**Báo cáo môn học Trí tuệ nhân tạo**

1. Thành Viên Nhóm  
    tên 5 đứa  
     
   II. Nội Dung Đề Tài
2. Tổng quan về đề tài
3. Cơ sở Lý thuyết
4. Demo BFS
5. Kết luận
6. **Tổng quan về đề tài**
   1. **Lý do chọn đề tài**

**Áp Dụng Kiến Thức Thực Tế:** Việc áp dụng kiến thức đã học về BFS vào thực tế là để củng cố và hiểu sâu hơn về thuật toán này.

**Giải Pháp Phổ Biến và Linh Hoạt:** BFS là một trong những thuật toán tìm kiếm cơ bản và phổ biến, có thể được áp dụng trong nhiều lĩnh vực khác nhau như trò chơi, truy xuất dữ liệu, xử lý đồ thị, và nhiều hơn nữa. Sự linh hoạt của BFS mang lại nhiều cơ hội cho việc nghiên cứu và ứng dụng.

* 1. **Mục tiêu**

Tạo ra một giải thuật tìm kiếm có thể áp dụng vào cho game hoặc xử các vấn đề về đồ thị

1. **Cơ Sở Lý thuyết**
   1. **Khái niệm về Python**

Python là ngôn ngữ lập trình máy tính bậc cao thường được sử dụng để xây dựng trang web và phần mềm, tự động hóa các tác vụ và tiến hành phân tích dữ liệu . Python là ngôn ngữ có mục đích chung, nghĩa là nó có thể được sử dụng để tạo nhiều chương trình khác nhau và không chuyên biệt cho bất kỳ vấn đề cụ thể nào

* 1. **Giới Thiệu Thuật Toán BFS**

**Có thể chia ra 3 phần là: 1 giới thiệu + 2 cách làm việc/phân tích + 3 code BFS ( không phải cái demo chỉ là code BFS )**

Thuật toán BFS (Breadth-First Search - Tìm kiếm theo chiều rộng) là một trong những thuật toán cơ bản và quan trọng nhất trong lĩnh vực truy cập đồ thị và tìm kiếm. Nó được sử dụng để tìm kiếm trong các cấu trúc dữ liệu đồ thị hoặc cây, bắt đầu từ một đỉnh cụ thể.

**Cách Hoạt Động:**

Thuật toán BFS hoạt động bằng cách duyệt đồ thị "theo chiều rộng", tức là từ đỉnh gốc, nó sẽ trước tiên duyệt qua tất cả các đỉnh kề với đỉnh gốc, sau đó duyệt qua tất cả các đỉnh kề với những đỉnh đã duyệt trước đó, và tiếp tục như vậy cho đến khi tìm được đỉnh mục tiêu hoặc duyệt qua toàn bộ đồ thị.

Thuật toán này thường được triển khai bằng cách sử dụng một hàng đợi (queue) để lưu trữ các đỉnh cần duyệt tiếp theo. Khi một đỉnh được duyệt qua, tất cả các đỉnh kề với nó sẽ được thêm vào hàng đợi, và quá trình duyệt tiếp tục cho đến khi hàng đợi trở thành rỗng.

**Ưu Điểm:**

Dễ Hiểu và Triển Khai: BFS là một thuật toán đơn giản và dễ hiểu, có thể triển khai một cách dễ dàng trong nhiều ngôn ngữ lập trình.

Đảm Bảo Tìm Kiếm Ngắn Nhất: BFS luôn đảm bảo tìm kiếm đường đi ngắn nhất từ đỉnh gốc đến mọi đỉnh khác trong đồ thị (nếu tồn tại).

Phát Hiện Chu trình: BFS có thể được sử dụng để phát hiện chu trình trong đồ thị, nếu có.

**Hạn Chế:**

Độ Phức Tạp: Trong một số trường hợp, đặc biệt là với đồ thị lớn, BFS có thể đòi hỏi một lượng lớn bộ nhớ để lưu trữ hàng đợi và dẫn đến hiệu suất kém.

Không Phù Hợp cho Đồ Thị Vô Hướng Có Trọng Số: BFS thường không phù hợp cho việc tìm kiếm đường đi ngắn nhất trên đồ thị vô hướng có trọng số.

Muốn nhét ảnh code vào thì tùy

1. **Demo BFS**

Show code chạy hết

1. **Kết Luận**

Giới thiệu được thuật toán BFS

Hiểu rõ được vấn đề cốt lỗi của thuật toán

Làm nền tảng phát triển game sau này