National University of Singapore
School of Computing
CS1010S: Programming Methodology
Semester I, 2016/2017

Mission 10.2 - Contest
2048 Solver

Release date: 10 October 2016
Due: 20 October 2016, 23:59

Required Files

- sidequest10.2-template.py
- puzzle_AI.py

Background

Grandwizard Ben has trained his pet toad to play the 2048 game. Unfortunately, the result are rather dismal, in fact, terrrible. You look at your adorable pet hamster. "Surely this little chap is a wee bit more intelligent. Surely, dear friend, you can't lose to a... toad." Your imagination begins to cloud with grand aspirations for your pet hamster.

Mission Brief

In this contest, you will compete to create the best solver for the game *2048*.

Entries will be judged foremost based on how often the solver manages to obtain the 2048 tile. The average score achieved will be used as a tie-breaker in the event where solvers have a comparable winning rate. Only one entry is allowed per apprentice. There will be a time limit of 10 minutes per entry. Anything done after the time limit will not be counted.

The template file `contest10.2-template.py` has been provided for you. In addition, another file `puzzle_AI.py` has also been provided. It contains functions that will render the graphical interface for your game as well as an implementation of all the functions you have developed in the sidequest.

After you have completed you solutions in the template file, copy only your function to coursemology. Do not `import puzzle_AI`.

Task

Write a function `AI(mat)` that takes in a game matrix `mat` and return one of the following strings: `'w'`, `'a'`, `'s'` or `'d'` which represent moves in the upward, leftward, downward

and rightward directions respectively. Your final submission should produce only valid moves.

Your solver should be able to continue any arbitrary game, including the one played halfway by Grandwizard Ben's pet toad. No undos are allowed. You are also not allowed to store information using global variables (or by any other means) from call to call.

Testing Your Solver

Uncomment the below lines to watch your solver's move-by-move execution:

```
game_logic['AI'] = AI
gamegrid = GameGrid(game_logic)
```

Press any key to see the next move each time. Uncomment

```
get_average_AI_score(AI, True)
```

instead to grade your solver according to the competition criteria.