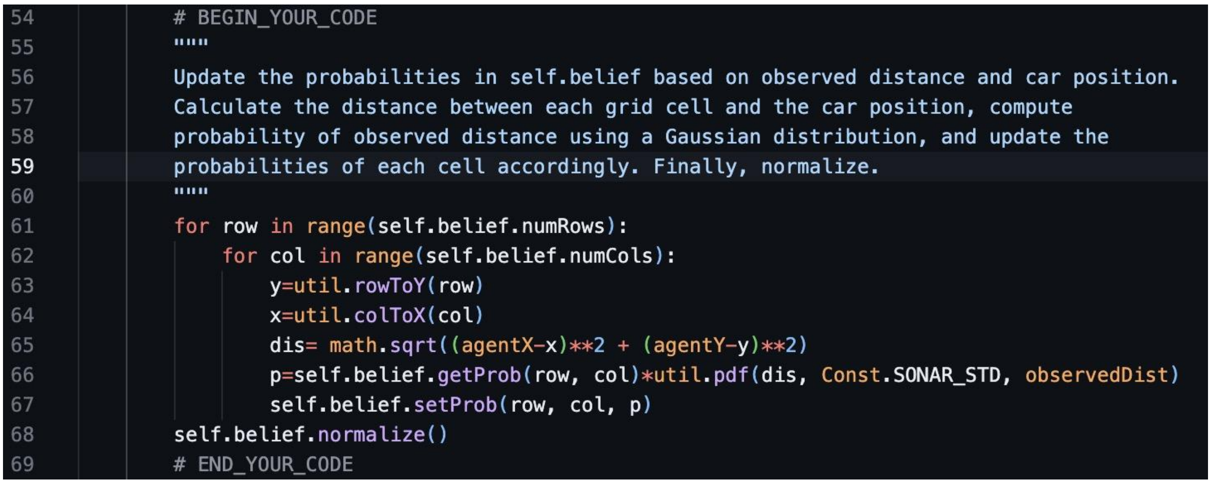
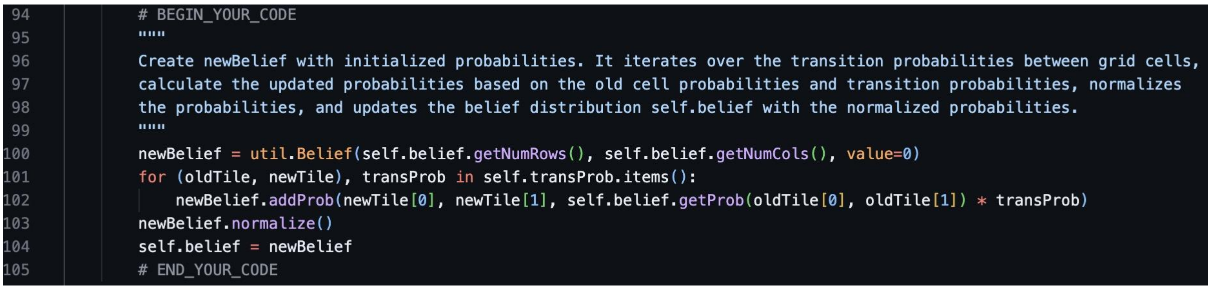
Homework 5: Car Tracking

110550012　黃鵬軒

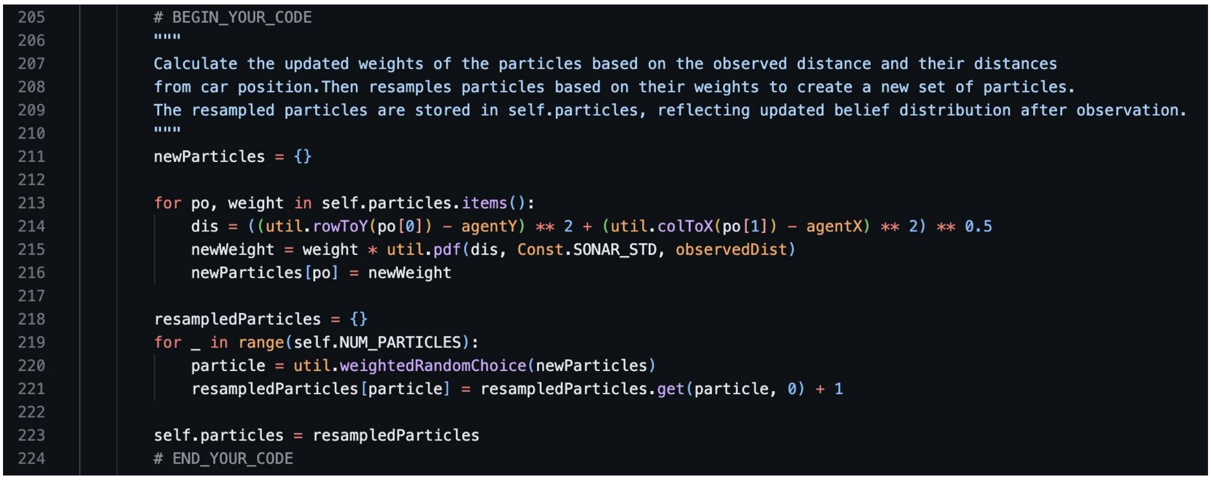
Part I. Implementation (15%):



```python
# BEGIN_YOUR_CODE
"""
Update the probabilities in self.belief based on observed distance and car position.
Calculate the distance between each grid cell and the car position, compute
probability of observed distance using a Gaussian distribution, and update the
probabilities of each cell accordingly. Finally, normalize.
"""
for row in range(self.belief.numRows):
    for col in range(self.belief.numCols):
        y=util.rowToY(row)
        x=util.colToX(col)
        dis= math.sqrt((agentX-x)**2 + (agentY-y)**2)
        p=self.belief.getProb(row, col)*util.pdf(dis, Const.SONAR_STD, observedDist)
        self.belief.setProb(row, col, p)
self.belief.normalize()
# END_YOUR_CODE
```



```python
# BEGIN_YOUR_CODE
"""
Create newBelief with initialized probabilities. It iterates over the transition probabilities between grid cells,
calculate the updated probabilities based on the old cell probabilities and transition probabilities, normalizes
the probabilities, and updates the belief distribution self.belief with the normalized probabilities.
"""
newBelief = util.Belief(self.belief.getNumRows(), self.belief.getNumCols(), value=0)
for (oldTile, newTile), transProb in self.transProb.items():
    newBelief.addProb(newTile[0], newTile[1], self.belief.getProb(oldTile[0], oldTile[1]) * transProb)
newBelief.normalize()
self.belief = newBelief
# END_YOUR_CODE
```

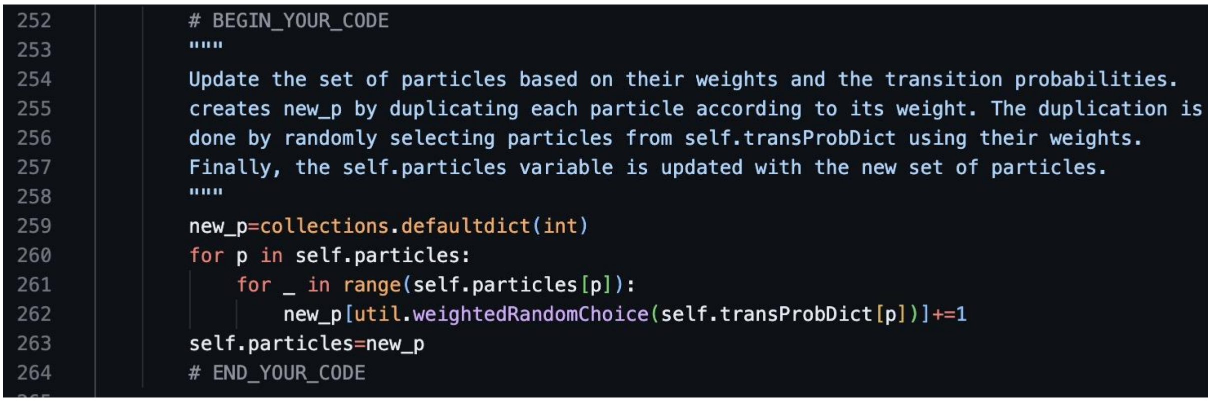


```python
# BEGIN_YOUR_CODE
"""
Calculate the updated weights of the particles based on the observed distance and their distances
from car position.Then resamples particles based on their weights to create a new set of particles.
The resampled particles are stored in self.particles, reflecting updated belief distribution after observation.
"""
newParticles = {}

for po, weight in self.particles.items():
    dis = ((util.rowToY(po[0]) - agentY) ** 2 + (util.colToX(po[1]) - agentX) ** 2) ** 0.5
    newWeight = weight * util.pdf(dis, Const.SONAR_STD, observedDist)
    newParticles[po] = newWeight

resampledParticles = {}
for _ in range(self.NUM_PARTICLES):
    particle = util.weightedRandomChoice(newParticles)
    resampledParticles[particle] = resampledParticles.get(particle, 0) + 1

self.particles = resampledParticles
# END_YOUR_CODE
```



```
        # BEGIN_YOUR_CODE
        """
        Update the set of particles based on their weights and the transition probabilities.
        creates new_p by duplicating each particle according to its weight. The duplication is
        done by randomly selecting particles from self.transProbDict using their weights.
        Finally, the self.particles variable is updated with the new set of particles.
        """
        new_p=collections.defaultdict(int)
        for p in self.particles:
            for _ in range(self.particles[p]):
                new_p[util.weightedRandomChoice(self.transProbDict[p])]+=1
        self.particles=new_p
        # END_YOUR_CODE
```

Part II. Question answering (5%):

When i run drive.py, terminal show following error:
DEPRECATION WARNING: The system version of Tk is deprecated and may be removed in a future release. Please don't rely on it. Set TK_SILENCE_DEPRECATION=1 to suppress this warning.

I tried some solution I found on the internet but it did not work and then I guessed it is a version problem again, so I changed the python version from 3.9.6 to 3.7.9, it solved my problem.