WID 3009
Artificial Intelligence Game Programming

Lecturer: Dr. Loo Chu Kiong

Maze Generation:
Evaluation of the Complexity of Constructive Procedural Content Generation

---

Wong Hao Shan
17122789/1 (WID170054)

Procedurally Content Generation of Maze: Hunt and Kill Algorithm

The Hunt and Kill algorithms is the PCG method selected for maze generation in Unity. Hunt and Kill use the idea of the recursive backtrack. The algorithm picks a random point and starts hunting until it hits a dead end. At this point, recursive backtracker would take a backtrack to the last vertex with unvisited neighbours while Hunt and Kill Algorithm does something different, it will scan the maze for an unvisited cell at restart the walking process at that location. It continues this process until all vertices are visited.

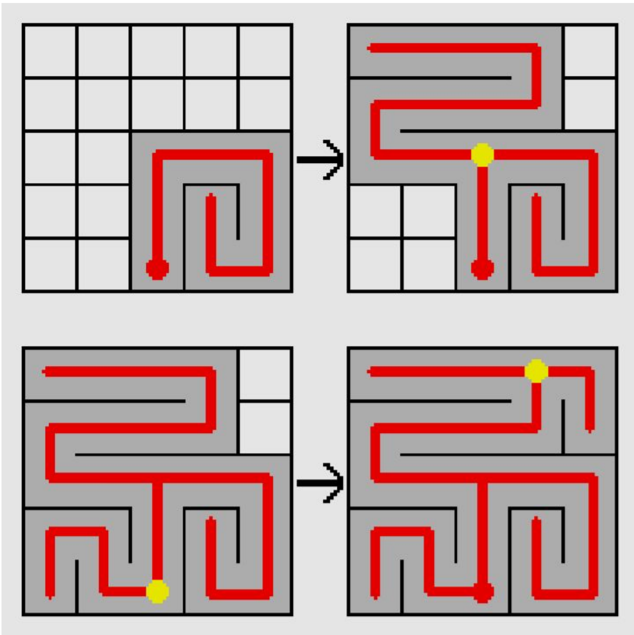
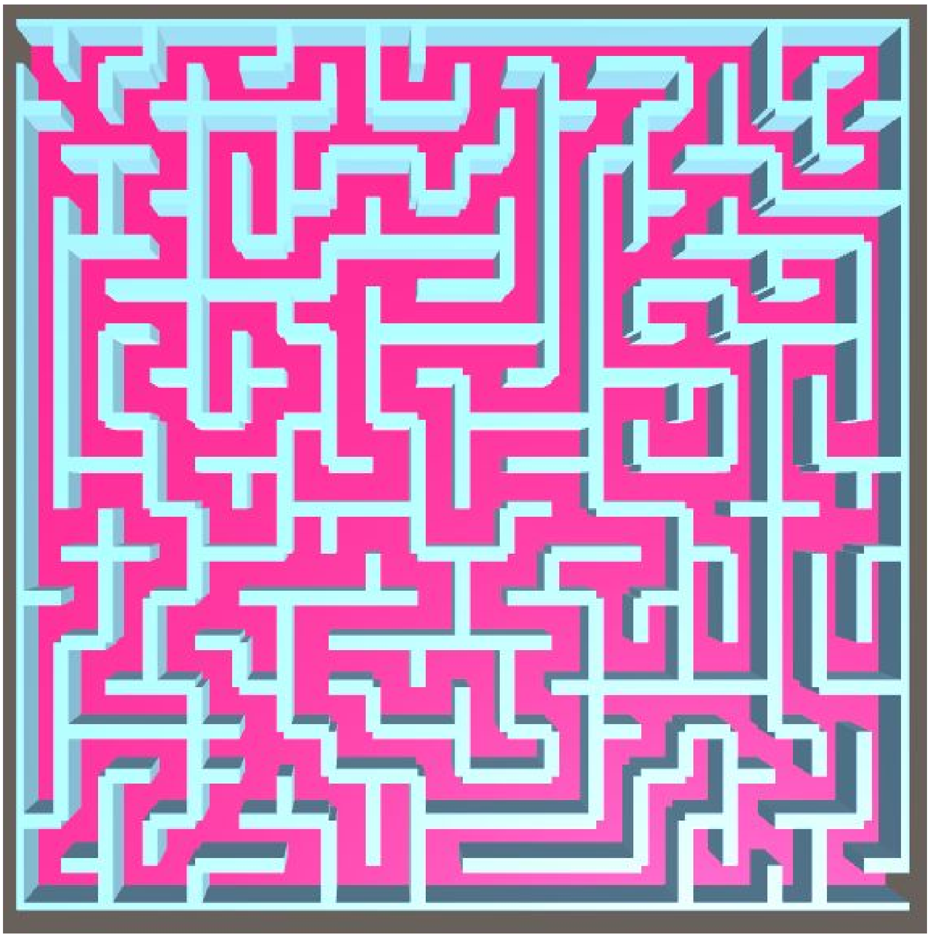


Illustration of Hunt and Kill Algorithm

Pseudocode of this algorithm is showing below:

1. A random cell is selected as a starting point.
2. A random walk is performed to carve passages to the unvisited adjacent cell, repeat until there has no unvisited adjacent cell is possible to travel.
3. Enter “hunt” mode, scan through the grid from top to bottom, left to right. The scanning is looking for an unvisited cell that is adjacent to a visited cell. If found, carve a passage between the two and let the formerly unvisited cell be the new starting location.
4. Repeat steps 2 and 3 of the hunting for the entire grid if the unvisited cell is found, else the algorithm is complete.

The time complexity of hunt and kill algorithm is O(|V|+|E|). The maze generated has no loops and it is a “perfect maze” with only one solution. Its advantage towards the recursive generation technique is to avoid the stack overflow issue. In this project, the maze generated can be specified by the user/player, the example shown below is with size 20x20 from the top view.



Maze Generated using Hunt and Kill Algorithm

```
Complexity Analysis================
                Total Numbers of Grid : 400
                Numbers of Dead Ends : 39
                Numbers of Straightways : 113
                Numbers of Turns : 211
                Numbers of T-junctions : 35
                Numbers of Crossroads : 2
Percentage=================
                Dead Ends : 9.75%
                Straightways : 28.25%
                Turns : 52.75%
                T-junctions : 8.75%
                Crossroads : 0.50%
==========================
```

The Statistic of Maze Complexity

Evaluation of Hunt and Kill Algorithm

The evaluation will be split into two main sessions as the maze generation and the maze solving. In the analysis of maze generation, the percentage of dead-ens, junctions, turns, straightaways and crossroads will be evaluated. Besides, Depth-first Search(DFS) Agent and Breadth-first Search(BFS) Agent is used to solving the maze in order to evaluate the completion time and maze complexity.

I. Maze Generation

The Statistic is computed from the average of 100 generated maze

Types of trait	Percentage	
	Maze 20x20	Maze 50x50
Dead Ends	9.68%	9.48%
Straightways	30.39%	29.94%
Turns	50.41%	51.25%
T-junctions	9.39%	9.17%
Crossroads	0.14%	0.16%

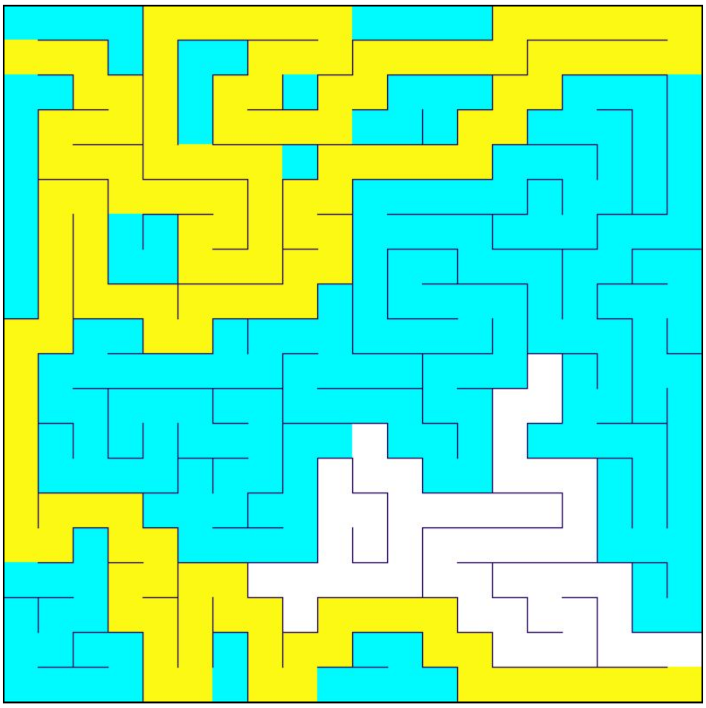
The table above shows that the complexity of the average of 100 generated maze has no noticeable difference. This indicates that the Hunt and Kill Algorithm is performed quite well and stable enough in generating different size of maze. Next session will analysis the difficulty of the maze.

II. Maze Solving

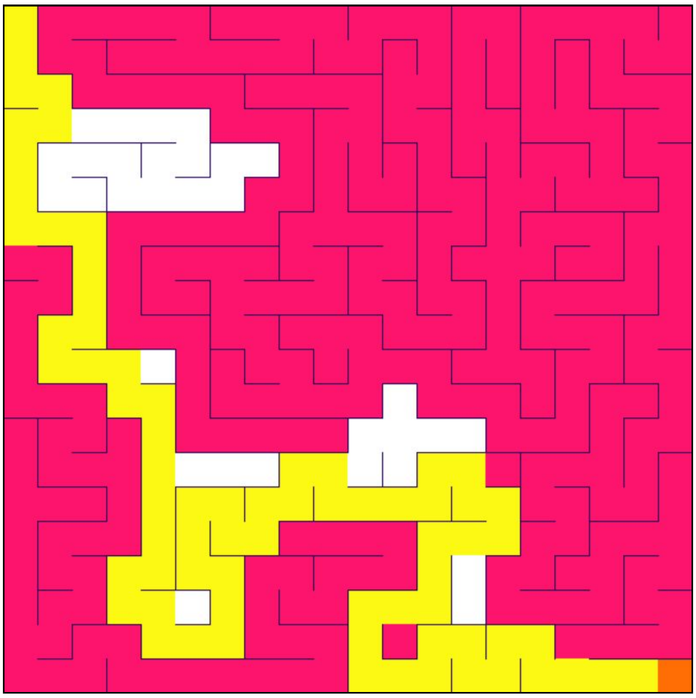
Both BFS and DFS Agent is carrying out the game test 3 times separately with different generated maze by Hunt and Kill algorithm. The number of moves and time of completion will be evaluated.

AI agent	Trials	Number of moves required from the entrance to exit	Time of maze completion(s)	Average of Tiles/Time taken
Breadth-first search (BFS)	1	106 steps	22.10	4.80
	2	72 steps	17.96	4.01
	3	138 steps	24.21	5.70
Depth-first search (DFS)	1	102 steps	20.09	5.08
	2	72 steps	26.82	2.68
	3	190 steps	27.21	6.98

The larger amount of time is required by the AI agents to complete the maze with higher complexity. The following shows the example of path solves by different agents. The maze generated is been converted into 2D and evaluate by the agent.



Example of Maze Solvd by Breadth-first Search Agent
(Attempts shown in cyan, Solution shown in yellow)



Example of Maze Solved by Depth-first Search Agent
(Attempts shown in pink, Solution shown in yellow)

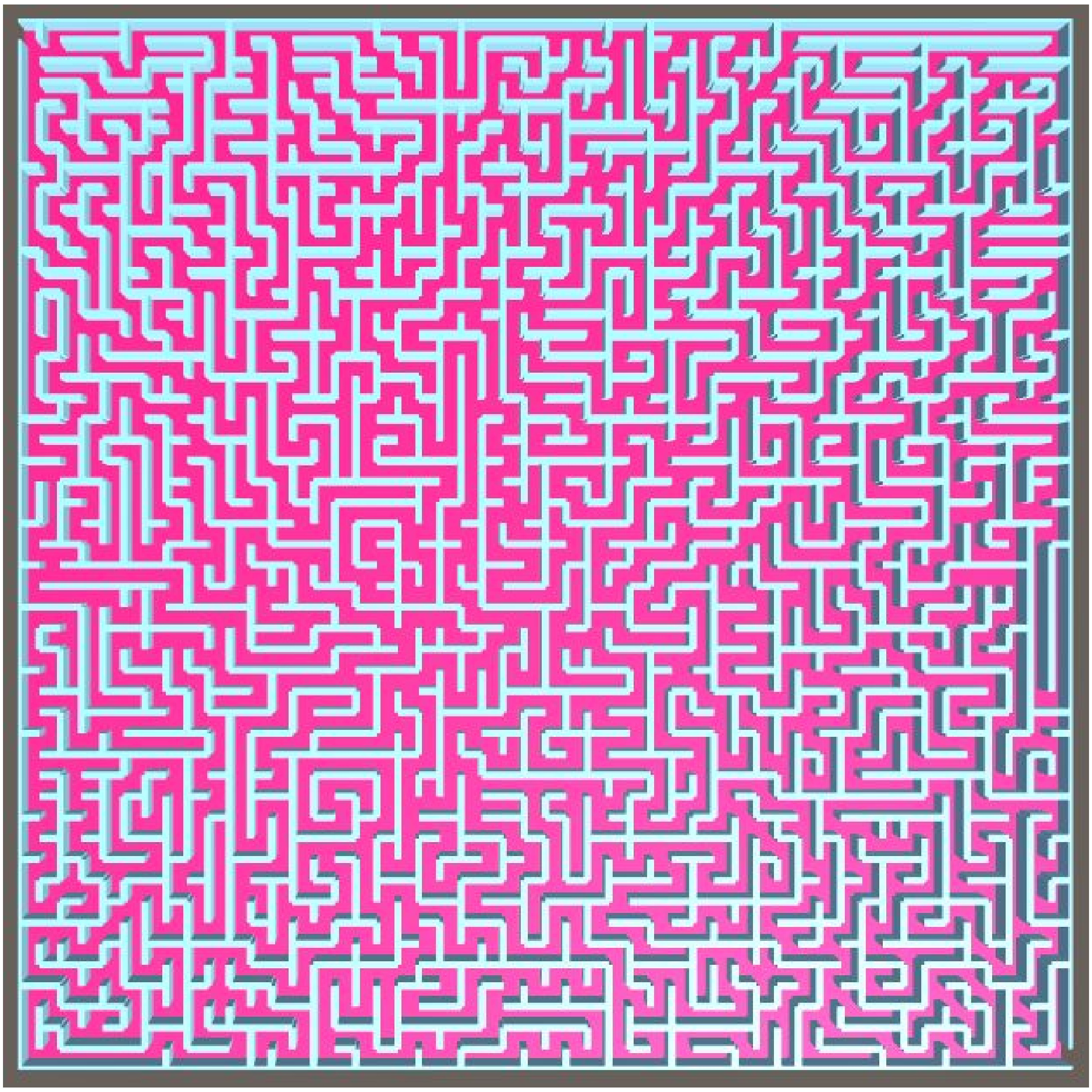
Appendix

```
Complexity Analysis==================
                Total Numbers of Grid : 40000
                Numbers of Dead Ends : 3870
                Numbers of Straightways : 12155
                Numbers of Turns : 20162
                Numbers of T-junctions : 3756
                Numbers of Crossroads : 57
Percentage========================
                Dead Ends : 9.68%
                Straightways : 30.39%
                Turns : 50.41%
                T-junctions : 9.39%
                Crossroads : 0.14%
=================================
```

100 Generations of 20x20 Maze

```
Complexity Analysis==================
                Total Numbers of Grid : 250000
                Numbers of Dead Ends : 23711
                Numbers of Straightways : 74841
                Numbers of Turns : 128131
                Numbers of T-junctions : 22923
                Numbers of Crossroads : 394
Percentage========================
                Dead Ends : 9.48%
                Straightways : 29.94%
                Turns : 51.25%
                T-junctions : 9.17%
                Crossroads : 0.16%
=================================
```

100 Generations of 50x50 Maze



Example of 50x50 Maze