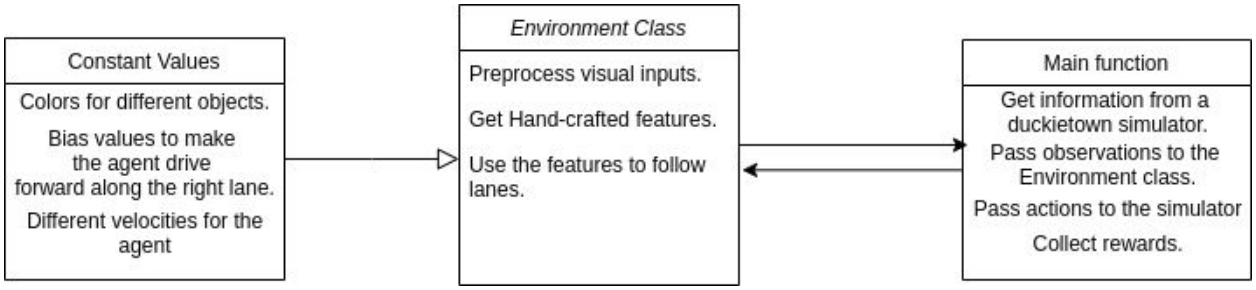
CS5478 Final Project

Shicheng Chen A0215003A

May 2020

The below figure shows the system diagram:



How the system processes the visual input?

I try to detect the yellow middle line and two white lines of the lane from observations (visual input).

1. I used line detection algorithms from OpenCV and RGB values from each pixel to detect these lines(middle yellow line and two white lines).
2. Then, I used morphological transformation operations to refine the results, which can help me to remove background information.
3. Finally, I resize the image to a small size to speed up hand feature extraction operations.

How it determines the position with respect to the lane

I want the agent to drive forward along the right lane in each tile. Firstly, I shift the whole image and then follow the middle yellow line, which means that the agent is actually drive in the right of the central line.

To follow the middle yellow line, we need half of the yellow line to be on the left of the image and the other half of the line to be on the right of the picture.

1. We assume the yellow line contains 100 pixels. We hope 50 pixels on the left of the image and 50 pixels on the other side.
2. Because we have a perspective view, so we need to refine the yellow line to be a single line to prevent a large number of pixels near to the agent and fewer pixels far from the agent.

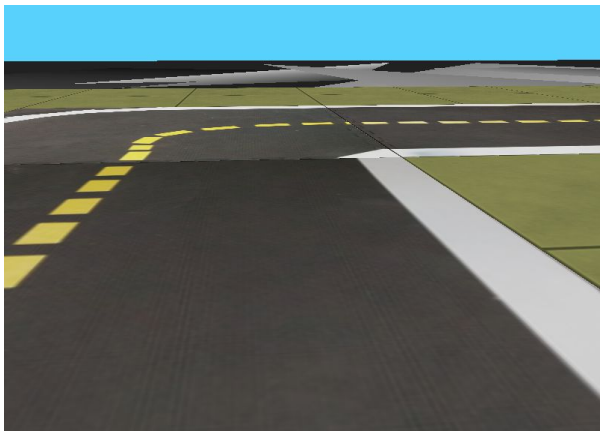
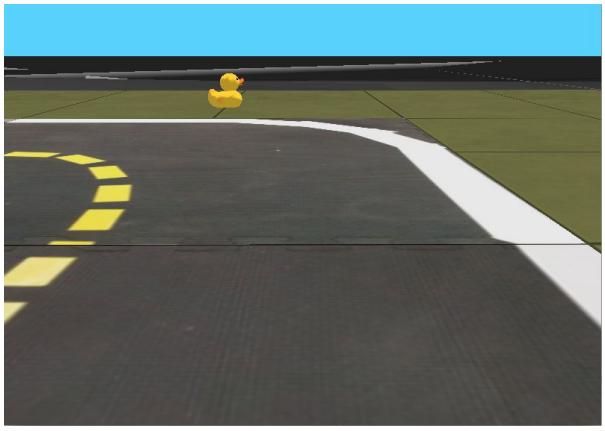
How it controls the vehicle

We assume the yellow line contains 100 pixels. LP represents that the left image contains LP yellow pixels. The right image contains RP yellow pixels.

1. If |LP-RP|<=10, the vehicle will go straight.
2. If LP-RP>10, the agent will turn right to balance the pixel number between two sides of the image.
3. If RP-LP>10, the car will turn left.

4. If the yellow line is straight, the car will move fast.
5. Curve yellow lines will make the car move slower.

Bag of tricks to get better rewards

1. The agent drives forward along the right lane and turns right to get more rewards than turning left, as shown in the above figures. I found that the car in the situation, as shown in the left figure, get more rewards than that of the right figure. Therefore, I come up with a constraint. If the agent does turning left operations continuously, I will turn the agent around to make the agent run in the other direction.


2. The agent gets more rewards when it drives along the straight lanes (as shown in the left figure). Too many sharp turns decrease the rewards that the agent receives (as shown in the right figure). Therefore, our agent tends to drive along the straight lanes. If there are too many sharp turns, the agent will turn around and look for straight lanes.
3. Velocity is the most critical hyper-parameter to get better rewards. Although I did not use reinforcement learning successfully, I used a grid search method for better hyper-parameters such as angular velocity and linear velocity.