Tutorial 1

Name: Zhaokun Su

Breakout group: 3

Zid: z5235878

Discussion results for group during tutorial time:

Percepts:
location of aircraft
Weather information
Runway conditions
Enemy or birds

Actions:
Allow landing
Allow takeoff
Redirect course

Goal:
Safely land and operate the aircraft

Environment:
airport
Airspace

Fully observable
Stochastic
Sequentia

(i) This agent is more complex than a reactive agent that merely acts on the current percepts on each cycle. In what environments does this agent have an advantage over the reactive agent? Run some experiments to test your ideas.

The agent in this exercise is more complicated than a reactive agent. It is more intelligent because this agent is goal-oriented, the agent has a horizon to find some food within a distance rather than just follow the complete condition-action rules to choose its next move. It is goal based. Besides that, it can be shown that this agent can record its behaviors and this is another reason why this agent is more complex.

As we can see, if we change the horizon to 100 (as if the environment is fully-observable), the result score will be better:

```
class Agent():
    # Agent maintains set of food items seen...but does not update it very well

    horizon = 100  # the agent can only see food up to this distance
```

```
Agent performs eat at (3, 1)
Agent performs move_right to (4, 1)
Agent performs move_right to (5, 1)
Food appears at (6, 2)
Agent performs move_right to (6, 1)
Agent score was 5
```

And if we change the horizon to 0 (as if no goals to discover), the result score will be so bad:

```
Agent performs move_right to (9, 2)
Food appears at (5, 1)
Agent performs move_right to (9, 2)
Agent performs move_right to (9, 2)
Agent performs move_right to (9, 2)
Agent performs move_right to (9, 2)
Food disappears at (3, 3)
Agent performs move_right to (9, 2)
Agent performs move_right to (9, 2)
Agent score was 0
```

Eventually, we can conclude that this agent has a higher level compared with a reactive agent, no matter in which environment, this agent will perform better.

And when the horizon distance increase, foods are more easily to be discovered, so this agent has an advantage over the reactive agent.

(ii) This agent stores all previous percepts in a set but does not update that set when food is observed to disappear (food that should be possible to see is no longer visible). Change the code to implement this functionality.

We need to add self.food.remove() at the end of the else block so that delete coresponding food in our food set.

```
    def clock_tick(self):
        # possibly delete food items -- copy elements not deleted into new
        food_set = set()
        for location in self.food:
            r = random.random()
            if (r > self.food_del_chance):
                food_set.add(location)
            else:  # when r < del_chance, food need to be deleted
                print("Food disappears at", end=" ")
                print(location)
                self.food.remove(location)
        self.food = food_set
        # possibly add new food item in random location
        r = random.random()
        if (r < self.food_add_chance):
            X = random.randint(0, self.grid_end)
            Y = random.randint(0, self.grid_end)
            if not ((X, Y) in self.food):
                self.food.add((X, Y))
                print("Food appears at", end=" ")
                print((X, Y))
```

(iii) (Harder) Consider how to define the agent action selection function when there are multiple agents in the world, all competing for food. Code the perception and action selection functions to handle this scenario and run some experiments. Note: It is straightforward using the existing code to create any number of agents.

It mainly logic can be like this:

There are multiple agents participating in this competition, and they both have the same horizon. Since the food is randomly appearing or disappearing, so:

a. When this food has been reached firstly by agent A, score A + 1, and this food disappears in the environment, a new food appears in anther location somewhere in this environment.
b. If two agents or more than two agents reach the same food at the same time. All of those food score + 1 and this food disappears in the environment, a new food appears in anther location somewhere in this environment.
c. It makes sense that different agents can have different horizons.