Interface Control Document (ICD) for Palomar CSCI 212 FINAL using osoyoo robot

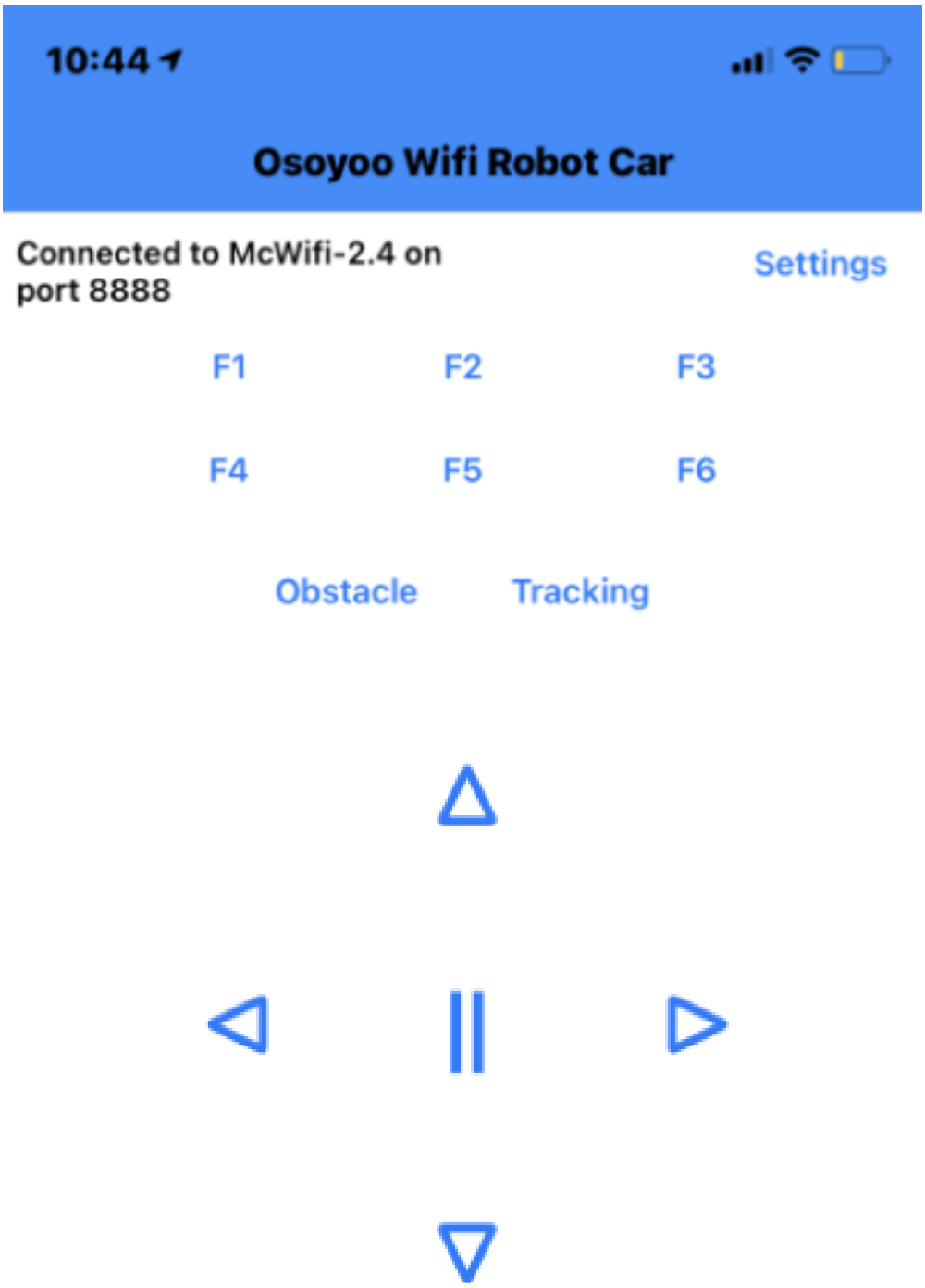
Version 1.2

Adam Amer
Timothy Liu
Cody McKinney
Nathan White

GROUP #3
05.29.2021

2.

USER INTERFACE / CONTROLLER



3.

Button	UDP message
F1	F
F2	G
F3	H
F4	I
F5	J
F6	K

▲	A
▼	B
▶	R
◀	L
square	E
obstacle	O
tracking	T

Controls for the osoyoo pi robot from iPhone or Android

△ - **Forward**. Both motors spin at an equal rate forward.
Returns ‘A’ in pi terminal.

```
case 'A': forward(fd, speed, speed); cur_status = p; break;
```

◁ - **Left**. The right motor spins at a higher rate turning the car left.
Returns ‘L’ in pi terminal.

```
case 'L': left(fd, speed, speed); delay(200); stop(fd); cur_status = p; break;
```

▷ - **Right**. The left motor spins at a higher rate turning the car right.
Returns ‘R’ in pi terminal.

```
case 'R': right(fd, speed, speed); delay(200); stop(fd); cur_status = p; break;
```

▽ - **Back**. Both motors spin at the same rate heading backwards.
Returns ‘B’ to the pi terminal.

```
case 'B': back(fd, speed, speed); cur_status = p; break;
```

⏸ - **Stop**. Stops all motor activity. Robot will stop all movement.
Returns ‘E’ to the pi terminal.

```
case 'E': stop(fd); cur_status = p; break;
```

4.

Obstacle - **Obstacle Avoid**. Obstacle Avoid button runs the function, avoid() which monitors object distance from 3 directions; left, right, center. If an object is within close proximity, the robot will avert the obstacle.

Returns an ‘O’ to the pi terminal.

```
case 'O': avoid(); break;
```

Tracking - **Follow**. Tracking button runs the function, follow() which scans left, right, and center facing forward to determine if an object is nearby. If an object is nearby, the robot will turn towards the object and attempt to get closer.

Returns a ‘T’ to the pi terminal.

```
case 'T': follow(); break;
```

F1 - **Increase speed**. F1 calls our increase_speed() method which will increment the robots speed by 200 each button press.

Returns an ‘F’ to the pi terminal.

```
case 'F': speed = increase_speed(speed); cur_status = p; break;
```

F4 - **Decrease speed**. F4 calls our decrease_speed method which will decrease the robots speed by 200 each button press.

Returns an ‘I’ to the pi terminal.

```
case 'I': speed = decrease_speed(speed); cur_status = p; break;
```