Objective- To have the computer write a tele-op program, that is based on user preferences, and still allow the program to be modified afterwards

Virtual Tele-Op Program

•••

By Neil

Background Information

- Teleoperation, long for tele-op, is the technical term for the remote control of a robot.
- In FTC, drivers can use Logitech or XBox controllers to control the robot.
- However, a tele-op program must be created in order to define each button's function on the controllers and the robot's motors.
- This program is processed through a phone at the Driver Station, which communicates directly with another phone on the robot.

“PrintStream” Class

- Since the tele-op program should be adjustable after it has been created, I have decided to write the source code into an external text file.
- In Java, in order to write a text file, the program has to import the PrintStream class.

```
PrintStream outfile = new PrintStream(new FileOutputStream("output.txt"));
```

class reference object Text file

reference Writes text

```
outfile.println(“”);
```

“Scanner” Class

- Initially, I used system input to write the tele-op program.
- In Java, in order to store user input, the program must import the Scanner class.

```
Scanner scanner= new Scanner(System.in);
```

class reference object System input

```
System.out.println("What would you like to call Motor?");
```

Ask user question

```
motorName = scanner.next();
```

Stores user’s answer into variable

Graphical User Interfaces (GUIs)

- For my second program, I decided to make the programming process more user-friendly by using graphics to configure the tele-op program.
- A GUI is basically a graphical frame that contains buttons and text fields (panels).
- To create a GUI, there has to be both a panel class, which defines the panels, and a driver class, which defines the frame itself (size, location of panels, etc.)

Graphical User Interfaces (GUIs)