AI 2002 – Artificial Intelligence

Assignment 1: Uninformed Search in a Grid Environment

Student Name: Shahzad Ahmed
Roll No: 22F3098

1. Introduction

In this assignment, I implemented six uninformed search algorithms in a 20x20 grid environment using Python and Pygame. The system visualizes how each algorithm explores nodes step-by-step from the Start node to the Target node. The environment contains static walls and dynamic obstacles that may appear randomly during execution.

2. System Implementation

Each cell in the grid represents a node. Colors are used to represent different states: Green for Start, Red for Target, Blue for Frontier nodes, Yellow for Explored nodes, Purple for Final Path, and Black for Obstacles. The movement order follows a clockwise direction including diagonal moves. A small probability is used to generate dynamic obstacles during runtime.

3. Algorithms Explanation

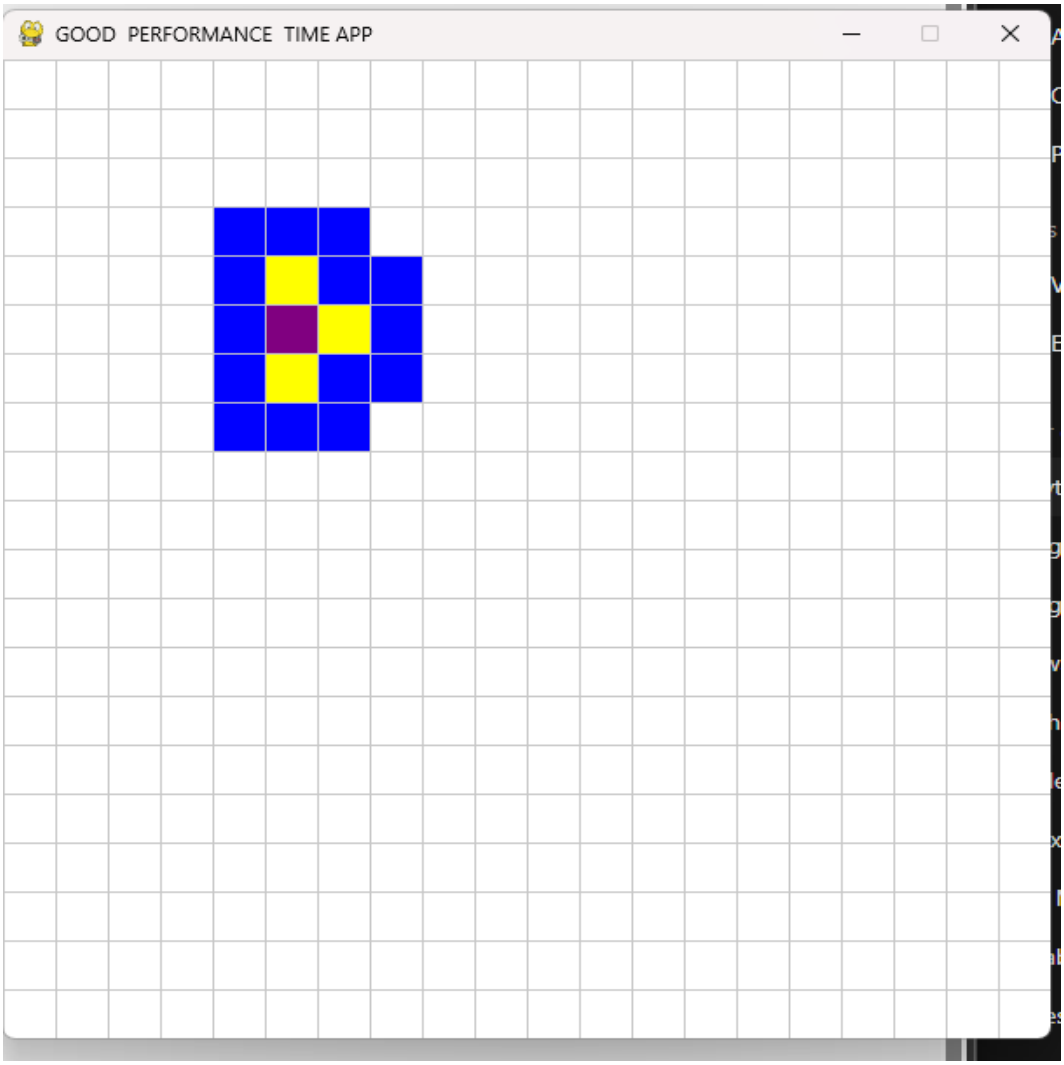
Breadth-First Search (BFS): Explores level by level using a queue and guarantees shortest path. Depth-First Search (DFS): Explores deeply using a stack and may not find shortest path. Uniform Cost Search (UCS): Expands node with lowest cost first using priority queue. Depth-Limited Search (DLS): DFS with a fixed depth limit. Iterative Deepening DFS (IDDFS): Repeatedly applies DLS with increasing depth. Bidirectional Search: Runs search from both start and goal until they meet.

4. Pros and Cons

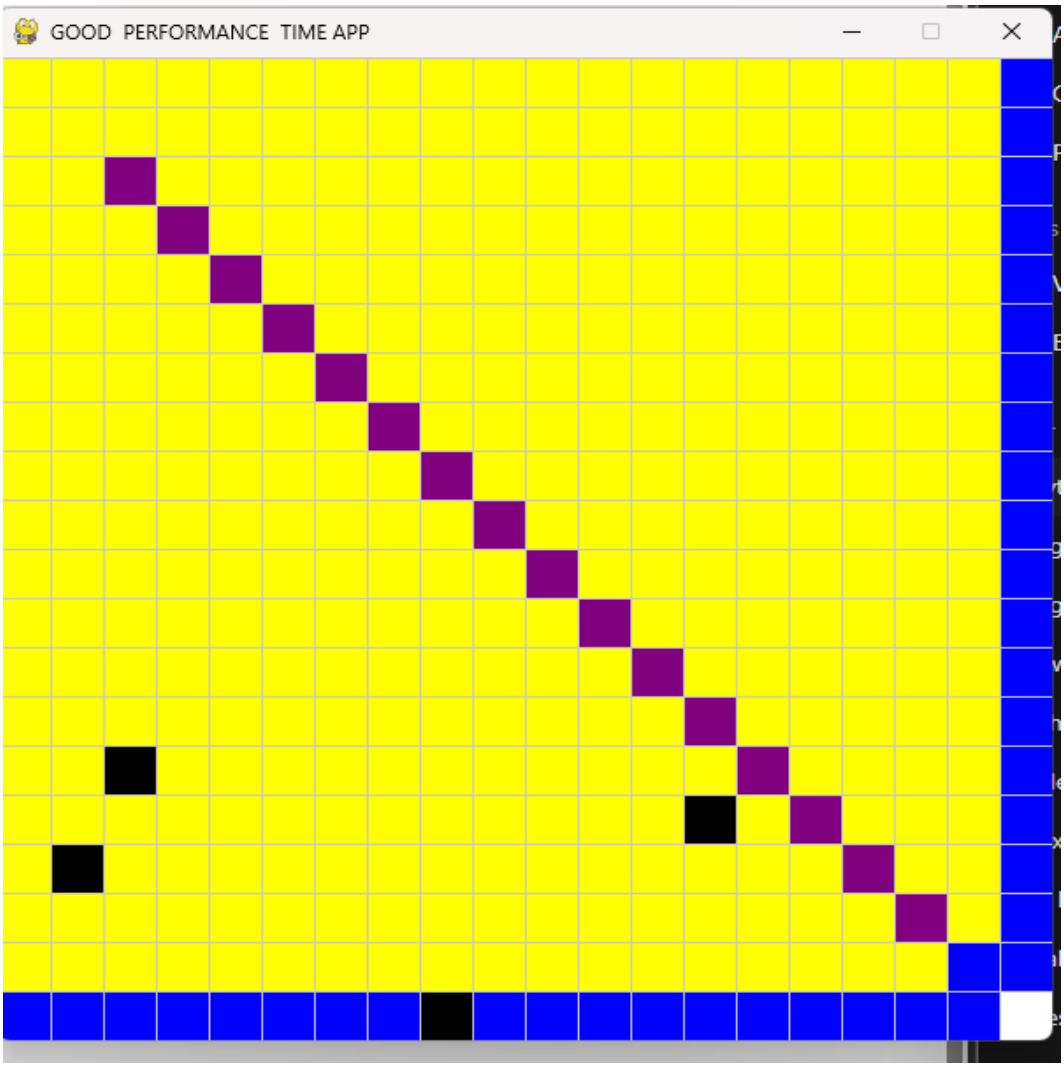
BFS: Complete and optimal but high memory usage. DFS: Low memory but not optimal. UCS: Optimal but slower. DLS: Prevents infinite depth but may miss solution. IDDFS: Memory efficient but repeats work. Bidirectional: Faster in many cases but complex implementation.

5. Experimental Results

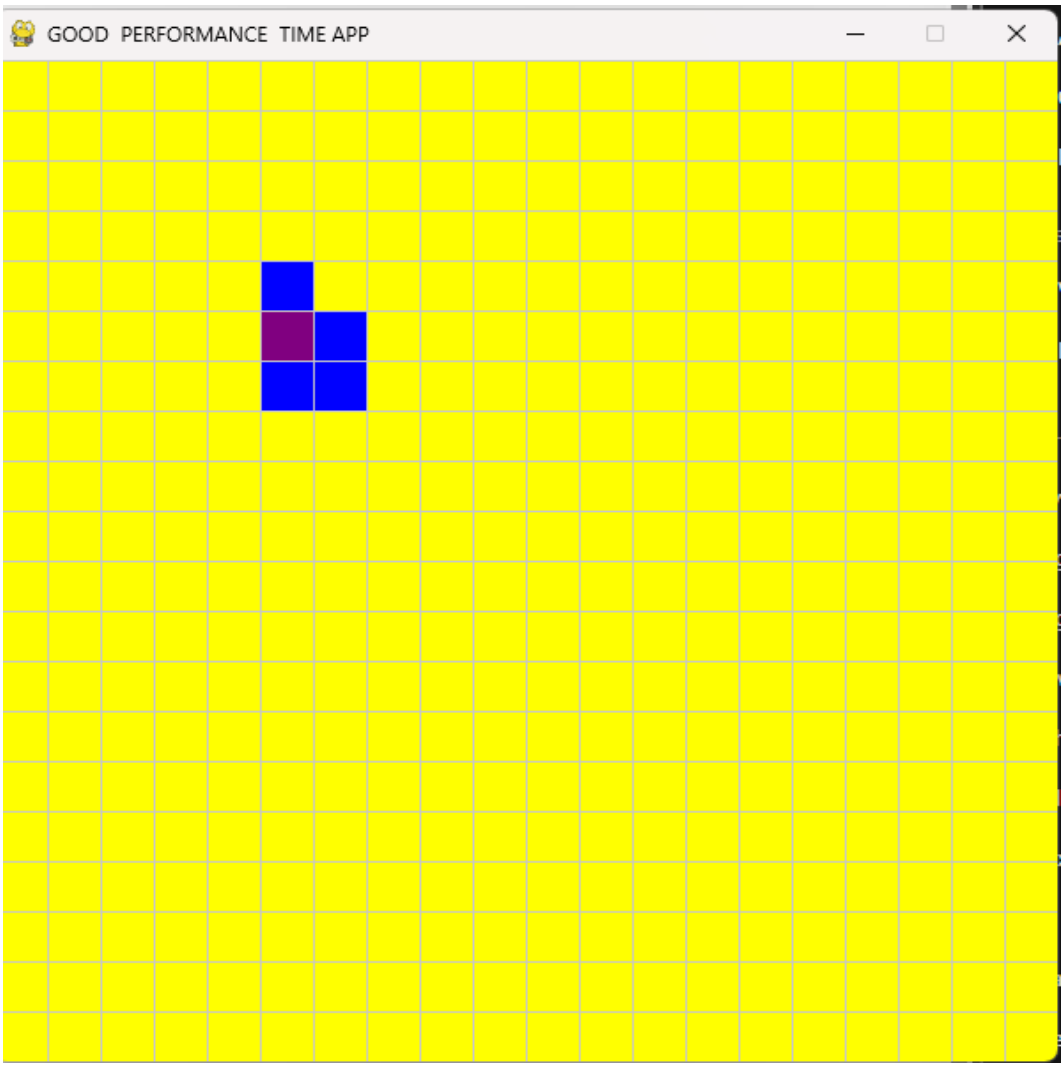
Breadth-First Search (BFS) – Best Case



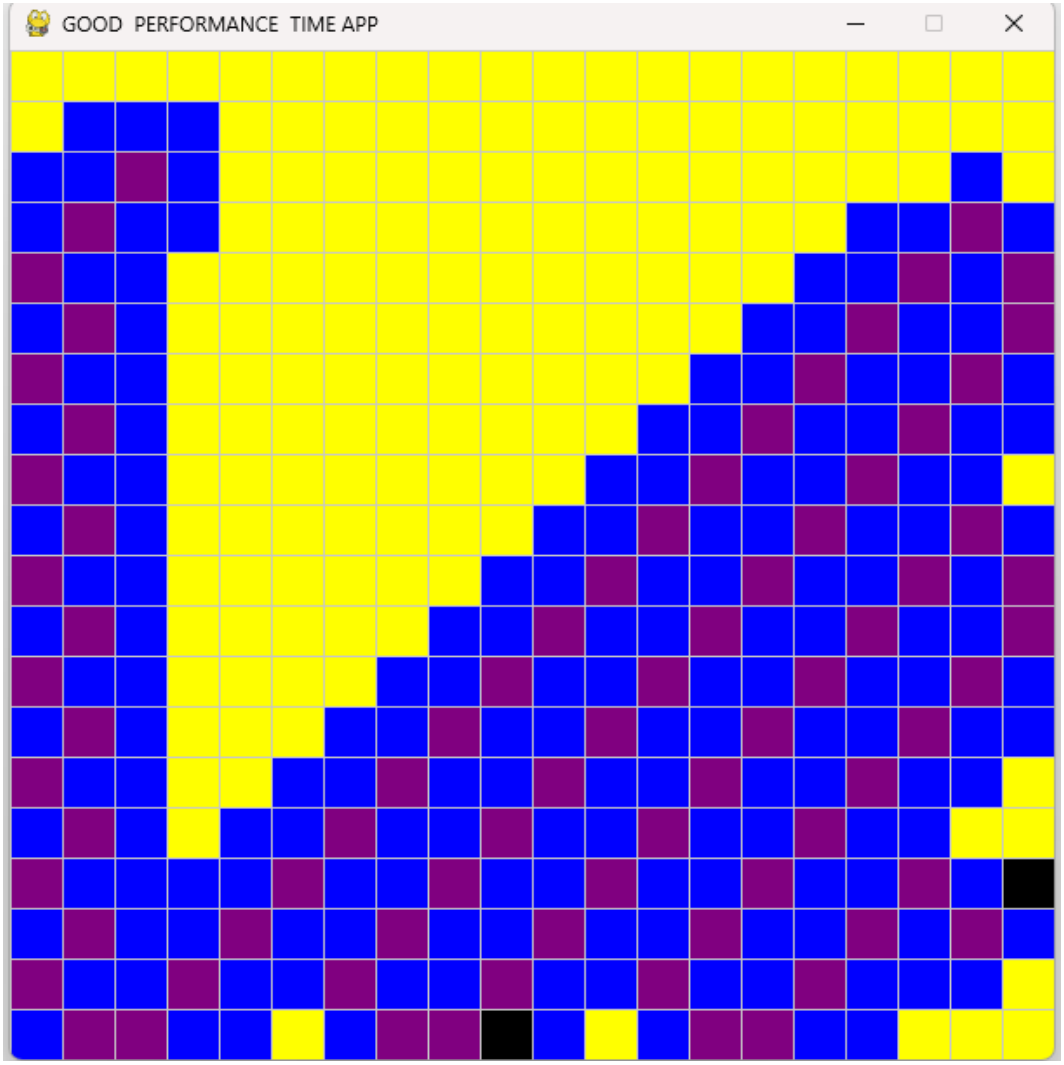
Breadth-First Search (BFS) – Worst Case



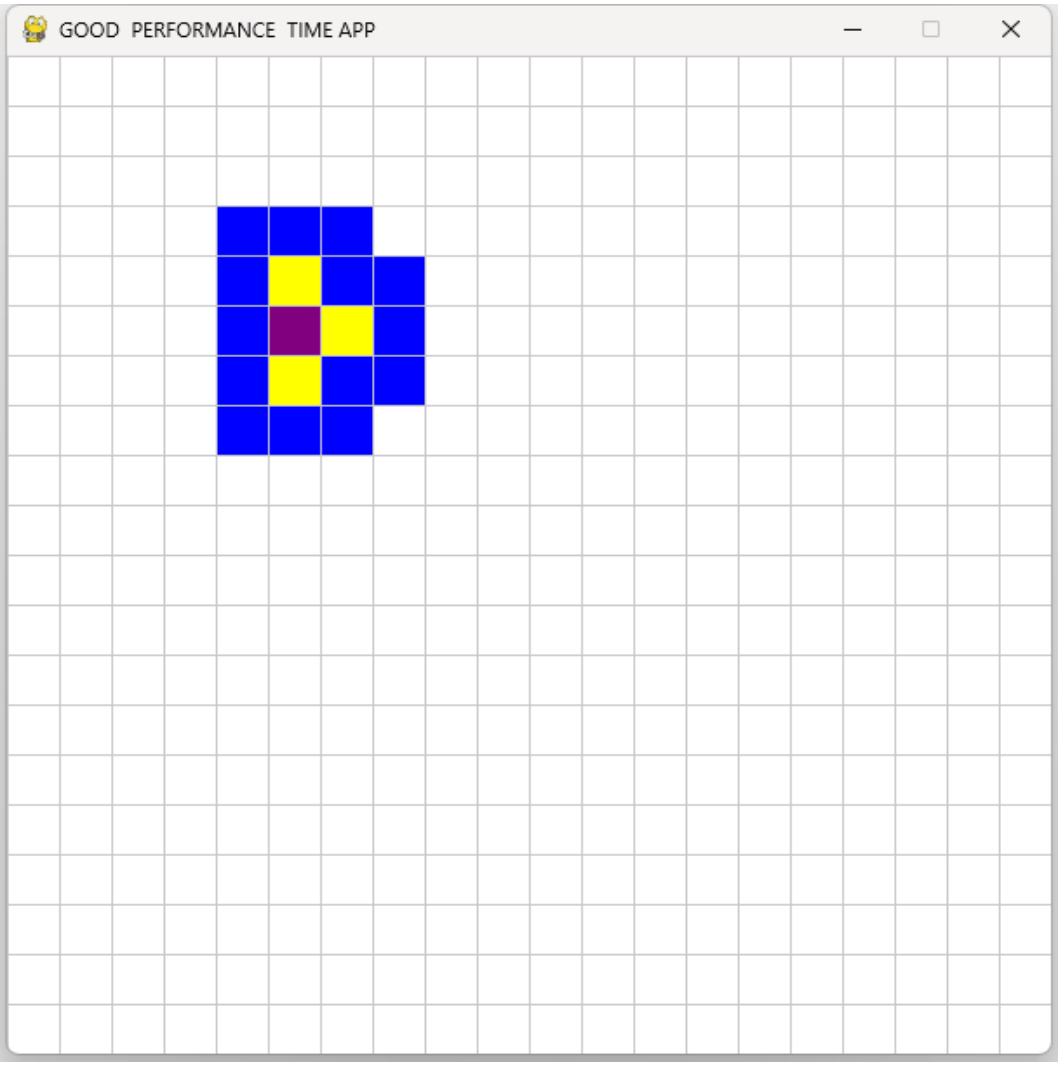
Depth-First Search (DFS) – Best Case



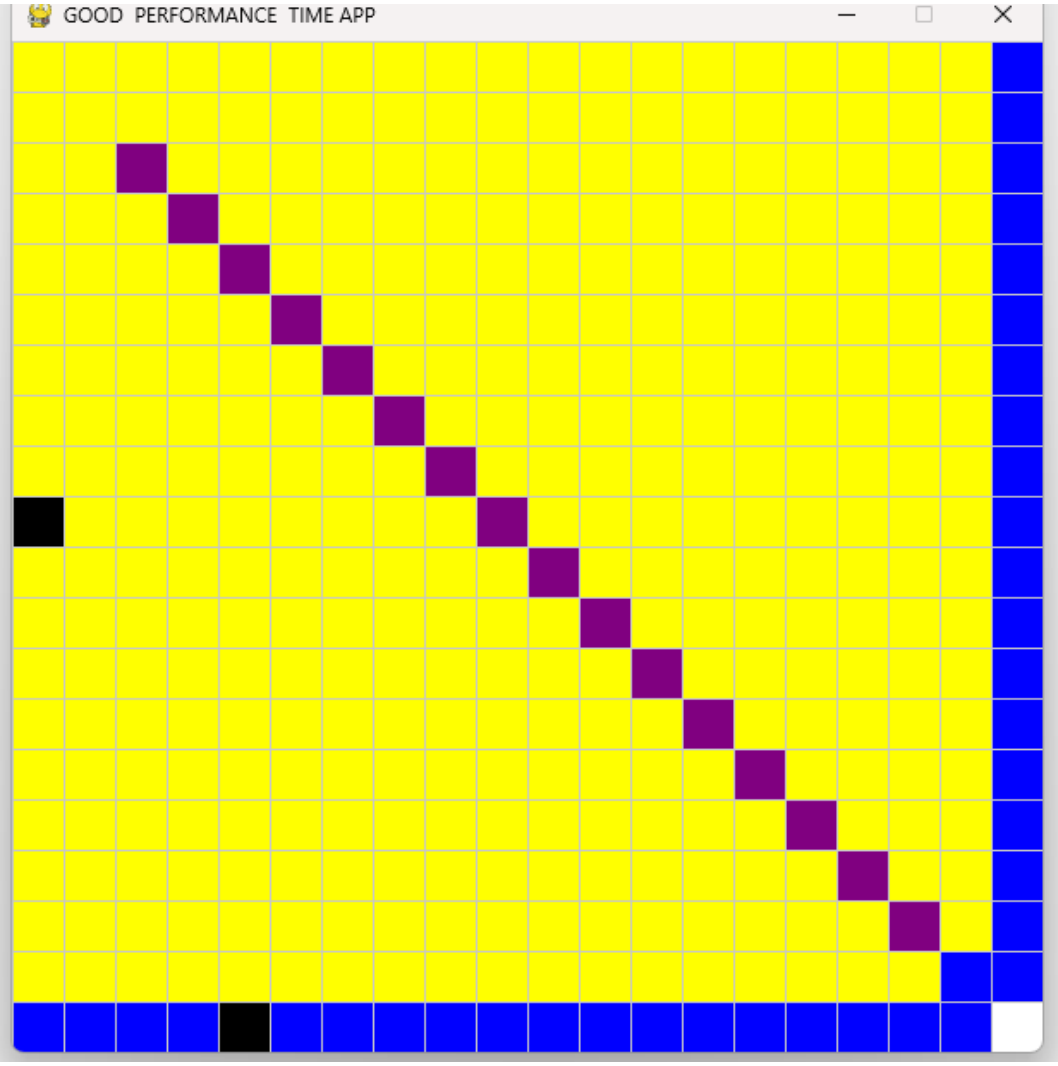
Depth-First Search (DFS) – Worst Case



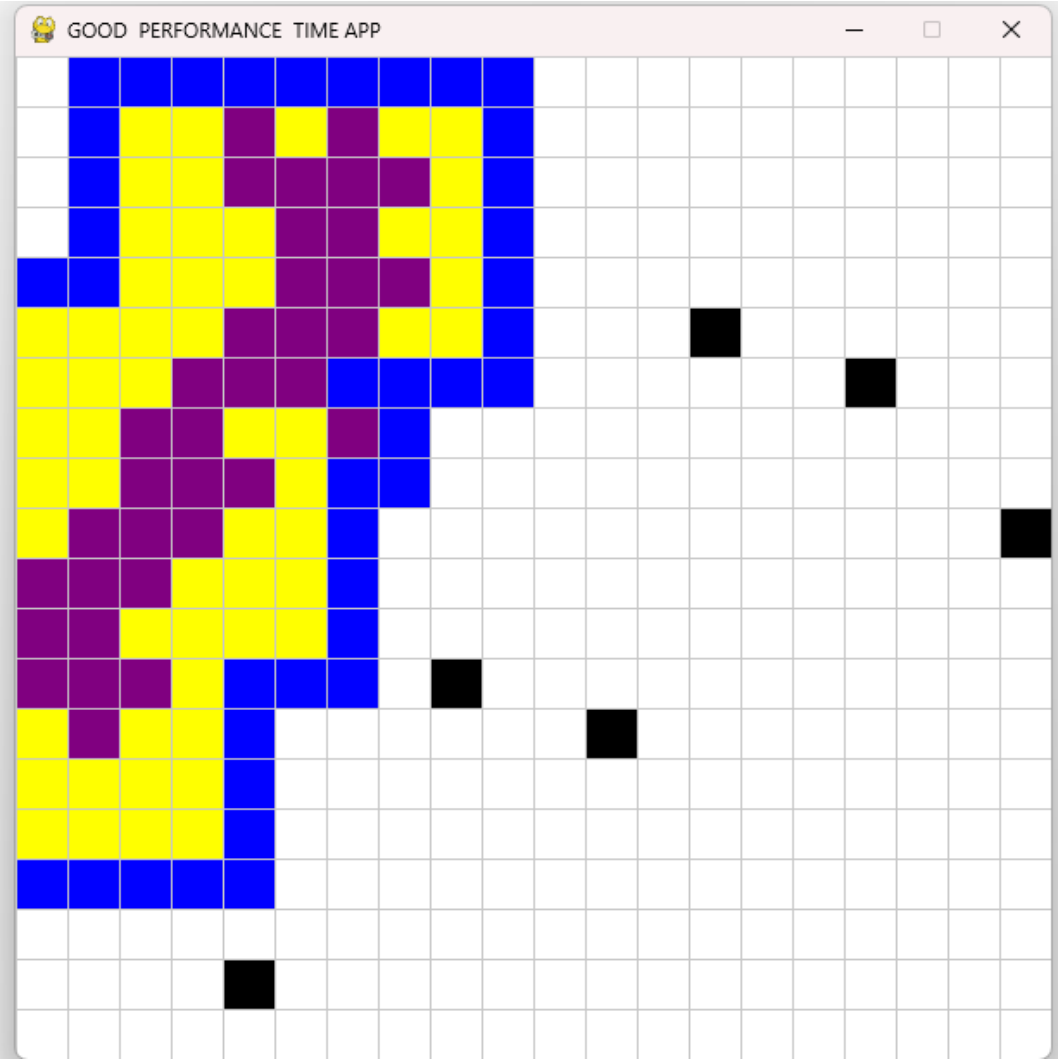
Uniform Cost Search (UCS) – Best Case



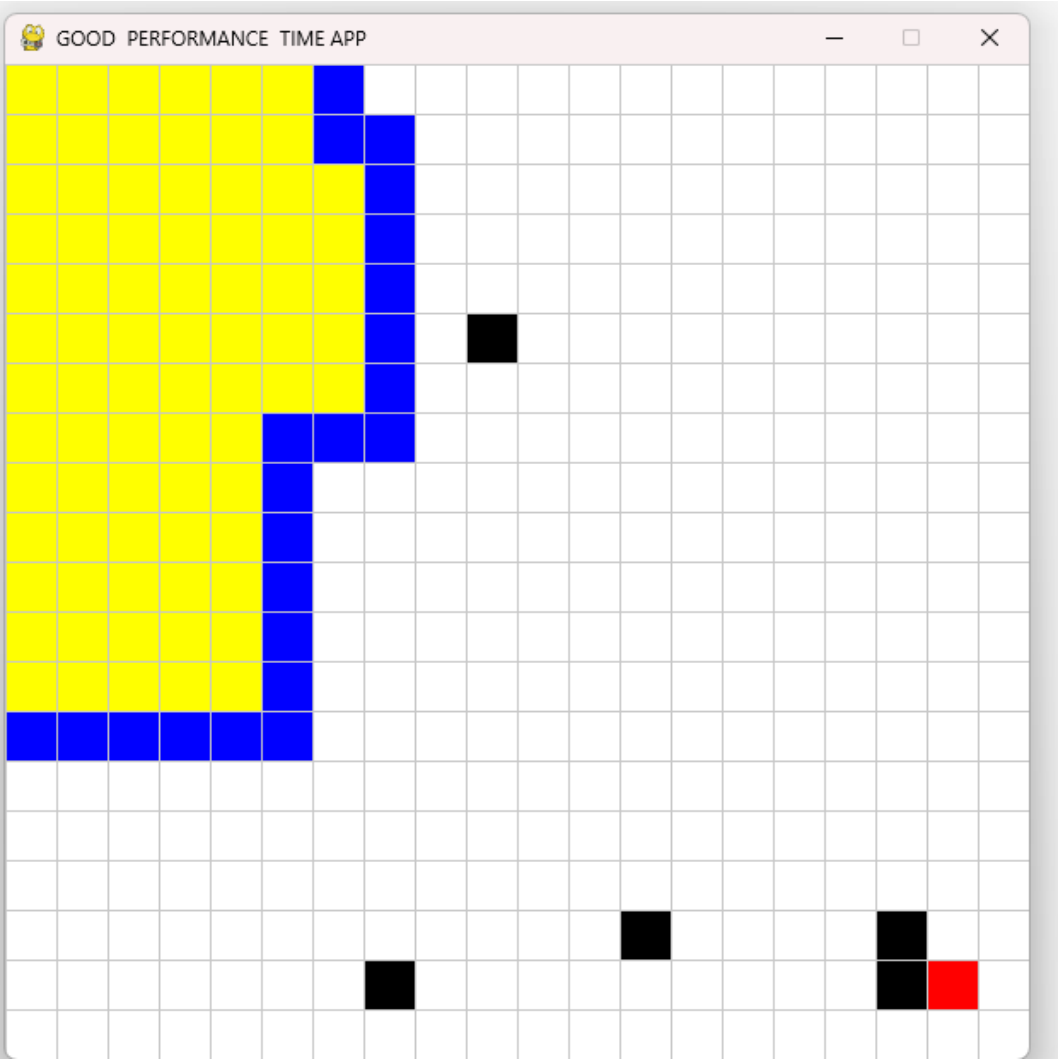
Uniform Cost Search (UCS) – Worst Case



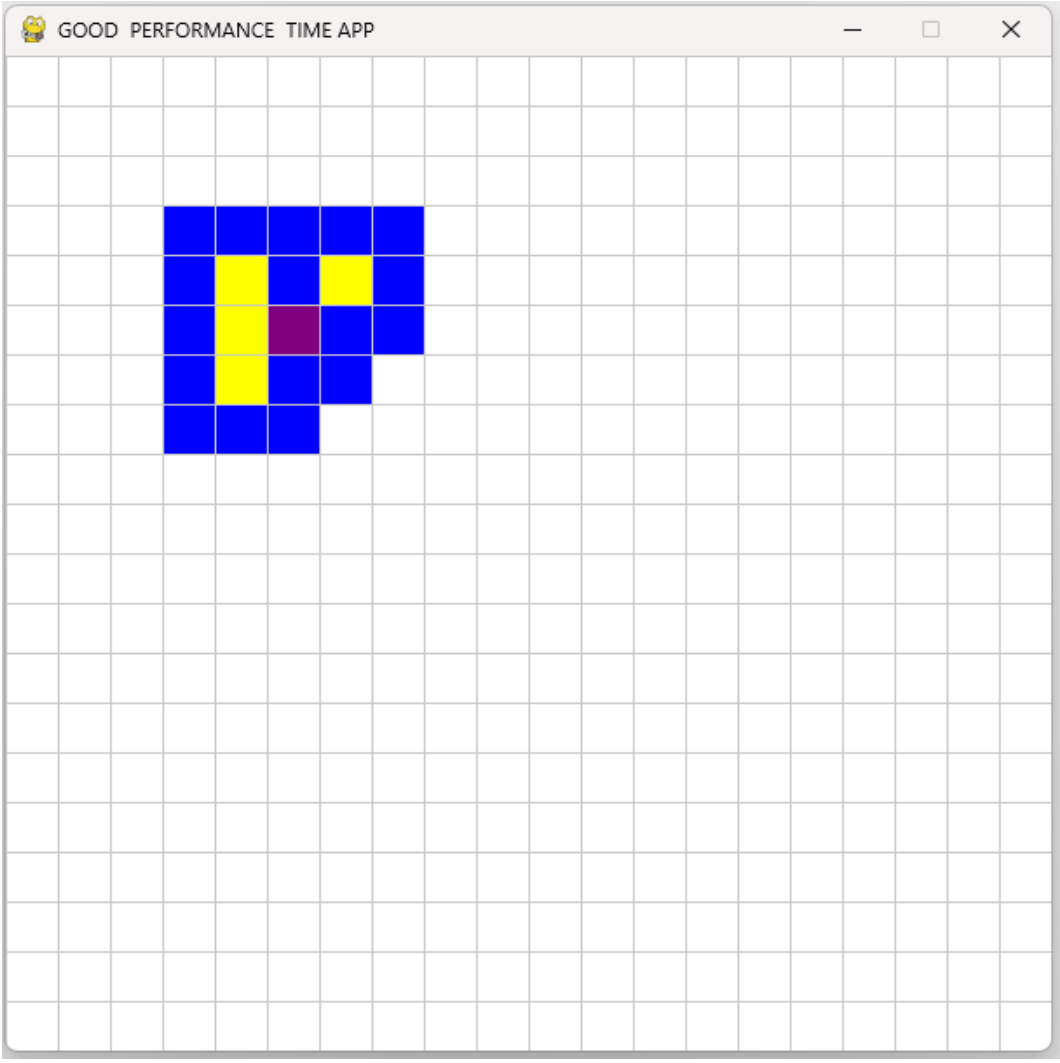
Depth-Limited Search (DLS) – Best Case



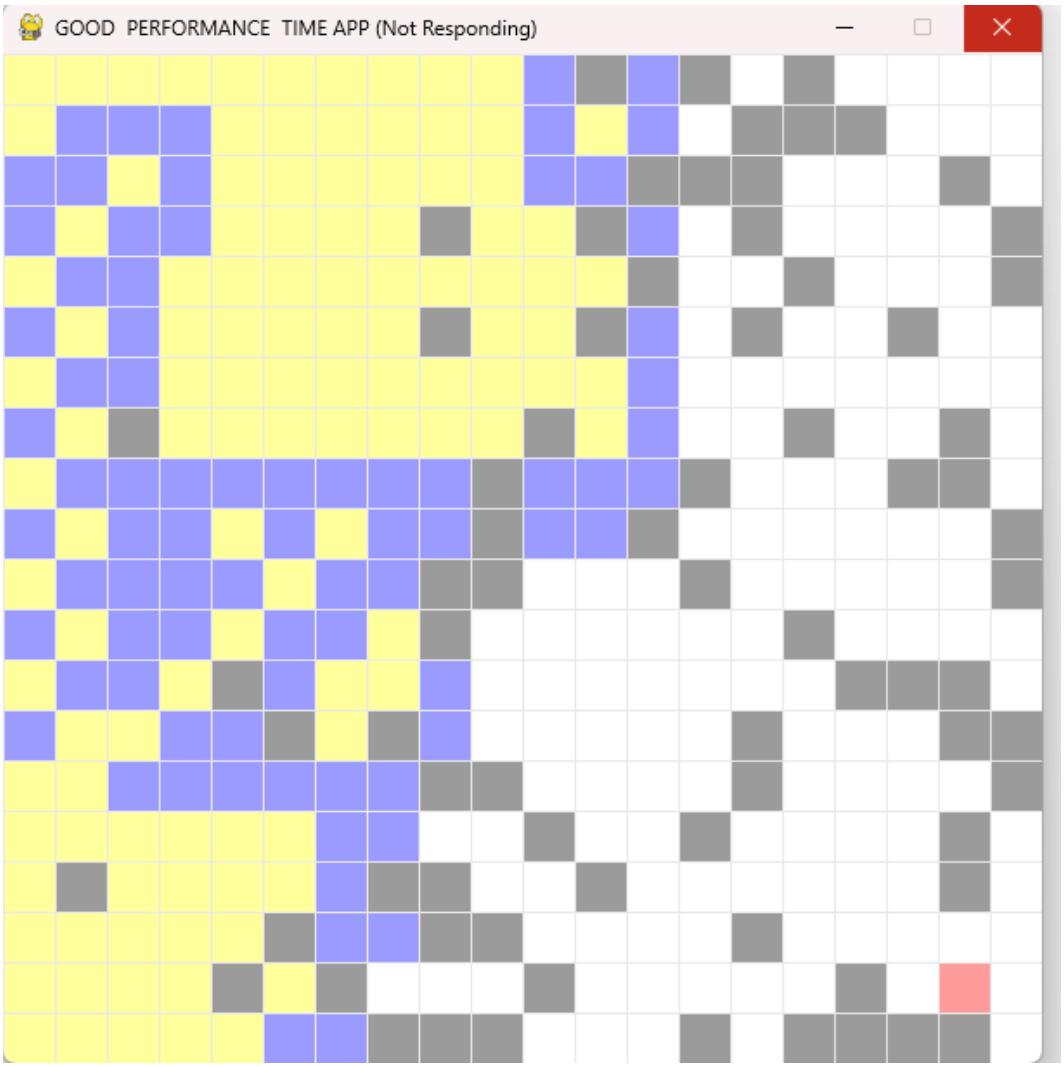
Depth-Limited Search (DLS) – Worst Case



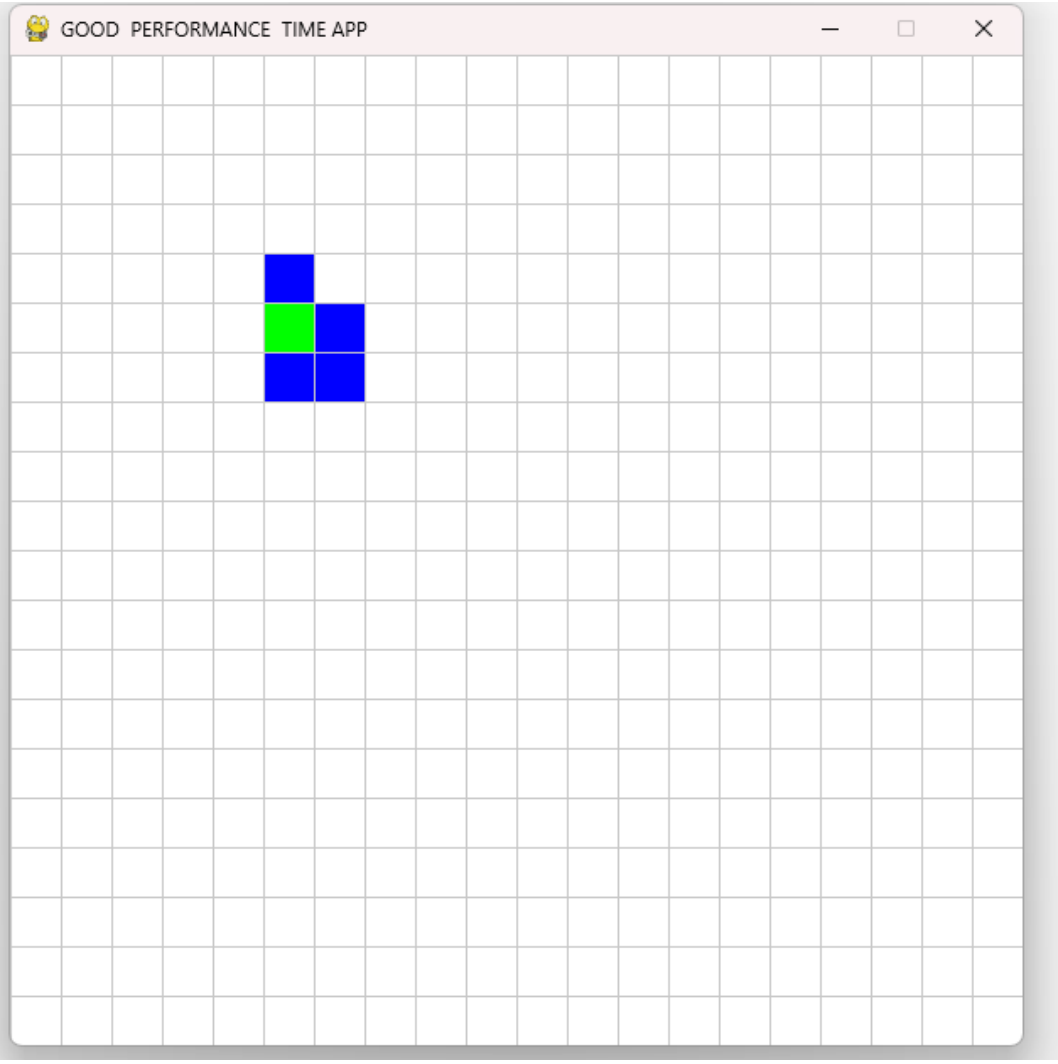
Iterative Deepening DFS (IDDFS) – Best Case



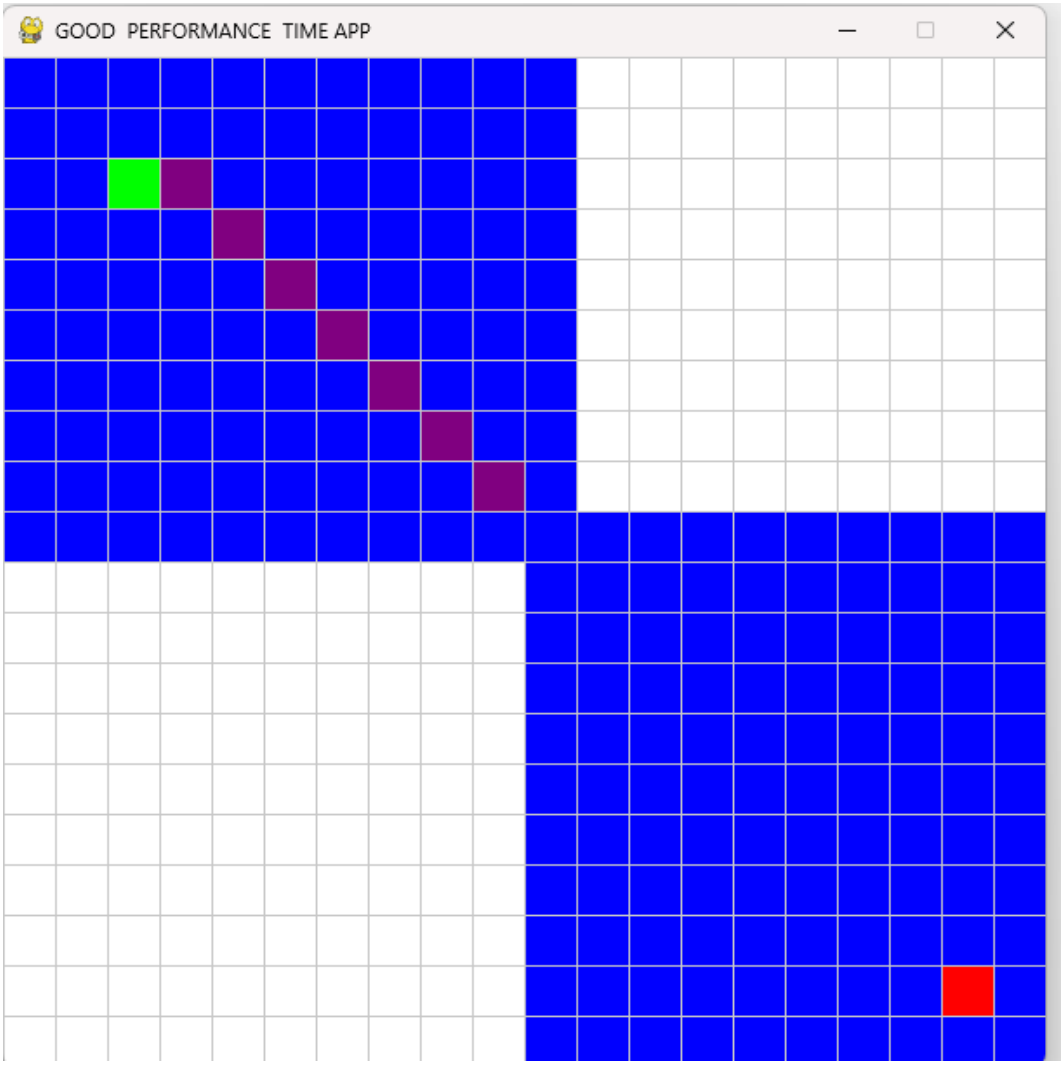
Iterative Deepening DFS (IDDFS) – Worst Case



Bidirectional Search – Best Case



Bidirectional Search – Worst Case



6. Conclusion

This assignment helped me understand how uninformed search algorithms differ in terms of exploration strategy, optimality, and memory usage. The visualization clearly demonstrated best and worst case behavior for each algorithm in a dynamic environment.