CS533 Assignment 1
Pac-Man MDP

1 State Space (S)

The state will be a tuple containing several items, each of which is given as a human :

1. Pacman position [`2D vector`]
2. Pacman power-up timer (represented in number of frames until ghosts can kill Pacman again) [`positive integer`]
3. List of Ghost
   (a) Position [`2D vector`]
   (b) Direction enumerated type (North, South, East, West) [`integer`]
4. Wall grid [`Boolean matrix`]
5. Dot grid (0 is no dot, 1 is basic dot, 2 is power-up dot) [`Ternary matrix`]

2 Action Space (A)

There are 5 actions available to our agent, four of which having the expected meaning (Pacman moves 1 square in the prescribed direction). The last action ($No - Op$) corresponds to Pacman remaining still for a timestep, while the ghosts move around him. $A = \{North, South, East, West, No - Op\}$

3 Reward Function (R)

There are 3 sources of reward:

1. *Eating a dot, either basic or power-up (1 point)*
   Given when computing the new state passes tests 4 or 5 in Section 4.
2. *Eating a ghost (100 points)*
   Given when computing the new state passes tests 2.(a) or 6.(c).i in Section 4.
3. *Eating all dots (1000 points)*
   Given when computing the new state passes tests 4 or 5 in Section 4 AND the dot grid is a 0 matrix.

4 Transition Function (T)

WLOG, we can lump the 4 MOVE actions together in discussing how the state transitions occur, since they are essentially isomorphic up to rotation. To compute the new state given the old state and an action, first we assume we are operating on a copy of the old state. Second, note that each of these rule applications are deterministic (happens with probability 1), other than 6.(a), which will be a uniform distribution over the available options. To compute the new state given the above assumptions, we must:

1. If the power-up timer is positive, decrement it in the new state
2. If the action is MOVE and destination contains GHOST
   (a) If the power-up timer is positive, eat the ghost (update position to center box, direction to something reasonable, and we may want to flag it so it does not move this timestep)
   (b) If the power-up timer is 0, lose the game
3. If the action is MOVE and destination contains no WALL, update pacman position
4. If the action is MOVE and destination contains BASIC DOT, remove it from the dot grid
5. If the action is MOVE and destination contains POWER-UP DOT, remove it from the dot grid and set the power-up timer to k.
6. For each ghost
   (a) If destination contains WALL, assign new unobstructed direction
   (b) Else update ghost position
   (c) If destination contains PACMAN
      i. If the power-up timer is positive, eat the ghost
      ii. If the power-up timer is 0, lose the game