**BỘ GIÁO DỤC & ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

--🙢🕮🙠--

**A picture containing text, device, gauge

Description automatically generated**

**BÁO CÁO**

**MÔN HỌC: Phát triển phần mềm mã nguồn mở**

***Đề tài: Trò chơi bắn xe tăng cổ điển***

***Giảng viên: Từ Lãng Phiêu***

***Lớp: DKP1201 Thực hiện: Nhóm 11***

***TP.HCM, ngày 19 tháng 4 năm 2023***

**DANH SÁCH THÀNH VIÊN NHÓM 11**

|  |  |
| --- | --- |
| **MSSV** | **Họ tên** |
| 3120410046 | Đặng Chí Bảo |
| 3120410412 | Nguyễn Thiên Phúc |
| 3120410034 | Nguyễn Tuấn Anh |
| 3120410030 | Nguyễn Đức Anh |

**MỤC LỤC**

**PHẦN 0. TỔNG QUAN GIAO DIỆN**

1. **Các thành phần trên giao diện**
2. **Phác thảo giao diện**
3. **Tạo đối tượng button và xử lí sự kiện**

**PHẦN 1: CHẾ ĐỘ TIÊU DIỆT**

**CHƯƠNG 1. GIỚI THIỆU**

**CHƯƠNG 2. HƯỚNG DẪN CÁCH CHƠI**

**CHƯƠNG 3. Ý TƯỞNG THỰC HIỆN**

**CHƯƠNG 4. MÔ HÌNH HÓA CHƯƠNG TRÌNH**

1. Cơ chế quá trình hoạt động.

2. Mô tả chi tiết các thành phần cần xử lí

**CHƯƠNG 5. QUY TRÌNH THỰC HIỆN CHƯƠNG TRÌNH**

1. Thư viện cần dùng.

2. Căn chỉnh cửa sổ.

3. Đưa vật thể lên màn hình.

4. Định dạng xe tăng.

5. Tạo chướng ngại vật (các ô tường).

6. Điều khiển xe tank của người chơi.

7. Tạo giới hạn di chuyển (khi va vào tường và rìa map).

8. Thao tác với đạn của xe tăng của người chơi.

9. Định dạng xe tăng địch.

10. Xe tăng địch tự động tính toán và di chuyển.

11. Xe tăng địch tự động tính toán bắn đạn.

12. Tăng số lượng xe tăng quân địch từ một lên sáu.

13. Trò chơi kết thúc.

14. Thêm tiếng nhạc nền, bắn đạn, nổ.

**PHẦN 2. CHẾ ĐỘ ĐỐI KHÁNG**

**CHƯƠNG 1. GIỚI THIỆU**

**CHƯƠNG 2. HƯỚNG DẪN CÁCH CHƠI**

**CHƯƠNG 3. Ý TƯỞNG THỰC HIỆN**

**CHƯƠNG 4. MÔ HÌNH HÓA CHƯƠNG TRÌNH**

1. Cơ chế quá trình hoạt động
2. Mô tả chi tiết các thành phần cần xử lí

**CHƯƠNG 5. QUY TRÌNH THỰC HIỆN CHƯƠNG TRÌNH**

1. **Thư viện cần dùng**
2. **Cài đặt các biến**
3. **Định dạng và đưa hình ảnh lên cửa sổ**
4. **Tùy chỉnh bố cục của của sổ.**
5. **Cơ chế điều khiển của của hai xe tăng**

a. Điều khiển xe tăng xanh

b. Điều khiển xe tăng đỏ

1. **Cơ chế xử lí đạn của cả hai xe tăng**
2. **Thông báo kết thúc trò chơi**
3. **Xử lí các sự kiện chính và các trường hợp xảy ra trong lúc chơi (hàm Main)**

a. Một số biến và list cơ bản để dễ thao tác

b. Cơ chế bắn đạn của cả hai xe tăng

c. Xử lí sự kiện bị bắn và đếm mạng

d. Kết thúc trò chơi bằng thông báo chiến thắng

e. Tự động reset lại trận đấu khi kết thúc

f. Thêm âm thanh, hiệu ứng tạo cảm giác chân thật, sống động.

**PHẦN 3: CHẾ ĐỘ SINH TỒN**

**CHƯƠNG 1. GIỚI THIỆU**

**CHƯƠNG 2. HƯỚNG DẪN CÁCH CHƠI**

**CHƯƠNG 3. Ý TƯỞNG THỰC HIỆN**

**CHƯƠNG 4. MÔ HÌNH HÓA CHƯƠNG TRÌNH**

1. Cơ chế quá trình hoạt động

2. Mô tả chi tiết các thành phần cần xử lí

**CHƯƠNG 5. QUY TRÌNH THỰC HIỆN CHƯƠNG TRÌNH**

1. Thư viện cần dùng
2. Cài đặt một số biến cơ bản
3. Định dạng hình ảnh và vị trí
4. Cơ chế điều khiển xe tăng
5. Tăng tốc độ của trò chơi
6. Xử lí khi xe tăng nhặt vật phẩm
7. Xử lí khi xe tăng va trúng chướng ngại vật
8. Thêm hiệu ứng âm thanh

**PHẦN 0. TỔNG QUAN GIAO DIỆN**

1. **Các thành phần trên giao diện**

Giao diện của chúng ta được thiết kế bằng PyQT5 gồm một cửa sổ có hình ảnh giới thiệu chương trình, trong đó phải có 3 nút ấn mở 3 chế độ và một nút thoát để thoát chương trình, ngoài ra còn có caption và icon chương trình.

1. **Phác thảo giao diện**

Graphical user interface, application

Description automatically generated

Mockup UI cửa sổ chính

1. **Tạo đối tượng button và xử lí sự kiện**

|  |
| --- |
| # Tạo 3 nút chế độ          btn1 = QPushButton('Tiêu diệt', self)          btn1.move(80, 350)          btn1.clicked.connect(self.on\_click\_mode1) |

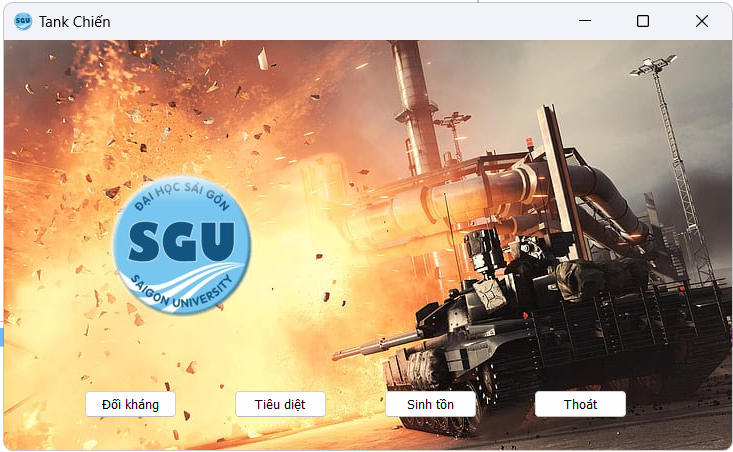
Ta tạo nút nhấn có tên “Tiêu diệt” tương ứng với chế độ cùng tên và set vị trí chúng cho phù hợp trên trục tọa độ Oxy (80,350), và tạo hàm xử lí để xử lí sự kiện này, các chế độ khác tương tự

|  |
| --- |
| @pyqtSlot()      def on\_click\_mode1(self):          subprocess.run(['python', 'mode1.py']) |

Trong hàm xử lí sự kiện này, ta sẽ thực thi bằng cách chạy file có định dạng ‘python’ tại một đường dẫn cho trước.

|  |
| --- |
| @pyqtSlot()      def on\_click\_exit(self):          # Thoát chương trình          QApplication.quit() |

Khi ta nhấn nút thoát, chương trình sẽ gọi đến hàm on\_click\_exit() sẽ gọi đến phương thức thoát chương trình.



Giao diện chương trình sau khi hoàn chỉnh

**PHẦN 1: CHẾ ĐỘ TIÊU DIỆT**

**CHƯƠNG 1. GIỚI THIỆU**

Nhắc đến trò chơi điện tử (hay game), chắc chắn không ai xa lạ với thuật ngữ này, đây là thứ gắn liền với tuổi thơ của biết bao thế hệ, gắn bó với mọi người, chắc chắn ai cũng từng chơi qua một lần.

Dạng sơ khai nhất của game chính là các trò chơi như cờ caro được chơi trên giấy, cờ vua, cờ tướng,... Hiện nay trò chơi đã được nâng lên một tầm cao mới, từ khi công nghệ cũng như internet phát triển, các lập trình viên đã mô hình hóa và đưa thành công nó lên các nền tảng thiết bị điện tử. Ngày nay trò chơi điện tử đã không ngừng được cải thiện lối chơi, đồ họa, các thể loại ngày càng hấp dẫn, thu hút đông đảo giới trẻ yêu thích.

Khi nói đến game thì nhiều người thường nghĩ rằng nó là một thứ vô bổ, lãng phí thời gian, ảnh hưởng đến sức khỏe và học tập. Và dĩ nhiên ở chiều ngược lại, các nhà khoa học đã chứng minh rằng, nếu chơi game điều độ, có chừng mực thì sẽ giúp chúng ta xả stress sau những giờ học tập và làm việc căng thẳng, và đối với một số tựa game như Minecraft, Lego còn khiến chúng ta rèn luyện khả năng tư duy sáng tạo cho trẻ.

Đến với nhóm em, em xin giới thiệu với quý thầy cô và các bạn về một trò chơi 2D đơn giản, đó chính là trò chơi bắn xe tăng, với lối chơi truyền thống đơn giản đi với đó là các giai điệu âm nhạc bắt tai phù hợp với mọi lứa tuổi.

Đầu tiên chính là chế độ tiêu diệt, chế độ này phù hợp với người chơi đơn, một mình, muốn một mình xông pha càn phá cả quân địch.

**A picture containing chart

Description automatically generated**

Tổng quan chương trình

**CHƯƠNG 2. HƯỚNG DẪN CÁCH CHƠI**

**A close up of a keyboard

Description automatically generated with low confidence**

**Bước 1:** Ta dùng các phím chức năng WSAD để di chuyển lên, xuống, trái, phải.

A hand on a keyboard

Description automatically generated with medium confidence

**Bước 2:** Ta dùng phím Space để bắn đạn

A picture containing graphical user interface

Description automatically generated

**Bước 3:** Dùng đạn để phá hủy tường để tiến lên đối

đầu với kẻ địch.

**Bước 4:** Cố gắng né tránh đạn từ kẻ địch, và đảm bảo khoảng cách và thời điểm phù hợp để hạ tất cả kẻ địch bằng đạn của mình.

**CHƯƠNG 3. Ý TƯỞNG THỰC HIỆN**

**Bước 1:** Ta tạo ra một vùng trống bị giới hạn bởi rìa bản đồ, sau đó thiết lập hai đối thủ (ta và địch), mỗi đối tượng sẽ đặt ở 2 góc bản đồ.

Diagram

Description automatically generated

**Bước 2:** Sau đó 2 chiếc xe tăng sẽ đối đầu lẫn nhau bằng cách bắn đạn nhằm hạ gục đối thủ, đồng thời cố gắng di chuyển để né đạn đối phương.

Diagram

Description automatically generated

**Bước 3:** Để lối chơi thêm sinh động và hấp dẫn ta thêm chướng ngại vật vào, đó là các bức tường chắn đạn để ngăn bắn đạn trực tiếp vào nhau, tạo ra các tình huống chân thật hơn.

Diagram

Description automatically generated

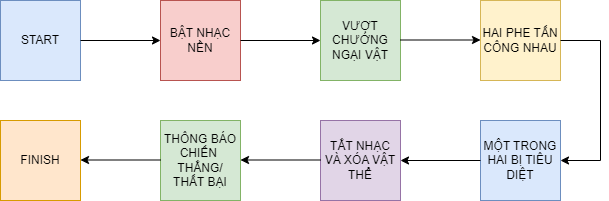
**Bước 4:** Để tăng độ khó cho game, ta sẽ tăng số lượng của địch lên 5 đối tượng.

Diagram

Description automatically generated

**CHƯƠNG 4. MÔ HÌNH HÓA CHƯƠNG TRÌNH**

1. **Cơ chế quá trình hoạt động**

****

1. **Mô tả chi tiết các thành phần cần xử lí**

|  |  |
| --- | --- |
| **Căn chỉnh cửa sổ, kích thước** |  |
| **Định dạng xe tăng của người chơi** | + Cơ chế các phím điều khiển.  + Hướng di chuyển của vật thể.  + Vị trí ban đầu của các vật thể trên map. |
| **Thiết lập cơ chế đạn của người chơi** | + Cơ chế phím bắn của đạn    + Hướng bay của đạn.    + Vị trí theo xe tăng khi bắn đạn.  + Loại bỏ đạn khi trúng tường, địch, ra khỏi map. |
| **Định dạng xe tăng địch** | + Cơ chế tự điều khiển bằng thuật toán (máy tính xử lí).  + Cơ chế tự xoay hướng bằng thuật toán (máy tính xử lí).  + Vị trí ban đầu trên map, khoảng cách giữa kẻ địch với nhau. |
| **Thiết lập cơ chế bắn đạn của địch** | + Cơ chế tự động bắn ngẫu nhiên bằng thuật toán (máy tính xử lí).  + Vị trí đạn bay ra theo từng kẻ địch cụ thể.  + Loại bỏ đạn khi trúng tường, người chơi, ra khỏi map. |
| **Xuất thông báo ra màn hình khi chiến thắng hoặc thua** | + Tô đen màn hình và hiện ra dòng chữ xanh hoặc đỏ ở vị trí được thiết lặp. |
| **Thêm vào các hiệu ứng như âm thanh, giao diện (nếu có)** | + Bật nhạc nền khi vào game và tắt khi kết thúc game.    + Phát ra âm thanh khi ta hoặc địch bắn.    + Phát ra âm thanh khi kẻ địch hoặc người chơi bị trúng đạn. |

**CHƯƠNG 5. QUY TRÌNH THỰC HIỆN CHƯƠNG TRÌNH**

1. **Thư viện cần dùng**

+ Pgzrun (Pygame Zero là một phần con của Pygame) bộ mô-đun [Python](https://vi.wikipedia.org/wiki/Python_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) [đa nền tảng](https://vi.wikipedia.org/wiki/%C4%90a_n%E1%BB%81n_t%E1%BA%A3ng) được thiết kế để viết [trò chơi điện tử](https://vi.wikipedia.org/wiki/Tr%C3%B2_ch%C6%A1i_%C4%91i%E1%BB%87n_t%E1%BB%AD). Nó bao gồm đồ họa máy tính và thư viện âm thanh được thiết kế để sử dụng với ngôn ngữ lập trình [Python](https://vi.wikipedia.org/wiki/Python_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)).

1. **Căn chỉnh cửa sổ**

|  |
| --- |
| WIDTH, HEIGHT = 800, 600 |

Ở đây ta chọn kích thước của cửa sổ chương trình, trong đó chiều dài và chiều rộng lần lượt là 800 x 600, đây là kích thước tiêu chuẩn cho game 2D, ngoài ra còn phù hợp với kích thước của hình nền.

**🡪** Tại đây ta được cửa sổ màu đen (chưa có gì bên trong)

1. **Đưa vật thể lên màn hình**

Graphical user interface

Description automatically generated with medium confidence

|  |
| --- |
| tank = Actor("tankplayer")  background = Actor("chientruong")  def draw():          background.draw()          tank.draw() |

Ta dùng lệnh Actor để thêm hình ảnh vào biến để thao tác, trong đó biến tank nhận đường dẫn đến hình ảnh đó, tương tự như vậy với nền.

Sau cùng ta đưa nó lên màn hình bằng lệnh draw.

Ở đây ta đã đưa thành công lên màn hình, mặc định nếu chưa định dạng vị trí thì xe tăng ở vị trí thứ (0,0).

Diagram

Description automatically generated with low confidence

Và theo đó vị trí mặc định thứ (0,0) được xác định bằng hình vẽ sau

🡪 Ta cần định dạng vị trí của chúng để có thể bắt đầu với vị trí phù hợp.

1. **Định dạng xe tăng**

|  |
| --- |
| SIZE\_TANK = 25 # Kích thước xe tăng bằng 1 nửa ô tường |

Cho kích thước xe tăng là 25, bởi vì nó phù hợp với kích thước của ô gạch (có thể nhỏ hơn).

|  |
| --- |
| tank.pos = (WIDTH/2, HEIGHT - SIZE\_TANK)  tank.angle = 90 |

**Tô đỏ:** vị trí của xe tăng phe mình được xác định tại vị trí như hình sau, trong đó ta chỉ sử dụng 4 góc 90, 180, 270, 360.

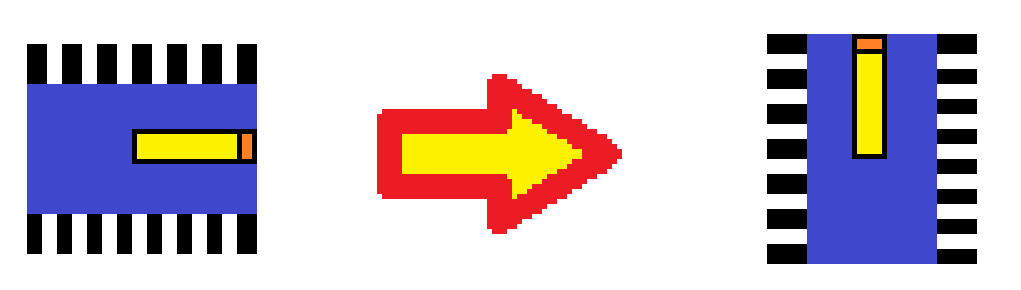
Chart, diagram

Description automatically generated with medium confidence

Diagram

Description automatically generated

**Tô xanh:** ta xoay xe tăng theo góc 90 độ để xe hướng về phía địch.



**5. Tạo chướng ngại vật (bức tường gạch)**

Text

Description automatically generated

**Tô đỏ:** ta duyệt toàn bộ bản đồ, chiều ngang 16 ô, chiều dọc 10 ô 🡪 16 x 10 = 160 ô tường (vừa đủ chỗ cho bản đồ 800 x 600)

**Tô xanh:** sau đó ta muốn chỗ có ô gạch để làm chướng ngại vật, chỗ không có ô gạch để cho đa dạng thì dùng thư viện random, khi đó máy tính sẽ tự sắp xếp vị trí các ô tường một cách ngẫu nhiên, do đó không màn nào giống màn nào.

🡪 Ta sẽ chọn chọn ngẫu nhiên khoảng 50 ô gạch làm tường.

**Tô đỏ:** sau khi lấy file hình ảnh add cho ô tường, thì ta định dạng chúng, ở đây ta lấy kích thước chuẩn của ô gạch là 50 (bằng 1 nửa so với xe tăng)

🡪 Cuối cùng ta dùng hàm append để thêm giá trị trên vào cuối LIST.

|  |
| --- |
| bucTuong = [] |

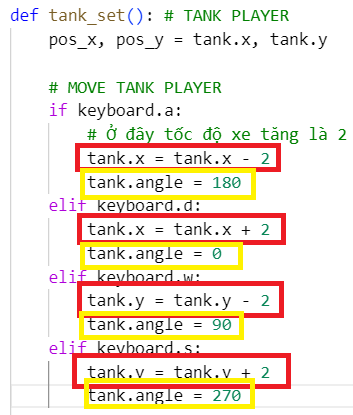
|  |
| --- |
| def draw():  for tuong in bucTuong:              tuong.draw() |

Cuối cùng là gọi và đưa nó lên màn hình.

**Lưu ý: SIZE\_TANK** được dùng trong trường hợp này fix lỗi lệch góc 🡪 thẩm mỹ.

**6. Điều khiển xe tank của người chơi**

Ta dùng các nút: A D W S tương ứng với lên, xuống, trái, phải (tốc độ = 2, có thể thay đổi cho phù hợp.

****

**Tô đỏ:** và hướng di chuyển được mô tả trên bản đồ theo hệ trục tọa độ Oxy như trên.

**Tô vàng**: và hướng xoay được thực hiện như mô tả bên hình, với ảnh gốc được đặt bên góc phải.

**Mũi tên xanh**: nếu chỉ dùng if thôi thì xe tăng xe đi chéo 🡪 không phù hợp.

Chart

Description automatically generated

Diagram, schematic

Description automatically generated

**7. Tạo giới hạn di chuyển (khi va vào tường và rìa map)**

**Va vào tường:**

|  |
| --- |
| def tank\_set(): # TANK PLAYER      pos\_x, pos\_y = tank.x, tank.y |

Ta tạo 2 biến lưu lại giá trị ban đầu (trước 1 thao tác) của vị trí.

|  |
| --- |
| # DK để tank không đi qua được tường      if tank.collidelist(bucTuong) != -1:          tank.x = pos\_x          tank.y = pos\_y |

Và khi xe tank xảy ra va chạm với tường (collidelist) thì xe tăng sẽ nhận lại vị trí ban đầu

🡪 Tóm lại xe tank sẽ không đi qua tường được (nếu cứ cố tiến thì sẽ sẽ bị rằng buộc điều kiện mãi)

**Va vào rìa map:**

Để xét được điều kiện này, ta phải phụ thuộc vào chiều dài, chiều cao của bản đồ mà ta đã set cho chương trình.

|  |
| --- |
| # DK để tank không đi ra khỏi map      if tank.x < SIZE\_TANK or tank.x > WIDTH - SIZE\_TANK or tank.y > HEIGHT - SIZE\_TANK:          tank.x = pos\_x          tank.y = pos\_y |

Cụ thể như hình dưới đây:

Graphical user interface, application

Description automatically generated

**Lưu ý:** ở đây ta thêm SIZE\_TANK để căn góc (do tính toán lỗi nền với kích thước cửa sổ).

**8. Thao tác với đạn của xe tăng của người chơi**

Để định dạng được đạn, ta cần 2 yếu tố chính:

**+ Thứ nhất,** vị trí của xe tăng, đạn sẽ xuất hiện tại đó.

**+ Thứ hai,** hướng đi của đạn phụ thuộc vào hướng nòng súng.

Ta lấy vị trí và phương hướng của đạn phụ thuộc vào vị trí và phương hướng của xe tăng. (pos và angle).

|  |
| --- |
| if keyboard.space:              # Định dạng đạn của player              dan = Actor("dan")              dan.angle = tank.angle  dan.pos = tank.pos  danPlayer.append(dan) |

|  |
| --- |
| for dan in danDich:          if dan.angle == 0:              dan.x += 5          if dan.angle == 180:              dan.x -= 5          if dan.angle == 90:              dan.y -= 5          if dan.angle == 270:              dan.y += 5 |

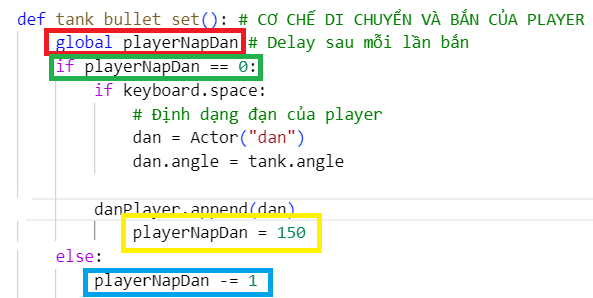
Cuối cùng là đạn sẽ được bắn ra, xe tăng hướng (angle) nào thì đạn bắn ra từ hướng đó, cứ tăng dần lên (tương tự cơ chế di chuyển của xe tăng được mô tả phía trên) với tốc độ đường đạn đi được lấy bằng 5.

**Tuy nhiên xe tăng không thể bắn liên thanh được, do đó ta phải thiết lập quá trình nạp đạn để khựng lại 1 khoảng thời gian mới bắn được tiếp.**

Đầu tiên ta tạo hai biến để đếm thời gian “chết” của đạn bên phe địch và người chơi.

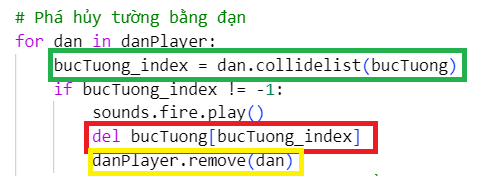
|  |
| --- |
| playerNapDan = 0  dichNapDan = 0 |

Ta chọn nó làm biến toàn cục, sau đó xét điều kiện, nếu thời gian chết bằng 0 thì được phép bắn tiếp, sau đó lại nó sẽ lại giảm từ 20 🡪 0, và được bắn tiếp, cứ tiếp diễn như vậy đến khi trò chơi kết thúc.



**Phá hủy tường**

Tương tự như trên , ta kiểm tra sự va chạm (collidelist) giữa đạn và tường



Nếu thỏa điều kiện va chạm ta sẽ xóa bức tường đó bằng cách remove viên đạn đó để tránh nó đi xuyên qua những bức tường phía sau.

**Xóa đạn thừa khi ra khỏi map**

Nếu không có bước này, chương trình vẫn chạy bình thường tuy nhiên càng bắn nhiều đạn thì chương trình càng tính toán nhiều hơn 🡪 Tiêu tốn nhiều tài nguyên máy tính, do đó ta sẽ xóa lượng đạn thừa không cần thiết.

|  |
| --- |
| # XÓA ĐẠN KHI CHẠM RÌA MAP --> GIẢM DUNG LƯỢNG RAM              if dan.x < 0 or dan.x > WIDTH or dan.y < 0 or dan.y > HEIGHT:                  danPlayer.remove(dan) |

Diagram

Description automatically generated

Vị trí của đạn khi bắn ra sẽ phụ thuộc vào vị trí của xe tank đang ở đâu (SIZE\_TANK: fix góc thừa)

|  |
| --- |
| if dan.angle == 0:                  dan.pos = (tank.x + SIZE\_TANK, tank.y)              if dan.angle == 90:                  dan.pos = (tank.x, tank.y + SIZE\_TANK)              if dan.angle == 180:                  dan.pos = (tank.x - SIZE\_TANK, tank.y)              if dan.angle == 270:                  dan.pos = (tank.x, tank.y + SIZE\_TANK) |

**9. Định dạng xe tăng địch**

Tương tự như xe tăng của người chơi, xe tăng của địch được định dạng như sau

Và cuối cùng xoay (rotate) sao cho xe tăng địch hướng xuống dưới (270 độ).

Diagram

Description automatically generated

**Text, email

Description automatically generated**

**10. Xe tăng địch tự động tính toán và di chuyển**

Đầu tiên ta tạo 2 biến toàn cục, mục đích để ghi lại thời gian chết khi xe tăng di chuyển và bắn đạn, tương tự như người chơi, sau đó nhận 2 biến lưu vị trí để xử lí di chuyển.

|  |
| --- |
| global dichDiChuyen, dichNapDan          pre\_x, pre\_y = dich.x, dich.y |

Sau đó ta dùng hàm random để nhận ngẫu nhiên một trong ba số: 0, 1, 2 (3 giá trị) 🡪 nhằm mục đích để máy tính tự do xử lí từng công việc cụ thể một cách không đoán trước.

|  |
| --- |
| choice = random.randint(0,2) |

**Khi xuất hiện ngẫu nhiên số 0:**

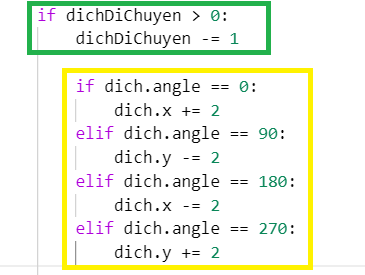
**+** Xe tăng địch sẽ duy chuyển

|  |
| --- |
| elif choice == 0: # Xe tăng địch di chuyển              dichDiChuyen = 30 |

**Tô xanh**: khi đó biến đếm duy chuyển từ 30 🡪 0:

Trong quá trình đếm đó xe tăng sẽ duy chuyển một cách ngẫu nhiên, do máy điều khiển.

**Tô vàng:** và phương thức di chuyển tương tự như xe tăng của người chơi, trường hợp góc 0, 90, 180, 270 thì theo thứ tự trên, trái, dưới, phải.



**Khi xuất hiện ngẫu nhiên số 1:**

A picture containing text

Description automatically generated

Thì nó cũng sẽ random ở 3 vị trí

Diagram

Description automatically generated

Nhân cho góc 90 độ xoay ngược chiều kim đồng hồ 🡪 Xe tăng địch đổi hướng.

Nếu đang ở vị trí này thì tank sẽ đổi sang vị trí kia, cụ thể:

A 🡪 D

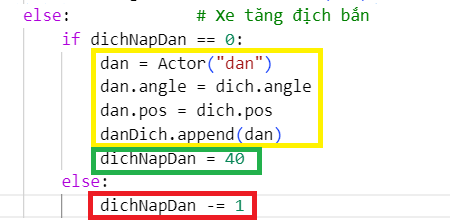
D 🡪 C

C 🡪 B

B 🡪 A

**Trường hợp ngẫu nhiên còn lại là 2:**

+ Xe tăng địch sẽ bắn đạn



**Tô vàng:** định dạng tương tự như xe tăng cửa người chơi.

Sau đó đạn bắn ra sẽ rơi vào thơi gian chết tức là đang nạp đạn, sau đó sẽ giảm thời gian từ 40 🡪 0: bắn đạn ngẫu nhiên để bắn tiếp.

🡪 Đạn bắn ra sẽ được đưa vào list

|  |
| --- |
| danDich = [] |

Cuối cùng là đưa đạn lên bản đồ.

|  |
| --- |
| for dich in keDich:              dich.draw() |

**11. Xe tăng địch tự động tính toán bắn đạn**

Thao tác bắn đạn của xe tăng địch về cơ bản giống so với người chơi, chỉ khác người chơi dùng phím để điều khiển, còn bot thì dùng hàm random để xử lí, do đó ta sẽ không nhắc lại cách thức bắn.

|  |
| --- |
| def enemy\_bullet\_set(): # CƠ CHẾ KHI BẮN CỦA KẺ ĐỊCH      global keDich, ketThuc      for dan in danDich:          if dan.angle == 0:              dan.x += 5          if dan.angle == 180:              dan.x -= 5          if dan.angle == 90:              dan.y -= 5          if dan.angle == 270:              dan.y += 5 |

Nếu đạn của địch va chạm với tường thì tiến hành xóa vị trí bức tường đó và dùng hàm remove để loại bỏ viên đạn đó tránh nó đi xuyên qua tất cả các vật thể.

Text

Description automatically generated with low confidence

Và đến cuối cùng tương tự như xe tăng của người chơi, đều xóa đạn khi đi ra khỏi map để tránh tiêu hao tài nguyên máy tính.

|  |
| --- |
| # Tương tự như người chơi, ta sẽ xóa đạn của kẻ địch khi chúng đi ra khỏi map              if dan.x < 0 or dan.x > WIDTH or dan.y > HEIGHT:                  danDich.remove(dan) |

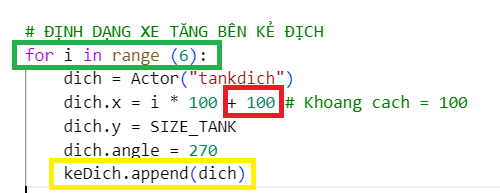
**12. Tăng số lượng xe tăng quân địch từ 1 lên 6**

Nếu chỉ có 1 chiếc xe tăng bên phe địch thì trò chơi sẽ rất vô vị, vì vậy để tăng thêm độ khó, tăng thêm sự sôi động ta sẽ tăng số lượng xe tăng địch lên.

**+ Tô xanh:** cách dễ nhất chính là dùng vòng lặp 1 🡪 6 chiếc tăng.

**+ Tô đỏ:** khoảng cách giữa các xe tăng địch phù hợp

**+ Tô vàng:** tạo List để chứa nhiều xe tăng hơn.

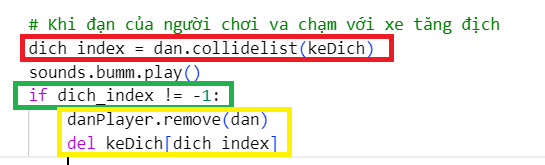


Cuối cùng là đưa đạn của kẻ địch lên cửa sổ trò chơi.

|  |
| --- |
| for dan in danDich:          dan.draw() |

**13. Trò chơi kết thúc**

Khi đạn của người chơi bắn toàn bộ xe tank phe địch, nếu đạn của người chơi bắn ra trúng kẻ địch thì xóa đạn + xóa tank đó.



Trường hợp ngược lại nếu đạn xe tăng địch bất kì bắn trúng xe tăng ta 🡪 giá trị game over mặc định là False sẽ chuyển thành True, là đúng.

|  |
| --- |
| if dan.colliderect(tank):                  ketThuc = True                  keDich = [] |

Khi kết thúc chỉ cần xuất ra thông báo thắng hoặc thua tùy vào trường hợp kẻ địch còn hay người chơi còn sống.

Sẽ thiết lặp cửa sổ là màu đen kèm dòng chữ thắng hoặc thua ở vị trí và màu sắc cho trước.

|  |
| --- |
| if ketThuc:          screen.fill(BLACK)          screen.draw.text('MISSION FAILED',(130,250), color = RED, fontsize = 100)      elif len(keDich) == 0:          screen.fill(BLACK)          screen.draw.text('MISSION SUCCESS',(130,250), color = GREEN, fontsize = 100) |

**14. Thêm tiếng nhạc, bắn đạn, nổ**

Để thêm hiệu ứng âm thanh sinh động và chân thật ta có thể dùng các file âm thanh được chuẩn bị từ trước, sau đó gọi lại mỗi khi cần sử dụng, ví dụ lúc bắn đạn, lúc vỡ tường, nhạc nền, ...

|  |  |
| --- | --- |
| Nhạc nền | sounds.astro.play() |
| Nổ xe tăng địch | sounds.bumm.play() |
| Xe tăng bắn đạn | sounds.guns.play() |
| Phá tường | sounds.fire.play() |
| Tắt nhạc nền | sounds.astro.stop() |
| Âm lượng nhạc nền | sounds.astro.set\_volume(0.5) |

**PHẦN 2. CHẾ ĐỘ ĐỐI KHÁNG**

**CHƯƠNG 1. GIỚI THIỆU**

Trò chơi vui nhất và mang lại nhiều kỉ niệm nhất là khi chơi nó với bạn bè, người thân, bởi lẽ chúng mang lại sự gắn bó, thân thiết, tạo ra nhiều kỉ niệm đẹp và hơn hết cùng hỗ trợ và đối đầu nhau giúp tăng thêm sự nhạy bén, trau dồi kĩ năng hơn .

**CHƯƠNG 2. HƯỚNG DẪN CÁCH CHƠI**

**Bước 1:** Người chơi thứ nhất, điều khiển chiếc xe tăng xanh lá sẽ dùng các phím W, S, A, D tương ứng với thao tác lên, xuống, trái, phải. Người chơi thứ hai, điều khiển chiếc xe tăng đỏ, sẽ dùng các phím mũi tên điều hướng lên, xuống, trái phải tương tự.

A picture containing electronics, jack, keyboard

Description automatically generated

**Bước 2:** Người chơi xe tăng xanh sẽ dùng phím SHIFT bên trái để bắn đạn, ngược lại người chơi xe tăng đỏ sẽ dùng phím SHIFT bên phải để bắn đạn.



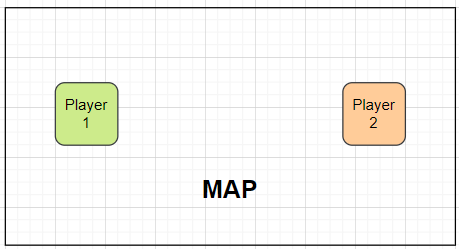
**Bước 3:** Hai xe tăng sẽ cố gắng né tránh đạn từ kẻ địch đồng thời đảm bảo khoảng cách và thời điểm phù hợp để hạ kẻ địch đối diện bằng đạn của mình.

****

**Lưu ý:** Chính giữa bản đồ sẽ có một ranh giới là bờ sông nên xe tăng sẽ không thể vượt qua biên giới của đối thủ, không có chướng ngoại vật sẽ khiến tăng độ đó cho trò chơi.

**CHƯƠNG 3. Ý TƯỞNG THỰC HIỆN**

**Bước 1:** Đưa xe tăng của hai người chơi lên bản đồ, xe tăng xanh lá sẽ ở bên trái, xe tăng màu đỏ sẽ nằm ở vị trí bên phải.

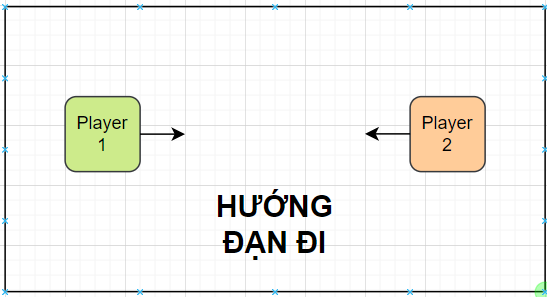


**Bước 2:** Hai xe tăng sẽ di chuyển theo 4 hướng, di chuyển lên xuống để né đạn của đối phương, di chuyển sang trái phải để tăng độ chính xác khi bắn, hoặc dụ kẻ địch vào tầm bắn.

Diagram, schematic

Description automatically generated

**Bước 3:** Để tránh đường đạn đi rối mắt và giúp đơn giản hóa trò chơi và cách thức xử lí đường đạn, thì xe tăng xanh sẽ chỉ bắn về hướng phải, xe tăng đỏ sẽ bắn về phía bên trái.



**Bước 4:** Để thỏa điều kiện chỉ bắn phải và trái thì ta sẽ giới hạn đường đi của 2 xe tăng trong phạm vi biên giới. Như xe tăng xanh chỉ được di chuyển trong ô vuông bên trái, còn xe tăng đỏ sẽ được di chuyển bên phía ô vuông trái.

A picture containing diagram

Description automatically generated

**CHƯƠNG 4. MÔ HÌNH HÓA CHƯƠNG TRÌNH**

**1. Cơ chế quá trình hoạt động**

**Diagram, schematic

Description automatically generated**

**2. Mô tả chi tiết**

Khái quát tất cả các đối tượng, các thao tác xử lí và các sự kiện có thể xảy ra trong quá trình chơi giữa hai người chơi.

|  |  |
| --- | --- |
| **Xe tăng xanh (player 1)** | + Cơ chế di chuyển (tốc độ, hướng đi, xoay hướng)  + Cơ chế bắn đạn  + Xử lí khi viên đạn bắn trúng kẻ địch  + Xử lí khi nạp đạn (delay)  + Xử lí khi đạn đã đi ra khỏi map |
| **Xe tăng đỏ (player 2)** | + Cơ chế di chuyển  + Cơ chế bắn đạn  + Xử lí khi viên đạn bắn trúng kẻ địch  + Xử lí khi nạp đạn  + Xử lí khi đạn đã đa ra khỏi map |
| **Hình ảnh, biên giới, âm thanh** | + Thiết lập vị trí ranh giới hạn có thể di chuyển  + Lấy hình ảnh và đưa lên cửa sổ trò chơi  + Lấy đường dẫn âm thanh và gọi lại khi xảy ra một sự kiện phù hợp.  + Một số thiết lắp FPS,… |
| **Thông báo** | + Xuất ra thông báo khi kết thúc chương trình và reset lại chương trình |

**CHƯƠNG 5. QUY TRÌNH THỰC HIỆN CHƯƠNG TRÌNH**

1. **Thư viện cần dùng**

|  |
| --- |
| import pygame  import os  pygame.init() |

Ta viện pygame sẽ hỗ trợ tốt hơn cho việc làm game 2D, trong đó nó hỗ trợ đưa các hình ảnh vào cửa sổ, truy xuất các đối tượng, điều chỉnh và xử lí các sự kiện, những trường hợp có thể xảy ra trong quá trình chơi. Trong đó cần có thư viện OS để có thể thao tác dễ dàng hơn trên cửa sổ. Sau cùng gọi ra init của pygame để xử lí âm thanh và font chữ.

1. **Cài đặt các biến**

|  |
| --- |
| WIDTH, HEIGHT = 900, 500  pygame.display.set\_caption("Tank Battle")  WHITE, BLACK, RED, GREEN, BLUE = (255, 255, 255), (0, 0, 0), (255, 0, 0), (34, 177, 76), (0, 162, 232)  WINDOWS = pygame.display.set\_mode((WIDTH, HEIGHT))  BULLET\_HIT\_SOUND = pygame.mixer.Sound('sounds/bumm.mp3')  BULLET\_FIRE\_SOUND = pygame.mixer.Sound('sounds/gun.mp3')  SOUND\_TRACK = pygame.mixer.Sound('sounds/astro.mp3')  MANG\_FONT = pygame.font.SysFont('comicsans', 40)  WINNER\_FONT = pygame.font.SysFont('comicsans', 100) |

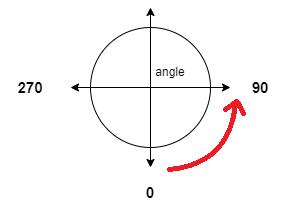
Ta thiết lập kích cỡ hình ảnh là 900 x 500, có thể căn chỉnh cửa sổ lớn hơn.

Gán các mã màu, đường dẫn âm thanh, kích thước màn hình và các biến để có thể dễ dàng gọi và xử lí ở dưới, font chữ cùng với kích thước chữ để hiện mạng và thông báo bằng text.

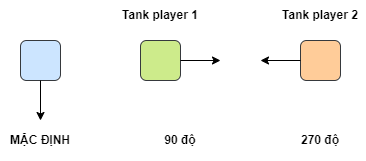
1. **Định dạng và đưa hình ảnh lên cửa sổ**

|  |
| --- |
| # Gán hình ảnh cho các biến để dễ dàng thao tác  GREEN\_TANK\_IMAGE = pygame.image.load(os.path.join('images', 'tankplayer.png'))  GREEN\_TANK = pygame.transform.rotate(pygame.transform.scale(GREEN\_TANK\_IMAGE, (TANK\_WIDTH, TANK\_HEIGHT)), 90)  RED\_TANK\_IMAGE = pygame.image.load(os.path.join('images', 'tankdich.png'))  RED\_TANK = pygame.transform.rotate(pygame.transform.scale(RED\_TANK\_IMAGE, (TANK\_WIDTH, TANK\_HEIGHT)), 270)  BGWAR = pygame.transform.scale(pygame.image.load(os.path.join('images', 'chientruong.png')), (WIDTH, HEIGHT)) |

Tương tự như khi cài đặt các biến, ta lưu đường dẫn của các hình ảnh, model xe tăng và nền (background) vào biến, tuy nhiên ta phải định dạng chúng lại cho phù hợp.

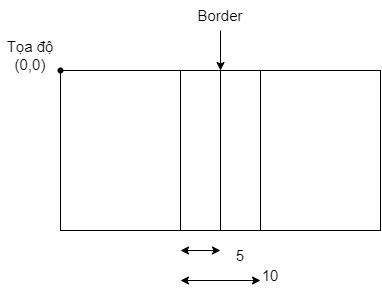


Trong đó file hình ảnh ban đầu sẽ là 0 độ, ta dùng rotate để xoay chúng, player 1 sẽ nhận giá trị 90 độ và hướng phải, player 2 sẽ nhận giá trị 270 độ và hướng trái.



Cụ thể như hình sau đây.

1. **Tùy chỉnh bố cục của của sổ.**

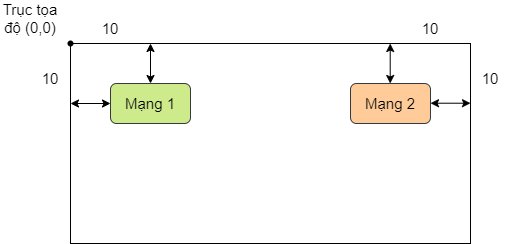
****

Ta lấy chiều rộng của của bản đồ WIDTH chia 2 để lấy điểm giữa (border) bản đồ nhằm chia 2 bên ranh giới.

Để tạo khoảng cách từ khoảng giữa là 5, thì ta trừ đi 5 để lấy điểm bên trái, sau đó cộng cho 10 để lấy điểm phải để hai bên phân chia được đều nhau.

|  |
| --- |
| BORDER = pygame.Rect(WIDTH//2 - 5, 0, 10, HEIGHT)  # Ranh giới phân chia hai chiến tuyến      pygame.draw.rect(WINDOWS, BLUE, BORDER) |

Sau đó đưa đường biên giới lên bản đồ, ở đây ta thiết lập chúng màu xanh (blue) để chúng đồng màu với bờ sông.



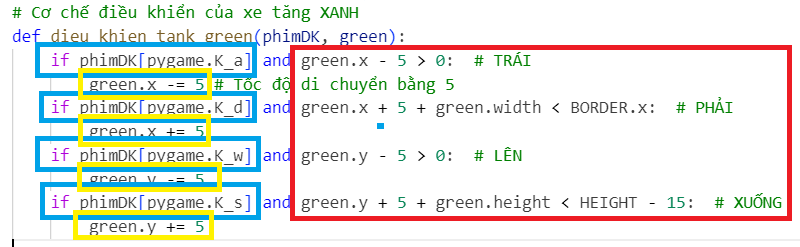
|  |
| --- |
| red\_mang\_text = MANG\_FONT.render("Mang Song: " + str(red\_mang), 1, RED)      green\_mang\_text = MANG\_FONT.render("Mang Song: " + str(green\_mang), 1, GREEN)      WINDOWS.blit(red\_mang\_text, (WIDTH - red\_mang\_text.get\_width() - 10, 10))      WINDOWS.blit(green\_mang\_text, (10, 10)) |

Ta đưa chuỗi số mạng còn lại ra cửa sổ để người dùng dễ quan sát và hình dung, chuỗi “mạng sống” của xe tăng xanh sẽ nhận vị trí cách trục tọa độ là (10,10), còn xe tăng đỏ sẽ lấy trục tọa độ width max trừ đi 10 để nằm bên phải.

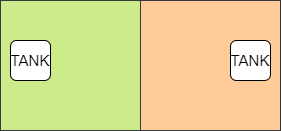
Tham số thứ hai của hàm render là 1sẽ làm chữ xuất ra có hiệu ứng nhẹ (antialiasing) tạo ra một đường viền mịn hơn và giúp chữ trông rõ ràng hơn. Nếu tham số này là 0, thì chữ sẽ được vẽ mà không có hiệu ứng antialiasing, tạo ra một đường viền rõ ràng hơn nhưng có thể không trông mịn màng bằng.

1. **Cơ chế điều khiển của của hai xe tăng**

**a. Điều khiển xe tăng xanh**

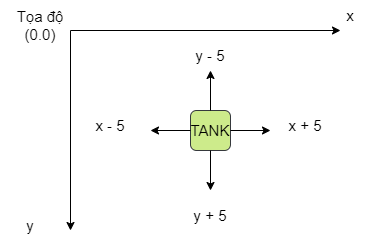


**Tô xanh:** Để thuận tiện cho việc điều khiển ta thiết lập cho người dùng bên trái điều khiển xe tăng xanh bằng các phím W, S, A, D tương ứng với lên, xuống, trái, phải.



**Tô đỏ:** điều kiện để xe tăng bên xanh không được phép vượt qua ranh giới ở giữa, vị trí trên và bên trái (A, W) trùng với góc tọa độ nên ta không cần phải thiết lập ranh giới cho chúng, đối với vị trí dưới (S) ta dựa vào chiều cao HEIGHT, vị trí phải (D) dựa vào điểm chia giữa (Border).

**Tô vàng:** nếu thoải điều kiện là đang điều khiển bằng nút nhấn và thỏa điều kiện trong ranh giới được phép điều khiển thì ta sẽ dịch chuyển



Dựa trên góc trục tọa độ Oxy, nếu muốn tiến phải thì + 5, lùi trái thì – 5, tiến xuống thì + 5, lùi lên thì – 5. Trong đó ta có thể tăng tốc độ di chuyển bằng cách tăng 5 lên.

**b. Điều khiển xe tăng đỏ**

|  |
| --- |
| # Cơ chế điều khiển của xe tăng ĐỎ  def dieu\_khien\_tank\_red(phimDK, red):      if phimDK[pygame.K\_LEFT] and red.x - 5 > BORDER.x + BORDER.width:  # TRÁI          red.x -= 5      if phimDK[pygame.K\_RIGHT] and red.x + 5 + red.width < WIDTH:  # PHẢI          red.x += 5      if phimDK[pygame.K\_UP] and red.y - 5 > 0:  # LÊN          red.y -= 5      if phimDK[pygame.K\_DOWN] and red.y + 5 + red.height < HEIGHT - 15:  # XUỐNG          red.y += 5 |

Ngược lại với xe tăng xanh, xe tăng đỏ ta sẽ dùng các phím mũi tên điều hướng bên phải của bàn phím để điều khiển như so với xe tăng xanh.

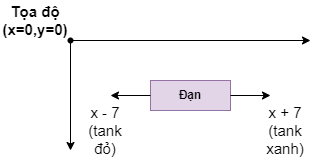
Tương tự như cơ chế điều khiển tiến lùi của xe tăng, bằng cách thao tác xy trên trục tọa độ Oxy.

Điểm khác biệt so với xe tăng xanh thì xe tăng đỏ đường ranh giới có thể di chuyển sẽ khác biệt. Vị trí trên (Up) trùng với trục tọa độ 0 nên ta sẽ không quy ước, vị trí xuống (Down) sẽ quy ước dựa vào độ dài của chiều cao, vị trí trái (Left) quy ước dựa vào điểm chia giữa, vị trí phải (Right) quy ước dựa vào chiều rộng.

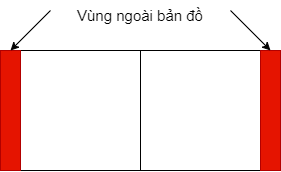
1. **Cơ chế xử lí đạn của cả hai xe tăng**



**Tô xanh:** Đạn của xe tăng xanh và xe tăng đỏ sẽ được khai báo riêng biệt, ta tạo một vòng lặp để duyệt và xét từng viên trong một đợt bắn đó, số đạn trong một đợt bắn ta sẽ khai báo sau.



**Tô vàng**: ta xác định hướng đi của đạn dựa vào trục tọa độ Ox, trong đó xe tăng xanh sẽ có đường đạn đi từ trái sang phải do đó x + 7, ngược lại xe tăng đỏ sẽ có đường đạn ngược lại sẽ có đường đạn x – 7.



**Tô đỏ:** để đảm bảo tính logic cho và chơi và quan trọng hơn hết chính là tối ưu tài nguyên của máy tính, tránh để máy tính xử lí những thông tin không cần thiết, như đạn sau khi bắn ra không cần dùng đến thì sẽ xóa bỏ chúng.

Trong đó có hai trường hợp ta sẽ xóa đạn của chúng, một là khi đạn bắn trúng đối phương, hai là khi đạn đã bắn ra khỏi bản đồ, ta sẽ xét điều kiện để chúng thỏa và xóa chúng.

1. **Thông báo kết thúc trò chơi**

|  |
| --- |
| # In ra thông báo thông báo thắng hoặc thua  def draw\_winner(text):      draw\_text = WINNER\_FONT.render(text, 1, WHITE)      WINDOWS.blit(draw\_text, (WIDTH/2 - draw\_text.get\_width() /2, HEIGHT/2 - draw\_text.get\_height()/2))      pygame.display.update()      pygame.time.delay(5000) # Đấu lại sau khoảng thời gian |

Khi một trong hai người chơi hết mạng, thỏa điều kiện trong hàm ta sẽ viết ở sau, thì chương trình sẽ ngắt đồng thời hiện lên dòng chữ thông báo bên nào thắng.

Chart, box and whisker chart

Description automatically generated

Dòng chữ đó sẽ được đặt tại vị trí giao điểm giữa phần chiều cao (Height) và chiều rộng (Width).

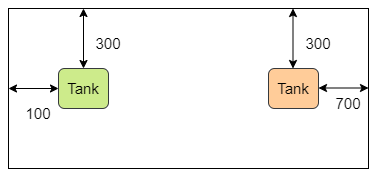
Và sau một khoảng thời gian nhất định, trò chơi sẽ được tự động lặp lại để chơi lại từ đầu.

1. **Xử lí các sự kiện chính và các trường hợp xảy ra trong lúc chơi (hàm Main)**

**a. Một số biến và list cơ bản để dễ thao tác**

|  |
| --- |
| green = pygame.Rect(100, 300, TANK\_WIDTH, TANK\_HEIGHT)  red = pygame.Rect(700, 300, TANK\_WIDTH, TANK\_HEIGHT)      red\_vien\_dan = []      green\_vien\_dan = []      red\_mang = 10      green\_mang = 10  MAX\_vien\_dan = 3 # Số lượng đạn tối đa có thể bắn trong 1 đợt  TANK\_WIDTH, TANK\_HEIGHT = 55, 40 |

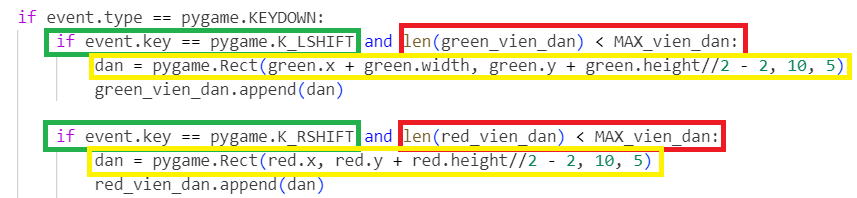
Ta khai báo kích thước (tank height và width) và vị trí ban đầu của hai xe tăng, chẳng hạn xe tăng xanh sẽ có vị trí (100,300), xe tăng đỏ sẽ có vị trí (700,300).



Ta khai báo hai list chứa vị trí của viên đạn để lát nữa có thể thao tác xóa bỏ hoặc xử lí va chạm cho dễ dàng.

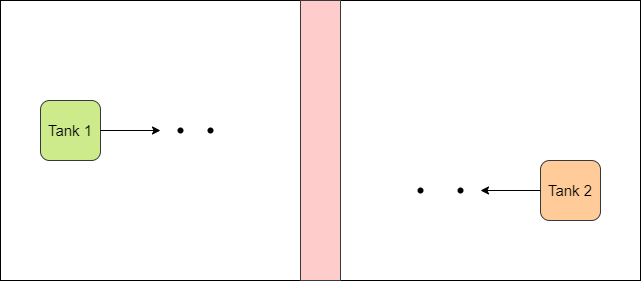
Tạo biến lưu số mạng của mỗi bên, số viên đạn tối đa trong một lần bắn và kích thước của xe tăng, ta cũng sẽ xử lí ở phía dưới.

**b. Cơ chế bắn đạn của cả hai xe tăng**



**Tô xanh:** điều kiện để xử lí khi nhận thông tin từ người chơi ấn phím SHIFT trái (xe tăng xanh), SHIFT phải (xe tăng đỏ).

**Tô đỏ:** điều kiện để được bắn tiếp là số đạn trên bản đồ phải nhỏ hơn số đạn tối đa được bắn, ví dụ ta set Max là 3 thì chỉ được phép bắn 3 phát đạn vào một đợt.



**Tô vàng:** Vị trí của viên đạn khi bắn ra sẽ phụ thuộc vào vị trí của chiếc xe tăng đó, ta xét dựa vào Ox để viên đạn đi theo hướng được định sẵn từ trước, ở đây ta phải viết thêm điều kiện đi để viên đạn đi được qua phía bên kia biên giới trong đó 5 x 10 tức là chiều dài và rộng của viên đạn.

**c. Xử lí sự kiện bị bắn và đếm mạng**

|  |
| --- |
| GREEN\_HIT = pygame.USEREVENT + 1 # Mã sự kiện 1  RED\_HIT = pygame.USEREVENT + 2 # Mã sự kiện 2 |

Ta gọi sự kiện xe tăng xanh bị bắn là 1, xe tăng đỏ bị bắn là 2.

|  |
| --- |
| pygame.event.post(pygame.event.Event(RED\_HIT))  pygame.event.post(pygame.event.Event(GREEN\_HIT)) |

|  |
| --- |
| if event.type == RED\_HIT:                  red\_mang -= 1  if event.type == GREEN\_HIT:                  green\_mang -= 1 |

Trong đó nếu sự kiện xảy ra thì ta sẽ kiểm tra chúng, nếu thỏa điều kiện thì mạng của xe tăng xanh hoặc đỏ sẽ giảm một mạng (bắt đầu là 10).

**d. Kết thúc trò chơi bằng thông báo chiến thắng**

|  |
| --- |
| winner\_text = ""          if red\_mang <= 0:              winner\_text = "GREEN WIN !"          if green\_mang <= 0:              winner\_text = "RED WIN !"          if winner\_text != "":              draw\_winner(winner\_text)              break |

Điều kiện để có thể xuất ra thông báo này phụ thuộc vào số mạng của một trong hai xe tăng, khi một trong hai xe tăng giảm tới 0 thì thông báo cho xe tăng còn lại là “chiến thắng”.

Ta lại kiểm tra thông báo một lần nữa, một khi thông báo đã xuất hiện thì sẽ dừng chương trình.

**e. Tự động reset lại trận đấu khi kết thúc**

|  |
| --- |
| pygame.time.delay(5000) # Đấu lại sau khoảng thời gian  ...   run = True      while run: |

Ta tạo một biến bool là run, mặc định ban đầu sẽ là True tức là trò chơi đang chạy, nếu trò chơi kết thúc thì nó sẽ trở thành False và sau một khoảng thời gian nhất đinh là 5000 nó sẽ reset biến run trở lại thành True, và tức là vòng lặp thỏa điều kiện, chương trình sẽ bắt đầu lại từ đầu.

**f. Thêm âm thanh, hiệu ứng tạo cảm giác chân thật, sống động.**

|  |  |
| --- | --- |
| Gọi nhạc nền khi chương trình bắt đầu thiết lặp âm lượng 50% | SOUND\_TRACK.play()  SOUND\_TRACK.set\_volume(0.5) |
| Gọi hiệu ứng “trúng đạn” khi đạn bắn trúng xe tăng. | BULLET\_HIT\_SOUND.play() |
| Khi xe tăng khi khai hỏa đạn | BULLET\_FIRE\_SOUND.play() |
| Ngắt nhạc khi trò chơi kết thúc | SOUND\_TRACK.stop() |

Cuối cùng gọi lại các hàm đã viết bên trên và gọi lại hàm main để thực thi trò chơi.

**PHẦN 3: CHẾ ĐỘ SINH TỒN**

**CHƯƠNG 1. GIỚI THIỆU**

Để tối ưu hóa trò chơi làm cho cách chơi đơn giản hơn nữa, phù hợp với mọi lứa tuổi và thiết bị, ta sử dụng cơ chế điều khiên bằng chuột hoặc điều khiển bằng cảm ứng trên các dòng điên thoại cảm ứng với duy nhất một thao tác điều khiển duy nhất.

**CHƯƠNG 2. HƯỚNG DẪN CÁCH CHƠI**

**Bước 1:** Cách chơi rất đơn giản, ta chỉ thao tác điều khiển xe tăng bằng chuột hoặc touchpad trên các dòng laptop, rê chuột sang sang hướng nào thì xe tăng sẽ di chuyển sang hướng đó.

A black computer mouse

Description automatically generatedA picture containing microwave, oven, appliance, kitchen appliance

Description automatically generated

**Bước 2:** Xe tăng sẽ tự động tiến lên và không thể dừng lại, vì thế hãy cố gắng nhặt những vật phẩm “kim cương” để tích điểm đồng thời né tránh các chướng ngại vật để có thể sống sót lâu nhất và tích điểm nhanh nhất.

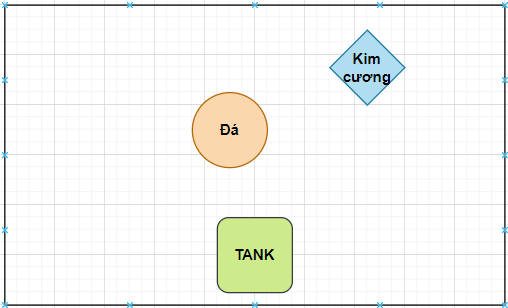
**A picture containing shape

Description automatically generated**

**Lưu ý:** Khi người chơi thu thập đủ nhiều kim cương thì tốc độ của trò chơi sẽ tăng lên càng lúc càng cao để tăng thêm độ khó cho trò chơi.

**CHƯƠNG 3. Ý TƯỞNG THỰC HIỆN**

**Bước 1:** Đưa ba đối tượng xe tăng, vật phẩm (kim cương), chướng ngại vật (đá) lên bản đồ.

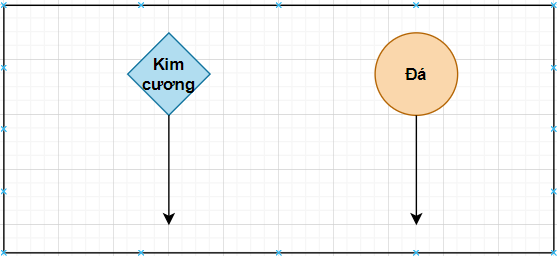


**Bước 2:** Để trò chơi trên nên thú vị, khó đoán trước được thì ta sẽ để vật phẩm và chướng ngại vật tự xuất hiện một cách ngẫu nhiên trong giới hạn của bản đồ.

A picture containing diagram

Description automatically generated

**Bước 3:** Thay vì xử lí để cửa sổ góc nhìn thay đổi thì ta để vật phẩm và chướng ngại vật rơi xuống tạo cảm giác như xe tăng ta đang di chuyển tốc độ cao.



**Bước 4:** Hướng người chơi đến việc né các chướng ngại vật đồng thời nhặt các vật phẩm.

A picture containing diagram

Description automatically generated

**CHƯƠNG 4. MÔ HÌNH HÓA CHƯƠNG TRÌNH**

1. **Cơ chế quá trình hoạt động**

**A picture containing text, clock, screenshot

Description automatically generated**

1. **Mô tả chi tiết**

|  |  |
| --- | --- |
| **Xe tăng** | + Cơ chế di chuyển theo chuột  + Xử lí khi va chạm với đá hoặc kim cương |
| **Đá (Rock)** | + Cơ chế di chuyển xuống dưới  + Tốc độ tăng dần theo điểm tích lũy |
| **Kim cương (items)** | + Cơ chế di chuyển xuống dưới  + Tốc độ tăng dần theo điểm tích lũy  + Biến mất khi tương tác với xe tăng |
| **Thông báo** | + Xuất thông báo kèm điểm số khi trò chơi kết thúc |
| **Hình ảnh, âm thanh** | + Lấy hình ảnh từ đường dẫn và đưa lên cửa sổ  + Lấy âm thanh và gọi lại sau mỗi khi xảy ra một sự kiện |

**CHƯƠNG 5. QUY TRÌNH THỰC HIỆN CHƯƠNG TRÌNH**

1. **Thư viện cần dùng**

|  |
| --- |
| import pgzrun, random  from random import randint |

Vì trò chơi của chúng ta sử dụng có quy mô nhỏ, xử lí sự kiện đơn giản nên chỉ sử dụng thư viện con của pygame là pgzero để xử lí, trong đó để các vật thể như vật phẩm và chướng ngại vật không lặp lại nhiều lần gây nhàm chán thì ta sẽ sử dụng thư viện random để chúng xuất hiện một cách ngẫu nhiên.

1. **Cài đặt một số biến cơ bản**

|  |
| --- |
| WIDTH, HEIGHT = 800, 600  WHITE, VIOLET = (255,255,255), (255,0,255)  diem = 0  ket\_thuc = False  background.draw() |

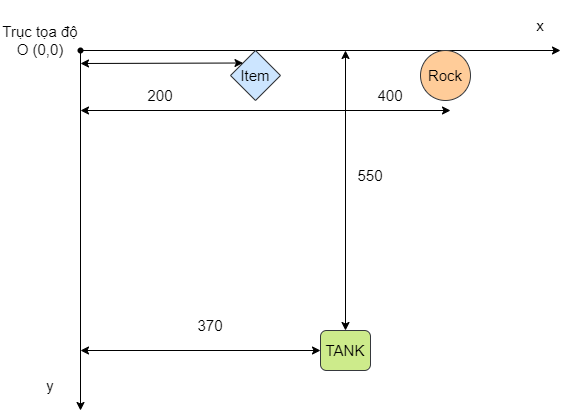
Ta thiết lặp kích thước cửa sổ là 800 x 600 không quá lớn để di chuyển và không quá nhỏ để thuận tiện cho quá trình điều khiển bằng chuột, cuối cùng là set hình nền và lưu giá trị các mã màu cho các biến để dễ dàng gọi lại.

Ta sử dụng biến điểm để tính điểm khi xe tăng của ta nhặt các vật phẩm, biến “kết thúc” để kiểm tra xem ta có bị va chạm với chướng ngại vật không.

1. **Định dạng hình ảnh và vị trí**

|  |  |
| --- | --- |
| # Định dạng hình nền  background = Actor("bg")  # Định dạng xe tăng  tank = Actor('tank')  tank.x, tank.y = 370, 550 | # Định dạng vật phẩm  item = Actor('item')  item.x, item.y = 200, 0  # Định dạng chướng ngại vật  rock = Actor('rock')  rock.x, rock.y = 400, 0 |

Ta sẽ gán đường dẫn của các hình ảnh cho các biến để dễ thao tác với chúng, vị trí của chúng được xác định dựa trên trục tọa độ Oxy.



Cụ thể vị trí mặc định ban đầu của xe tăng là x = 370, y = 550, vật phẩm là y = 0, x= 200, đá là y = 0, x= 400. Đây là vị trí tạm thời của chúng, vì khi chơi xe tăng sẽ được điều khiển bởi chuột, còn đá và kim cương sẽ xuất hiện một cách ngẫu nhiên.

1. **Cơ chế điều khiển xe tăng**

Diagram

Description automatically generated

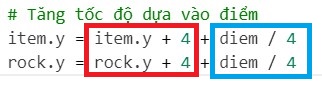
|  |
| --- |
| # Cơ chế điều khiển xe tăng bằng chuột  def on\_mouse\_move(pos,rel,buttons):      tank.x = pos[0]      tank.y = pos[1] |

Hàm này sẽ lấy vị trí của chuột sau đó cập nhật vị trí x và y trên trục tọa độ cho xe tăng, do đó chuột của ta nằm ở đâu thì xe tăng sẽ ở vị trí đó.

1. **Tăng tốc độ của trò chơi**

**Diagram

Description automatically generated with medium confidence**

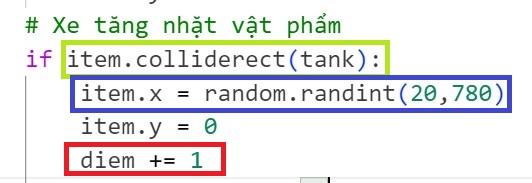
****

**Tô đỏ:** tốc độ ban đầu của vật thể và chướng ngại vật.

**Tô xanh:** Số điểm ta nhận được trong mỗi lần nhặt vật phẩm được chia 4 để đảm bảo tiến độ chơi không quá nhanh.

Sau đó sẽ được cập nhật liên tục vào để tốc độ rơi xuống càng lúc càng nhanh tăng.

1. **Xử lí khi xe tăng nhặt vật phẩm**



**Tô xanh lá:** Ta dùng hàm colliderect để kiểm tra xem xe tăng có “va chạm” với các vật phẩm item không.

**Tô đỏ:** nếu đúng ta sẽ đếm tăng số điểm lên một, cứ như thế cho đến khi kết thúc trò chơi.

**Tô xanh dương:** Ta dùng hàm random để vật phẩm xuất hiện ngẫu nhiên theo trục Ox tức là theo chiều ngang, vật phẩm sẽ xuất hiện tại vị trí đầu của trục Oy và rơi xuống.

**Lưu ý:** để tránh trường hợp hàm random hoạt động sai, ví dụ chúng random các vật phẩm ra khỏi kích thước của bản đồ, ta phải giới hạn vị trí của chúng.

|  |
| --- |
| # Xử lí giới hạn của màn hình      if item.y > 600:          item.y = 0      if rock.y > 600:          rock.y = 0 |

1. **Xử lí khi xe tăng va trúng chướng ngại vật**

|  |
| --- |
| # Xe tăng tông trúng cục đá      if rock.colliderect(tank):          ket\_thuc = True |

Khi va chạm xảy ra giữa chướng ngại vật và xe tăng xảy ra khi kiểm tra bằng hàng colliderect thì biến “kết thúc” sẽ là True, thỏa điều kiện kết thúc chương trình, vì thế chương trình sẽ ngắt và hiện lên thông báo.

|  |
| --- |
| if ket\_thuc:          screen.draw.text('Tro choi ket thuc',(360,300), color = WHITE, fontsize=60)          screen.draw.text('Tong diem: '+ str(diem),(360,350), color = VIOLET, fontsize=80) |

Hai thông báo trò chơi kết thúc và tổng điểm hiện ra ở cuối chương trình theo vị trí được thiết lập sẵn để người chơi biết được số điểm cuối cùng của mình.

1. **Thêm hiệu ứng âm thanh**

|  |  |
| --- | --- |
| Nhạc nền | sounds.astro.play() |
| Âm lượng nhạc nền | sounds.astro.set\_volume(0.5) |
| Va vào đá | sounds.bumm.play() |
| Nhặt vật phẩm | sounds.bruh.play() |
| Ngừng nhạc nền | sounds.astro.stop() |

**----- HẾT -----**