**TRUY TÌM KHO BÁU TRÊN ĐẢO HOANG**

1️. Trạng thái đầu (Initial State)

Bối cảnh ban đầu:

* Người chơi bắt đầu tại một điểm ngẫu nhiên trên đảo hoang.
* Không biết vị trí kho báu.
* Có một số tài nguyên cơ bản (ví dụ: nước, thức ăn, công cụ).
* Chưa có bản đồ đầy đủ, chỉ thấy khu vực gần xung quanh.

Các yếu tố xác định trạng thái đầu:

Vị trí nhân vật: (X, Y) trên bản đồ.  
Lượng tài nguyên: Sức khỏe = 100%, Nước = 50%, Thức ăn = 50%.  
 Bản đồ: Chỉ mở ra một phần nhỏ gần vị trí ban đầu.  
Thời gian trong ngày: Có thể là sáng hoặc chiều tùy thiết kế game.

2️. Trạng thái đích (Goal State)

Điều kiện thắng game:

* Người chơi tìm thấy và đào được kho báu.
* Hoặc người chơi trốn thoát khỏi đảo (nếu có cơ chế thoát khỏi đảo).

Các yếu tố xác định trạng thái đích:  
 Người chơi đứng đúng vị trí của kho báu.  
Có công cụ cần thiết để đào kho báu.  
Không bị chết do đói, khát, hoặc bị tấn công.

(Nếu có nhiều cách kết thúc game)  
 Thắng game: Tìm được kho báu hoặc trốn thoát thành công.  
 Thua game: Chết vì đói, khát, thú dữ hoặc rơi vào bẫy.

1.Thuật toán tìm đường & khám phá bản đồ

BFS (Breadth-First Search) – Tìm đường qua khu vực chưa biết

Dùng khi:

* Người chơi khám phá đảo theo từng khu vực và muốn tìm đường đi an toàn nhất.
* AI NPC (ví dụ: quái vật hoặc đồng đội) cần khám phá khu vực theo từng bước.

Ứng dụng:

* Tìm kiếm tài nguyên hoặc manh mối trên đảo.
* AI trợ lý (nếu có) giúp người chơi tìm đường.
* Xác định vùng an toàn tránh thú dữ hoặc bẫy.

DFS (Depth-First Search) – Khám phá hang động & lối đi bí mật

Dùng khi:

* Người chơi muốn khám phá sâu một khu vực (ví dụ: đi vào hang động).
* Xác định vị trí kho báu nếu có một chuỗi manh mối dẫn đến nó.

Ứng dụng:

* Tìm kho báu trong hang động hoặc trong các khu vực hẹp, nơi có nhiều ngõ cụt.
* Khám phá bản đồ với tính năng ngẫu nhiên, khi có nhiều đường đi nhưng chỉ một đường đúng.

*A (A-Star) – Tìm đường tối ưu tránh nguy hiểm*\*

Dùng khi:

* Người chơi cần tìm đường nhanh nhất đến một điểm (ví dụ: kho báu, nguồn nước, nơi trú ẩn).
* AI cần đuổi theo hoặc trốn tránh người chơi theo cách hiệu quả nhất.

Ứng dụng:

* AI quái vật rượt đuổi người chơi thông minh hơn.
* Người chơi tìm đường nhanh nhất qua rừng rậm hoặc đầm lầy.
* Chế độ chạy trốn, khi cần nhanh chóng tìm nơi trú ẩn.

Dijkstra – Tìm đường tối ưu khi có nhiều loại địa hìnhDùng khi:

* Địa hình có nhiều loại với mức độ khó khác nhau (ví dụ: đi trên đường đá nhanh hơn lội qua sình lầy).

Ứng dụng:

* Người chơi tính toán lộ trình an toàn nhất để tiết kiệm sức lực.
* AI quyết định lộ trình hợp lý để không đi vào khu vực nguy hiểm.

2. Thuật toán AI & Hành vi của sinh vật trên đảo

Finite State Machine (FSM) – AI quái vật & động vật hoang dã

Dùng khi:

* AI cần có nhiều trạng thái hành vi (tuần tra, phát hiện người chơi, săn đuổi, quay về).

Ứng dụng:

* Quái vật sẽ tuần tra, đuổi theo và tìm kiếm khi mất dấu người chơi.
* Động vật trên đảo có thể sợ hãi hoặc tấn công tùy theo tình huống.

Ví dụ AI Quái vật FSM:

* Tuần tra – Đi theo tuyến cố định.
* Phát hiện người chơi – Nếu thấy dấu vết (âm thanh, di chuyển).
* Săn đuổi – Rượt theo người chơi bằng thuật toán A\*.
* Mất dấu – Quay lại trạng thái tuần tra nếu mất dấu.

Minimax & Alpha-Beta Pruning – AI thông minh hơn khi đấu trí

Dùng khi:

* AI quái vật hoặc NPC cần đưa ra quyết định chiến thuật khi đấu trí với người chơi.

Ứng dụng:

* Nếu người chơi đặt bẫy, AI sẽ tìm cách né hoặc phá hủy bẫy.
* AI sẽ phán đoán đường đi của người chơi thay vì chạy bừa bãi.

Monte Carlo Tree Search (MCTS) – AI học cách săn đuổi thông minh

Dùng khi:

* Quái vật hoặc NPC có thể tìm hiểu hành vi của người chơi theo thời gian.

Ứng dụng:

* AI có thể học từ các lần chơi trước, trở nên khó đối phó hơn.
* Nếu người chơi thường chạy theo một hướng, AI sẽ chặn đường đó.

3. Thuật toán giải đố & tạo manh mối kho báu

Backtracking – Giải câu đố & tạo mê cung kho báu

Dùng khi:

* Cần tạo câu đố mở khóa cửa hoặc mê cung dẫn đến kho báu.

Ứng dụng:

* Tạo câu đố logic (sắp xếp ký hiệu, mở khóa bằng số).
* Sinh mê cung để kho báu không dễ tìm thấy ngay.

Procedural Generation – Tạo đảo hoang ngẫu nhiên

Dùng khi:

* Muốn bản đồ thay đổi mỗi lần chơi, tránh sự nhàm chán.

Ứng dụng:

* Tạo địa hình đảo bằng Perlin Noise (rừng, sông, núi, hang động).
* Ngẫu nhiên hóa vị trí kho báu, làm mỗi lượt chơi khác nhau.

4. Quản lý sinh tồn & tài nguyên

Hệ thống mô phỏng sinh tồn – Cân bằng tài nguyên

Dùng khi:

* Cần tính toán sức khỏe, năng lượng, thức ăn, nước uống của người chơi.

Ứng dụng:

* Lượng thức ăn giảm dần, buộc người chơi phải săn bắn hoặc hái lượm.
* Khát nước có thể gây chết nếu không tìm ra nguồn nước.
* Quản lý nhiệt độ cơ thể, có thể chết nếu quá lạnh vào ban đêm.