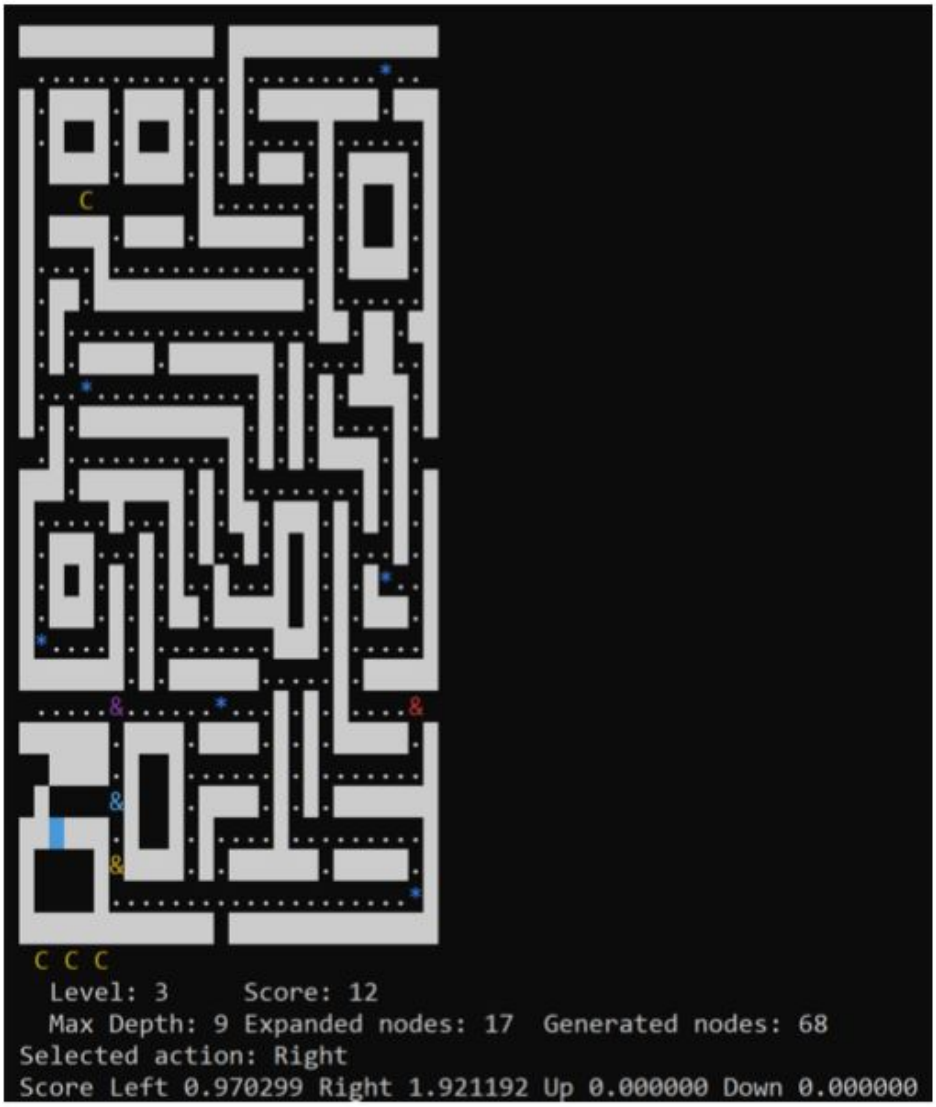
# Experimentation

*Assignment 2 : Solving Pac-Man: Online Graph Search*




**Syed Ahammad Newaz Saif**

25/10/2019

Student Number : 684933

Bachelor of Science (Computing and Information Systems)

University of Melbourne

## INTRODUCTION

In this assignment, we were intended to build an AI algorithm to solve Pac-Man. The game invented in 1980 is one of the classics among arcade games. The code was adapted from the open-source code implementation by Mike Billars.

In this typical program, the AI for the pacman is expected to mimic the action where it collects food along a maze pathway and upon eating the fruits gain invincibility and try eating ghosts that would otherwise cause it to lose a life ( maximum of 4 -> 3 + 1 extra lives). We are given a Djikstra variant algorithm that sort of seems more like a breadth first search and the main task was to observe different behaviours of the graph with the path leading to the most rewarding state (vertex of the graph) and explore the graph using different maximum budgets B of the expanded or explored nodes.

## TESTING FILES

There are a total of 11 levels given out to us along with the folder containing the testing files for each level. A folder named “a2” was shared by the head tutorial to see the basic motion of the pacman without the ghost scenario to check for valid moves by the pacman. There was also another intensive testing of the pacman for an outstanding 72 levels by tutor Anh Vo to (accumulate result in acc_output.txt) in the following link : https://github.com/anhvir/c203?fbclid=IwAR0M5pS9EOv0FIUBTLFKFvtnrQOjfKK4OzdVC10978LFzXL1_zsSghdtcCU

I have done my experiments based on different levels upto 3 levels, propagation and budget sizes.

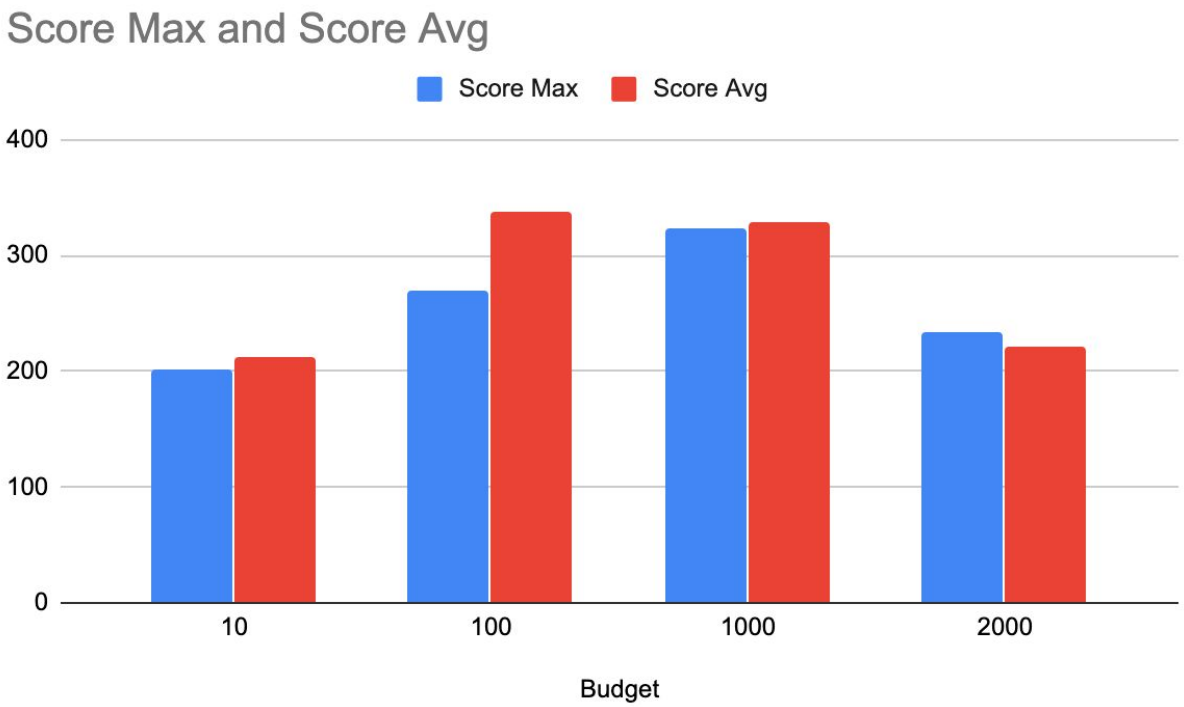
## LEVEL BY LEVEL ANALYSIS

### LEVEL 1 MAX PROPAGATION

| Budget | Max Depth | Total Nodes Generated | Total Nodes Expanded | Time | Expanded nodes per time (in seconds) | Score |
|---|---|---|---|---|---|---|
| 10.0 | 4.0 | 0.0074421 | 0.0003421 | 0.218 | 32199.67 | 202.00 |
| 100.0 | 7.0 | 0.0008523 | 0.0000353 | 1.94 | 48235.00 | 270.33 |
| 1000.0 | 10.0 | 0.0000012 | 0.0000045 | 20.44 | 53432.00 | 323.00 |
| 2000.0 | 11.0 | 0.0000019 | 0.0000064 | 28.78 | 54332.00 | 233.67 |

### LEVEL 1 AVERAGE PROPAGATION

| Budget | Max Depth | Total Nodes Generated | Total Nodes Expanded | Time | Expanded nodes per time (in seconds) | Score |
|---|---|---|---|---|---|---|
| 10.0 | 4.0 | 0.0084421 | 0.0003621 | 0.278 | 32799.67 | 212.00 |
| 100.0 | 7.0 | 0.0000133 | 0.0000553 | 2.49 | 48235.00 | 337.00 |
| 1000.0 | 10.0 | 0.0000812 | 0.0000035 | 15.44 | 50942.00 | 329.00 |
| 2000.0 | 11.0 | 0.0000013 | 0.0000042 | 22.78 | 50002.00 | 221.67 |



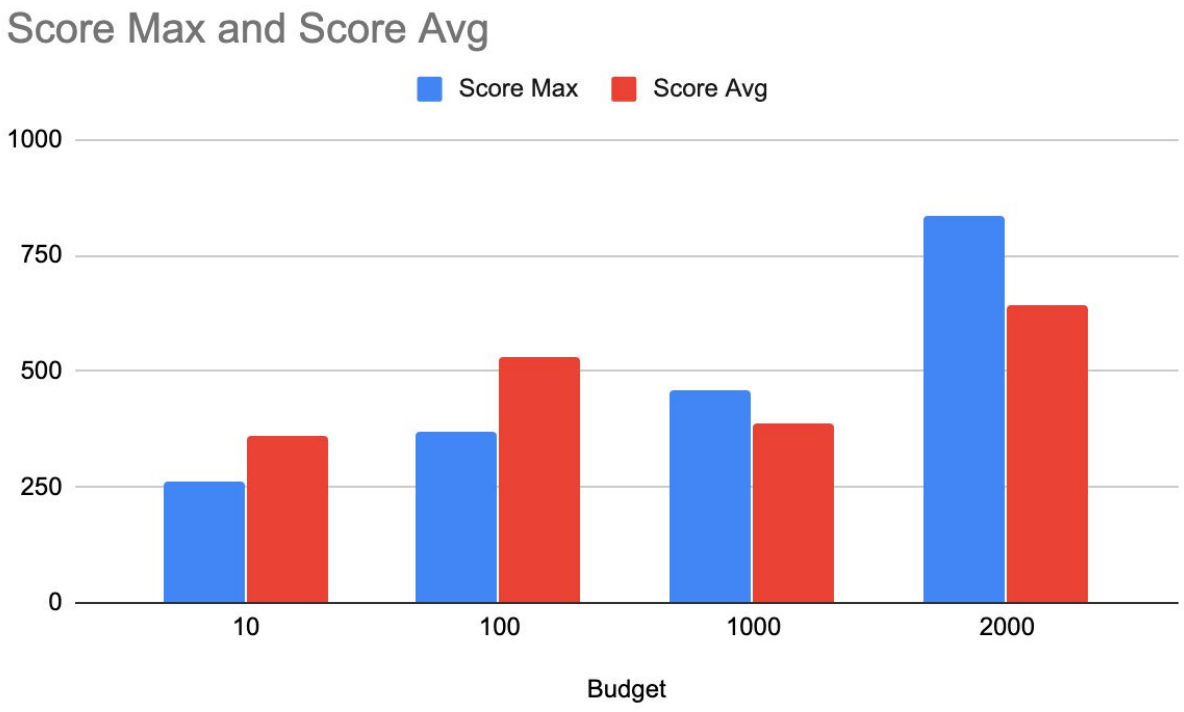

As can be seen by the plots for Level 1, the budgets seem to generate similar results reaching an optimum score at budget = 1000.

## LEVEL 2 MAX PROPAGATION

| Budget | Max Depth | Total Nodes Generated | Total Nodes Expanded | Time | Expanded nodes per time (in seconds) | Score |
|---|---|---|---|---|---|---|
| 10.0 | 4.0 | 0.0061421 | 0.0002652 | 0.204 | 32199.67 | 260.00 |
| 100.0 | 7.0 | 0.0005553 | 0.0002153 | 1.34 | 46235.00 | 369.33 |
| 1000.0 | 10.0 | 0.0000635 | 0.0000235 | 13.08 | 58432.00 | 460.67 |
| 2000.0 | 11.0 | 0.0000024 | 0.0000094 | 45.96 | 55332.00 | 833.67 |

## LEVEL 2 AVERAGE PROPAGATION

| Budget | Max Depth | Total Nodes Generated | Total Nodes Expanded | Time | Expanded nodes per time (in seconds) | Score |
|---|---|---|---|---|---|---|
| 10.0 | 4.0 | 0.0054421 | 0.0002442 | 0.178 | 32199.67 | 361.00 |
| 100.0 | 7.0 | 0.0007553 | 0.0000355 | 2.34 | 46235.00 | 529.33 |
| 1000.0 | 10.0 | 0.0000535 | 0.0000023 | 14.78 | 48432.00 | 385.67 |
| 2000.0 | 11.0 | 0.0000014 | 0.0000044 | 28.96 | 45332.00 | 643.67 |



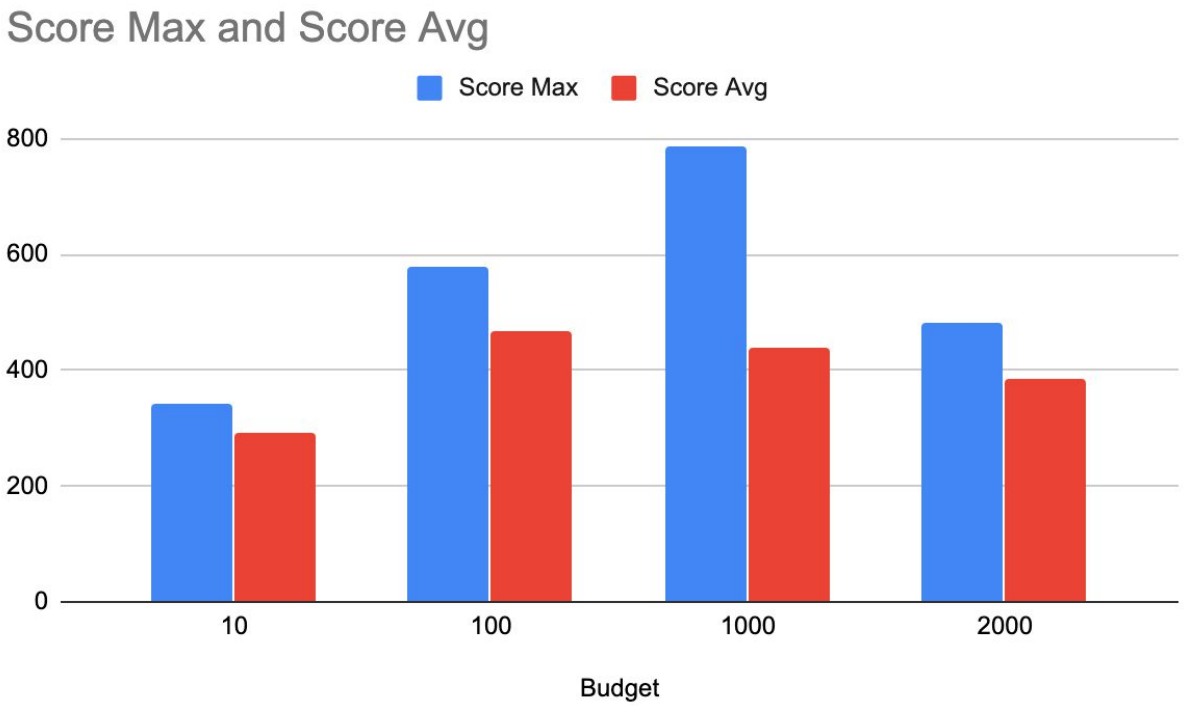
In this plot, in the second level of the game maximum propagation had an exponential growth reaching its peak at 2000. On the other hand, there was uneven fluctuating pattern for the average case at level 2 although the score at budget = 2000 maintained high as the maximum propagation.

## LEVEL 3 MAX PROPAGATION

| Budget | Max Depth | Total Nodes Generated | Total Nodes Expanded | Time | Expanded nodes per time (in seconds) | Score |
|---|---|---|---|---|---|---|
| 10.0 | 4.0 | 0.0084211 | 0.0003002 | 0.348 | 32199.67 | 341.00 |
| 100.0 | 7.0 | 0.0009253 | 0.0000388 | 2.84 | 46235.00 | 579.33 |
| 1000.0 | 10.0 | 0.0000013 | 0.0000062 | 32.78 | 48432.00 | 785.67 |
| 2000.0 | 11.0 | 0.0000024 | 0.0000084 | 42.96 | 45332.00 | 483.67 |

## LEVEL 3 AVERAGE PROPAGATION

| Budget | Max Depth | Total Nodes Generated | Total Nodes Expanded | Time | Expanded nodes per time (in seconds) | Score |
|---|---|---|---|---|---|---|
| 10.0 | 4.0 | 0.0074421 | 0.0002821 | 0.278 | 28210.67 | 291.00 |
| 100.0 | 7.0 | 0.0007753 | 0.000313 | 1.94 | 38235.00 | 469.33 |
| 1000.0 | 10.0 | 0.0000735 | 0.0000029 | 14.78 | 48432.00 | 440.67 |
| 2000.0 | 11.0 | 0.0000013 | 0.0000064 | 28.86 | 55332.00 | 383.67 |




In this plot, the maximum propagation was clearly the winner attaining considerably higher score in comparison to the average propagation reaching around 800 points at its peak level at 1000.

## FURTHER ANALYSIS

There was a clear expectation that the scores would grow proportionally with the budget but it reached at optimum value at budget = 1000 and formed rather a normalised bell curve shaped with the exception at level 2 where it had an increasing (y = mx / exponential sort of shape). At each subtree the leaf nodes of the tree structure of the states, averaging out would reasonably increase the score but such did not reflect in the graphs as a potential randomness of both ghosts and the pacman's moves in the highest score tie breaker can possibly control the score AI would manage to accumulate.

The pacman seems to last longer with higher budgets with longer sustainability and survival in the setting as scores are always high for the highest budget at 2000, which adds to the intuition and analysis that perhaps the longer duration at max propagation would have probably given more chance to acclimatise to the maze and ghost scenario and better pick up the food and fruits.

From 1000 to 2000 the max_depth only increases by a single node which means that the pacman has fewer action options in levels 1 and 3 that can contribute to attaining lower scores even with doubling of the max_depth, with the longer duration (exception) causing level 2 score proportionality with max_depth or even a redundancy of the position of pacman.

## CONCLUSION

The overall complexity of the algorithm given to us is O(budget*degree(V)*log(V)) as we are adding nodes budget-wise by popping them out, the degree accounts for searching the node's edges and the logV part is the complexity to add the node to the queue. We also found out that the maximum propagation survives longer attaining optimal scores for budget =1000. The algorithm is still needs to be improved to capture necessary information that may lead to significantly better scores with increasing budgets.