

## YetiBorg Qualification Challenge

26<sup>th</sup> April 2022, the Time-Schedule will be sent by email this Monday at around 12.00.

The main goal of the YetiBorg Qualification Challenge is to ensure that each team is able to remotely control the assigned Yetiborg both manually and by automatic script. It consists of two parts:

1. Manually drive your YetiBorg to a position in front of one of the small pictures at the perimeter of the environment of the track and take a photograph of it.
2. Manually drive the YetiBorg to one of the 2 positions on the track (indicated by the green circles) with its camera headed in the direction of the blue arrow (See Figure 1.). From this position the YetiBorg should drive by itself by starting the drive.py script.

**Deliverables:** Send the picture made in Step 1 and report your findings for Step 2 by email to [erwin@liacs.nl](mailto:erwin@liacs.nl) within 2 days of your session (See schedule sent separately by email ).

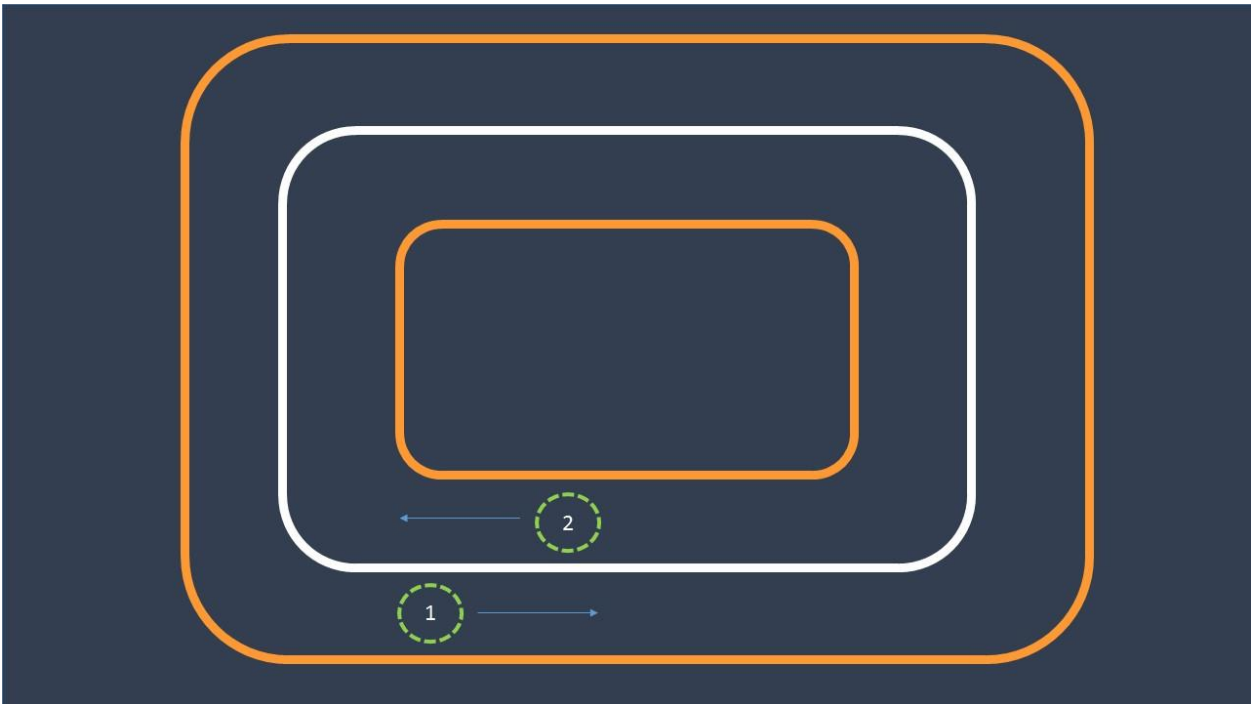


Figure 1. Overview of the YetiBorg track. During the Final Challenge, the YetiBorg should always drive in between two lines in the direction of one of the blue arrows, always having a white line to its left and an orange line to its right. The green circles indicate the 2 starting positions. The yellow arrows roughly indicate the positions of the small pictures.

### Some characteristics of the track:

- The lines are ~3 cm wide and are made of somewhat reflective white and orange tape. They will not be completely smooth and are subject to change.
- The background color will be light yellow/beige.
- The width of every lane of the track accommodates one YetiBorg, but only just.
- The area of the bounding box of the track is about 3m x 3m.
- During the Qualification Challenge the shape of the track will be as depicted, during the final Challenge it will have a similar shape.

### Getting Started

- See the schedule ( sent by email, separately ) for the time slot and credentials for the Yetiborg that has been assigned to your team. Please follow the schedule strictly, i.e., do not try to login outside of your team's time-slot.
- Connect to the Robotics Classroom by Kaltura. Here you will see almost a complete overview of the track and can follow the movements of your and other YetiBorgs.  
I will be monitoring the class room on a regular basis. If something needs my attention, please use the chat to notify me. If not in a meeting, I will try to respond as soon as possible.
- Connect to the YetiBorg using ssh.
- Go to the directory `/home/pi/teamx`, with x equal to your team number.
- Furthermore you will find code for manually controlling the YetiBorg, and initial code for automatically driving around the track.
  - You can manually control the Yetiborg by issuing the command:

```
python manual_yetiborg.py
```

 (Thanks to [1])

After pressing a random key you can control your YetiBorg by pressing and holding the key 'w', 'a', 's', 'd'. The program can be stopped by pressing 'q'.

- A still image can be taken by using the command:

```
raspistill -o cam.jpg
```

- Initial code is provided for automatically driving around the track. Note that it assumes the YetiBorg is located on the track. Only start this code when you manually drove the YetiBorg to either Position 1) or 2) on the track The code is started by issuing the command:

```
python yetiRace.py
```

- Use CTRL-C to interrupt and stop the code, if you notice the YetiBorg drives off the track.

Note: The following necessary packages already have been installed:

- `bash <(curl https://www.piborg.org/installer/install-picoborgrev.txt)`
- `bash <(curl https://www.piborg.org/installer/install-zeroborg.txt)`
- `bash <(curl https://www.piborg.org/installer/install-yetiborg-v2.txt)`
- `sudo apt-get -y install python-picamera`
- `sudo apt-get -y install libcv-dev libopencv-dev python-opencv`

For further information and code examples please see the related articles on page:

- <https://www.piborg.org/robots/yetiborg-v2>

### **Optional: Prepare for the final YetiBorg Challenge.**

During the Final Challenge, the YetiBorg should drive in between two lines in the direction of one of the blue arrows (see Figure 1), always having a white line to its left and an orange line to its right. The green circles roughly indicate the 2 starting positions (note: they are not present at the actual track). During the drive it should try to recognize several traffic signs and obey traffic lights.

If there is time left during your session you may want to try to change/improve the initial code so that the YetiBorg drives around the track as required during the Final Challenge.

### **Important Final Remarks:**

- 1. NB Be sure to always copy and save your code to your own PC by using ssh or a program like WinSCP or Putty. Delete your own code (only) just before the end of your session.**
- 2. Drive your YetiBorg to the lower right corner of the camera image of the track and shut the system down ( command: `shutdown` ).**
- 3. The YetiBorgs are made available on my home network. This network is also used by several other students working on their BSc or MSc projects. So please use it only during the assigned time-slot and act responsibly.**

### **References:**

[1] Manual\_yetiborg.py code by J. Hasenoot, M. Zwetsloot, E. Karakas.