**Homework #4: Blackjack**

# Acknowledgements and Disclaimers

I collaborated with Nick Landy and Megan Knight in formulating my answers to this homework.

# Problem 1

## Problem 1a

For this problem, we have n+1 players consisting of a single agent (Pacman) and the n opposing players (ghosts). In order to understand how to write down our value function, we can express each state as follows:

* **At the end state**, the result is simply the utility derived from that state as determined by the game’s utility function.
* **At the maximum depth**, the agent uses the evaluation function for the maximizing selection instead of searching further into the tree
* **If the player is the agent,** the value function will be recursively called for all of the possible successor states, and the maximum of these will be selected (if the maximum depth has not yet been reached)
* **If the player is an opponent, but NOT the last opponent,** the value function will be recursed and the minimum will be taken (since all opponents are min agents)
* **If the player is the LAST opponent,** the value function will be recursed and the depth decremented by one (since a single depth layer constitutes the movement of the agent and ALL of the opponents)

Therefore, the recurrence of the value function can be written as:

# Problem 3

## Problem 3a

The expectimax function should be identical to the minimax function defined for Problem 1 in all respects except for the behavior of the ghost players, since their behavior is being changed from a minimizing behavior to the expected value of their behavior based on the random policy . Therefore, the expectimax recurrence is given as:

# Problem 4

## Problem 4b

At a high level, I wanted my evaluation function to “augment” the base score of the given state s (i.e. the default evaluation function used for the other parts of this assignment) with additional information. This information needed to allow Pac-Man to trade off between parameters that are relevant for estimating the utility of a given state, which fundamentally reduce to a cost-benefit calculus between the ease with which food can obtained and the propensity of coming into contact with a ghost.

* The total distance between Pac-Man and all of the other ghosts on the board (higher is better)
* The distance between Pac-Man and the nearest ghost (higher is better)
* The distance between Pac-Man and the nearest food capsule (lower is better)
* The distance between Pac-Man and the food capsules more generally, which was calculated as the distance to the centroid of the food capsule coordinate pairs (lower is better)