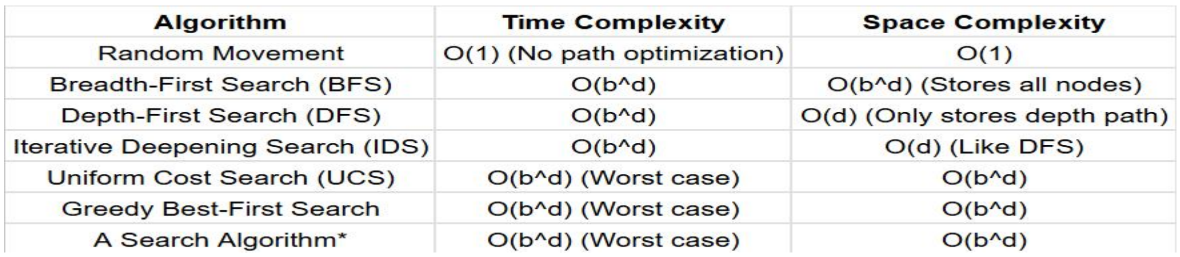
Search Algorithms in AI Snake Game

Introduction

This report analyzes the implementation of various search algorithms used in an AI-controlled Snake game. The game utilizes pathfinding techniques to navigate obstacles and reach the food. The following algorithms are implemented:

1. Random Movement Algorithm
2. Breadth-First Search (BFS)
3. Depth-First Search (DFS)
4. Iterative Deepening Search (IDS)
5. Uniform Cost Search (UCS)
6. Greedy Best-First Search
7. A* Search Algorithm

Time and Space complexity

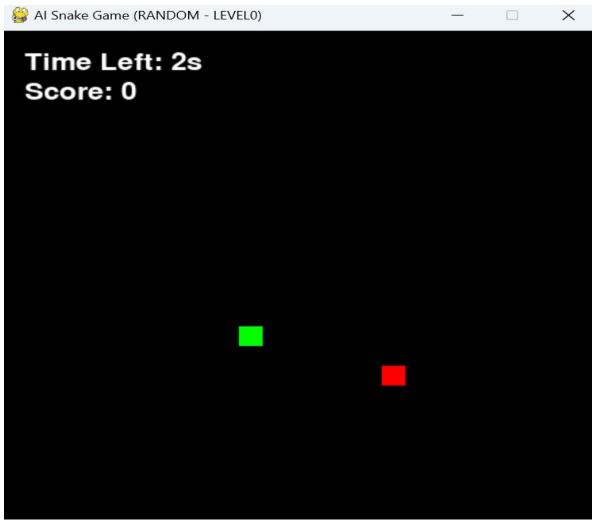


Algorithm	Time Complexity	Space Complexity
Random Movement	O(1) (No path optimization)	O(1)
Breadth-First Search (BFS)	O(b^d)	O(b^d) (Stores all nodes)
Depth-First Search (DFS)	O(b^d)	O(d) (Only stores depth path)
Iterative Deepening Search (IDS)	O(b^d)	O(d) (Like DFS)
Uniform Cost Search (UCS)	O(b^d) (Worst case)	O(b^d)
Greedy Best-First Search	O(b^d) (Worst case)	O(b^d)
A Search Algorithm*	O(b^d) (Worst case)	O(b^d)

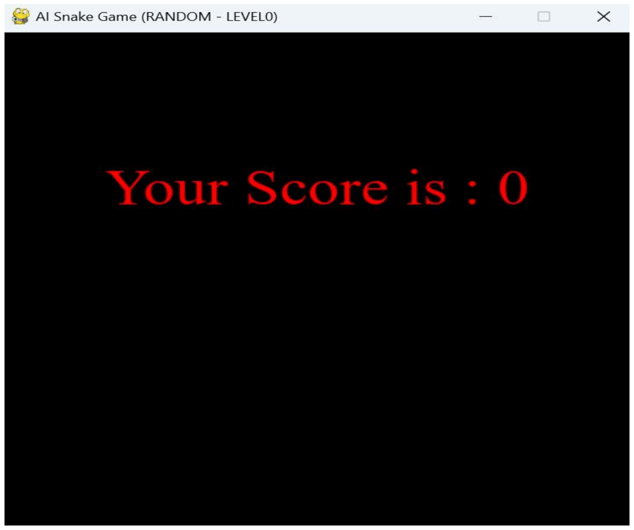
Implemented Search Algorithms

1. Random Movement Algorithm

Random Search Algorithm in AI is a simple optimization technique that explores the search space by randomly selecting solutions and evaluating their performance. It does not use gradients or systematic exploration, making it useful for non-differentiable, noisy, or complex functions.

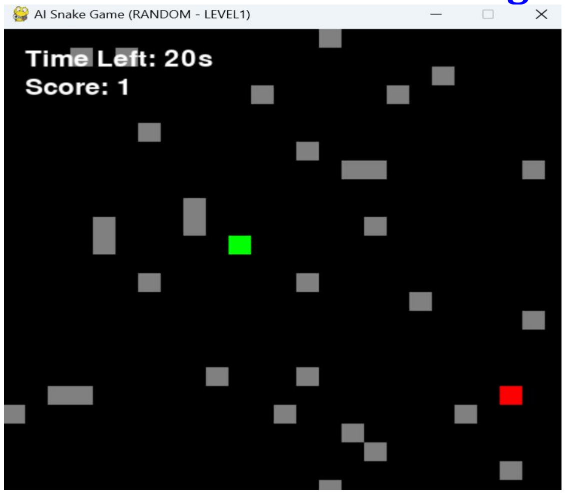


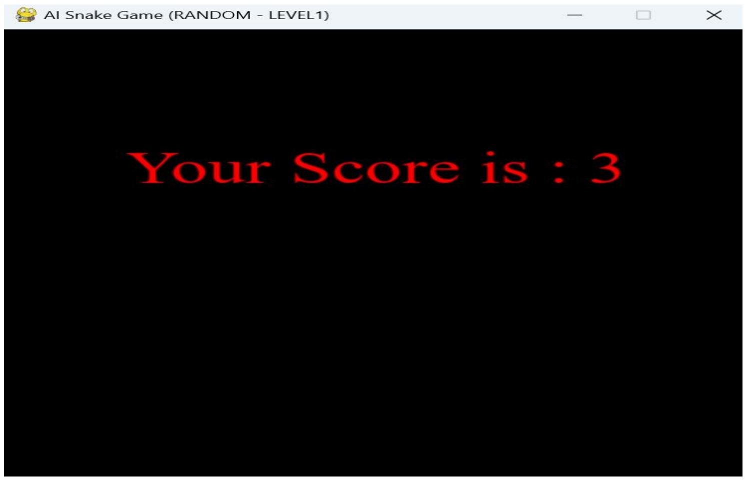
Time Left: 2s
Score: 0

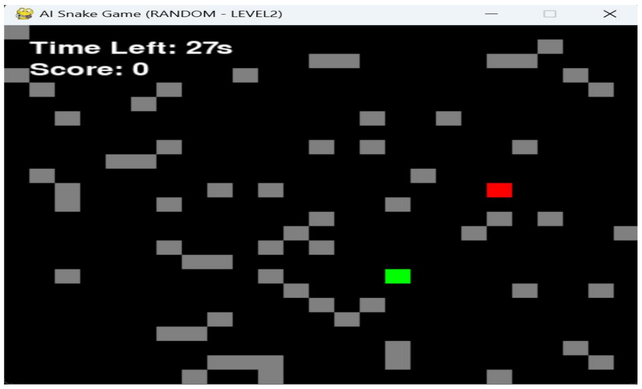


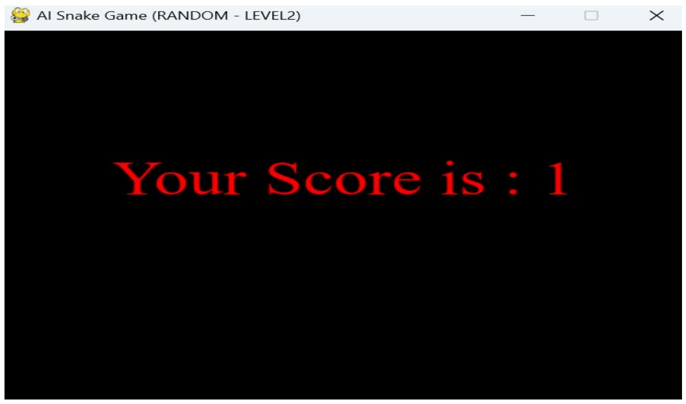
Your Score is : 0

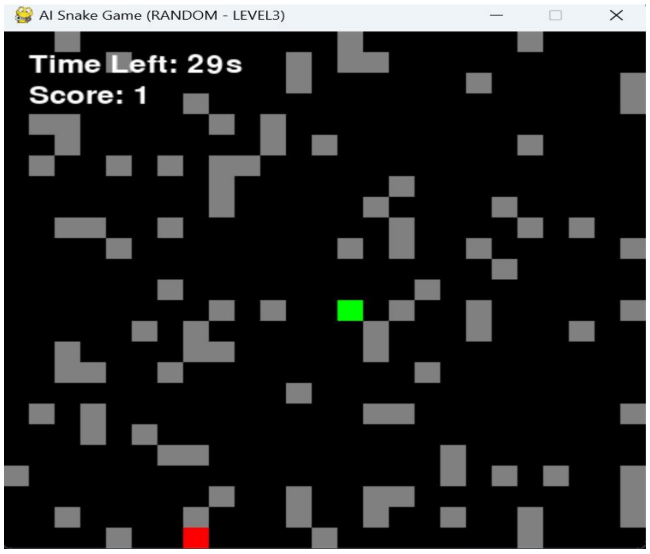
Search Algorithms in AI Snake Game







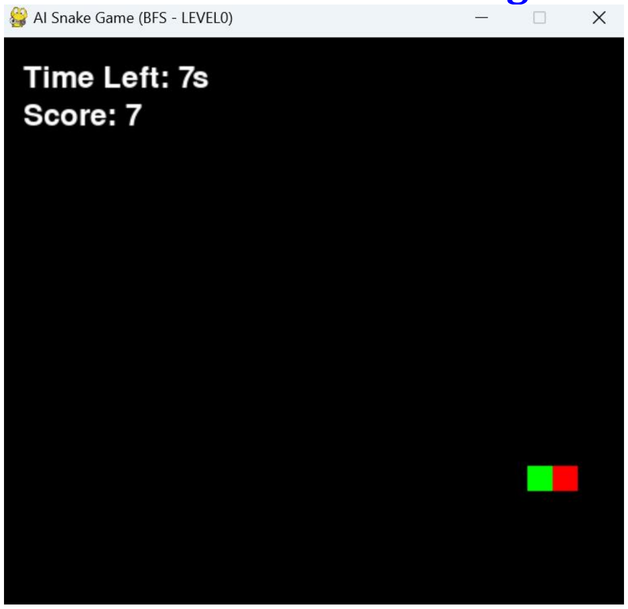




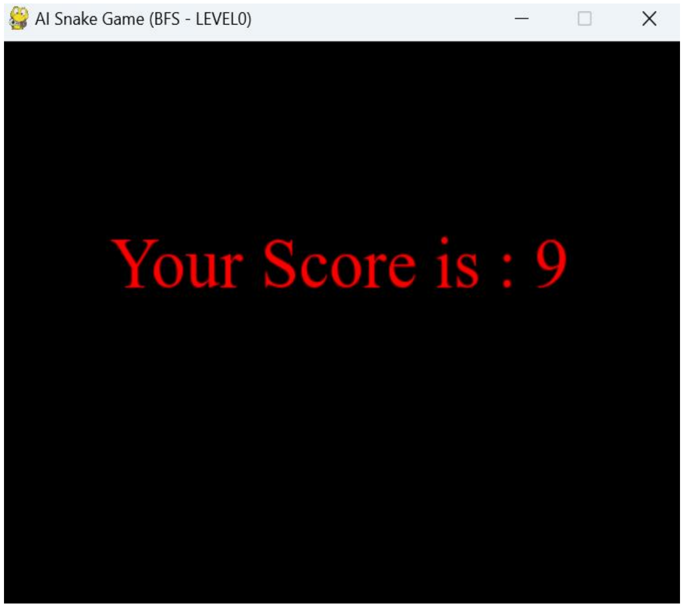
2. Breadth-First Search (BFS)

BFS (Breadth-First Search) is a graph traversal algorithm that explores all neighbors of a node before moving to the next level. It uses a queue (FIFO) to track nodes to visit.

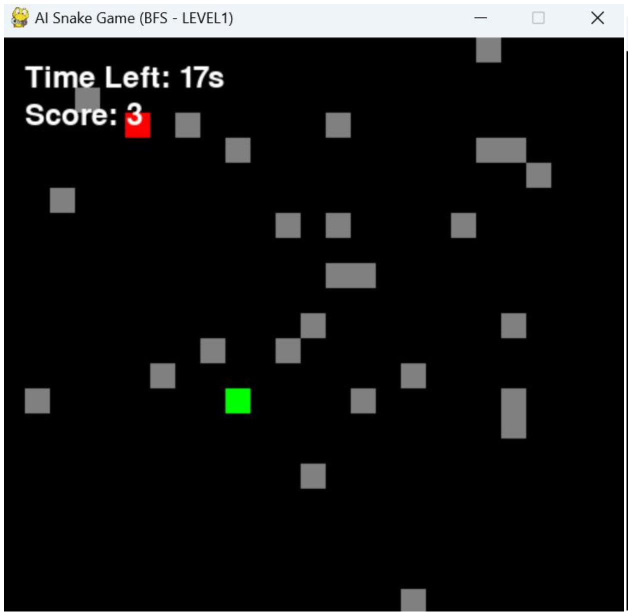
Search Algorithms in AI Snake Game


AI Snake Game (BFS - LEVEL0)

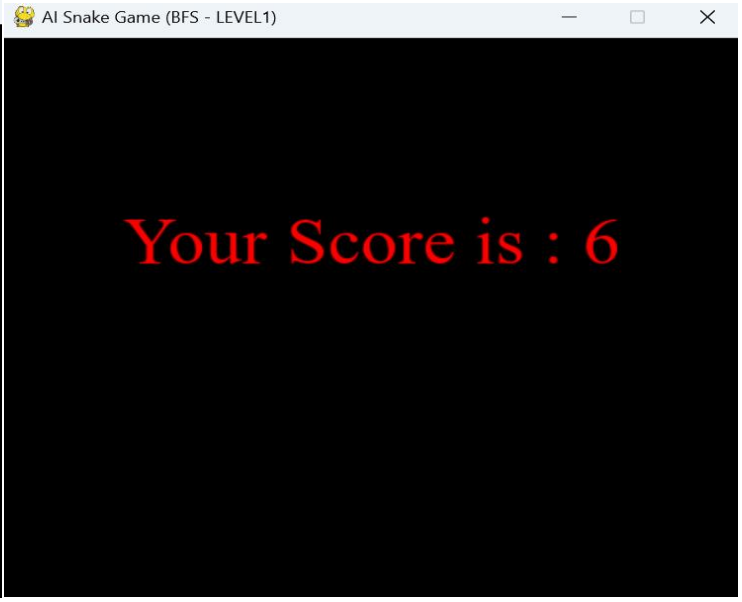
Time Left: 7s
Score: 7


AI Snake Game (BFS - LEVEL0)

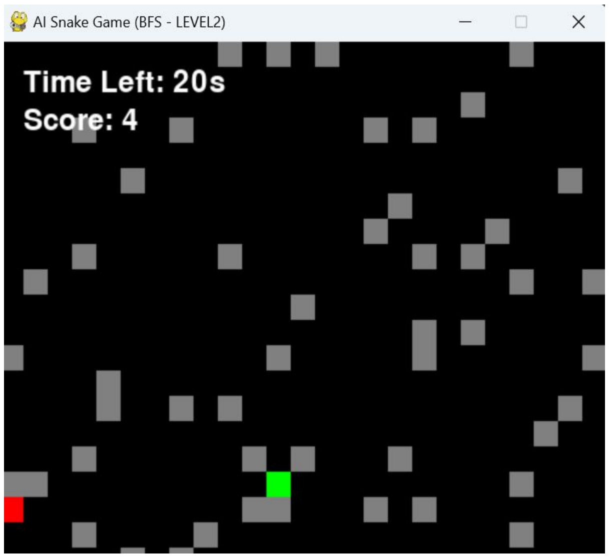
Your Score is : 9


AI Snake Game (BFS - LEVEL1)

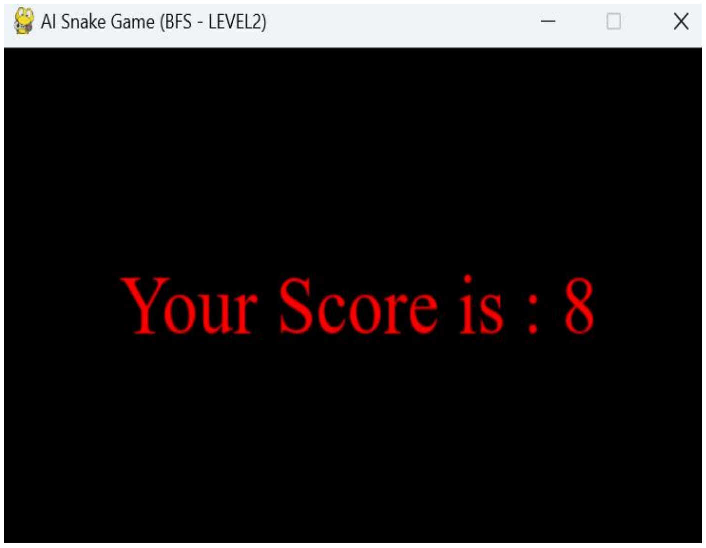
Time Left: 17s
Score: 3


AI Snake Game (BFS - LEVEL1)

Your Score is : 6

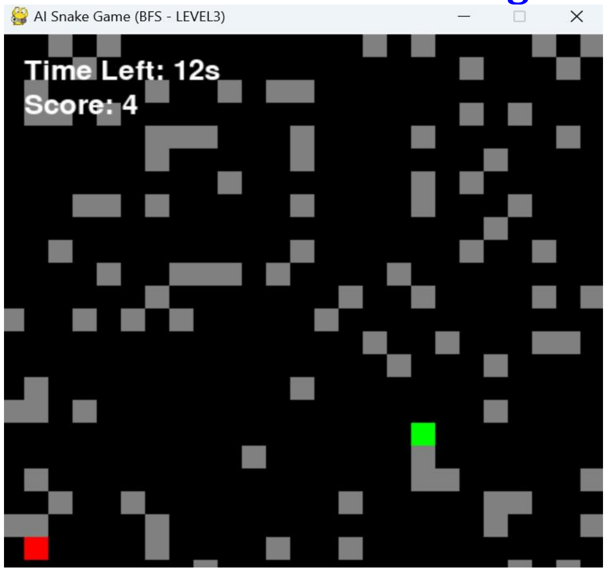

AI Snake Game (BFS - LEVEL2)

Time Left: 20s
Score: 4

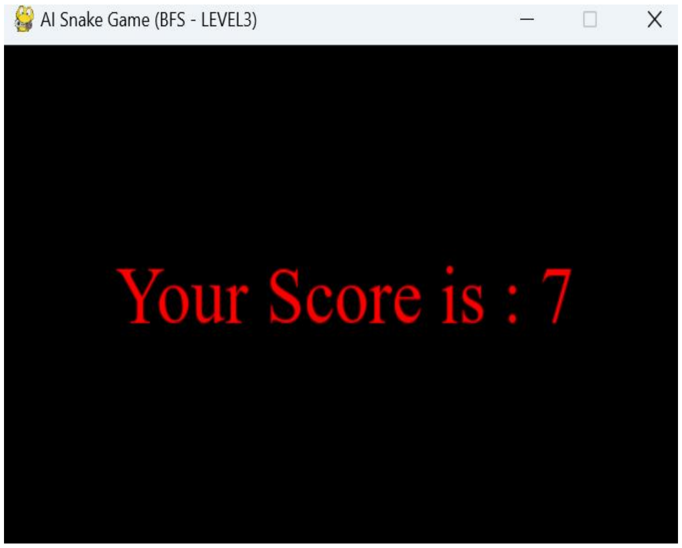

AI Snake Game (BFS - LEVEL2)

Your Score is : 8

Search Algorithms in AI Snake Game


AI Snake Game (BFS - LEVEL3)

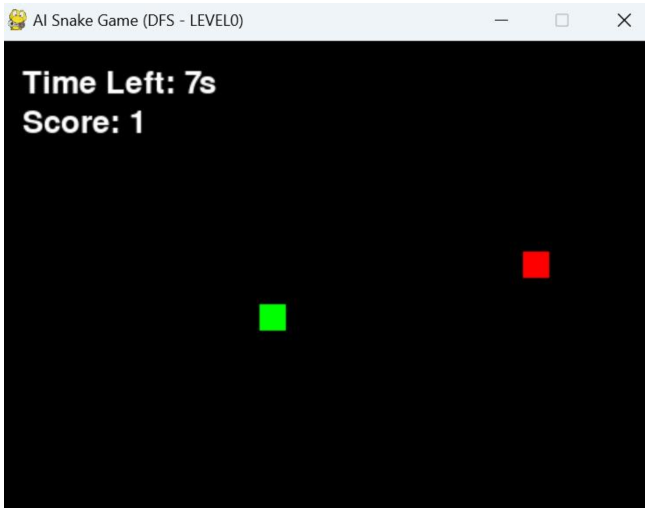
Time Left: 12s
Score: 4


AI Snake Game (BFS - LEVEL3)

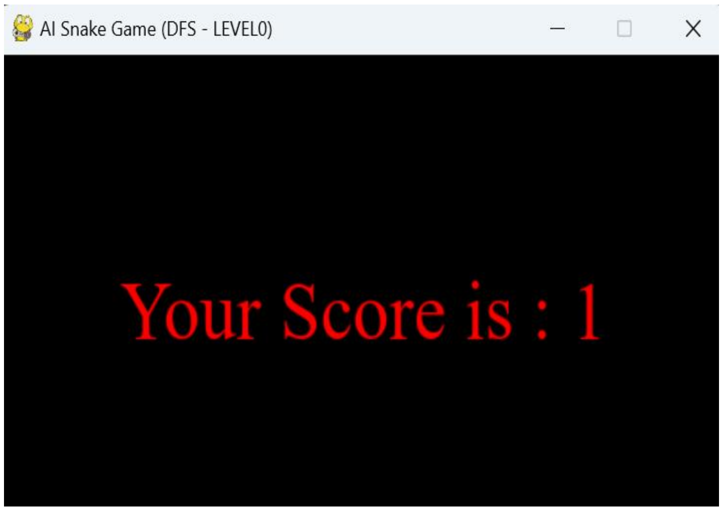
Your Score is : 7

3. Depth-First Search (DFS)

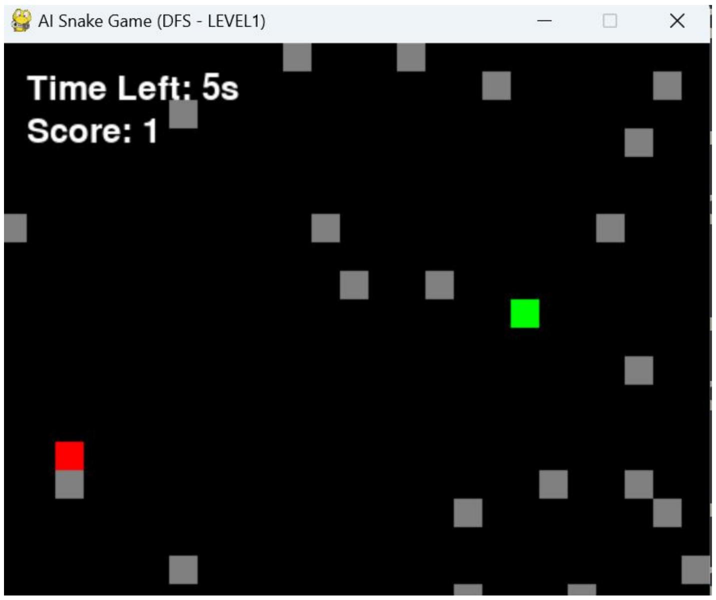
DFS (Depth-First Search) is a graph traversal algorithm that explores as far as possible along a branch before backtracking. It uses a stack (explicit or recursion) to track nodes.


AI Snake Game (DFS - LEVEL0)

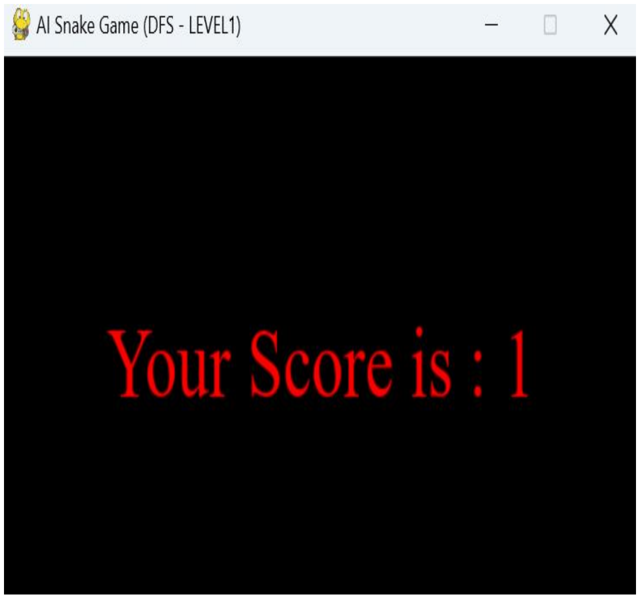
Time Left: 7s
Score: 1


AI Snake Game (DFS - LEVEL0)

Your Score is : 1

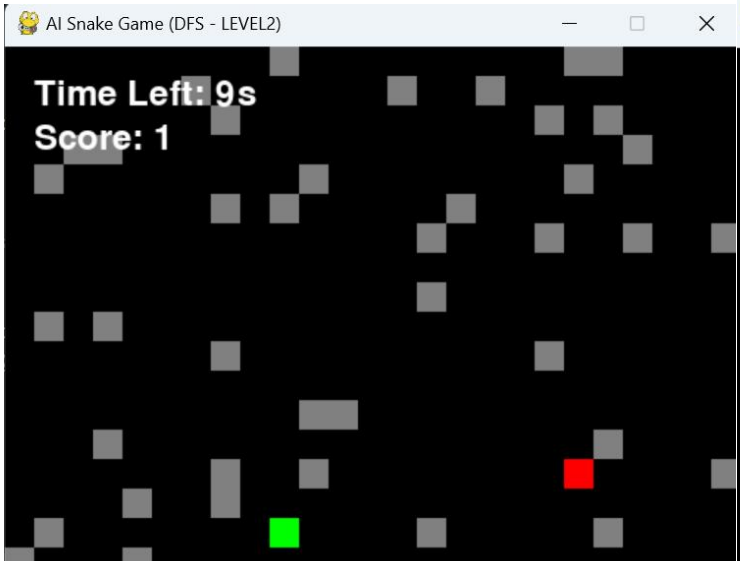

AI Snake Game (DFS - LEVEL1)

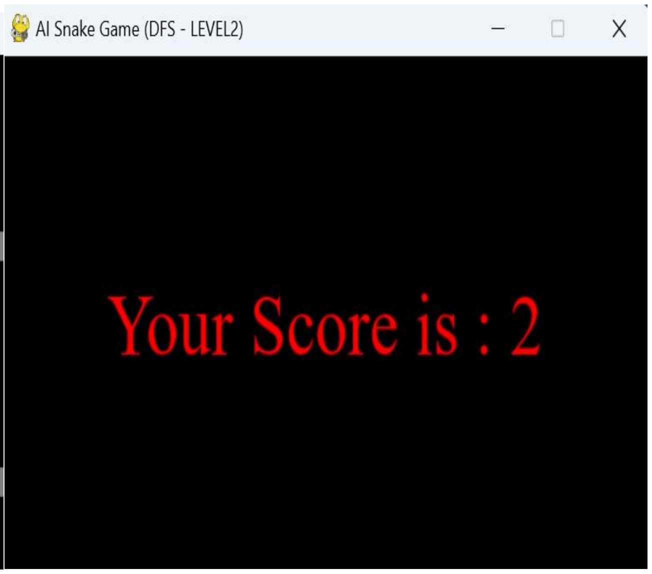
Time Left: 5s
Score: 1

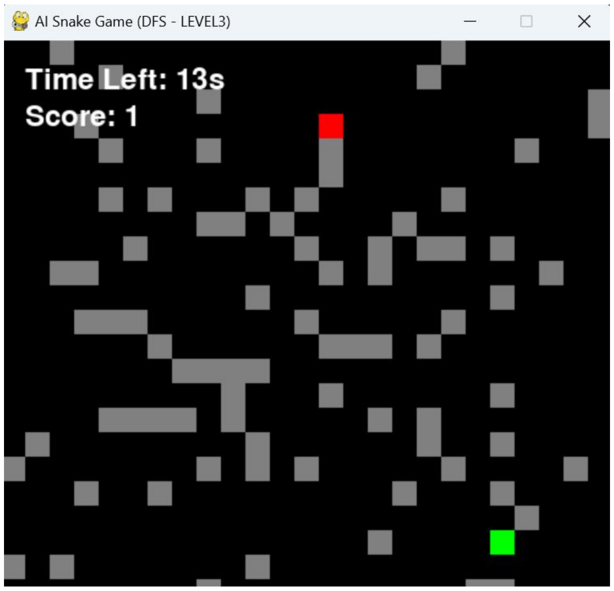

AI Snake Game (DFS - LEVEL1)

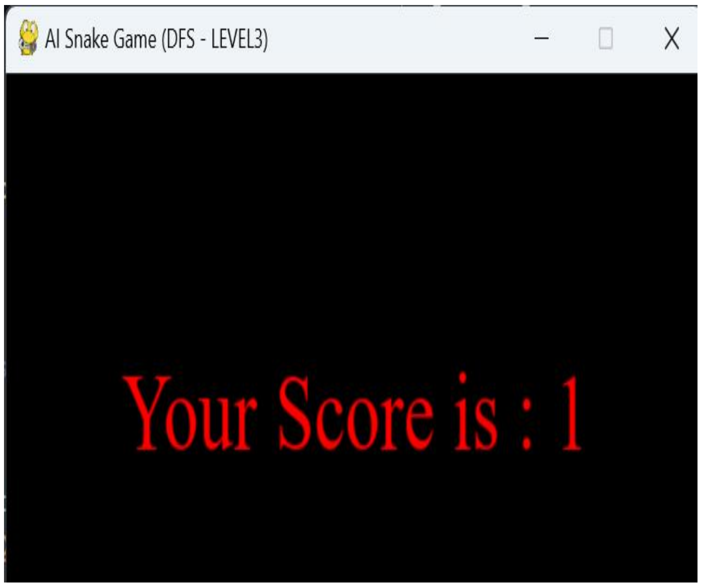
Your Score is : 1

Search Algorithms in AI Snake Game



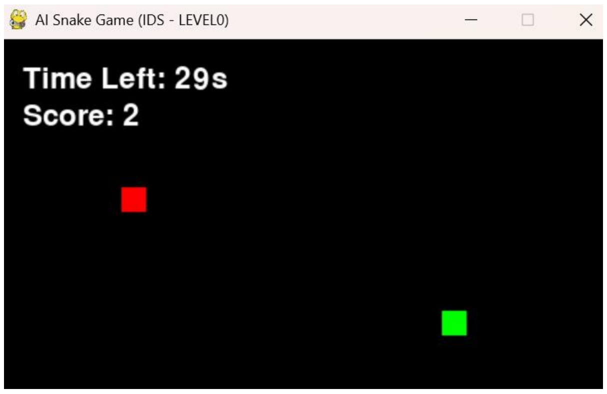


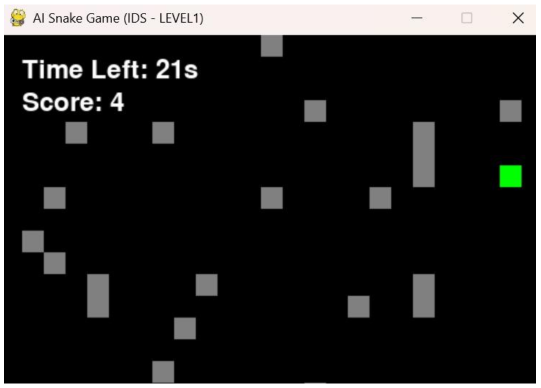


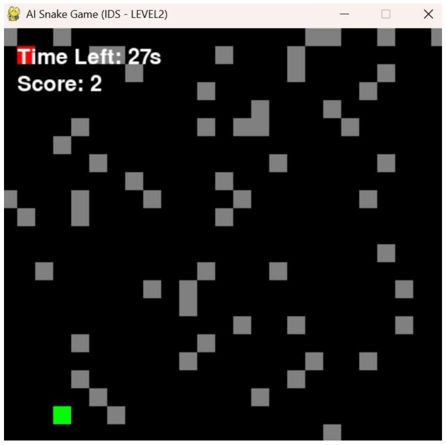


4. Iterative Deepening Search (IDS)

Iterative Deepening Search (IDS) is a graph traversal algorithm that combines Depth-First Search (DFS) and Breadth-First Search (BFS) to find the shortest path in an uninformed search. It repeatedly runs DFS with increasing depth limits until the goal is found.



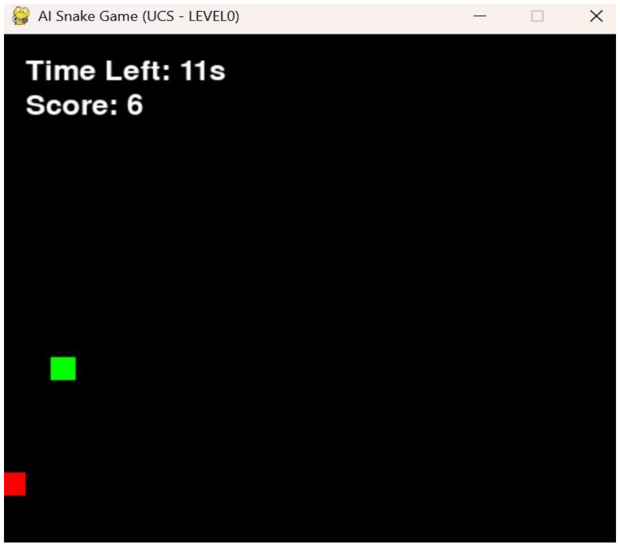


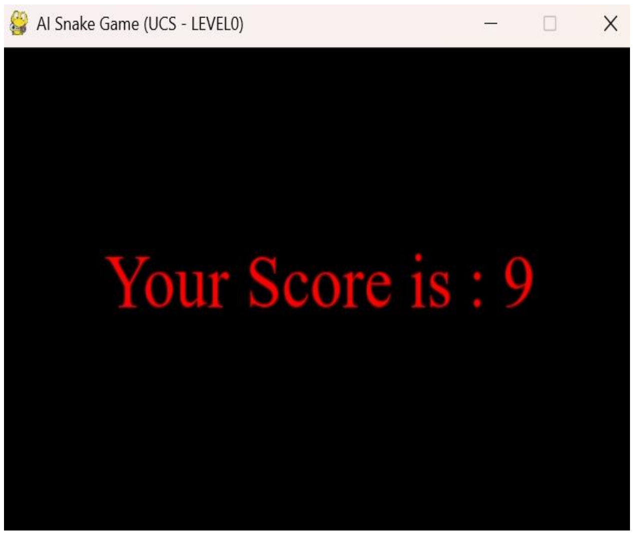


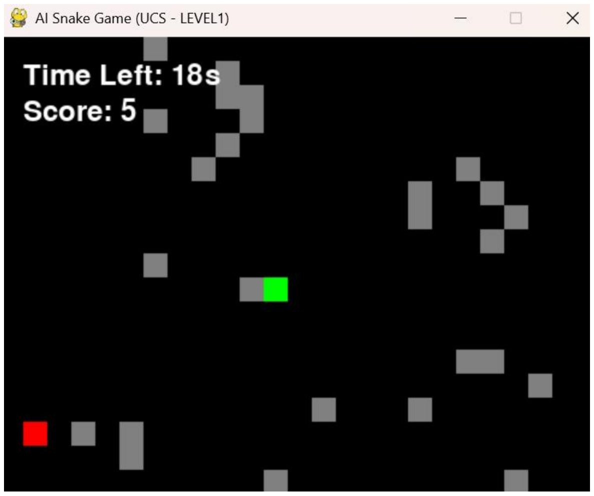
THE IDS CODE FOR SCORE IS NOT RESPONDING

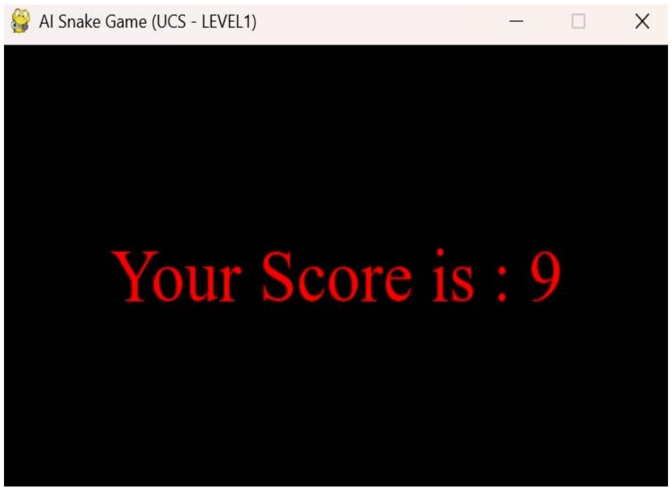
5. Uniform Cost Search (UCS)

Uniform Cost Search (UCS) is an uninformed search algorithm used to find the least-cost path in a weighted graph. It is similar to Dijkstra's algorithm and operates using a priority queue.

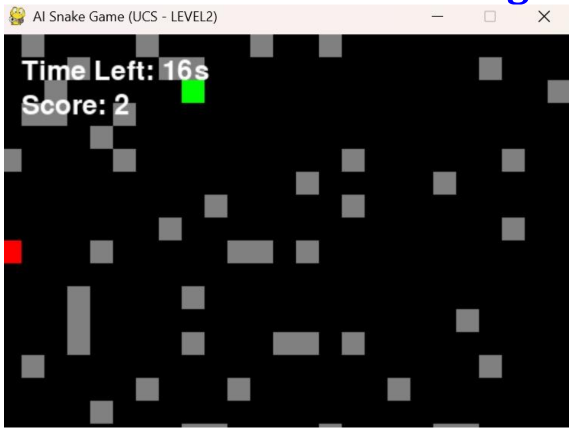




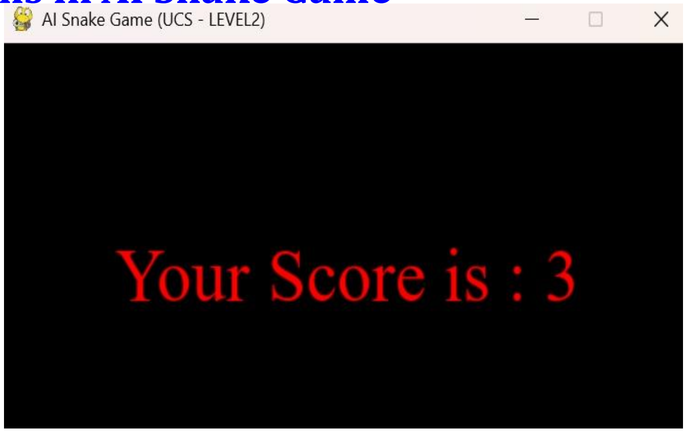




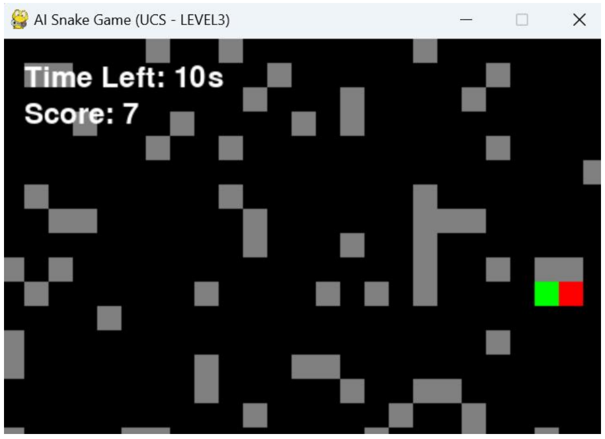
Search Algorithms in AI Snake Game


AI Snake Game (UCS - LEVEL2)

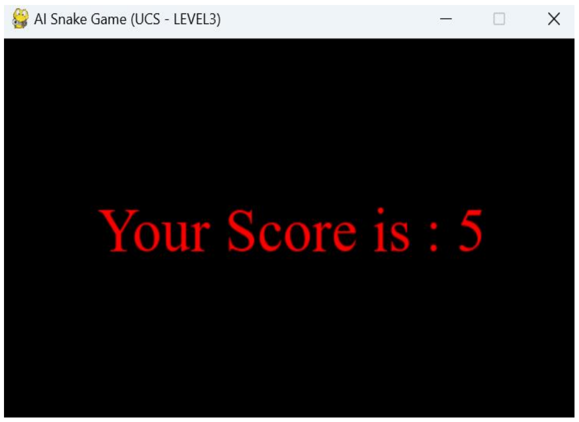
Time Left: 16s
Score: 2


AI Snake Game (UCS - LEVEL2)

Your Score is : 3


AI Snake Game (UCS - LEVEL3)

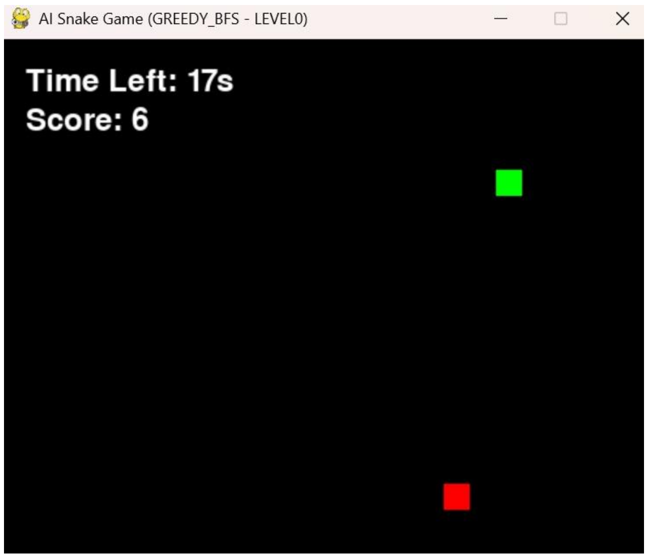
Time Left: 10s
Score: 7


AI Snake Game (UCS - LEVEL3)

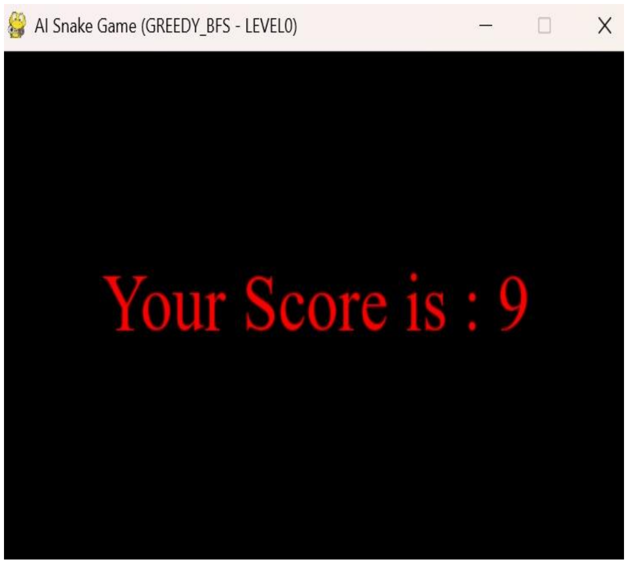
Your Score is : 5

6. Greedy Best-First Search

Greedy Best-First Search (GBFS) is an informed search algorithm that expands the node that appears closest to the goal, using a heuristic function $h(n)$. It does not consider the cost from the start node, only the estimated cost to the goal.

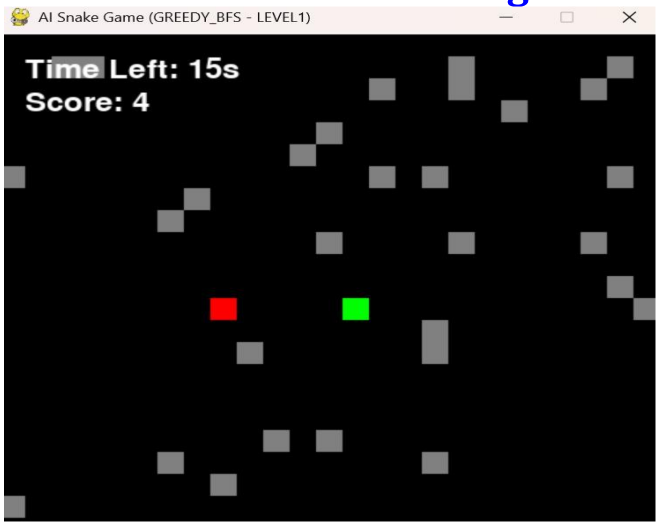

AI Snake Game (GREEDY_BFS - LEVEL0)

Time Left: 17s
Score: 6

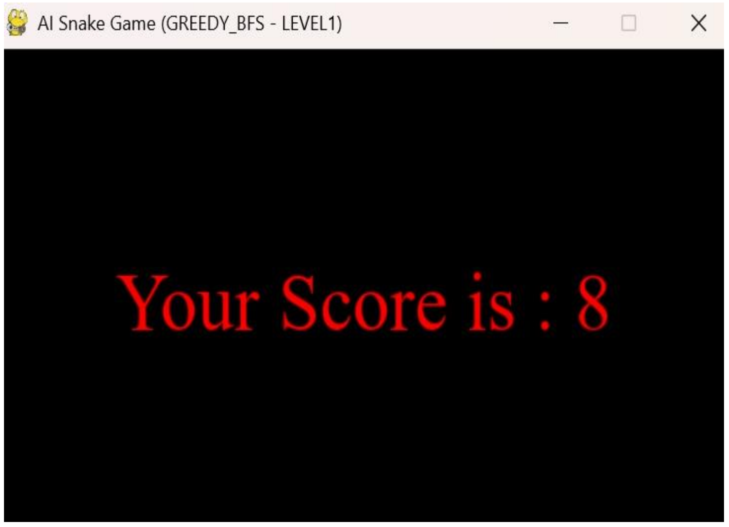

AI Snake Game (GREEDY_BFS - LEVEL0)

Your Score is : 9

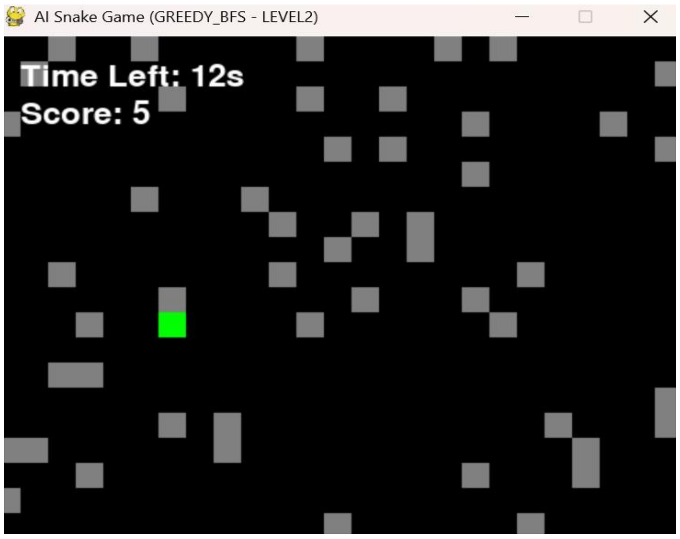
Search Algorithms in AI Snake Game


AI Snake Game (GREEDY_BFS - LEVEL1)

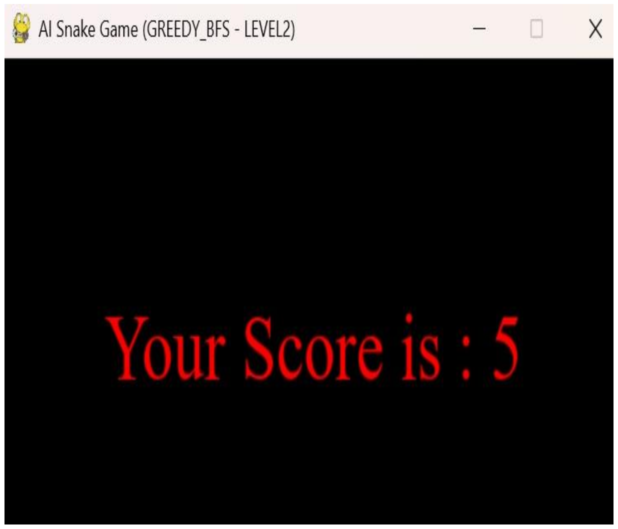
Time Left: 15s
Score: 4


AI Snake Game (GREEDY_BFS - LEVEL1)

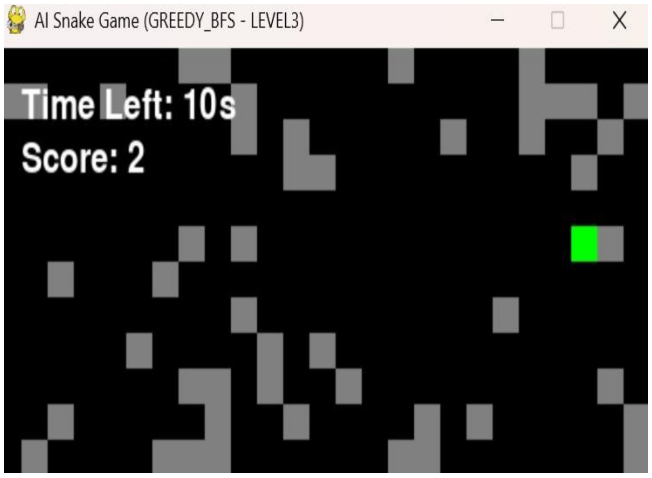
Your Score is : 8


AI Snake Game (GREEDY_BFS - LEVEL2)

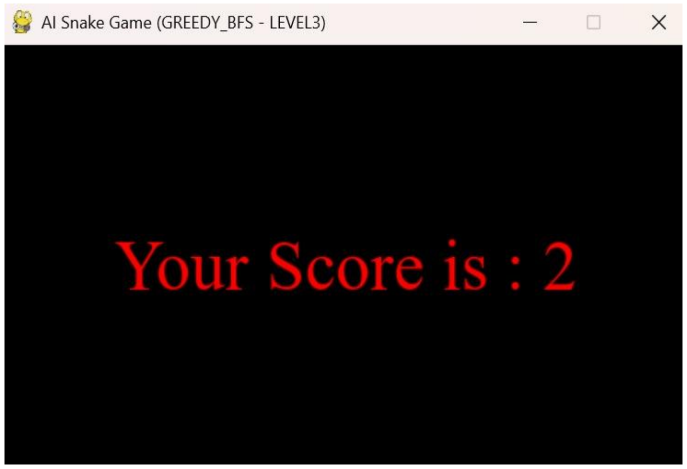
Time Left: 12s
Score: 5


AI Snake Game (GREEDY_BFS - LEVEL2)

Your Score is : 5


AI Snake Game (GREEDY_BFS - LEVEL3)

Time Left: 10s
Score: 2

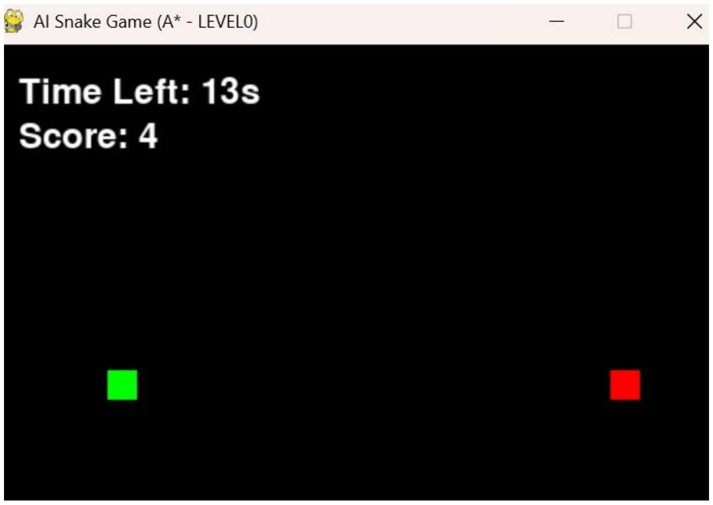

AI Snake Game (GREEDY_BFS - LEVEL3)

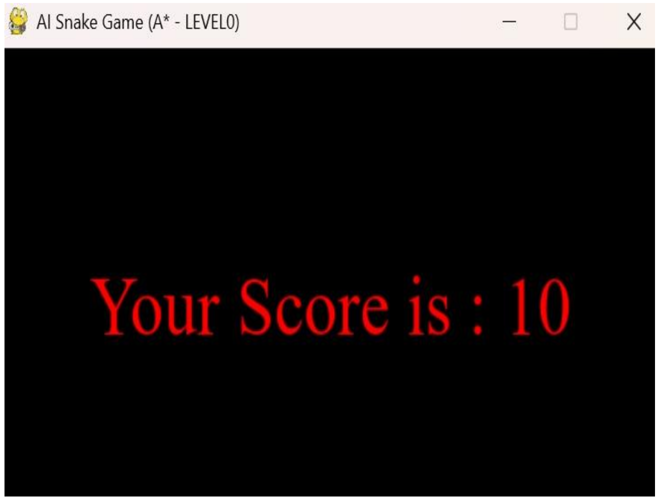
Your Score is : 2

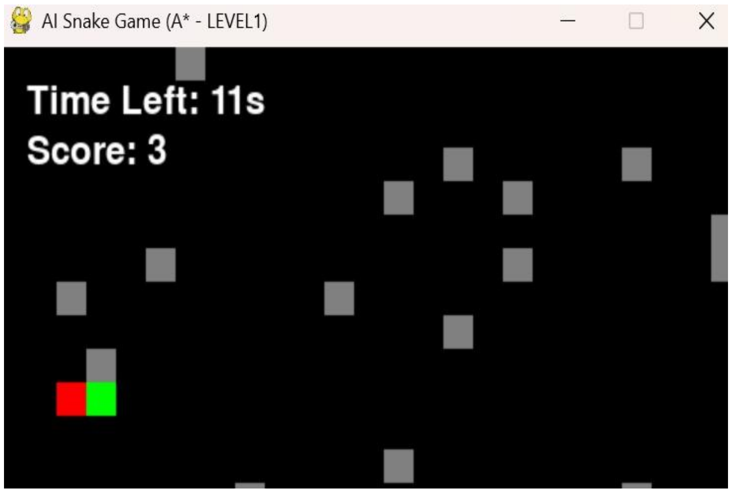
Search Algorithms in AI Snake Game

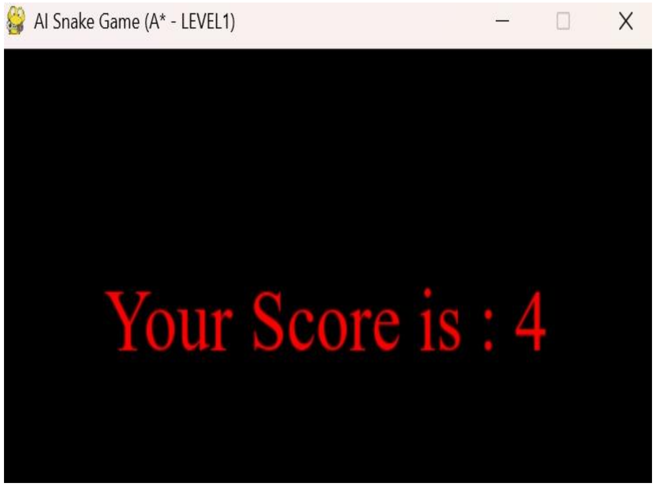
7. A* Search Algorithm

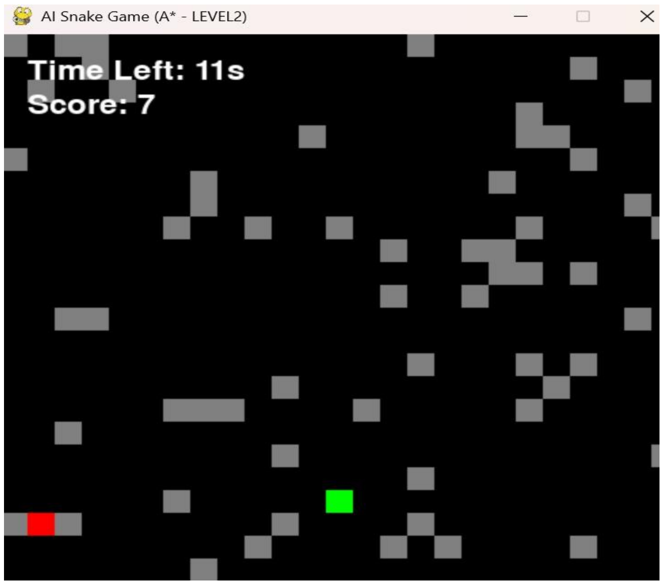
a* is an informed search algorithm that finds the shortest path in a weighted graph using both path cost and heuristics. It balances Uniform Cost Search (UCS) and Greedy Best-First Search (GBFS) for optimal and efficient pathfinding.

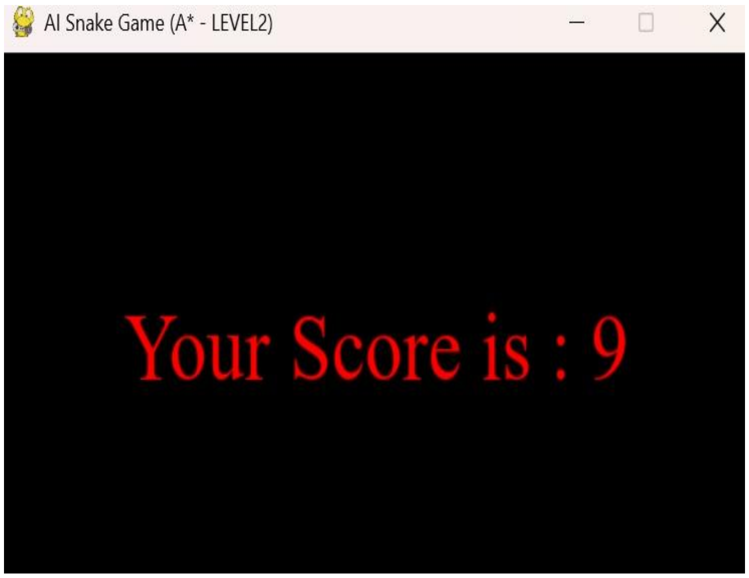




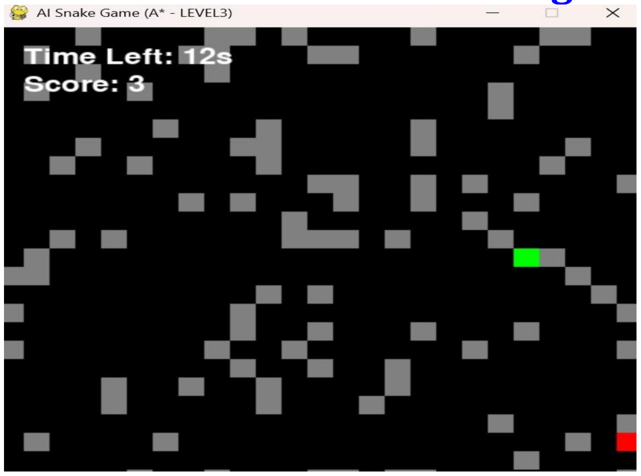


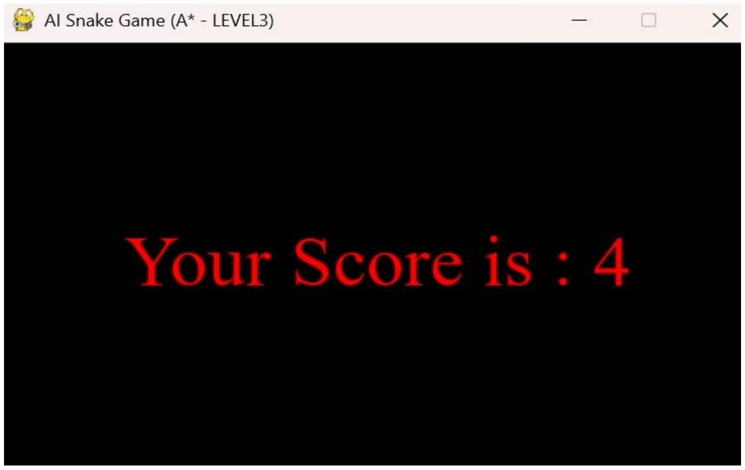






Search Algorithms in AI Snake Game





Results and Observations

Algorithm	Level 0	Level 1	Level 2	Level 3
Random	0	3	1	1
BFS	9	6	8	7
DFS	1	1	2	1
IDS (Not responding)	-	-	-	-
UCS	9	9	3	5
Greedy BFS	9	8	5	2
A*	10	4	9	4

Conclusion

Comparison of Search Algorithms

From the given results, we can analyze the performance of different search algorithms at each level based on their effectiveness in finding solutions.

Random Search performs poorly, with inconsistent results across levels, highlighting its inefficiency in systematic search.

BFS (Breadth-First Search) maintains relatively high performance, especially at Level 0 (9) and Level 2 (8), indicating its reliability in exploring all possible paths evenly. However, its performance fluctuates across levels.

DFS (Depth-First Search) has low values across all levels, showing its inefficiency in finding optimal solutions, as it can get stuck in deep, unfruitful paths.

IDS (Iterative Deepening Search) is not responding, possibly due to high computational requirements or inefficient implementation for deeper levels.

UCS (Uniform Cost Search) performs consistently well in Level 0 (9) and Level 1 (9) but drops significantly at deeper levels, suggesting that cost-based expansion is useful in early stages but may struggle in complex scenarios.

Search Algorithms in AI Snake Game

Greedy BFS shows strong performance in early levels (9, 8) but degrades in deeper levels, indicating that heuristic-driven search is efficient but can lead to suboptimal paths due to local optima.

A* performs best overall, excelling in Level 0 (10) and Level 2 (9), indicating that the combination of heuristic and path cost leads to optimal and efficient search. However, its performance is lower at Levels 1 and 3, possibly due to heuristic variations or search space complexity.

Key Takeaways:

A* is the most effective algorithm overall, balancing cost and heuristic efficiently.

BFS and UCS perform well in structured search but can be inefficient in large or deep search spaces.

Greedy BFS is efficient but can be misled by heuristics.

DFS and Random Search are the least effective due to their lack of systematic exploration and optimization.

IDS did not perform, indicating issues with scalability or execution in this case.

For optimal search, A* is recommended, followed by BFS or UCS depending on the problem constraints.

Presented By:

Sandeep Kumar Sah

22054082

CSE-38

Sandip Kumar Sah

22054083

CSE-38

SUBMITTED TO:

VIJAY KUMAR MEENA

FCS

THANKS…