2017 Fall AI **National Taiwan University**
Instructor: Tian-Li Yu **B03901133 Gary Shih**
HW2 **October 18, 2017**

Question 5

In question 5, I implement the following techniques:

1. Real Distance
   Considering walls and reasonable ghosts, we can more reliably calculate distances to all points than Manhattan Distance by Breadth-First Search. Then, once we have distances to foods, capsules and ghosts, we will be able to let Pacman know where it should go in some priorities.

   However, how do we define reasonble ghosts? We know that there are two states of ghost, normal and scared. We should avoid to meet normal ghosts while in the case of scared ghosts, if we can catch them before they become normal, we can get much more score. Therefore, the definition of a reasonable ghost in my design is that a normal ghost or a scared ghost which Pacman approximately is not able to catch it before it becomes normal.

2. Rewards
   After we know distances to those useful targets, we can apply some value to them and let Pacman know where it should go and avoid. In my implementation, I do linear combination to the reciprocal of these distances and the following is the weights I use:

   - Food: 1.0
   - Capsule: 100.0
   - Scared ghost Pacman can catch: 200.0

3. currentGameState.getScore()
   This part is the same as scoreEvaluationFunction which provide scores that includes basic win, lose, and time penalty conditions.

Note

An important point is that I do not additionally penalize the condition where Pacman may meet a normal ghost because I found that if I do, Pacman will become too conservative to eat foods and capsules. Besides, in our testcase, there are just two ghosts randomly move. As a result, currentGameState.getScore() is enough to let Pacman avoid ghosts.