# Hand-In for Homework 2 – Uninformed and Informed Search Algorithms on Pacman

## Description

The uninformed and Informed search algorithms are implemented on the framework provided as part of the CIG conference 2016 http://cig16.image.ece.ntua.gr/competitions/ Game AI competition.

I have used Bread First Search for uninformed and A-star algorithm for informed search.

For uninformed search I have used BFS due to the following advantages:

1) Easier to implement
2) BFS will never get trapped exploring the useless path forever.
3) Also, if there is a solution, BFS will definitely find it.

For informed search I have used A-star has the advantage that among all shortest-path algorithms using the given heuristic function h(n). A* algorithm expands the fewest number of nodes.

## Files

1) Package main.java.entrants.pacman.neethu.controllers :
   BFSPacMan.java

   AStarPacMan.java

2) Package main.java.entrants.pacman.neethu.util (used by AStarPacMan.java)
   MazeNode.java
   Util.java

## Instructions for compiling and running

### Step 1

1) For executing BFS, you can uncomment the following line in Main.java
   executor.runGameTimed(bfs, ghosts, true);
2) For executing A-star, you can uncomment the following line in Main.java
   executor.runGameTimed(astar, ghosts, true);

   By default, I have used POCommGhosts.
   You can pass randomGhosts object as executor.runGameTimed(astar, randomGhosts, true) if you want to use RandomGhosts

### Step 2

If you are using Eclipse IDE, you can run the code by selecting Run-> Run as -> Java Application and execute the Main.java file.

# Analysis

## Space Complexity

### Breadth First Search (BFSPacMan.java)

Considering the number of nodes in the maze as n.

ArrayList path and HashSet stepsTaken, which stores the BFS path to target and unique steps taken by pacman respectively, can utmost have n elements,

Same is the case with pills, powerPills, targetsArray, q, visited, paths where

- ‘pills’ is an integer array which stores the indices of pills in the maze,
- ‘powerPills’ store indices of power pills in the maze,
- ‘targetsArray’ stores (pills + power pills)
- ‘q’ is a Queue data structure which stores the open nodes
- ‘visited’ is a Queue data structure which stores the visited nodes

So space complexity is O(n).

### A-Star (AStarPacMan.java, MazeNode.java, Util.java)

Considering the number of nodes in the maze as n.

Graph, path, stepsTaken, pills, powerPills, targetsArray, q, visited, open, visited has space complexity O(n) as these data structures can have atmost n elements.

## Time Complexity

### Breadth First Search (BFSPacMan.java)

Time complexity is $O(b^d)$ where b is the number of child nodes( branching factor) and d is the depth at which target is present.

### A-Star (AStarPacMan.java, MazeNode.java, Util.java)

Time complexity is $O(b^d)$ in the worst case. But since the heuristics used in admissible (shortest distance), it prunes away many of the $b^d$ nodes that an uninformed search would expand