PA3 Report

Padakanti Akshay Kiran, 200050093

Task 1:

- I followed the straight path connecting the car and the road (350,0)
- This method was my first approach and only approach, and it worked well
- In every iteration of the next_action function, I did the following:
  - First I figured out the angle needed to reach the point (350,0) using the following formula:
    angle_needed=math.atan2((-y),(350-x))*180/math.pi
    angle_needed += 360 if it is negative
  - Then I checked whether the angle of the car was close to the angle_needed, with threshold of 3 degrees by checking if the diff was less than 3 or more than 357 (because 359 is close to 1, but the diff is large)
  - If the angle was not close enough, I then figured out which way to turn the car (clockwise or anti-clockwise) to reach the angle faster by choosing clockwise if angle - angle_needed is > 180 or between -180 and 0.
  - I also applied brakes to the car if the angle was wrong (acc = -5)
  - If the angle was close enough, I accelerated the car forward
- Slight Inconsistency in given code: The angle given in the very first iteration is in the range (-180, 180) but it is in the (0,360) range for every other iteration. So I added 360 to the angle if it was negative

Task 2:

- I followed only horizontal or vertical paths to reach the destination (another approach explained later)
- The axes are safe because the pits are centered at least 110 distance from both axes, implying that the edges of the pits are at least 60 distance (110-50) from the axes. The pits are initialized in separate quadrants.
- Path:
  - First, if the car was away from the x-axis, I tried to reach the x-axis by moving vertically if possible.
  - If there was a pit blocking car's path to the x-axis, I moved the car horizontally right until it was no longer blocked by any pit and then followed the same vertical path to reach the x-axis

- ➢ After reaching the x-axis, I went horizontally right to reach (350,0)

❖ I made a variable called cen_list in the init() function and updated it to ran_cen_list in the controller before calling next_action

❖ To check whether a pit is blocking the car's path to the x-axis, I iterated through the cen_list and checked if any of the pits had its center within (x-70, x+70) and its y coordinate between the car and x-axis. (70 was chosen to be safe from the variances in steering angle)

❖ Moving horizontally or vertically:
- ➢ The approach is entirely the same as the task1.
- ➢ Angle_needed:
  - ■ 270 if y > 0 and 90 if y < 0 (vertical movement)
  - ■ 0 (horizontal movement)
- ➢ If the angle is not within the threshold, figure out which direction (clockwise or anti-clockwise) helps you reach the angle faster and steer in that direction while applying brakes (acc = -5)
- ➢ If the angle is correct, accelerate (acc = 5) without steering
- ➢ Finally, if the target was to reach any of the axes, I decelerated as soon as reaching within 30 units of the axis

❖ I tried the following approach first but failed to reach the end safely consistently:
- ➢ I first checked points on the x-axis (to the right of the car) that could be reached without intersecting any pit.
- ➢ Then I followed a straight right path after reaching that point
- ➢ If no point was reachable, I tried to reach the y-axis first, then the center, then straight right towards the road. Both axes are safe as explained above
- ➢ Problems I faced with this approach:
  - ■ Hard to check whether the line segment joining the car to the target point intersects any of the pits (My Code was hard to debug)
  - ■ The variance in the angle caused the car to hit the pits even if the path was safe
  - ■ Hard to decide when to start decelerating as the path is now slanted
  - ■ If I applied brakes too quickly, the car would not reach the target and would overshoot otherwise