

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO:
ĐỒ ÁN KẾT THÚC MÔN LẬP TRÌNH GAME

Đề tài:
GAME TETRIS

Giảng viên hướng dẫn: Nguyễn Thị Tuyết Hải

Sinh viên thực hiện: Huỳnh Ngô Anh Thái

Mã số sinh viên: N19DCCN188

Lớp: D19CQCN02-N

TPHCM, ngày 14 tháng 03 năm 2023

MỤC LỤC

LỜI CẢM ƠN.....	3
MỞ ĐẦU	4
CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI	5
1. Giới thiệu chung.....	5
1.1. Lịch sử của game Tetris.....	5
1.2. Tổng quan	5
1.3. Cách chơi	5
2. Mô tả đề tài.....	6
3. Link github:	6
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	7
1. Tổng quan về thư viện pygame	7
1.1. Lịch sử	7
1.2. Kiến trúc và tính năng tổng quan.....	7
2. Cài đặt và các tính năng được sử dụng trong pygame	7
2.1. Cài đặt pygame	7
2.2. Các tính năng chính được sử dụng trong pygame	8
CHƯƠNG 3: THỰC NGHIỆM.....	26
1. Màn hình chính	26
2. Giao diện trò chơi.....	29
KẾT LUẬN	35
Kết quả đạt được của đồ án.....	35
Hạn chế của đồ án	35
TÀI LIỆU THAM KHẢO	36

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến cô Nguyễn Thị Tuyết Hải – giảng viên bộ môn “Lập trình Game” trong khoa Công nghệ thông tin 2 đã trang bị cho em những kiến thức, kỹ năng cơ bản cần có để hoàn thành đồ án này.

Tuy nhiên trong quá trình nghiên cứu đồ án, do kiến thức chuyên ngành còn hạn chế nên em vẫn còn nhiều thiếu sót khi tìm hiểu, đánh giá và trình bày về đồ án. Rất mong nhận được sự quan tâm, góp ý của cô để đồ án của em được đầy đủ và hoàn thiện hơn.

Em xin chân thành cảm ơn.

TPHCM, ngày 14 tháng 03 năm 2023

MỞ ĐẦU

Trò chơi điện tử (video game) được ra mắt lần đầu tiên vào tháng 10 năm 1958, đến nay cũng đã hơn 60 năm và có những bước tiến vượt bậc về lối chơi cũng như chất lượng đồ họa. Từ những video game chuyển động trái phải lên xuống trên màn hình 2 chiều với đồ họa trắng đen, thì bây giờ con người đã có thể tạo ra những game có lối chơi đa dạng, đồ họa chân thật đến từng sợi tóc.

Trong quá trình đó, không thể không kể đến những video game được gọi là “huyền thoại”, tạo bước đệm cho việc hình thành nên các thể loại và lối chơi game sau này. Game TETRIS cũng thuộc một trong các số đó và đến nay TETRIS vẫn còn tồn tại đến tận bây giờ và là một trong số ít các game huyền thoại được đem ra thi đấu trên thế giới.

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1. Giới thiệu chung

1.1. Lịch sử của game Tetris

Tetris (tiếng Nga: Тетрис) là một trò chơi điện tử đầu tiên được thiết kế và phát triển bởi nhà khoa học máy tính người Liên Xô Alexey Pajitnov. Trò chơi được tạo ra vào ngày 6 tháng 6 năm 1984, trong lúc ông đang làm việc tại Trung tâm Tính toán Dorodnicyn của Viện hàn lâm Khoa học Liên Xô tại Moskva. Ông lấy tên của trò chơi từ tiền tố "tetra-" của tiếng Hy Lạp, có nghĩa là "bốn" (mỗi bộ phần trong trò chơi, gọi là Tetromino, có bốn phần) và quần vợt (*tennis*), trò thể thao Pajitnov thích nhất.

Trò chơi (hay một biến thể của nó) được bán cho hầu hết mọi máy trò chơi điện tử và hệ điều hành máy tính, cũng như trong các máy tính đồ họa, điện thoại di động, máy nghe nhạc di động, PDA, và nhiều thứ khác. Tuy nhiên biến thể của *Tetris* được bán trên thị trường vào thập niên 1980, trò chơi cầm tay của Game Boy, ra mắt năm 1989 đã biến trò chơi thành một trong những trò chơi thịnh hành nhất. Số thứ 100 của tạp chí *Electronic Gaming Monthly* gọi Tetris là "Trò chơi Vĩ đại nhất trong Mọi thời đại". Nó đã bán được hơn 70 triệu phiên bản.

1.2. Tổng quan

Trò chơi có bảy loại khối hình: I (thẳng đứng), J, L, O (vuông), S, T, Z. Ta thấy mỗi khối gạch được cấu tạo từ 4 hình vuông nhỏ xếp lại với nhau. Ta có thể coi các khối gạch đó như là những hình chữ nhật có kích thước khác nhau.

Các hình khác được tạo ra khi xoay các khối cơ bản này các góc tương ứng 90 độ, 180 độ, 270 độ.

Một chuỗi ngẫu nhiên của Tetriminos rơi xuống sân chơi (một trục đứng hình chữ nhật, được gọi là "tốt" hay "ma trận").

1.3. Cách chơi

Mục tiêu của trò chơi là di chuyển các khối gạch đang rơi từ từ xuống trong kích thước hình chữ nhật 20 hàng x 10 cột (trên màn hình). Chỗ nào có gạch rồi thì không di chuyển được tới vị trí đó. Người chơi xếp những khối hình sao cho khối hình lấp đầy 1 hàng ngang để ghi điểm và hàng ngang ấy sẽ biến mất.

Một nhóm 4 khối sẽ rơi từ phía trên cùng của màn hình, di chuyển các khối và xoay chúng cho đến khi chúng rơi xuống phía dưới cùng của màn hình, sau đó nhóm 4 khối tiếp theo sẽ rơi xuống.

Nếu để cho những khối hình cao quá màn hình, trò chơi sẽ kết thúc.

Trò chơi kết thúc khi khối gạch không rơi xuống được nữa.

Tất cả các Tetriminos có khả năng hoàn thành một và hai dòng. J, L có thể có ba. Chỉ có Tetrimino chữ I có khả năng để xóa bốn dòng cùng một lúc, và điều này được gọi là một "Tetris". Xóa nhiều nhất chỉ được 4 hàng/1 lần.

Phím tắt:

- Phím mũi tên lên: xoay khối.
- Phím mũi tên trái: di chuyển sang trái.
- Phím mũi tên phải: di chuyển sang phải.
- Phím mũi tên xuống: tăng tốc độ rơi.

2. Mô tả đề tài

Đề tài này được viết bằng ngôn ngữ python với sự trợ giúp chủ yếu từ thư viện pygame. Pygame là một bộ mô-đun Python đa nền tảng phục vụ cho việc viết các game.

Trong đề tài, sau khi chạy chương trình, sẽ có một menu hiện ra. Trong menu sẽ có các chức năng chính gồm: nút chơi, nút bật tắt nhạc nền, nút chọn độ khó, nút xem bảng xếp hạng, nút xem thông tin trò chơi. Trong đó:

- Nút chơi: sẽ đưa người chơi vào một giao diện chính, trong đó giao diện sẽ cho phép người chơi điều khiển các khối gạch rơi xuống và có báo cho người chơi biết khối gạch nào tiếp theo sẽ rơi xuống. Trò chơi cũng có hiển thị điểm để người chơi theo dõi thành tích của mình. Ngoài ra còn có một nút dừng để người chơi có thể tạm dừng trò chơi trong bao lâu tùy thích. Trong nút tạm dừng sẽ cho phép người chơi chơi tiếp, bật tắt âm thanh game, thoát ra màn hình chính.

- Nút bật tắt nhạc nền: ban đầu sẽ mặc định nhạc nền được bật trong suốt quá trình. Như chức năng của nó, sẽ cho phép người chơi bật tắt nhạc nền theo ý thích.

- Nút chọn độ khó: có 3 độ khó lần lượt là dễ, trung bình, khó. Mặc định ban đầu sẽ là dễ, người chơi có thể chọn độ khó trung bình hoặc khó tùy theo ý thích. Cứ 100 điểm tốc độ rơi sẽ tăng dần, nhưng sẽ khó phát hiện ra cho đến khi đến một mức nào đó thì người chơi sẽ nhận thấy tốc độ rơi sẽ tăng dần theo thời gian.

- Nút xem bảng xếp hạng: sẽ cho phép người chơi xem thành tích từ cao xuống thấp, cụ thể là top 5 thành tích cao nhất mà người chơi đạt được.

- Nút xem thông tin trò chơi: sẽ cho phép người chơi xem thông tin người phát triển trò chơi.

3. Link github:

https://github.com/truongjr/tetris_game

Trong link github trên em cũng đã kèm thêm báo cáo ở hai dạng là docx và pdf.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

1. Tổng quan về thư viện pygame

Pygame là một bộ mô-đun Python đa nền tảng được thiết kế để viết trò chơi điện tử. Nó bao gồm đồ họa máy tính và thư viện âm thanh được thiết kế để sử dụng với ngôn ngữ lập trình Python.

1.1. Lịch sử

Pygame ban đầu được viết bởi Pete Shinnars để thay thế PySDL sau khi quá trình phát triển của nó bị đình trệ. Đây là một dự án cộng đồng từ năm 2000 và được phát hành theo phần mềm miễn phí mã nguồn mở GNU Lesser General Public License.

Pygame phiên bản 2 được lên kế hoạch là "Pygame Reloaded" vào năm 2009, nhưng việc phát triển và bảo trì pygame đã hoàn toàn dừng lại cho đến cuối năm 2016 với phiên bản 1.9.1. Sau khi phát hành phiên bản 1.9.5 vào tháng 3 năm 2019, việc phát triển phiên bản 2 mới đang hoạt động theo lộ trình.

Pygame 2.0 được phát hành vào ngày 28 tháng 10 năm 2020, nhân ngày sinh nhật thứ 20 của pygame.

1.2. Kiến trúc và tính năng tổng quan

Pygame sử dụng thư viện Simple DirectMedia Layer (SDL), với mục đích cho phép phát triển trò chơi máy tính trong thời gian thực mà không cần cơ chế bậc thấp của ngôn ngữ lập trình C và các dẫn xuất của nó. Điều này dựa trên giả định rằng các chức năng đắt tiền nhất bên trong trò chơi có thể được trừu tượng hóa khỏi logic trò chơi, do đó có thể sử dụng ngôn ngữ lập trình bậc cao, chẳng hạn như Python, để cấu trúc trò chơi.

Các tính năng khác mà SDL không có bao gồm toán học vectơ, phát hiện va chạm, quản lý đồ họa 2d, hỗ trợ MIDI, camera, thao tác mảng pixel, chuyển đổi, lọc, hỗ trợ phông chữ freetype nâng cao và vẽ.

Các ứng dụng sử dụng pygame có thể chạy trên điện thoại và máy tính bảng Android với việc sử dụng Bộ phụ pygame cho Android (pgs4a). Âm thanh, rung, bàn phím và gia tốc kế được hỗ trợ trên Android.

2. Cài đặt và các tính năng được sử dụng trong pygame

2.1. Cài đặt pygame

Ta có thể theo các bước như trang web này hướng dẫn để cài đặt pygame:

- Hệ điều hành Windows: <https://www.geeksforgeeks.org/how-to-install-pygame-in-windows/>
- Hệ điều hành Linux: <https://www.geeksforgeeks.org/install-pygame-in-linux/>

- Hệ điều hành MacOS: <https://www.geeksforgeeks.org/install-pygame-in-macos/>

2.2. Các tính năng chính được sử dụng trong pygame

2.2.1. Tạo cửa sổ game

Để có thể thấy được những gì diễn ra bên trong game thì ta phải làm cho nó hiện ra cửa sổ lên màn hình. Sau đây là một đoạn code để thực hiện việc đó:

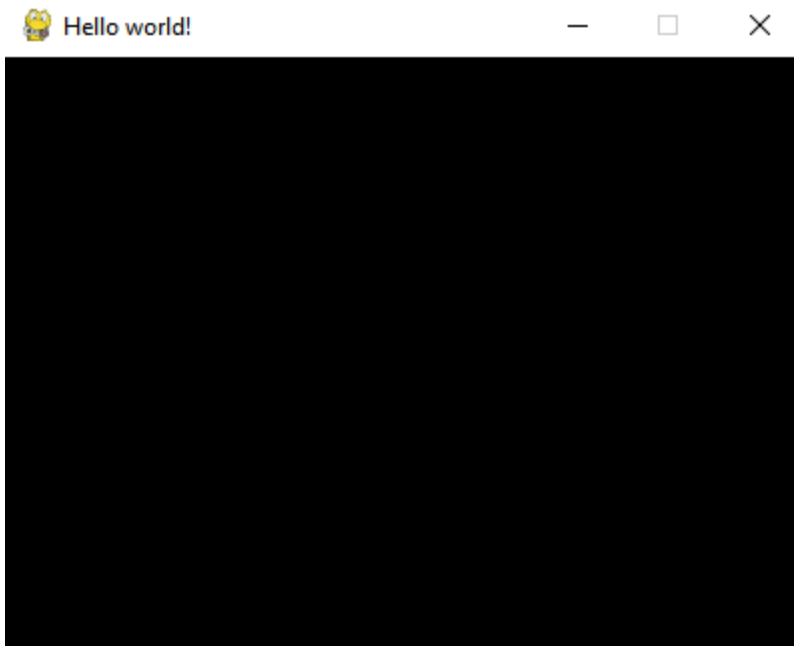
```
import pygame

pygame.init()

win = pygame.display.set_mode((400, 300))

pygame.display.set_caption("Hello world!")
```

Đoạn code trên để tạo ra một cửa sổ có chiều rộng 400 pixel và chiều dài 300 pixel được lưu trong một biến “win”. Ngoài ra ta cũng thay đổi caption của cửa sổ thành dòng chữ “Hello World”.



Đây là kết quả khi chạy chương trình, vì không có vẽ gì lên màn hình nên ta chỉ có thể nhìn thấy một màu đen.

2.2.2. Vòng lặp game và sự kiện trong game

Các chuyển động trong game cũng giống như trong phim hoạt hình, tức là để tạo ra các chuyển động trong game thì người ta sẽ phát liên tục các hình ảnh. Trong game cũng

tương tự thế, vì vậy ta cần có một vòng lặp để “vẽ” liên tục các hình ảnh. Vòng lặp while trong đoạn code dưới đây là để làm việc đó.

```
run = True
while run:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            run = False

pygame.quit()
```

Như có thể thấy ở đoạn code trên, ta cũng thấy có sự xuất hiện của event (sự kiện), đây cũng là một trong các phần quan trọng khi ta viết game. Sự kiện trong game có thể là click chuột, ấn phím,... Đoạn code đó có tác dụng: Khi click vào nút X trên cửa sổ thì kết thúc game và đóng cửa sổ lại. Dùng biến event trong vòng for để lấy các sự kiện xảy ra. Dòng if để kiểm tra sự kiện có phải là “click nút X” hay không. Và dòng run = False để khiến cho vòng lặp dừng lại và nhảy xuống dòng pygame.quit() để đóng cửa sổ hiện ra mà không gây ra lỗi.

Tóm lại, ta có thể hiểu rằng: trong vòng lặp game có 3 việc chính là vẽ, bắt sự kiện, thay đổi đối tượng.

2.2.3. Vẽ vài thứ lên màn hình

Như đã tìm hiểu về vòng lặp game, những lệnh vẽ sẽ nằm trong vòng lặp game. Chúng ta hãy thử tô lên cửa sổ một cái nền màu trắng và vẽ một hình chữ nhật màu đỏ.

```
import pygame


pygame.init()
win = pygame.display.set_mode((400, 300))
pygame.display.set_caption("Hello world!")

run = True
while run:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            run = False
```

```
win.fill((255, 255, 255))
pygame.draw.rect(win, (255, 0, 0), (100, 80, 150, 50))
pygame.display.update()

pygame.quit()
```

Và đây là kết quả khi chạy chương trình:

 Hello world!

— □ ×



```
win.fill((255, 255, 255))
```

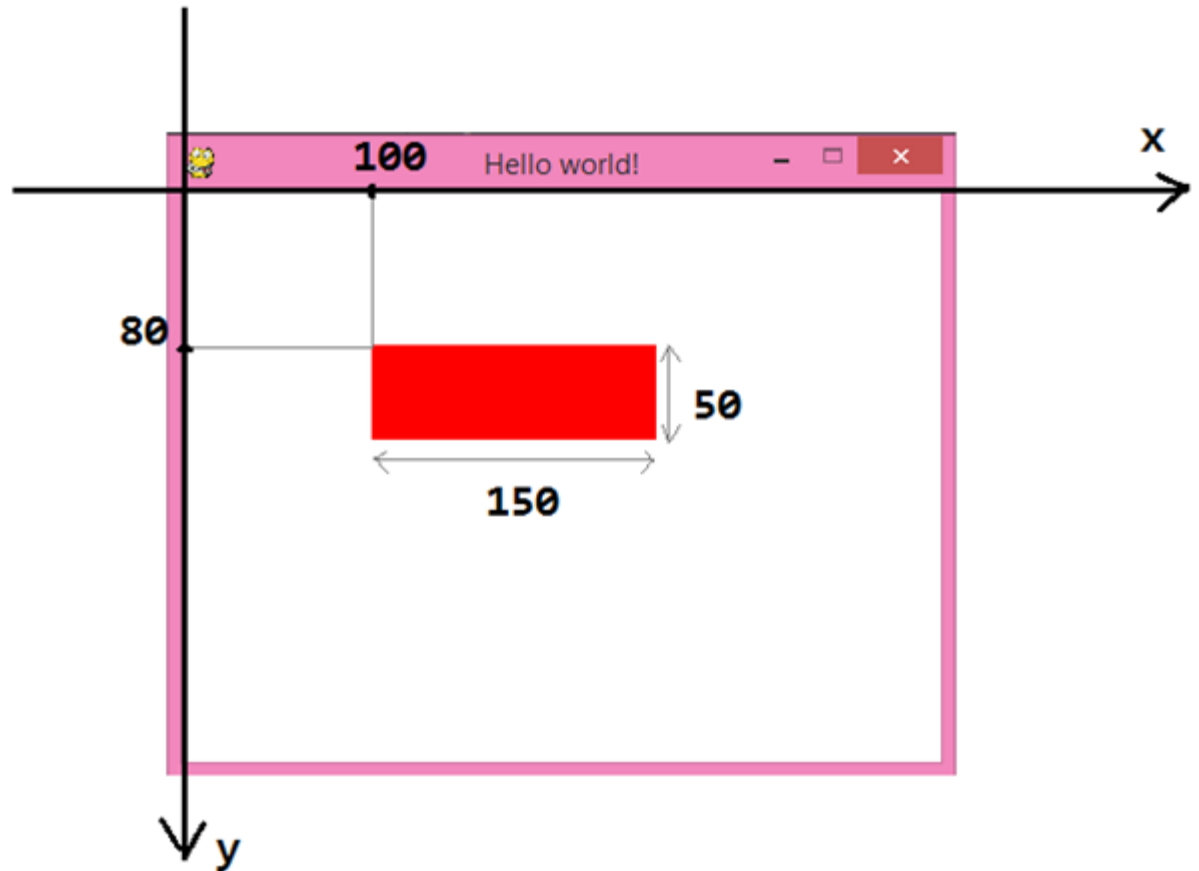
Dòng trên dùng để tô màu cho cửa sổ game. Hàm fill có tham số là một **tuple** (hoặc list) thể hiện cho màu sắc. (255, 255, 255) thể hiện màu trắng trong hệ màu RGB.

```
pygame.draw.rect(win, (255, 0, 0), (100, 80, 150, 50))
```

Dòng này dùng để vẽ hình chữ nhật. Trong hàm rect có các tham số cơ bản như sau:

pygame.draw.rect(surface, color, rect, width)

- **surface** là “chỗ” vẽ vào, cụ thể ở đây là DISPLAYSURF (tức là cái cửa sổ đấy)
- **color** là một tuple (hoặc list) thể hiện màu sắc, cụ thể ở đây là màu đỏ
- **rect** là một tuple (hoặc list) có 4 phần tử thể hiện cho các thông số của hình chữ nhật: hoành độ góc trên bên trái, tung độ góc trên bên trái, chiều dài, chiều cao. Gốc tọa độ là góc trên cùng bên trái, trục hoành nằm ngang hướng sang phải, trục tung thẳng đứng hướng xuống dưới, đơn vị tọa độ và độ dài là pixel.



- **width** là độ dày của nét vẽ, nếu không truyền vào tham số này thì mặc định là hình chữ nhật sẽ được tô kín.

2.2.4. Một số hàm vẽ thường gặp

Thực tế, trong các game người ta ít dùng các hàm để vẽ những hình ảnh, mà người ta sẽ dùng những file ảnh để thêm vào trong game. Ta sẽ dùng các hàm vẽ này trong lúc test thử hoạt động của game trước khi đặt những hình ảnh vào.

```
import pygame

pygame.init()
win = pygame.display.set_mode((400, 300))
pygame.display.set_caption("Hello world!")
# Tạo sẵn các màu sắc
BLACK = ( 0, 0, 0)
WHITE = (255, 255, 255)
```

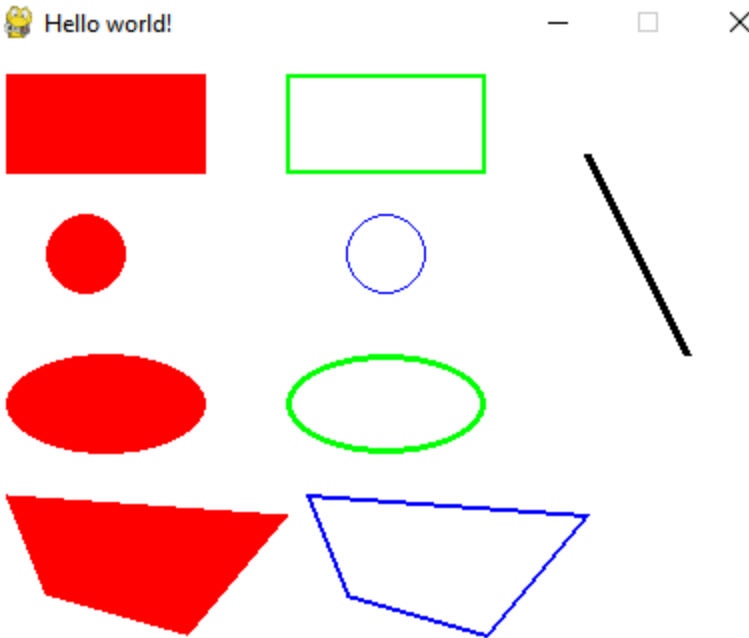
```

RED    = (255,  0,  0)
GREEN  = (  0, 255,  0)
BLUE   = (  0,  0, 255)
run = True
while run:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            run = False
    win.fill((255, 255, 255))
    pygame.draw.rect(win, RED, (10, 10, 100, 50))# Hình chữ nhật
    pygame.draw.rect(win, GREEN, (150, 10, 100, 50), 2)# Hình chữ nhật
rỗng
    pygame.draw.circle(win, RED, (50, 100), 20) # Hình tròn
    pygame.draw.circle(win, BLUE, (200, 100), 20, 1)# Hình tròn rỗng
    pygame.draw.ellipse(win, RED, (10, 150, 100, 50))# Hình elip
    pygame.draw.ellipse(win, GREEN, (150, 150, 100, 50), 3)# Hình elip
rỗng
    pygame.draw.polygon(win, RED, ((10, 220), (150, 230), (100 ,290),
(30, 270)))# Đa giác
    pygame.draw.polygon(win, BLUE, ((160, 220), (300, 230), (250
,290), (180, 270)), 2)# Đa giác rỗng
    pygame.draw.line(win, BLACK, (300, 50), (350, 150), 4)# Đoạn thẳng
    pygame.display.update()

pygame.quit()

```

Đây là kết quả sau khi chạy chương trình:



Sau đây là một số hàm vẽ khác:

pygame.draw.circle(surface, color, center, radius, width) Hàm này dùng để vẽ hình tròn.

- **surface** là nơi để vẽ lên.
- **color** là màu được vẽ.
- **center** là một tuple (hoặc list) thể hiện tọa độ tâm hình tròn.
- **radius** là bán kính hình tròn.
- **width** là độ dày nét vẽ (tương tự hình chữ nhật).

pygame.draw.ellipse(surface, color, rect, width) Hàm này dùng để vẽ hình elip.

- **surface** là nơi để vẽ lên.
- **color** là màu để vẽ.
- **rect** là một tuple (hoặc list) gồm 4 phần tử là các thông số hình chữ nhật (như đã tìm hiểu ở phần trước). Hình elip được vẽ nội tiếp hình chữ nhật đó.
- **width** là độ dày nét vẽ (tương tự hình chữ nhật).

pygame.draw.polygon(surface, color, points, width) Hàm này dùng để vẽ đa giác.

- **surface** là nơi để vẽ lên.
- **color** là màu được vẽ.
- **points** là tuple (hoặc list) thể hiện các đỉnh của đa giác. Mỗi đỉnh là một tuple (hoặc list) thể hiện tọa độ.
- **width** là độ dày nét vẽ (tương tự hình chữ nhật).

pygame.draw.line(surface, color, start_pos, end_pos, width) Hàm này dùng để vẽ một đoạn thẳng.

- **surface** là nơi để vẽ lên.
- **color** là màu được vẽ.
- **start_pos** là tuple (hoặc list) thể hiện toạ độ điểm đầu của đoạn thẳng.
- **end_pos** là tuple (hoặc list) thể hiện toạ độ điểm cuối của đoạn thẳng.
- **width** là độ dày nét vẽ.

Các giá trị toạ độ hay độ dài là những số nguyên, đơn vị là pixel.

2.2.5. Tạo chuyển động trong game

Giả sử ta có một đoạn code vẽ một hình chữ nhật để cho nó chạy từ trái qua phải:

```
import pygame

WINDOWWIDTH = 400 # Chiều dài cửa sổ
WINDOWHEIGHT = 300 # Chiều cao cửa sổ
# Tạo sẵn các màu sắc
WHITE = (255, 255, 255)
RED = (255, 0, 0)
pygame.init()

win = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
pygame.display.set_caption("Hello world!")
# Xác định FPS
FPS = 60
fpsClock = pygame.time.Clock()
# Hoành độ của hình chữ nhật
rect_x = 0
run = True
while run:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            run = False
    win.fill(WHITE)
```

```

pygame.draw.rect(win, RED, (rect_x, 100, 50, 40))
rect_x += 2
if rect_x + 50 > WINDOWWIDTH:
    rect_x = WINDOWWIDTH - 50
pygame.display.update()
t = fpsClock.tick(FPS)

pygame.quit()

```

Kết quả là ta sẽ được một hình chữ nhật màu đỏ ở giữa màn hình đi từ mép trái sang mép phải.

Trước khi tìm hiểu những dòng code thì các bạn cần biết thêm 1 khái niệm là **FPS (Frames Per Second)**. Hiểu theo nghĩa tiếng Việt thì **FPS** là số lượng khung hình trong 1 giây. Ví dụ: nếu FPS là 60 thì trong 1 giây sẽ có 60 khung hình xuất hiện liên tiếp. Có thể ai cũng biết vòng lặp while chạy rất nhanh, trong 1 giây sẽ lặp rất nhiều lần và tốc độ phụ thuộc vào nhiều yếu tố như code trong vòng lặp, CPU... Vì thế để đảm bảo game hoạt động ổn định thì cần phải thiết lập FPS để vòng lặp game thực hiện với tốc độ nhất định. Để thiết lập FPS thì các bạn hãy chú ý những dòng code sau:

- Đặt 2 dòng này bên ngoài vòng lặp game, trường hợp này thì FPS là 60:

```

FPS = 60
fpsClock = pygame.time.Clock()

```

- Đặt dòng này ở cuối vòng lặp game:

```

fpsClock.tick(FPS)

```

Trong đó fpsClock = pygame.time.Clock() sẽ tạo một đối tượng để quản lý thời gian.

Dòng fpsClock.tick(FPS) được gọi một lần cho mỗi khung hình. Nó sẽ tính toán bao nhiêu mili giây đã trôi qua kể từ lần gọi trước. Nếu mà vượt qua đối số tốc độ khung hình tức là FPS thì hàm sẽ delay để giữ cho trò chơi chạy chậm hơn số tick mỗi giây. Điều này có thể được sử dụng để giúp giới hạn tốc độ thời gian chạy của trò chơi. Ví dụ: bằng cách gọi Clock.tick(40) một lần cho mỗi khung hình, chương trình sẽ không bao giờ chạy với tốc độ hơn 40 khung hình mỗi giây.

Trong vòng lặp game có dòng code như sau:

```

pygame.draw.rect(win, RED, (rect_x, 100, 50, 40))

```

Dòng code này sẽ vẽ hình chữ nhật màu đỏ với hoành độ rect_x và tung độ là 100, với kích thước là 50 x 40

```

# Thay đổi vị trí hình chữ nhật

```

```
rect_x += 2
if rect_x + 50 > WINDOWWIDTH:
    rect_x = WINDOWWIDTH - 50
```

Sau mỗi vòng lặp thì hoành độ của hình chữ nhật tăng lên 2 pixel. Điều kiện trong if để chiếc xe không chạy ra ngoài màn hình, khi phần đầu của hình chữ nhật ra khỏi màn hình thì vẽ chiếc xe sát lại lề bên phải.

2.2.6. *Vẽ chữ*

Đoạn code sau đây sẽ vẽ chữ ra màn hình:

```
import pygame

WINDOWWIDTH = 400 # Chiều dài cửa sổ
WINDOWHEIGHT = 300 # Chiều cao cửa sổ

WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
pygame.init()

win = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
pygame.display.set_caption("Hello world!")

FPS = 60
fpsClock = pygame.time.Clock()

font = pygame.font.SysFont('consolas', 30)
text = font.render('Hello World', True, GREEN, RED)
run = True
while run:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            run = False
    win.fill(WHITE)
```



```
win.blit(text, (50, 100))
pygame.display.update()
fpsClock.tick(FPS)
```

```
pygame.quit()
```

Trước tiên cần phải xác định font chữ cần vẽ. Dòng dưới đây để tạo font chữ:

```
font = pygame.font.SysFont('consolas', 30)
```

Biến font dùng đại diện cho font chữ đó, font được dùng là consolas, size chữ là 30.

Tiếp theo là một surface và vẽ chữ lên đó. Dòng dưới này có nhiệm vụ đó:

```
text = font.render('Hello World', True, GREEN, RED)
```

Tham số 'Hello world' là chữ cần vẽ. Tham số thứ 2 là True, cái này gọi là antialias, nó dùng để làm "mềm" các nét vẽ. Tham số thứ 3 là màu chữ, ở đây là màu xanh. Tham số cuối là màu nền, ở đây là màu đỏ. Nếu không truyền vào tham số cuối thì chữ sẽ không có màu nền.

2.2.7. Thêm hình ảnh

Trước giờ ta chỉ những surface rồi vẽ hình lên đó. Tuy nhiên, có những hình phức tạp (như một con mario) thì không thể dùng các hàm để vẽ được, hoặc nếu vẽ được thì rất là phức tạp. Vì thế, người ta phải dùng những file hình ảnh có sẵn để thêm vào trong game. Nào, bây giờ hãy tìm hiểu cách để thêm hình ảnh vào. Để cho đơn giản thì mình sẽ dùng phần code chiếc xe ở phần trước. Mình sẽ thêm hình ảnh chiếc xe vào để thay thế cho các hàm vẽ.

Trước hết chúng ta sẽ tìm hình 1 chiếc xe trước, và sẽ lưu nó cùng chung thư mục với file chạy chương trình và tên file hình này là car.png:



Và ta chỉnh sửa một chút code như sau:

```
import pygame

WINDOWWIDTH = 400 # Chiều dài cửa sổ
WINDOWHEIGHT = 300 # Chiều cao cửa sổ

WHITE = (255, 255, 255)
RED = (255, 0, 0)
```

```

GREEN = ( 0, 255, 0)
pygame.init()

win = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
pygame.display.set_caption("Hello world!")

FPS = 60
fpsClock = pygame.time.Clock()

image = pygame.image.load('car.png')
car_x = 0

run = True
while run:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            run = False
    win.fill(WHITE)
    win.blit(image, (car_x, 100))
    car_x += 2
    if car_x + 100 > WINDOWWIDTH:
        car_x = WINDOWWIDTH - 100
    pygame.display.update()
    fpsClock.tick(FPS)

pygame.quit()

```

Ở đây có dòng code:

```

image = pygame.image.load('car.png')
car_x = 0

```

Nó sẽ load file tên car.png và lưu vào một surface tên là image. Ngoài ra ta còn lưu hoành độ của chiếc xe bằng biến car_x.

Để có thể đưa hình lên trên màn hình ta sẽ dùng đoạn code sau:

```
win.blit(image, (car_x, 100))
```

Đoạn code này có tác dụng như đã nêu và bố trí hình ảnh ở tọa độ (car_x, 100) để cho nó chuyển động từ mép trái sang mép phải của màn hình. Và đoạn code if để xử lý di chuyển được giữ nguyên như lúc trước, nhưng chỉ thay đổi từ 50 thành 100 để cho phù hợp với chiều dài của hình car.png.

2.2.8. Sự kiện

Một game cần phải có sự tương tác với người chơi. Tất nhiên người chơi game thích điều khiển một chiếc xe hơn là xem nó tự chạy. Vì thế, ta hãy có ý tưởng là thế này, khi nhấn giữ phím mũi tên trái thì chiếc xe chạy sang trái, khi ấn giữ phím mũi tên phải thì chiếc xe chạy sang phải.

Đoạn code sau đây sẽ thực hiện điều đó:

```
import pygame
from pygame.constants import K_LEFT, K_RIGHT, KEYDOWN, KEYUP

WINDOWWIDTH = 400 # Chiều dài cửa sổ
WINDOWHEIGHT = 300 # Chiều cao cửa sổ

WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
pygame.init()

win = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
pygame.display.set_caption("Hello world!")

FPS = 60
fpsClock = pygame.time.Clock()

image = pygame.image.load('car.png')
run = True
car_x = 0
moveLeft = False
moveRight = False
```

```

while run:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            run = False

        if e.type == KEYDOWN:
            if e.key == K_LEFT:
                moveLeft = True
            if e.key == K_RIGHT:
                moveRight = True
        if e.type == KEYUP:
            if e.key == K_LEFT:
                moveLeft = False
            if e.key == K_RIGHT:
                moveRight = False

    win.fill(WHITE)
    win.blit(image, (car_x, 100))

    if moveLeft == True:
        car_x -= 2
    if moveRight == True:
        car_x += 2

    if car_x + 100 > WINDOWWIDTH:
        car_x = WINDOWWIDTH - 100
    if car_x < 0:
        car_x = 0
    pygame.display.update()
    fpsClock.tick(FPS)

pygame.quit()

```

Khi chạy chương trình, nếu ta ấn giữ phím mũi tên trái thì xe chạy sang trái, khi ấn giữ phím mũi tên phải thì xe chạy sang phải.

Bây giờ ta xét hai tham số được thêm vào là `moveLeft` và `moveRight` dạng boolean:

```
moveLeft = False
moveRight = False
```

Chúng được dùng để kiểm tra xem xe đang chạy sang trái, sang phải hay đứng im. Ở đây, ta sẽ mặc định là cả hai tham số đều `False`, nghĩa là chiếc xe đang đứng im tại chỗ. Tiếp theo, ta xem những dòng code sau:

```
if moveLeft == True:
    car_x -= 2
if moveRight == True:
    car_x += 2

if car_x + 100 > WINDOWWIDTH:
    car_x = WINDOWWIDTH - 100
if car_x < 0:
    car_x = 0
```

Những dòng code này có nghĩa là nếu đi sang trái hoặc sang phải thì hoành độ của xe sẽ thay đổi. Nếu đụng mép trái hoặc mép phải thì ta sẽ thay đổi hoành độ để cho chiếc xe luôn nằm trong màn hình mà không rời khỏi màn hình.

Chú ý đến phần bắt sự kiện bên trong vòng lặp `for`:

```
if e.type == KEYDOWN:
    if e.key == K_LEFT:
        moveLeft = True
    if e.key == K_RIGHT:
        moveRight = True

if e.type == KEYUP:
    if e.key == K_LEFT:
        moveLeft = False
    if e.key == K_RIGHT:
        moveRight = False
```

Dòng `if e.type == KEYDOWN`: để kiểm tra sự kiện `KEYDOWN` có xảy ra hay không. `KEYDOWN` là sự kiện xảy ra khi có 1 phím được ấn xuống. Lưu ý là sự kiện này chỉ xảy ra vào khoảng khắc phím được ấn xuống, khi phím đang giữ thì sự kiện này không xảy ra nữa (trừ khi có phím khác được ấn). Trong đoạn if đó thì có 2 dòng if nữa để kiểm tra phím được ấn là phím nào, dòng `if e.key == K_LEFT`: để xem phím được ấn có phải là phím mũi tên trái không, nếu đúng thì gán `moveLeft = True` để xe chạy sang trái, tương tự với trường hợp nhấn phím mũi tên phải.

Dòng `if e.type == KEYUP`: để kiểm tra sự kiện `KEYUP` có xảy ra hay không. `KEYUP` là sự kiện xảy ra khi có 1 phím được thả ra. Cũng như `KEYDOWN`, sự kiện `KEYUP` chỉ xảy ra vào khoảng khắc phím được thả ra. Các đoạn if bên trong cũng tương tự như phần trên. Khi mũi tên trái được thả ra thì `moveLeft = False` (xe không chạy sang trái nữa), tương tự với trường hợp phím mũi tên bên phải.

2.2.9. Thêm nhạc và âm thanh vào

Làm game thì không thể thiếu việc có nhạc nền và âm thanh game, ta sẽ tải âm thanh game từ các trang này:

<https://downloads.khinsider.com/game-soundtracks/album/tetris-gameboy-rip>: ở đây ta sẽ tải file `tetris-gameboy-02`

<https://www.findsounds.com/ISAPI/search.dll?keywords=tetris>: ở đây ta sẽ tải `tetris-fall`

Sau khi đã tải xong thì ta sẽ bỏ vào cùng 1 thư mục với file game source code. Và thêm một số dòng code sau đây:

```
import pygame
from pygame.constants import K_LEFT, K_RIGHT, KEYDOWN, KEYUP

WINDOWWIDTH = 400 # Chiều dài cửa sổ
WINDOWHEIGHT = 300 # Chiều cao cửa sổ

WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
pygame.init()

win = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
pygame.display.set_caption("Hello world!")
```

```

FPS = 60
fpsClock = pygame.time.Clock()

image = pygame.image.load('car.png')
car_x = 0
run = True
moveLeft = False
moveRight = False
pygame.mixer.init()
pygame.mixer.music.load("tetris-gameboy-02.mp3")
pygame.mixer.music.set_volume(0.5)
pygame.mixer.music.play(-1, 0.0)
while run:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            run = False

        if e.type == KEYDOWN:
            if e.key == K_LEFT:
                moveLeft = True
            if e.key == K_RIGHT:
                moveRight = True
        if e.type == KEYUP:
            if e.key == K_LEFT:
                moveLeft = False
            if e.key == K_RIGHT:
                moveRight = False

    win.fill(WHITE)
    win.blit(image, (car_x, 100))

```

```

    if moveLeft == True:
        car_x -= 2
    if moveRight == True:
        car_x += 2

    if car_x + 100 > WINDOWWIDTH:
        car_x = WINDOWWIDTH - 100
        hit = pygame.mixer.Sound("fall.wav")
        hit.play()
    if car_x < 0:
        car_x = 0
        hit = pygame.mixer.Sound("fall.wav")
        hit.play()
    pygame.display.update()
    fpsClock.tick(FPS)

pygame.mixer.music.stop()
pygame.quit()

```

Ở đây ta có thêm các dòng sau, trước khi vào vòng lặp while:

```

pygame.mixer.init()
pygame.mixer.music.load("tetris-gameboy-02.mp3")
pygame.mixer.music.set_volume(0.5)
pygame.mixer.music.play(-1, 0.0)

```

trong đó `pygame.mixer.init()` để khởi tạo mô-đun mixer.

`pygame.mixer.music.load("tetris-gameboy-02.mp3")` sẽ tải tệp tên là “tetris-gameboy-02.mp3” và chuẩn bị để phát.

`pygame.mixer.music.set_volume(0.5)` lúc này sẽ chỉnh âm lượng của nhạc về một nửa, và số truyền vào sẽ là số thực trong khoảng từ 0.0 đến 1.0.

`pygame.mixer.music.play(-1, 0.0)` sẽ cho phép nhạc bắt đầu từ 0.0 và sẽ lặp vô hạn. Trong hàm này có đến 3 tham số như sau: `play(loops = 0, start = 0.0, fade_ms = 0)` → None

- `loops` có nghĩa là số lần mà nhạc sẽ lặp lại, mặc định là 0. Nhạc lặp vô hạn nếu đối số này được đặt thành -1.

- start có nghĩa là vị trí thời gian mà nhạc bắt đầu phát. Vị trí bắt đầu phụ thuộc vào định dạng của nhạc được phát. MP3 và OGG sử dụng vị trí theo thời gian tính bằng giây.

- fade_ms có nghĩa là khoảng thời gian (tính bằng mili giây) mà nhạc sẽ giảm tăng dần từ 0.0 lên âm lượng tối đa (hoặc âm lượng đã được set trước đó bởi set_volume()).

```
hit = pygame.mixer.Sound("fall.wav")  
hit.play()
```

Dòng đầu tiên có nghĩa là tạo một đối tượng Sound từ file tên fall.wav để có thể lồng nhạc vào âm thanh nền mà không phải dừng âm thanh nền. Dòng tiếp theo để chạy âm thanh đó. Ta bỏ vào điều kiện kiểm tra mép trái và phải để biểu hiện cho sự va chạm vào mép của xe, lúc đó thì ta sẽ cho âm thanh bật lên.

CHƯƠNG 3: THỰC NGHIỆM

1. Màn hình chính

Sau khi chạy chương trình, thì ta sẽ nhận được màn hình như sau:



Trong màn hình chính trên, dòng chữ TETRIS được làm chuyển động lên và xuống để cho sinh động hơn. Khi ta rê chuột vào các nút thì chúng sẽ có hiệu ứng Hover, hiệu ứng này sẽ làm cho nút nhỏ hơn bình thường. Các nút hiện trên màn hình từ trên xuống dưới từ trái qua phải gồm:



Đây là nút chơi, nút này sẽ vào giao diện chính của trò chơi.



Đây là nút bật tắt nhạc nền, mặc định sẽ là bật nhạc nền. Báo cho người chơi biết rằng nhạc nền đang được bật.



Đây là nút bật tắt nhạc nền khi ta bấm vào, hình sẽ được cập nhật thành có dấu chéo để báo cho người chơi biết rằng nhạc nền đã được tắt.

định ban đầu sẽ là dễ, khi mũi tên ở trên chữ LV. này và chọn độ khó thì ở hiện ra 3 mức độ cho người



Đây là nút chọn độ khó, mặc người chơi chọn độ khó khác nhau thì sẽ khác nhau. Khi ấn vào nút mũi tên ngay bên dưới bên phải của nút này sẽ chơi chọn, khi chọn màu nào thì mũi tên sẽ lập tức biến thành màu đó.



Đây là trạng thái khi ta chọn độ khó cho việc bắt đầu trò chơi, khi chọn độ khó nào thì màu của mũi tên sẽ theo màu của các mức độ mà nó đang mang.



Dễ(Easy)

Trung bình(Medium)

Khó(Hard)



Đây là nút xem bảng xếp hạng, bảng xếp hạng sẽ lấy top 5 điểm cao nhất để đưa chọn phép người chơi xem thành tích của bản thân. Khi ấn vào ta sẽ thấy được bảng xếp hạng như sau:

TETRIS

— □ ×

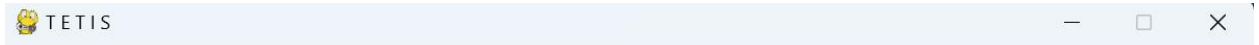


1	30
2	20
3	10
4	0
5	0

Để có thể trở lại màn hình chính thì ta sẽ ấn chuột thêm 1 lần nữa để trở lại màn hình chính.



Đây là nút xem thông tin của trò chơi, khi ấn vào thì sẽ hiện ra thông tin của người phát triển trò chơi này, đây là màn hình khi ta ấn vào nút xem thông tin.



HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
Posts and Telecommunications Institute of Technology

Đồ án :

GAME TETRIS

Giảng viên hướng dẫn : TS. Nguyễn Thị Tuyết Hải

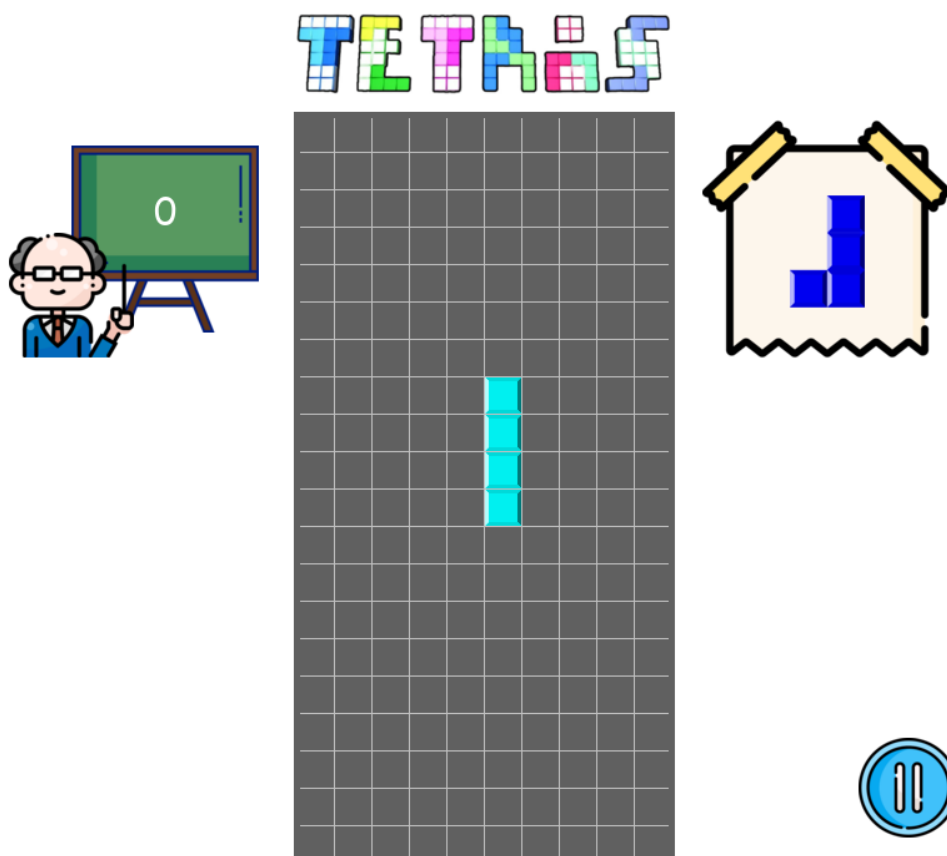
Sinh viên thực hiện : Huỳnh Ngô Anh Thái - N19DCCN188

TP Hồ Chí Minh Ngày 01 Tháng 03 Năm 2023

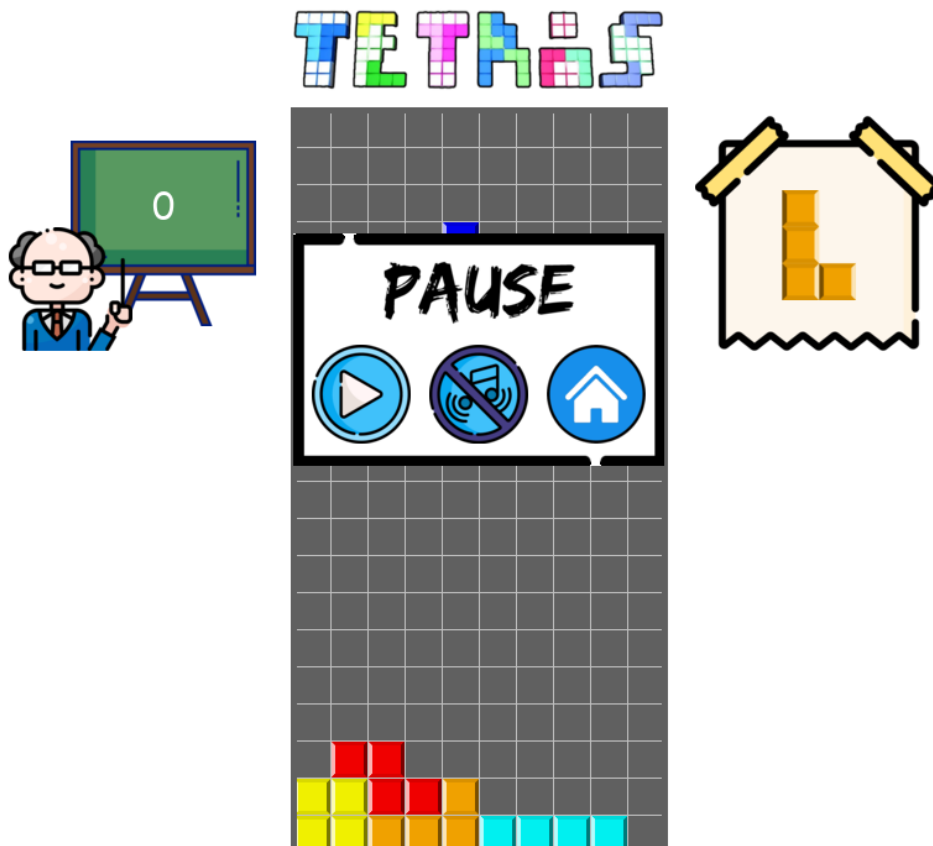
Để có thể trở lại màn hình chính, ta hoàn toàn có thể ấn chuột như lúc xem bảng xếp hạng.

2. Giao diện trò chơi

Khi ấn vào nút chơi, ta sẽ có màn hình như sau:

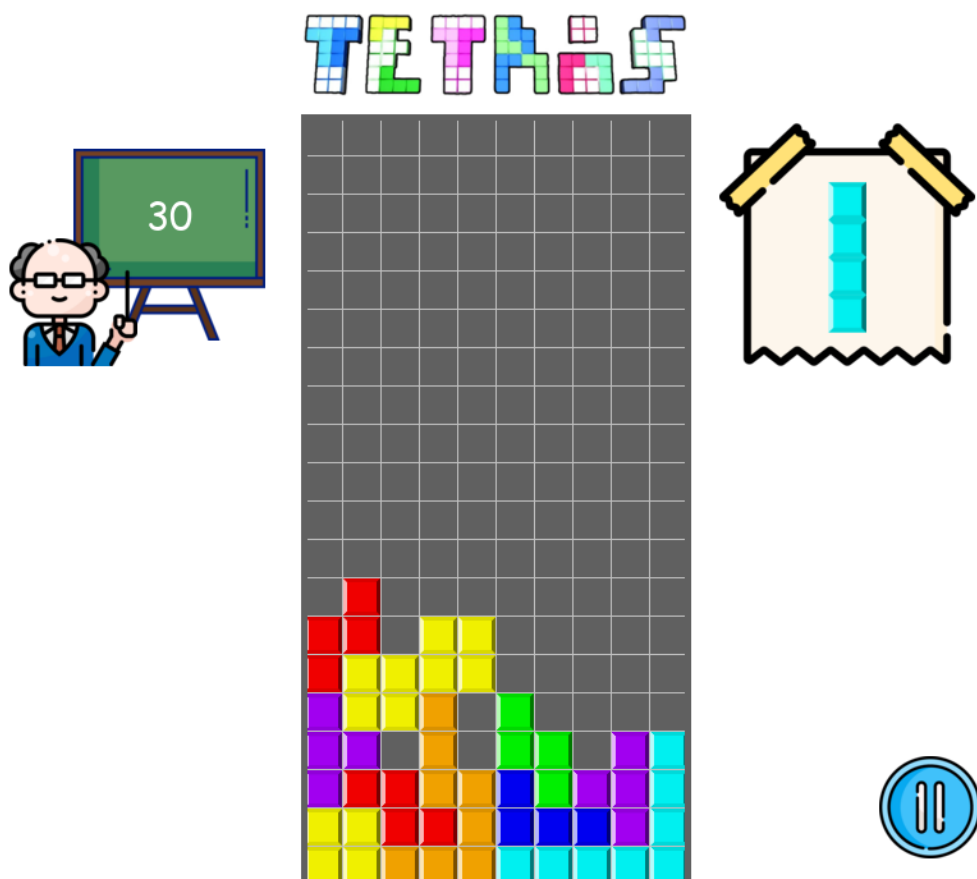


Như ta có thể thấy trên màn hình, ở giữa là khu vực chơi, người chơi sẽ điều khiển khối gạch đang rơi. Ở phía bên trái là điểm của người chơi, mỗi lần được một hàng đầy gạch là sẽ cộng thêm 10 điểm. Bên phải phía trên báo cho người chơi biết rằng gạch nào sẽ rơi xuống tiếp theo để người chơi tính toán chiến thuật. Ở góc dưới bên phải là nút tạm dừng, khi ấn tạm dừng, trò chơi sẽ hiện ra giao diện sau:

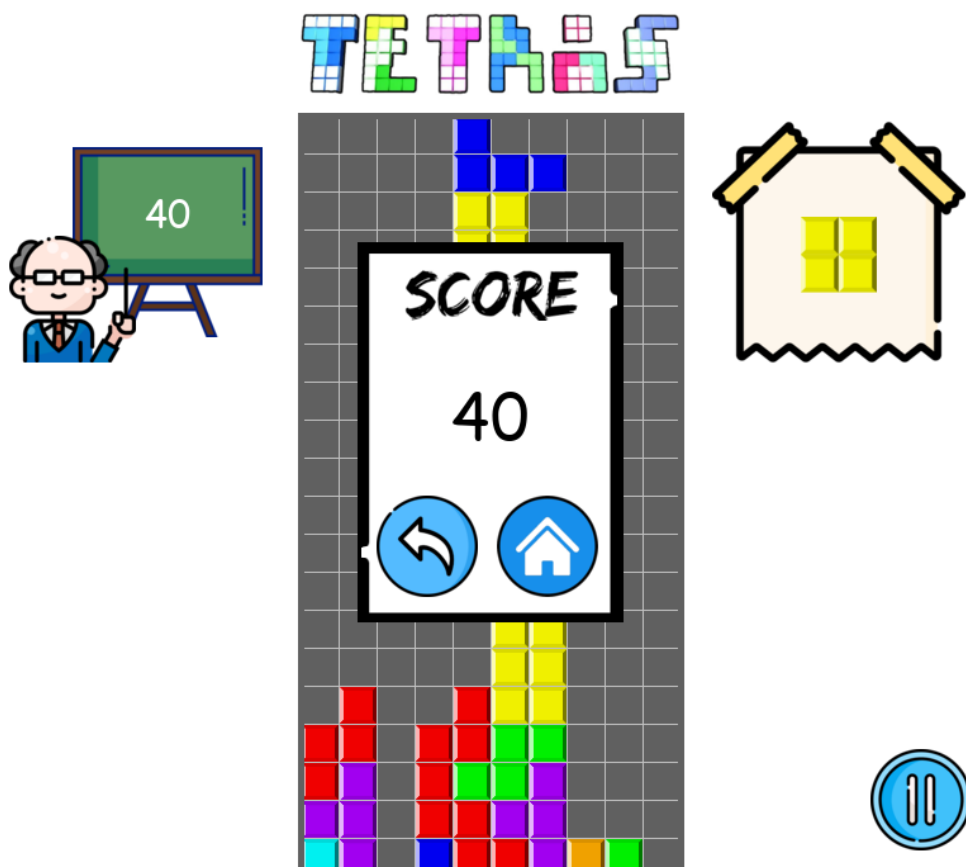


Trong pause này sẽ có 3 nút, nút đầu tiên sẽ cho phép người chơi quay lại màn chơi hiện đang dang dở, nút thứ hai để bật tắt nhạc nền, nút thứ ba để người chơi quay lại màn hình chính.

Khi đủ một hàng chứa đầy các gạch thì ta sẽ thấy hiệu ứng trong trò chơi, nhưng do chức năng chụp màn hình không bắt được khoảnh khắc đó:



Như hình trên, ta có thể thấy có 3 hàng dưới cùng đã đầy gạch, sau khi thực hiện xong hiệu ứng và âm thanh hoàn thành các dòng, ta có thể thấy điểm sẽ được cộng dồn vào. Khi ta chơi bị thua, thì màn hình sẽ hiện ra giao diện như sau:



Ở đây, sau khi tiếng nhạc thua vang lên, ta có thể thấy có hai nút điều khiển, nút đầu tiên bên trái sẽ cho phép người chơi chơi một màn mới. Còn nút hình ngôi nhà bên cạnh sẽ cho ta quay trở lại màn hình chính. Ta chọn quay trở lại màn hình chính, và vào mục bảng xếp hạng, và điểm 40 là điểm cao nhất nên ta có kết quả như sau:



Bảng xếp hạng

1 40

2 30

3 20

4 10

5 0

Đến đây là kết thúc phần demo game!

KẾT LUẬN

Kết quả đạt được của đồ án

Sau một thời gian tập trung triển khai đề tài, em đã hoàn thành được đồ án “Game TETRIS” với giao diện và các thao tác sử dụng dễ dàng, thân thiện với người dùng. Các cam kết tụi em đưa ra cũng đã cơ bản được đáp ứng.

Sau khi hoàn thành xong được đồ án này, em đã được mở rộng thêm các kiến thức chuyên ngành mới nhờ việc tìm hiểu một số kiến thức lập trình để áp dụng vào việc phát triển đồ án này.

Nhờ vậy, kỹ năng tự học và vận dụng các kỹ năng mới của em được nâng cao hơn. Khi thực hiện và hoàn thành dự án, em cũng đã biết thêm được nhiều kinh nghiệm quý giá khi xây dựng ứng dụng nói riêng và phát triển phần mềm. Những kỹ năng này sẽ là nền tảng để giúp em nâng cao trình độ bản thân, kinh nghiệm trong thực tế để có thể làm việc ngoài các doanh nghiệp sau này. Ngoài ra, kỹ năng giao tiếp với bạn đồng hành và phối hợp làm việc nhóm của em cũng được cải thiện rõ rệt, đồng thời cũng tăng tính kỷ luật cho cả hai trong quá trình làm việc nhóm.

Hạn chế của đồ án

Quá trình xử lý logic của chương trình chưa thực sự được tối ưu. Nên đôi lúc cũng gây chậm trễ cho chương trình. Vì vấn đề thời gian nên chức năng giữ nút chưa được xử lý, có thể gây nên sự khó chịu cho người chơi.

TÀI LIỆU THAM KHẢO

1. <https://vi.wikipedia.org/wiki/Tetris>
2. <https://vi.wikipedia.org/wiki/Pygame>
3. <https://www.geeksforgeeks.org/how-to-install-pygame-in-windows/>
4. <https://www.geeksforgeeks.org/install-pygame-in-linux/>
5. <https://www.geeksforgeeks.org/install-pygame-in-macos/>
6. <https://www.pygame.org/docs/>
7. <https://codelearn.io/sharing/lap-trinh-game-co-ban-voi-pygame>
8. <https://codelearn.io/sharing/lap-trinh-game-co-ban-voi-pygame-p2>
9. Tham khảo trong sách making games with python & pygame