

CISC 489/689 Intro to Multi-Agent Systems  
Programming Assignment 2  
Simulation of Competitive Agents in a Tileworld  
Due: **Friday, May 13, 11:59 pm**

### The Simulation Environment

- Please download the simulator from [here](#). You can use any IDE you like (e.g. PyCharm, VSCode, etc). The basic environment is almost the same as the last time, except that the two agents will be generated at random positions (instead of the top left corner).

### Task Description

- There are three parts of this programming assignment, that is to make the following simulation cases work:
  - Case 0 (0 points): one `demoPlayer()` and one `randPlayer()` are in the Tileworld, where you can control the `demoPlayer()` and get familiar with the rules.
  - Case 1 (10 points): two `randPlayer()`'s are in the Tileworld, which is used as the benchmark (this is the base case: there is nothing for you to write, just make sure it works).
  - Case 2 (40 points): one `randPlayer()` and one self-designed agent (i.e. `PlayerA()`) are in the Tileworld. Your design should show your agent is rational.
  - Case 3 (50 points): two self-designed agents are in the Tileworld, where you test if your agents are competing well with each other. Your design (especially `PlayerB()`) should show how effectively it competes with others.
- Note: This is a test environment for you to try out your strategies, you can make changes, say the size of the map (i.e. N), the number of coins generated each round, the length of time for each round, etc. Your code will be tested in a different set of variables (like a larger map, more rounds, longer game time).

### Submission

- 1. Your code (`compAgent.py`).  
Basically, only `compAgent.py` will be tested to ensure fairness (and class `PlayerB()` will be used for the tournament, in which your designed agents will compete with one another).  
If you made any changes to the other files in order for your code to work (e.g. certain Python packages), please point them out. In this case you may also want to upload the whole set of your code just in case.
- 2. A short report (no limitation on the number of pages, but preferably less than 2 pages not including appendices).

Please describe how your design works (e.g. features, algorithms, etc), the comparison among the three cases, and the advantages of your design (e.g. how your agents compete, how your design of a certain agent gets higher utility, etc).

### Team Formation

Each team can have up to 2 students. You can find a teammate by posting on [Discord](#). Each team only needs to make one submission.

### Bonus Point

- Try to polish your design. We will run your designed agent (`PlayerB()`) against each other in pairs like a sport tournament. Therefore, please make sure your (`compAgent.py`) works well with the original Tileworld environment (you may just redownload it and replace the `compAgent.py` file with yours). If your `compAgent.py` file cannot work with the original environment, that means we cannot make your agent compete with other agents, then you'll be considered as waiving the rights for extra credit (and probably also lose points since we cannot test how well your agent (`PlayerB()`) competes with others).
- We'll run the tournament in five different environments. Every agent who wins 3 of the games will move to the next level until we have a champion.
- The top three agents will receive additional rewards (+20% for 1st place, +10% for 2nd place, and +5% for 3rd place.).

### Integrity

Your code must be your own work. If you use ideas, code pieces from others, please cite them properly in the report.

### Questions?

Please post them on [Discord](#), and consider replying to your classmate's questions. You can also attend TA's office hours or contact the TA.

### Resources

- Pygame Docs: <https://www.pygame.org/docs/>
- Pygame on github: <https://github.com/pygame/pygame>
- A brief Pygame tutorial: <https://www.youtube.com/watch?v=FfWpgLFMI7w>
- Python / Pygame tutorial: Collisions between static and moving objects: [https://www.youtube.com/watch?v=1\\_H7InPMjaY](https://www.youtube.com/watch?v=1_H7InPMjaY)