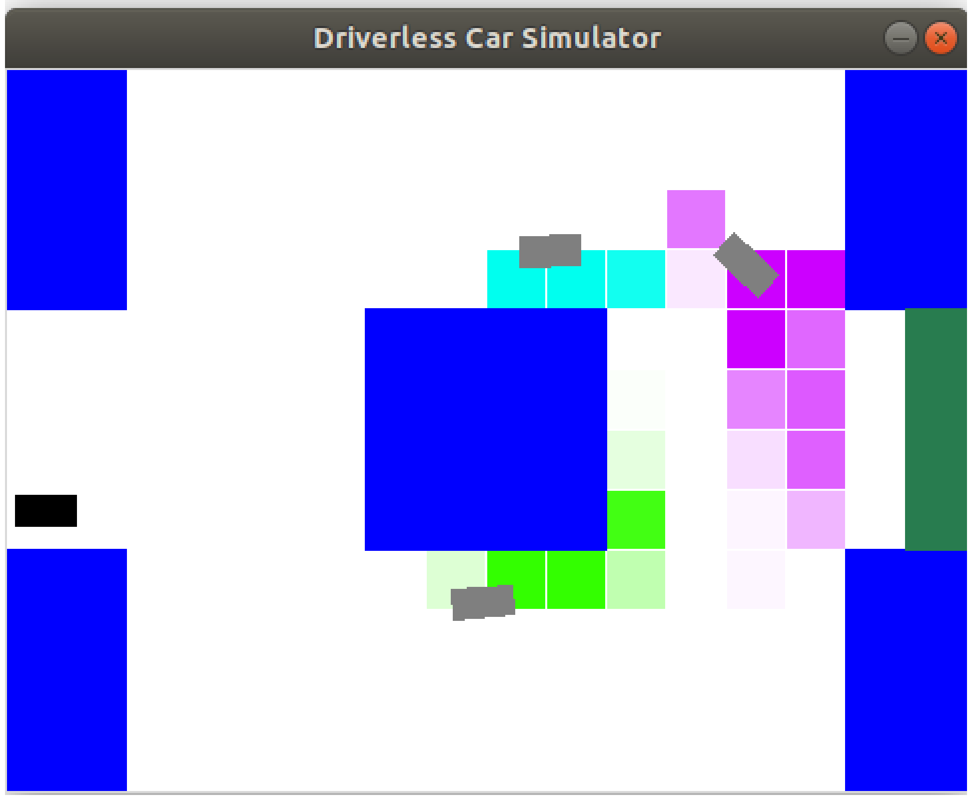
Introduction to Artificial Intelligence

Homework 4: Car Tracking

TA 周千貿

Introduction

- Building an autonomous driving system is an incredibly complex endeavor.
- In this assignment, you will focus on the sensing system, which allows us to track other cars based on noisy sensor readings.



Getting started

- This code base is a modified version of the Driverless Car written by Chris Piech at Stanford University.(https://stanford-cs221.github.io/winter2021/assignments/car/index.html)
- You can only execute it on a local machine.
- Google Colab cannot execute it because it has GUI.
- Please install python 3 on your own machine and be familiar with run code with CLI.

Getting started

- Let's start by trying to drive manually, run the following command: `python drive.py -l lombard -i none`
- Other arguments for `python drive.py` will describe in the homework document.
- The drive.py file is not used for any grading purposes, it's just there to visualize the code you will be writing and help you gain an appreciation for how different approaches result in different behaviors (and to have fun!).

Autogrding

- TAs will use an grader to grade your implementation.
- This grader is different from homework 3
- What autograder will run on your implementation?
  - Just check your output. You can see the test cases in grader.py.
- After tests, It will show the score you will get.

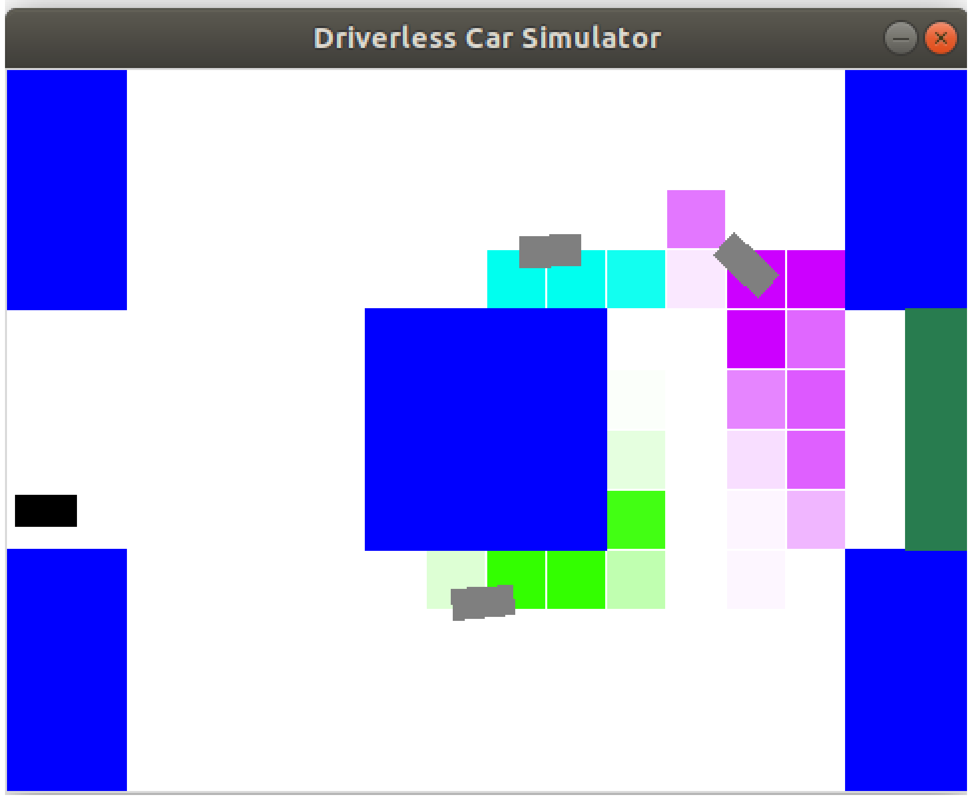
```
========== END GRADING [80/80 points + 0/0 extra credit]
```

Autogrding

- The autograder has been included in the code base. You can use the following command to test by yourself:
  ```
  python grader.py
  ```
- To run a single test (e.g., part1-1), run the following command:
  ```
  python grader.py part1-1
  ```

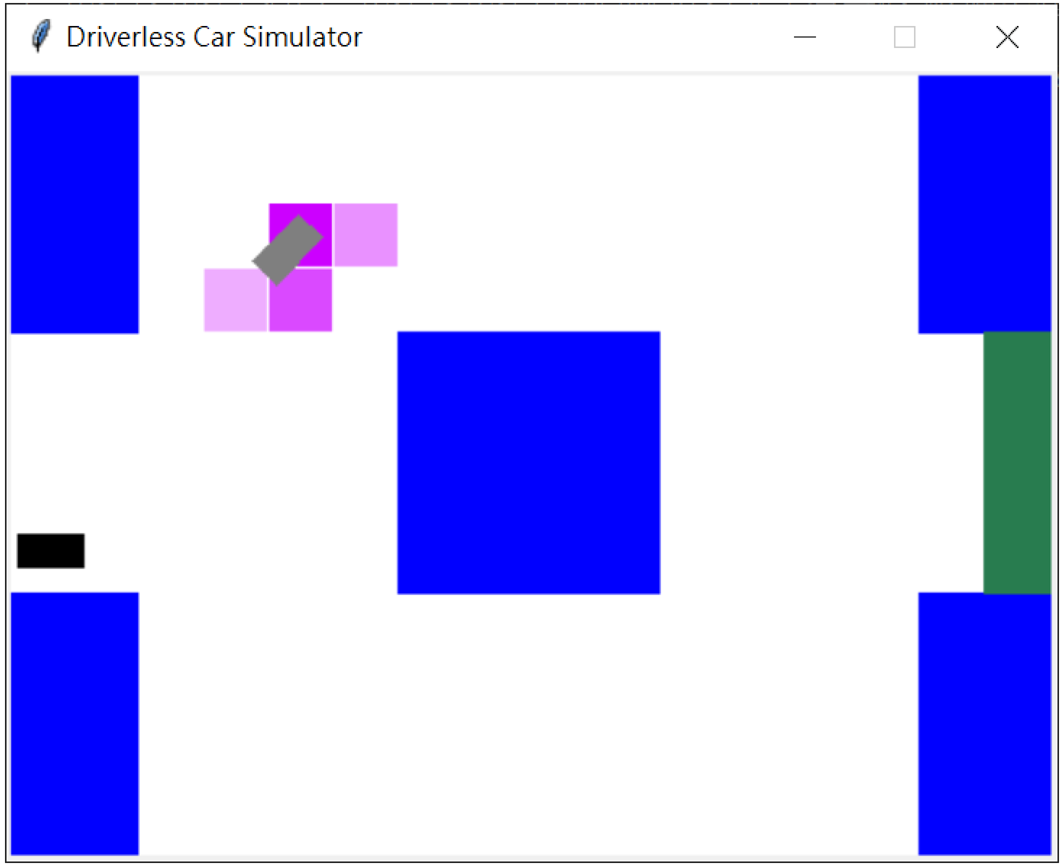
Modeling car locations

- The world is a two-dimensional rectangular grid.
- There are your car and K other cars.
- At each time step t, your car gets a noisy estimate of the distance to each of the cars.
- We assume that each of the other cars move independently and that the noise in sensor readings for each car is also independent.



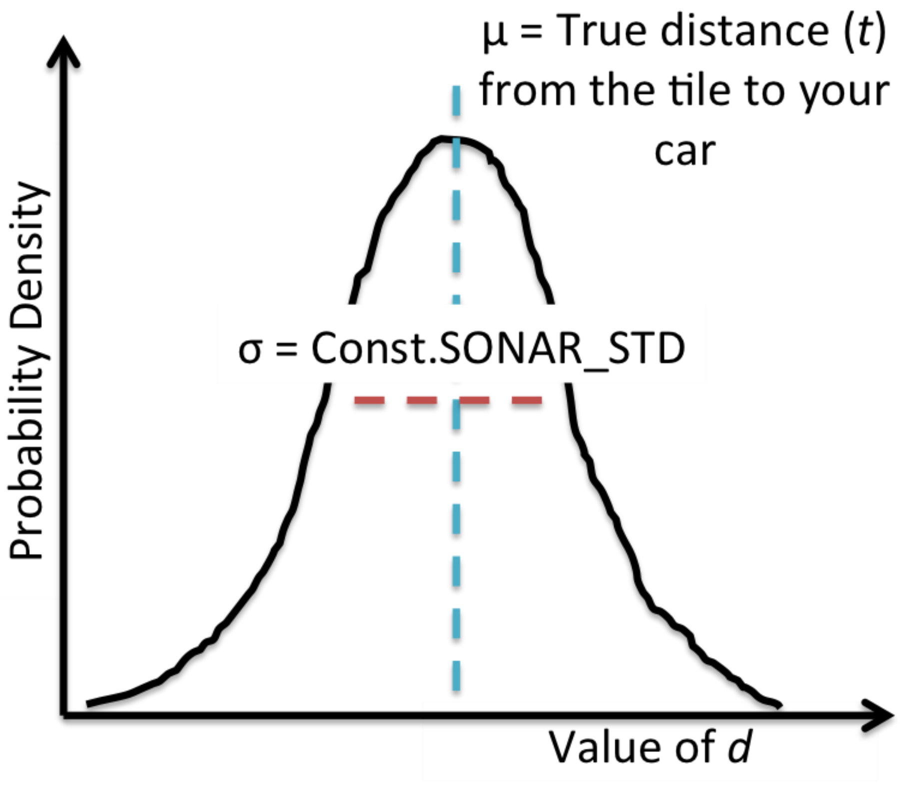
Modeling car locations

- We will assume there is just one other car.
- $H_t \in R^2$ is the actual position of the other car.
- $p(h_t | h_{t-1})$ governs the car's movement.
- $a_t \in R^2$ is your car's position.
- To minimize costs, we use a simple sensing system based on a microphone. The microphone provides us with distance $E_t \sim N(\|a_t - H_t\|_2, \sigma^2)$



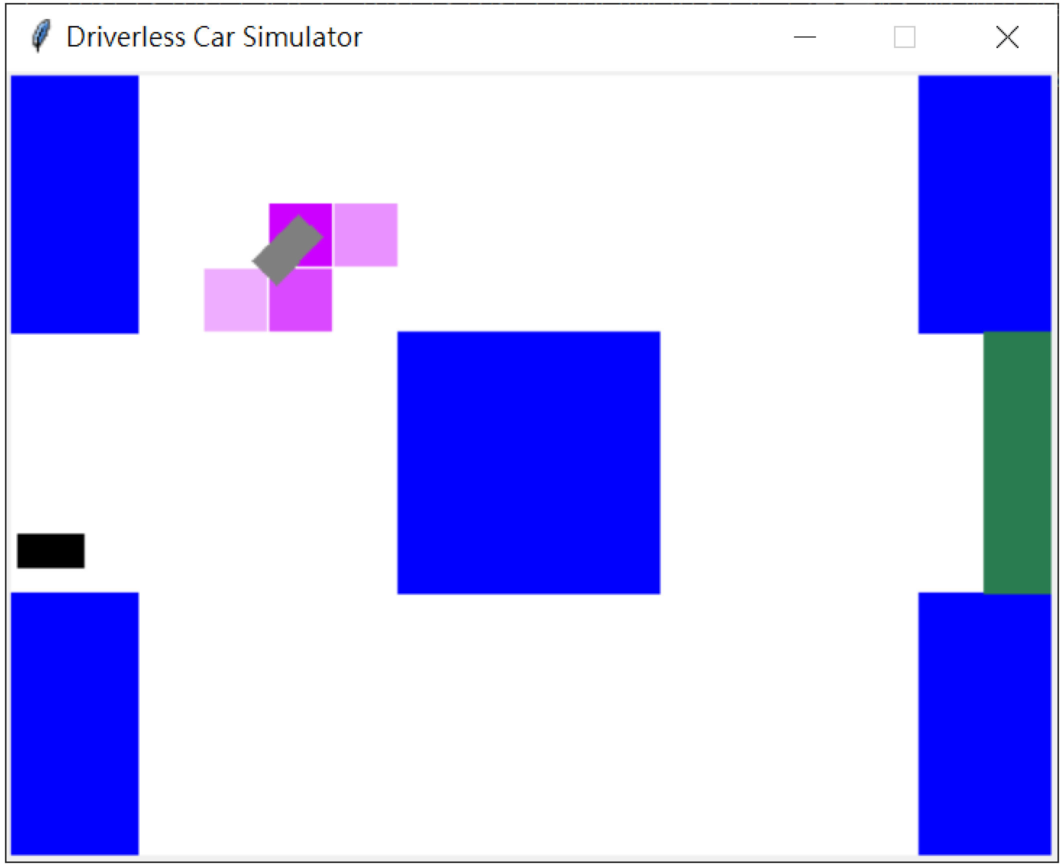
Modeling car locations

- The microphone provides us with distance $E_t \sim N(\|a_t - H_t\|_2, \sigma^2)$
- For example, if your car is at $a_t = (1,3)$ and the other car is at $H_t = (4,7)$, then the actual distance is 5 and E_t might be 4.6 or 5.2, etc.
- When you coding, please use Const.SONAR_STD and util.pdf(mean, std, value) to calculate the probability.



Modeling car locations

- Your job is to implement a car tracker that computes the posterior distribution $P(H_t|E_1 = e_1, \cdots, E_t = e_t)$ and update it for each time step t.
- To simplify things, we will discretize the world into tiles represented by (row, col) pairs.
- To convert from a tile to a location, use util.rowToY(row) and util.colToX(col).



Requirements

- Please modify the codes in submission.py between # Begin your code and # End your code.
- In addition, do not import other packages.
- In part 1 and 2, you will implement ExactInference, which computes a full probability distribution of another car's location over tiles (row, col).
- In part 3, you will implement ParticleFilter, which works with particle-based representation of this same distribution.
- This assignment will be one of the most conceptually challenging assignments. Please start early!

Part 1: Emission probabilities (20%)

- In this part, we assume that the other car is stationary (e.g., $H_t = H_{t-1}, \forall\, t$).
- You will implement a function observe that upon observing a new distance measurement $E_t = e_t$ updates the current posterior probability from $P(H_t | E_1 = e_1, \cdots, E_{t-1} = e_{t-1})$ to $P(H_t | E_1 = e_1, \cdots, E_t = e_t) \propto P(H_t | E_1 = e_1, \cdots, E_{t-1} = e_{t-1}) p(e_t | h_t)$.
- Emission probabilities $p(e_t | h_t)$ described earlier under "Modeling car locations".
- The current posterior probability $P(H_t | E_1 = e_1, \cdots, E_{t-1} = e_{t-1})$ is stored as self.belief in ExactInference.

Part 2: Transition probabilities (20%)

- Now, let's consider the case where the other car is moving according to transition probabilities $p(h_{t+1}|h_t)$.
- We have provided the transition probabilities for you in self.transProb.
- You will implement a function elapseTime that updates the posterior probability about the location of the car at a current time step $P(H_t = h_t|E_1 = e_1, \cdots, E_t = e_t)$ to the next time step

$$P(H_{t+1} = h_{t+1}|E_1 = e_1, \cdots, E_t = e_t) \propto \sum_{H_t} P(H_t = h_t|E_1 = e_1, \cdots, E_t = e_t) p(h_{t+1}|h_t)$$

Part 3: Particle filtering (40%)

- Though exact inference works well for the small maps, it wastes a lot of effort computing probabilities for every available tile. We can solve this problem using a particle filter.
- In this part, you will implement observe and elapseTime functions for the ParticleFilter class in submission.py.
- Some of the code has been provided for you. For example, the particles have already been initialized randomly.

Report (20%)

- A written report is required.
- The report should be written in English.
- Save the report as a .pdf file.
- For part 1 ~ 3, please take some screenshots of your code and explain how you implement codes **in detail**.
- Describe problems you meet and how you solve them.

Important Rules

- **Due Date: 2021/6/1 23:55**

- **Submission**
  - Please prepare your submission.py and report (.pdf) into STUDENTID_hw4.zip.

- **Late Submission Policy**
  - 20% off per late day

Reminders

- More detail will be in the homework document.
- Read through the homework document before you get started. Make sure you understand what to do.
- If there are any updates or problems of the homework, we will announce on E3.
- If you have any questions for homework please mail me.
  - TA 周千貿: ya11235813@gmail.com
  - TA 胡瑞麟: linhu.cs09g@nctu.edu.tw
  - TA 張桂華: amy09921@gmail.com