

Cơ sở Trí tuệ nhân tạo

18_22

BÁO CÁO THỰC HÀNH

Trò chơi 3D TicTacToe (AI)

Hà Văn Duy – 18120339

MỤC LỤC

I. Giới thiệu 2

II. Xây dựng..... 2

 1. Giao diện..... 2

 2. Xử lý sự kiện..... 3

 3. Tìm đường đi với AI 4

III. Mã nguồn và Demo..... 5

I. Giới thiệu

Họ và tên: Hà Văn Duy

MSSV: 18120339

Mô tả sản phẩm:

Game được viết bằng ngôn ngữ HTML + CSS + Javascript và chạy trực tiếp trên trình duyệt.

II. Xây dựng

1. Giao diện

Mã nguồn của giao diện được viết trong file “index.html”, và được trang trí bởi file “style.css”.

Với sự kết hợp của JQuery, mã nguồn của giao diện game và board game được sinh theo cơ chế động sau khi trang được load lên trình duyệt.

Bố cục:



Giao diện trò chơi rất đơn giản, bao gồm board game ở chính giữa là khối lập phương, và các tùy chọn cho trò chơi được cố định ở phía bên trái. Phía dưới còn có một nút để cho phép người chơi bắt đầu game.

Để thực hiện một nước đi trong trò chơi, ta thực hiện di chuột vào một “tầng” bất kì, tầng đó sẽ mở rộng ra cho phép ta chọn chính xác ô muốn đi.



2. Xử lý sự kiện

Các hàm chức năng trong file “script.js” giúp xử lý các thao tác nhấp chuột của người dùng. Đồng thời, trong đó còn chứa các biến toàn cục để lưu board game và thông số cài đặt.

Với phân mã nguồn này, trò chơi đã có thể được chơi ở chế độ người với người.

Tên hàm	Chức năng
ShowMatrix(size)	Thêm mã nguồn giao diện của board game vào “index.html” theo cơ chế động.
Narray(...dimensions)	Khởi tạo ma trận ($n \times n \times n$) chứa giá trị của các ô trong board game.

clearDOMBoard()	Đặt lại trạng thái trò chơi.
StartGame()	Hàm được gọi khi người chơi click vào nút start, dùng để chuyển trạng thái game từ chế độ chờ sang chế độ chơi.
myClick(a, b, c)	Hàm được gọi khi người dùng nhấp vào một ô có tọa độ (a, b, c) trong board game.
play(player, a, b, c)	Hàm có tác dụng đi một nước đi trên giao diện, và cập nhật thông báo trạng thái trò chơi.
successHandler(d)	Cũng là hàm cập nhật giao diện, tuy nhiên được CPU gọi khi thực hiện xong một nước đi.

3. Tìm đường đi với AI

Các hàm chức năng trong file “worker.js” phục vụ cho việc tìm đường đi của CPU. Việc tìm đường này được chạy trong một thread riêng nên không ảnh hưởng tới trải nghiệm của người chơi.

Thuật toán được sử dụng để tìm đường là giải thuật Minimax, có kết hợp với cắt tỉa Alpha-Beta để tối ưu thời gian. Đồng thời, mã nguồn còn được cài đặt với độ sâu duyệt tối đa là $MAX_DEPTH = 4$, nếu chạm đến mức này thì chương trình sẽ gọi hàm heuristic để đánh giá thay vì tiếp tục duyệt xuống dưới giúp giới hạn thời gian duyệt.

Tên hàm	Chức năng
findBestMove(board, playerName)	Tìm một nước đi tốt nhất cho CPU dựa trên giải thuật minimax.
findBestMoveOptimize(board, playerName)	Tìm một nước đi tốt nhất cho CPU dựa trên giải thuật minimax kết hợp với cắt tỉa Alpha-Beta.
checkStateGame(matrix)	Kiểm tra trạng thái của trò chơi, nếu có người thắng cuộc thì trả về tên người chơi thắng.
isMoveLeft(matrix)	Kiểm tra xem còn nước đi nào có thể đi trên bàn không.
heuristic(board, player)	Hàm đánh giá độ tốt của bàn chơi hiện tại cho một người chơi bằng cách đếm số đường đi có thể thắng còn lại của người chơi đó.
minimax(...)	Thuật toán tìm đường đi.
minimaxOptimize(...)	Thuật toán tìm đường đi có cộng thêm cắt tỉa Alpha-Beta.

isEqualPlayerOrNull(player, item)	Kiểm tra giá trị của một ô trong bàn có phải trùng với tên người chơi cho trước hoặc rỗng không.
-----------------------------------	--

III. Mã nguồn và Demo

Mã nguồn của game: <https://github.com/viplazylmht/CubeTicTacToe>

Demo: <https://viplazylmht.github.io/CubeTicTacToe/>