TRIFORK.

ORBITLAB & TRIFORK

DonkeyCars Guide

TRIFORK.

PREREQUISITES

Clone the repo on your computer

```
git clone https://github.com/trifork/donkeycar-quickstart
```

Install Docker

```
https://docs.docker.com/get-docker/
```

Have functioning terminal (WSL, Mac terminal, etc)

Use the `donkeycars` wifi
Connect to the car with `ssh`

```
» ssh pi@<HOSTNAME>
```

Make sure the Xbox Controller connects

```
» sudo bluetoothctl

» disconnect <MAC-Adress>

» connect <MAC-Adress>

» paired-devices
» scan on
```

Hostname	Password
triforkblack	triforkBlack
triforkred	triforkRed
~~triforkblue~~	~~triforkBlue~~
orbitlabblack	orbitlabBlack

OBS notice the color starts with a capital letter for the password

TRIFORK.

Start the car with the following command

```
» cd mycar
» python manage.py drive --js
```

You should now be able to drive the car and se that it is recording. Recorded pictures should be in the `data` folder.

Move the data to your own computer using `scp`

```
» scp pi@triforkblack:/home/mycar/data/tup_1_25_04_22 <YOUR_FOLDER>
```

```
+------------------+--------------------------+
|     control      |          action          |
+------------------+--------------------------+
|     a_button     |       toggle_mode        |
|     b_button     | toggle_manual_recording  |
|     x_button     |   erase_last_N_records   |
|     y_button     |      emergency_stop      |
|  right_shoulder  |  increase_max_throttle   |
|  left_shoulder   |  decrease_max_throttle   |
|     options      | toggle_constant_throttle |
|      circle      | show_record_acount_status |
|        R2        |     enable_ai_launch     |
| left_stick_horz  |       set_steering       |
| right_stick_vert |       set_throttle       |
|  right_trigger   |      set_magnitude       |
|   left_trigger   |      set_magnitude       |
+------------------+--------------------------+

» throttle_scale: 0.49
» throttle_scale: 0.5
» auto record on throttle is enabled.
» recording: False
» recorded 10 records
» recorded 20 records
» recorded 30 records
» recorded 40 records
» recorded 50 records
» erased last 100 records.
```

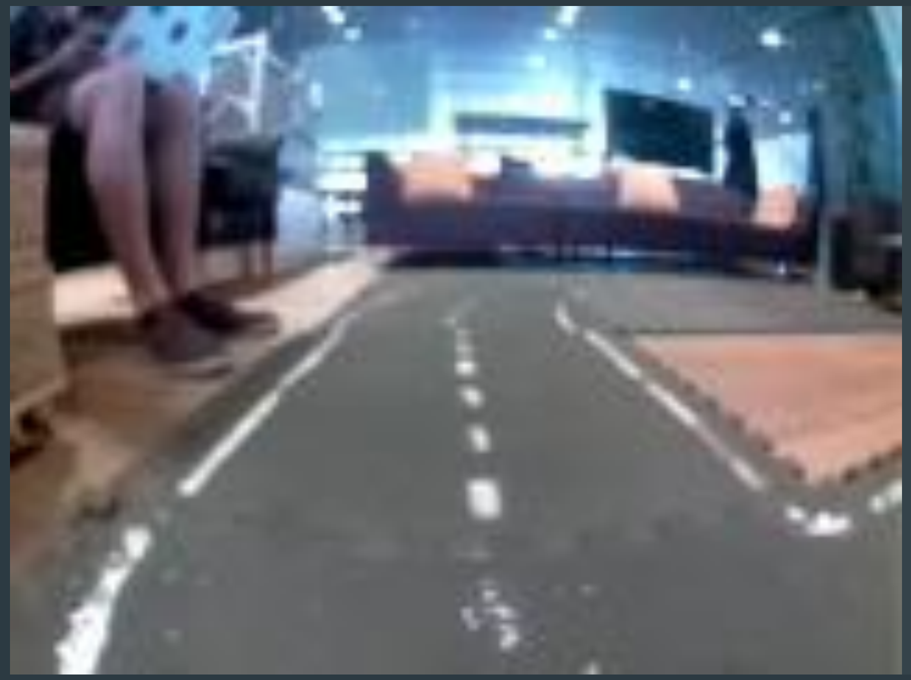
TRIFORK.

COLLECT DATA

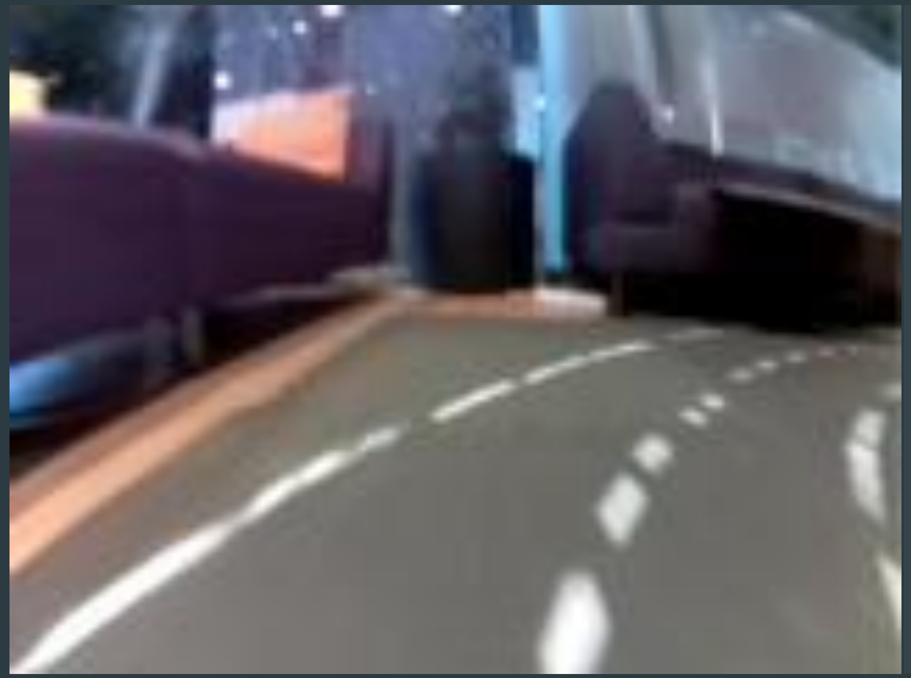
Make to capture diverse data. Your data determines what your car can learn.

Consider different configurations. Ex. Is what we see above the road useful?

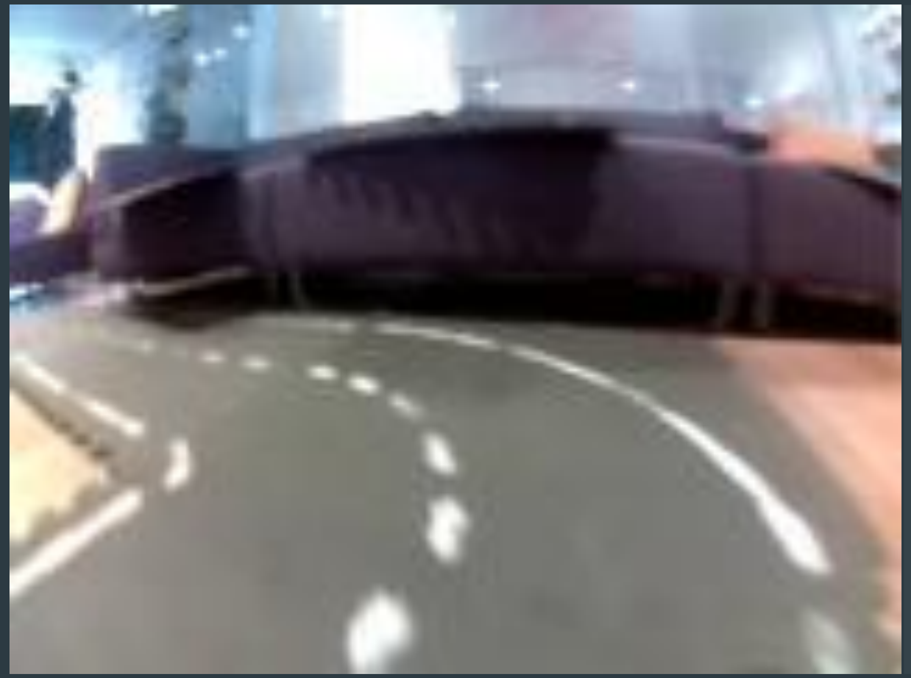
Good driving gives better data!



{"cam/image_array": "1_cam-image_array_.jpg", "user/angle": 0.0, "user/throttle": -0.5, "user/mode": "user", "milliseconds": 26724}



{"cam/image_array": "221_cam-image_array_.jpg", "user/angle": 1.0, "user/throttle": 0.06881923886837366, "user/mode": "user", "milliseconds": 56644}



{"cam/image_array": "468_cam-image_array_.jpg", "user/angle": -1.0, "user/throttle": 0.5, "user/mode": "user", "milliseconds": 104997}

The training and collection of data is configured using the `myconfig.py` file

Study the file to see all the possible configurations

Some configurations might not work because of data format or hardware

Use the same config for training and running the car!

```
# #TRAINING

# # tensorflow models: (linear|categorical|tflite_linear|tensorrt_linear)

# DEFAULT_MODEL_TYPE = 'linear'

# BATCH_SIZE = 128
# TRAIN_TEST_SPLIT = 0.8
MAX_EPOCHS = 20
SHOW_PLOT = False
# USE_EARLY_STOP = True
# EARLY_STOP_PATIENCE = 5
# MIN_DELTA = .0005
# OPTIMIZER = None                #adam, sgd, rmsprop, etc.
# LEARNING_RATE = 0.001           #only used when OPTIMIZER specified
# LEARNING_RATE_DECAY = 0.0       #only used when OPTIMIZER specified

…

# PRUNE_CNN = False  #This will remove weights from your model.
# PRUNE_PERCENT_TARGET = 75       # The desired percentage of pruning.
# PRUNE_PERCENT_PER_ITERATION = 20 # % of pruning that is perform per iteration.
# PRUNE_VAL_LOSS_DEGRADATION_LIMIT = 0.2 # The max validation loss during pruning.

…

# # Region of interst cropping
# # only supported in Categorical and Linear models.

# ROI_CROP_TOP = 0 # rows of pixels to ignore on the top of the image
# ROI_CROP_BOTTOM = 0 # rows of pixels to ignore on the bot of the image

…

#JOYSTICK

# USE_JOYSTICK_AS_DEFAULT = False
# JOYSTICK_MAX_THROTTLE = 0.5
# JOYSTICK_STEERING_SCALE = 1.0
# AUTO_RECORD_ON_THROTTLE = True
```

Build and run the `donkeycar-quickstart` program

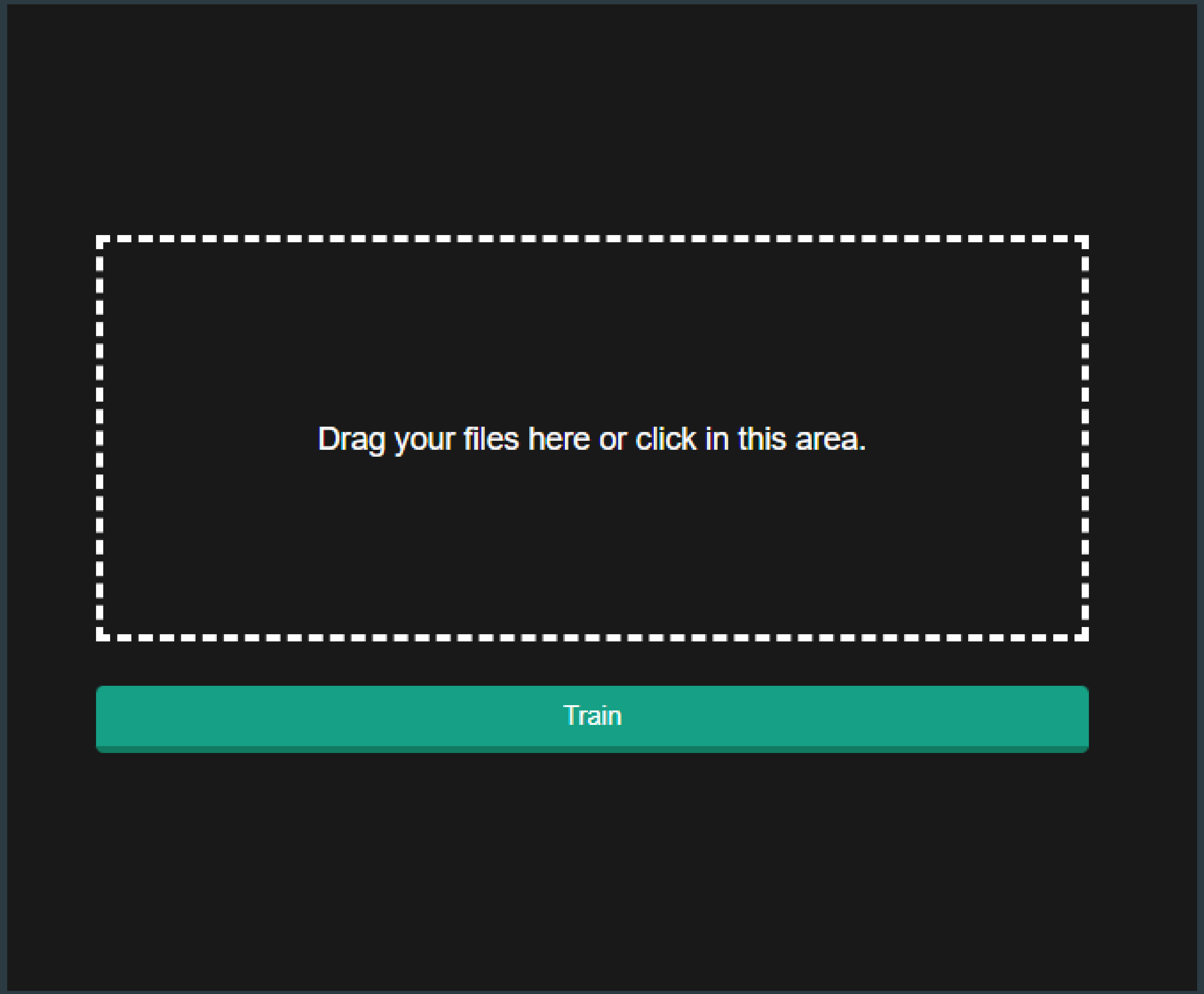
```
» docker image build -t donkey_train_web_api .
» docker run -p 5000:5000 -d donkey_train_web_api
```

You should now be able to see the program at `localhost:5000`

See the logs

```
» docker log -f <CONTAINER-ID>
```

The finished model should automatically download



TRIFORK.

Move the finished model to the car

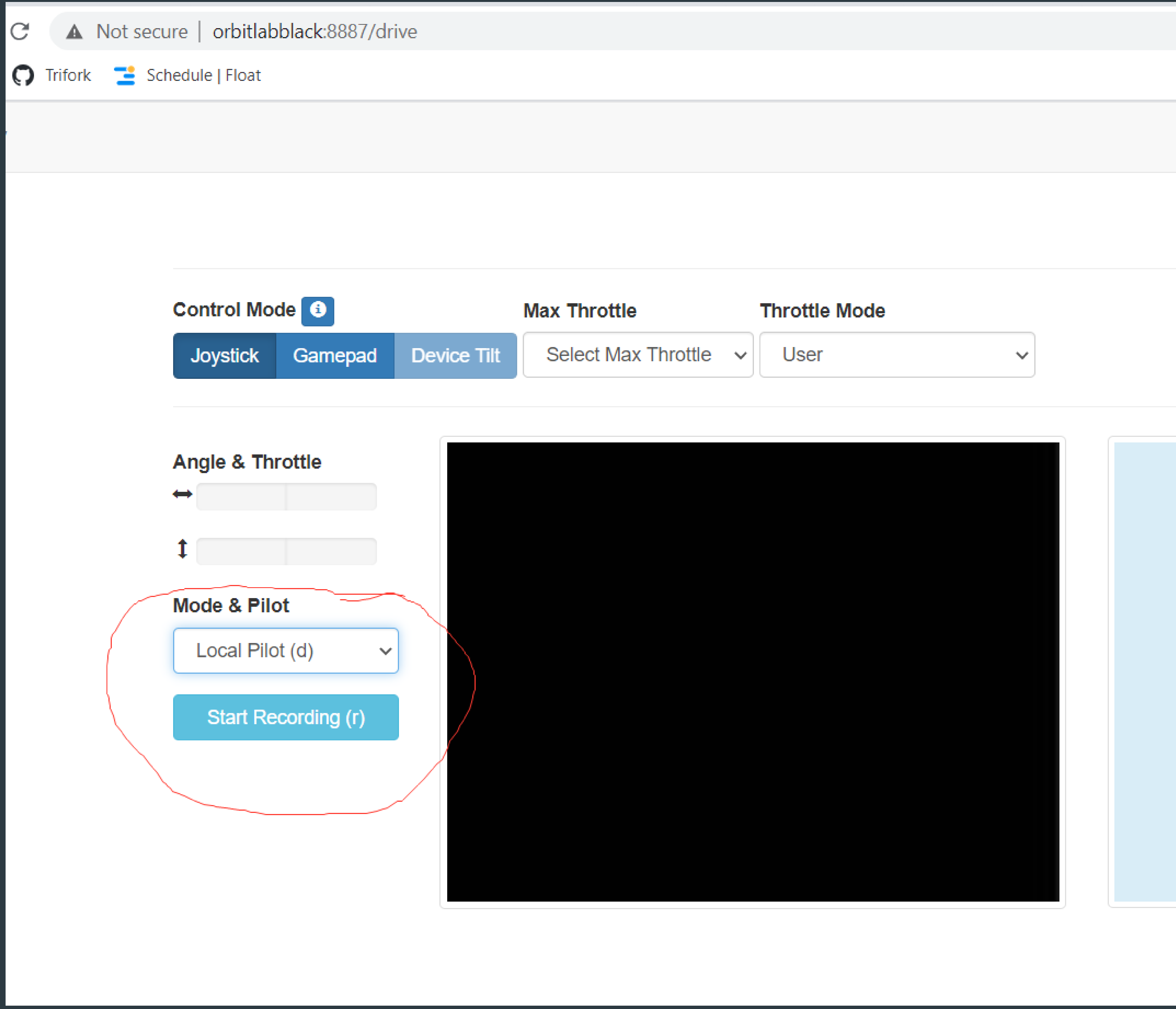
```
» scp <YOUR_FOLDER>/model.h5 pi@triforkblack:/home/mycar/models
» scp <YOUR_FOLDER>/myconfig.py pi@triforkblack:/home/mycar
```

Start the car with the model

```
» cd mycar
» python manage.py drive -model ./models/model.h5
```

```
» You can now go to <your hostname.local>:8887 to drive your car.
» Starting vehicle at 20 Hz
```

Start a browser and go to `<your car>.local:8887` **and, when you are ready, under** `Mode & Pilot` **select** `Local Pilot (d)`


Not secure | orbitlabblack:8887/drive

Control Mode: Joystick | Gamepad | Device Tilt

Max Throttle: Select Max Throttle

Throttle Mode: User

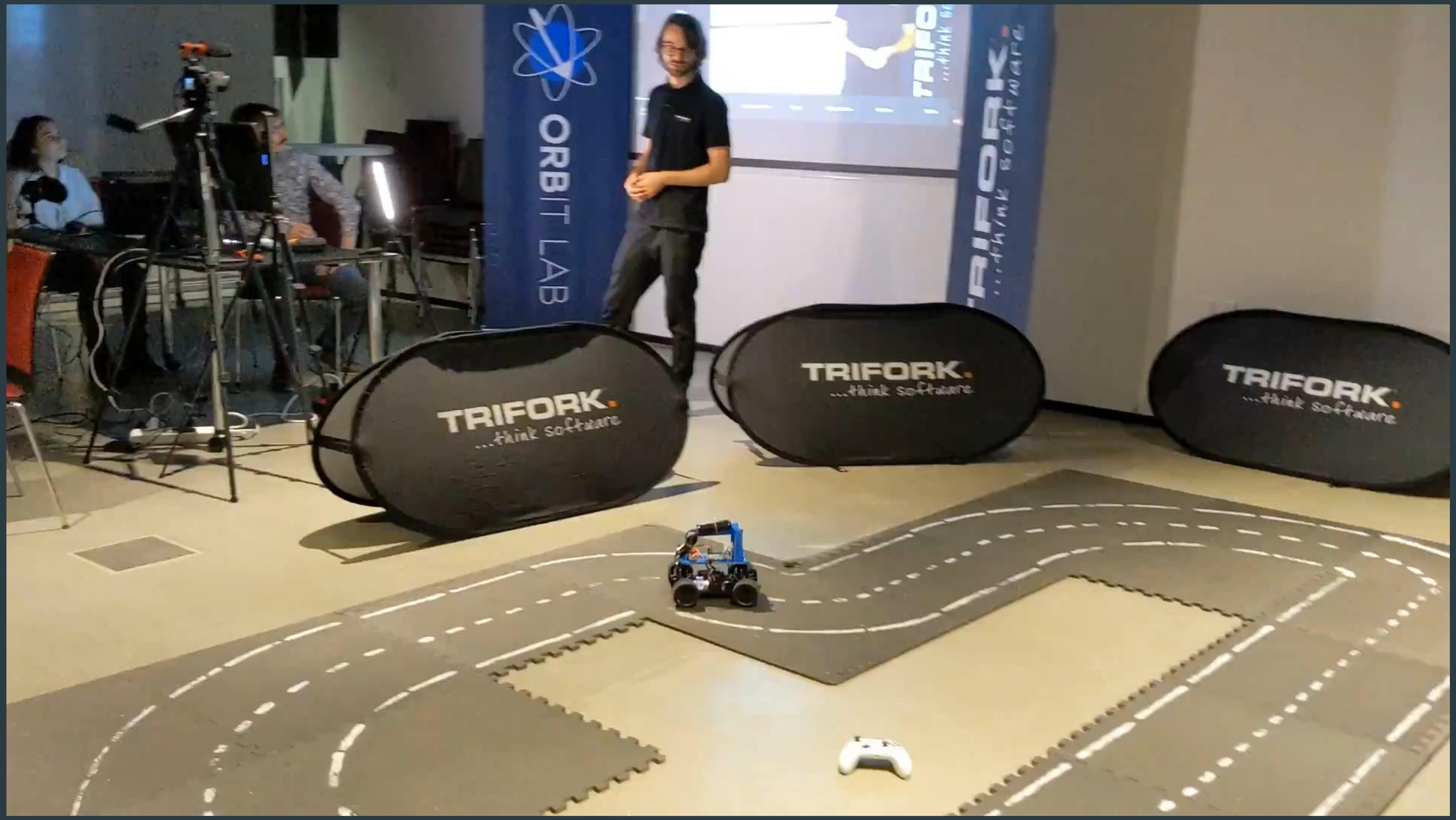
Mode & Pilot: Local Pilot (d)

COMPETE!

Each team gets 5 mins to drive their car around the track.

If the car goes off the track it can be put back at the point it went off

Most rounds win!



TRIFORK.