Semester 2 2014
COMP3702/7702 ARTIFICIAL INTELLIGENCE
ASSIGNMENT 2

Note:

- You can do this assignment individually or in a group of 2-3 students.
- For those who choose to work in a group:
  - All students in the group must be enrolled in the same course code, i.e., all COMP3702 students or all COMP7702 students.
  - Send the group name and members (name and student ID) via email to comp3702-staff@itee.uq.edu.au with subject **group for Assignment2** before **11.59pm on Friday, September, 26th**. If we do not receive your e-mail by then, you will need to work on the assignment individually.
- The due date for the code is **Wednesday, October, 8th**, at demo time.
- You need to choose a time slot to demo your program. Stay tuned for the exact time and venue.
- For the demo, you can use your own laptop or one of the desktops in the lab.
- Please submit a hardcopy format of your report before **midday on Monday, October, 13th** via the assignment chute at Hawken. You need to use the assignment cover sheet (https://student.eait.uq.edu.au/coversheets/?pid=67354&sid=52122&aiid=193681 for COMP3702 and https://student.eait.uq.edu.au/coversheets/?pid=66019&sid=50787&aiid=193720 for COMP7702). Each group only needs to submit one report.

Clarifications 25/09:

- The number of steps your cycle(s) can take in a race is at most 100n, where n is the length of the track. This requirement is to avoid having a race with infinite number of steps.
- The goal cells will always be blank.
- In the event of tie, you will win. We're nice ☺
- **Hint:** Hardcoded policy is unlikely to get you a win in cases more difficult than Novice.

Clarifications 23/09:

- If you compute an initial policy offline and revise it online, then the time limit for the offline part of your computation follows the offline requirement and the time limit during online compuation follows the online requirment, as set in Update 22/09.
- The total money you spend to buy autonomous cycles and register on races can **NOT** be more than the initial capital. However, the total money you have at the end of Tour de UQ may be negative, due to the repairs you need to do.
- For NE and SE actions: If the cycle ends up in a position outside the grid cells, then it will remain where it is.
- The cycle will get the reward for winning (if it wins) when it enters a goal cell.
- About #11, par. 2 in "Rules of Tour de UQ". Essentially, it means the penalty is given when a cycle and a distractor are in the same cell at the same time

step, i.e., R(s = (cycle in cell-i, distractor in cell-i), a = any action, s' = any state) = penalty as described in this Assignment.

Update 22/09:

- Your code must choose the cycle(s) and track(s) in less than 5 minutes. If you compute the solution offline, you need to compute the strategy for each race in less than 1.5 minutes for novice and medium, and less than 3 minutes for advance (including advance + >1 cycles per race). If you compute the solution online, you need to compute the strategy in less than less than 1 sec per step for all.
- For advance + >1 cycles per race, each cycle has two additional actions, i.e.,
  - TO: Turns a cycle into an obstacle with probability 0.7 (with probability 0.3 it remains a cycle). While a cycle becomes an obstacle, it cannot move.
  - TC: Turns a cycle that has been turned into an obstacle back to become a cycle with probability 1.0.

Congratulations!!! You have been hired as a strategic consultant to help the ITEE cycling team win Tour de UQ. Tour de UQ is a week-long autonomous cycle competition between various schools at UQ. It has many races at various tracks, from a few meter in-campus tracks to tens of kilometers inter-campus tracks. Each team can choose which track(s) they want to compete. Of course, different race has different registration fee, provides different challenges and different prizes. Each team is provided with an initial capital they can spend on bicycle(s) and registration fees. The goal of the competition is to accumulate as much profit as possible. Your task is to design and develop a software that will automatically compute the strategies on what bicycle(s) to buy, which race the team should compete in, and of course how the autonomous cycle should move so as to win the race.

For this assignment, you can assume:

1. Each track is rectangular in shape and is represented as uniform grid cells.
2. The world is fully observable, but non-deterministic.
3. The possible actions of are:
   - FS: Move East at the rate of 1 cell/step.
   - FM: Move East at the rate of 2 cells/step.
   - FF: Move East at the rate of 3 cells/step.
   - NE: Move North East at the rate of 1 cell/step.
   - SE: Move South East of the current cell at the rate of 1 cell/step.
   - ST: Stay.
4. There are no error in the opponents' movement, but their action follows a stochastic policy.
5. Within a race, the autonomous cycle may be damaged. We assume that these damages can be fixed instantaneously and do not affect the performance of the cycle at the next time step or at the next race. However, it will reduce the amount of money the team has because the team needs to pay for the repair.

The rules of Tour de UQ are:

1. Your team will be given a one off initial capital of $x. You win a race at a particular track when you reach one of the goal cells faster than your opponents. If there are no opponents, you win the race when you reach one of the goal cells in 2n steps, where n is the #columns in the track. Each cell at the right-most side of the track is a goal cell. At this cell, regardless of what action is performed, the cycle will move to a terminal state.
2. The team must choose the tracks they will compete in and pay the registration fees at the start of Tour de UQ.
3. A team can only buy the autonomous cycle(s) from the provided list.
4. The team can buy more than one autonomous cycle, provided it has enough money to do so.
5. The team can only buy the cycles at the beginning of Tour de UQ.
6. The team can use different autonomous cycle for different track.
7. The team can register more than one cycles to participate in the same race. Of course, the registration fee must be paid for each autonomous cycle that participates in the race.
8. A team can participate in at most three races.
9. A team can choose a particular track at most once.
10. A track may contain static obstacles.
11. A track may be occupied by one or more distractors. Each distractor is associated with a cell, where the distractor may appear or may not appear. You will be given information on the probability that the distractor will appear.
    You can assume that the appearance of each distractor within a time step does not change. Hence if at the beginning of the step a distractor appears at cell-i, it will remain to appear there during this step. The state of the distractor only changes at the end of the step. This assumption ensures the state is always fully observable.

You will be given the following information:

1. A set of autonomous cycles your team can choose from. For each cycle, you will have the following information:
    - Its name.
    - Its speed, classified into Slow, Medium, and Fast. A Slow cycle does not have FM and FF action, a Medium cycle does not have FF action, and a Fast cycle has all three speed variants (FS, FM, FF).
    - Its reliability, classified as Reliable and NotReliable. When a cycle occupies a cell with a distractor in it, the cycle will be damaged. However, a Reliable cycle will only suffer minor damages, which costs your team $10 to repair. A NotReliable cycle will suffer major damages and will cost your team $75 to repair. Your team is required to do this repair, regardless of whether you will be using the cycle again or not.
    - Its endurance, which indicates its suitability for offroad racing. It has two classes: Domesticated and Wild. When an action passes through or ends at an obstacle cell, a Domesticated cycle will stop at the last cell before the obstacle and incur major damages, while a Wild cycle

will stop at the intended cell and incur only minor damages. The cost to repair major damages is $50 per hit, while the cost to repair minor damages is $5 per hit.
- Its price.

You can assume the following is true for any type of cycle.
- All cycles have the same error for action NE and SE. They move to the intended destination only 70% of the time and stay where they are 10% of the time. For NE, they will move to the cell at their North and East 10% of the time, respectively. For SE, they will move to the cell at their South and East 10% of the time, respectively.
- When an action hits the boundary of the track, the cycle will remain at the boundary cell.

2. A set of available tracks your team can race in. For each track, you will have the following information:
   - Track name.
   - The size of the tracks (in #horizontal cells X #vertical cells) and the map.
   - Positions of obstacles, which is deterministic.
   - Possible distractors, their positions, and the probability the distractions will occur.
   - Number of opponents and the strategies of each of them.
   - Registration fee to compete in this track.
   - Prize for winning the race in this track.

Input Format

The input consists of n+2 files, where n is the number of available tracks. The first file contains the set of autonomous cyle and its information. The second file is a meta-track file, containing the file names of the available tracks. Each of the next n files contain information about each track.

<u>The cycle file</u> consists of m+1 lines, where m is the number of available cycles. The first line is the number of autonomous cycles available. Each of the following lines is an information about one of the cycles. The format of this line is:

    Name SpeedClass ReliabilityClass EnduranceClass Price

<u>The meta-track file</u> contains k+1 lines, where k is the number of tracks available during Tour de UQ. The first line is:

    k startupMoney

Each of the next k lines is the file name of the available track.

<u>The track file.</u>

The first line is the size of the map, the number of opponents, registration fee, and prize money, in the following format:

#horizontalCells #verticalCells #opponents registrationFee Prize.

The next lines are the map, where 0 means empty cell, 1 means obstacle, each capital letter between A and J indicates the starting position of opponent#1 to opponent#10 respectively, each capital letter between K and Z indicates the starting position of the team's cycle, and each lower case letter means possible occurance of a distractor. For example,

A0a0
S010

Means that there is an obstacle in cell-(1,2), opponent-A starts from cell-(0,0), the team's cycle starts from cell-(1,0), and there is a non-zero probability that distractor-a appears in cell-(0,2).

The next lines after the map is the behavior of each opponent, represented as a stochastic policy. The first line for an opponent is the opponent ID. The next lines is the stochastic policy of the particular opponent. It is represented as a matrix, where:

- Each row represents the cells in the order of left-right and up-down.
- Each column represent action in the order of FS, FM, FF, NE, SE.
- Each element represents the probability that at the cell represented by the row, the opponent performs the action that corrresponds to the column.

For example, in the 2X4 track above, the matrix may be as follows (without the title in the first row & first column):

	FS	*FM*	*FF*	*NE*	*SE*	*ST*
0,0	0.5	0.2	0.3	0	0	0
0,1	0.2	0.5	0.3	0	0	0
0,2	0	0	0	0	1	0
0,3	0	0	0	0	1	0
1,0	0.3	0.2	0.5	0	0	0
1,1	0.5	0.5	0	0	0	0
1,2	1	0	0	0	0	0
1,3	0	0	0	0	0	1

This means that P(opponent-A performs FS | opponent-A is at cell-(0,0)) = 0.5, P(opponent-A performs FM | opponent-A is at cell-(0,0)) = 0.2, etc.

The next lines after the opponent is the behavior of each distractor. The first line for a distractor is the distractor ID. The next line is the probability that the disctractor will appear. For example,

a
0.7

Output Format

The format of the output file is:

Line1: The names of the cycle file and meta-track file
Line2: The cycles type you've bought
Line3: The ID of the first track compete in
Line4 to Line-i (where i = (4+(#stepsX(#rows+1))+#rows+1): Containing the game state at every step, represented in a map. If there are more than one cycles in the same cell, then we write the ID of each cycle in the cell separated by '-' and inside a square brackets (without space). For example if we use the track as shown in p.4 and if at some point in time, both your team's cycle and the opponent is at cell-(0,1) and the distractor does not appear, then the map should be shown as:

    0[A-S]00
    0010

Each game state is followed by a line of the action taken by your team's cycle in the format of cycleID-action. After all the game states have been printed, there is an additional line on the amount of prize gathered minus the total cost to repair the autonomous cycls(s).
The following lines are similar to Line3 until the previous line, but for your second race. These lines are then followed by lines representing the same information but for your last race.
Last line: The total amount of money at the end of Tour de UQ.

Note:

- **We will provide codes for simulating the opponent, distractor, and the cycles behavior, for reading input files, and writing output files.**

Grading

Solution & program: 15%.
Report: 5%.

Marking scheme for solution & software: Please take a look at Marking Scheme for Assignment 2 in the class website.