Artificial Intelligence Spring 2022
Term Project #1 Due Date: 06/10/2022 11:59PM

1. Project Objective

Utilize what you have learned in class plus any other relevant knowledge to write a program that plays against a human player (e.g., you) in a two-player board game – CANTRIS (Candy Crush + Tetris).

2. Language

C++ (C++14) / Python3.10

3. Games Rules

(1) An example the initial configuration of 8×4 game board with 32 numbered tiles. Each tile is indexed by its (row, column) coordinates, e.g., the tile 3 in a red circle is indexed by (2, 1).

My points = 0
Computer's points = 0

4	2	2	4
3	1	3	4
1	3	1	2
3	2	4	1
1	1	3	2
2	4	1	1
2	3	4	3
4	4	2	3

(2) Rules

2-1. When it is your turn, you can either remove only one tile or take two tiles off the board sequentially, detailed in (3) below. The number of the selected tile is then added to your points.

2-2. All the tiles on the same column above the one you remove will drop by one position after the removal, e.g., after the tile 3 is removed, both the tiles 2 and 1 above it will drop by one position to fill the vacancy. Your total points are now 3. See below.

My points = **3**
Computer's points = 0

4		2	4
3	2	3	4
1	1	1	2
3	2	4	1
1	1	3	2
2	4	1	1
2	3	4	3
4	4	2	3

2-3. After the tiles fall to their lawful positions, in case of three or more adjacent tiles with the same number on the same row will be further removed from the board, and the sum of the numbers of the tiles is added to the points. **The steps 2-2 and 2-3 are repeated if applicable until no more tile can be removed.** After that, the game switches to Computer's turn, and all the rules above apply in the same way.

E.g.,

My points = 3
Computer's points = 0

4		2	4
3	2	3	4
1	1	1	2
3	2	4	1
1	1	3	2
2	4	1	1
2	3	4	3
4	4	2	3

My points = **6**
Computer's points = 0

			4
4		2	4
3	2	3	2
3	2	4	1
1	1	3	2
2	4	1	1
2	3	4	3
4	4	2	3

2-4. The game is over when any column on the board is cleared.

2-5. When the game ends, the player that has higher total points wins.

(3) There are three different board dimensions, i.e., 6×3, 8×4 and 10×5.

- 6×3: the tiles are numbered from 1 to 3
- 8×4: the tiles are numbered from 1 to 4.
- 10×5: the tiles are numbered from 1 to 5.

Remove only one tile or two tiles sequentially in turn.

- 6×3: can remove only one tile.
- 8×4: can remove only one tile.
- 10×5: can remove two tiles sequentially, i.e., remove one tile first, and let tiles fall to positions, and then remove the second and let tiles fall to positions.

(4) Testing

- Your program will be tested twice on each of all board dimensions,6×3, 8×4 and 10×5. In the two tests, your program runs as 1st and 2nd player, respectively.
- Board.txt will be replaced by TA while testing, but same specifications.
- Environment: Win10, 11th Gen i5-11400, 48GB RAM, GPU is not allow.

4. What to submit

(1) Your report (student-ID.pdf) (40%), e.g., 309515003.pdf
(2) Your source code (student-ID.xx) (60%), e.g., 309515003.py
Source code covers three parts:
(2-1) Select one tile (set row = 6, col = 3), e.g., 309515003_1.py
(2-2) Select two tiles, e.g., 309515003_2.py
(2-3) Select one tile (set row = 8, col = 4), e.g., 309515003_3.py

(3) Please zip all your files into student-ID.zip
(4) Any violation of the file name format will incur 5 pts penalty deduction.

5. Grading Policy (total 100 points)

- Coding: total 60 pts

(1) Baseline: Any executable code without plagiarism gets 30 pts; otherwise, 0 pt.

(2) For the remaining 30 pts, any of the following cases will incur a penalty point deduction.

- Running time over limit (30 sec for one player's turn): -5 pts

- Late project submission: -10 pts per day

You (i.e., computer programs) will be randomly grouped, and a double elimination tournament will be held. The more games you win, the higher score you get (30 pts total).

- Documentation: total 40 pts

(1) Design philosophy (20 pts)

- If your algorithm completely relies on randomness, you get zero point.
- Clearly describe your algorithm.
  - Search strategy
  - Heuristics (if there is any)

(2) Discussion (20 pts)

- Motivation behind your search strategy and heuristics
- Challenges in implementation
- Improvement by your methods
- Any lesson learned from project

Note.

- You only need to make your own move, TA will input your opponent's move.
- If you have any question about the project, please contact TAs.