Project Proposal: BFS-Based Maze Game in Python

Project Title:

Maze Explorer: A Pathfinding Game with BFS Algorithm Visualization

Objective:

To develop an interactive maze game using Python and the Pygame library, where players can either solve the maze themselves or visualize the Breadth-First Search (BFS) algorithm automatically finding the shortest path.

Motivation:

Pathfinding algorithms are foundational in AI and robotics. This game visualizes the BFS algorithm in action, making it an engaging way for students and enthusiasts to see how search algorithms operate.

Key Features:

- Maze Generation using recursive backtracking.
- Game Modes: Play Mode and Solution Mode (BFS Visualization).
- BFS Visualization with visited cells and shortest path highlighting.
- Dynamic maze scaling with level progression.
- Graphical interface using Pygame.
- Smooth gameplay using Pythons asyncio.

Technologies Used:

Python 3.10+, Pygame, Asyncio, BFS Algorithm

Target Users:

Students, educators, and game developers interested in algorithm visualization and procedural generation.

Project Timeline:

Project Proposal: BFS-Based Maze Game in Python

Phase	Description	Duration
Requirements Gathering	Define core features and game mechanics	2 days
Maze Generator	Implement recursive backtracker	3 days
BFS Pathfinding Logic	Implement BFS with visualization	3 days
Game UI & Controls	Add user interface and keyboard controls	3 days
Play Mode Logic	Add win condition and level handling	2 days
Solution Mode Logic	Integrate BFS visualization	2 days
Testing & Debugging	Ensure reliability and fix bugs	2 days
Documentation	Write code and user documentation	1 day

Expected Outcomes:

- A playable maze-solving game with BFS solution mode.
- Visual learning tool for BFS.
- Code base extensible to other pathfinding algorithms.

Future Improvements:

- Add lives and scoring system.
- Integrate more algorithms (A*, Dijkstra).
- Export solutions as images or files.
- Build web/mobile versions.