

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

Tên đề tài

Xây dựng game Tank

GVHD: ThS.Từ Lăng Phiêu
Nhóm 28

SV: 3120410390 - Võ Lê Trường Phát
3120410166 - Ngô Thanh Hiệp
3119410471 - Lê Minh Trường

TP. HỒ CHÍ MINH, THÁNG 5/2024

Mục lục

1	Phần giới thiệu	2
1.1	Giới thiệu game Tank	2
1.2	Chức năng của phần mềm	2
1.3	Hướng dẫn cách chơi-Sử dụng các phím	2
1.4	Cơ chế hoạt động	3
2	CƠ SỞ LÝ THUYẾT	4
2.1	Ngôn ngữ lập trình Python	4
2.2	Phần mềm lập trình Visual Studio Code	4
2.3	Thư viện hỗ trợ	4
3	Thiết kế ứng dụng	6
3.1	Ý tưởng trò chơi	6
3.2	Sử dụng thư viện	7
3.3	Thực hiện trò chơi	7
4	KẾT LUẬN	18

1 Phần giới thiệu

1.1 Giới thiệu game Tank

Như chúng ta đã biết, hiện nay đã có rất nhiều game bắn xe tăng khác nhau, dường như trò chơi bắn xe tăng đã trở nên quá phổ biến. Để có thể đáp ứng một cách tốt nhất thị hiếu cho người chơi thì game cần phải có những tính năng ưu việt hơn các phần mềm khác.

Chúng ta sẽ tìm hiểu về phát triển một trò chơi bắn xe tank sử dụng ngôn ngữ lập trình Python. Trò chơi này sẽ cho phép hai người chơi trên cùng một máy tính và cạnh tranh để tiêu diệt xe tank của đối thủ.

Một trong những tính năng quan trọng của phần mềm play audio là khả năng điều khiển phát nhạc, bao gồm chức năng play, pause, stop, next, previous, shuffle, repeat... Người dùng có thể dễ dàng chọn và thay đổi các chức năng này bằng cách sử dụng các nút được hiển thị rõ ràng trên giao diện chính.

1.2 Chức năng của phần mềm

Các chức năng cơ bản của game tank bao gồm:

- Chế độ đa người chơi: Phần mềm cho phép hai người chơi tham gia vào trò chơi cùng một máy tính. Mỗi người chơi điều khiển một chiếc xe tank và mục tiêu là tiêu diệt xe tank của đối phương.
- Điều khiển xe tank: Người chơi có thể điều khiển chiếc xe tank của mình bằng cách sử dụng các phím mũi tên hoặc các phím điều khiển được đặt trước.
- Bắn và tiêu diệt đối thủ: Mục tiêu chính của trò chơi là tiêu diệt xe tank của đối thủ bằng cách sử dụng vũ khí trên xe tank của mình. Người chơi điều khiển nút bắn để nhắm chính xác vào đối thủ.
- Trường chiến động rất đa dạng: Trò chơi cung cấp nhiều trường chiến động đất khác nhau với các vật cản và cấu trúc để tạo ra sự thú vị và thách thức cho người chơi.
- Điểm số và thời gian chơi: Phần mềm ghi nhận điểm số của mỗi người chơi dựa trên số lần tiêu diệt đối thủ.
- Âm thanh và hiệu ứng hình ảnh: Phần mềm có âm thanh và hiệu ứng hình ảnh sống động, tạo ra một trải nghiệm chơi game độc đáo và thú vị.

1.3 Hướng dẫn cách chơi-Sử dụng các phím

Lúc vào game người chơi sẽ được chọn xe tăng mình thích. Có 2 loại xe tăng:

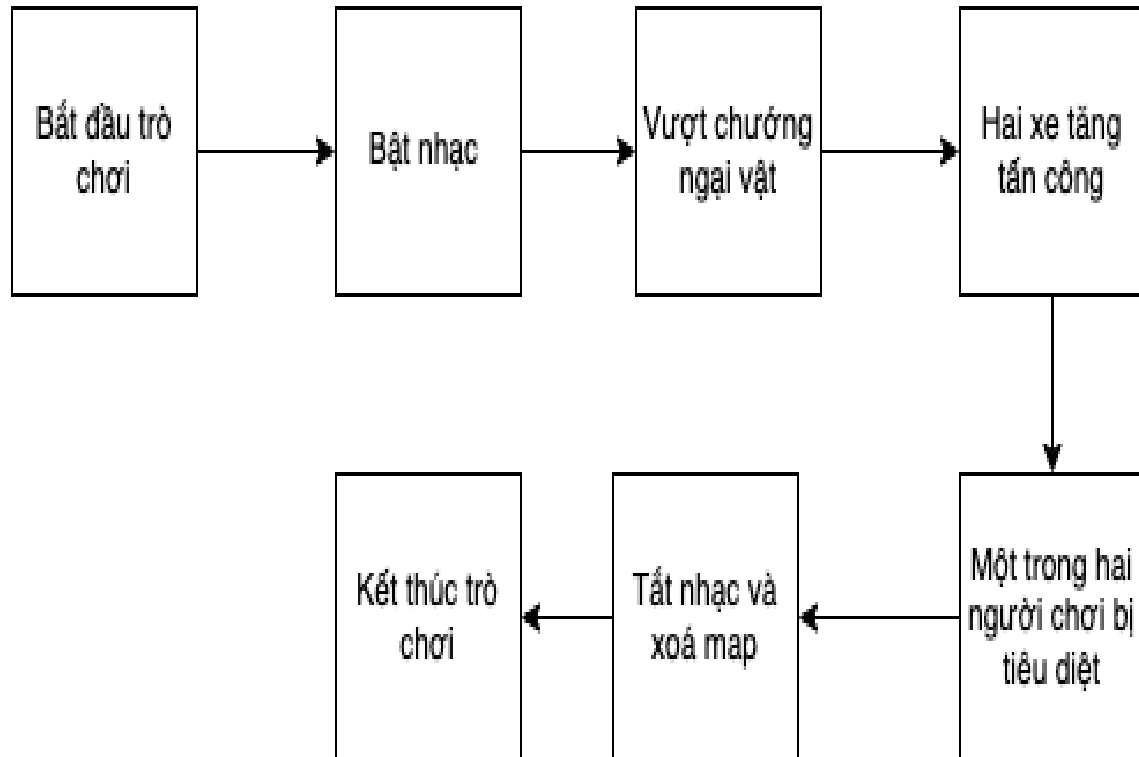
- loại 1: Ở phần xe tăng này sẽ có health là 3, Speed là 5.
- loại 2: Ở phần xe tăng này sẽ có health là 5, Speed là 3.

Về phần sử dụng phím để di chuyển:

Ở người chơi 1: Sử dụng các phím W,A,S,D để di chuyển và dùng phím 1 để bắn ra những viên đạn đến đối thủ.

Ở người chơi 2: Sử dụng các phím mũi tên lên xuống trái phải để di chuyển và dùng phím m để bắn ra những viên đạn đến đối thủ.

1.4 Cơ chế hoạt động



2 CƠ SỞ LÝ THUYẾT

2.1 Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

Những lợi ích của Python bao gồm:

- Các nhà phát triển có thể dễ dàng đọc và hiểu một chương trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh.
- Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một chương trình Python.
- Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.
- Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++.
- Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, bạn sẽ có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.
- Trên Internet có rất nhiều tài nguyên hữu ích nếu bạn muốn học Python. Ví dụ: bạn có thể dễ dàng tìm thấy video, chỉ dẫn, tài liệu và hướng dẫn dành cho nhà phát triển.
- Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix.

2.2 Phần mềm lập trình Visual Studio Code

Visual Studio Code chính là ứng dụng cho phép biên tập, soạn thảo các đoạn code để hỗ trợ trong quá trình thực hiện xây dựng, thiết kế website một cách nhanh chóng. Visual Studio Code hay còn được viết tắt là VS Code. Trình soạn thảo này vận hành mượt mà trên các nền tảng như Windows, macOS, Linux. Hơn thế nữa, VS Code còn cho khả năng tương thích với những thiết bị máy tính có cấu hình tầm trung vẫn có thể sử dụng dễ dàng.

Visual Studio Code mang rất nhiều ưu điểm vượt trội so với bất kỳ IDE nào khác:

- Hỗ trợ đa nền tảng: Linux, Mac, Windows, ...
- Hỗ trợ đa ngôn ngữ: C/C++, JavaScript, JSON, Visual Basic, HTML, CSS, ...
- Ít dung lượng
- Tính năng mạnh mẽ
- Intellisense chuyên nghiệp
- Giao diện thân thiện
- Kiến trúc mạnh mẽ và người dùng có thể khai thác mở rộng
- Số lượng người sử dụng lớn tạo nên cộng đồng hỗ trợ rộng rãi

2.3 Thư viện hỗ trợ

Thư viện Pygame

Pygame là một thư viện mã nguồn mở cho phép lập trình game và đa phương tiện trên Python. Thư viện này cung cấp các công cụ cho việc tạo các ứng dụng đa phương tiện, chẳng hạn như các trò chơi, âm nhạc, đồ họa và các ứng dụng tương tự.

Ưu điểm của Pygame:



- Dễ học: Pygame có cú pháp đơn giản và dễ hiểu, cho phép người dùng mới bắt đầu nhanh chóng học được cách sử dụng thư viện.
- Đa nền tảng: Pygame có thể hoạt động trên nhiều nền tảng khác nhau, bao gồm Windows, Mac OS X, Linux và nhiều hệ điều hành khác.
- Đa dạng: Pygame có nhiều tính năng và hỗ trợ đầy đủ cho các yêu cầu phức tạp trong lập trình game và đa phương tiện.
- Cộng đồng lớn: Pygame có một cộng đồng lớn với nhiều tài liệu hướng dẫn và các ví dụ, giúp người dùng tìm hiểu và giải quyết các vấn đề.

Nhược điểm của Pygame:

- Khả năng tối ưu hóa không tốt: Pygame không được tối ưu hóa tốt cho các ứng dụng có đồ họa phức tạp, do đó, nếu không được xử lý tốt, có thể dẫn đến giảm hiệu suất.
- Hạn chế đồ họa: Pygame không hỗ trợ các tính năng đồ họa cao cấp như DirectX hay OpenGL, vì vậy không thể sử dụng để tạo ra các trò chơi và ứng dụng đồ họa cao cấp.
- Không được bảo trì thường xuyên: Pygame không được bảo trì và cập nhật thường xuyên, do đó, có thể không hoạt động tốt với các phiên bản Python mới nhất.
- Không hỗ trợ 3D: Pygame không hỗ trợ lập trình 3D, do đó không phù hợp để tạo ra các trò chơi và ứng dụng có đồ họa 3D phức tạp.

3 Thiết kế ứng dụng

3.1 Ý tưởng trò chơi



Trò chơi bắt đầu bằng việc import thư viện pygame, khởi động Pygame, thiết lập Pygame Mixer cho âm thanh, khởi tạo màu sắc và biến font. Sau đó, tạo cửa sổ trò chơi, thiết kế hình nền, nạp âm thanh, hình ảnh bản đồ và xe tăng, đặt điểm xuất hiện xe tăng, tạo nhóm người chơi, nhân vật, thêm xe tăng, quản lý tường và viên đạn, và cuối cùng đặt biến điều khiển cho hàm và vòng lặp trong trò chơi.

Ghi điểm vào file điểm khi có, thiết lập số khung hình trên giây để chạy mượt mà, gọi hàm menu để hiển thị giao diện người chơi và xử lý sự kiện chuột. Khi người dùng chọn "chơi trò chơi", hàm menu sẽ kết thúc và dừng phát nhạc nền. Khi chọn "thoát trò chơi", hàm menu sẽ kết thúc, đặt giá trị false cho biến điều khiển và dừng phát nhạc.

Nhóm tải và phát nhạc nền khi call màn hình lựa chọn xe tăng, hiển thị tên các mục lựa chọn và hình ảnh, yêu cầu người chơi sử dụng phím điều khiển để chuyển đổi xe tăng. Khi người chơi nhấn Enter, xác nhận xe tăng được chọn và cập nhật thông số. Sau đó, chọn bản đồ và kết thúc hàm lựa chọn. Nếu người chơi thoát, hàm sẽ kết thúc và dừng phát nhạc.

Nhóm hẹn giờ cho việc bắn đạn khi màn hình chiến đấu được kích hoạt, sau đó nạp và phát nhạc nền. Trong vòng lặp, họ di chuyển người chơi, xử lý việc bắn đạn và đóng cửa sổ trò chơi. Duyệt qua đạn để xử lý va chạm và hiển thị trạng thái trên màn hình. Khi sức khỏe của người chơi giảm về 0, kết thúc trò chơi và dừng nhạc nền.

Nhóm sẽ tải và phát nhạc nền khi hàm kết thúc trò chơi được gọi, sau đó cập nhật điểm số từ tệp và xác định người chiến thắng. Người chiến thắng sẽ được cộng thêm một điểm và điểm số mới sẽ được ghi lại. Tiêu đề kết quả sẽ được hiển thị trên màn hình và trong vòng lặp, điểm số sẽ được hiển thị và xử lý sự kiện như dừng phát nhạc, thiết lập lại điểm số hoặc kết thúc trò

chơi sẽ được thực hiện.

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. Đưa xe tăng của hai người chơi lên bản đồ, player 1 đặt ở góc trái bên trên map, player 2 đặt ở góc dưới bên phải map.

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

Để tạo ra một trải nghiệm chơi đầy sinh động và hấp dẫn hơn, nhóm em sẽ thiết kế các vách tường gỗ đặc biệt, những bức tường chắn đạn sẽ ngăn cản việc bắn trực tiếp vào nhau. Điều này sẽ tạo ra các tình huống chiến đấu trong game trở nên chân thực hơn, khi người chơi phải tìm cách vượt qua các vách tường này để tiến xa hơn trong trận đấu

3.2 Sử dụng thư viện

Các thư viện được sử dụng:

```
import Math
import pygame from pygame.locals import *
import random
import time
pygame.init()
```

- Math: Thư viện toán học chuẩn của Python, cung cấp các hàm toán học như lũy thừa, căn bậc hai, các hàm lượng giác.

- pygame: Thư viện chuyên cho việc phát triển trò chơi, hỗ trợ tạo cửa sổ, xử lý đồ họa, âm thanh, và xử lý sự kiện.

- from pygame.locals import : Nhập tất cả các hằng số có sẵn từ module pygame.locals, thường được sử dụng để dễ dàng tham chiếu tới các nút bấm, hằng số hệ thống mà không cần phải gọi module pygame.locals.

- random: Thư viện hỗ trợ tạo ra các số ngẫu nhiên, sử dụng cho việc lựa chọn ngẫu nhiên, xáo trộn dữ liệu.

- time: Thư viện cung cấp các hàm để làm việc với thời gian, bao gồm đo hạn, đợi (sleep), và xuất lại thời gian hệ thống

- pygame.init(), thư viện pygame được khởi tạo, đây là bước cần thiết để bắt đầu làm việc với pygame.

3.3 Thực hiện trò chơi

***Khởi tạo menu:**


```
1 usage
def gameMenu(thisStage):
    #shows the game menu, for pseudocode plz see above
    '''
    ~~~~~
    parameters: thisStage variable, if it is true, function executes, otherwise, function will not
    return: none
    '''
    global run, selecting, battling, end    #needs to be global because the change to these variables
                                           #global update the changes outside of this function
    pygame.mixer.music.load(musicList[0])  #variables
    pygame.mixer.music.set_volume(0.4)
    pygame.mixer.music.play(-1)

    Title = my_font.render( text: 'T. O. D.', antialias: True, color: (212,212,212))
    Start_label = my_font2.render( text: 'Start Game', antialias: True, color: (212,212,212))
    Exit_label = my_font2.render( text: 'Exit Game', antialias: True, color: (212,212,212))

    Srect = Rect( *args: 210, 280, 421, 339)    #rects here are used to see if the mouse coordinates
    Erect = Rect( *args: 245, 340, 380, 430)
```

function gameMenu trong ứng dụng Pygame có vai trò thiết lập màn hình menu chính cho trò chơi. Ở trung tâm của hàm gameMenu là tham số thisStage, cho phép xác định các giai đoạn hoặc trạng thái cụ thể trong giao diện menu. Tham số này phục vụ như một yếu tố linh hoạt thích ứng với tương tác của người dùng, hướng dẫn họ qua menu một cách mượt mà.

Việc khai báo các biến toàn cục như run, selecting, battling và end cung cấp cho hàm sự linh hoạt để sửa đổi các giá trị này trong quá trình thực thi. Những biến này hoạt động như các cờ kiểm soát, ảnh hưởng đến hành vi của menu dựa trên đầu vào của người dùng và sự kiện hệ thống.

***Sự kiện thoát game**

```
while thisStage:

    x, y = pygame.mouse.get_pos()
    Opt = 0 #Opt means options, this variable smooth out t

    if Srect[0] < x < Srect[2] and Srect[1] < y < Srect[3]:
        my_font2.set_underline(True)
        Start_label = my_font2.render(text: 'Start Game', antialias: True, color: (212,212,212))

        screen.blit(Start_label, dest: (210,280))
        pygame.display.flip()

        Opt = 1 #if the mouse is over the titl
                #underline the titles,(a cool

    elif Erect[0] < x < Erect[2] and Erect[1] < y < Erect[3]:

        my_font2.set_underline(True)
        Exit_label = my_font2.render(text: 'Exit Game', antialias: True, color: (212,212,212))

        screen.blit(Exit_label, dest: (220, 340))
        pygame.display.flip()

    Opt = 2

for ev in pygame.event.get():
    if ev.type == QUIT:
        pygame.mixer.music.stop()
        run = False
        starting = False
        selecting = False
        battling = False
        end = False
        thisStage = False

    elif ev.type == MOUSEBUTTONDOWN:
        if Opt == 1:
            pygame.mixer.music.stop()
            starting = False
            thisStage = False
        elif Opt == 2:
            pygame.mixer.music.stop()
            run = False
            starting = False
            selecting = False
            battling = False
            end = False
            thisStage = False
```

Thiết lập của vòng lặp cho menu trò chơi. Quản lý việc hiển thị và tương tác của người dùng

với menu chính, xử lý việc bắt đầu hay thoát trò chơi dựa trên đầu vào từ chuột. Trong vòng lặp while này, các sự kiện chuột và tương tác người dùng được xử lý:

- Hàm `pygame.mouse.get_pos()` lấy vị trí chuột.
- Opt được khởi tạo với giá trị 0, có thể được sử dụng để theo dõi lựa chọn của người chơi trong menu.
- if `Srect[0] < x < Srect[2]` and `Srect[1] < y < Srect[3]`: kiểm tra nếu con trỏ chuột đang ở bên trên nút "Start Game". Nếu đúng, thì chữ "Start Game" được gắn dấu gạch dưới.
- elif `Erect[0] < x < Erect[2]` and `Erect[1] < y < Erect[3]`: thực hiện việc kiểm tra tương tự cho nút "Exit Game". Nếu đúng, thì chữ "Exit Game" được in đậm và màu đỏ, được thể hiện bằng nhãn Exit.

*Thiết lập màn hình chọn xe tăng

```
def selectionScreen(thisStage):  
  
    global run, selecting, battling, end  
    pygame.mixer.music.load(musicList[1])  
    pygame.mixer.music.set_volume(0.4)  
    pygame.mixer.music.play(-1)  
  
    Opt1 = 0  
    Opt2 = 0  
  
    instructionP1 = pygame.image.load('instruction2.png')  
    instructionP2 = pygame.image.load('instruction1.png')  
  
    Title = my_font3.render(text='Select Your Tank', antialias=True, color=(9,225,242))  
    p1Title = my_font4.render(text='P1', antialias=True, color=(212,212,212))  
    p2Title = my_font4.render(text='P2', antialias=True, color=(212,212,212))  
  
    instruction = my_font4.render(text='Use your left and right controls to switch between tanks.', antialias=True, color=(175,44,33))  
    instruction2 = my_font4.render(text='Press enter to start the battle!', antialias=True, color=(175,44,33))  
  
    Title = my_font3.render(text='Select Your Tank', antialias=True, color=(9,225,242))  
    p1Title = my_font4.render(text='P1', antialias=True, color=(212,212,212))  
    p2Title = my_font4.render(text='P2', antialias=True, color=(212,212,212))  
  
    instruction = my_font4.render(text='Use your left and right controls to switch between tanks.', antialias=True, color=(175,44,33))  
    instruction2 = my_font4.render(text='Press enter to start the battle!', antialias=True, color=(175,44,33))  
  
    options = [imageUp_v1, imageUp_v2]  
    stats = {0: ['3', '5', '3'], 1: ['5', '3', '1']}  
  
    DisplaySurf = pygame.Surface((100,100))  
    DisplaySurf.fill((255,255,255))  
    DisplaySurf2 = pygame.Surface((100,100))  
    DisplaySurf2.fill((255,255,255))  
  
    while thisStage:  
        for ev in pygame.event.get():  
            if ev.type == QUIT:  
                pygame.mixer.music.stop()
```

Quản lý quá trình chọn xe tăng và cài đặt các thông số trước khi bắt đầu trò chơi. Lựa chọn và thao tác được thực hiện thông qua các phím điều hướng và phím ENTER. Hàm này xử lý màn hình lựa chọn xe tăng trước khi bắt đầu trận chiến.



Ta thiết lập biến toàn cục gồm run, selecting, battling, và end. Hàm tải và phát nhạc nền từ musicList[1] với âm lượng đã được đặt là 0.4 và lặp lại vô hạn.

***Thiết lập định dạng cho xe tăng**

```
players = pygame.sprite.Group()

player1 = pygame.sprite.Sprite()
player1.rect = pygame.Rect((20, 20), (24, 24))
player1.direction = 'up'
player1.keys = (K_a, K_d, K_w, K_s, K_1)

player2 = pygame.sprite.Sprite()
player2.rect = pygame.Rect((560, 390), (24, 24))
player2.direction = 'up'
player2.keys = (K_LEFT, K_RIGHT, K_UP, K_DOWN, K_m)

players.add(player1)
players.add(player2)

bulletgroup = pygame.sprite.Group()
walls = pygame.sprite.Group()
```

- Khai báo hai danh sách ver1 và ver2 chứa các hình ảnh khác nhau cho hai phiên bản người chơi, với các hướng di chuyển khác nhau được biểu diễn qua các hình ảnh lên, xuống, phải và trái.

- Tạo nhóm players sử dụng pygame.sprite.Group() để quản lý các đối tượng người chơi.

- Tạo ra hai đối tượng player1 và player2 bằng cách sử dụng pygame.sprite.Sprite(). Mỗi đối tượng người chơi được cấp một hình chữ nhật (rect) để xác định vị trí và kích thước, một thuộc tính direction để biết hướng di chuyển, và một set các keys để xác định các phím điều khiển di chuyển và bắn đạn.

- Thêm hai đối tượng player1 và player2 vào nhóm players bằng phương thức add.

- Nhóm bulletgroup sử dụng pygame.sprite.Group(), dùng để quản lý các viên đạn trong trò chơi.

- Nhóm walls cũng dùng pygame.sprite.Group(), dùng để quản lý các bức tường trong game.

***Cách thức hoạt động điều khiển xe tăng**

```
def movement(player):  
    key = pygame.key.get_pressed()  
  
    if key[player.keys[0]]:  
        player.rect.left -= player.speed  
        player.direction = 'left'  
        player.image = player.pics[3]  
  
    elif key[player.keys[1]]:  
        player.rect.left += player.speed  
        player.direction = 'right'  
        player.image = player.pics[2]  
  
    elif key[player.keys[2]]:  
        player.rect.top -= player.speed  
        player.direction = 'up'  
        player.image = player.pics[0]  
  
    elif key[player.keys[3]]:  
        player.rect.top += player.speed  
        player.direction = 'down'  
        player.image = player.pics[1]  
  
if len(pygame.sprite.spritecollide(player, players, dokill: False)) > 1 or pygame.sprite.spritecollide(player, walls, dokill: False):  
  
    if player.direction == 'left':  
        player.rect.left += player.speed  
    if player.direction == 'right':  
        player.rect.left -= player.speed  
    if player.direction == 'up':  
        player.rect.top += player.speed  
    if player.direction == 'down':  
        player.rect.top -= player.speed
```

- Hàm movement nhận đối tượng player làm đối số và xử lý các sự kiện đầu vào từ bàn phím để di chuyển player.
- Hàm sử dụng biến key để lấy trạng thái hiện tại của các phím được bấm.
- Mỗi phím kiểm soát một hướng di chuyển cụ thể của player (sang trái, sang phải, lên trên, xuống dưới) và cập nhật hình ảnh player tương ứng với hướng đó.
- Hàm cũng xử lý va chạm để ngăn player không di chuyển xuyên qua các đối tượng khác như người chơi hoặc tường: + Sử dụng pygame.sprite.spritecollide để kiểm tra va chạm giữa player và nhóm players hoặc walls.
 - + Nếu phát hiện va chạm len() của danh sách va chạm lớn hơn 1 hoặc spritecollide trả về True, player sẽ phải di chuyển theo hướng ngược lại để tránh tình trạng mắc kẹt.
 - + Đối với mỗi hướng di chuyển, nếu phát hiện va chạm, player sẽ được đẩy ngược lại một khoảng bằng player.speed để không đè lên player khác hoặc các bức tường trong trò chơi.

***Tạo map**

```
def readMapQ():  
    Map = open(random.choice(maps), 'r')  
  
    x = 0  
    y = 0  
  
    for l in Map:  
        builtin = l.split(',')  
        builtin[-1] = builtin[-1].strip('\n')  
        for D in builtin:  
            if D == '.':  
                tile = pygame.sprite.Sprite()  
                tile.image = tiles  
                tile.rect = pygame.Rect(x, y, 11, 11)  
                walls.add(tile)  
                y += 11  
            else:  
                y += 11  
        x += 11  
        y = 0  
  
    Map.close()
```

Hàm readMap được tạo ra để đọc dữ liệu từ bản đồ của trò chơi và thiết lập màn hình chơi dựa theo nó, sử dụng sprites để tạo ra các đối tượng "bức tường", và đặt chúng vào vị trí tương ứng trên màn hình chơi của trò chơi.

Lệnh open mở tệp, và random.choice chọn một tệp một cách ngẫu nhiên từ danh sách đó. Hai biến x và y được khởi tạo để theo dõi vị trí hiện tại trên bản đồ, nơi các đối tượng của trò chơi sẽ được đặt.

Sau đó duyệt qua từng dòng của tệp bản đồ (for l in Map):

- Mỗi dòng được chia (split) bởi dấu phẩy để tạo thành một danh sách các ký tự, mỗi ký tự đại diện cho một phần của bản đồ.

- Hàm loại bỏ ký tự xuống dòng n ở cuối mỗi dòng (builtin[-1] = builtin[-1].strip('\n')).

Tiếp theo, hàm duyệt qua từng ký tự trong dòng (for D in builtin):

- Nếu ký tự là dấu chấm (.), nó tượng trưng cho một "bức tường" trong trò chơi:
- Một đối tượng sprite mới được tạo ra (tile = pygame.sprite.Sprite()).
- Ảnh đại diện cho bức tường, biến tiles, được thiết lập làm hình ảnh của sprite này (tile.image = tiles).
- Đối tượng sprite mới đó được gán một hình chữ nhật (Rect) với vị trí (x, y) và kích thước (12, 12).
- Đối tượng sprite được thêm vào nhóm walls, đã được định nghĩa trước đây như một nhóm của spritesPygame, để dễ dàng quản lý các bức tường trong trò chơi.
- y được tăng thêm 12 để di chuyển đến vị trí tiếp theo trên cùng một dòng, nơi một sprite khác được đặt.
- Nếu ký tự không phải là dấu chấm, y vẫn được tăng lên và không có sprite nào được tạo ra.

***Xử lý màn hình trận đấu**

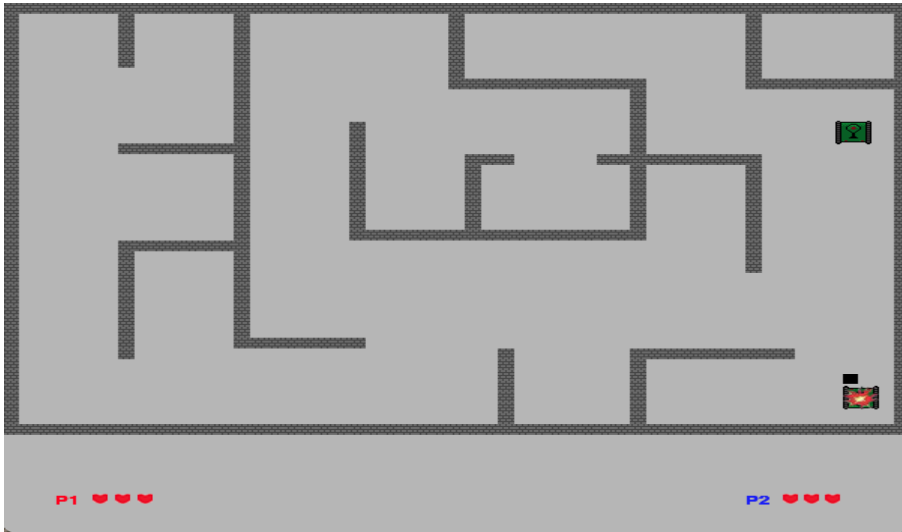
```
def battleScreen(thisStage):  
    global run, end  
    pygame.mixer.music.load(musicList[2])  
    pygame.mixer.music.set_volume(0.4)  
    pygame.mixer.music.play(-1)  
  
    timer = 0  
    timer2 = 0
```

Thiết lập và quản lý màn hình chiến đấu trong game, bao gồm việc khởi tạo âm nhạc và điều chỉnh âm lượng, bắt đầu vòng lặp, và khởi tạo bộ đếm thời gian cho các sự kiện trong trò chơi.

***Xử lý bắn đạn**

```
if key[player1.keys[4]]:  
    if timer >= player1.cooldown:  
        shoot.play()  
        bullet(player1)  
        timer = 0  
  
if key[player2.keys[4]]:  
    if timer2 >= player2.cooldown:  
        shoot.play()  
        bullet(player2)  
        timer2 = 0
```

Quản lý việc bắn đạn cho hai người chơi trong trò chơi. Khi người chơi nhấn nút bắn đạn (được chỉ định bởi player1.keys[4] và player2.keys[4]), và nếu thời gian đếm đã đủ lớn so với khoảng thời gian hồi chiêu (cooldown), âm thanh bắn đạn sẽ được phát, một viên đạn mới sẽ được tạo từ vị trí của người chơi đó, và bộ đếm thời gian sau đó sẽ được đặt lại về 0 để bắt đầu đếm cho lần bắn tiếp theo.



*Kiểm tra tank nổ khi trúng đạn

```
for bullets in bulletgroup:
    if pygame.sprite.collide_rect(bullets, player1):
        bulletgroup.remove(bullets)
        player1.health -= 1
        explode.play()
        screen.blit(explosion, player1.rect)
        pygame.display.flip()
        time.sleep(1)
        player1.rect.center = spawnpoints[random.randint(a: 0, b: 2)]

    if pygame.sprite.collide_rect(bullets, player2):
        bulletgroup.remove(bullets)
        player2.health -= 1
        explode.play()
        screen.blit(explosion, player2.rect)
        pygame.display.flip()
        time.sleep(1)
        player2.rect.center = spawnpoints[random.randint(a: 0, b: 2)]

    if pygame.sprite.spritecollide(bullets, walls, dokill: False):
        explode.play()
        bulletgroup.remove(bullets)

    if len(pygame.sprite.spritecollide(bullets, bulletgroup, dokill: False)) > 1:
        pygame.sprite.spritecollide(bullets, bulletgroup, dokill: True)
```

Xử lý va chạm của đạn với các nhân vật trong trò chơi và các đối tượng khác nhau trên màn hình. Khi một viên đạn từ bulletgroup va chạm vào người chơi player1 hoặc player2, viên đạn đó sẽ bị loại bỏ khỏi nhóm, người chơi sẽ mất một lượng máu, âm thanh nổ sẽ được phát và một hình ảnh vụ nổ sẽ xuất hiện tại vị trí của người chơi. Người chơi sau đó sẽ được chuyển đến điểm spawn xa nhất so với vị trí hiện tại của người chơi kia. Nếu viên đạn va chạm vào tường, nó cũng sẽ bị loại bỏ và phát ra âm thanh nổ.

Nếu có nhiều viên đạn va chạm vào nhau (được kiểm tra bởi len(pygame.sprite.spritecollide(bullets, bulletgroup, False)) > 1), tất cả những viên đạn va chạm sẽ được loại bỏ.

*Kết thúc ván đấu


```
if player1.health == 0 or player2.health == 0:  
    pygame.mixer.music.stop()  
  
    battling = False  
    thisStage = False  
    endScreen(end)
```

Âm nhạc trong trò chơi sẽ dừng lại sử dụng `pygame.mixer.music.stop()`

Quản lý việc kết thúc trận đấu khi một người chơi đã mất hết health và quá trình dọn dẹp liên quan sau trận đấu, xử lý tình huống nếu một trong hai người chơi hết máu (health bằng 0). Khi một người chơi hết máu

*Hiển thị kết quả

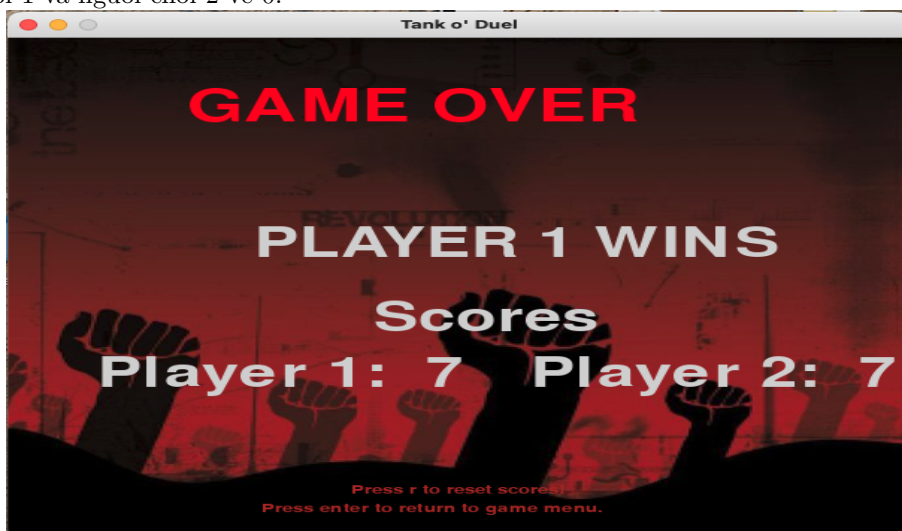
```
while thisStage:  
    p1Scores = my_font2.render('Player 1: ' + str(scoredata['1']), antialias=True, color=(212,212,212))  
    p2Scores = my_font2.render('Player 2: ' + str(scoredata['2']), antialias=True, color=(212,212,212))  
  
    screen.blit(screenPIC, dest=(0,0))  
    screen.blit(Title, dest=(120,50))  
    screen.blit(WinnerTitle, dest=(165,200))  
    screen.blit(scoreTitle, dest=(245,280))  
    screen.blit(p1Scores, dest=(60,340))  
    screen.blit(p2Scores, dest=(350,340))  
    screen.blit(instruction, dest=(230,480))  
    screen.blit(instruction2, dest=(170,500))  
    pygame.display.flip()  
  
    for ev in pygame.event.get():  
        if ev.type == QUIT:  
            pygame.mixer.music.stop()  
            run = False  
            end = False  
            thisStage = False  
        elif ev.type == KEYDOWN:  
            if ev.key == K_RETURN:  
                pygame.mixer.music.stop()  
                thisStage = False  
            elif ev.key == K_r:
```

```
for ev in pygame.event.get():
    if ev.type == QUIT:
        pygame.mixer.music.stop()
        run = False
        end = False
        thisStage = False
    elif ev.type == KEYDOWN:
        if ev.key == K_RETURN:
            pygame.mixer.music.stop()
            thisStage = False
        elif ev.key == K_r:
            scoredata['1'] = 0           #reset the scores if one player gets mad and wants to restart a
            scoredata['2'] = 0

scores = open('Scores Count.txt', 'w')
scores.write('1,' + str(scoredata['1']) + '\n')
scores.write('2,' + str(scoredata['2']) + '\n')           #final recording and updating just to be safe
scores.close()

starting = True
selecting = True
```

- Hàm sử dụng một vòng lặp while để liên tục kiểm tra xem màn hình kết thúc trò chơi đang được hiển thị hay không (thisStage là True).
- Trong mỗi lần lặp của vòng lặp này, hàm tạo ra các surface chứa text cho điểm số của người chơi 1 và người chơi 2 sử dụng font đã chỉ định trước và màu sắc cụ thể.
- Các surface này sau đó được xuất (blit) lên màn hình tại vị trí xác định. Cận cảnh của màn hình kết thúc trận đấu, tiêu đề chiến thắng, điểm số, và hướng dẫn cũng được xuất lên màn hình, pygame.display.flip() được gọi để cập nhật toàn bộ màn hình hiển thị với những thay đổi vừa được xuất lên.
- Hàm cũng xử lý sự kiện từ người dùng: Nếu phát hiện sự kiện thoát (QUIT), hàm sẽ dừng phát nhạc nền, cập nhật trạng thái của các biến điều khiển trò chơi (run, starting, selecting), và thoát khỏi vòng lặp while, dẫn đến kết thúc hàm. Nếu phát hiện sự kiện nhấn phím (KEYDOWN), hàm kiểm tra xem phím được nhấn là phím nào:
- Nếu phím Enter được nhấn, hàm sẽ dừng phát nhạc nền và thoát khỏi vòng lặp while, có thể là để trở về menu chính của trò chơi. phím R được nhấn, hàm sẽ reset điểm số của người chơi 1 và người chơi 2 về 0.





4 KẾT LUẬN

Trong quá trình thực hiện đồ án phát triển phần mềm mã nguồn mở Game tank, nhóm chúng em đã học hỏi được nhiều kỹ năng và kiến thức mới về lập trình, chúng em đã tạo ra một môi trường game thú vị với đồ họa hấp dẫn và âm thanh sống động. Dự án đã giúp chúng ta áp dụng và củng cố kiến thức về lập trình và phát triển game. Việc xây dựng game bắn xe tank không chỉ đòi hỏi kiến thức vững về lập trình mà còn yêu cầu khả năng tư duy logic và sáng tạo trong việc thiết kế gameplay và các tính năng. Tuy nhiên, vẫn còn một số cải tiến có thể thực hiện trong tương lai để nâng cao trải nghiệm của người chơi, bao gồm việc thêm vào các tính năng mới, tối ưu hóa hiệu suất và mở rộng phạm vi gameplay. Nhóm em xin chân thành cảm ơn!



Tài liệu

- [1] Documentation of Pygame "<https://www.pygame.org/docs/>"