Rolls-Royce

GETTING STARTED GUIDE

WITH

RACE YOUR CODE VIRTUAL

FormulaPi.com

The below is insight from the people who create and build the robots you'll be racing on how the challenge works and the code you will receive.

THE CHALLENGE

Thanks for taking part in the Rolls–Royce Race Your Code virtual racing. The racing is being hosted by Formula Pi in Cambridge, in the UK.

In this competition, you will edit and write code which will be used to control virtual robots. These robotswill compete against each other in 4 virtual races and one real world race.

To do this, you take pre–programmed robot code which can navigate the racetrack on it's own, improve the code and submit it to the Formula Pi server to race against other competitors on virtual tracks. This could be tricky as your only sensor input is from a camera mounted at the front of the robot, but improvements can be as simple as changing the lane on the track, or as complicated as major changes, such as a complete code rewrite.

The race dates are as follows:

9th April 2019   Virtual Test round 1

24th April 2019   Virtual Race round 1

30th April 2019   Virtual Race round 2

7th May 2019   Virtual Race round 3

14th May 2019   Virtual Race Round 4

21st May 2019   Real World Race

Watch the Live stream

All races are live streamed on YouTube so you can watch the virtual racing live, or watch it back at any time after the event.

There are 4 virtual tracks and one real world track. These are in order:

Silverstone, UK
Cota, USA
Hockenheim, Germany
Marina Bay, Singapore

The real track is the Formula Pi circuit in Cambridge, UK.

Code can be improved in various ways. You could for example vary the lane choice the robot tries to navigate around, but be careful – the robots move in unexpected ways!

All races will be broadcast at 12:30pm UK time

STEPS FOR COMPETITORS

The laptop you are using has been specially configured to simplify the process of editing, testing, uploading code, and viewing the live stream.

You will need to connect the laptop to a Wi-Fi connection in order to upload your code and to view the stream. We recommend using the guest Wi-Fi or an external Internet connection.

Use the links on the desktop

The laptop desktop has several shortcuts which make it easy to edit your code, test it works in the simulator, and upload it to the server ready for each race.

Desktop icons

From the links you will be able to view and edit the code, get updated instructions, run the simulator, upload your code for the competition and watch the live stream.

The link that opens this instruction document will attempt to update every time you click on the link to view them, so you should always have the latest information.

EDITING CODE

Restore code to the original version

Double click on the "Restore code" icon on the desktop to restore code to the original version. This deletes all changes you have made to the code without backing up! – so only do this if you really want to go back to the original code. We recommend doing this first before doing any code editing, to make sure to clear code any previous teams have written.
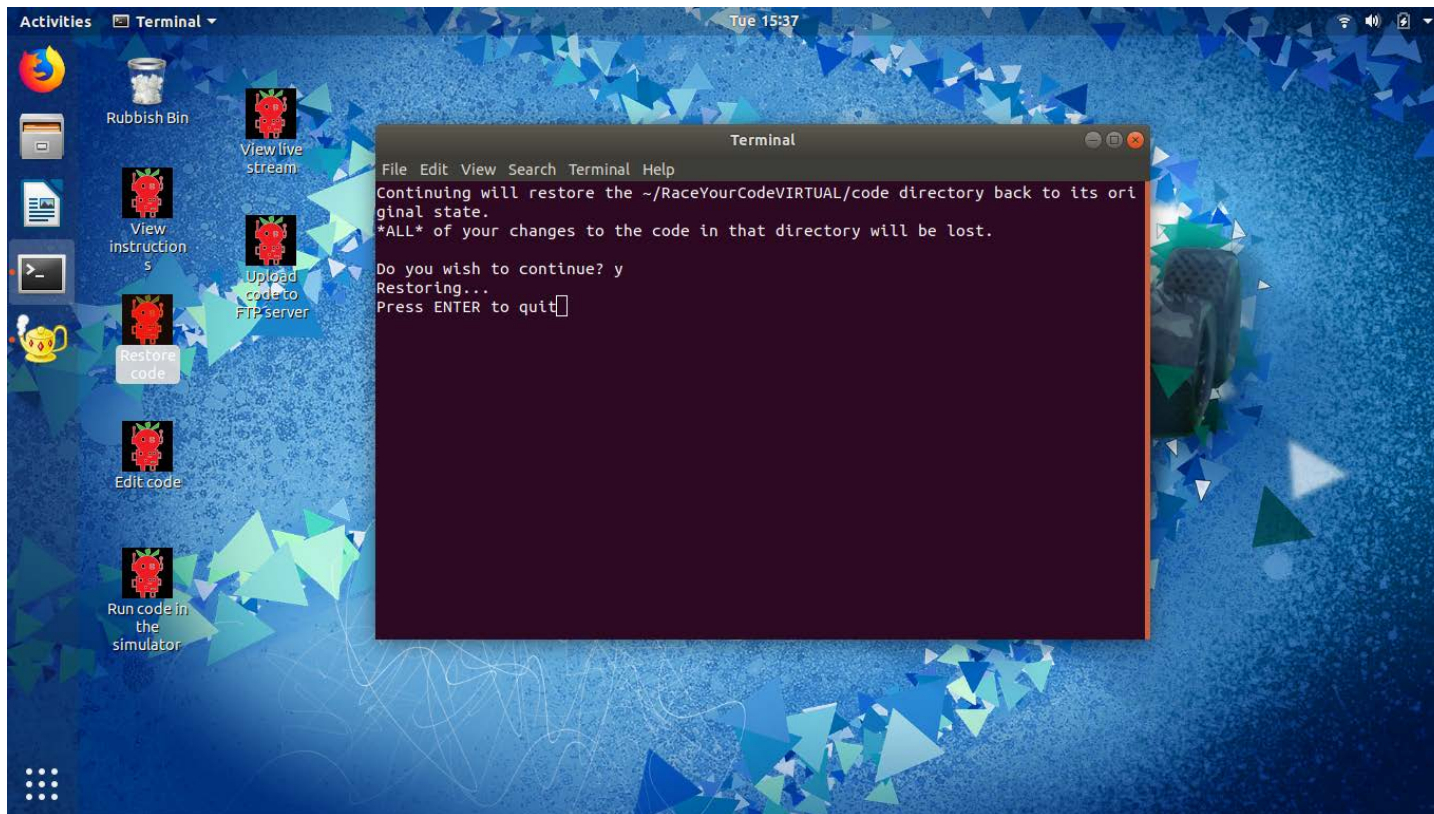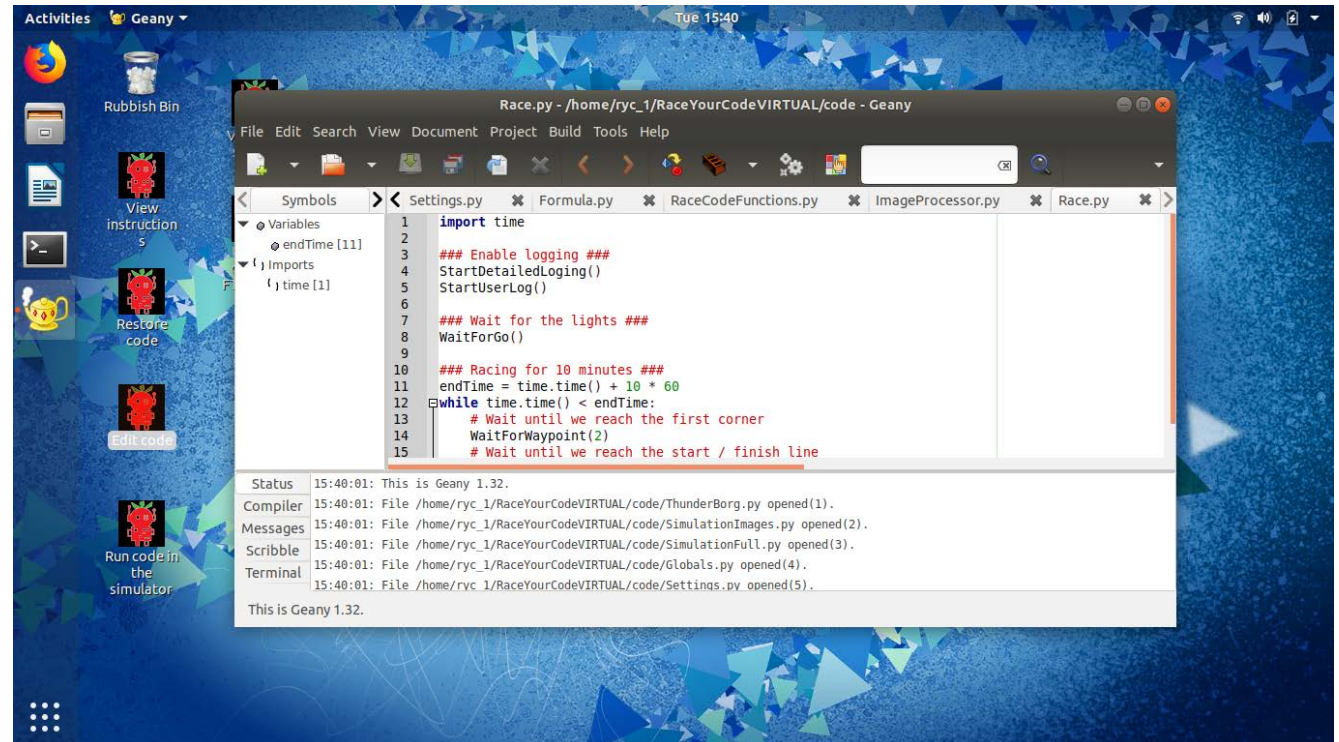
EDITING CODE

Edit code

This link opens the Geany editor with all the relevant python files opened in tabs.

The rightmost tabs have the code you will most likely wish to edit.

There are additional files that are not opened by default. Advanced coders may wish to edit these.

Code warning

The code is licensed for your use with a non-open source license, check the LICENSE.txt file for the full license terms.
For more information see https://www.formulapi.com/comment/240#comment-240

THE STANDARD CODE

Our standard code is made up of a number of files most of which is run in both the simulator and on the real robots.

There are more detailed explanations of the scripts and their parts in the Guides folder provided with the code. You should be given access details to all of the code for experimenting and development as part of your registration

Important note

Most of the code is common with our Formula Pi Monster series code. As such, there is a lot of useful information about the race code which may prove useful here:

https://www.formulapi.com/race-code

and also some insight into how our logic works here:

https://www.formulapi.com/tags/image-processing

SimulationFull.py /SimulationImages.py

These are explained in more detail later on, but these are scripts to help with development of the code without the robots, allowing you to run the bulk of the code using the provided simulator to generate camera data and track position.

Settings.py / Globals.py

These scripts hold settings and defaults respectively for the standard processing code. Most of the settings can be changed on the fly by overwriting Settings.py while the scripts are in control, but not all of them. The values in both of these files are shared between all of the scripts.

ImageProcessor.py

This is where the bulk of the code lives. This file is responsible for both the camera processing and the calculations for the motor output speeds. The code in this file is heavily math and OpenCV based, but has been kept fairly simple. If you wish to alter the image processing behaviour you will likely want to modify this script.

Race.py

This script runs alongside the image processing and is mostly intended for determining target track positions and other race type decisions. There are a number of examples provided for how this thread could be used, but it can access and alter most values in the image processing code as well.

RaceCodeFunctions.py

This script provides the simplified race API used in our Race.py examples. They provide functions such as logging and feedback from the processing code. A full list of the functions can be found here: https://www.formulapi.com/race-code/functions

ThunderBorg.py

This script provides a simple API to control the on-board ThunderBorg motor controller without the need to write I2C messages by hand.
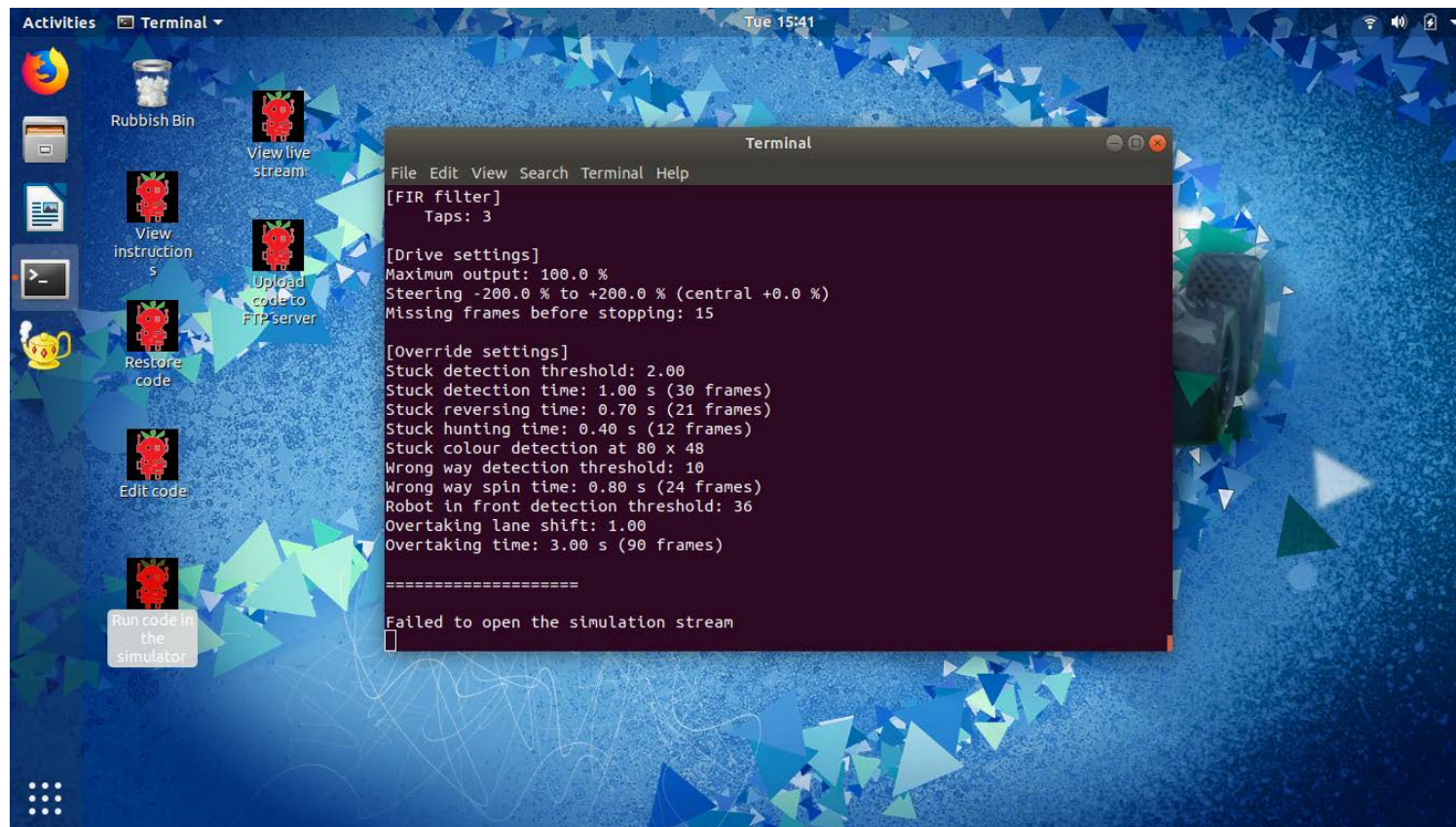
SIMULATING

Testing the code in the simulator

Run code in the simulator icon

This icon will start the simulator. The terminal window will state a failure. This is
not a concern – it just shows that the simulation stream has not started.

Start the simulation stream

The simulator shows the image the robot sees at the top left, the overview of the track at the bottom right and a top view of the robot displaying the both the current control outputs and the physics model output.

On the top right of the simulator, there is a button to start the video streaming. Click on this and wait for a few seconds. The green light should be visible on the robot outline (the top view of the robot)

Start the light sequence

Click on the "Green" button under the start settings. The lights should be visible in the robot view window. Then click the "Red" button. On clicking the "Green" for the second time, the robots should drive off.

Please note the light sequence needs to be Green–Red–Green

The robot should stop automatically after 10 minutes as per a timeout in the code.

Reset the robot positions

To reset the robots back to the start position within the 10 minutes, click on the "Reset" button under Bot simulation.

Close the simulator and re-open each time you want to make changes to code, or to re-run the light start sequence.

Note

You will need to close and restart the simulator after the 10 minutes has elapsed to run again.
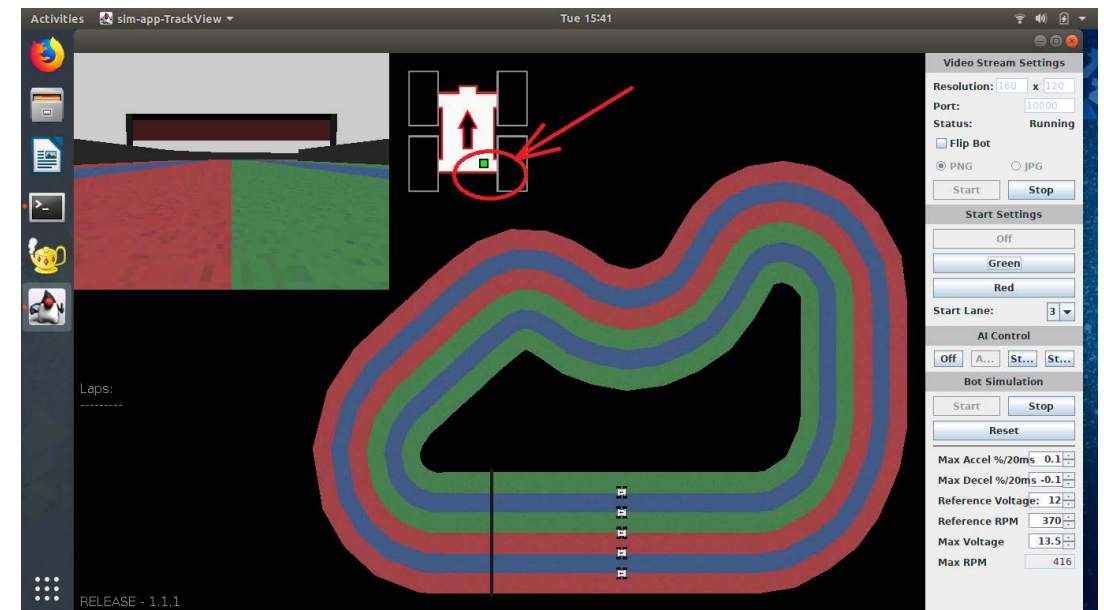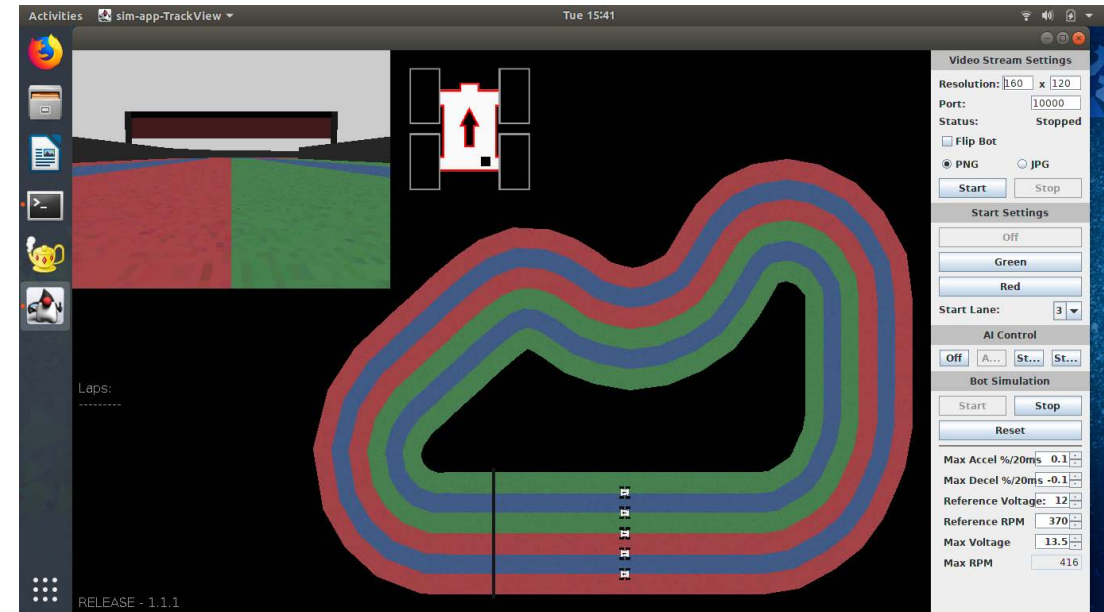
Simulator vs real world track

The simulator does have some limitations compared to the real robots:

Everything is evenly lit, there are no shadows or bright spots
Outside of the track is purely grey, in reality there are a lot of things in the background
The start lights do not project light like the real ones
There is no collision detection, the robots can run straight though the track walls
The robots are only roughly modelled, they do not behave exactly like the real robots

The final race will be on the real world track. It is possible to write code that works well in the simulator but not in real life. Please consider the real world aspects when designing your code.
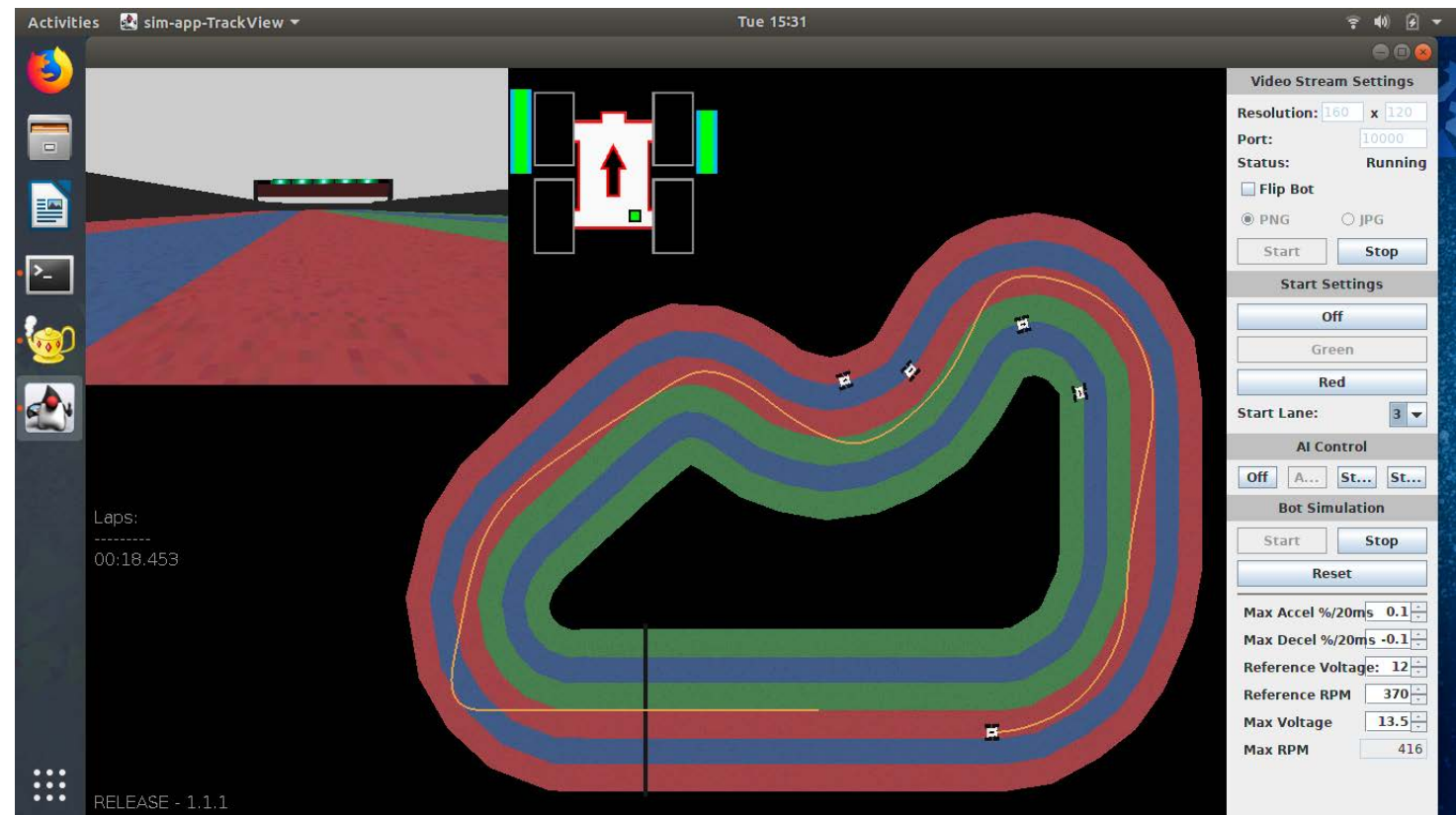
Logging

The terminal displays log information which is very useful for debugging. It displays information such as when the start line is detected, or the overtaking code has been executed. It also displays the current frame rate which should not deviate too far from 30fps.
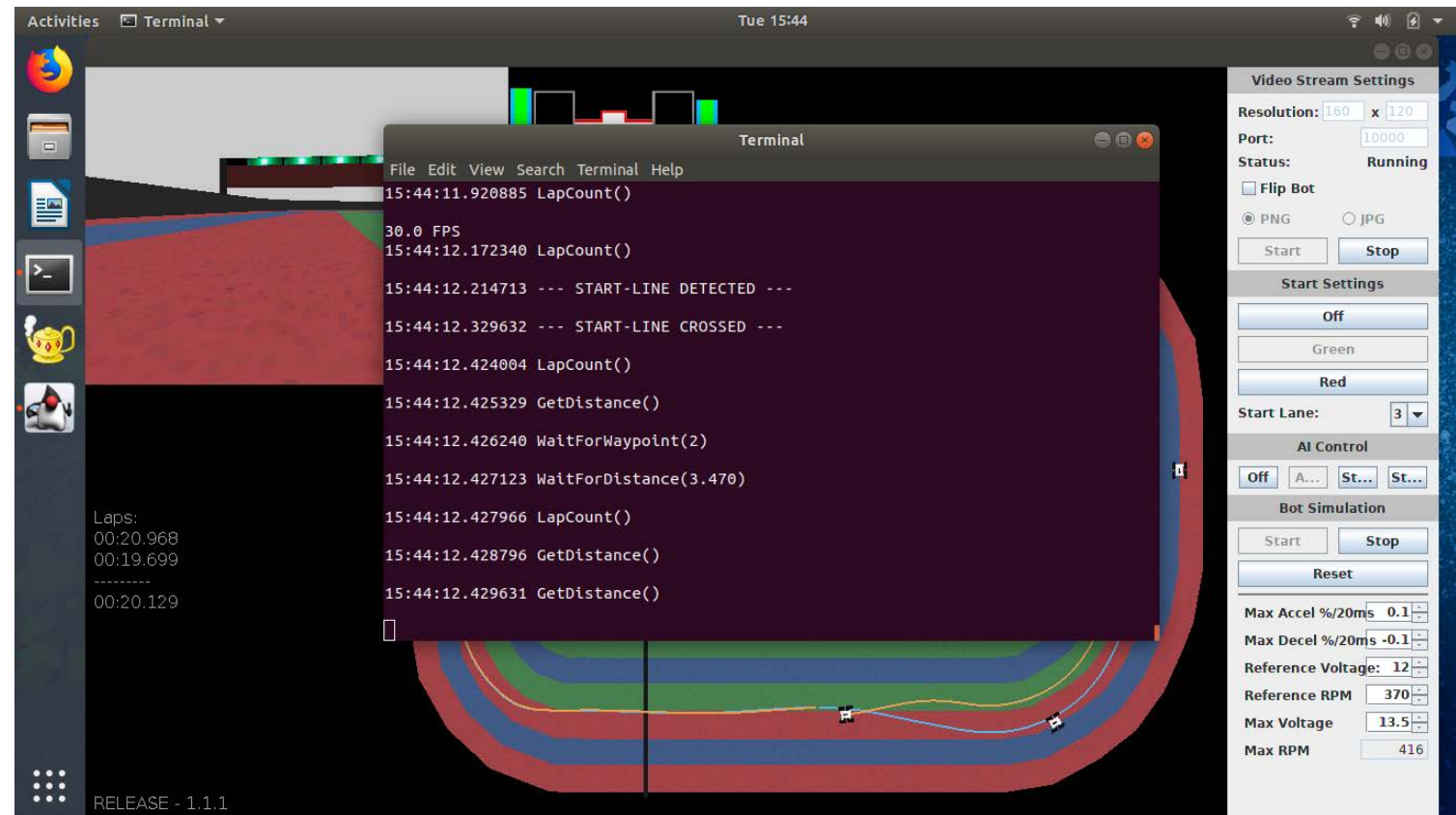
Your first code edit

One of the first things you may wish to do is to correct a bug in the code. Open the settings.py file from the editor. Change the value of the firstStraightMin to 1.0 and the value of firstStraightMax to 2.0

This change should prevent the situation where the robot drives off straight ahead after the start lights and crashes over the barrier as pictured below.
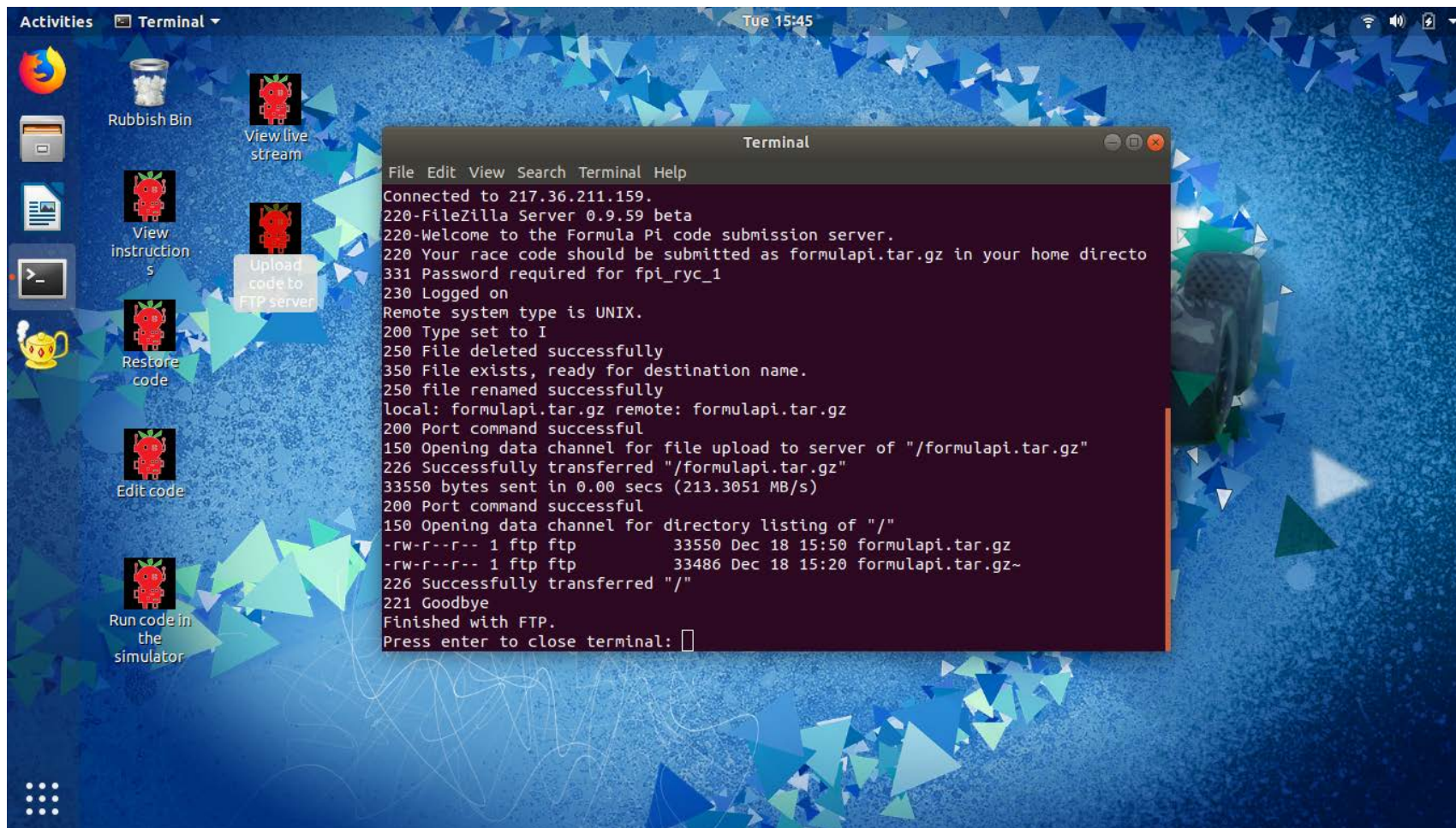
The simulator does not have collision detection, however there is in the virtual racing, meaning collisions will slow you down significantly.

Save the file, close and re-open the simulator. Now the robot should not experience this error.
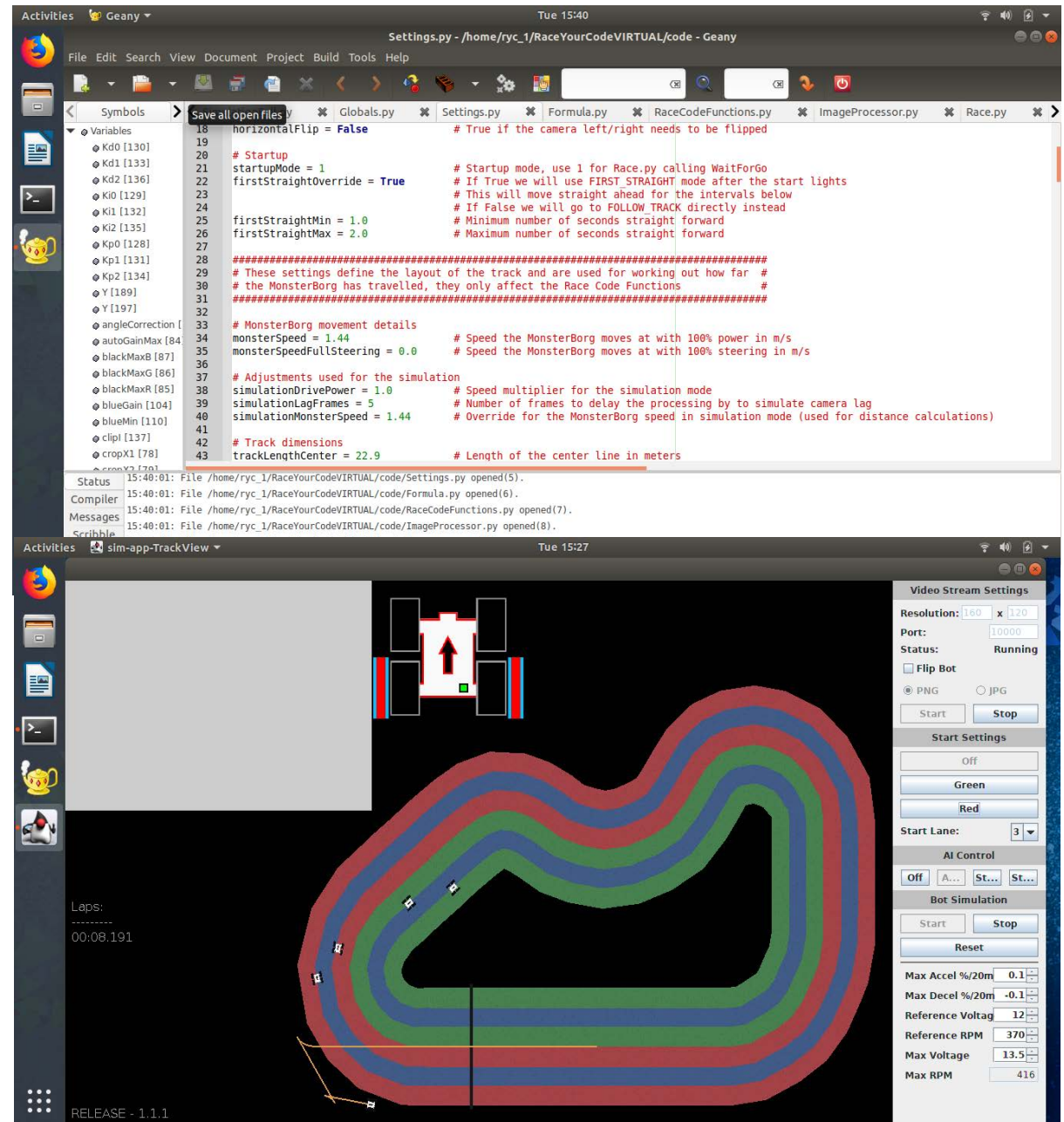
Upload the code

You will need to upload the code at least 24 hours before each race. Make sure to re-upload your code even if it hasn't changed since last race. The "Upload code to FTP server" icon will do this for you. Double check to see that the directory listing shows the formulapi.tar.gz file and that the date is correct (our server is GMT). The link creates a backup of the last uploaded code and calls this formulapi.tar.gz~

View the stream

The view live stream icon will open a firefox window connected to twitch.tv/raceyourcode. This is where we intend to stream the racing from. Please be sure to be there on 16th January for the test round 1. If you sign up for a twitch account, you will be able to use the chat room.

If we have problems with the stream we may use the YouTube channel which is open in the firefox browser on the next tab along.
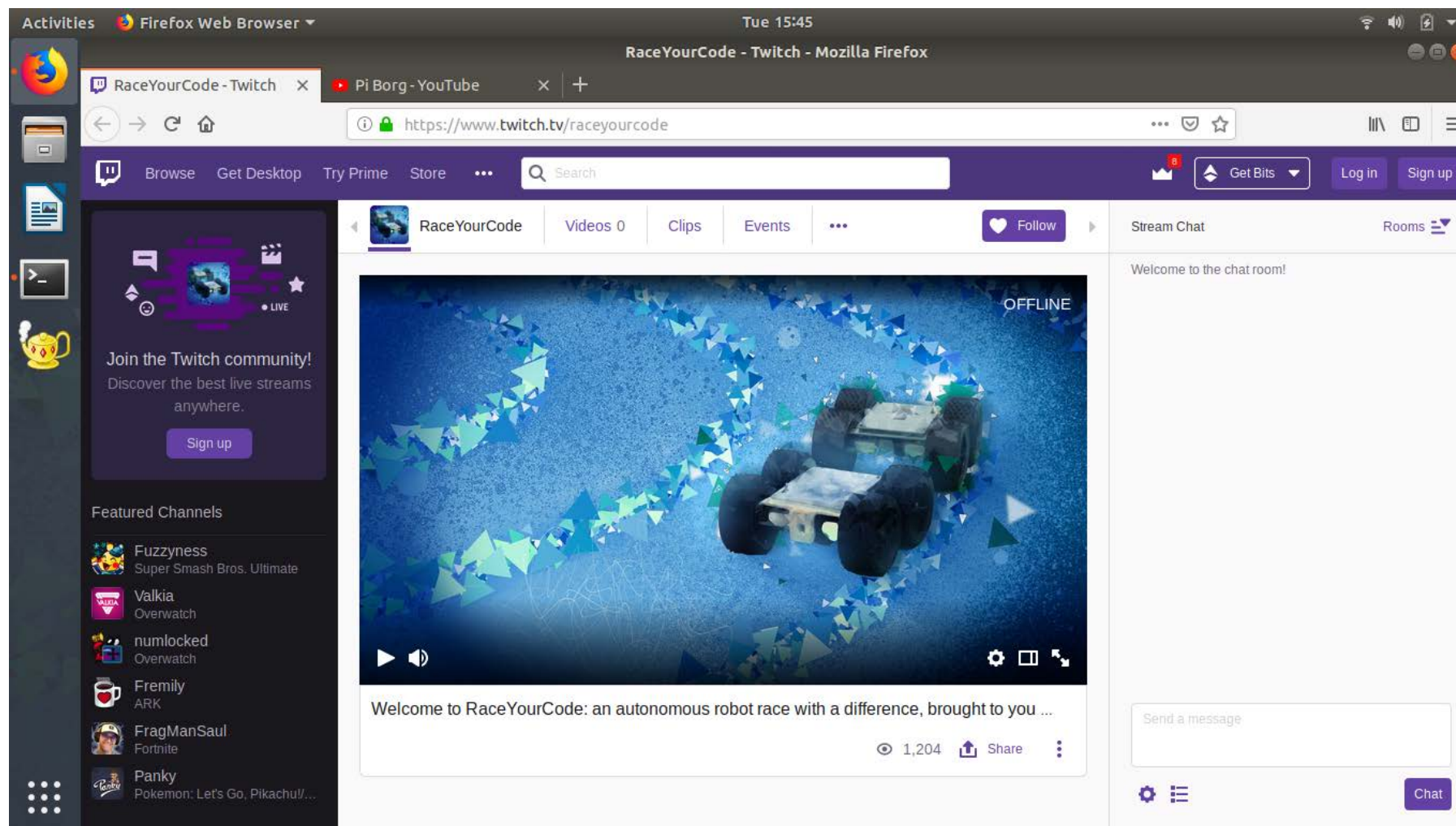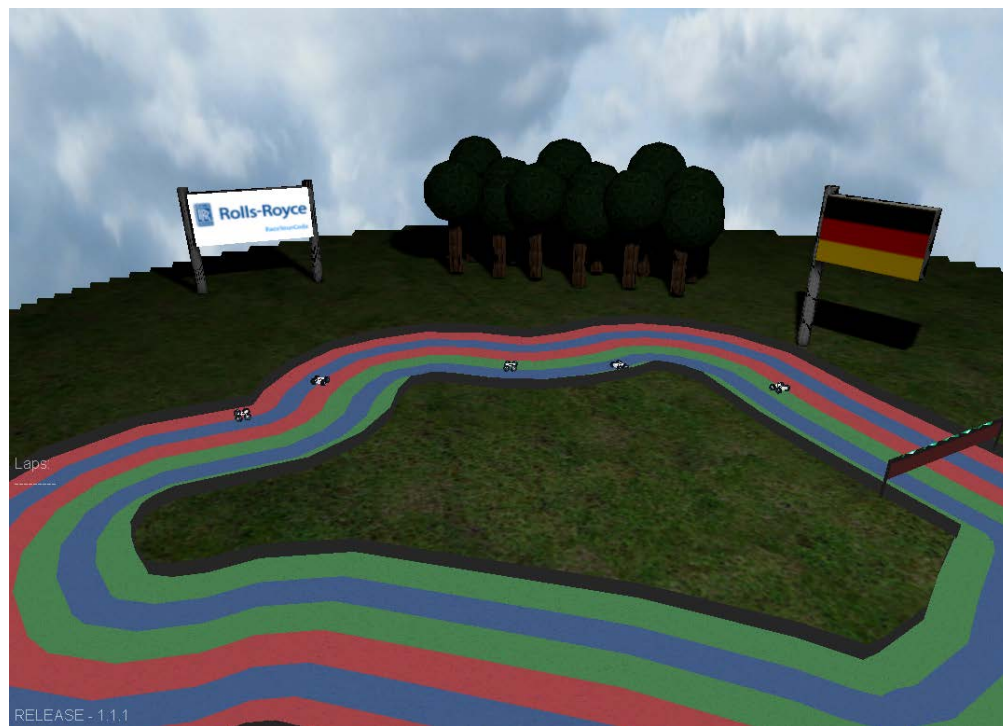
More information and help

Loads more information is available at www.formulapi.com

Click on the "Race Code" link at the top for detail information about the code and how it works. The "Race code processing results" shows a few simple animations which show the difference between offset, angle and curvature calculations

There is a specific RaceYourCode thread on the forum where you can post any questions you have.

The forums and the blog show lots of useful information.

Thanks for competing, and we'd love to hear your feedback!