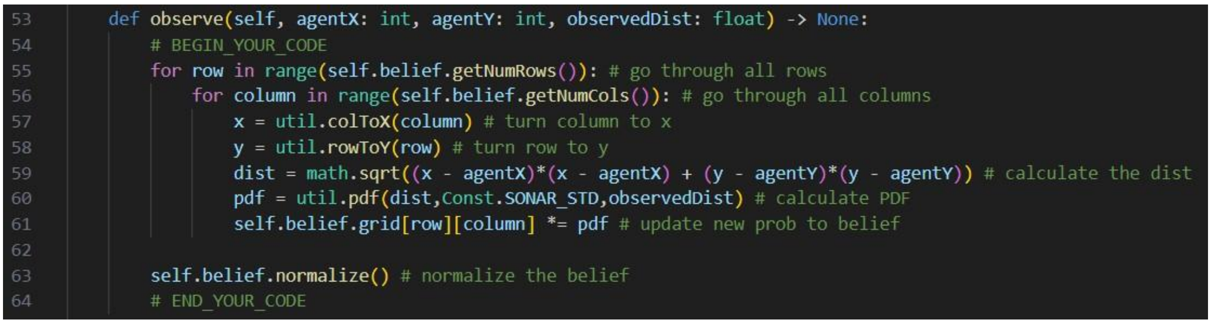
Homework 5: Car Tracking

110550168

Part I. Implementation (15%) :

Please screenshot your code snippets of Part 1 ~ Part 3, and explain your implementation.

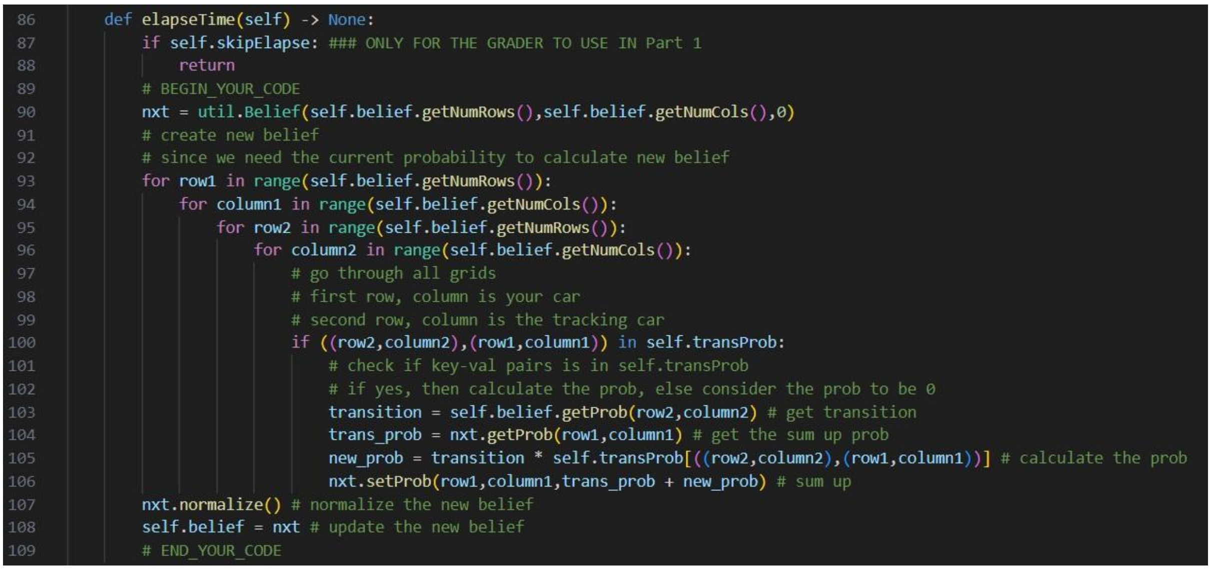
Part 1:



```python
def observe(self, agentX: int, agentY: int, observedDist: float) -> None:
    # BEGIN_YOUR_CODE
    for row in range(self.belief.getNumRows()): # go through all rows
        for column in range(self.belief.getNumCols()): # go through all columns
            x = util.colToX(column) # turn column to x
            y = util.rowToY(row) # turn row to y
            dist = math.sqrt((x - agentX)*(x - agentX) + (y - agentY)*(y - agentY)) # calculate the dist
            pdf = util.pdf(dist,Const.SONAR_STD,observedDist) # calculate PDF
            self.belief.grid[row][column] *= pdf # update new prob to belief

    self.belief.normalize() # normalize the belief
    # END_YOUR_CODE
```

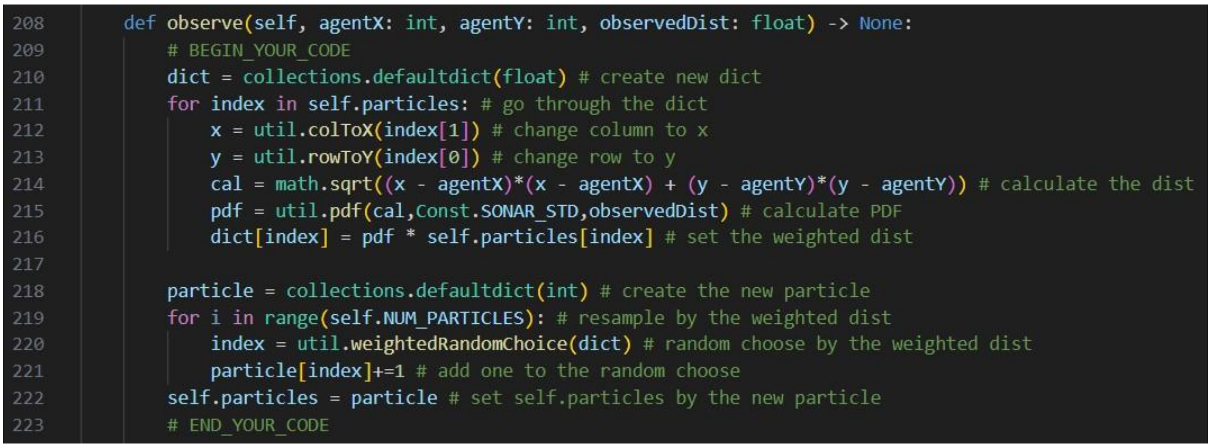
Part 2:



```python
def elapseTime(self) -> None:
    if self.skipElapse: ### ONLY FOR THE GRADER TO USE IN Part 1
        return
    # BEGIN_YOUR_CODE
    nxt = util.Belief(self.belief.getNumRows(),self.belief.getNumCols(),0)
    # create new belief
    # since we need the current probability to calculate new belief
    for row1 in range(self.belief.getNumRows()):
        for column1 in range(self.belief.getNumCols()):
            for row2 in range(self.belief.getNumRows()):
                for column2 in range(self.belief.getNumCols()):
                    # go through all grids
                    # first row, column is your car
                    # second row, column is the tracking car
                    if ((row2,column2),(row1,column1)) in self.transProb:
                        # check if key-val pairs is in self.transProb
                        # if yes, then calculate the prob, else consider the prob to be 0
                        transition = self.belief.getProb(row2,column2) # get transition
                        trans_prob = nxt.getProb(row1,column1) # get the sum up prob
                        new_prob = transition * self.transProb[((row2,column2),(row1,column1))] # calculate the prob
                        nxt.setProb(row1,column1,trans_prob + new_prob) # sum up
    nxt.normalize() # normalize the new belief
    self.belief = nxt # update the new belief
    # END_YOUR_CODE
```

Part 3:

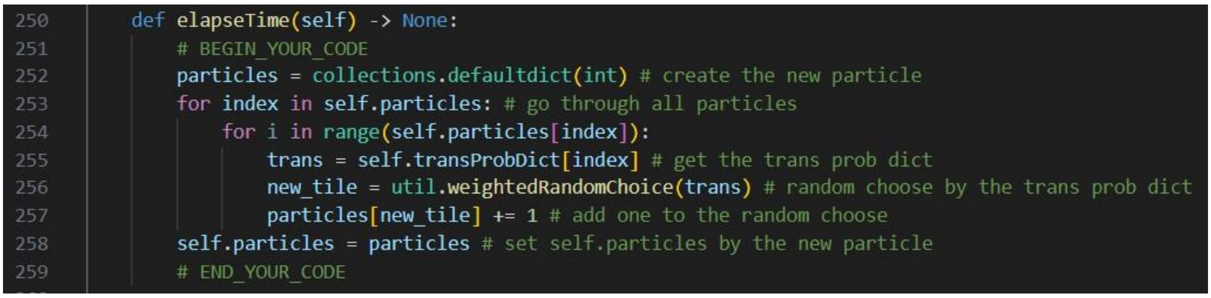
observe :



```
    def observe(self, agentX: int, agentY: int, observedDist: float) -> None:
        # BEGIN_YOUR_CODE
        dict = collections.defaultdict(float) # create new dict
        for index in self.particles: # go through the dict
            x = util.colToX(index[1]) # change column to x
            y = util.rowToY(index[0]) # change row to y
            cal = math.sqrt((x - agentX)*(x - agentX) + (y - agentY)*(y - agentY)) # calculate the dist
            pdf = util.pdf(cal,Const.SONAR_STD,observedDist) # calculate PDF
            dict[index] = pdf * self.particles[index] # set the weighted dist

        particle = collections.defaultdict(int) # create the new particle
        for i in range(self.NUM_PARTICLES): # resample by the weighted dist
            index = util.weightedRandomChoice(dict) # random choose by the weighted dist
            particle[index]+=1 # add one to the random choose
        self.particles = particle # set self.particles by the new particle
        # END_YOUR_CODE
```

elapseTime :



```
    def elapseTime(self) -> None:
        # BEGIN_YOUR_CODE
        particles = collections.defaultdict(int) # create the new particle
        for index in self.particles: # go through all particles
            for i in range(self.particles[index]):
                trans = self.transProbDict[index] # get the trans prob dict
                new_tile = util.weightedRandomChoice(trans) # random choose by the trans prob dict
                particles[new_tile] += 1 # add one to the random choose
        self.particles = particles # set self.particles by the new particle
        # END_YOUR_CODE
```

Part II. Question answering (5%) :

Please describe problems you met and how you solved them.

1. I misunderstood part 2's meaning and used self.belief to store the new belief.
   sol : I read through the instructions above then figure out my mistake
2. For part 3, I forgot to resample, so I got the wrong answer.
   sol : I ask my friends and also make clear what I should do.