Homework 5: Car Tracking

110550080 何曉嫻

Part I. Implementation (15%):

Part 1

```
54          # BEGIN_YOUR_CODE
55          for row in range(self.belief.getNumRows()):
56              for col in range(self.belief.getNumCols()):
57                  # Calculate the distance between the tile and my car
58                  x= util.colToX(col)
59                  y= util.rowToY(row)
60                  dis= math.hypot(agentX- x, agentY- y)
61                  # p(et|ht)
62                  p= util.pdf(dis, Const.SONAR_STD, observedDist)
63                  # Update the current posterior probability with following :
64                  # P(Ht|E1=e1~et)= p(et|ht)* P(Ht|E1=e1~et-1)
65                  post= p*self.belief.getProb(row, col)
66                  self.belief.setProb(row, col, post)
67          # Normalize the posterior probability
68          self.belief.normalize()
69          # END_YOUR_CODE
```

Part 2

```
94          # BEGIN_YOUR_CODE
95
96          # Create a new belief to make sure we use
97          # the CURRENT self.belief distribution to compute updated beliefs
98          new= util.Belief(self.belief.getNumRows(), self.belief.getNumCols(), 0)
99          # Calaulate the sum of probability
100         for (oldTile, newTile) in self.transProb:
101             # P(Ht=ht|E1=e1~et) * p(h_t+1|ht)
102             old= self.belief.getProb(oldTile[0], oldTile[1])
103             tp= self.transProb[(oldTile, newTile)]
104             post= old* tp
105             new.addProb(newTile[0], newTile[1], post)
106         # Normalize the posterior probability
107         new.normalize()
108         # Update the belief
109         self.belief = new
110
111         # END_YOUR_CODE
```

Part 3-1

```
211         # BEGIN_YOUR_CODE
212
213         # Reweight the particle distribution with emission probability
214         # associated with the observed distance.
215         for (row, col) in self.particles:
216             x= util.colToX(col)
217             y= util.rowToY(row)
218             dis= math.hypot(agentX-x, agentY-y)
219             p= util.pdf(dis, Const.SONAR_STD, observedDist)
220             post= self.particles[(row, col)]* p
221             self.particles[(row, col)]= post
222
223         # create a new dictionary
224         new= collections.defaultdict(int)
225         # Re-sample the particles from the reweighted distribution
226         for i in range(self.NUM_PARTICLES):
227             particle= util.weightedRandomChoice(self.particles)
228             new[particle]+= 1
229         self.particles= new
230         # END_YOUR_CODE
```

Part 3-2

```
258         # BEGIN_YOUR_CODE
259         # create a new dictionary
260         new= collections.defaultdict(int)
261         # Update the particles with transition probability in each tile
262         # Get the particle locations at time $t+1$
263         for cur in self.particles:
264             for i in range(self.particles[cur]):
265                 # Choose one particle from taht locations
266                 particle= util.weightedRandomChoice(self.transProbDict[cur])
267                 new[particle]+= 1
268         self.particles= new
269         # END_YOUR_CODE
```

Part II. Question answering (5%):

I don’t know how to resample in Part 3. After realizing the meaning of resample, I know that I should choose the sample randomly (based on distribution proportional to the weights) and calculate the times it is chosen.