# Cosmic Wayfinder: Search & Problem-Solving Agent

This project implements an intelligent agent navigating a 2D space environment to reach a target location efficiently. Unlike standard examples such as the 8-puzzle or simple mazes, this domain combines obstacles, variable-cost regions, and optional teleportation points (wormholes) to create a non-trivial pathfinding problem.

The agent traverses a grid containing nebulas, asteroids, black holes, and wormholes. Nebulas slow movement, asteroids increase traversal cost, black holes are impassable, and wormholes allow teleportation. Critically, the agent must decide whether to use a wormhole or continue along the current path, balancing speed and cost. This introduces a strategic layer beyond standard grid navigation.

## Comparative Evaluation

Algorithms were tested on multiple random maps, using **nodes expanded**, **path cost**, **computation time**, and **space usage** as metrics:

| Algorithm | Nodes Expanded | Path Cost | Time | Optimal | Notes |
|---|---|---|---|---|---|
| UCS | High | Lowest | Medium | Yes | Explores all low-cost paths; guaranteed optimal. |
| A* | Moderate | Lowest | Low | Yes | Heuristic reduces nodes expanded while maintaining optimality. |
| Greedy | Low | Higher | Very Low | No | Fastest exploration but may skip optimal paths. |

**Analysis:**

- **UCS** guarantees optimality but expands many nodes, consuming significant memory ($O(b^d)$ worst-case space).
- **A*** reduces node expansions using heuristics, maintaining optimality with moderate space and low runtime.
- **Greedy** has minimal memory usage and fastest runtime but can produce suboptimal paths.

Teleportation affects performance differently: UCS considers it only if it reduces cumulative cost, A* evaluates it alongside heuristic distance, and Greedy may skip it, highlighting the interaction between algorithm strategy and domain-specific choices.

## Design Justification

This domain introduces complexity beyond classical examples by combining variable-cost obstacles and optional teleports. Agents must evaluate multiple path options rather than blindly following the shortest distance.

The algorithms were chosen to illustrate trade-offs between completeness, optimality, and efficiency. UCS and A* demonstrate optimality, while Greedy emphasizes speed.

The interactive Pygame visualization enhances understanding by showing agents, obstacles, and wormholes in real time. "Race mode" demonstrates decision-making differences between A* and Greedy, providing intuitive insight into path selection and teleportation usage.

## Usage

Run the program in a terminal with: *python cosmic_wayfinder.py*

This will launch the interactive environment, allowing you to observe the agent navigating the grid, testing UCS, A*, and Greedy strategies, and experimenting with teleportation and obstacles.

## Conclusion

The Cosmic Wayfinder project demonstrates a **non-trivial search problem** requiring strategic decision-making. Comparative evaluation highlights the trade-offs between completeness, time/space complexity, and path optimality. Optional teleportation challenges agents to evaluate multiple strategies. Overall, this project provides a clear, interactive demonstration of search and problem-solving in a complex environment while remaining modular, reproducible, and well-documented.