Virtual Robot – Detailed Installation Instructions

You can use Virtual Robot with either the IntelliJ IDEA Integrated Development Environment (IDE) or Android Studio. The steps are essentially the same, as Android Studio is built on top of IntelliJ.

This guide will demonstrate the use of IntelliJ. If you already have Android Studio, just follow the steps here, except for downloading and installing IntelliJ. If you would prefer to use Android Studio but need to install it, it is easy to obtain and install (just google it).

A word about JDK and JavaFX

In addition to your IDE (IntelliJ or Android Studio) and the Virtual Robot project, you wil need a Java Development Kit (JDK) and JavaFX. The JDK includes a Java Compiler and various Java libraries and programming tools. JavaFX is a platform for creating desktop applications using Java.

Some JDKs have JavaFX built-in. Others require a separate download and installation of JavaFX. The separate download of JavaFX requires some additional steps in your IDE (IntelliJ or Android Studio) that can be somewhat burdensome and error-prone. For that reason, this guide will only discuss JDKs that have JavaFX built-in. Those are: Amazon Correto 8, Bell Liberica 8, and Bell Liberica 17. Amazon Correto 8 has JavaFX built-in for Windows, but not for Mac OS. Bell Liberica 8 and Bell Liberica 17 have JavaFX built-in for Windows, and *I believe* for Mac OS as well. I have tried Bell Liberica 21 on Windows, and could not get it to work. Bell Liberica 8 or 17 is therefore recommended as the JDK for Windows or Mac.

If you have a Mac, and find that Bell Liberica 8 or Bell Liberica 17 does not work for you, please let us know (Ftc8397@gmail.com). We will help you get set up with a different solution, and edit this guide.

Virtual Robot – Detailed Installation Instructions—IntelliJ IDEA

I. If you don't already have a Java Development Kit (JDK) installed on your system, you need it. Current Recommendation: Liberica 8 or Liberica 17.

To get Liberica 17 as your JDK (new Java version; supports PC and Mac):

Navigate to: [Download Java | Java 8, Java 11, Java 17, Java 21 - OpenJDK Builds for Linux, Windows & macOS (bell-sw.com)]

Select "JDK 17 LTS" at top right of the page.

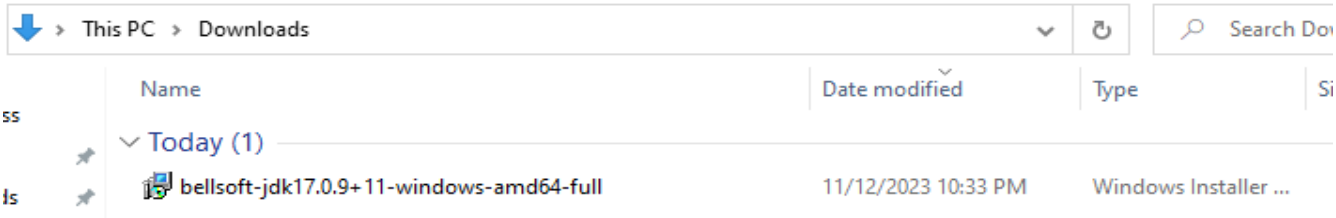
Do NOT click on the big black buttons for Windows or Mac. Those will give you the "standard" version; you need the "full" version, which includes JavaFx.

Scroll down to find your Operating System (e.g., Windows 64 bit, or macOS), and under the operating system name, select your processor type (almost certainly x86). Use the "Package" drop-down to select the "Full JDK" package.

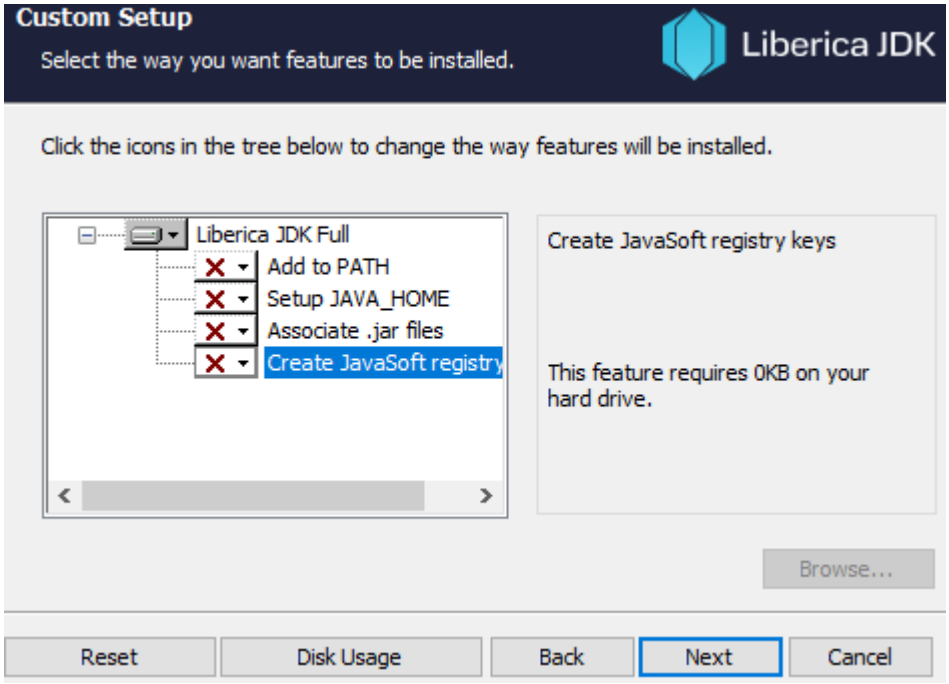
Then download the installer file (MSI for Windows, DMG for macOS).

Once you have downloaded the installer file, double-click it to install the Liberica JDK.

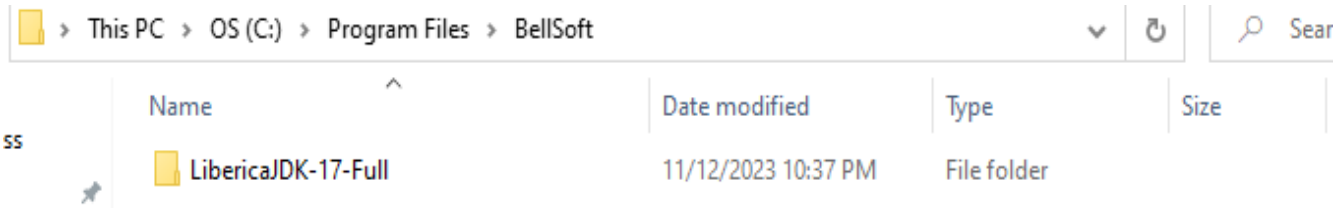
The downloaded Liberica 19 installer file on my system:



If just using Liberica 17 to run virtual_robot on IntelliJ or Android Studio, you can reject the "Add to PATH", "Setup JAVA_HOME, "Associate .jar file", and "Create JavaSoft registry" features.



Running the Liberica 17 installer on my system installs the JDK to:

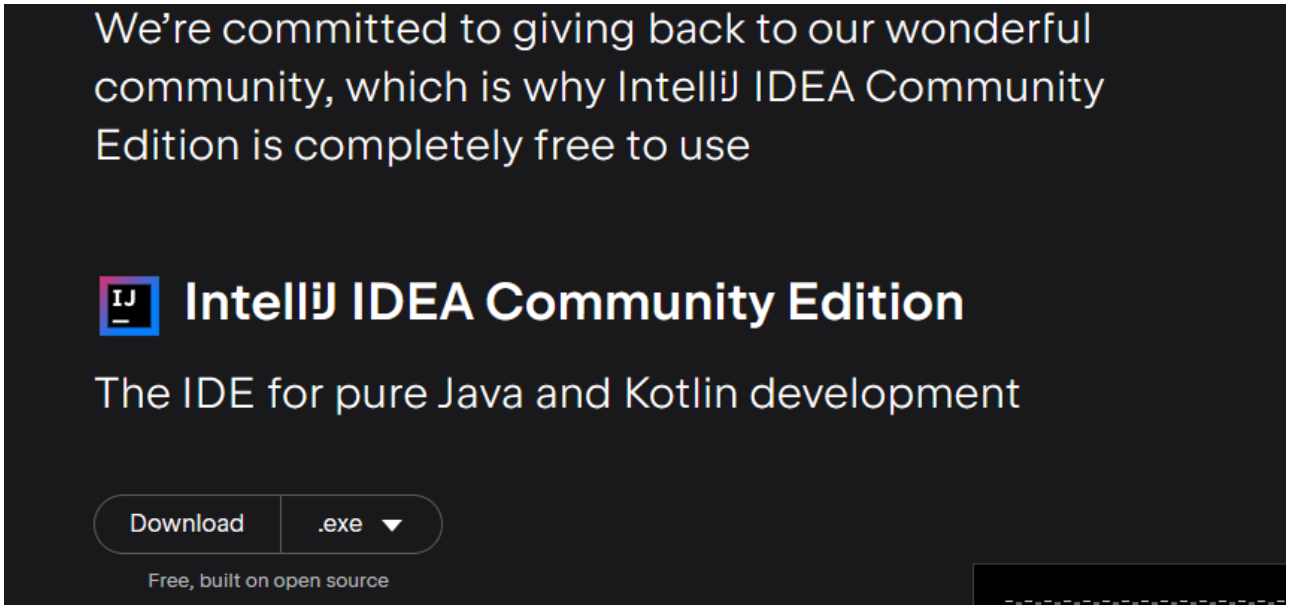


II. If you don't already have IntelliJ IDEA on your system, you'll need to install it.

(SKIP if using Android Studio)

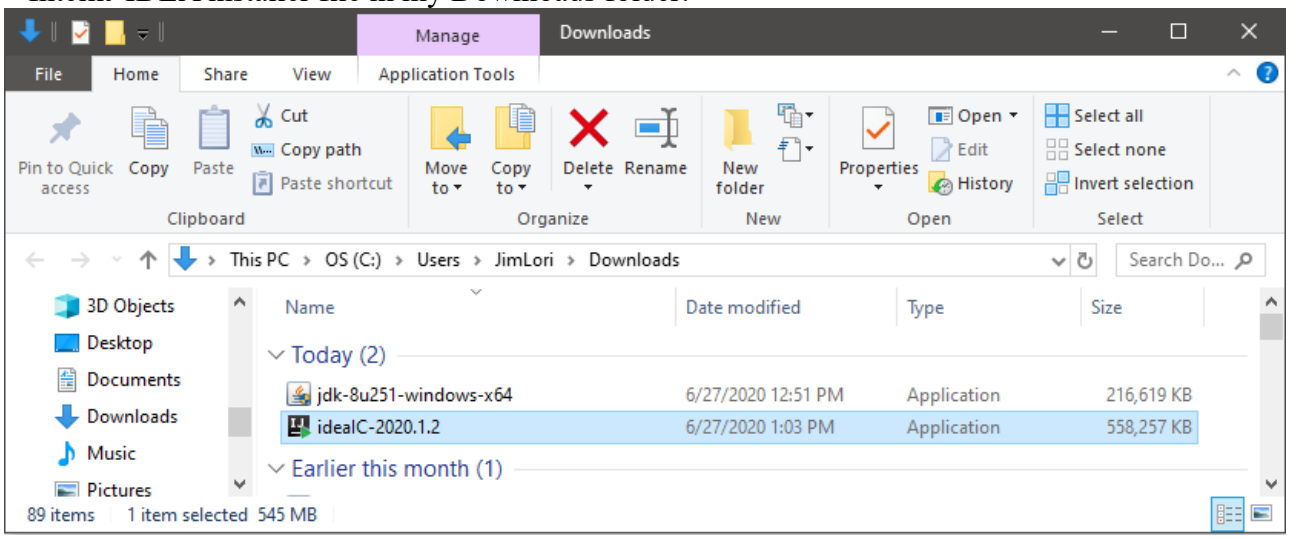
https://www.jetbrains.com/idea/download/#section=windows

Select "Windows" or "MacOS", then scroll down to the Community Edition section.



Click to download the .exe (Windows) or .dmg (MacOS) for the free Community edition of IntelliJ IDEA. For example, this might be named idealC-2020.1.2.exe.

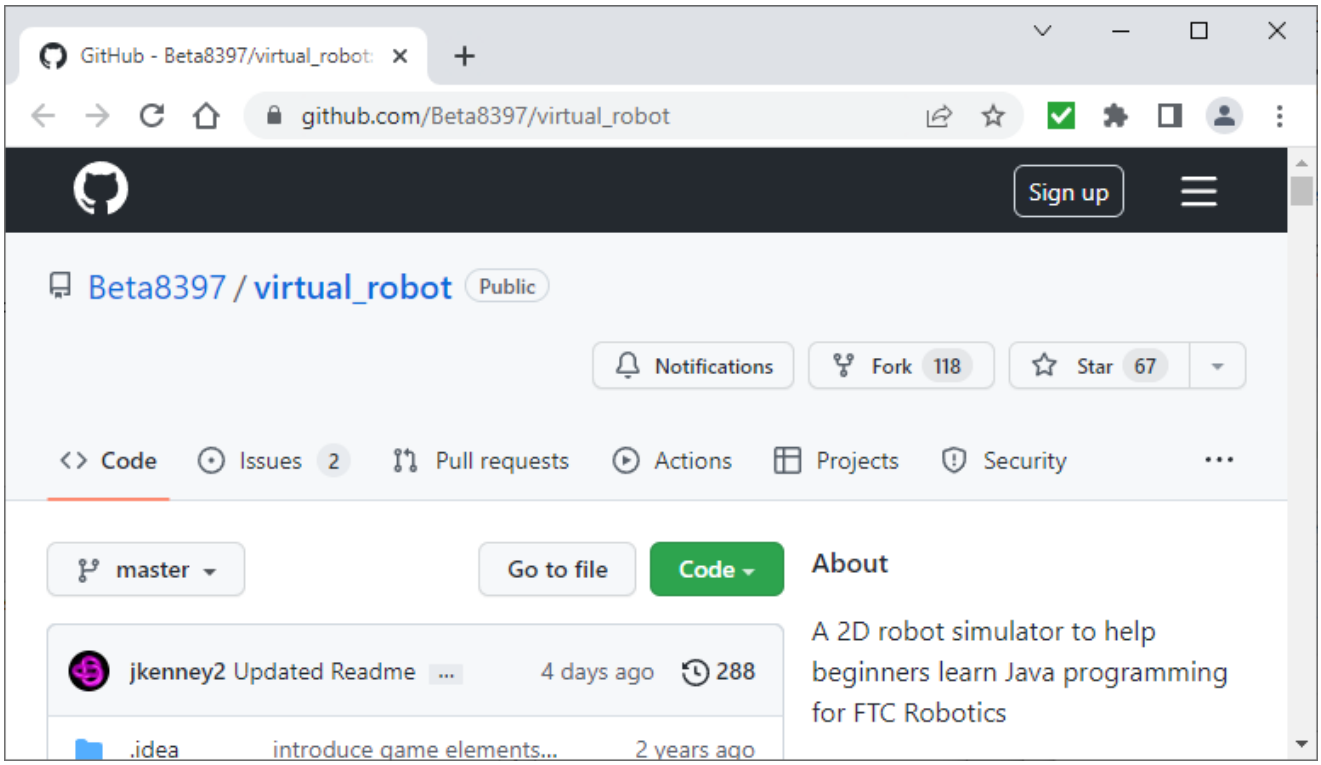
IntelliJ IDEA installer file in my Downloads folder:



Run this .exe file to install IntelliJ IDEA.

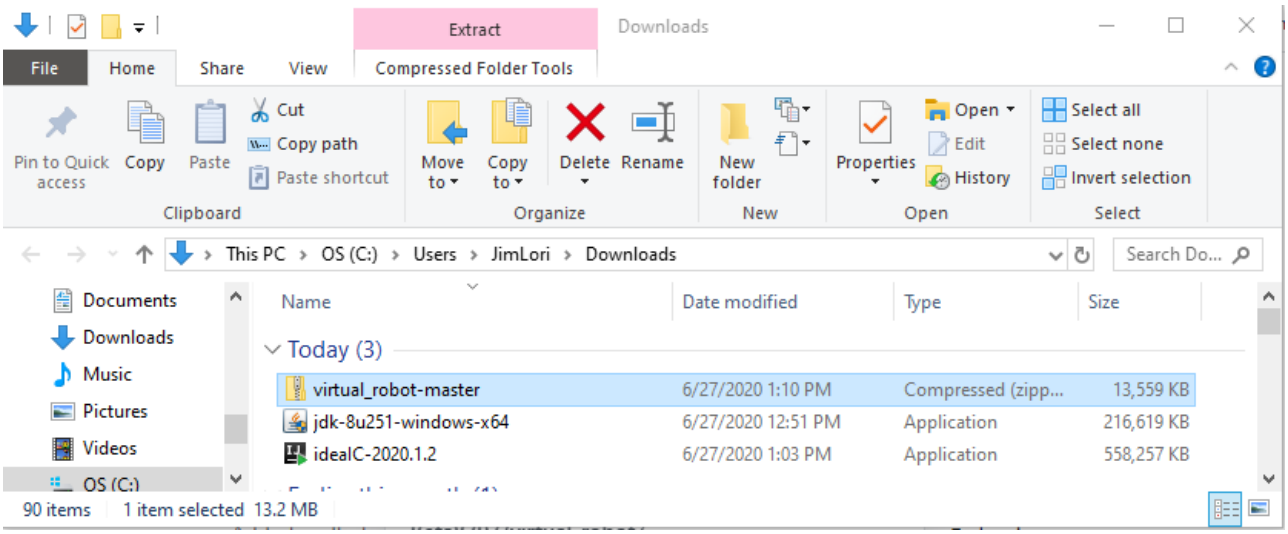
III. Download the Virtual Robot project.

Open a browser and navigate to: https://github.com/Beta8397/virtual_robot



Click the green button ("Code") on the right and select the "Download Zip" option. You'll find the .zip folder in your downloads directory: virtual_robot-master.zip.

virtual_robot-master.zip file in my Downloads folder:



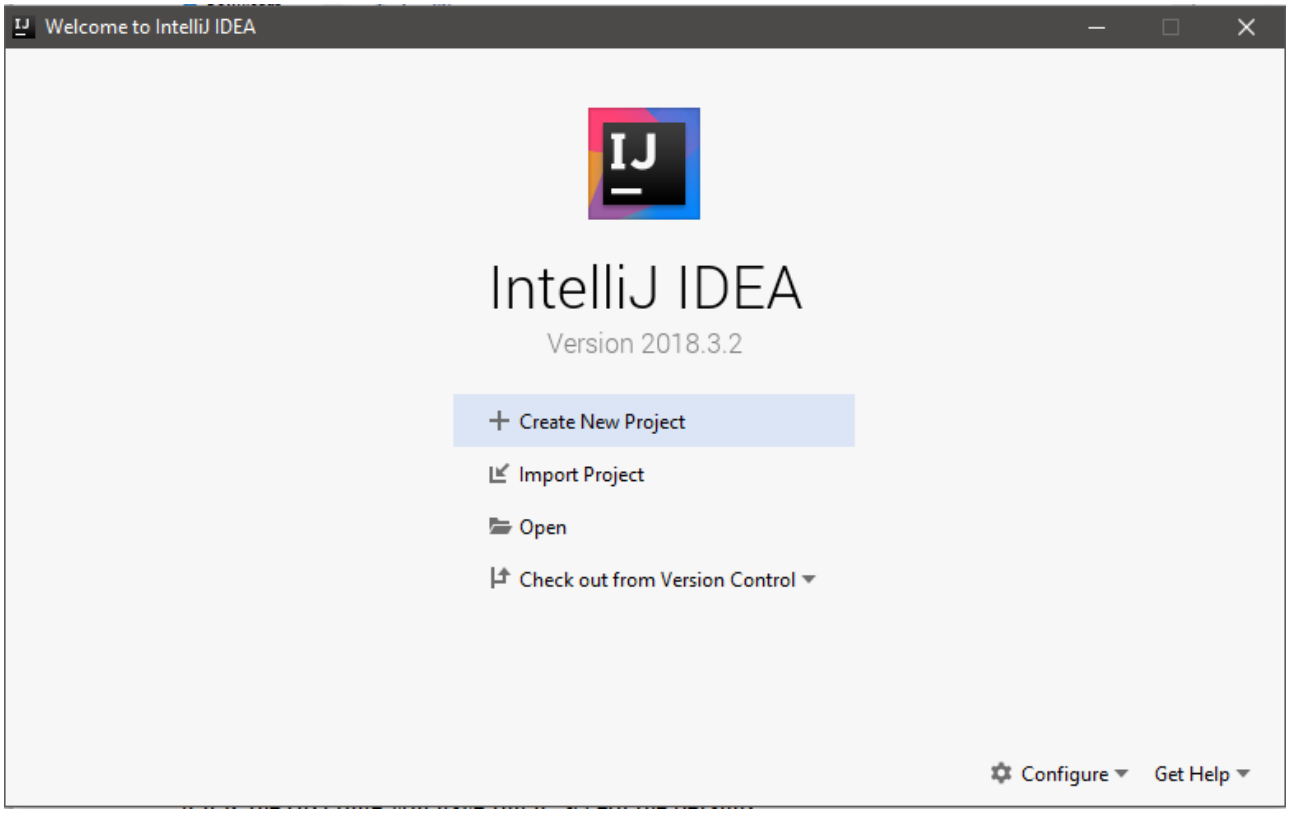
Right-click the zip folder and select "Extract all...". Extract to the folder of your choice. You'll wind up with an unzipped folder, for example: C:\Users\Me\Downloads\virtual_robot-master.

Within this folder, you should find a single directory: virtual_robot-master. That is the project folder. So for example, the path to the project folder might be:

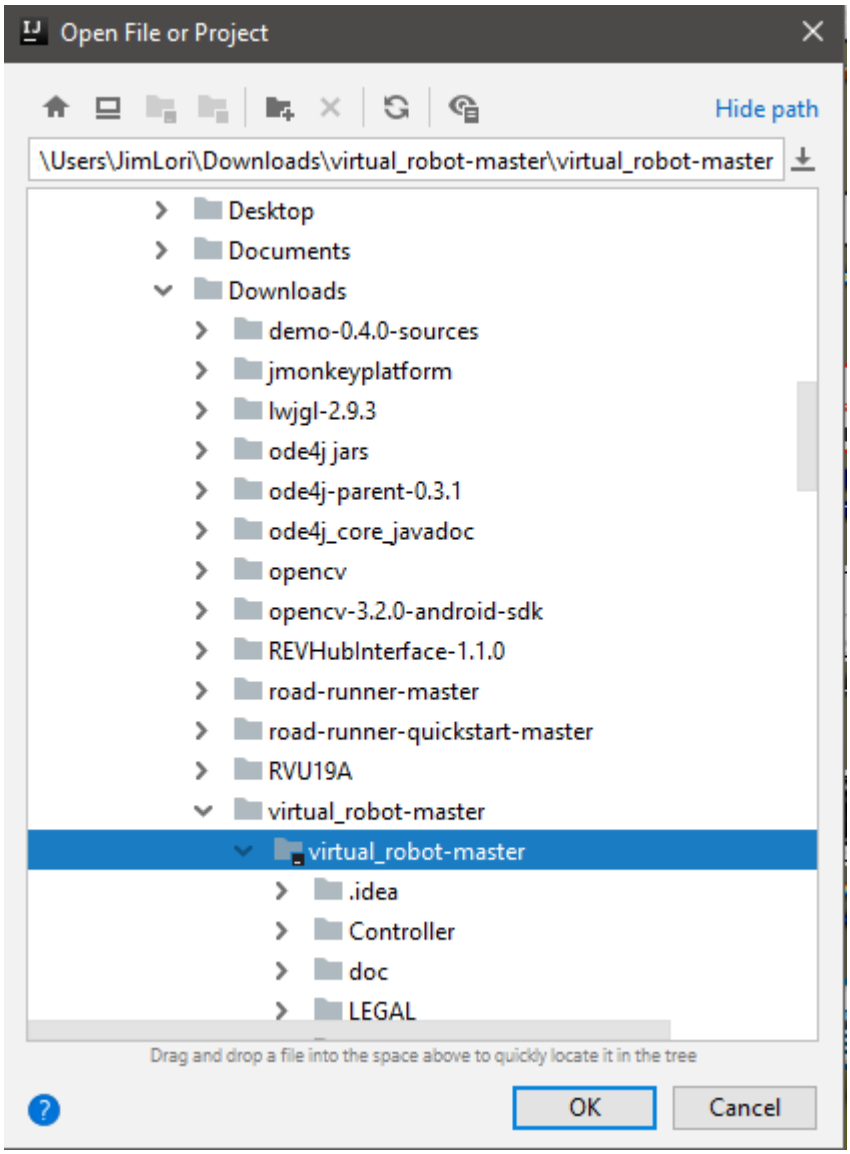
C:\Users\Me\Downloads\virtual_robot-master\virtual_robot-master

IV. Run the IntelliJ IDEA application (or Android Studio, if using that).

If it is the first time you have run it, accept the defaults.



Click "Open", and navigate to your project folder, virtual_robot-master. Note that if you have unzipped into a folder named virtual_robot-master, then you will have another folder named virtual_robot-master nested within it. The project folder is the inner one.



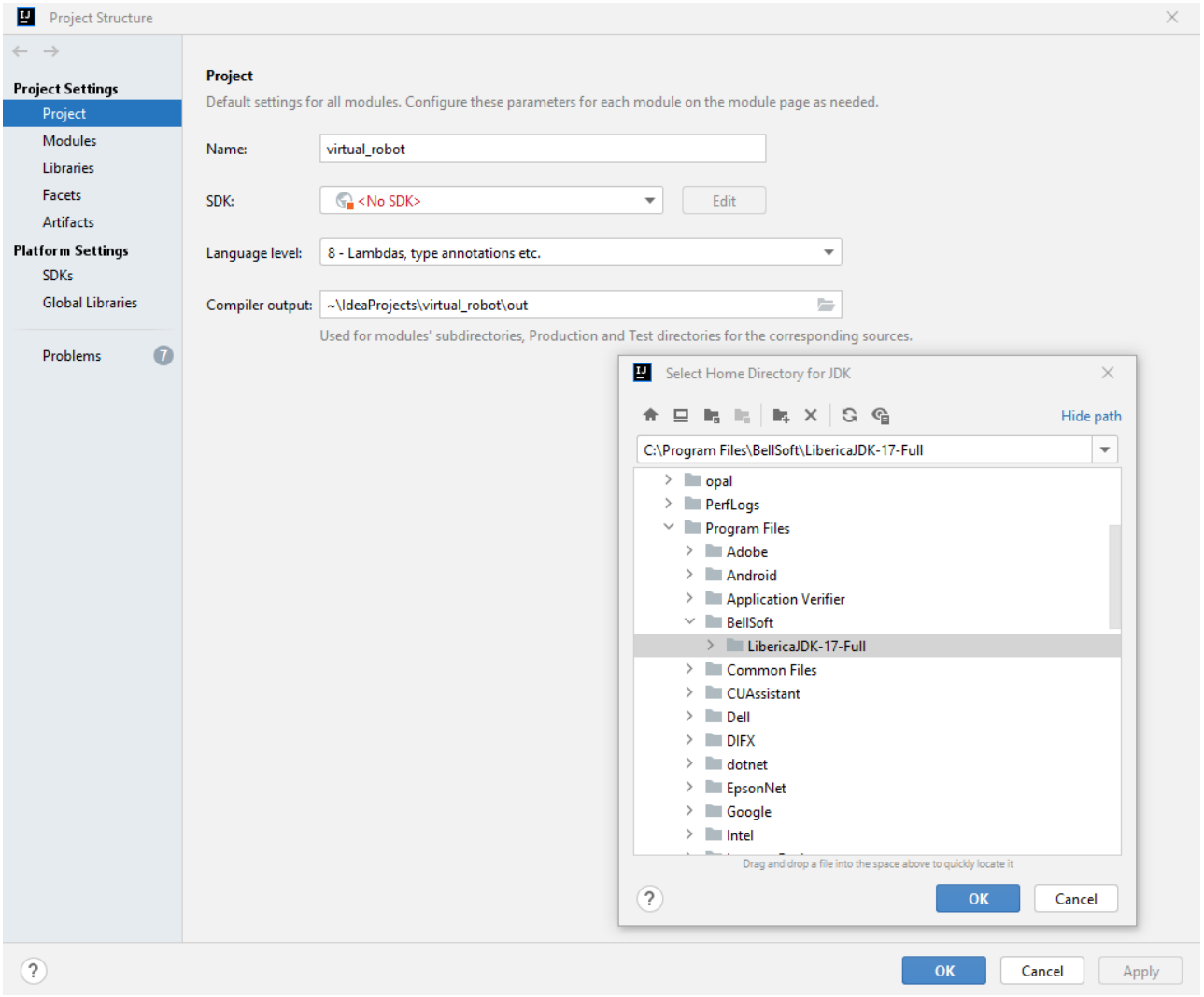
Click "OK", and the project will open.

V. Make sure the correct JDK is selected.

Go to: File --> Project Structure

and select "Project".

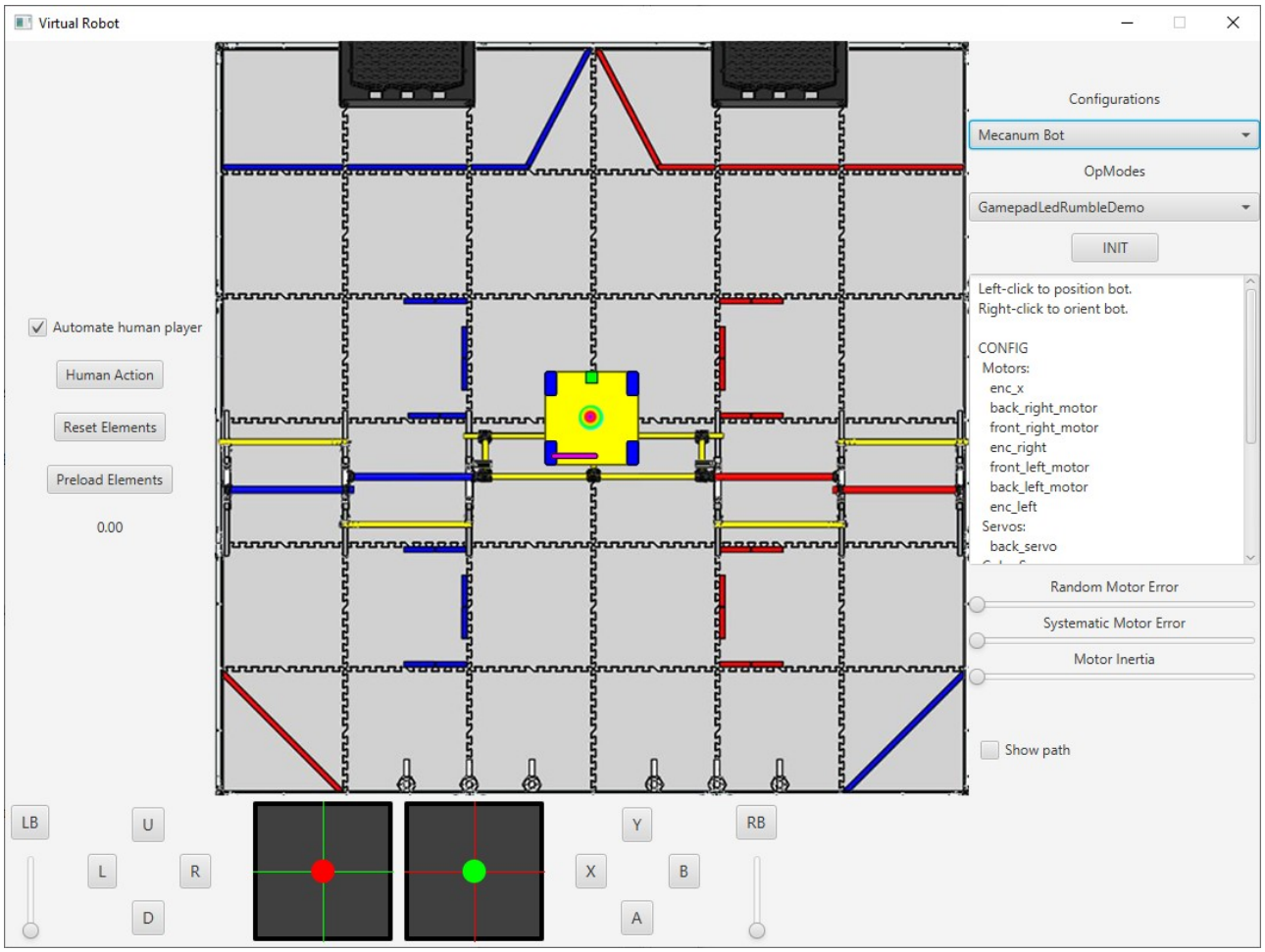
Under "Project SDK", expand the dropdown and look for the JDK (e.g., Bell Liberica 17) that you installed. If present, select it. If not, click "Add SDK" at the bottom of the expanded SDK dropdown, then JDK. Navigate to the JDK that you installed, select it, then click "OK" on the "Select Home Directory for JDK" dialog.



Under "Language Level", select "8" if you are using Liberica 8 JDK, or "17" if you are using Liberica 17 JDK.

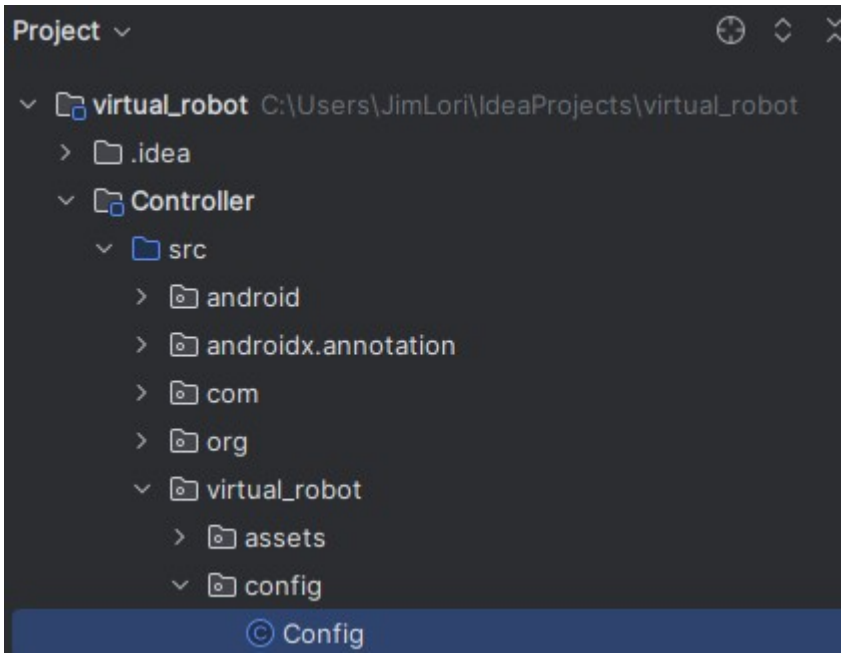
Then, in the "Project Structure" dialog, click "Apply", then "OK".

Now you should be able run the app by clicking the green arrowhead.



VI. Modify the Config.java file to suit your needs.

By default, virtual_robot provides a "virtual gamepad" at the bottom of the window. If desired, you can plug in one or two real gamepads instead. To do this, in the Controller module, navigate to the "src->virtual_robot->config" folder and open Config.java.



In the Config.java file, change the value of the USE_VIRTUAL_GAMEPAD variable from true to false. Then plug in your real gamepad(s) and go. Don't forget to press "Start-A" on the gamepad once you have started the application.

virtual_robot uses a bitmap (.BMP) image of the FTC playing field as the background. By default, the resolution of this image is 648x648 pixels. On some laptops, when using the virtual gamepad, this results in an application window that is too big to fit completely on the screen. This is easy to fix, as follows:

Open the Config.java file (see above).
Change the value of the FIELD_WIDTH variable from 648 to 576.
Change the name of the backgrount image from centerstage648.bmp to centerstage576.bmp.

```java
/**
 *  Width of the field, in pixels
 */
public static final double FIELD_WIDTH = 576;  2 usages

/**
 * Whether to use "Virtual Gamepad" (true -> Virtual gamepad, false -> Real gamepad)
 */
public static final boolean USE_VIRTUAL_GAMEPAD = true;  1 usage

/**
 * The image object for the field.
 */
public static final Image BACKGROUND = new Image( url: "/virtual_robot/assets/centerstage576.bmp");
```

VII. Now you are ready to write and test your own op modes.