Software and Perception Team Tasks

Deadline: Midnight March 1, 2020

1 GENERAL INSTRUCTIONS

The following tasks are intended to check your skills in the field of Computer Vision and Path Planning. You'll receive the invitation to a google group for discussion during the task round. You're encouraged to figure out the problems on your own, however, feel free to contact Yash Soni, Debjoy Saha or Archit Rungta, visit the lab or create a thread on the google group if you feel you're not making any progress. We will also have routine meetings to check on your progress. The final code must be submitted on a github repo in C++ or Python along with appropriate documentation. The documentation format will also be provided to you. We had fun in creating these tasks and we hope that you'll enjoy them too. Do not copy code from friends as we will use automated plagiarism checkers. Also, if you use any library functions, you might be asked to explain in detail how it works. Ping us for more problems if you don't find this challenging enough. Good luck!

2 Basic Tasks

2.1 Tic-Tac-Toe Minimax Environment

In the first task you have to write an agent to play the Tic-Tac-Toe game using minimax algorithm. Minimax algorithm basically is just an exhaustive search of your game space. You'll be using the gym-tictactoe environment. If you want, you may implement more advanced methods like Q-Value Learning and Genetic Algorithms. Feel free to contact us if you have any trouble in getting the environment setup. A skeleton for player-vs-minimax agent is provided at here.

2.2 Billiards Game Prediction

Given images of a billiard pool table with the cue and ball to be hit indicated find pairs of collisions assuming perfectly elastic ball-ball collisions and ball-table collisions. Obviously, the balls' centers might not be in-line and this would lead to two divergent ball movements. To simplify this, we assume that in case of ball-ball collision, the incoming ball stops and the incident-ball moves with the same velocity as the incoming ball.

2.3 Face Detection and Motion Tracking

For the first task you're supposed to implement algorithms to detect a face and to track its motion. You can use OpenCV's inbuilt function for detecting faces, however, you're encouraged to think of methods to detect faces while not relying on inbuilt face detection functions. For tracking the motion of the face, you may not use any inbuilt tracking methods in OpenCV. Once you have detected and have working face tracking methods implement a game where the user has to navigate his face through obstacles. This video(only the first ten seconds are relevant) is an example of one such implementation. You can play with that example as the Flappy bird filter on Instagram.

3 Advanced Tasks

3.1 Path Planning for AI agent in Game Environment

In this task you're given a game environment and a basic strategy for movement in the same environment. This environment can be setup to run on either Windows or Ubuntu. Your task is to write a strategy for your agent that can guide it to weapons in the complex maps that the environment provides. Furthermore, code your agent to also pick up a weapon and fight. We will test out different strategies against one another so you can spend as much time as you want on improving your strategy for this game.

3.2 Tracking Multiple Objects

Imagine you're writing the code for a military aircraft radar. Your electrical engineers have given you a csv file containing the detected 3D location of enemy air-crafts at all given time frames. However, this detection is not very reliable. There are time-steps on which an aircraft might not appear on the radar, on other instances - a bird or civilian aircraft might pop up as an enemy aircraft. In general the problem is that your detection might report false positives for some small spans of time, and it might miss true positives for some small time stamps. Your job is to filter out all such false positives and report the count of what you think are true positives over all time stamps. Also, report the entire trajectory for some subset of these true positives .

4 Helpful Links

- Install Ubuntu
- Basic Terminal Tutorial
- Git Online Practice