Nhóm 9:

Thành viên:

22119054 Lưu Trọng Dũng

22110062 Đào Nguyên Phúc

Đề tài: Design and implement Maze game using AI algorithms

**1. Giới thiệu**

**Mục tiêu**: Xây dựng một trò chơi tìm đường AI trong môi trường mê cung, ứng dụng các thuật toán tìm kiếm đường đi tối ưu nhằm dẫn dắt nhân vật từ điểm bắt đầu đến điểm kết thúc trên bản đồ được tạo bằng thuật toán DFS.

**Phát triển**: Tạo ra một trò chơi nhỏ với giao diện đồ họa cho phép người chơi tương tác, lựa chọn các thuật toán tìm đường khác nhau như DFS, BFS, và A\*, …

**Chế độ tạo bản đồ**: Người dùng có thể tùy chỉnh vị trí của điểm bắt đầu, điểm kết thúc, và các bức tường.

**Màn hình trò chơi**: Hiển thị bản đồ cùng với các bước di chuyển của nhân vật theo thuật toán được lựa chọn.

**2. Ngôn ngữ sử dụng**

**Ngôn ngữ lập trình**: Python

**Thư viện chính**:

**Pygame**: Được sử dụng để xây dựng giao diện đồ họa và tương tác người dùng.

**Numpy**: Hỗ trợ xử lý các phép toán ma trận nhằm xây dựng và thao tác trên bản đồ mê cung.

**Deque, List**: Sử dụng các cấu trúc dữ liệu để triển khai các thuật toán tìm đường.

**3. Thiết kế hệ thống**

**Kiến trúc chính**:

**Class GameAI**: Chịu trách nhiệm xử lý toàn bộ logic của trò chơi, giao diện người dùng, và các thuật toán AI.

**Thuật toán tìm đường** (dự kiến sự dụng):

**DFS (Depth-First Search)**: Duyệt theo chiều sâu, hiệu quả cho việc khám phá tất cả các đường đi có thể.

**BFS (Breadth-First Search)**: Tìm đường ngắn nhất bằng cách duyệt theo chiều rộng, đảm bảo tìm kiếm mọi điểm trên cùng một cấp độ trước khi chuyển sang cấp độ tiếp theo.

***A* (A-star)**: Sử dụng hàm chi phí dự đoán (heuristic) để tối ưu hóa đường đi, kết hợp chi phí thực tế đã đi với chi phí ước lượng còn lại nhằm tìm ra đường ngắn nhất hiệu quả nhất.

**Hill Climbing**: Áp dụng chiến lược tìm kiếm địa phương để chọn hướng di chuyển tại mỗi bước dựa trên giá trị "happiness" hoặc chi phí. Nếu tìm thấy hướng di chuyển mang lại giá trị tốt hơn, thuật toán sẽ tiếp tục di chuyển theo hướng đó cho đến khi không còn hướng nào tốt hơn.

**Greedy**: Tương tự như Hill Climbing, nhưng thuật toán này chọn bước đi theo cách tham lam, tức là luôn chọn bước đi tốt nhất tại mỗi bước mà không quan tâm đến hệ quả dài hạn, điều này có thể dẫn đến đường đi không tối ưu.

**Giao diện người dùng**:

**Menu chính**: Cung cấp các lựa chọn cho người chơi, bao gồm: bắt đầu trò chơi, tự tạo bản đồ, chọn thuật toán tìm đường, và thoát trò chơi.

**Tạo map**: cung cấp khả năng tự thiết kế bản đồ riêng để thử nghiệm các thuật toán khác nhau.

**4. Kết luận**

Dự án "Thiết kế và triển khai trò chơi mê cung sử dụng các thuật toán AI" không chỉ cung cấp một môi trường thú vị cho người chơi và so sánh các hiệu xuất của thuật toán tìm đường khác nhau. Qua việc triển khai trò chơi, người chơi có thể hiểu rõ hơn về cách thức hoạt động của các thuật toán như DFS, BFS, A\*, Hill Climbing và Greedy.

Thông qua việc tùy chỉnh bản đồ và lựa chọn thuật toán, người chơi không chỉ trải nghiệm mà còn có thể nhận diện ưu nhược điểm của từng thuật toán trong việc tìm kiếm đường đi tối ưu.

Dự án mở rộng ra khả năng phát triển thêm các tính năng mới và cải tiến trong tương lai, như việc áp dụng các thuật toán AI phức tạp hơn hoặc cải thiện giao diện người dùng để nâng cao trải nghiệm của người chơi.