Assignment 2 – Algorithms and Data Structures

Tom Hanson – 912033 – hansont – 23/10/19

# Experimentation

To ascertain the effectiveness of my ai and its 2 propagation types, experimentation of the ai was undertaken. For each propagation method, maximum and average, 4 budget sizes (10,100,1000 and 2000) were ran a total of 3 times. Each time, the total time, expanded/second and final score. The average and standard deviation (of the final score only) of these three runs were calculated and graphed. This was repeated 2 more times on different levels to reach a total of 72 runs. The graphs of the final scores vs budget can be found in the figures (1,2 and 3) below. The comprehensive table of results can be found in appendix one.

*Figure 1, Final score is graphed against budget for the 2 propagation types, tested on level 1*

*Figure 2, Final score is graphed against budget for the 2 propagation types, tested on level 2*

*Figure 3, Final score is graphed against budget for the 2 propagation types, tested on level 3*

# Analysis

From the graphs we can see that, generally, maximum propagation w results in a higher final score than average propagation. I believe this is due to how each of the program handles losing lives and when food becomes sparser. In the average case, the points associated with food becomes insignificant compared to the value of the heuristic, and thus, will not make moves towards ghosts even if the last remaining food is in that direction. Compared to maximum, which is generally more aggressive towards food and will deal with the ghosts when they become an issue.

The only case where average gets ahead of maximum is in the budget = 10 case. This is because a more cautious approach, loosing life has a greater weighting in the average case, works better when the AI has less possible future states to compare.

In level 2 and 3, a budget of 1000 was found to be optimal, while in level 1, budget = 2000 was optimal. This is probably because was the levels become more complex, many future actions lead to life loss. This results in paths that previously appeared safe become more dangerous. Thus, Pacman is less likely to follow these paths, even if this is where the food is located.

My implementation made one optimisation. I found that the Pac-man would oscillate backwards and forwards until a ghost would come nearby. To counteract this I implemented a condition that Pac-man cannot backtrack (ie go left after going right), unless not doing so would loose a life. This optimisation functions well for the most part. However, it can fall into issues around corners, or if there is an equal but odd number of squares between Pac-man and a ghost (ie. they arrive at the same square at the same time). Future implementations of this optimisation could allow for earlier backtracks, and backtracks allowed for other conditions such as eating, food, fruit and ghosts while invincible.

# Appendix 1

The full table of results for all levels, propagation types and budgets, including standard deviation of the final score for each budget.

