

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 6 năm 2018

CHƯƠNG I. GIỚI THIỆU CÀI ĐẶT GAME

I.1. Giới thiệu chung:

Game bắn tank là game giải trí, người chơi điều khiển xe tank của mình với mục đích là bảo vệ thành công tính mạng, nhà chính của mình và tiêu diệt được hết các địch thủ. Sau khi thông qua nâng cấp của nhóm thì nó không còn cần phải bảo vệ nhà chính nữa mà chỉ cần bảo vệ được chính bản thân mình và tiêu diệt hết tank địch để qua màn. Game được tạo thành bởi 11 class gồm có :

I.2. Myrect: 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,...)

I.3. Time: 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.

I.4. Bonus: 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 sheld,tăng đạn bắn ra, mạng tank, đóng bang các tank địch)

I.5. Bullet: 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.

I.6. Label: cài đặt font chữ cho các chữ sẽ xuất hiện trên màn hình game.

I.7. Explosion: Tạo và vẽ các sự kiện các vật thể trong game bị nổ hay phá hủy.

I.8. Level: Load map cho game khi xe hoàn thành xong màn chơi nào đó ở level phía trước, tạo số tank địch tối đa ở level đó, tạo các thành phần load và vẽ các vật thể như gạch, đá, nước, rừng, băng, mỗi tile_size là 16. Chuyển và tạo ra hình ảnh, giao diện cho các vật thể trong game. Số level max là 35 nhưng khi vượt qua 35 màn và lên các màn cao hơn thì nó sẽ chuyển về lại các map của level đầu. Tạo nên những âm thanh mà viên đạn được phá hủy khi va chạm.

I.9. Tank: Tank có thể di chuyển lên xuống trái phải, 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.10. Enemy: Tạo các sự khác biệt cho tank địch và tank mình, tạo ra các loại tank(tank chạy nhanh, tank thường, tank có lượng chống chịu cao, tank mạnh), nó được khởi tạo cho xuất hiện một cách random, tạo hình giao diện cho các tank enemy này, có thể di chuyển lên xuống trái phải như tank player

I.11. Player: Tạo ra tank của bản thân player, tạo hình, giao diện, hầu hết đều giống như tạo nền cơ bản của các tank, là tank có thể nhặt bonus.

I.12. Game: 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) so với ban đầu là Map(26*26), tự tạo lại cho map rộng ra với các địa hình. Cửa sổ được mở rộng ra từ trước kia là screen(480*416) thành screen(704*640). Tạo nền các giao diện cho game, các âm thanh, thực hiện hóa các font chữ và nền các font chữ đó(chữ bằng gạch, chữ thường), độ cao, rộng, tọa độ các thành phần trong giao diện. Tạo các sự kiện bonus, tạo shield, spawn tank địch và khởi tạo tank mình, kết thúc game, tính toán và show điểm.



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



Hình 2. Game BattleCity trước sau cải tạo (40x40)

CHƯƠNG II. HƯỚNG TIẾP CẬN VÀ PHƯƠNG PHÁP

II.1. Thuật giải tìm đường cho Player

Xe tăng địch thường xuất hiện ở phần phía trên bản đồ, kiểm tra xem chúng có nằm trong bán kính ở giữa hay không, nếu có sẽ ưu tiên tiêu diệt trước.

Đầu tiên sẽ sắp xếp các xe tăng địch ưu tiên phải bị giết từ trung tâm. Dùng thuật giải A* với hàm heuristic do khoảng cách nữa xe tăng địch với playertank bằng cách sử dụng hàm Manhattan-distance.

Giải thuật A* tìm đường đi cho player tank:

Khởi tạo biến bắt đầu $start = (start.x, start.y)$

Khởi tạo biến đích $goal = (goal.x, goal.y)$

Khởi tạo priorityqueue: $frontier = PriorityQueue()$

Mảng $came_from$ lưu vị trí trước đó (cha)

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

Mã giả:

```
While not frontier.empty()
```

```
    current = frontier.top();
```

```
    Nếu current == goal thì break
```

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

```
    Điều kiện xét:      + Nếu đụng độ với địch thì dừng
```

```
                      + Nếu đụng độ với tường ( trừ cỏ ) 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

```
next = None
```

```
dir_cmd = None
```

```

While current !=start
    Next = current
    Current = came_from[current]
    If next
        If current.y > next.y
            Dir_cmd=0
        elif current_top < next_top:
            dir_cmd = 2
        elif current_left > next_left:
            dir_cmd = 3
        elif current_left < next_left:
            dir_cmd = 1
    return dir_cmd
với move_dir:          0:Up 1:Right 2:Down 3:Left 4:None

```

II.2. Thuật giải né đạn cho Player

II.2.1 Hàm xác định vị trí tương đối của enemy tank và player tank

- Nếu 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.2.1 Hàm né đạn

Sắp xếp thông tin của tất cả đạn đang có ở trên bản đồ bằng khoảng cách từ viên đạn đến player tank, với hàm heuristic tính khoảng cách euclidean distance.

Dựa vào danh sách đạn ở trên lọc ra viên đạn gần nhất – gây nguy hiểm nhất gọi là viên đạn a, với hàm khoảng cách đến player tank là $e(x)$.

Nếu như $e(a) < 120$ (gần với player tank)

Lấy thông tin của viên đạn bao gồm hướng bắn là rect. Nếu viên đạn đang cực gần với player tank $e(a) < 5$ và viên đạn đang cùng hướng di chuyển thì không thể kịp di chuyển để né thì phải bắn trả để né đạn. Nếu $5 < e(a) < 25$ thì player tank sẽ có thời gian để đổi hướng di chuyển để né.

Mã giả:

Tạo mảng sắp xếp tất cả các viên đạn đang có trên map với khoảng cách đến player tank.

Hàm heuristic tính khoảng cách là euclidean distance

Xuất ra viên đạn có khoảng cách nhỏ nhất (min_dist_with_bullet)

if min_dist_with_bullet <= 120

Lấy thông tin của viên đạn: vị trí, hướng bắn

if abs(bullet_rect.centerx - player_rect.centerx) <= 25:

if abs(bullet_rect.centerx - player_rect.centerx) <= 5:

if bullet_direction == 0 and bullet_rect.top > player_rect.top:

directions.append(2)

shoot = 1

if bullet_direction == 2 and bullet_rect.top < player_rect.top:

directions.append(0)

shoot = 1

else

if bullet_rect.left > player_rect.centerx:

directions.append(3)

else:

directions.append(1)

elif abs(bullet_rect.centery - player_rect.centery) <= 25:

if abs(bullet_rect.centery - player_rect.centery) <= 5:

if bullet_direction == 1 and bullet_rect.left <

player_rect.left:

directions.append(3)

shoot = 1

if bullet_direction == 3 and bullet_rect.left >

player_rect.left:

```

        directions.append(1)
        shoot = 1
    else:
        if bullet_rect.top > player_rect.centery:
            directions.append(0)
            directions.append(2)
        else:
            directions.append(2)
            directions.append(0)
với move_dir:          0:Up 1:Right 2:Down 3:Left 4:None

```

II.3. Thuật giải kết hợp xác định hướng di chuyển và bắn đạn cho Player

Kết hợp hàm xác định vị trí tương đối ở II và đường đi của thuật giải A* ở I khi vị trí tương đối của enemy tank == hướng đi của player tank thì sẽ bắn để tiêu diệt enemy tank.

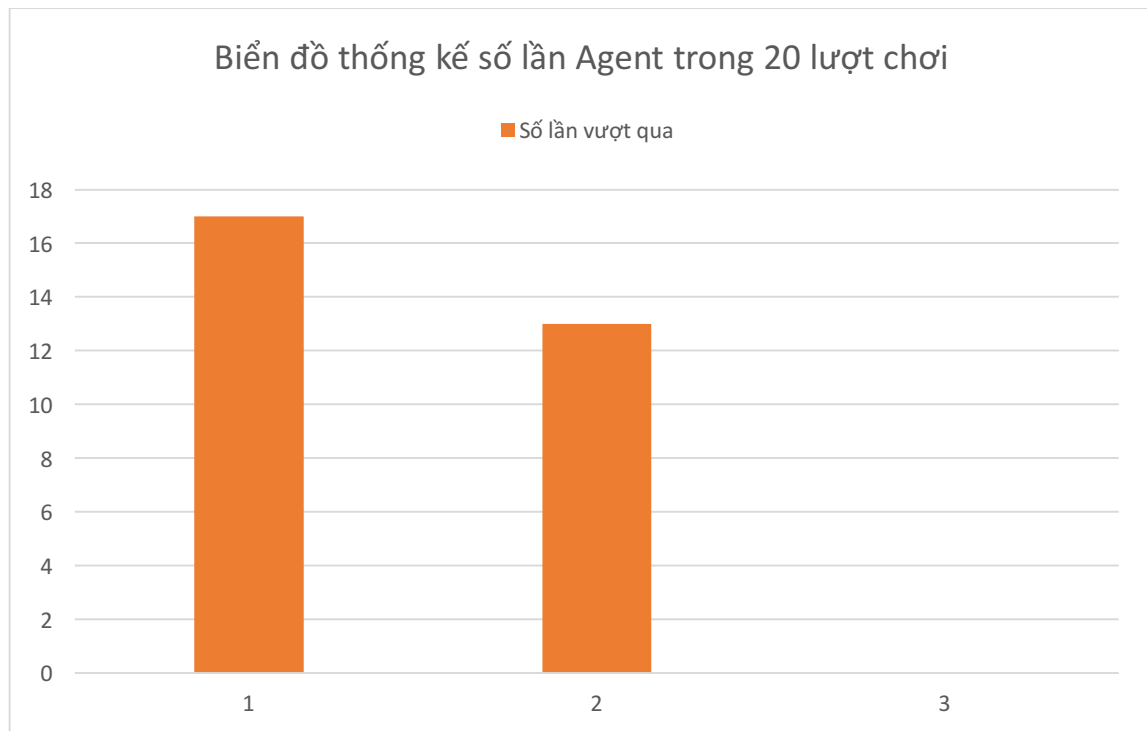
Kết hợp giữa hàm né đạn và đường đi của thuật giải A*, khi hướng đạn của các enemy tank (không phải enemy gần nhất) gần với hướng đi của player tank thì sẽ xuất hiện hàm cảnh báo, gỡ bỏ bước di chuyển đó và thực hiện lại A*

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



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.

NGUỒN THAM KHẢO

- [1] <https://www.pygame.org/project-BattleCity+Tanks-1614-.html>, Raitis G, Remake of BattleCity in Python.
- [2] <http://www.cs.huji.ac.il/~ai/projects/2015/BattleCity/>, Itai Segev and Shlomi Shasha, BattleCity AI Project.