Robotic Navigation and Exploration

HW2: Path Tracking

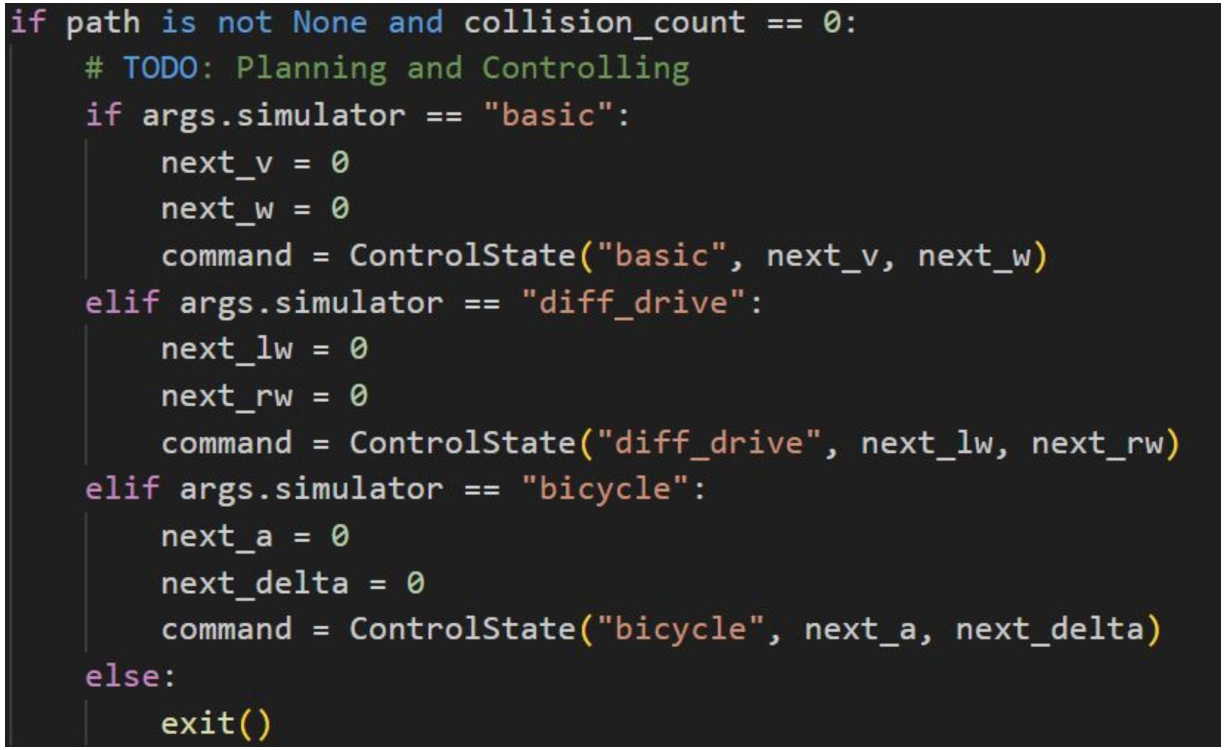
Min-Chun Hu anitahu@cs.nthu.edu.tw
CS, NTHU

Score

- **Path Tracking**
  - PID (Basic / Differential Drive / Bicycle) (5+5+5=15%)
  - Pure Pursuit (Basic / Differential Drive / Bicycle) (10+10+10=30%)
  - Stanley (Bicycle) (10%)
  - LQR (Basic / Differential Drive / Bicycle) (10+10+10=30%)
- **Collision Handling** (5%)
- **Report** (10%)
- **Deadline: 2025/03/16 (11:59 pm)**
- **Homework Upload**
  - zip the **full program code** and **report.pdf** into **[student id]_hw2.zip** (without brackets)
  - 10 points will be deducted for incorrect naming or filename extension

Control & Planning

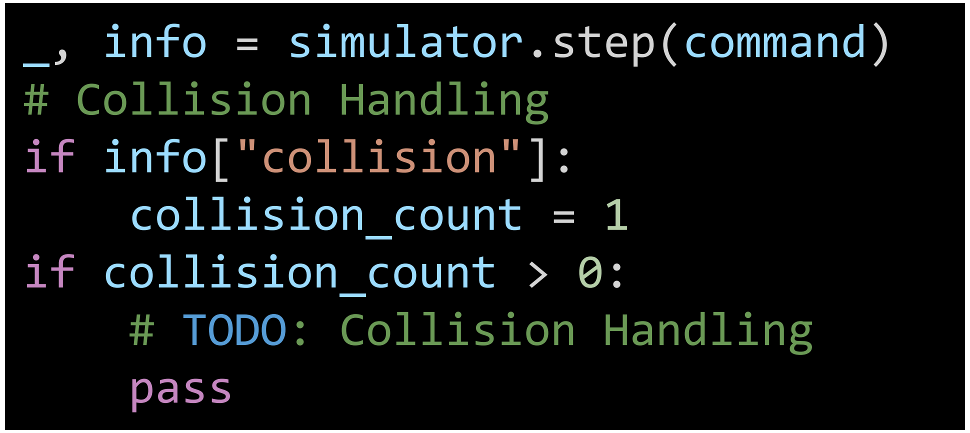
- Complete the "TODO" for each file.



```
if path is not None and collision_count == 0:
    # TODO: Planning and Controlling
    if args.simulator == "basic":
        next_v = 0
        next_w = 0
        command = ControlState("basic", next_v, next_w)
    elif args.simulator == "diff_drive":
        next_lw = 0
        next_rw = 0
        command = ControlState("diff_drive", next_lw, next_rw)
    elif args.simulator == "bicycle":
        next_a = 0
        next_delta = 0
        command = ControlState("bicycle", next_a, next_delta)
    else:
        exit()
```

Collision Handling

- Handle the situation of collision.
- Hint: move backward and re-planning the path.



```python
_, info = simulator.step(command)
# Collision Handling
if info["collision"]:
    collision_count = 1
if collision_count > 0:
    # TODO: Collision Handling
    pass
```

Run Code

```
python navigation.py -s [basic/diff_drive/bicycle] (simulator)
            -c [pid/pure_pursuit/stanley/lqr] (controller)
            -p [a_star/rrt/rrt_star] (planner)
                      -m [map path] (Just use the default)
```