YetiBorg Qualification Challenge

3rd May 2021, the Time-Schedule has been sent by email earlier.

The main goal of the YetiBorg Challenge is to develop an automatic script that can drive the YetiBorg in either direction of the track depicted in Figure 1, whereby:

1. The YetiBorg stays with all its wheels within its track. Starting from Position 1 and driving in the direction indicated by the blue arrow, the YetiBorg always holds the outer orange line to its right and the inner white line to its left, and similar rules apply when the YetiBorg starts from Position 2. Wheels should not completely cross an orange or white line.
2. There are traffic lights that automatically alternate between red and green lights, for each lane. The YetiBorg should automatically stop for a red light, and start driving again when the light is green.

Deliverables: Send your .py script, that implements the above, together with explanatory comments and instructions in a single .zip file by email to erwin@liacs.nl before 24th May 2021.

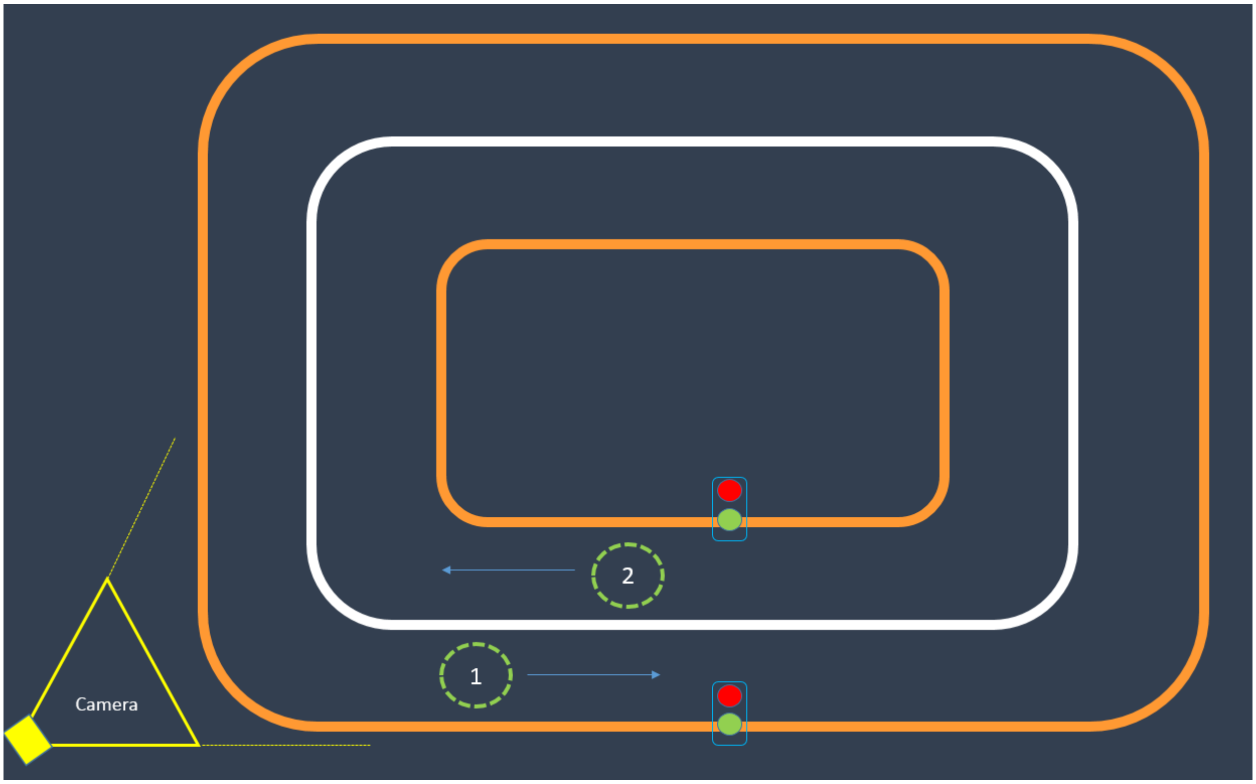

Figure 1. Overview of the YetiBorg track. During the Final Challenge, the YetiBorg should always drive in between two lines in the direction of one of the blue arrows, always having a white line to its left and an orange line to its right. The green circles indicate the 2 starting positions.

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.

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/team_NAME, with NAME equal to your team name.
- 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
      ```

    - If you notice that the YetiBorg is heading in the wrong direction use **`CTRL-C`** to interrupt the code, and stop the motors using the python script stop.py, i.e., issue the command: `python stop.py`
    - See also a short video of the debug-mode (not easily available through ssh) on the website to get an impression of the working of the program.

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-1/yetiborg-v2

Important Final Remarks:

1. **NB Be sure to always and regularly 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. **Note that copy and pasting the ssh command from the pdf with the schedule introduces some invisible control characters in the command leading to a permission denied. It is advised to enter the ssh command by typing.**
3. **Drive your YetiBorg to the lower right corner of the camera image of the track and shut the system down (command:** `shutdown` **).**
4. 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.

Grading:

Passing the Qualification Challenge is rewarded with 3 points. The rest of the grade is based on the performance and quality of the delivered code, the implemented methods and ideas to solve the challenge, and the comments and instructions for the code.

References:

[1] Manual_yetiborg.py code by J. Hasenoot, M. Zwetsloot, E. Karakas.