FIRST FTC Coding Camp

By: Future Martians

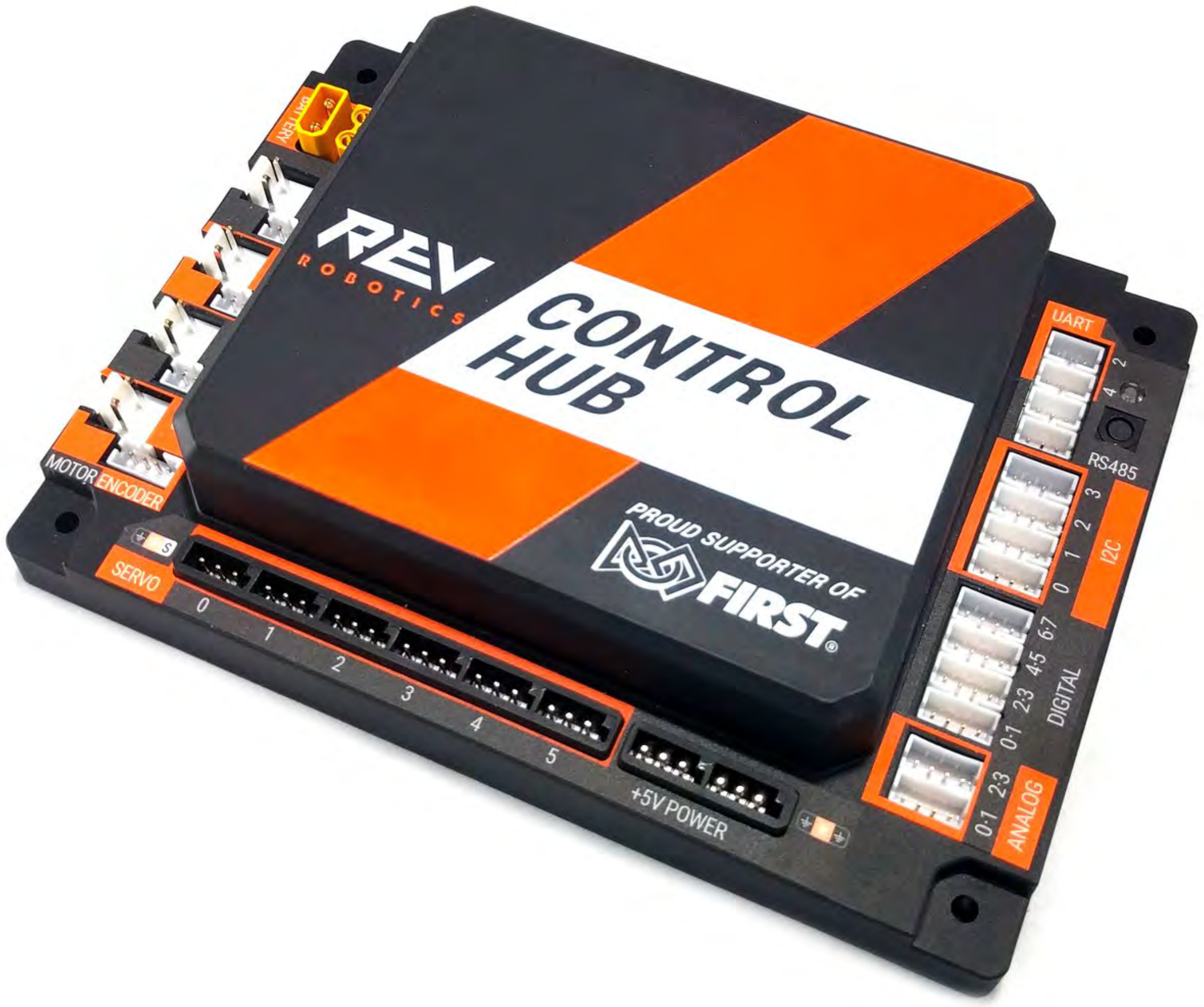
Who are we?

- *Welcome and thanks for joining our FTC Coding Camp today*
- *We are FRC team 9023, Future Martians!*
- *Last season, we competed in our first ever FIRST Robotics Competition as a Rookie team and learned some incredible things about engineering, tech skills, programming, and so much more!*
- *We had the amazing opportunity to make it all the way to the World championships in Houston, Texas and won the Rookie All-Star Award in our division.*
- *Most of the students on this team have experience with the other FIRST programs, such as FLL and FTC and have worked our way up to the Future Martians FRC team*

REV Hardware - Control Hub, Expansion Hub, Driver Hub, and REV Hardware Client

A quick overview of the Control Hub

- The Control Hub is the robot's "brain", controlling all the motors, sensors, and servos connected to it.
- The Control Hub runs the Android x86 Operating system, a common OS used in many phones and other devices
- The Control Hub has 4 DC motor ports, 6 servo ports, 8 digital I/O ports, an internal IMU, and 4 analog ports

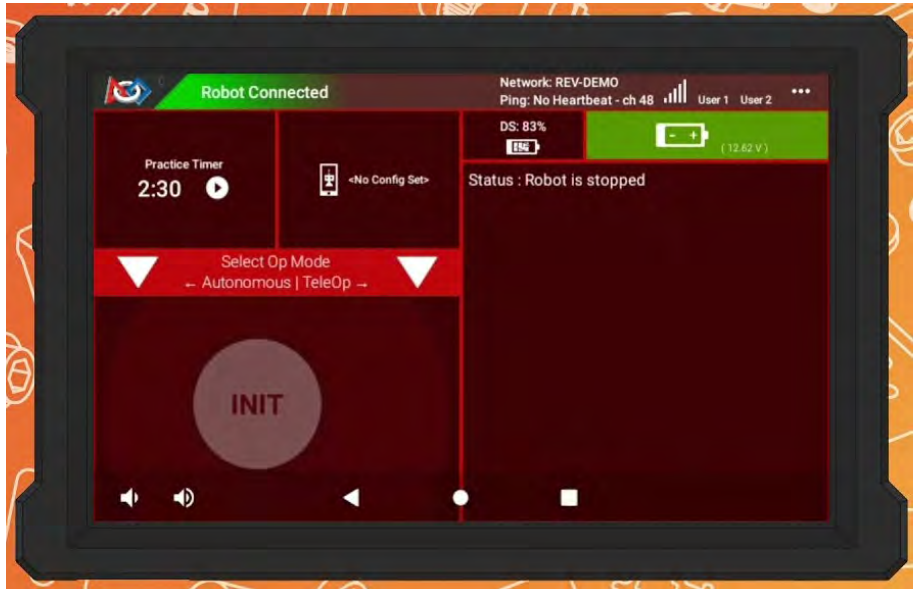


Expansion Hub

- The Expansion Hub is a hardware controller add-on to the Control Hub, and contains the same ports as the Control Hub, allowing for more motors, servos, and other devices to be connected.
- Before the introduction of the Control Hub, two Expansion Hubs were used along with an Android OS based cell phone which provided the wireless access. With the supply chain issues for Control Hubs resolved, we will not be working with the old hardware system.

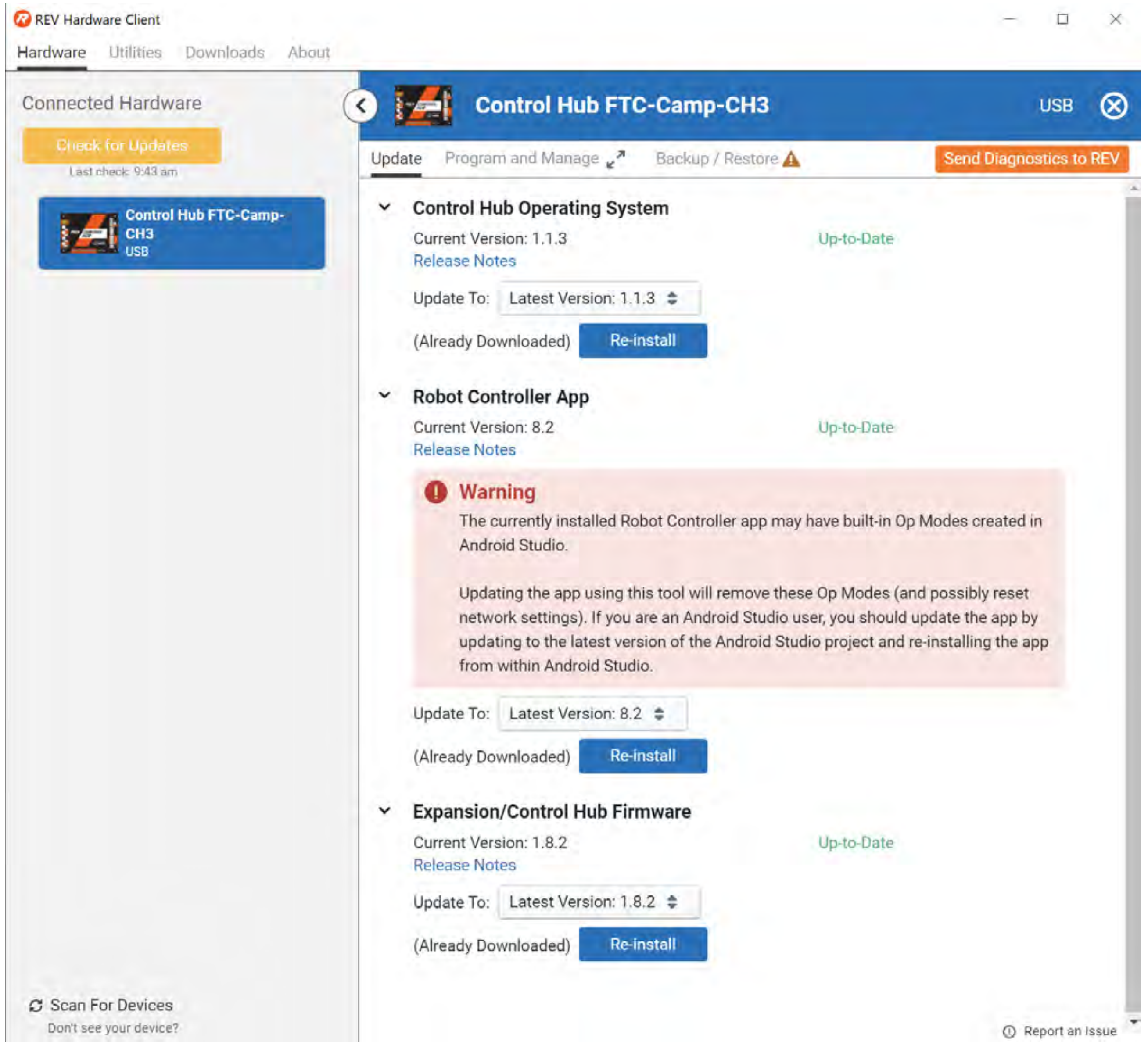
Driver Hub

- The Driver Hub is an Android OS based mobile device designed to interact wirelessly with the Control Hub. It runs a specialized app called "Driver Station" for interfacing with and configuring the Control Hub.
- This app provides several functionalities, including initializing and starting the robot, changing configurations, viewing a camera (if one is connected), and viewing logs and errors.
- The Driver Hub contains 3 USB-A Ports and up to two controllers can be plugged in.
- Prior to the introduction of the Driver Hub, an Android OS based cell phone was used to provide the wireless interaction with the other Android OS based cell phone on the Robot. We will not be working with this old hardware system.



REV Hardware Client

- REV Hardware client is an application that allows you to view and update the Control Hub and Expansion Hub.
- You can plug in your REV device into your computer, then choose from multiple options, including updating the REV device's firmware, backing up your device's files, and resetting the device.
- The Control Hub Operating system is the Android OS.
- The Robot Controller App will be updated by the FTC SDK version being used in Android Studio.
- The Control Hub Firmware fixes issues with the electronics.
- Link: https://docs.revrobotics.com/rev-hardware-client/



Control Hub Operating System
Current Version: 1.1.3 Up-to-Date
Release Notes
Update To: Latest Version: 1.1.3
(Already Downloaded) Re-install

Robot Controller App
Current Version: 8.2 Up-to-Date
Release Notes

Warning
The currently installed Robot Controller app may have built-in Op Modes created in Android Studio.

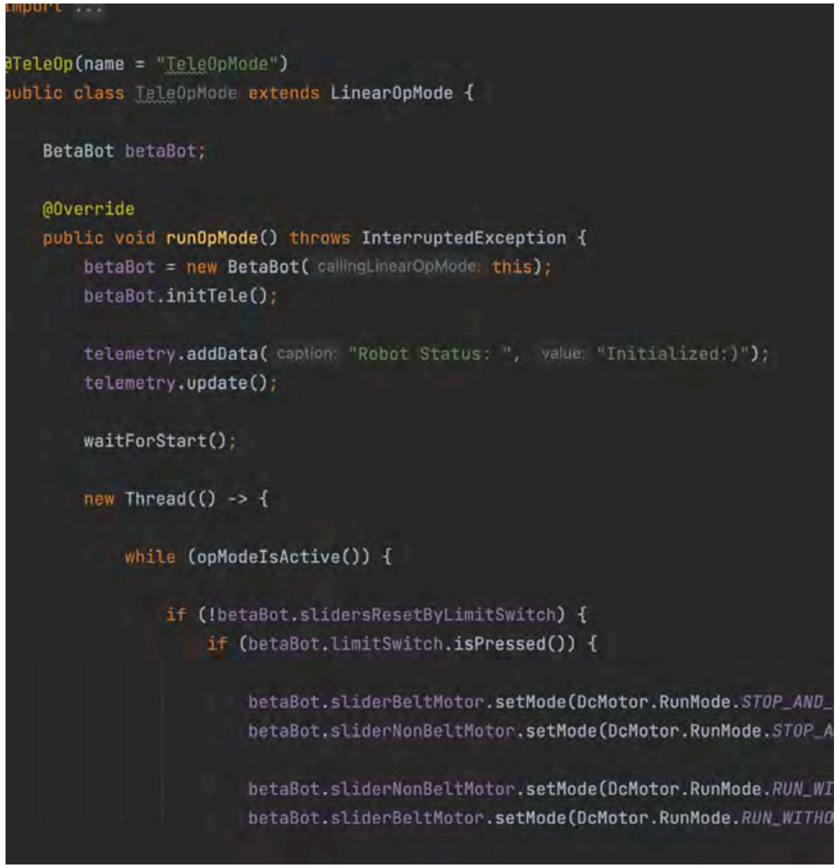
Updating the app using this tool will remove these Op Modes (and possibly reset network settings). If you are an Android Studio user, you should update the app by updating to the latest version of the Android Studio project and re-installing the app from within Android Studio.

Update To: Latest Version: 8.2
(Already Downloaded) Re-install

Expansion/Control Hub Firmware
Current Version: 1.8.2 Up-to-Date
Release Notes
Update To: Latest Version: 1.8.2
(Already Downloaded) Re-install

Introduction to FTC Coding

- Java is the programming language used for FTC. We will not be teaching Java in this camp. It will be helpful if you are already familiar with Java and Object-Oriented Programming, but this camp will focus on setting you up for being able to code with Java and allow you to run examples.
- There are 3 main options for coding: Android Studio, Onbot java, and block coding (we will mainly discuss Android Studio, and touch on Onbot Java)
- Github is used for source control and collaboration among the team.



```
import ...

@TeleOp(name = "TeleOpMode")
public class TeleOpMode extends LinearOpMode {

    BetaBot betaBot;

    @Override
    public void runOpMode() throws InterruptedException {
        betaBot = new BetaBot( callingLinearOpMode: this);
        betaBot.initTele();

        telemetry.addData( caption: "Robot Status: ",  value: "Initialized:)");
        telemetry.update();

        waitForStart();

        new Thread(() -> {

            while (opModeIsActive()) {

                if (!betaBot.slidersResetByLimitSwitch) {
                    if (betaBot.limitSwitch.isPressed()) {

                        betaBot.sliderBeltMotor.setMode(DcMotor.RunMode.STOP_AND_
                        betaBot.sliderNonBeltMotor.setMode(DcMotor.RunMode.STOP_A

                        betaBot.sliderNonBeltMotor.setMode(DcMotor.RunMode.RUN_WI
                        betaBot.sliderBeltMotor.setMode(DcMotor.RunMode.RUN_WITHO
```

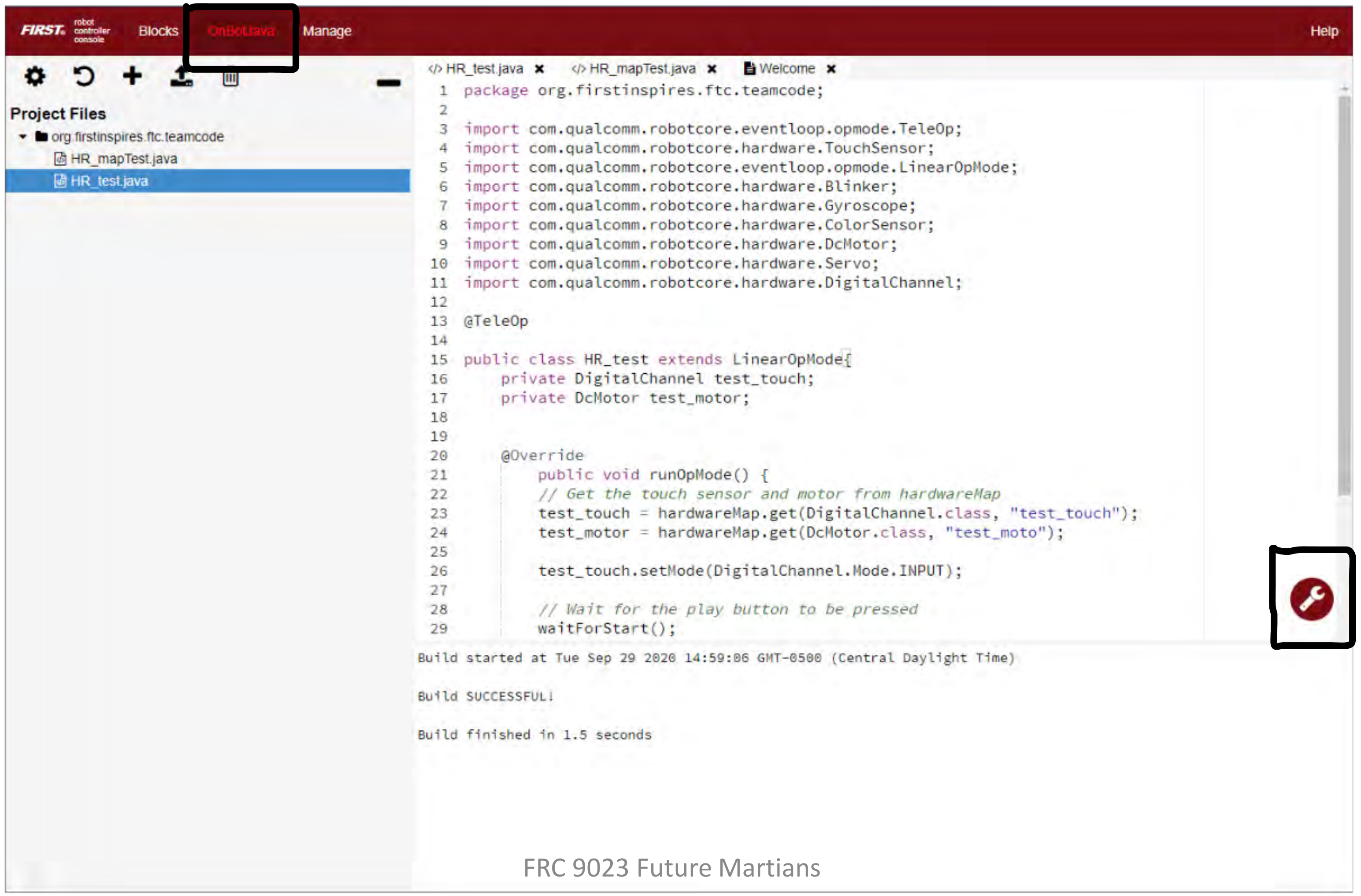
What is OnBot Java?

- OnBot Java is a programming tool provided by FIRST
- It is used on a web browser, so there is no app installation required
  - FIRST recommends using Google Chrome
- It is hosted on the Control Hub
  - So it allows you to write code directly onto the device without having to download it on your computer first
- You only need to press a button to compile the code to make it available to run from the Driver Station
- The FTC SDK and Java tools are available on OnBot Java without any additional download
- OnBot Java is very easy to collaborate on; everyone can access updated versions almost instantly

How does OnBot Java work?

- You first need to **connect to your Control Hub** from your computer
  - *Enter "192.168.43.1:8080" into your browser to open OnBot Java*
  - Click the "OnBot Java" tab at the top of the screen
- Create a new file and start coding
  - There are certain templates available for this: you can either choose an Autonomous OpMode or a TeleOp
  - These templates will set up the necessary configurations in the code for it to appear and run correctly on the Driver Station
- Once you are ready to run, click the **build** button in the bottom right corner of the screen, and it will compile your code so it can be available to run from the Driver Station

How does OnBot Java look?



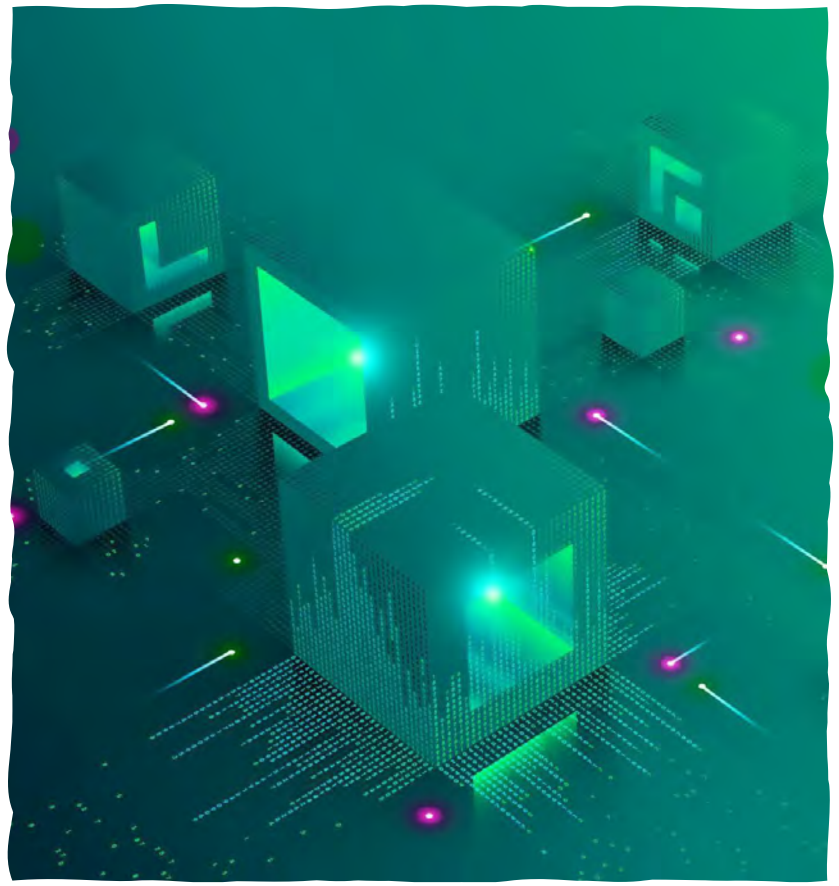
```
package org.firstinspires.ftc.teamcode;

import com.qualcomm.robotcore.eventloop.opmode.TeleOp;
import com.qualcomm.robotcore.hardware.TouchSensor;
import com.qualcomm.robotcore.eventloop.opmode.LinearOpMode;
import com.qualcomm.robotcore.hardware.Blinker;
import com.qualcomm.robotcore.hardware.Gyroscope;
import com.qualcomm.robotcore.hardware.ColorSensor;
import com.qualcomm.robotcore.hardware.DcMotor;
import com.qualcomm.robotcore.hardware.Servo;
import com.qualcomm.robotcore.hardware.DigitalChannel;

@TeleOp

public class HR_test extends LinearOpMode{
    private DigitalChannel test_touch;
    private DcMotor test_motor;


    @Override
        public void runOpMode() {
        // Get the touch sensor and motor from hardwareMap
        test_touch = hardwareMap.get(DigitalChannel.class, "test_touch");
        test_motor = hardwareMap.get(DcMotor.class, "test_moto");

        test_touch.setMode(DigitalChannel.Mode.INPUT);

        // Wait for the play button to be pressed
        waitForStart();
```

```
Build started at Tue Sep 29 2020 14:59:06 GMT-0500 (Central Daylight Time)

Build SUCCESSFUL!

Build finished in 1.5 seconds
```

Advantages and Disadvantages of OnBot Java

Advantages	Disadvantages
• No app installation required • Whereas in order to use Android Studio, you need to install three applications: Android Studio, FTC SDK, and Java. • Very easy to collaborate on • Build time is very quick; compiles code directly onto the Control Hub • Whereas Android Studio first compiles the code into a file that gets downloaded onto your laptop and then you need to put it onto the Control Hub	• Cannot use Git or GitHub – hard to track changes to your code • Has limited capability to load external libraries • Need to be connected to the Control Hub while programming • There will be no internet connection • Cannot code if you are away from the Control Hub – however, you can download the code while you are connected and work on it later asynchronously

What is Android Studio?



- Android Studio is an IDE (Integrated Development Environment) that allows you to code applications for Android OS.
- If you remember from previous slides, the Control Hub runs Android OS, meaning that the use of Android studio is the only way we can put code onto the Control Hub.

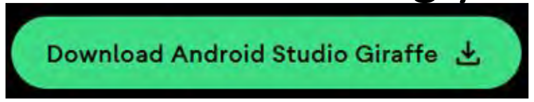
What do we need to install?

1. Android Studio - https://developer.android.com/studio
2. Java SDK (optional since Android Studio includes it) - https://www.oracle.com/java/technologies/downloads/#java17
3. FTC Software Development Kit* - https://github.com/FIRST-Tech-Challenge/FtcRobotController

*The FTC SDK requires a different installation process that we will go over in later slides.

Download Android Studio

Downloading Android Studio is a simple but long process.

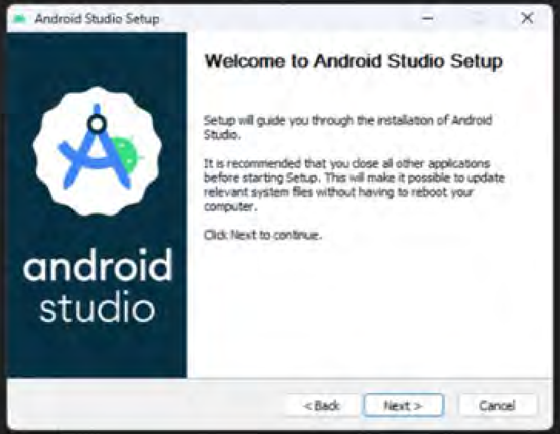
1. Navigate to the website in the previous slide.
2. You will see a large green button asking you to install the latest version of android studio. **CLICK IT** 

3. After accepting the Terms of Service, wait for the executable file to install, then double click on it in your Downloads folder or whatever location you downloaded it too.
4. Wait for the installation window to open

If you are having ANY problems, please tell us

Installing Android Studio

- When you first install, a UAC prompt may show up. If you do not have admin access to your computer, you will NOT be able to install Android studio.
- Keep clicking next until you see an install button. Click Install. Android Studio will take some time to install depending on your computer.
- Click next and finish once it finished installing.

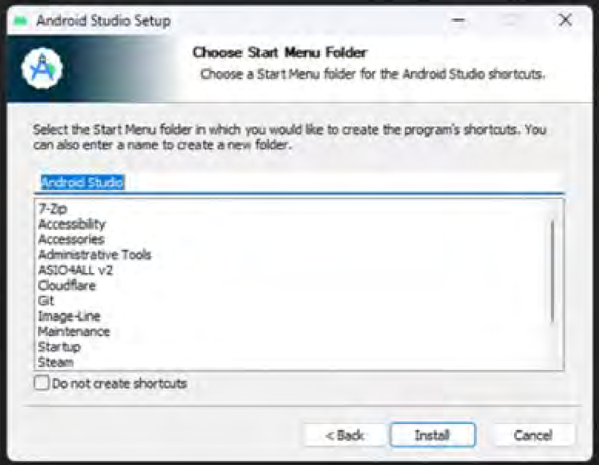


Welcome to Android Studio Setup

Setup will guide you through the installation of Android Studio.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.



Choose Start Menu Folder

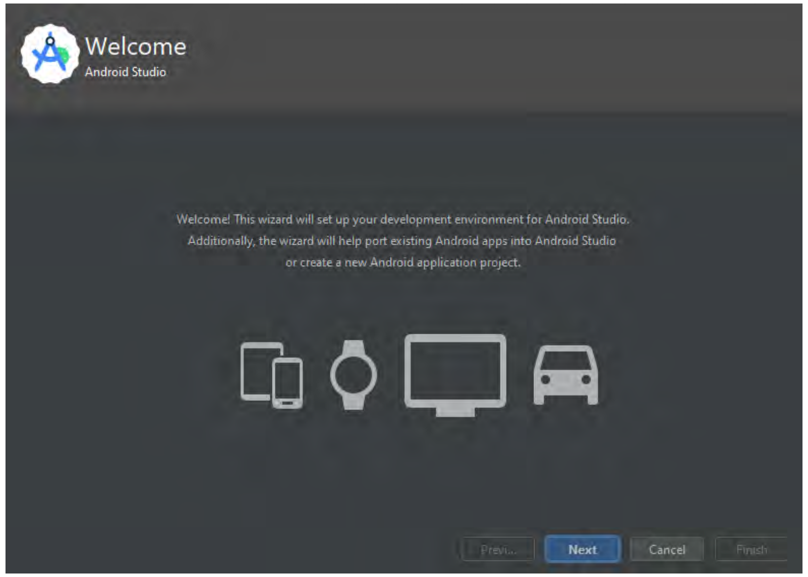
Choose a Start Menu folder for the Android Studio shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

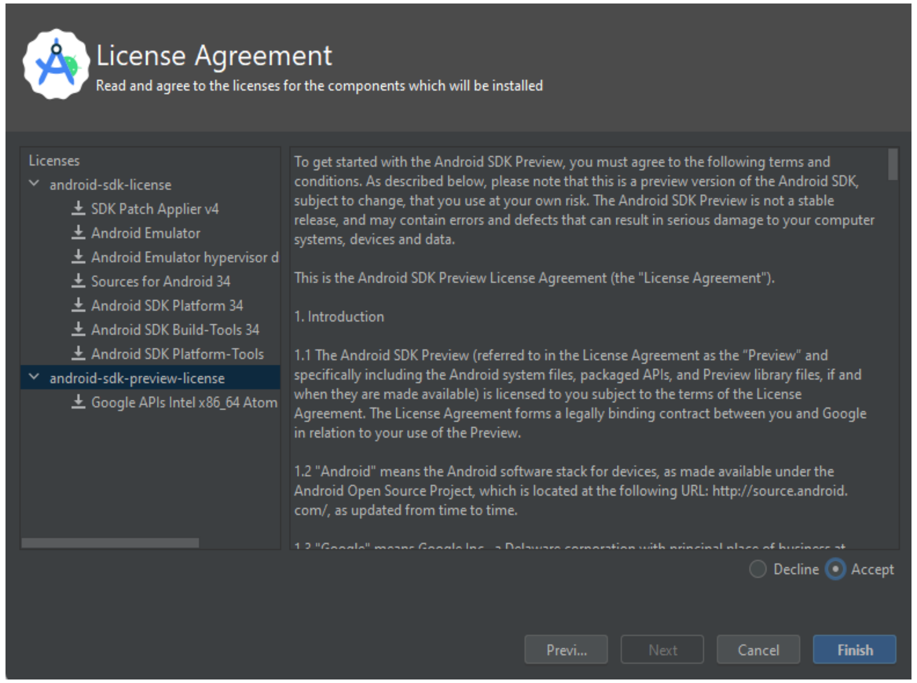
Android Studio

7-Zip
Accessibility
Accessories
Administrative Tools
ASIO4ALL v2
Cloudflare
Git
Image-Line
Maintenance
Startup
Steam

☐ Do not create shortcuts



Welcome! This wizard will set up your development environment for Android Studio.
Additionally, the wizard will help port existing Android apps into Android Studio
or create a new Android application project.


License Agreement
Read and agree to the licenses for the components which will be installed
Previ...
Next
Cancel
Finish

Licenses
- android-sdk-license
  - SDK Patch Applier v4
  - Android Emulator
  - Android Emulator hypervisor d
  - Sources for Android 34
  - Android SDK Platform 34
  - Android SDK Build-Tools 34
  - Android SDK Platform-Tools
- android-sdk-preview-license
  - Google APIs Intel x86_64 Atom

To get started with the Android SDK Preview, you must agree to the following terms and conditions. As described below, please note that this is a preview version of the Android SDK, subject to change, that you use at your own risk. The Android SDK Preview is not a stable release, and may contain errors and defects that can result in serious damage to your computer systems, devices and data.

This is the Android SDK Preview License Agreement (the "License Agreement").

1. Introduction

1.1 The Android SDK Preview (referred to in the License Agreement as the "Preview" and specifically including the Android system files, packaged APIs, and Preview library files, if and when they are made available) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and Google in relation to your use of the Preview.

1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: http://source.android.com/, as updated from time to time.

Decline [✓ Accept]

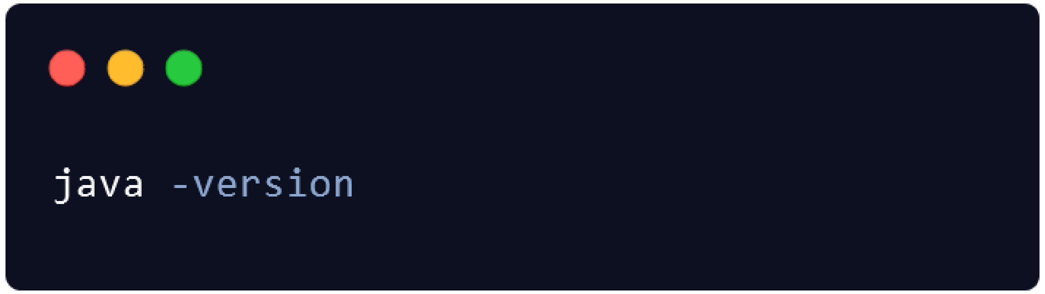
Installing Android Studio Continued

- Once finished installing, Android Studio will open, but you're not done yet.
- Click "Ok" on the Import Settings Screen, then click "Don't Send" on the Help improve Android Studio Screen.
- You will see a screen saying "Welcome", keep clicking next on it. Choose "Standard Install" and click next. Pick which theme you want your IDE to be and click next. Click next on the Android SDK screen, unless you want to change its install location. Keep clicking next and Accept all the terms and services. Then click "Finish". Android Studio will install more files, and this will take some time.

<u>IF YOU ARE HAVING ANY ISSUES, PLEASE TELL US</u>

Installing Java 17

- Go to this link: https://www.oracle.com/java/technologies/downloads/#java17 and download Java 17 for your specific operating system
- Open the installer and follow the instructions to install Java onto your computer. (A UAC prompt may open)
- To see if Java is installed, open your terminal or command prompt, and run this command. -------------------------->



```
java -version
```

Installing FTC Java SDK

- Open up: https://github.com/FIRST-Tech-Challenge/FtcRobotController/releases/tag/v8.2
- Click on the Green "Code" button in the middle of the screen, and then click "Download ZIP".
- Once the ZIP downloads, extract it.
- Open Android Studio, click open, and choose the FTC Robot Controller folder. This should open a new window.

It's important to note that the FTC SDK is a preseason version and may change once the 2023-24 season starts.

Git, Github, and Source-Control

What is Git and Github?

- Git is a source-control system that is used to track changes in your code over time.
- GitHub is the online cloud provider that hosts the changes in your Git Folders (Called Repositories)
- Why do we use Git?
- If we assume that I make changes to the code, and I tell everyone except Rishi about the changes, then he will not know that we changed the code.
- If I put my changes to Git, and upload them to Github, all Rishi has to is to is check Github for changes.

Key Terms to know

Repository (Repo) - A collection of files combined with their history that are managed by Git.

Commit – A group of changes made to files along with a description of the changes.

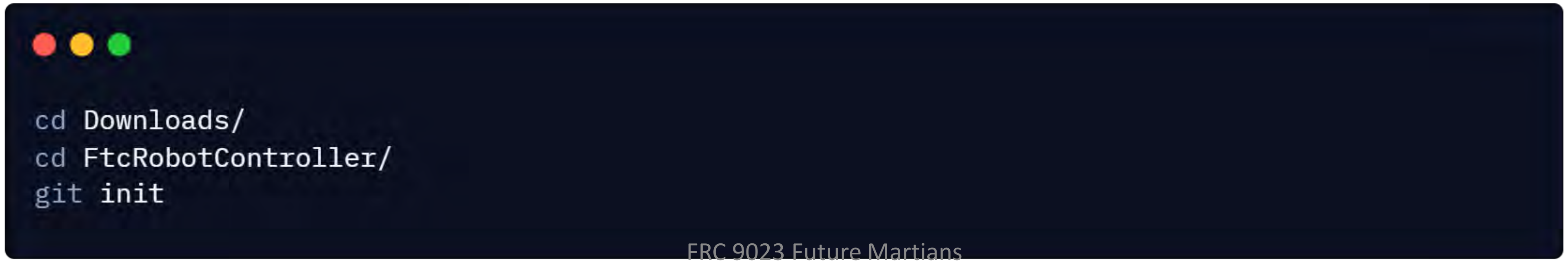
Clone - Creating a Local Copy of a repository on your computer.

Push - Uploading your Commits to Github

Pull - Getting changes from Github and downloading them

Installing Git and creating your first Repo

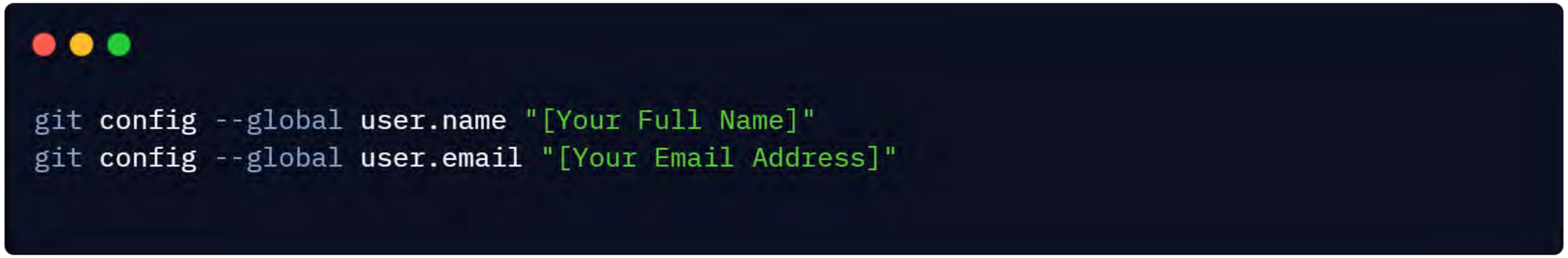
- First of all, we need to install Git.
  - If you are on Windows, go to the following link to install Git.
  - https://git-scm.org
  - If you are on MacOS or Linux, you already have Git installed by default.
- Second after installing Git
  - Open up Git Bash on your Computer (Windows Only)
  - Open Up Terminal on your Computer (MacOS and Linux Only)
  - Follow the commands below:



```
cd Downloads/
cd FtcRobotController/
git init
```

Some Simple Setup

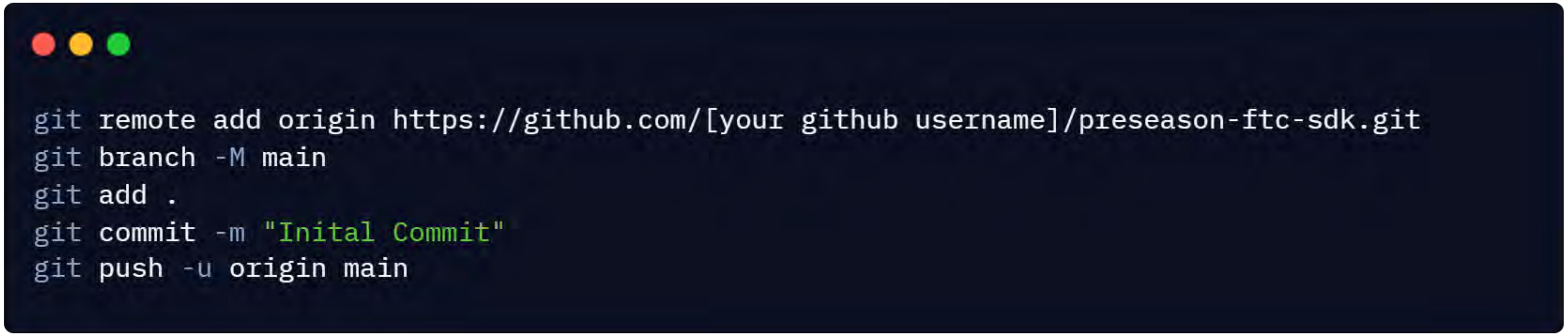
- There are a few setup things you need to do before you can continue using Git and Github effectively.
- First of all, you need to sign up for a GitHub account, if you have not done so already. You can do so here: https://github.com/signup
- After you have done this you need to run the following commands in your Terminal:



```
git config --global user.name "[Your Full Name]"
git config --global user.email "[Your Email Address]"
```

Using Github

- Now, navigate to GitHub on your web browser and if you are logged in, in the left-hand corner, there should be an option that says, “New”.
- After Choosing New, a screen showing settings for a new Repository should open. Enter a name (set it to “preseason-ftc-sdk”) and set it to Private. Click Create.
- Now run the following commands:



```
git remote add origin https://github.com/[your github username]/preseason-ftc-sdk.git
git branch -M main
git add .
git commit -m "Inital Commit"
git push -u origin main
```

- It will ask you to log in during these steps. We will show you how to log in in the next slide.

Logging into Git

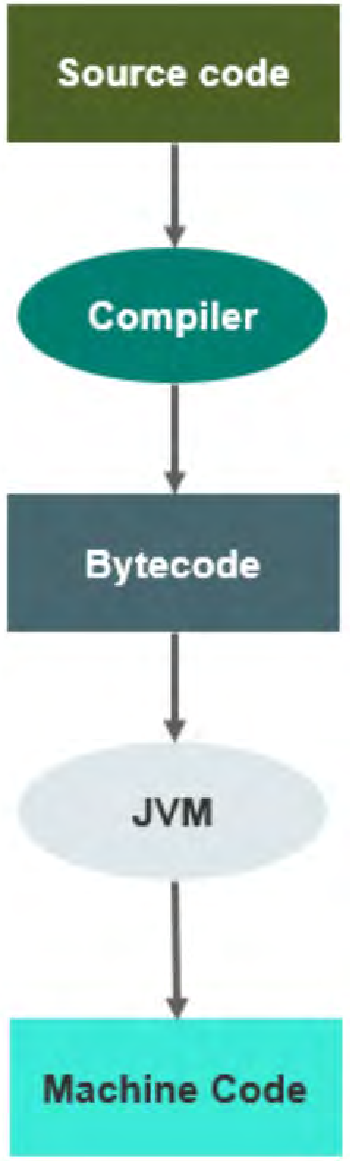
- When running some commands in Git for the first time, it may ask you to log in.
- The First option should be your username. Enter your Github Username.
- Now, in the second option don’t enter your password. Instead:
    - Open Github.com
    - Click on your Profile Picture in the Top Right Corner.
    - Go all the way down and select Settings
    - Now, once the Settings Page has opened, on the left hand side, scroll down and choose "Developer Settings"
    - After that, choose "Personal Access Tokens (classic)" and then click "New".
    - Now, in this screen, Enter any Name you want for the name field and check all the checkboxes.
    - Finally, click Continue.
    - Now, copy the token that it has generated and paste it in the Password Field.
- Now retry the command that you tried earlier and it should work.

Git and Android Studio

- Open up Android Studio on your computer.
- Click the Open Button, navigate to your Downloads, and choose the “FtcRobotController” Folder. Wait until the project opens.
- Now, wait for about a minute, while Android Studio sets up the project.
- After that, head over to the “Commit” Tab in the left hand side of the screen.
- You should see 5 different files here, and at the bottom, an option for a “Commit Message”. Choose the checkbox and select each of the five files.
- In the Commit Message, type “Project Setup” and then click the “Commit and Push” button.
- Now, check Github and you should see the files added to Github.

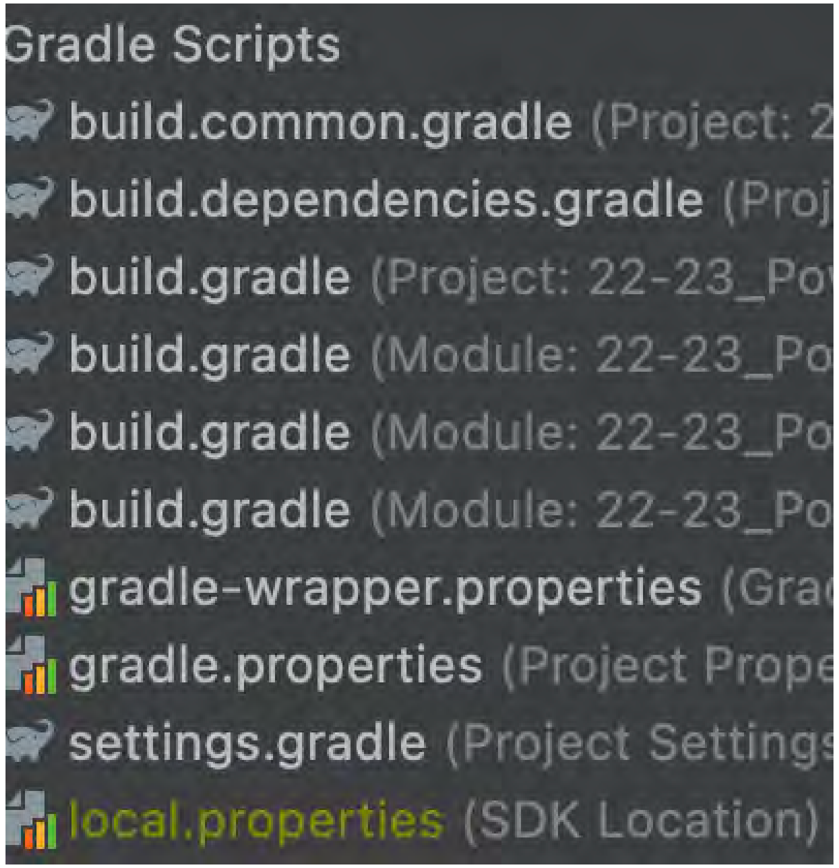
Java – Compiling and Running code

- After a Java source file (example: "Test.java") has been created, it needs to be compiled. The compiler compiles the code in that file to generate a Bytecode file ("Test.class"). This is done using the "javac" command of the Java Software Development Kit (Java SDK).
  ***"javac Test.java"* compiles it to create Test.class**
- This bytecode file can be executed on any platform (Operating System - OS), but it cannot simply be run on the OS.
- Execution of the bytecode ("Test.class" file) requires an interpreter which will convert the bytecode instructions to instructions for the OS to execute.
  ***"java Test"* runs the bytecode Test.class**
- The JVM (Java Virtual Machine) is the interpreter that can run the bytecode generated after compilation, by interpreting it into machine level instructions for the OS.



Gradle

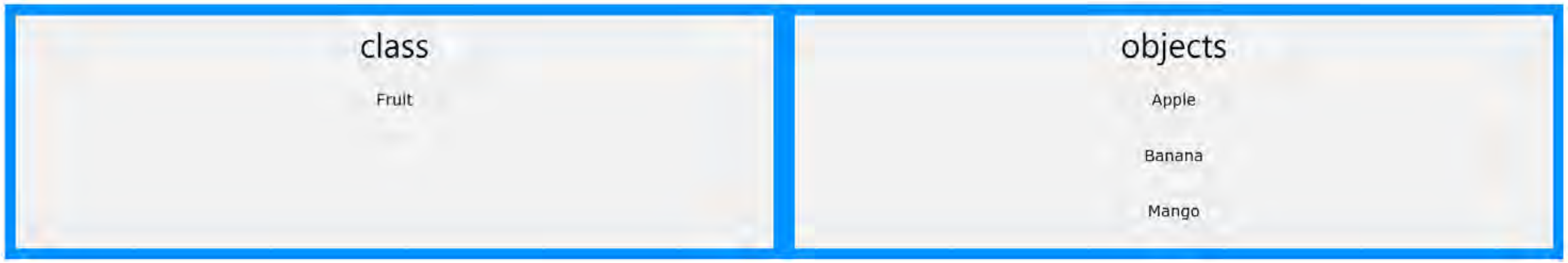
- The most common processes in building any software include compiling the source code, and packaging the compiled output to a compressed format like apk(android), JAR, TAR, ZIP, etc.
- We might also perform some other tasks like publishing the artifacts to some repository or transfering the apk to the Android device
- If there is no build tool to automate this process; then, we will need to perform all of these actions manually, which would be slow and have chances of human errors.
- Gradle is the build tool used in the FTC SDK to automate these processes


Gradle Scripts
build.common.gradle (Project: 2
build.dependencies.gradle (Proj
build.gradle (Project: 22-23_Po
build.gradle (Module: 22-23_Po
build.gradle (Module: 22-23_Po
build.gradle (Module: 22-23_Po
gradle-wrapper.properties (Gra
gradle.properties (Project Prope
settings.gradle (Project Settings
local.properties (SDK Location)

Object Oriented Programming

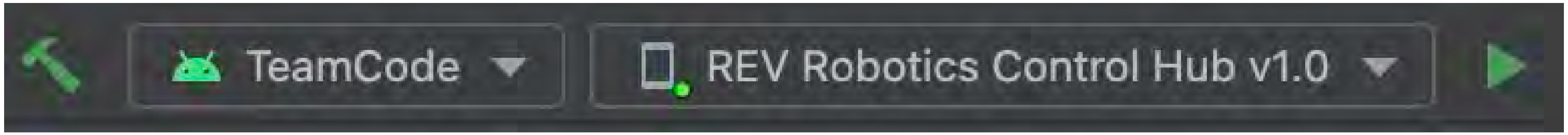
- Object Oriented Programming (OOP) involves classes and objects, along with their attributes and methods. Java uses OOP.
- Example: In real life, an **Apple** is an **object** and it is an instance of a **Fruit** which would be the **class**. The Apple has **attributes**, such as **weight** and **color**, and **methods** are what can be done by the fruit, such as **grow** and **rot**.

  If **Jill** (an object of class **Person**) is **eating** the apple, then "**eat**" is a **method** that the class "**Person**" has.
- A Class is like an object constructor, or a "blueprint" for creating objects.
- Look at the following illustration to see the difference between class and objects:



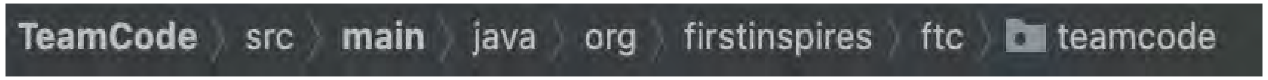
Downloading Code for the First Time

- Connect a cable into the usb c port on the control hub
- Connect the other usb end into your computer
- Click the build button on the application menu at the top of your screen(You don't have to do this everytime)
- After the build finishes click the run button
- You just downloaded code for the first time :)!!!

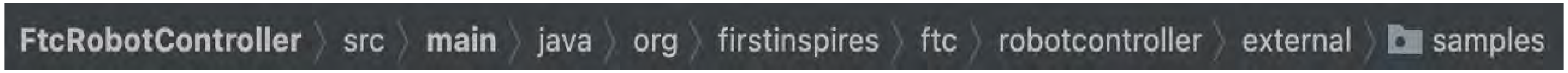


Navigating the SDK

- You will be writing your own custom code here:

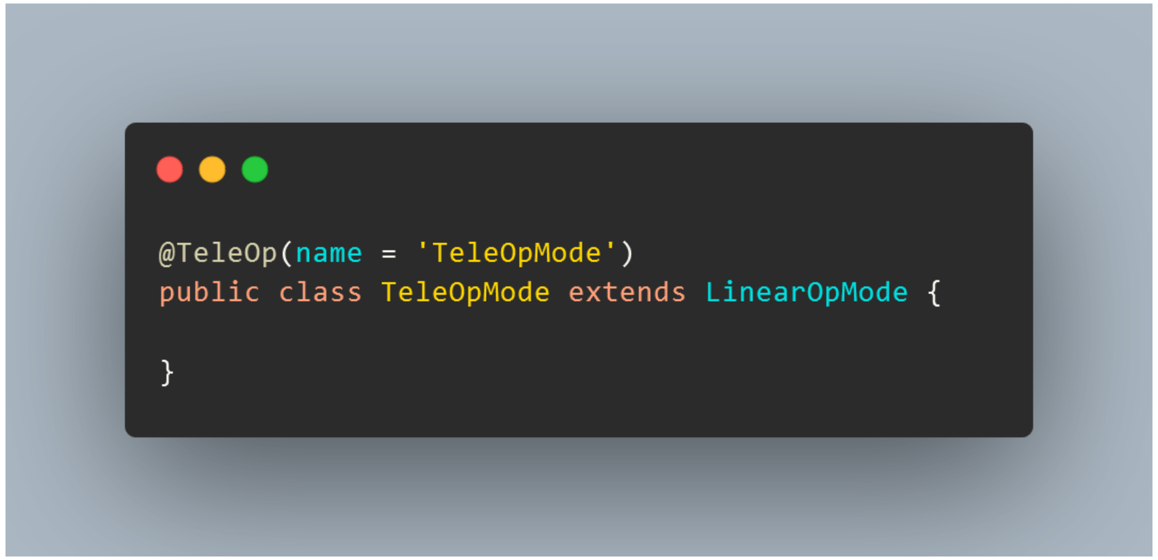

TeamCode 〉 src 〉 main 〉 java 〉 org 〉 firstinspires 〉 ftc 〉 teamcode

- You can view sample code here:


FtcRobotController 〉 src 〉 main 〉 java 〉 org 〉 firstinspires 〉 ftc 〉 robotcontroller 〉 external 〉 samples

- Sample code is a very powerful resource, as it provides examples of various coding applications such as moving a motor, vision processing, etc.

Stages of a Program and OpMode Classes



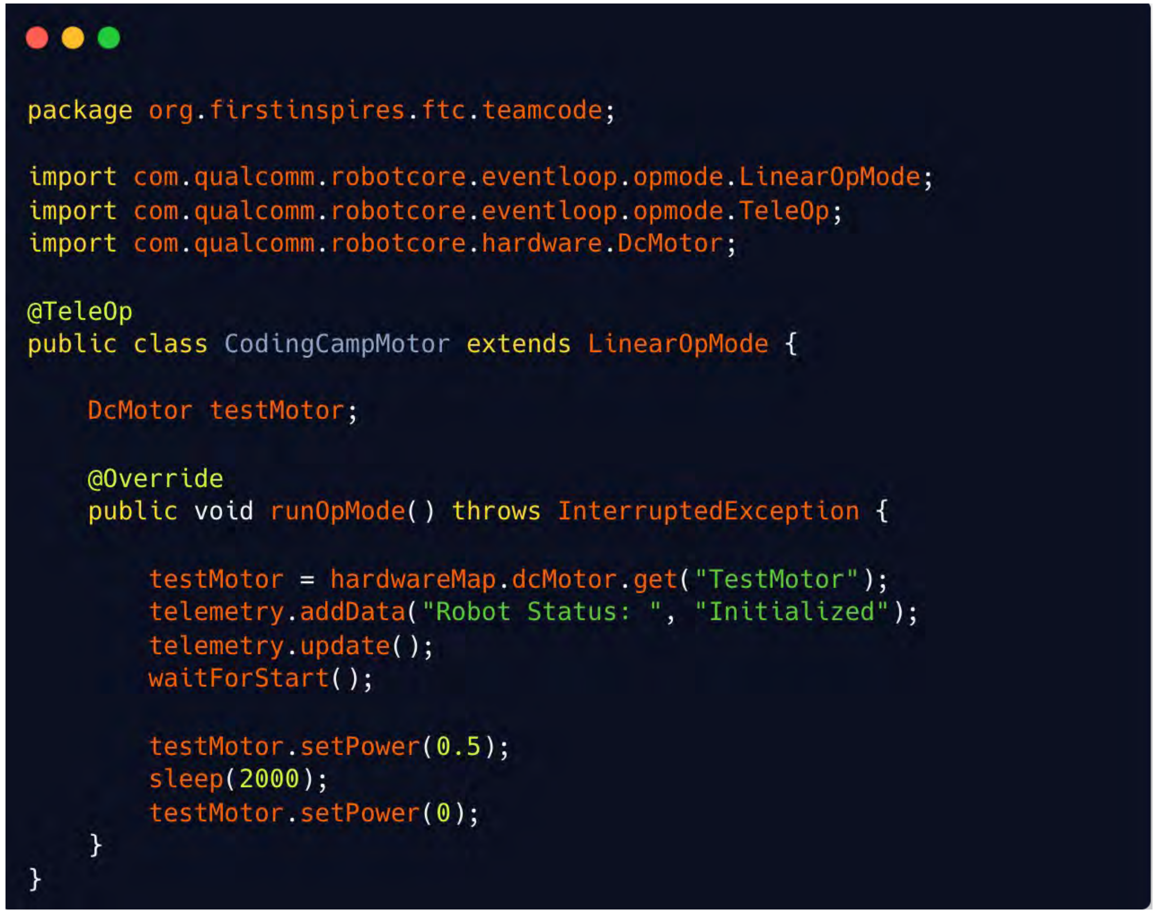
```
@TeleOp(name = 'TeleOpMode')
public class TeleOpMode extends LinearOpMode {

}
```

- The FTC SDK offers two Opmode classes: LinearOpMode and OpMode, we will be using LinearOpMode in this camp.
- The initialization stage is where you run all the things necessary to setup your robot before you press play (Motor Configs, Webcam setup, etc.)
- The start stage is where you run your actual Auto or TeleOp programs.

Resources: https://gm0.org/en/latest/docs/software/getting-started/linear-opmode-vs-opmode.html

Writing your First Program

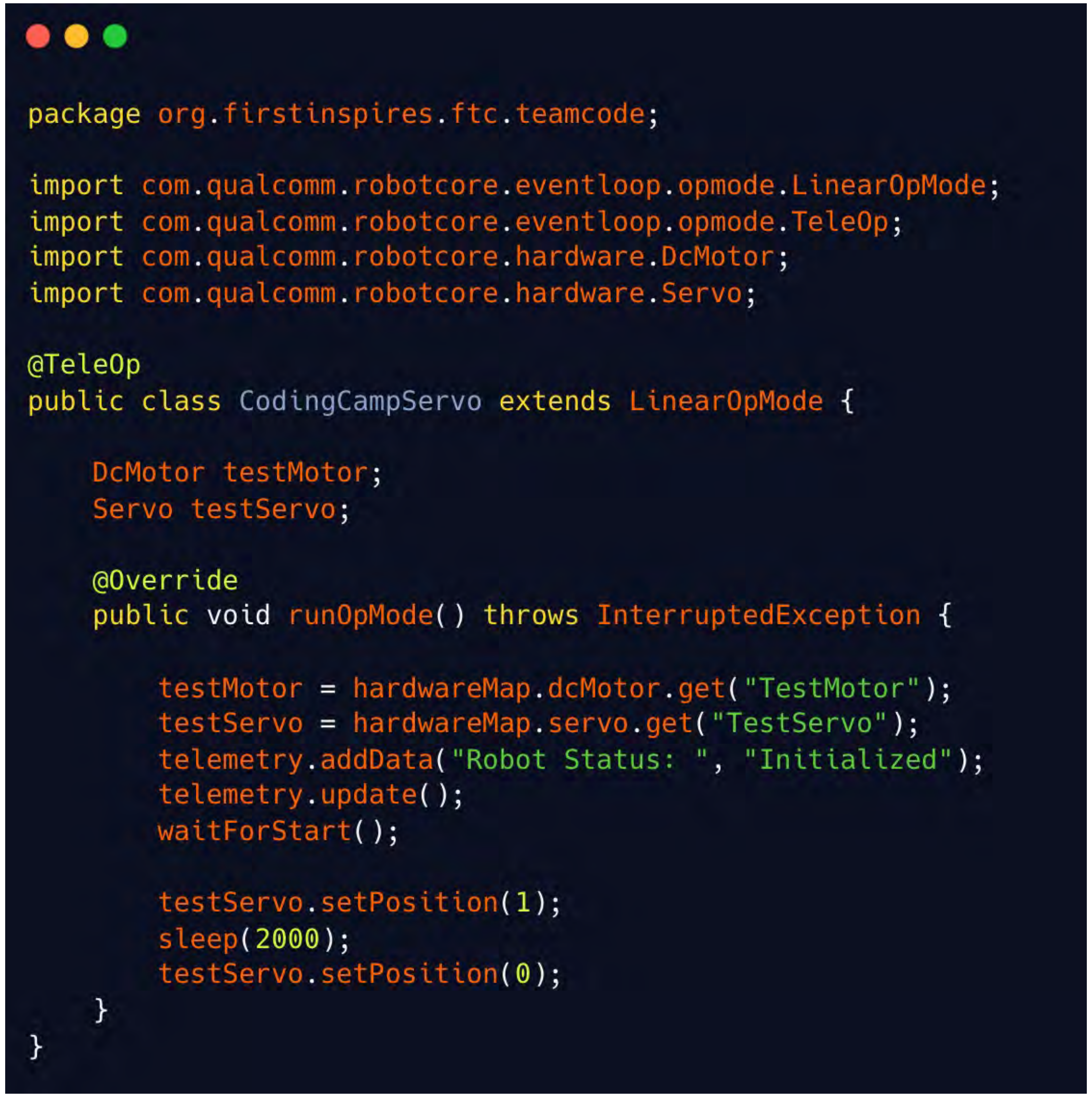
- Extend LinearOpMode
- Implement RunOpMode
- Add @TeleOp above Class
- "DcMotor testMotor;" Outside of RunOpMode method
- "testMotor = hardwareMap.dcMotor.get("TestMotor");" inside RunOpMode method
- Add telemetry to show init stage is complete
- "waitForStart();" under telemetry
- Add code to start a motor at 50% speed, wait 2 seconds, then stop the motor
- You Just Wrote Your First Program:)!!!



```java
package org.firstinspires.ftc.teamcode;

import com.qualcomm.robotcore.eventloop.opmode.LinearOpMode;
import com.qualcomm.robotcore.eventloop.opmode.TeleOp;
import com.qualcomm.robotcore.hardware.DcMotor;

@TeleOp
public class CodingCampMotor extends LinearOpMode {

    DcMotor testMotor;

    @Override
    public void runOpMode() throws InterruptedException {

        testMotor = hardwareMap.dcMotor.get("TestMotor");
        telemetry.addData("Robot Status: ", "Initialized");
        telemetry.update();
        waitForStart();

        testMotor.setPower(0.5);
        sleep(2000);
        testMotor.setPower(0);
    }
}
```

Coding Servos

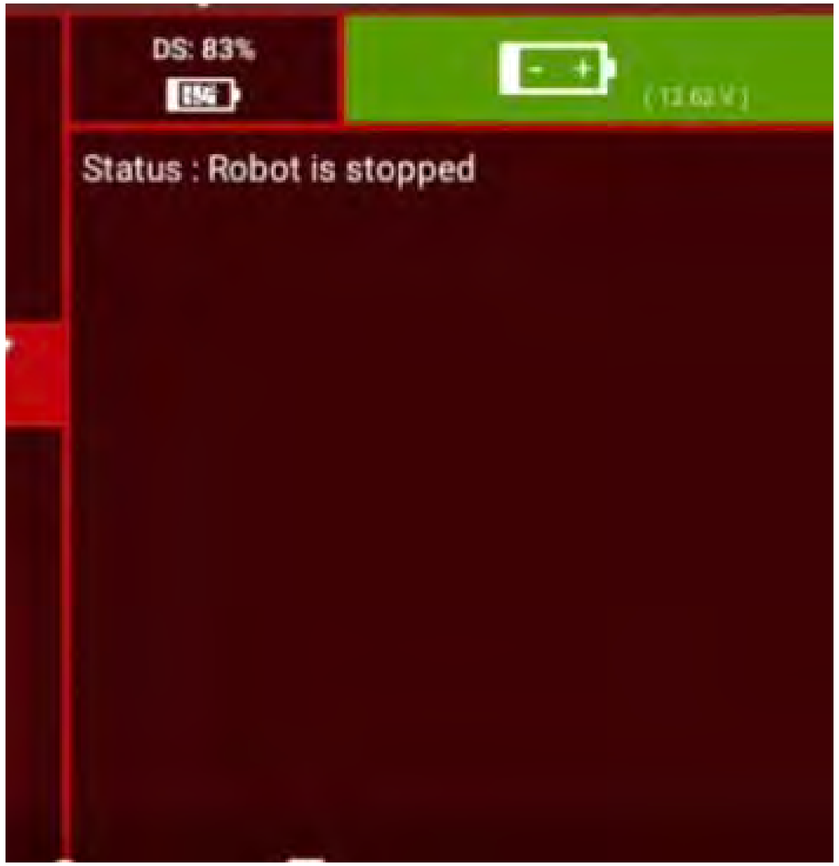
1. Create Servo object
2. Add it to the hardware map
3. Change the motor moving code to servo moving code



```java
package org.firstinspires.ftc.teamcode;

import com.qualcomm.robotcore.eventloop.opmode.LinearOpMode;
import com.qualcomm.robotcore.eventloop.opmode.TeleOp;
import com.qualcomm.robotcore.hardware.DcMotor;
import com.qualcomm.robotcore.hardware.Servo;

@TeleOp
public class CodingCampServo extends LinearOpMode {

    DcMotor testMotor;
    Servo testServo;

    @Override
    public void runOpMode() throws InterruptedException {

        testMotor = hardwareMap.dcMotor.get("TestMotor");
        testServo = hardwareMap.servo.get("TestServo");
        telemetry.addData("Robot Status: ", "Initialized");
        telemetry.update();
        waitForStart();

        testServo.setPosition(1);
        sleep(2000);
        testServo.setPosition(0);
    }
}
```

Telemetry

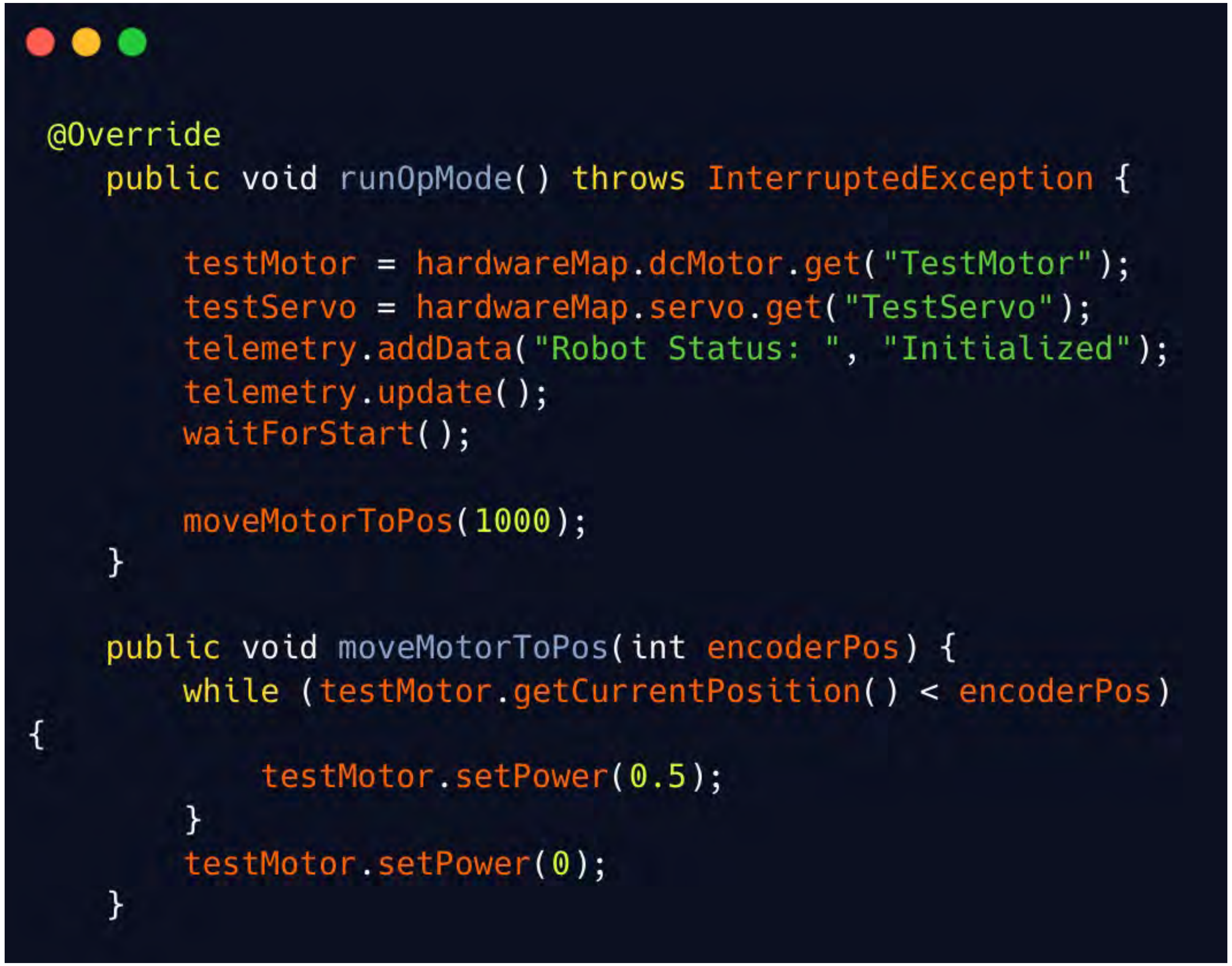
- In previous slides you might have seen us use telemetry to know when the init stage was complete, this one of the many ways we can use telemetry to help us in our programming.
- Telemetry provides a logging system so we can constantly see the value of variables throughout a program
- This can help with debugging, tuning, and many other things
- Telemetry shows up on the driver hub screen
- In a program you can use telemetry.addData() to add data, and telemetry.update() to make it show up on the driver hub screen



Status : Robot is stopped

Coding a Motor with Encoders

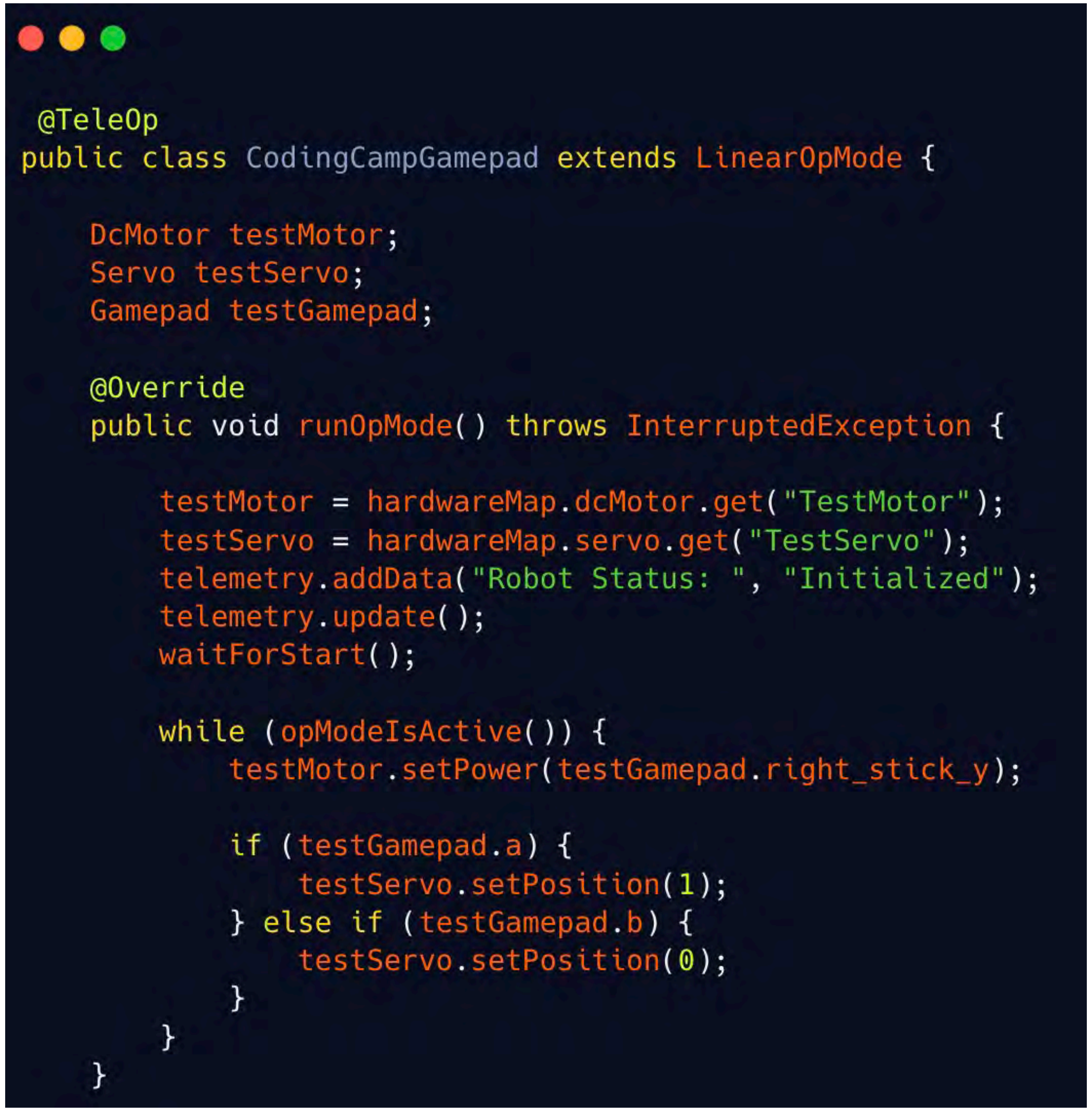
- Create moveMotorToPos Method
- Use a while loop to spin the motor at 50% speed until it reaches the target pos
- Stop the motor once it reaches the target pos
- Replace servo movement code with this method



```
@Override
    public void runOpMode() throws InterruptedException {

        testMotor = hardwareMap.dcMotor.get("TestMotor");
        testServo = hardwareMap.servo.get("TestServo");
        telemetry.addData("Robot Status: ", "Initialized");
        telemetry.update();
        waitForStart();

        moveMotorToPos(1000);
    }

    public void moveMotorToPos(int encoderPos) {
        while (testMotor.getCurrentPosition() < encoderPos)
{
            testMotor.setPower(0.5);
        }
        testMotor.setPower(0);
    }
```

Programming a Gamepad

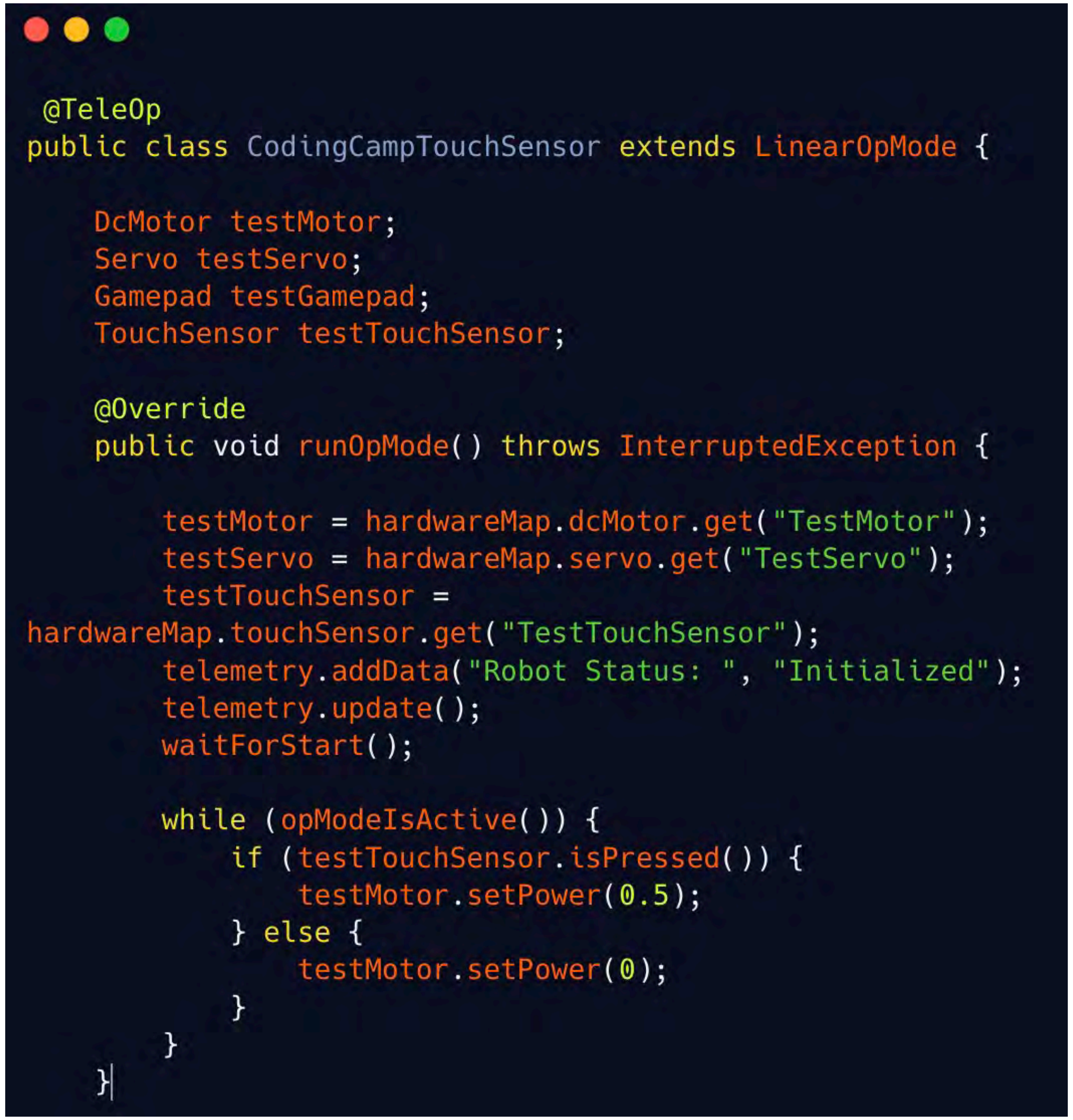
- Add gamepad object
- Create while loop
- Set motor power to gamepad joystick
- If a or b pressed on gamepad, move servo to a corresponding position



```
 @TeleOp
public class CodingCampGamepad extends LinearOpMode {

    DcMotor testMotor;
    Servo testServo;
    Gamepad testGamepad;

    @Override
    public void runOpMode() throws InterruptedException {

        testMotor = hardwareMap.dcMotor.get("TestMotor");
        testServo = hardwareMap.servo.get("TestServo");
        telemetry.addData("Robot Status: ", "Initialized");
        telemetry.update();
        waitForStart();

        while (opModeIsActive()) {
            testMotor.setPower(testGamepad.right_stick_y);

            if (testGamepad.a) {
                testServo.setPosition(1);
            } else if (testGamepad.b) {
                testServo.setPosition(0);
            }
        }
    }
```

Coding a touch sensor to move a motor

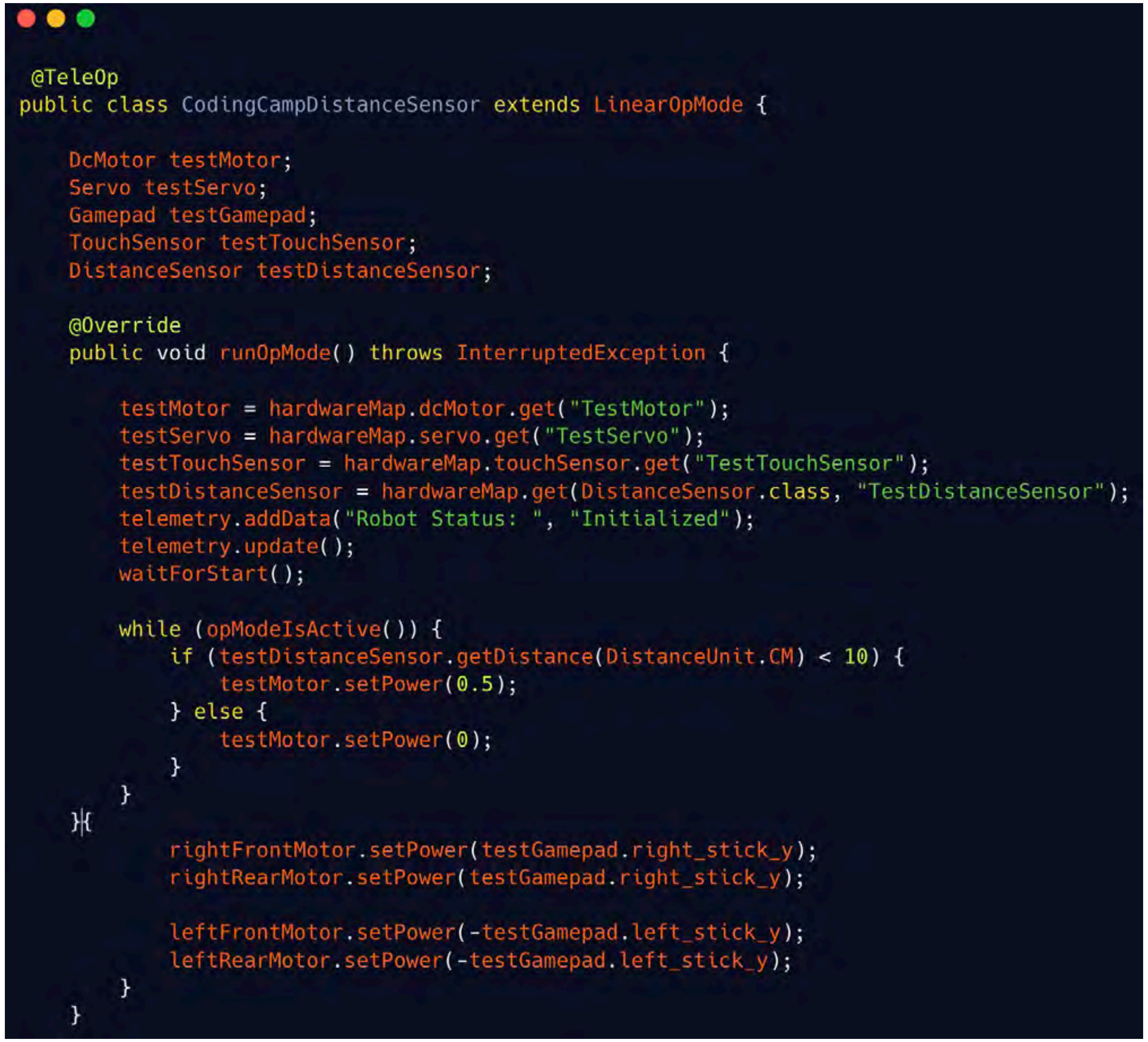
- Create a touch sensor object
- Add it to the hardware map
- Make a condition to move the motor if it is pressed



```
 @TeleOp
public class CodingCampTouchSensor extends LinearOpMode {

    DcMotor testMotor;
    Servo testServo;
    Gamepad testGamepad;
    TouchSensor testTouchSensor;

    @Override
    public void runOpMode() throws InterruptedException {

        testMotor = hardwareMap.dcMotor.get("TestMotor");
        testServo = hardwareMap.servo.get("TestServo");
        testTouchSensor =
hardwareMap.touchSensor.get("TestTouchSensor");
        telemetry.addData("Robot Status: ", "Initialized");
        telemetry.update();
        waitForStart();

        while (opModeIsActive()) {
            if (testTouchSensor.isPressed()) {
                testMotor.setPower(0.5);
            } else {
                testMotor.setPower(0);
            }
        }
    }
```

Coding a distance sensor to move a motor

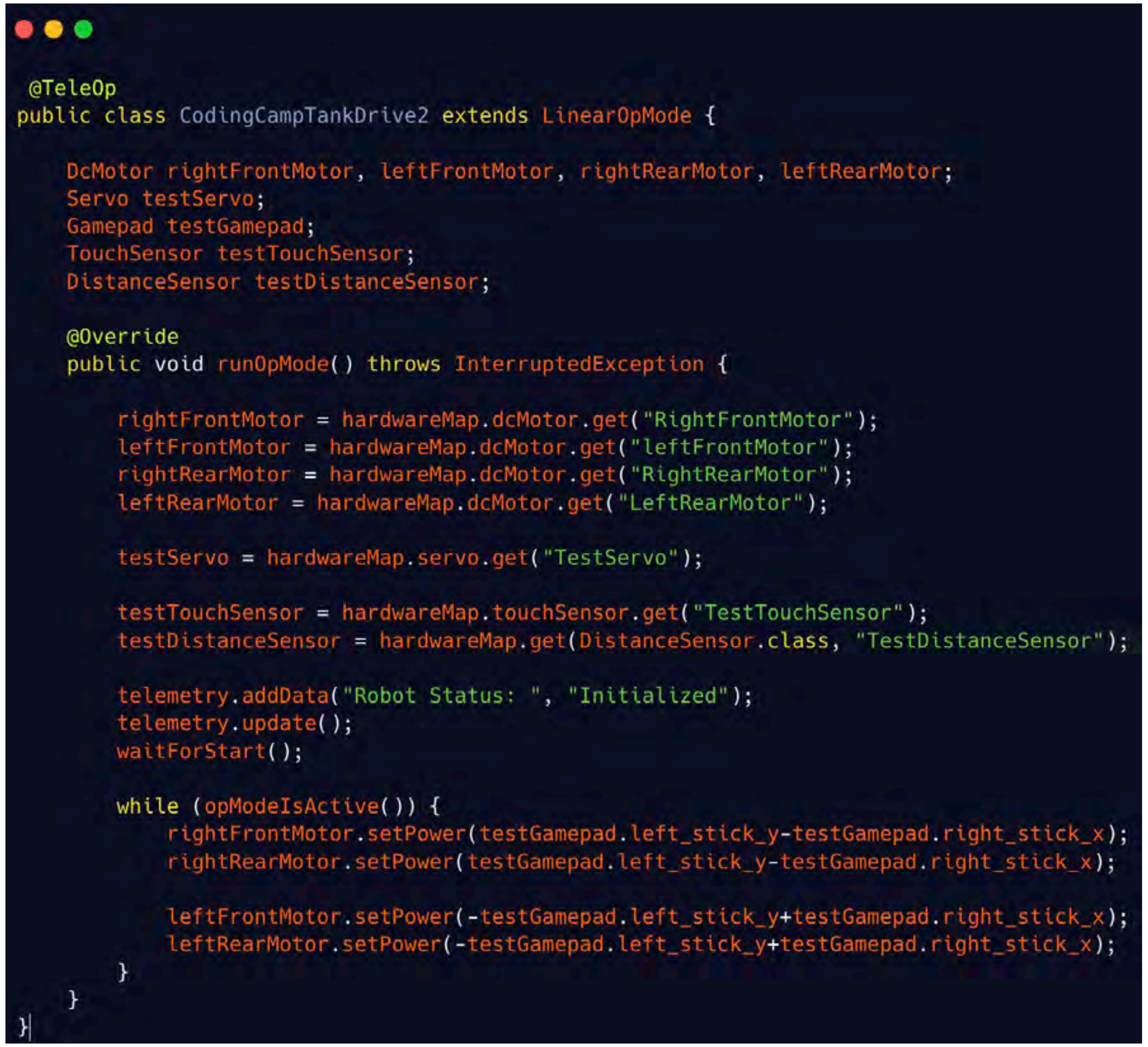
- Create a distance sensor object
- Add it to the hardware map
- Make a condition to move the motor if the distance is less than 10cm



```java
 @TeleOp
public class CodingCampDistanceSensor extends LinearOpMode {

    DcMotor testMotor;
    Servo testServo;
    Gamepad testGamepad;
    TouchSensor testTouchSensor;
    DistanceSensor testDistanceSensor;

    @Override
    public void runOpMode() throws InterruptedException {

        testMotor = hardwareMap.dcMotor.get("TestMotor");
        testServo = hardwareMap.servo.get("TestServo");
        testTouchSensor = hardwareMap.touchSensor.get("TestTouchSensor");
        testDistanceSensor = hardwareMap.get(DistanceSensor.class, "TestDistanceSensor");
        telemetry.addData("Robot Status: ", "Initialized");
        telemetry.update();
        waitForStart();

        while (opModeIsActive()) {
            if (testDistanceSensor.getDistance(DistanceUnit.CM) < 10) {
                testMotor.setPower(0.5);
            } else {
                testMotor.setPower(0);
            }
        }
    }{

            rightFrontMotor.setPower(testGamepad.right_stick_y);
            rightRearMotor.setPower(testGamepad.right_stick_y);

            leftFrontMotor.setPower(-testGamepad.left_stick_y);
            leftRearMotor.setPower(-testGamepad.left_stick_y);
        }
    }
```

Coding a Tank Drivebase pt2

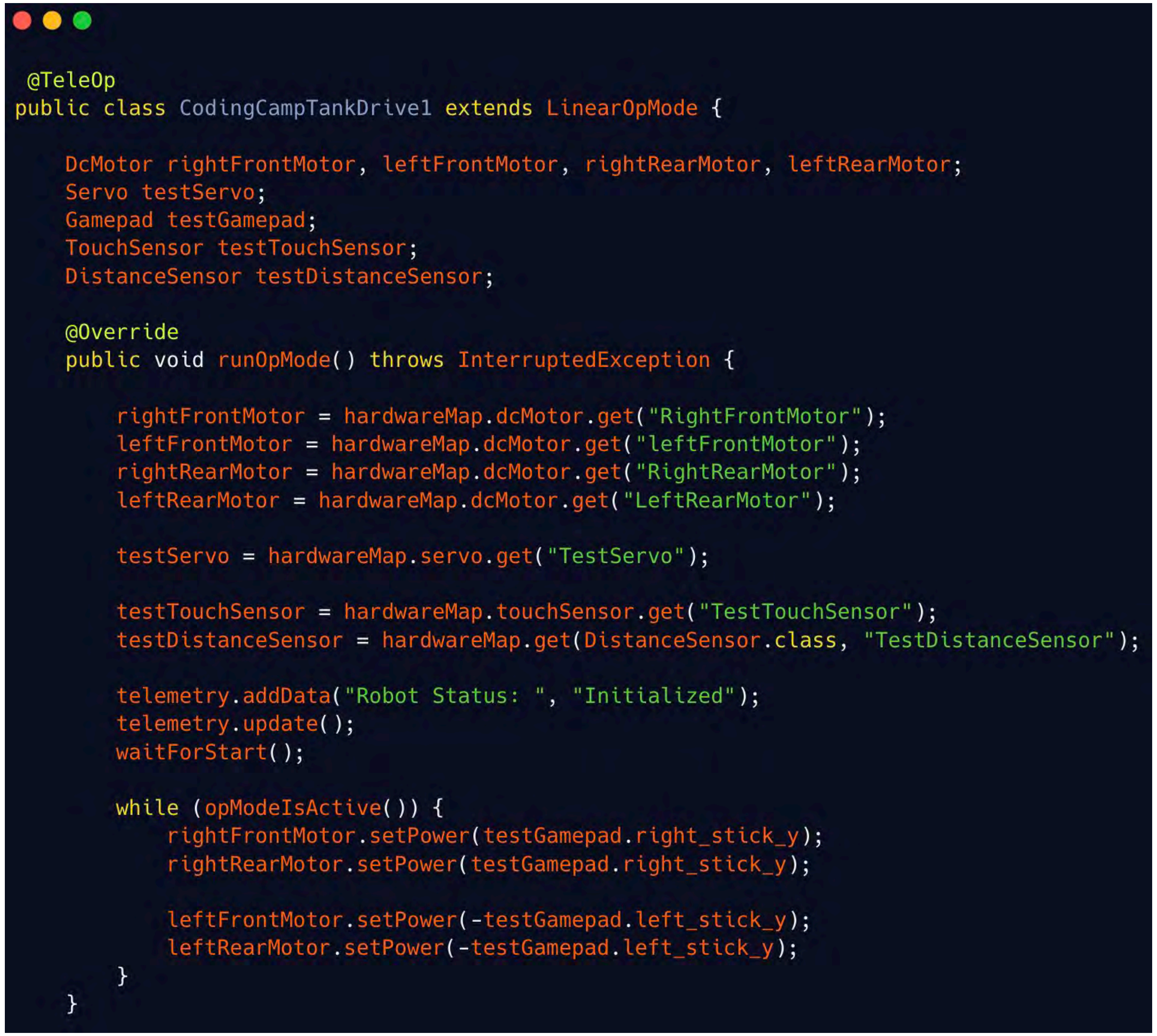
- Change the right-side motors to be set to the leftjoystick_y, and subtract the value of the rightjoystick_x
- Change the left-side motors to add the value of the rightjoystick_x



```java
@TeleOp
public class CodingCampTankDrive2 extends LinearOpMode {

    DcMotor rightFrontMotor, leftFrontMotor, rightRearMotor, leftRearMotor;
    Servo testServo;
    Gamepad testGamepad;
    TouchSensor testTouchSensor;
    DistanceSensor testDistanceSensor;

    @Override
    public void runOpMode() throws InterruptedException {

        rightFrontMotor = hardwareMap.dcMotor.get("RightFrontMotor");
        leftFrontMotor = hardwareMap.dcMotor.get("leftFrontMotor");
        rightRearMotor = hardwareMap.dcMotor.get("RightRearMotor");
        leftRearMotor = hardwareMap.dcMotor.get("LeftRearMotor");

        testServo = hardwareMap.servo.get("TestServo");

        testTouchSensor = hardwareMap.touchSensor.get("TestTouchSensor");
        testDistanceSensor = hardwareMap.get(DistanceSensor.class, "TestDistanceSensor");

        telemetry.addData("Robot Status: ", "Initialized");
        telemetry.update();
        waitForStart();

        while (opModeIsActive()) {
            rightFrontMotor.setPower(testGamepad.left_stick_y-testGamepad.right_stick_x);
            rightRearMotor.setPower(testGamepad.left_stick_y-testGamepad.right_stick_x);

            leftFrontMotor.setPower(-testGamepad.left_stick_y+testGamepad.right_stick_x);
            leftRearMotor.setPower(-testGamepad.left_stick_y+testGamepad.right_stick_x);
        }
    }
}
```

Coding a Tank Drivebase pt1

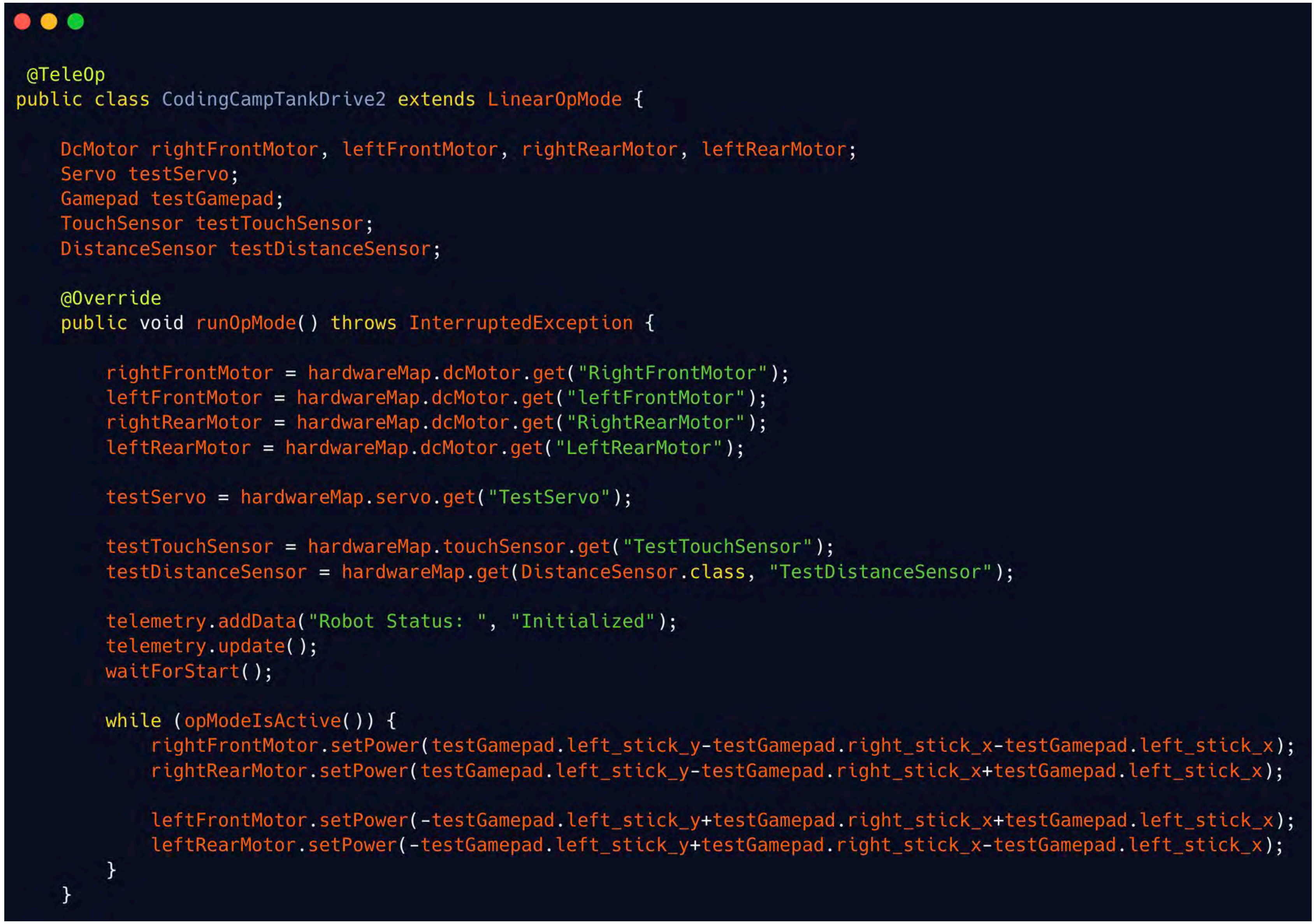
- Create an object for each motor
- Code the left joystick to control left two motors, and right joystick to control right 2 motors



```
 @TeleOp
public class CodingCampTankDrive1 extends LinearOpMode {

    DcMotor rightFrontMotor, leftFrontMotor, rightRearMotor, leftRearMotor;
    Servo testServo;
    Gamepad testGamepad;
    TouchSensor testTouchSensor;
    DistanceSensor testDistanceSensor;

    @Override
    public void runOpMode() throws InterruptedException {

        rightFrontMotor = hardwareMap.dcMotor.get("RightFrontMotor");
        leftFrontMotor = hardwareMap.dcMotor.get("leftFrontMotor");
        rightRearMotor = hardwareMap.dcMotor.get("RightRearMotor");
        leftRearMotor = hardwareMap.dcMotor.get("LeftRearMotor");

        testServo = hardwareMap.servo.get("TestServo");

        testTouchSensor = hardwareMap.touchSensor.get("TestTouchSensor");
        testDistanceSensor = hardwareMap.get(DistanceSensor.class, "TestDistanceSensor");

        telemetry.addData("Robot Status: ", "Initialized");
        telemetry.update();
        waitForStart();

        while (opModeIsActive()) {
            rightFrontMotor.setPower(testGamepad.right_stick_y);
            rightRearMotor.setPower(testGamepad.right_stick_y);

            leftFrontMotor.setPower(-testGamepad.left_stick_y);
            leftRearMotor.setPower(-testGamepad.left_stick_y);
        }
    }
```

Coding a Mecanum Drivebase

- Add the leftjoystick_x to the leftFront and rightRear motor
- Subtract the leftjoystick_x from the rightfront and leftRear motor



```
 @TeleOp
public class CodingCampTankDrive2 extends LinearOpMode {

    DcMotor rightFrontMotor, leftFrontMotor, rightRearMotor, leftRearMotor;
    Servo testServo;
    Gamepad testGamepad;
    TouchSensor testTouchSensor;
    DistanceSensor testDistanceSensor;

    @Override
    public void runOpMode() throws InterruptedException {

        rightFrontMotor = hardwareMap.dcMotor.get("RightFrontMotor");
        leftFrontMotor = hardwareMap.dcMotor.get("leftFrontMotor");
        rightRearMotor = hardwareMap.dcMotor.get("RightRearMotor");
        leftRearMotor = hardwareMap.dcMotor.get("LeftRearMotor");

        testServo = hardwareMap.servo.get("TestServo");

        testTouchSensor = hardwareMap.touchSensor.get("TestTouchSensor");
        testDistanceSensor = hardwareMap.get(DistanceSensor.class, "TestDistanceSensor");

        telemetry.addData("Robot Status: ", "Initialized");
        telemetry.update();
        waitForStart();

        while (opModeIsActive()) {
            rightFrontMotor.setPower(testGamepad.left_stick_y-testGamepad.right_stick_x-testGamepad.left_stick_x);
            rightRearMotor.setPower(testGamepad.left_stick_y-testGamepad.right_stick_x+testGamepad.left_stick_x);

            leftFrontMotor.setPower(-testGamepad.left_stick_y+testGamepad.right_stick_x+testGamepad.left_stick_x);
            leftRearMotor.setPower(-testGamepad.left_stick_y+testGamepad.right_stick_x-testGamepad.left_stick_x);
        }
    }
```

What is Machine Learning?

- Machine learning is a branch of Artificial Intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way humans learn, gradually improving its accuracy
- In traditional programming, complex rules are added to the computer program and those are used to analyze input data and output an answer. Example: if the input data is an image of a flower, and if the programming rules can recognize the flower, then the answer “flower” is output
- To enable a traditional program to recognize the differences between multiple different kinds of flowers, would require significantly more complex programming
- Instead, machine learning focuses on providing examples to a machine learning algorithm or “model” – providing data and answers – and allowing the model to build its own rules to determine the relationships between the examples provided to it.
- By training a model over successive steps, the model attempts to improve its accuracy in correctly identifying previously unseen variations of the data

TensorFlow

- In FIRST Tech Challenge, the machine learning platform used is TensorFlow
- TensorFlow is an open-source platform for machine learning with a good set of tools, libraries, and resources to enable developers to create tools such as the FTC Machine Learning tool
- TensorFlow has been utilized in FTC for a number of years, allowing teams to recognize individual game pieces and clusters of game pieces via pre-built models developed by FTC engineers
- Now FTC is empowering teams to build their own custom models!

Links / References

You have listened to so much at the camp, what if you can't remember something?

- FTC Control System:
  https://ftc-docs.firstinspires.org/en/latest/programming_resources/shared/control_system_intro/The-FTC-Control-System.html

  https://ftc-docs.firstinspires.org/en/latest/programming_resources/shared/program_and_manage_network/Connecting-a-Laptop-to-the-Program-%26-Manage-Network.html
- Wiring Guide:
  https://ftc-docs.firstinspires.org/en/latest/_images/B1.svg
- Java for FTC (PDF):
  https://github.com/alan412/LearnJavaForFTC/blob/master/LearnJavaForFTC.pdf
- Install Android Studio:
  https://ftc-docs.firstinspires.org/en/latest/programming_resources/tutorial_specific/android_studio/installing_android_studio/Installing-Android-Studio.html
- FIRST's document on using Android Studio for FTC:
  https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/android-studio-guide.pdf
- Get the FTC SDK app:
  https://github.com/FIRST-Tech-Challenge/FtcRobotController
- Learn to use Github to manage the project source code:
  https://ftc-docs.firstinspires.org/en/latest/programming_resources/tutorial_specific/android_studio/fork_and_clone_github_repository/Fork-and-Clone-From-GitHub.html

Links / References (continued)

You have listened to so much at the camp, what if you can't remember something?

- Coding for FTC:
  https://ftc-docs.firstinspires.org/en/latest/programming_resources/tutorial_specific/android_studio/creating_op_modes/Creating-and-Running-an-Op-Mode-%28Android-Studio%29.html

  https://ftc-docs.firstinspires.org/en/latest/programming_resources/tutorial_specific/android_studio/controlling_a_servo/Controlling-a-Servo-%28Android-Studio%29.html

  https://ftc-https//ftc-docs.firstinspires.org/en/latest/programming_resources/tutorial_specific/android_studio/using_sensors/Using-Sensors-%28Android-Studio%29.html
- Machine Learning and Tensor Flow in FTC:
  https://ftc-docs.firstinspires.org/en/latest/ftc_ml/index.html

  https://ftc-ml.firstinspires.org/
- Tensor Flow in 2022-23 (PowerPlay season):
  https://ftc-docs.firstinspires.org/en/latest/programming_resources/vision/tensorflow_pp_2022/tensorflow_pp_2022.html
- April Tags:
  https://ftc-docs.firstinspires.org/en/latest/apriltag/vision_portal/apriltag_intro/apriltag-intro.html
- Functional Java and Virtual Robot Simulator (provided by a Coach from Illinois):
  https://docs.google.com/presentation/d/17kHtmei_IZTAIO69mz2FqwuGX23APpWheFP3XcndoxM/edit?fbclid=IwAR0W6P39yB1G1UKCCJorrjjaoMPRJo09Xwfa8E1HFiHlsx43fh92UqtXWms#slide=id.g23160c7411d_0_533