Project Report

**Project Title: Smart Fire Fighter**

**Submitted By: M. Afzal (22K-4323)**

**Course: AI**

**Instructor: Sir Khalid**

**Submission Date: May 09, 2025**

# 1. Executive Summary

This project simulates a grid-based game where an AI agent must escape from a fire-spreading environment. It uses several AI techniques including A\*, BFS, DFS, UCS, and Greedy Search. The objective was to compare the performance and intelligence of these algorithms within a dynamic hazard-based simulation. We also introduced innovations such as fire spread, real-time decisions, and interactive visualization using a GUI.

# 2. Introduction

## Background:

Inspired by pathfinding AI and survival scenarios, this project builds a game-like simulation that is not based on any traditional board game. Instead, it presents a unique environment where fire spreads unpredictably and the AI must find its way to safety, demonstrating decision-making under pressure.

## Objectives of the Project:

• Design a simulation game with dynamic obstacles  
• Implement and compare multiple classical AI search techniques  
• Analyze the performance of each AI approach under changing conditions  
• Visualize the AI’s decision-making in real time

# 3. Game Description

## Original Game Rules:

There is no traditional game base. Instead, the environment follows grid navigation logic where an agent must reach a goal node from a start node, avoiding fire, walls, and other hazards.

## Innovations and Modifications:

• Fire spreads randomly in each round  
• Water buckets and walls create strategic movement constraints  
• Multiple AI algorithms are integrated for comparison  
• GUI enables real-time decision visualization

# 4. AI Approach and Methodology

## AI Techniques Used:

• A\* Search  
• Breadth-First Search (BFS)  
• Depth-First Search (DFS)  
• Uniform Cost Search (UCS)  
• Greedy Search

## Algorithm and Heuristic Design:

• Heuristics used include Manhattan distance and distance from fire  
• Each algorithm was evaluated based on safety, optimality, and speed  
• Decision-making balanced between escaping and avoiding dynamic threats

## AI Performance Evaluation:

Performance was measured by success rate in reaching the goal, average time per decision, and the number of steps taken. A\* generally performed best in terms of optimality, while Greedy Search was fastest but often unsafe.

# 5. Game Mechanics and Rules

## Modified Game Rules:

• Agent starts at a defined cell and must reach a goal cell  
• Fire spreads after each move  
• Buckets can slow fire spread; walls are impassable

## Turn-based Mechanics:

1. Fire spreads randomly  
2. Agent evaluates and makes its move  
3. GUI updates the board  
4. Repeat until game ends

## Winning Conditions:

The agent wins by reaching the goal cell before being consumed by fire.

# 6. Implementation and Development

## Development Process:

The game was implemented in Python using Pygame. Each AI algorithm was written as a module. The GUI displays the board, agent movements, fire spread, and interactive components.

## Programming Languages and Tools:

• Language: Python  
• Libraries: Pygame, NumPy,Tkinter for gui  
• Tools: VSCode, GitHub

## Challenges Encountered:

• Fire spread logic with randomness and fairness  
• Synchronizing GUI updates with decision-making steps  
• Heuristic tuning for performance balance

# 7. Team Contributions

• M. Ali (22K-4438): Implemented BFS, DFS, and UCS search algorithms. Helped debug grid logic.

• Sameed ur Rehman (22K-4269): Built the fire spread system, random hazard generation, and GUI display.

• M. Afzal (22K-4323): Developed A\*, Greedy Search, and integrated all components into a working simulation. Also handled documentation.

# 8. Results and Discussion

A\* showed the best performance with the highest success rate and shortest path in most scenarios. Greedy Search had the fastest average time but often resulted in failure. UCS and BFS provided reliable but slower results. The visualization helped highlight the difference in decision-making strategies.

# 9. References

• Russell, S., & Norvig, P. (2016). Artificial Intelligence: A Modern Approach  
• https://www.geeksforgeeks.org/search-algorithms-in-ai/  
• https://www.pygame.org/docs/  
• Project implementation files and AI textbook notes

# 7.Group Members

**SAMEED UR REHMAN (22K-4269)**

**MUHAMMAD AFZAL(22K-4323)**

**MOHAMMAD ALI(22K-4438)**