SYSC 3010 Project Design Solution

R Tanks (Group W7)

Group Members:

Jake Bridge, 101072342
Thomas Sloan, 101023007
Mahmoud Al Sous, 101071213
Marc Donovan, 101073218

Context and Motivation

The current market is saturated with home based laser tag game where play a large indoor space is optimal. What R tanks wishes to accomplish is to bring the laser tag experience to an indoor space requiring a fraction of the space for an enjoyable experience.

Playing also has many health benefits for both children and adults. In children it is necessary for social and physical development and cognitive and emotional well-being[1]. In adults it can be a stress relief, improve brain function and can help relationship development[2].

Objective

R tanks is a live action game which refreshes the traditional game of tanks by putting it in the real world. Let the digital world come to life as R tanks storms the battlefields near you. R tanks will allow users to face 1 v 1 in fast paced combat using sensors to tag the enemy. After the battle is done, users will get to see their score and compete for the top spot. R tanks wants to create an experience that is worth coming back to.

Scope

R tanks is a game which allows users to do the following:

- Have the user sign in with their unique credentials
- Let the user control the tanks with a smartphone application(Android/Apple)
- Have fully mobile tanks with a rotating turret
- Track the scores of the user that play

Design - System Architecture:

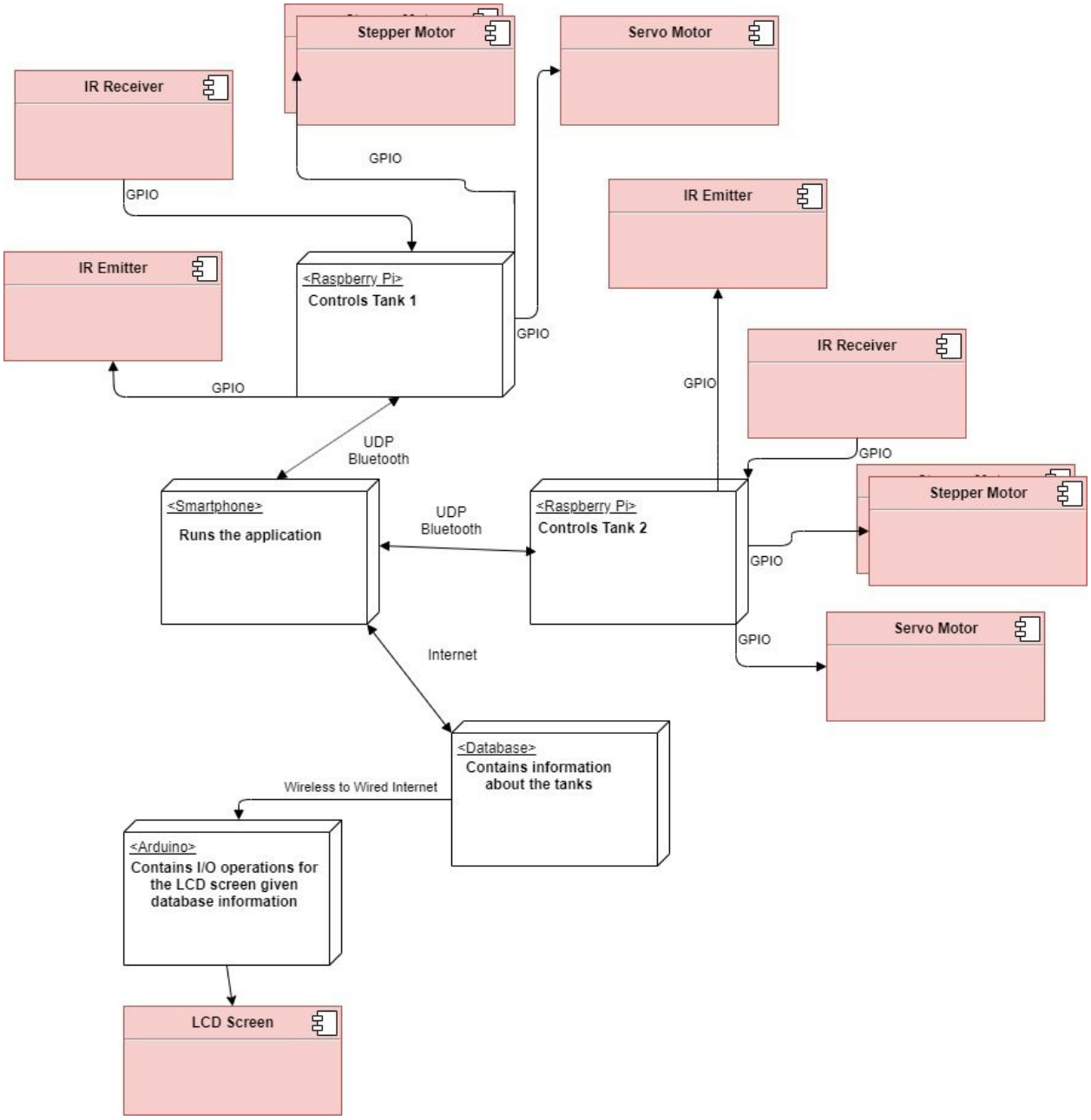


Figure 1: System architecture diagram

Design - Communications Protocol:

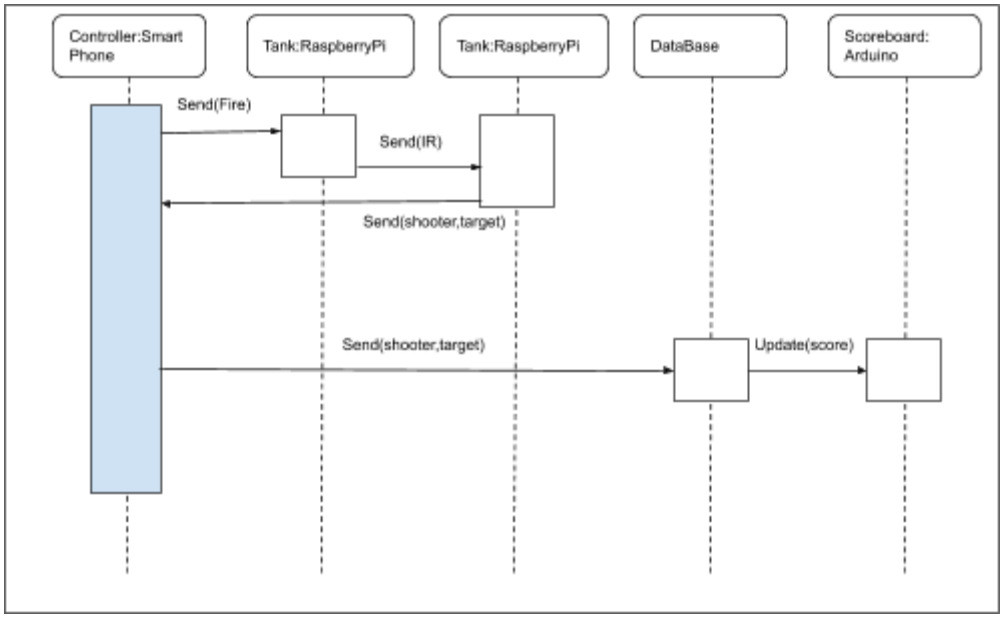


Figure 2: Firing IR beam sequence diagram

1. Player presses the “shoot" button on our app on their smartphone
2. The Raspberry Pi receives the “shoot” message and shoots an IR beam
3. If the IR beam is picked up by the other Pi’s IR receiver, it sends the “shooter” tank and “target” tanks information to the smartphone
4. The smartphone app relays this information to the database
5. The database updates the scoreboard, awarding a point to the “shooter” tank
    a. The “target” tanks information is necessary for player data collection. This allows players to view their stats after a game

Shooting cannon sequence table:

Sender	Receiver	Message	Format
Phone	Tank	Fire	Bluetooth: MQTT
Tank(IR emitter)	Tank(IR receiver)	*IR light* - actual IR light is sent	IR light
Tank	Phone	Shooter,Target	Bluetooth: MQTT
Phone	Database	Shooter,Target	Bluetooth: MQTT
Database	Scoreboard	score	UDP

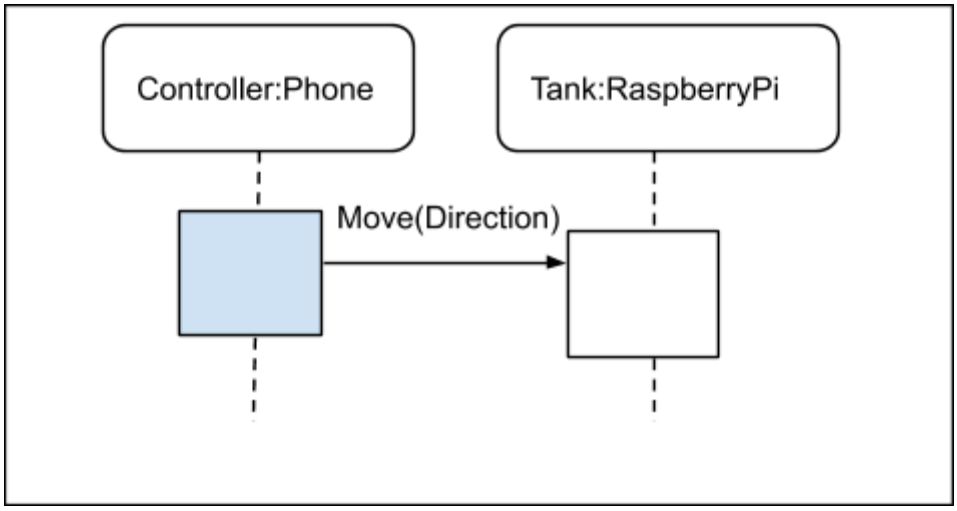


Figure 3: Tank movement sequence diagram

1. Player presses an arrow on the app in the direction they want to go
2. Tank receives this message and moves in the corresponding direction

Move sequence table:

Sender	Receiver	Message	Format
Phone	Tank	Direction (Left, Right, Forward, Backward)	Bluetooth: MQTT

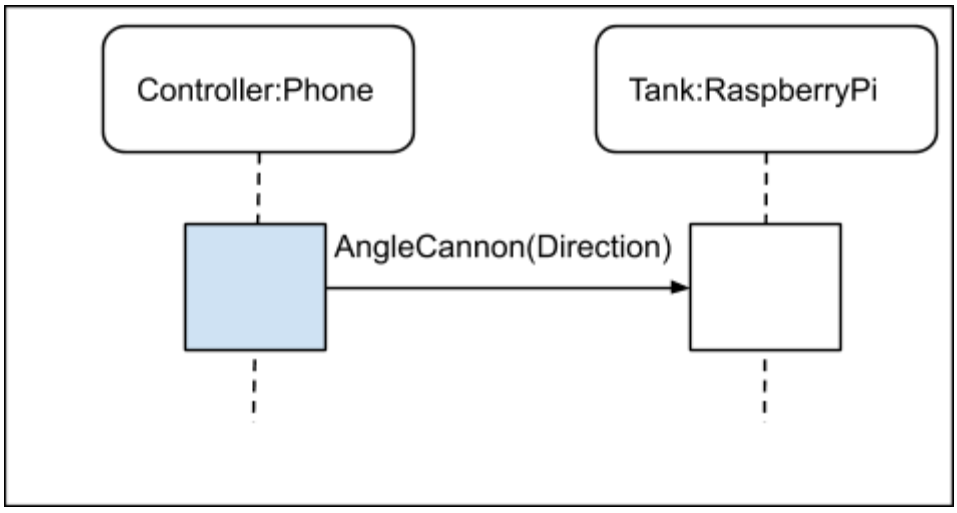


Figure 4: Cannon angle sequence diagram

1. Player moves the joystick on the app in the direction they want the cannon to go
2. The angle of the cannon moves in the corresponding direction

Cannon angle sequence table:

Sender	Receiver	Message	Format
Phone	Tank	Direction (Clockwise, Counter-clockwise)	Bluetooth: MQTT

Error Scenarios

Error	*Effect*	*Solution*
Dropped UDP Packet	This causes the arduino to miss an instruction	The arduino will have an if statement that checks the received packets. When it noticed that a packet was missed the arduino will ignore and poll the firebase again.
Bluetooth connection to tank lost	This will cause a player to be unable to control the tanks	Current game is ended and players must reconnect to their tanks
One phone's internet connection is dropped	This will cause the scores to not be updated in the firebase by one phone	The game will continue as normal with the scores being updated by the one phone
Both phone's internet connection is dropped	This will cause the scores to not be updated in the firebase	The arduino will display the latest score, and the phones will not send the data to the firebase. The game will be lost
Firebase is down	The scoreboard and the arduino will not be updated	The arduino will display the latest score, and the phones will not send the data to the firebase. The game will be lost
Dropped MQTT Packet	This causes the RTank to miss and instruction	The RTanks will be in a constant loop, waiting for instructions. No instruction will be critical, so if a packet is dropped it will be ignored.

Database:

The database will be stored in an online cloud database, Firebase created by Google. This will allow easy connection to our application, and allow the results of each game to update the leaderboard displayed on each device.

Database Schema:

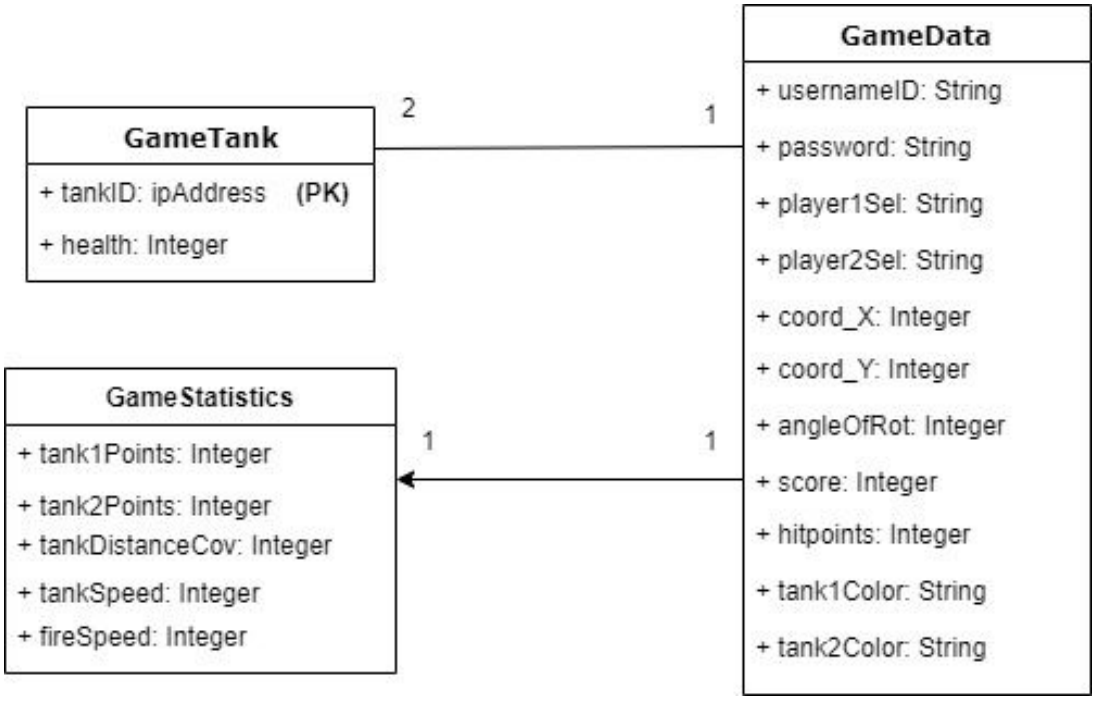


Figure 5: Database relational tables schematic

Figure 5 above shows the connections of the tables within the database. For each game, there will be two tanks connected with unique IDs as well as one set of statistics which are calculated from the game data.

Design - Hardware:

Game Design:

The game will feature multiple raspberry pi's as unique units that are able to "shoot" lasers at each other with sensors to detect when one is shot.

RTank Unit:

The RTank will have a car design with the wheels attached to the 3D printed shell. Inside the shell will be the battery, raspberry pi, and the stepper motors. One stepper motor will move the vehicle while the servo motor is used precision steering. The second stepper motor will be positioned at the top of the vehicle, with its shaft attached to rotate around the z-axis. Attached to the shaft of the second stepper motor will be the infrared emitter, which acts as the RTanks cannon, and the infrared receiver(s). The infrared receivers will be used to detect when the other RTank has had a hit it. Each RTank will then be controlled through the respective raspberry pi's Bluetooth connection. In this setup both Pis will be in headless mode connected over Bluetooth to a smartphone and communicate using bi-directional UDP. The sensor used is the IR receiver which starts a feedback loop back updating the score. The system will be non-expandable. To protect against IR spread, we will be placing the IR emitter in a tube similar to a real tank cannon barrel.

Smartphone Application:

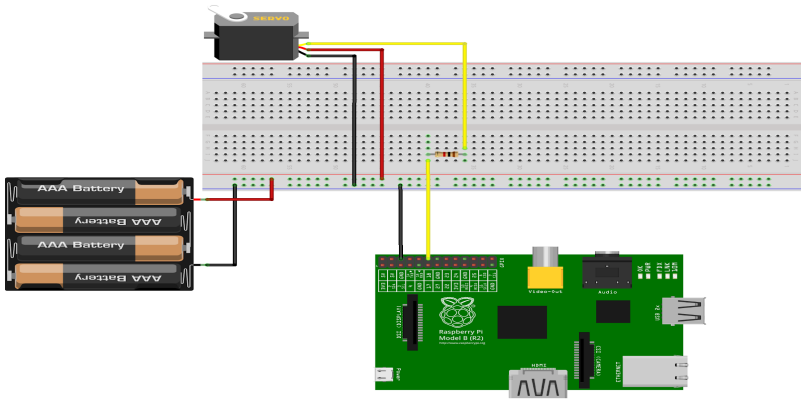
The smartphone application with be created using Flutter in the Dart language. It will have several different views. The primary view will be a create account/login screen. The login options include google accounts and account created in the app. The information of the users will be stored in the database. The second view will have a start game and a scoreboard button. The scoreboard screen will have the leaderboard with the win-loss records of all the players. This information will also be stored in the database. The start game view will have a select RTank view (i.e. RTank 1 or RTank 2). When all tanks have been selected the game will begin.

The game controller interface will have digital joystick on the top left to control speed and direction. The bottom right hand corner will have three buttons, left right and fire. The current score will be displayed in the middle top, and a "heat bar" will be displayed at the top. The heat bar is a programmed limit to how many shots can be fired in a time period.

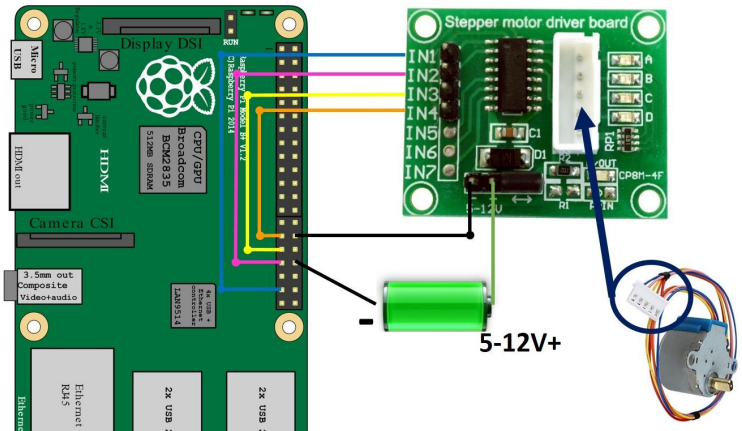
Arduino:

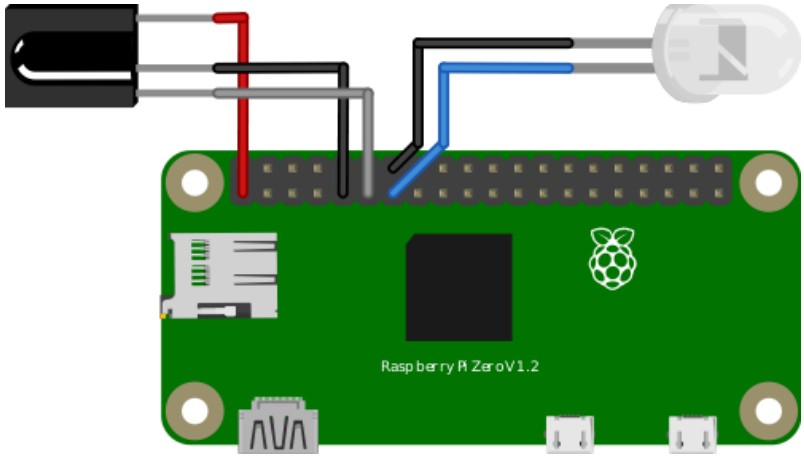
The arduino will be connected to the internet via a hardwire ethernet cord and have an LCD screen. It will then periodically poll changes from the Firebase database and display them on the LCD display. The arduino will be this systems wired network connection.

Components:

Part	Servo Motor
Part Number	SG90
Description	Lightweight servo motor that will be used to control
Key Parameters	Voltage: 4.8-6 V Torque: 2.5 kg-cm
Circuit	 [3] https://www.bluej.org/raspberrypi/ServoMotor.html
Hardware Interface	Yellow → PWM Red → VCC Brown → Ground
Software Interface	Can control position based on PWM signal.

Part	5V DC Stepper Motor with driver board

Part Number	Motor: 28byj-48 Driver Board: ULN2003
Description	The motors are used to control the RTank's movement and the turret direction. The drivers are to make communication with the raspberry pi easier.
Key Parameters	*Motor:* Rated Voltage: 5V DC Number of Phases: 4 Stride Angle: 5.625°/64 Pull in torque: 0.3 kg-cm Insulated Power: 600VAC/1mA/1s Coil: Unipolar 5 lead coil *Driver:* Voltage drop from source to motor: ~1V
Circuit	 **[4]** https://www.hackster.io/erickbp/stepper-motor-from-windows-10-iot-core-d3c5d6
Hardware Interface	Blue → Coil 1 Pink → Coil 2 Yellow → Coil 3 Orange → Coil 4 Green → VCC Black → Ground
Software Interface	Can control the direction and speed of the motor by making the GPIOs high or low in a specific manner.

Part	IR Emitter/IR Receiver
Part Number	Emitter: 5003MD Reciever : TSOP1738
Description	These devices either emit IR light or detect IR light.
Key Parameters	Diameter: 5MM Wavelength: 940nm
Circuit	 **[5]** https://camp.isaax.io/en/isaax-examples/ir-control-via-lirc-on-raspberry-pi-zero-w
Hardware Interface	*Receiver (Black)* Red → VCC Black → Ground Grey → Signal *Emitter (Clear)* Blue → VCC Black → Ground
Software Interface	*Receiver (Black)* Uses manufacturer library to output pulse. *Emitter (Clear)*

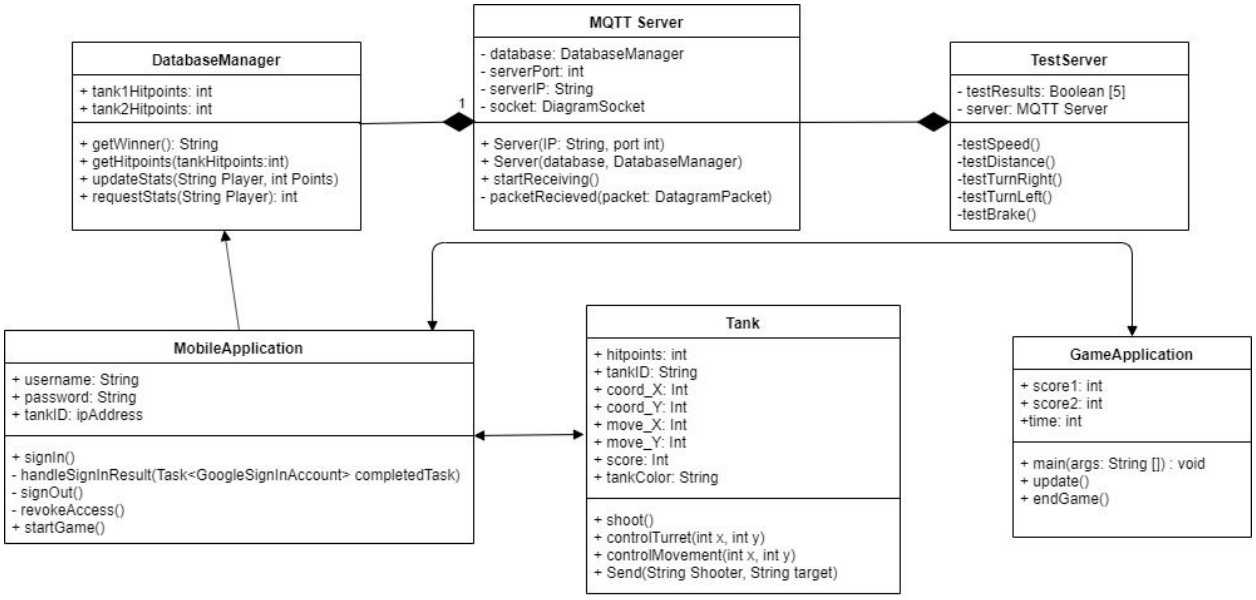
	Emitters light if GPIO is set to high.

Testing:

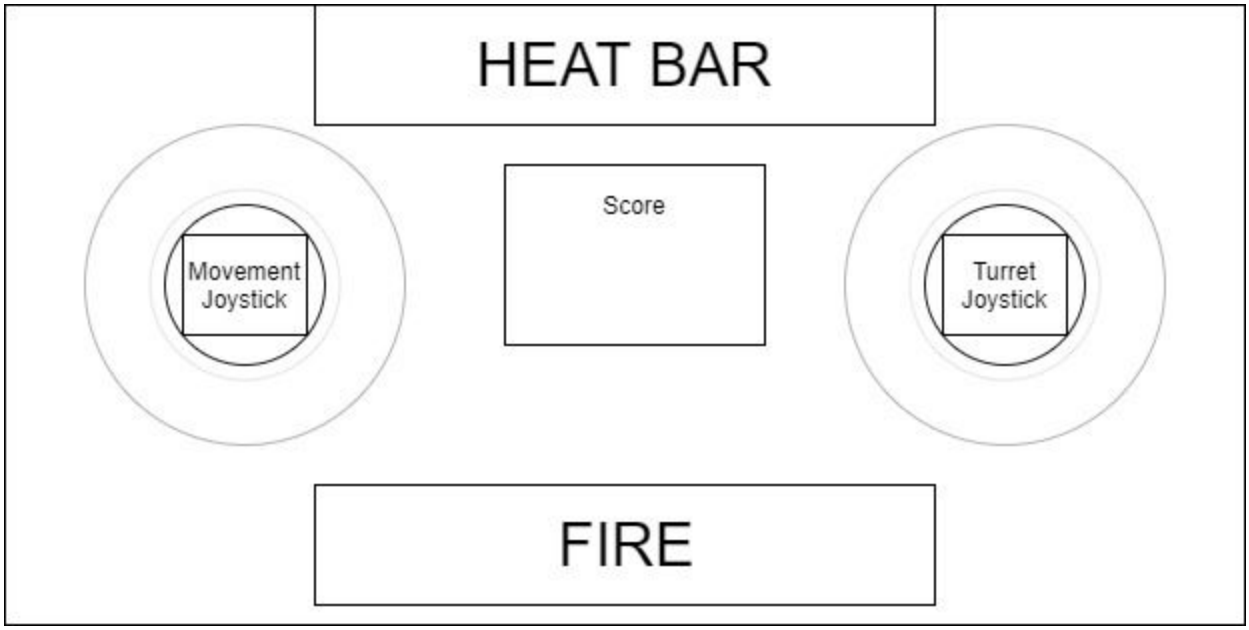
Part	Function	Functions to Test
Servo Motor	Cannon position	1. Left/right movement - To prove our servo motors function 2. Tracking position - Keeping track of the current angle of the motor will be important for aiming the cannon 3. Speed of rotation - The speed that a user can change the position of their cannon will be important for gameplay These tests will be done by writing test code for the Raspberry Pi's and observing the results.
Stepper Motor	Tank movement	1. Steering - We need to test and set the limits for how far left and right a tank can turn 2. Speed - We have to make sure the tanks are not too slow as well as not to fast so that they are agile without being too difficult to control 3. Forward/Backwards - The tank must be able to go forwards as well as backwards for navigation purposes These tests will be done by writing test code for the Raspberry Pi's and observing the results.

IR Receiver	Tank hitbox	1. Intensity - The intensity of IR that a receiver can read will be important to determine if a tank was hit by a shot This test will be done by shooting IR light at different intensities and seeing if the receiver reacts to it. Our goal is to find an intensity that will trigger a reaction within 2 meters.
IR Emitter	Tank cannon	1. Intensity - The intensity of the IR light that it can produce will be important to determine if a cannon shot hits a tank 2. Spread of IR light - We need to test the spread of the light to make sure that only accurate shots aimed at the tanks will register as hits Test 1 will be done with the above test (IR Receiver). Test 2 will be done by moving the receiver left to right while shooting IR light and seeing the range that it picks up the signal. Our goal is to have a spread of no more than 10 degrees.

Design - Software:



Controlling the Tank: Once a game has been started, the application will have the user control the tank. Below is an example of what the user will see when controlling the tank. Each label indicates what each button will do



Add a google sign in section for the application: Once the application is loaded, there will be a button prompting the user to sign in using their google account. This will use Google's API. The application requires the user to have a pre existing google account.Users will be able to

sign in and out of their account. Users will also be able to disconnect their account with the application so if they decide to use another account.

Game Mechanics: Refer to the communication protocol for how the tank will base on who gets hit. The game will have the tanks going. A point will be given to the player

Testing:

Feature	Functions to Test
Sending controls to tanks	1. Type of controller - Joystick - Arrows (Similar to keyboard) 2. Sending multiple controls at a time - For example move and shoot at the same time Test 1 will be done after the model tanks have been built and we can decide which will suit our tanks best. We can try multiple controller types if we are unsure. Test 2 can be done by using LEDs in place of the real hardware and lighting one up for each different function. If we can light 2 LEDs simultaneously the test passes.
Receiving data from tanks	1. Check if tank was hit This test will show if a tank was hit and if one is hit, the other tank hit it. This covers all of the information we need from the tanks. We will need to test that we can receive a signal as well as where it came from.
Updating leaderboards	1. Receiving data 2. Data is sorted correctly - Specific data goes to correct location 3. Data can be organized - Used to show top players Test 1 will be done by sending test code sent from the Raspberry Pi. Then we can use this test code to send the information that we

	want to. Tests 2/3 can be done without the tanks. We can just test our code normally with test cases as the sorting and manipulation of data has nothing to do with the hardware.
User sign in/out	1. What sign-in procedure to use - Custom - Google - Facebook 2. Only one instance of the account can be open at a time We are using Flutter to design the app. We will likely use whichever is easiest to implement using that software. If we need specific information from the user, we may need to create our own custom login procedure.

Bibliography:

[1] Reading, Richard. “The Importance of Play in Promoting Healthy Child Development and Maintaining Strong Parent–Child Bonds.” *Child: Care, Health and Development*, vol. 33, no. 6, 2007, pp. 807–808., doi:10.1111/j.1365-2214.2007.00799_8.x.

[2]Reading, Richard. “The Importance of Play in Promoting Healthy Child Development and Maintaining Strong Parent–Child Bonds.” *Child: Care, Health and Development*, vol. 33, no. 6, 2007, pp. 807–808., doi:10.1111/j.1365-2214.2007.00799_8.x.

[3] “Controlling a Servo Motor.” *BlueJ*, BlueJ, www.bluej.org/raspberrypi/ServoMotor.html.

[4] Bacallao, Erick. “Stepper Motor from Windows 10 IoT Core.” *Hackster.io*, 15 Sept. 2015, www.hackster.io/erickbp/stepper-motor-from-windows-10-iot-core-d3c5d6.

[5] Isaax. “IR Control via LIRC on Raspberry Pi Zero W.” *Isaax Camp*, 5 Feb. 2018, camp.isaax.io/en/isaax-examples/ir-control-via-lirc-on-raspberry-pi-zero-w.