

Báo cáo bài tập lớn 1.

Môn Trí Tuệ Nhân Tạo

Đề tài: Tìm kiếm (Searching).

GVHD: Vương Bá Thịnh.

Thành viên:

Nguyễn Minh Cường-51100465.

Nguyễn Văn Duy-51300610.

Nguyễn Phát Đạt-51300792.

**Yêu cầu:**

Chúng ta có 1 bài toán cho bài tập lớn (BTL) này : Mummy Maze, sinh viên phải hiện thực 3 giải thuật: Depth-first search, Breadth-first search, và Simple Hill Climbing (hoặc 1 giải thuật heuristic khác). Đối với giải thuật Simple Hill Climbing sinh viên cần chọn 1 hàm lượng giá (evaluation function) và đánh giá hiệu năng của giải thuật Hill Climbing so với DFS và BFS.

Sinh viên cần tạo ra 10 input cho bài toán. Sinh viên cần hiện thực có tính năng demo step-by-step có hiển thị trực quan (như game).

Sinh viên phải viết lại báo cáo để nêu rõ quá trình tìm hiểu và hiện thực bài toán này. Trong báo cáo, sinh viên phải trình bày bảng số liệu về thời gian và sự tiêu tốn bộ nhớ của từng giải thuật đối với từng bài toán, và giải thích.

**Tìm hiểu & hiện thực:**

**DFS**

* Sử dụng stack
* Lặp cho đến khi nào mà stack không còn phần tử nào thì thoát khỏi vòng lặp hoặc tìm được goal
* Node là node lá thì sinh ra các trạng thái có thể xảy ra
* Những node mà người chơi bị mummy giết hoặc bị trùng với node cha của nó sẽ bị bỏ qua

**BFS**

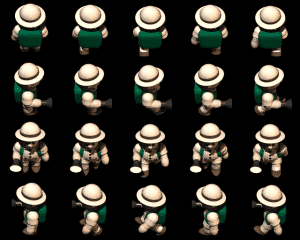
* Sử dụng queue
* Lặp cho đến khi nào mà stack không còn phần tử nào thì thoát khỏi vòng lặp hoặc tìm được goal
* node là node lá thì sinh ra các trạng thái có thể xảy ra
* Những node mà người chơi bị mummy giết hoặc bị trùng với node cha của nó sẽ bị bỏ qua.

**TreeState**

* self.board: Map có kích thước 6x6 sẽ được biểu diễn thành mảng có kích thước 13x13
  + 6x6 sẽ lưu vị trí của player và mummy
  + 7x7 sẽ lưu vị trí của wall và goal
  + Mảng 6x6 và 7x7 sẽ kết hợp xen kẻ với nhau tạo thành mảng 13x13
* self.parent: lưu trữ node cha của node đó. Dùng cho việc
  + Tìm kiếm sol để giải map
  + Kiểm tra trạng thái đó có trùng với trạng thái cha của nó không
* self.playerIsKilled: nếu trạng thái đó dẫn đến player bị mummy giết thì sẽ set thành True
* self.goLeft: lưu trữ trạng thái khi player đi sang trái
* self.goRight: lưu trữ trạng thái khi player đi sang phải
* self.goUp: lưu trữ trạng thái khi player đi lên
* self.goDown: lưu trữ trạng thái khi player đi xuống
* self.noMove: lưu trữ trạng thái khi player không di chuyển
* def playersMove(self, direction): Di chuyển 1 trong 4 hướng (trái, phải, trên, dưới) hoặc không di chuyển
* def mummyMove(self): Dựa vào những bước di chuyển của người chơi mà mummy quyết định bước đi
  + mummy ưu tiên đi theo trục x đến khi xmummy == xplayer thì mummy sẽ di chuyển theo trục y
  + mummy luôn di chuyển về player
* def isDuplicateState(node, parent): kiểm tra node đó có trùng với những trạng thái trước đó không. Để tránh cho những trường hợp lặp vô hạn.

**GUI**

* Các đối tượng player, mummy, wall, goal sẽ sử dụng Sprite Sheet để load từng frame

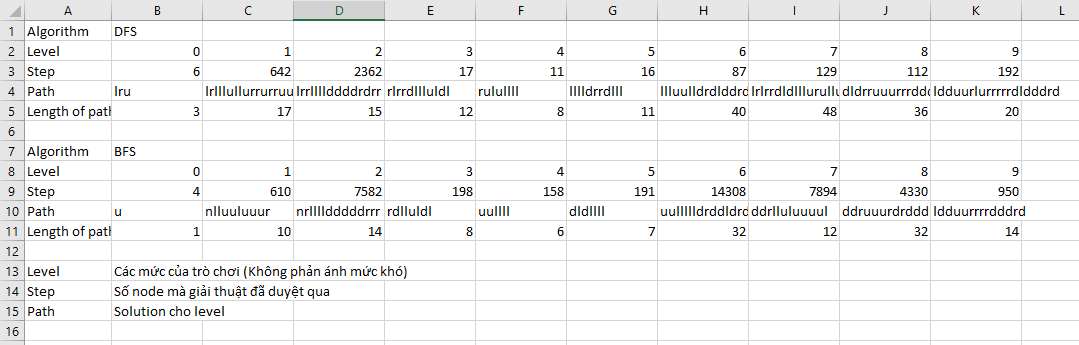


* Map được load từ file txt
  + 'x': wall
  + '-':
  + 'p': player
  + 'm': mummy
  + 'g': goal

**Main.py**

* Load map từ file txt vào mảng 13x13
* Người dùng sẽ chọn giải thuật rồi chọn Run
* Mảng 13x13 sẽ được gọi giải thuật đó với input đầu vào là mảng 13x13
* Giải thuật đó sẽ trả về số bước và danh sách các bước đi
* Player sẽ hiển thị kết quả theo giải thuật đã chọn.

**Bảng số liệu:**



Giải thích:

**Hình ảnh demo minh họa.**

