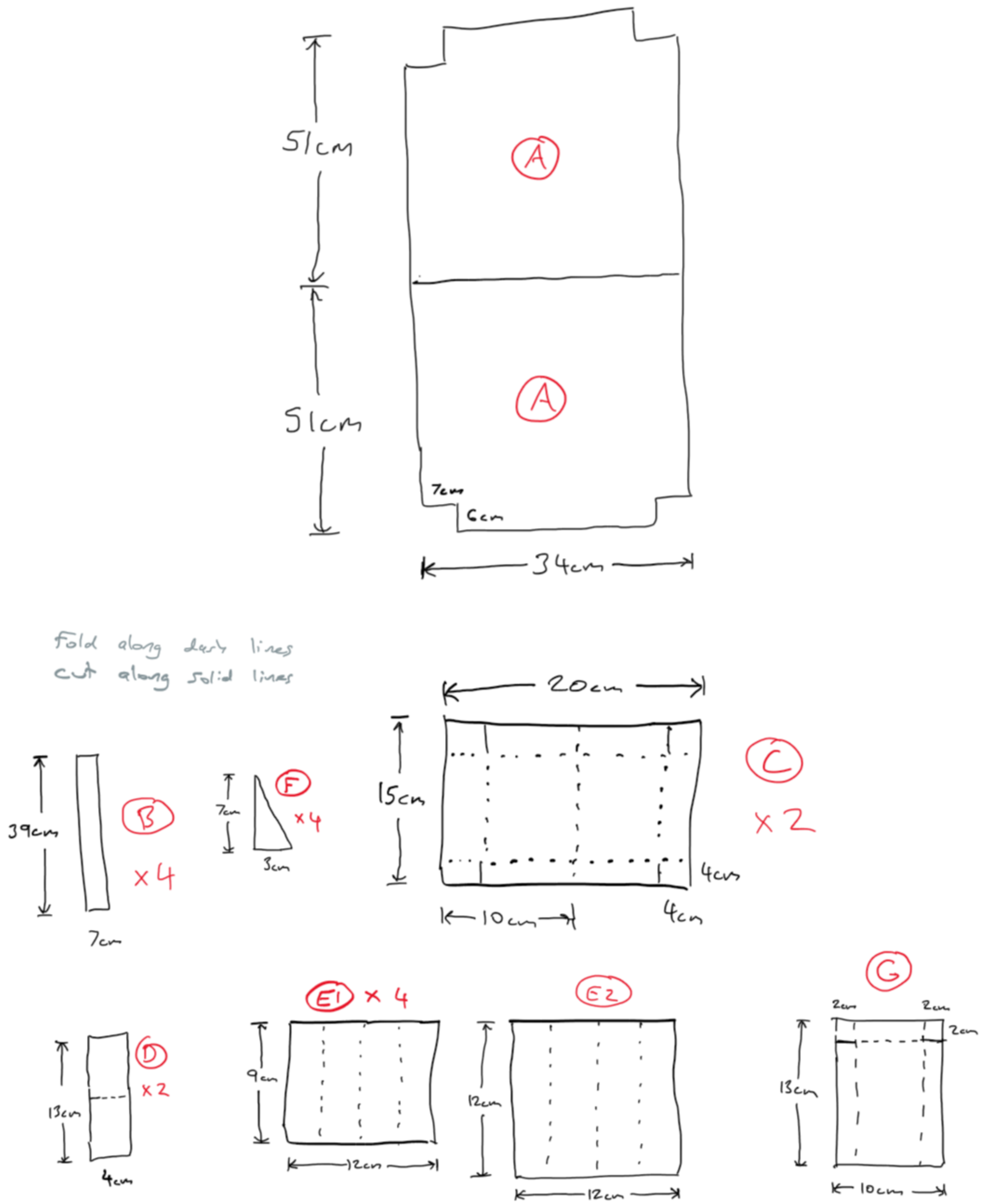
- Capacitive touch sensor:
    - As opposed to using regular buttons, we saw value in using the capacitive touch sensor, as it allowed us to create buttons of varying shapes and sizes.
    - The main purpose of the capacitive touch sensor is to use it as a boolean value to check which state the servo should be in.
    - The programming is set up so that when the user touches the capacitive touch, this sets the state of the servo to its up-position, and vice versa when the capacitive touch is not being touched.
- Proximity Sensor:
    - We wanted to implement a type of score tracking system in our final design, so we used the proximity sensor to increment a value when the ball falls into the goal.
    - To make this work, we set the proximity sensor to a low range setting and made the threshold of the incremental value very large, so there were not any false goals. When the ball enters the range of the proximity sensor, the score value increases.
    - A problem was also encountered when the user grabs the ball from the goal, the value increments again, so we set a sleep statement for the proximity sensor so that it does not double count a goal.
- Mini OLED
    - The mini OLED screen was used to display the score of the game. The opposing player's score can be found on your goal, and vice versa. The text on the OLED is set to the score value that is incremented by the proximity sensor each time through the while loop that controls the rest of the program.

Assembly:

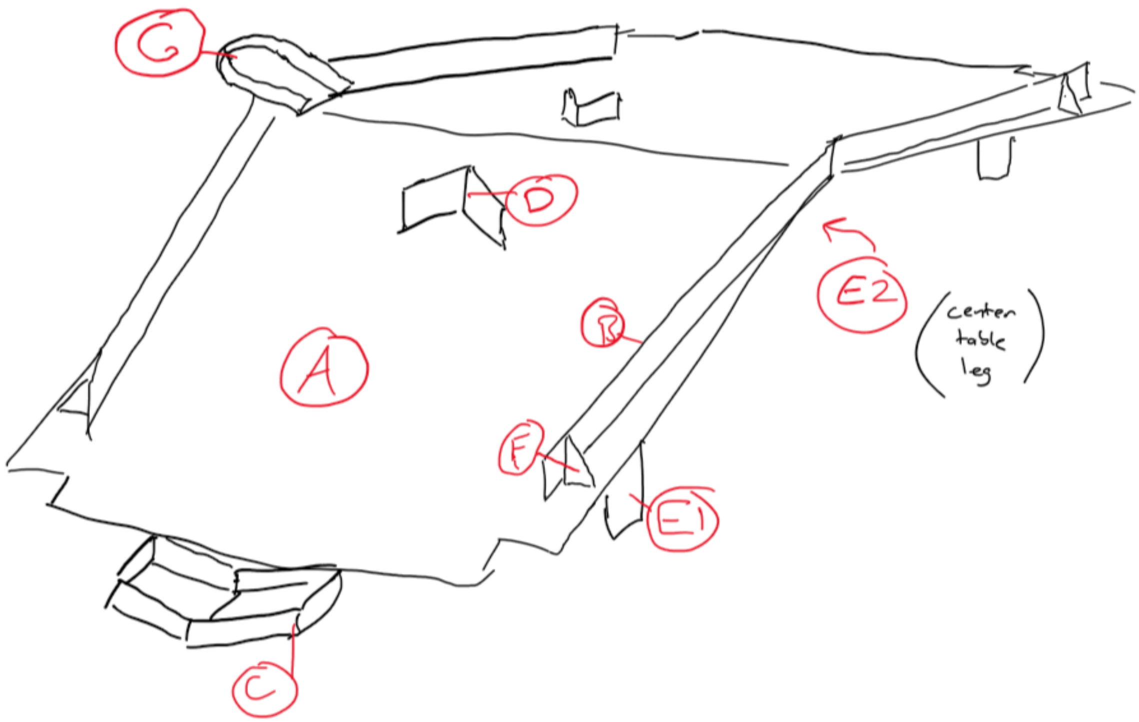
Sensor configuration:



Required cardboard pieces:



Cardboard configuration:



Assembly Process:
Hot glue was used to fasten all cardboard pieces together in the orientation seen above. The recommended angle from the edge to the center of the board is 15 degrees, but can be increased or decreased depending on the power/speed of your servo motors.

The Pi's and sensors were all fastened to the completed cardboard design with the use of masking tape. The location and orientation of most sensors can be changed to better fit the individual design, but since we only had short cables, all sensors were taped to the bottom of the board near the goal of each side.

Reflections:

There were a few technical issues we didn't anticipate before beginning to work on the project. The first was the quality of the servo motors as flippers. The servo motors only spin at a fixed speed which was too slow to propel the marble all the way up the table. To fix this issue, we considered using a gear ratio, but thought it would make the design too complicated. Instead, we used a simpler mechanical advantage by just making the lever arm longer. We also reduced the angle of the table. Another issue we had was the popsicle sticks were too thin from the side to strike the ball consistently. To fix this, we added a thin strip of cardboard barrier on top to thicken the flipper.

As far as design interaction, I think our device interaction was simple and intuitive. Their directive was made clear by the shape of the board and the fact that our device built on their familiarity with the one-player version of the same game. In other words, the users already had a mental framework for interacting with the device (playing a game) and could quickly adjust to our twist.

All users knew how to interact with the touch sensitive controller pads even before picking them up. All users also knew where to put the ball once a rally had ended, which we think is because of the overall simplicity of the device; a ball rolling down a ramp. One issue that we did find in our final design was that the users of the touch-controllers tended to hit the cardboard in places where the conductive foil was not positioned, causing no movement in the pinball paddles. This could have been easily avoided had we increased the amount of conductive tape on the controllers, but if we did that, then we would no longer be able to maintain the shapes to signify which controller controlled which pinball paddle.

As a result of an intuitive design, users did not need a long debrief before playing, and could jump in quickly, straight to the gameplay.