Q1: The main concern of my reflex agent is to not get eaten, which returns an evaluation result of -10000. After that the agent checks if the action consumes a food and adds 1000 to the result if it does. Afterwards I subtract the Manhattan distance to all food particles left on the map, I used subtraction here because larger distances should result in worse states. Similarly, I subtract the Manhattan distances to ghosts while they are scared and add them if they are not scared. It may seem counterintuitive to count distances to ghosts as positive but keeping them away gives a higher positive value than being near them. Finally, I subtract 500 from my evaluation if the move in question is a STOP move, because they are very rarely beneficial and can cause Pacman to get stuck. I used several negative values and I believe it is a good idea, but I did not use any reciprocals. This is because they are beneficial for multiplication and division, and I did not use these operators.

Q2: AlphaBetaAgent acts considerably faster than MinimaxAgent because it prunes the redundant leaves. Because both agents always consider that their adversaries will take the best course of action, it is possible to detect that a path will never be taken without checking its entire leaves and AlphaBetaAgent uses this to decrease its search cost. Pruning is more likely to occur if a max node's early checked leaves is very large or if a min node's early checked leaf is very small.

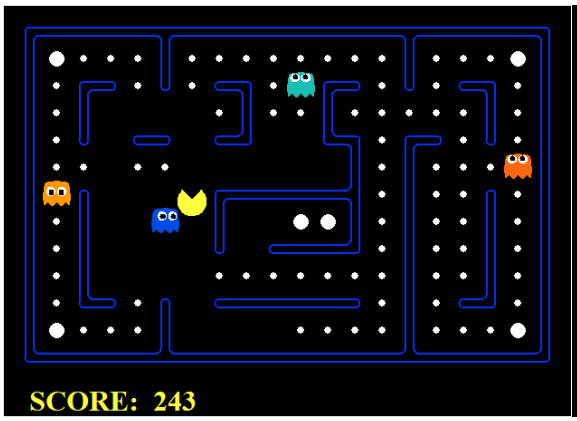

Figure 1: Minimax

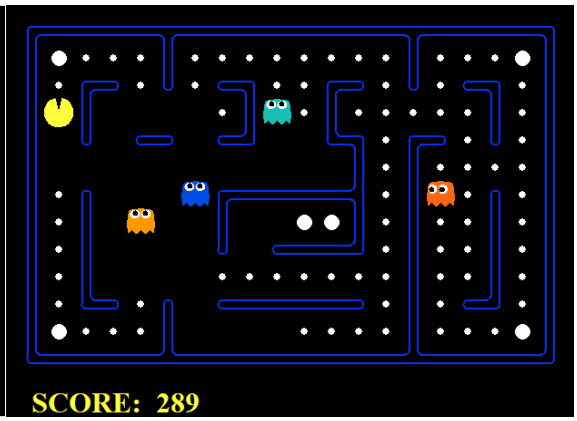

Figure 2: Alpha-Beta at 20sec

Q3: They behave in the exact same manner because while the value of intermediate nodes may be different in Alpha-Beta Pruning, the result of the top node is always the same because it only prunes the redundant nodes.

Q4: Expectimax works at a similar speed when compared to Minimax as they do the same amount of calculations, because both do not prune redundant leaves. It also followed a different path than the others because it does not expect its adversaries to take optimal paths like the other two.

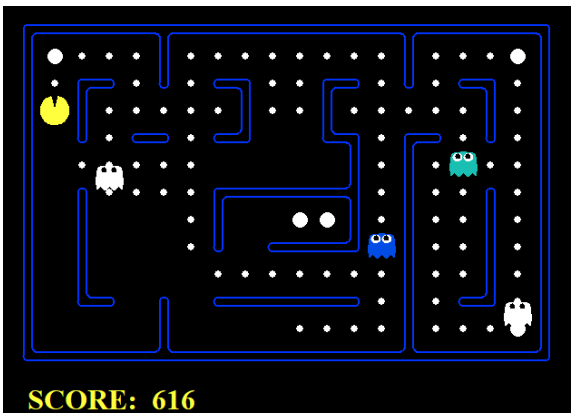

Figure 3: Expectimax at 20sec

Q5: I used the score function because this time a state's goodness when compared to all other states is important. I removed the successor-action based checks from my evaluation and added a check for distances to capsules because the point constraints were harsher in this question. I multiplied my additions to my evaluation result with weights to better represent the value or danger of the surrounding objects. My Pacman is also paranoid of coming next to a ghost because it decreases the evaluation result severely.

Q6: For Q1 I did not use weights, I only awarded special cases like eating a food and made Pacman hate being eaten or stopping. For the final question I gave weights to distances to food, capsules and ghosts, being 10, 20 and 100 (- or + depending on being scared) respectively. I chose to value capsules more than food because it gives the opportunity to earn several points, and I prioritized ghosts the most because they net high points when they are scared and stay in this state for a limited time. I also used hard constraints for the case of endangering Pacman or consuming food.