Maze Generation and Solving Game with Ai

This Python script utilizes the Pygame library to create a simple maze generation and solving game. The game involves generating a random maze using a tree-based algorithm and allows the user to interactively select starting and ending points. The program then finds and displays the path between these two points using a pathfinding algorithm.

Code Overview:

- **Root Class:**
  - The **Root** class represents a node in a tree structure used for maze generation and pathfinding.
  - Each node has a value, a reference to its parent, and a list of child nodes.
  - The **addChild** method adds a child node to the current node.
- **Game Class:**
  - The **Game** class manages the main game loop and interactions.
  - It includes methods for drawing the maze canvas, handling user input, generating mazes, and finding paths.
- **Maze Generation:**
  - The maze is generated using a recursive tree-based algorithm.
  - The **generate_mage** method initializes a root node and builds a tree representing the maze.
  - The **build_tree** method recursively generates the maze by connecting neighboring nodes in a random order.
- **Pathfinding:**
  - The game allows the user to select starting and ending points by clicking on the maze.
  - The **find_solution** method uses a recursive approach to find a path from a given starting point to an endpoint in the generated maze.
- **User Interaction:**
  - The game responds to mouse clicks for selecting start and end points and keyboard input for resetting the maze and initiating pathfinding.
- **Pygame Visualization:**
  - The Pygame library is used for graphics and user interface.
  - The maze canvas is displayed on the screen, and the solution path is drawn using circles and lines.

Algorithms Used:

- **Maze Generation:**

    - The maze generation algorithm employs a recursive backtracking technique.
    - The algorithm explores each cell in the grid, randomly connecting neighboring cells to form a maze.
- **Pathfinding:**
    - Pathfinding is achieved using a modified depth-first search (DFS) algorithm.
    - The **find_solution** method recursively explores the tree structure to find a path from the starting point to the endpoint.

How to Play:

- **Selecting Start and End Points:**
    - **Left-click** on a cell to set the starting point (indicated by a white circle).
    - **Right-click** on a cell to set the ending point (indicated by a white rectangle).
- **Generating a New Maze:**
    - Press the **SPACE** key to generate a new random maze.
- **Solving the Maze:**
    - Press the **RETURN** key to find and display the solution path from the selected start to end points.
- **Exiting the Game:**
    - Click the close button or press **ALT + F4** to exit the game.