Final Project due Dec 11, 2020 16:00 PST
starter code

The final project is open-ended. The task is to program a SuperTuxKart ice-hockey player.

In `agent/player.py`, implement the class `HockeyPlayer`. This class provides an `act` function, which takes an image (in `numpy` format) and the state of the player (ground truth position and camera parameters). You should return an action (steering, acceleration, ...). `random_agent` contains an example of a random hockey player than just drives straight with minor random perturbations.

We also provide you with a hockey tournament script. Run it with

```
python -m tournament.play player1 player2 ...
```

where `playeri` is any agent you code up (e.g. `agent`) or `AI` to use the game AI to play. The teams of the agents alternate. For example

```
python -m tournament.play random_agent AI random_agent AI
```

sets up a match between two random agents and two AI agents.

Use the `--save` flag to store your agent's output.

The rules

This project is completely open-ended, however, there are a few ground rules:

1. You may work in a team of up to 4 students.
2. Teams may share data, but not code.
3. You may organize test races between each other. However, make sure to delete

your agents from the shared filesystem after the tournament (no sharing of code).

4. In our grader, your agent will run in complete isolation from the rest of the system. Any attempt to circumvent this will result in your team failing.
5. You may use `pystk` and all its state during training, but not during gameplay in the tournament.
6. Your code must be efficient (< 100ms / step) on a **CPU**. There is no GPU access during testing.

The tournament

We will play an ice hockey tournament between all submissions. Two of your agents will play an opponent. Your agents will run in the same process and may communicate with each other. Each game will play up to 3 goals, or a maximum of 1200 steps (2 minutes). The team with more goals wins. Should one of the teams violate the above rules, it will lose the game. Should both teams violate the rules, the game is tied.

Depending on the number of submissions, we will let all submissions play against each other, or we will first have a group stage and then have the top 10 submissions play against each other. The submission with the most victories wins the tournament. The goal difference breaks ties. Additional, matches further break ties.

Grading

- 40pts originality of the idea. How many things did you try? Do you understand how your strategy works?
- 30pts average number of goals scored per game. Linearly from 0 to 1. We will compute the average against the game AI or the rest of the class and use the better of the two.
- 30pts writeup. Structure of the writeup, exposition, how well is the idea motivated and explained.

- 15 pts extra credit for the top 3 teams in the competition (15pts winner, 10pts second, 5pts third).

Hints

- Test your code on a Linux machine (either CS lab or colab).
- We will organize several mock tournaments before the final tournament. Make use of them.

Submission

Once you finished the project, create a submission bundle using

```
python bundle.py agent [YOUR UT ID]
```

Make sure to place your writeup in the `writeup` directory, and remove any non-essential files from it.

Grading

100.0 points possible

Please upload this assignment through canvas.