CSE 348: AI Game Programming
Project 2: Pathfinding in games
Due date: Wednesday, February 29

We will use the UC Berkeley Pacman software for this project. Here are the links:

- General pacman description and python tutorial:

  http://www-inst.eecs.berkeley.edu/~cs188/pacman/pacman.html

- The actual tasks that you need to perform are described below and correspond to some of the questions as they appear in:

  http://www-inst.eecs.berkeley.edu/~cs188/pacman/projects/search/search.html

Tasks for you to be performed:

1. **Task 1**: Question 1 from UC Berkeley project (**depth-first search (DFS)**)

2. **Task 2**: Question 2 from the UC Berkeley project (**breadth-first search (BFS)**)

3. **Task 3**: Question 4 of the UC Berkeley project (**A* search**).

4. **Task 4**: We are going to implement **transition tables**. As explained in class, commercial games frequently use transition tables to maintain all paths in a map (these paths are precomputed). The transition table is a two-dimensional array with one column and one row for each cell in the map. Given any two cells, A and B, in the map then

   transition[A,B] = "the cell neighboring A that is on the shortest path to B"

Create a program that automatically fills in the transition table for the given map and then compute the shortest path between the two given nodes using the table.

5. **Task 5**: (in-class competition): Question 7 of the UC Berkeley project (**eat as fast as possible**). We will run all programs in class in the same machine and the winner will be the program that runs the fastest. Notes:

   A) Ignore the consistency issues discussed in Question 7
   B) Transition tables (or any memory build beforehand) are not allowed for the competition.

Deliverables:

- You have until 11:59PM EST on Wednesday, February 29, 2012 to email your code with all five (5) tasks as described above to:

  "James F. Ahlum (Graduate Student - EN/)" jfa306@lehigh.edu

- Please bring for class on Thursday, March 1st a short presentation (1-3 slides) indicating how you performed Task 5.
- Bring for class on Thursday, March 1st a short report (2 pages maximum): where you report **for each Task 1 to 4** above: (1) what is your solution for the task, (2) what is the running time for that task, (3) how does the solution speed for this task compares to the other three tasks, (4) why you think this is the case

  Notes:

  a. For task 4, only compute the time to find the path **after** the transition table is constructed.
  b. For tasks 1 to 4, please use the "bigMaze.lay" map from the distribution
  c. You are going to need to need data structures for this project. You are strongly advised to use the ones implemented in the pacman software (i.e., util.py).