

```
import random
```

```
import sys
```

```
from queue import Queue
```

```
# Constants for maze symbols
```

```
WALL = "\033[91m█\033[0m"
```

```
OPEN_SPACE = "\033[94m○\033[0m"
```

```
START = "S"
```

```
END = "E"
```

```
PATH = "\033[92m●\033[0m"
```

```
def generate_maze(n, wall_percentage):
```

```
    # Create an n x n maze with walls and open spaces
```

```
    maze = [[WALL if random.random() < wall_percentage else OPEN_SPACE for _ in range(n)] for _ in range(n)]
```

```
    maze[0][0] = START
```

```
    maze[n - 1][n - 1] = END
```

```
    return maze
```

```
def print_maze(maze):
```

```
    for row in maze:
```

```
        print("".join(row))
```

```
def find_path(maze):
```

```
    # BFS to find a path from start to end
```

```
    start = (0, 0)
```

```
    end = (len(maze) - 1, len(maze) - 1)
```

```
    q = Queue()
```

```
    q.put(start)
```

```

visited = set()

while not q.empty():

    current = q.get()

    if current == end:

        break

    for dx, dy in [(1, 0), (-1, 0), (0, 1), (0, -1)]:

        x, y = current[0] + dx, current[1] + dy

        if 0 <= x < len(maze) and 0 <= y < len(maze) and maze[x][y] == OPEN_SPACE and (x, y) not in visited:

            q.put((x, y))

            visited.add((x, y))

            maze[x][y] = PATH

def main():

    n = 10 # Change the maze size as needed

    wall_percentage = 0.3 # Adjust the wall percentage as needed

    while True:

        maze = generate_maze(n, wall_percentage)

        print("Generated Maze:")

        print_maze(maze)

        user_choice = input("Enter 'P' to print the path, 'G' to generate another puzzle, or 'Q' to quit: ").strip().lower()

        if user_choice == 'p':

            find_path(maze)

            print("\nMaze with Path:")

            print_maze(maze)

        elif user_choice == 'g':

            continue

```

```
elif user_choice == 'q':
```

```
    sys.exit("Goodbye!")
```

```
else:
```

```
    print("Invalid choice. Please enter 'P', 'G', or 'Q'.")
```

```
if __name__ == "__main__":
```

```
    main()
```