

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

ĐỒ ÁN MÔN HỌC
TRÍ TUỆ NHÂN TẠO

TRÍ TUỆ NHÂN TẠO TRONG TRÒ CHƠI
XE TĂNG BATTLE CITY



GIẢNG VIÊN HƯỚNG DẪN:
PHẠM NGUYỄN TRƯỜNG AN

NHÓM SINH VIÊN THỰC HIỆN:
1. Nguyễn Minh Thư – 16521201
2. Phạm Lê Gia Thịnh – 16521183
3. Võ Lê Minh Trí – 16521291

TP.HCM, tháng 7 năm 2018

MỤC LỤC

| | |
|---|----|
| CHƯƠNG I. TỔNG QUAN VỀ ĐỒ ÁN..... | 2 |
| I.1 Giới thiệu đồ án..... | 2 |
| I.1.1 Trò chơi Bắn xe tăng (Battle City)..... | 2 |
| I.1.2 Mục tiêu của đồ án..... | 2 |
| I.2 Những lớp đối tượng trong trò chơi..... | 2 |
| I.2.1 Lớp “Myrect”..... | 3 |
| I.2.2 Lớp “Time”..... | 3 |
| I.2.3 Lớp “Bonus”..... | 3 |
| I.2.4 Lớp “Bullet”..... | 3 |
| I.2.5 Lớp “Label”..... | 7 |
| I.2.6 Lớp “Explosion”..... | 7 |
| I.2.7 Lớp “Level”..... | 7 |
| I.2.8 Lớp “Tank”..... | 8 |
| I.2.9 Lớp “Enemy”..... | 9 |
| I.2.10 Lớp “Player”..... | 9 |
| I.2.11 Lớp “Game”..... | 9 |
| I.2.12 Một số tùy chỉnh khác..... | 11 |
| CHƯƠNG II. TIẾP CẬN VÀ PHƯƠNG PHÁP..... | 13 |
| II.1 Sơ lược ý tưởng của thuật toán..... | 13 |
| II.2 Tổng quan cấu trúc của file ai.py..... | 13 |
| II.3 Cách thu thập dữ liệu map..... | 14 |
| II.4 Thuật giải tìm đường đi cho Player..... | 14 |
| II.5 Hàm xác định vị trí tương đối của tank player và tank enemy..... | 16 |
| II.6 Thuật giải né đạn cho Player..... | 17 |
| II.7 Tổng thể lối chơi..... | 18 |
| CHƯƠNG III. KẾT QUẢ VÀ ĐÁNH GIÁ..... | 19 |
| III.1 Phân tích định lượng..... | 19 |
| III.2 Đánh giá mặt hạn chế và hướng giải quyết..... | 20 |

CHƯƠNG I. TỔNG QUAN VỀ ĐỒ ÁN

I.1 Giới thiệu đồ án

I.1.1 Trò chơi Bắn xe tăng (Battle City)

Bắn xe tăng (“Battle City”) là một tựa game cổ điển, xây dựng trên nền tảng đồ hoạ đơn giản, có tổng cộng 26x26 ô vị trí trên màn hình của người chơi. Người chơi điều khiển xe tăng của mình bằng các phím điều hướng để di chuyển tìm kiếm kẻ địch và sử dụng một phím được quy định để bắn đạn tiêu diệt địch hoặc phá tường thành cản trở việc di chuyển. Kẻ địch của người chơi gồm nhiều xe tăng tự động được lập trình để tiêu diệt người chơi và tiêu diệt cờ của người chơi. Trò chơi sẽ qua màn chơi mới khi người chơi tiêu diệt được hết toàn bộ tăng địch ở mỗi màn chơi, trong khi đó, trò chơi sẽ kết thúc nếu người chơi bị tiêu diệt hết số mạng của mình hoặc bị bắn mất cờ.

Những chi tiết cụ thể hơn về trò chơi sẽ được đề cập ở phần sau.

I.1.2 Mục tiêu của đồ án

Trong bài đồ án này, nhóm tác giả đặt mục tiêu hiện thực lại trò chơi cổ điển Battle City bằng ngôn ngữ Python. Thông qua trò chơi được cài đặt sẵn, nhóm sẽ có những bước cải thiện mức độ khó của trò chơi đồng thời thử nghiệm những thuật giải được học trong môn “Trí tuệ nhân tạo” vào trò chơi để tạo ra trình điều khiển tự động cho xe tăng của người chơi, cố gắng vượt qua nhiều màn chơi nhất có thể cũng như thử nghiệm thêm một số thuật giải nâng cao khác nằm ngoài chương trình học.

Bài đồ án này được thực hiện nhằm mục đích tìm hiểu cụ thể hơn những nội dung được học trong môn học “Trí tuệ nhân tạo” và cách mang những kiến thức trừu tượng ứng dụng vào thực tế. Đồng thời, bài đồ án cũng là bài tập rèn luyện kỹ năng coding của nhóm tác giả cũng như giúp cho nhóm có thời gian tìm hiểu thêm những kỹ thuật mới, ngôn ngữ lập trình mới và có kinh nghiệm trong việc xử lý những lỗi phát sinh trong quá trình lập trình - thứ mà nhóm tác giả có thể sẽ phải dành cả quãng đời còn lại để đối phó với.

I.2 Những lớp đối tượng trong trò chơi

Bài đồ án này được thực hiện dựa trên mã nguồn có sẵn của game với tên gọi “Remake of Battle City on Python” viết dựa trên ngôn ngữ python.

Trong phần này, để người đọc hiểu rõ được cấu tạo cơ bản của game, nhóm tác giả sẽ đề cập đến các lớp (class) có sẵn trong game cũng như các thuộc tính để thực hiện chức năng của các lớp này, cùng với đó, những tình hình không liên quan đến phần thuật giải điều khiển xe tăng người chơi trong game được nhóm thực hiện cũng sẽ được nêu ra trong phần này.

I.2.1 Lớp “Myrect”

Lớp này được tạo thành với các thuộc tính (left, top, width, height, type) hỗ trợ cho việc tạo các chỉ số cơ bản cho các đối tượng chúng ta tạo ra sau này (gạch, đá, tank,...). Phần này không cần chỉnh sửa bởi vì nó là nền và không ảnh hưởng đến các tính năng, các cài đặt khác về sau mà ngược lại nó còn hỗ trợ khá tốt cho việc cài đặt.

I.2.2 Lớp “Time”

Lớp này tính toán thời gian, tạo lập thời gian xuất hiện các xe tăng, thời gian giữa những phát bắn, phục vụ cho các yêu cầu đòi hỏi thời gian cho các class sau này. Phần này cũng tương tự như myrect, không cần chỉnh sửa bất kỳ cái gì về cấu trúc cũng như cài đặt bên trong nó.

I.2.3 Lớp “Bonus”

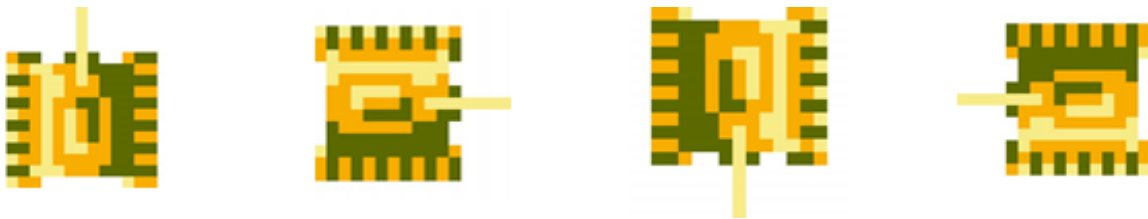
Lớp này tạo ra các hiệu ứng, item được bonus cho tank player khi sử dụng các vật phẩm rơi ra (dùng thời gian, cộng điểm, nổ tank địch, được bảo vệ bởi shield, tăng đạn bắn ra, mạng tank, đóng băng các tank địch). Vì không có yêu cầu thêm hay bớt ứng dụng về phần này nên sẽ không chỉnh sửa gì. Nhưng tank không ưu tiên chạy và chiếm lấy bonus cho dù nó có lợi đến mức nào dù có là bonus làm nổ toàn bộ các tank và hoàn thành nhiệm vụ thì AI vẫn không màng đến mà ưu tiên các mục tiêu khác. Điều này là một điểm tạo bất lợi cho chính AI và tăng lợi thế cho tank địch.

I.2.4 Lớp “Bullet”

Lớp này thì viên đạn được hình thành và sẽ được chia ra làm hai loại là tank địch và tank của player, có các trạng thái như:

- Đạn đang tồn tại (được bắn ra và đang bay)
- Đạn sẽ phát nổ khi gặp mục tiêu hoặc tự nổ khi ở cuối cửa sổ màn hình không gặp mục tiêu nào trên đường bay.

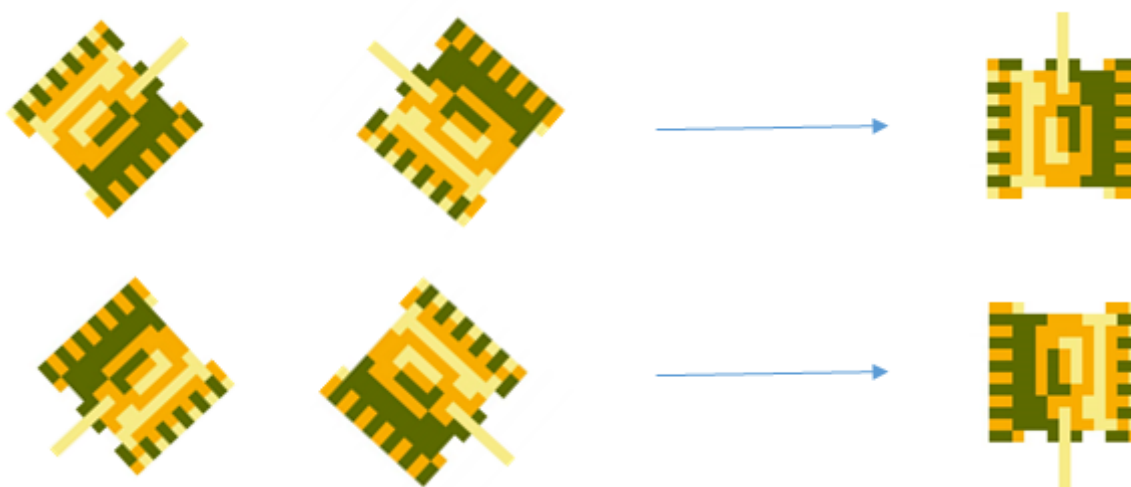
- Đạn được xóa bỏ đi(dùng để xóa hình ảnh viên đạn sau khi rơi vào trạng thái nổ)
- Đạn được cài đặt theo yêu cầu số viên đạn bắn nhiều nhất có thể, tốc độ viên đạn bay, sức công phá của viên đạn, giới hạn đạn bay trong cửa sổ khung hình và nếu đạn của tank địch và tank player chạm nhau thì sẽ phát nổ.
- Những tinh chỉnh thêm: Vì được yêu cầu thêm ứng dụng đi chéo sau kỳ báo cáo đợt đầu tiên, nhóm đã nghiên cứu và tiến hành làm và hoàn thiện thêm được trường hợp đi chéo cho cả tank địch và tank player. Sau khi có một loạt thay đổi và kiểm nghiệm nhóm đã rút ra được một số kinh nghiệm trong việc thêm ứng dụng đi chéo và được nộp trong file tanks_experiment.py . Trong phần chỉnh sửa này cho phép tank đi chéo nên chúng ta cũng phải sửa lại hàm đạn để có thể phù hợp với tính năng đó. Các tank đã có thêm quyền được đi chéo và chính vì vậy đạn phải được quy ước theo một cách riêng theo từng trường hợp. Thực tế vì game thuộc loại đơn giản nên hình ảnh của game tank hay đạn chỉ có 4 hướng đó là: hướng lên, hướng xuống, hướng phải, hướng trái.



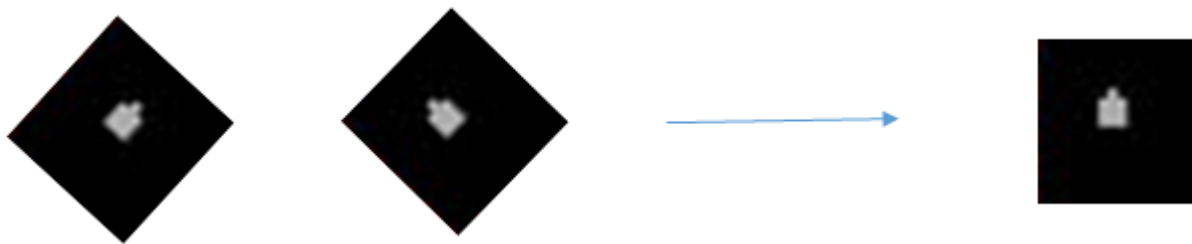
Đạn:



Chính những ví dụ trên cho ta thấy được nếu là xe tank đi chéo mà chỉnh sửa lại tank nghiêng một góc 45 độ và viên đạn cũng chỉnh theo nghiêng một góc 45 độ thì sẽ làm mất đi tính chất của game và đồ họa trông rất kì cục. Vì vậy nhóm đi đến quyết định là đồng nhất hình ảnh và đường đi của tank. Diễn hình như tank đi chéo lên và lệch sang phải thì tọa độ của tank sẽ thay đổi là lên một bậc và chệch sang phải một bậc nhưng riêng về phần hình ảnh ta sẽ cho là tank đang hướng lên (Thực chất có hai sự lựa chọn trong trường hợp này đó là hình ảnh tank sẽ hướng lên hoặc hình ảnh tank sẽ hướng sang phải, nhóm đã tự đưa ra quyết định là chọn hình ảnh tank hướng lên). Cũng tương tự các trường hợp khác, tank đi chéo lên và chệch sang bên phải hoặc bên trái đều là đồng bộ thành hình ảnh tank đang hướng lên, tương tự hình ảnh tank đi chéo xuống và chệch sang phải hoặc trái cũng sẽ được đồng bộ hình ảnh là tank đang hướng xuống. Tương tự đạn cũng vậy, vì hình ảnh tank là hướng khi đi chéo chỉ có là hướng lên hoặc hướng xuống nên viên đạn không thể tự nhiên lại bay chéo màn hình được, nó sẽ chỉ có bay lên nếu đi chéo lên hoặc đạn sẽ xuống nếu tank đi chệch xuống.



Dạn:



Nhóm đã thực hiện thêm vào các thông số: (DIR_UP, DIR_RIGHT, DIR_DOWN, DIR_LEFT) đại diện cho chuyển động lên, dịch sang phải, dịch xuống dưới, dịch sang trái chuyển thành (DIR_UP, DIR_RIGHT, DIR_DOWN, DIR_LEFT, DIR_UPRIGHT, DIR_UPLEFT, DIR_DOWNRIGHT, DIR_DOWNLEFT) đại diện cho các chuyển động lên, dịch sang phải, dịch xuống dưới, dịch sang trái, dịch lên trên chéo sang bên phải, dịch lên trên chéo sang bên trái, dịch xuống dưới chéo sang bên phải, dịch lên xuống dưới chéo sang bên trái bao gồm các việc thay đổi thông số tọa độ địa chỉ tương ứng theo từng trường hợp. Đạn thì chỉ cần di chuyển lên, di chuyển xuống, di chuyển sang trái, di chuyển sang phải nhưng vẫn được cài đặt và cấu tạo lại như cấu trúc của tank là chuyển từ (DIR_UP, DIR_RIGHT, DIR_DOWN, DIR_LEFT) đại diện cho chuyển động lên, dịch sang phải, dịch xuống dưới, dịch sang trái chuyển thành (DIR_UP, DIR_RIGHT, DIR_DOWN, DIR_LEFT, DIR_UPRIGHT, DIR_UPLEFT, DIR_DOWNRIGHT, DIR_DOWNLEFT) đại diện cho các chuyển động lên, dịch sang phải, dịch xuống dưới, dịch sang trái, dịch lên trên chéo sang bên phải, dịch lên trên chéo sang bên trái, dịch xuống dưới chéo sang bên phải, dịch lên xuống dưới chéo sang bên trái bao gồm các việc thay đổi thông số tọa độ địa chỉ tương ứng theo từng trường hợp. Đạn tuy có 4 trường hợp di chuyển nhưng vẫn phải chỉnh thêm và đồng bộ theo giống như tank vì một chút cài đặt nhỏ trong game.

Theo như yêu cầu đầu tiên đạn được chỉnh sửa là có thể đi xuyên qua mọi bức tường, không gây ra các va chạm hoặc phát nổ hủy hoại bất kì (dù đặc tính tường có khác nhau hay sức mạnh của đạn, chỉ bị giới hạn bởi khung cửa sổ và phát nổ kha va chạm vào tank của địch), nhưng sau khi được giáo viên góp ý và giúp chỉnh sửa thì được yêu cầu là đạn cũng có thể bắn xuyên tường địa hình nào đó và có một số loại tường thành địa hình mà tank không thể bay qua được. Đạn không thể xuyên qua tường được chia ra

các trường hợp đó là tường thành mỏng có thể phá hủy(có thể phá hủy cùng lúc một hoặc hai địa hình nhỏ của chướng ngại vật tùy vào đường đi của viên đạn được bắn ra). Một loại địa hình nữa là địa hình không thể phá hủy dù có bắn phá ra sao, bắn góc độ nào, sức mạnh viên đạn, tất cả đều không thể phá hủy địa hình, chướng ngại vật.

I.2.5 Lớp “Label”

Lớp này chỉ đơn giản là cài đặt font chữ cho các chữ sẽ xuất hiện trên màn hình game(vì không có quy định gì nên nhóm lấy font chữ là Arial và cỡ chữ là 13). Phần này nhóm không chỉnh sửa gì bởi nó chỉ là một số cài đặt nhỏ của game không phải trọng tâm đang nhắm đến.

I.2.6 Lớp “Explosion”

Lớp này tạo ra và vẽ các sự kiện các vật thể trong game bị nổ hay phá hủy. Phần này nhóm cũng không thay đổi vì nó đúng tính chất của game khi những vật thể bị tiêu diệt do một lý do nào đó như va chạm hay hiệu ứng nào đó và muốn xóa đi sự hiện diện của sự vật mà chúng ta cần xóa thì điều nhất thiết chúng ta cần phải tạo ra hiệu ứng nổ, hiệu ứng hình ảnh cho người chơi biết được điều đó.

I.2.7 Lớp “Level”

Lớp này thực hiện chức năng chính là load map cho game khi bắt đầu lượt chơi hoặc sau khi xe tăng của người chơi hoàn thành xong một màn chơi nào đó và được chuyển qua màn chơi mới. Ở mỗi màn chơi, map sẽ được tạo ra một cách ngẫu nhiên dựa trên thư viện map. Để thuận tiện cho việc dễ dàng tạo map tùy theo ý muốn của người dùng, mỗi block vật thể trong map được gán với một kí tự, những vật thể và ký hiệu tương ứng cùng thuộc tính của vật thể này trong map gồm:

- Kí hiệu “#” dùng để vẽ khối vật thể là gạch, có thể bị phá bởi đạn có sức mạnh bằng 1 và không thể đi xuyên qua.
- Kí hiệu “@” dùng để vẽ khối vật thể tường đá, không thể bị phá vỡ bởi đạn thông thường những có thể bị bắn phá nếu như nhận được bonus item sức mạnh đạn. Khối này không thể đi xuyên qua.
- Kí hiệu “~” dùng để vẽ khối vật thể là nước, nước có thể bắn xuyên qua nhưng không thể đi xuyên qua được.

- Kí hiệu “%” dùng để gán cho khối cỏ, cỏ có thể đi xuyên qua và bắn xuyên qua, dùng làm nơi ẩn tránh.

Mỗi khối vật thể này có kích thước là 16, có đồ hoạ được tạo từ file ảnh “sprites.gif” trong thư mục “/images”.

Trong quá trình mỗi màn chơi diễn ra, những cập nhật mới như mảng gạch/đá bị phá huỷ cũng sẽ được thực hiện trong lớp này, mỗi lần phá huỷ sẽ phát ra âm thanh phá huỷ đồng thời một block phá huỷ cũng được tạo ra trong 0.5 giây để biểu thị cho việc biến mất của block.

Số lượng tank địch được xuất hiện đồng thời được quy định trong thuộc tính “max_enemies”, qua mỗi màn chơi, số lượng tank địch xuất hiện cùng lúc sẽ được tăng lên 1 đơn vị, cho đến khi số lượng tank xuất hiện cùng lúc là 5. (Nhóm đã giới hạn số lượng tank địch xuất hiện cùng một lúc vì mất nhiều tài nguyên (process) cho việc điều khiển tự động mỗi xe tank địch, do đó số lượng tank địch xuất hiện cùng lúc càng nhiều, thời gian thực hiện thuật giải tìm đường đi cho xe tank sẽ bị chậm lại, dẫn tới một số tình huống xe tank địch bị đứng yên hoàn toàn)

I.2.8 Lớp “Tank”

Tank như đã trình bày và giải thích ở phần trên(mục II.4) có thể di chuyển lên, di chuyển xuống, di chuyển sang trái, di chuyển sang phải, di chuyển lên trên và lệch sang phải, di chuyển lên trên và lệch sang trái, di chuyển xuống dưới và lệch sang phải, di chuyển xuống dưới và lệch sang trái, tạo ra nơi xuất hiện ban đầu cho tank, tạo ra các level cho tank địch, phân biệt tank địch và tank của player và các trạng thái cho tank cũng giống như đạn:

- Tank đang còn sống di chuyển và hoạt động trên bản đồ map game.
- Tank bị chết bởi những phát bắn từ đối thủ gây ra cho mình khi máu của tank giảm xuống 0.
- Tank bị phát nổ sau trạng thái chết.

Khởi tạo lượng máu cho tank, các trường hợp cho tank khi di chuyển, tốc độ di chuyển của tank, số đạn bắn ra, địch hay đồng minh. Tạo ra các hình ảnh biểu diễn sự

xuất hiện của tank. Tank có thể bắn ra những viên đạn và có sức mạnh khác nhau, xoay các hướng khác nhau.

I.2.9 Lớp “Enemy”

Đây là lớp đối tượng tạo ra cho xe tank địch, kế thừa các thuộc tính của lớp “Tank” và có những đặc tính riêng biệt. Ở lớp này, xe tank địch được định nghĩa gồm nhiều loại tank khác nhau, phù hợp với yêu cầu của đề án đề ra để có được sự phong phú về các loại tank địch (tank chạy nhanh, tank thường, tank có lượng chống chịu cao, tank mạnh).

Vị trí xuất hiện của tank địch được quy định ở lớp này, nhóm đã cài đặt để xe tank địch xuất hiện ngẫu nhiên ở 5 vị trí nếu như cài đặt thuộc tính xe tank địch xuất hiện không cùng lúc hoặc xuất hiện đồng thời ở 5 vị trí này nếu như cài đặt thuộc tính các xe tank xuất hiện đồng thời (cùng một lúc). Những vị trí này nằm ở phía trên của bản đồ. Ở lớp này, xe tank địch cũng xác định hướng di chuyển của mình dựa trên hàm “generatePath”. Ở hàm này, xe tank địch ban đầu được khởi tạo một hướng đi random, nhưng sau đó hướng đi tiếp theo của xe tank địch sẽ dựa trên vị trí hiện tại của nó tương đối so với bản đồ và vị trí của cờ địch. Tuy nhiên, do lớp cờ địch đã bị xóa đi nên đôi khi hướng di chuyển của xe tank địch bị ảnh hưởng, gây ra hiện tượng xe tank bị đơ, không di chuyển.

I.2.10 Lớp “Player”

Lớp này tạo ra tank của bản thân player, tạo hình, giao diện, hầu hết đều giống như tạo nên cơ bản của các tank. Điểm khác biệt là hướng đi của xe tank người chơi thay vì được tạo ra một cách random thì được lấy vào từ phím điều hướng của người chơi, ngoài ra, tank của người chơi có thể nhặt bonus và thay đổi thuộc tính (sức mạnh đạn, tốc độ chạy, vòng bảo vệ,...) của mình.

Trong bài đồ án của nhóm, xe tank của người chơi thay vì được điều khiển bởi bàn phím sẽ được định hướng đi và bắn đạn bằng giải thuật tìm đường tiêu diệt địch, sẽ được trình bày cụ thể hơn ở phần II của bài báo cáo.

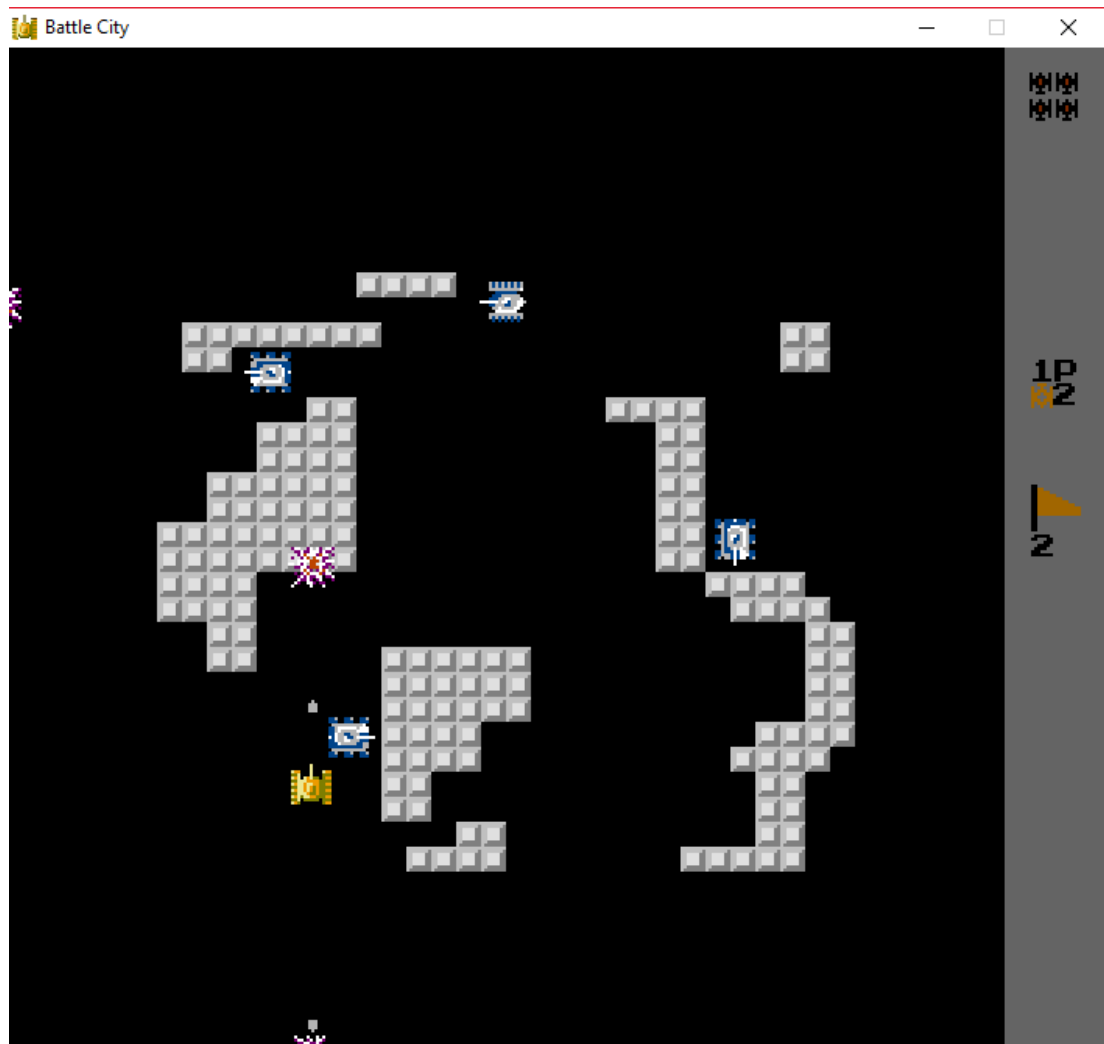
I.2.11 Lớp “Game”

Lớp này tạo lại map và khởi tạo lại cửa sổ màn hình game với các thông số khác để map được to và rộng ra hơn. Cải thiện Map(40*40) theo yêu cầu của đề án so với ban

đầu là Map(26*26), tự tạo lại cho map rộng ra với các địa hình khác nhau mà mỗi level sẽ xuất hiện sau khi qua màn (chiến thắng màn trước). Cửa sổ được nới rộng ra từ trước kia là screen(480*416) thành screen(704*640). Tạo nền các giao diện cho game, các âm thanh mà trò chơi yêu cầu để tạo ra sự chân thực cho game (tiếng đạn bắn ra, đạn nổ, tiếng chướng ngại vật bị phá hủy hay chướng ngại vật phá hủy đạn, ăn bonus,...), thực hiện hóa các font chữ và nền các font chữ đó (chữ bằng gạch, chữ thường) xuất hiện ra lúc mở đầu game và một số trường hợp khác nữa, độ cao, rộng, tọa độ các thành phần trong giao diện. Hoàn thành các bước tạo ra các sự kiện (bonus, tạo shield, spawn tank địch và khởi tạo tank mình, kết thúc game, tính toán và show điểm). Ví dụ minh họa được đưa ra bởi chính những hình ảnh được chụp từ trong game ra:



Hình 1. Game Battle city trước khi cải tạo(26x26)



Hình 2. Game Battle city sau khi cải tạo(40x40)

I.2.12 Một số tùy chỉnh khác

Ngoài các tùy chỉnh đã được thực hiện trong nội bộ của các lớp thuộc tính, nhóm còn mang một số biến thuộc tính ra toàn cục (ở phía trên cùng của file tanks.py chính) để người sử dụng có thể dễ dàng tùy chỉnh theo ý muốn của mình, những thuộc tính đó có những ý nghĩa như sau:

- Thuộc tính “allspeed”: quy định tốc độ của toàn bộ nhân tố trong game (xe tank địch, xe tank người chơi, đạn). Ở giá trị 1, thuộc tính này cài đặt game với tốc độ mặc định. Khi tăng tốc toàn cục của game lên cao, đôi khi sẽ sinh ra các lỗi, nguyên nhân là do xe tank địch và xe tank người chơi được điều khiển bởi thuật giải tìm đường. Các gợi ý hướng đi từ thuật giải được đưa vào hàng đợi, nếu tốc

độ game diễn ra nhanh hơn kết quả từ thuật giải tìm đường, hàng đợi sẽ bị rỗng và xe tank sẽ không có hướng đi cụ thể, gây ra lỗi. Ở thuộc tính này, nhóm khuyến nghị người dùng để cài đặt thuộc tính ở tốc độ lớn nhất là 2.

- Thuộc tính “max_enemies”: quy định số lượng tank địch xuất hiện tối đa cùng một lúc trong game (khác với số lượng mạng của tank địch trong game, được quy định hàm “nextLevel” của lớp Game), thuộc tính này quy định mức độ khó của mỗi màn chơi, giá trị được gán cho max_enemies là giá trị số lượng tank địch xuất hiện ở màn chơi đầu tiên, giá trị này sẽ tự động tăng dần (+1) qua mỗi màn chơi tiếp theo. Tuy nhiên, nhóm đã giới hạn số lượng tank địch xuất hiện cùng một lúc lại là 5 vì mỗi xe tank địch yêu cầu một số lượng process nhất định để xác định hướng đi cho xe tank của mình. Số lượng xe tank quá nhiều sẽ làm game mất ổn định vì số lượng process yêu cầu quá nhiều, và đôi khi gây ra hiện tượng hàng đợi rỗng như trình bày ở trên. Giá trị mặc định của thuộc tính này là 3.
- Thuộc tính “Enemy_exist_sametime”: thuộc tính này quy định việc bắt đầu mỗi màn chơi, xe tank địch sẽ xuất hiện cùng một lúc (giá trị True) hay xuất hiện lần lượt (giá trị False). Giá trị mặc định của thuộc tính này là False, tức là xe tank địch xe xuất hiện lần lượt tại các vị trí định sẵn.
- Thuộc tính “mode_cross”: Thuộc tính này xuất hiện trong file tanks_experiment.py, điều chỉnh chế độ đi chéo trong game. Vì vấn đề đi chéo chưa được giải quyết triệt để và còn để lại một số tình huống không mong muốn nên hiện tại nhóm không để thuộc tính này trong file chính thức. Người dùng có thể chạy thử file này để quan sát việc di chuyển chéo của các đối tượng trong game.

CHƯƠNG II. TIẾP CẬN VÀ PHƯƠNG PHÁP

II.1 Sơ lược ý tưởng của thuật toán

Di chuyển: Dùng thuật giải A* để định hướng di chuyển của tank người chơi đến gần hơn hoặc cùng hướng với tank địch. Tạo điều kiện thuận lợi nhất cho tank người chơi để bắn đạn, vì số lượng đạn của tank người chơi có hạn nên không thể bắn liên tục hoặc lung tung.

Né đạn: dựa vào đường di chuyển bằng thuật giải A*, xem xét các trường hợp đạn ảnh hưởng tới tank người chơi. Nếu đạn quá gần với tank người chơi không thể né thì tank người chơi sẽ bắn trả, ngược lại đạn ở một khoảng cách nhất định so với tank người chơi thì tank sẽ di chuyển sang hướng ngược lại để né đạn.

Thu thập dữ liệu map: bao gồm hình dạng map, tổng số tank địch, vị trí của tank địch, hướng di chuyển của tank địch, vị trí và hướng đạn của địch ... để phục vụ cho việc xử lý di chuyển cũng như né đạn của tank người chơi.

II.2 Tổng quan cấu trúc của file ai.py

Bao gồm các hàm :

- Định nghĩa Priority Queue
- Lớp “ai_agent”:
 - + Định nghĩa operations dùng để điều khiển di chuyển và bắn đạn liên kết với file tank.py. Hàm này giúp ta định nghĩa tổng quan lối chơi của tank người chơi
 - + Định nghĩa hàm Get_mapInfo
 - + Định nghĩa hàm Update_Strategy: đây là hàm điều khiển di chuyển và bắn đạn. Hàm này được gọi liên tục trong quá trình tank người chơi đi thực hiện nhiệm vụ.
 - + Định nghĩa giải thuật A*
 - + Định nghĩa hàm manhattan_distance dùng để tính khoảng cách theo kiểu manhattan
 - + Định nghĩa hàm eulidean_distance tính khoảng cách theo kiểu eulidean
 - + Định nghĩa hàm find_neighbor là hàm con của giải thuật A* giúp tìm ra vị trí hợp lệ gần với vị trí ban đầu

+ Định nghĩa hàm `inline_with_enemy` hàm xác định vị trí tương đối của tank người chơi với tank địch

+ Định nghĩa hàm `bullet_avoidance` hàm kết hợp né đạn và di chuyển cho tank người chơi

II.3 Cách thu thập dữ liệu map

Từ ý tưởng ban đầu với nhóm đã tạo ra hàm `get_mapinfo` với

- *Input: Hiện trạng map hiện tại*
- *Output: Thông tin của tất cả thành phần của map*

Với các đối tượng chính là bullets, enemies, tile, player dùng cấu trúc dữ liệu list với tên `mapinfo` gồm 4 chỉ mục từ 0 đến 3 cho lần lượt các đối tượng trên. Với mỗi đối tượng trên có những đối tượng nhỏ hơn cụ thể:

- Bullets: rect, direction, speed
- Enemies: rect, direction, speed, type
- Tile: rect, type
- Player: rect, direction, speed, `Is_shielded` (với rect bao gồm left, top, width, height)

Lần lượt duyệt các đối tượng lớn ở trên hiện trạng hiện tại của map rồi append list `mapinfo`. Trả về list `mapinfo` (output của hàm `get_mapinfo`)

Kết hợp với xử lý đa luồng multiprocessing của python để tăng tốc độ xử lý (vì hiện trạng map thay đổi liên tục) vừa cập nhật thông tin map (`get_mapinfo`) vừa ra lệnh cho tank người chơi di chuyển và bắn đạn.

Quá trình thu thập dữ liệu map sẽ được thực hiện liên tục khi chương trình còn chạy.

II.4 Thuật giải tìm đường đi cho Player

Dùng thuật giải A* để định hướng cho tank người chơi.

Input: Vị trí ban đầu, vị trí đích, tốc độ di chuyển của tank người chơi

Output: hướng di chuyển tiếp theo để tiếp cận vị trí đích

Khởi tạo ban đầu

Biến bắt đầu `start = (start.x, start.y)`

Biến đích goal = (goal.x,goal.y)

Priorityqueue: frontier = PriorityQueue() với cấu trúc priority queue là cấu trúc queue với phần tử đầu tiên nhỏ nhất được định nghĩa trước

Mảng came_from lưu vị trí trước đó (cha)

Mảng cost_so_far lưu giá trị độ dài của vị trí kế tiếp

While not frontier.empty()

Khởi tạo current = frontier.top();

Nếu current == goal thì break

Xét tất cả các vị trí lân cận điểm đang xét: phía trên, phía dưới, bên trái, bên phải

Điều kiện xét: + Nếu đụng độ với đích thì dừng

+ Nếu đụng độ với tường (trừ có) thì dừng

Nếu thỏa mãn tất cả điều kiện trả về điểm được phép đi (next)

Tính giá trị new_cost = cost_so_far[current]+speed trong đó speed là tham số tốc độ di chuyển

Nếu next không thuộc trong mảng cost_so_far hoặc có giá trị cost lớn hơn

Cập nhập lại giá trị cost_so_far[next] = new_cost

priority = new_cost + self.heuristic(goal, next)

frontier.put(next, priority)

came_from[next] = current

với hàm heuristic là hàm manhattan_distance tính khoảng cách giữa next và goal

khởi biến next = None

khởi tạo dir_cmd = None

Đi ngược trong mạng came_from để tìm vị trí trước (cha)

While current !=start

Next = current

Current = came_from[current]

If next

If current.y > next.y

Dir_cmd=0 đi lên trên

elif current_top < next_top:

dir_cmd = 2 đi xuống dưới

elif current_left > next_left:

dir_cmd = 3 đi sang trái

elif current_left < next_left:

dir_cmd = 1 đi sang phải

return dir_cmd

với move_dir: 0:Up 1:Right 2:Down 3:Left 4:None

Định nghĩa hàm heuristic là hàm manhattan_distance

Input: hai vị trí cần tính a,b

Output: khoảng cách manhattan

Trả về $|a.x - b.x| + |a.y - b.y|$

Định nghĩa hàm euclidean_distance

Input: hai vị trí cần tính a,b

Output: khoảng cách euclidean_distance

Trả về $\sqrt{(a.x-b.x)^2 + (a.y-b.y)^2}$

II.5 Hàm xác định vị trí tương đối của tank player và tank enemy

Input: Player rect, enemy rect

Output: xuất ra 1 số tương ứng vị trí của enemy so với player. Số 0 là ở trên, 1 là bên phải, 2 là ở dưới, 3 là bên trái tương tự với move_dir.

Mục đích để xác định tank người chơi nằm ở phía nào so với tank địch, nằm ở trên, dưới hay bên trái, phải.

Bằng cách xác định vị trí chính giữa của tank người chơi so sánh với vị trí của tank địch. Vì trong tự game này 1 con tank bao gồm nhiều khối gộp lại với nhau và vị trí chính giữa chính là trung bình cộng của bên trái và phải hoặc trên và dưới

- Nếu điểm chính giữa player tank nằm ở giữa 2 thuộc tính rect_left và rect_right của enemy tank và enemy tank có rect_bottom > rect_top player tank rect top thì sẽ xác định được rằng enemy tank đang ở phía trên của player tank. Ngược lại nếu rect_bottom < rect_top player tank xác định được rằng enemy tank đang ở phía dưới của player tank.
- Nếu player tank nằm ở giữa 2 thuộc tính rect_top và rect_bottom của enemy tank và enemy tank có rect_left > rect_right player tank rect top thì sẽ xác định được rằng enemy tank đang ở phía bên trái của player tank. Ngược lại nếu rect_left < rect_right player tank xác định được rằng enemy tank đang ở phía phải của player tank.

II.6 Thuật giải né đạn cho Player

Input: thông tin của player, thông tin đạn, hướng di chuyển của A, vị trí tương đối của địch*

Output: giúp xe tank người chơi tránh đạn

Khởi tạo:

- Mảng direction lưu hướng di chuyển
- Player_rect = thông tin vị trí của người chơi
- Khởi tạo mảng sorted_bullet_info_list lưu dữ liệu đã sắp xếp thông tin của tất cả các viên đạn đang có trên bản đồ bằng khoảng các viên đạn tới tank người chơi. Với hàm tính khoảng cách là euclidean distance, ở đây nhóm dùng hàm euclidean distance thay vì manhattan distance vì euclidean cho khoảng cách thẳng tới viên đạn, viên đạn có khoảng cách nhỏ hơn sẽ có mức độ nguy hiểm cao hơn nhiều.
- Khởi tạo biến shoot để điều khiển bắn đạn.
- Xét mảng sorted_bullet_info_list lấy ra viên đạn gần nhất với player tank
- Nếu như nó nhỏ hơn khoảng cách 150 (số 150 là số ngẫu nhiên nhóm chọn có thể thay đổi thông số này) thì lấy thông tin chi tiết của viên đạn bao gồm rect và hướng bắn. Rồi tiếp tục xét nếu như viên đạn đã quá gần so với người tank người chơi (cách người chơi 1~2 ô trên bản đồ) thì bắt buộc phải bắn trả vì không còn thời gian di chuyển né, ngược lại nếu viên đạn có khoảng cách đủ để tank di chuyển né thì ra lệnh cho tank người đi di chuyển ngược với hướng đạn bắn hoặc vị trí tương đối của nó có cùng với hướng di chuyển sắp tới của tank người chơi hay không, nếu như trùng thì bắn để tiêu diệt tank địch.
- Nếu như khoảng cách của viên đạn gần nhất với player tank lớn hơn 150 thì xét. Đồng thời xét hướng đi của viên đạn có cùng với hướng di chuyển của tank người chơi hay không nếu như trùng thì xóa nó khỏi mảng direction
- Dựa vào mảng direction sẽ xuất ra hướng đi cho tank player.

II.7 Tổng thể lối chơi

Ưu tiên diệt những tên địch gần với tank người chơi nhất. Di chuyển đến gần tên địch hơn và bắn tiêu diệt nó. Trên đường di chuyển sẽ kết hợp với tìm kiếm đường đi tốt nhất và né đạn nguy hiểm .

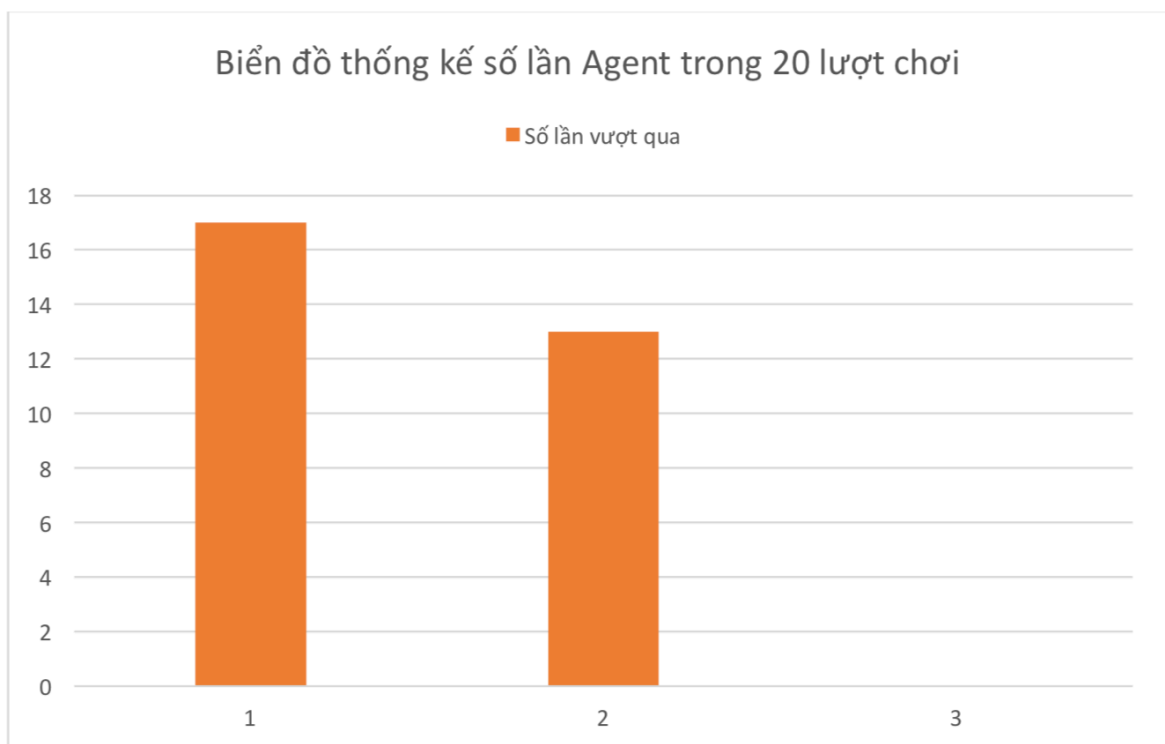
Sắp xếp tất cả các tên địch đang có trên bản đồ từ gần nhất đến xa nhất so với tank người chơi, với hàm tính khoảng cách Manhattan distance

Lấy ra thông tin của tên địch gần nhất sau đó tính vị trí tương đối của địch với tank người chơi, tính đường đi tiếp cận địch bằng A* sau đó kết hợp vừa di chuyển vừa né đạn.

CHƯƠNG III. KẾT QUẢ VÀ ĐÁNH GIÁ

III.1 Phân tích định lượng

Dựa trên mô hình game và Trí tuệ nhân tạo được cài đặt từ hai chương trước, nhóm đã thực hiện thử nghiệm 20 lần khác nhau để xem khả năng ứng biến để chiến thắng của Agent qua các màn chơi như thế nào và có thể qua được bao nhiêu màn. Kết quả thử nghiệm mặc dù không thoả mãn kì vọng của nhóm với khả năng vượt qua màn 2 của Agent là khoảng 65%, tuy nhiên Agent hầu như chưa thể vượt qua màn thứ 3 của game. Kết quả thử nghiệm được thể hiện cụ thể ở biểu đồ dưới đây



Thống kê số lượng màn chơi vượt qua trong 20 lần thử

Kết quả trên có thể giải thích bởi việc qua mỗi màn chơi, số lượng quân địch ngày càng nhiều và nhiều quân địch có sức mạnh hơn, trong khi Agent có số mạng ngày càng giảm vì không thể giữ nguyên số mạng qua các màn chơi trước. Quá trình đo lường thử nghiệm này nhóm đã cố tình tăng tốc độ toàn cục của game lên gấp 2 lần để có thể nhanh chóng quan sát và ghi nhận lại kết quả. Hướng tiếp cận này có thể đã ảnh hưởng đến kết quả đo đạt vì khi tốc độ tăng đòi hỏi máy tính phải xử lý nhanh

hơn làm Agent đôi khi hoạt động không được ổn định, một số lỗi cũng phát sinh trong quá trình này mà nhóm vẫn đang tìm kiếm cách khắc phục.

III.2 Đánh giá mặt hạn chế và hướng giải quyết

Trong quá trình thực hiện cài đặt và chạy thử thuật toán như đã nêu ở chương II, nhóm phát hiện ra một số mặt hạn chế của Agent khi tuân thủ một cách cố định thuật giải A^* và thuật giải né đạn, một vài mặt hạn chế đó gồm:

- Khi Agent sử dụng thuật giải A^* để tìm kiếm đối phương, nếu đối phương đang chạy song song và cùng hướng với Agent, Agent sẽ đuổi theo đối phương đó ở hướng song song trong khi đối phương đang đi ở phía trước, do đó sẽ không thể gặp được đối phương để có thể tiêu diệt nếu đối phương không chuyển hướng. Trong một số trường hợp, tank địch trong game chỉ chạy tiến tới và tiến lùi, ở trường hợp này, Agent bị động chạy theo tank địch trong một khoảng thời gian dài gây nên việc không hiệu quả và tăng khả năng bị tiêu diệt bởi các tank ở hướng khác.

- Ở một số trường hợp khác, Agent bị động đứng yên do gặp vật cản và hướng đi kế tiếp liên tục là đi vào hướng vật cản (trong khi việc cải tạo lại game không cho phép Agent phá hủy đi gạch). Việc này làm cho game bị gián đoạn và không thể tiếp tục. Trường hợp này xảy ra cũng chủ yếu là do phía tank địch không di chuyển nên không thể tiêu diệt Agent để trở về vị trí ban đầu và tính toán lại. Hai vấn đề điển hình nêu trên gây ra nguyên nhân chính bởi việc Agent quá tuân thủ theo thuật giải được cài đặt trong khi những thuật giải Trí tuệ nhân tạo chưa được tinh chỉnh để phù hợp với môi trường chơi (các vật cản, điểm mù, ...), trong khi mỗi màn chơi lại có một cấu trúc địa hình khác nhau, nên khó có thể tinh chỉnh thuật toán cho phù hợp với cụ thể từng màn được. Ở vấn đề này, nhóm dự định sẽ áp dụng thêm một hàm để phát hiện ra các di chuyển liên tục không tạo ra kết quả (như giết địch) trong một thời gian dài, ngay sau khi phát hiện, Agent sẽ ngừng tuân theo giải thuật đã cài đặt trước đó mà thực hiện chạy ngẫu nhiên trong một vài nước trước khi quay lại cập nhật các thuật giải và sử dụng các thuật giải để tiếp tục quyết định các bước đi.