

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Khoa Công nghệ thông tin



BỘ MÔN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
BÁO CÁO ĐỒ ÁN
TRÒ CHƠI BĂNG QUA ĐƯỜNG



Giáo viên hướng dẫn : Thầy Trương Toàn Thịnh

Sinh viên thực hiện : Doãn Anh Khoa – 21127076

Nguyễn Hồng Quế Anh – 2112578

Đỗ Hoàng Khánh Duy – 2112601

Trần Hoàng Khải – 21127620

Nhóm : 4 - Lớp: 21CLC03

MỤC LỤC

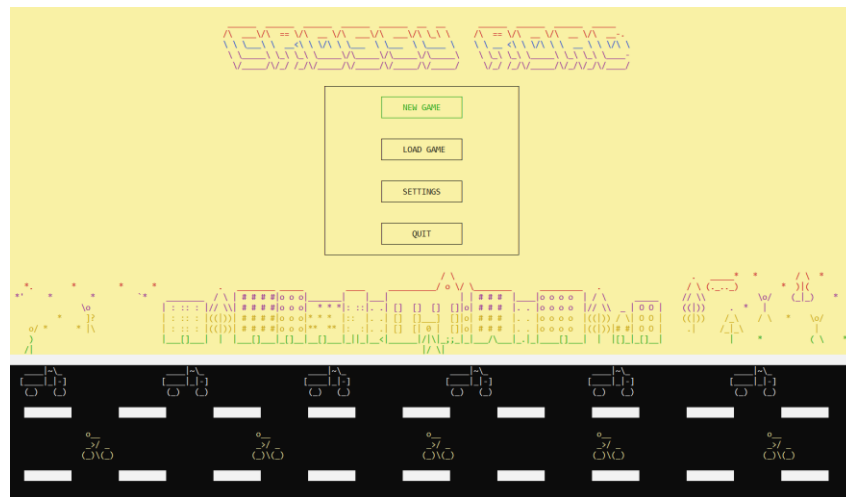
I.	Giới thiệu.....	2
II.	Kịch bản và giao diện của trò chơi.....	2
III.	Các lớp đối tượng được xây dựng trong game.....	6
1.	Lớp CPOSITION:	6
2.	Lớp CPEOPLE:.....	6
3.	Lớp COBSTACLE, CBIRD, CBUTTERFLY, CCAR, CDUCK	6
4.	Lớp CONELANE, CLANE	7
5.	Lớp CLEVEL.....	8
6.	Lớp CMAP	8
7.	Lớp CGAME:.....	8
8.	Sơ đồ lớp đối tượng tham gia trò chơi:	9
IV.	Các kỹ thuật xử lý trò chơi.....	10
V.	Mã giả.....	11
VI.	Nguồn tài liệu tham khảo	13

I. Giới thiệu

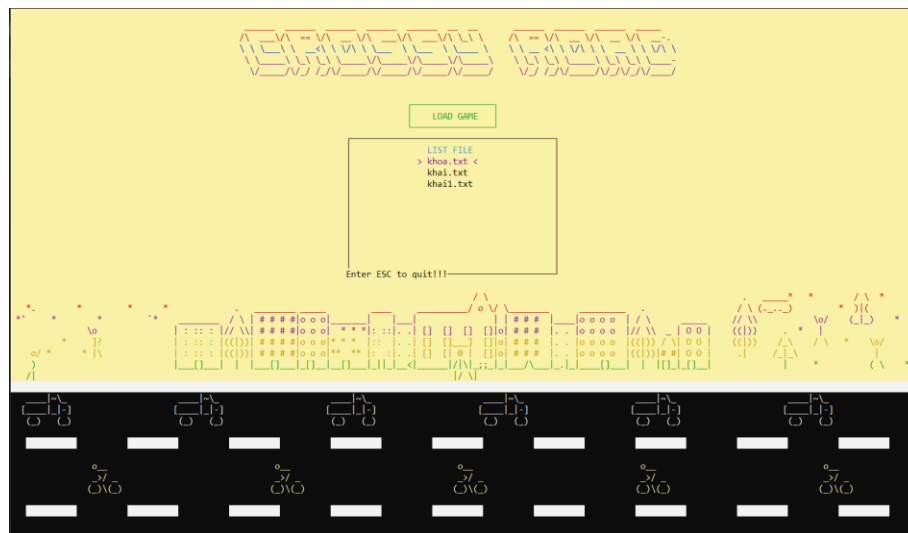
- Chúng em là thành viên của nhóm 4 lớp 21CLC03 đang học môn Lập trình hướng đối tượng. Trong đồ án trò chơi Băng qua đường, chúng em đã vận dụng, phối hợp những kiến thức, kĩ thuật và cấu trúc dữ liệu cơ bản như xử lí tập tin, kĩ thuật đa tiến trình và thêm những kĩ thuật để thiết lập những tính năng có thể thêm âm thanh cho trò chơi thêm sinh động. Ngoài ra cũng áp dụng những kiến thức được học từ bộ do thầy Trương Toàn Thịnh giảng dạy.

II. Kịch bản và giao diện của trò chơi

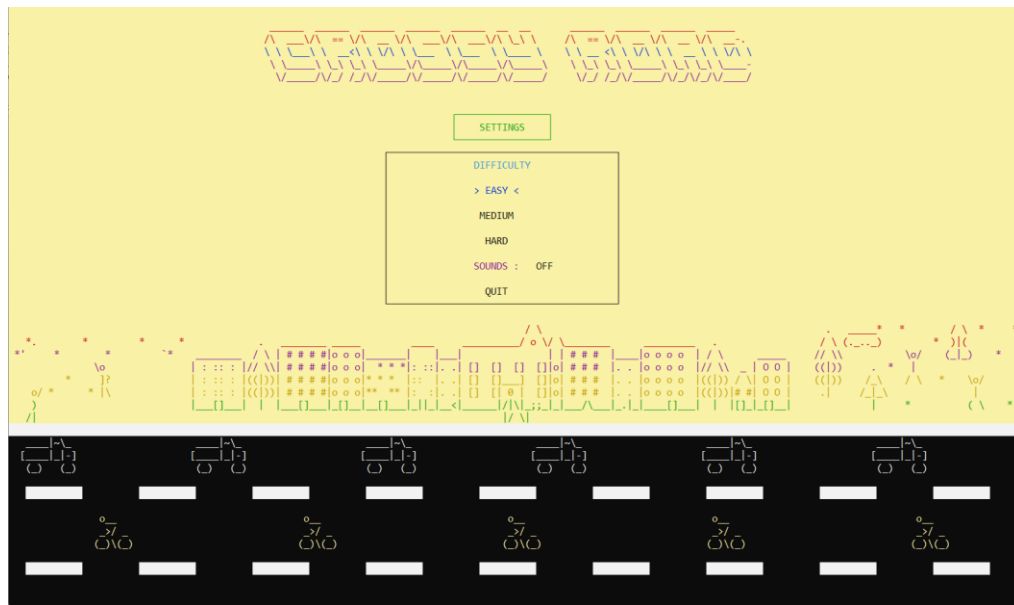
- Đồ án trò chơi băng qua đường được xây dựng với ngôn ngữ lập trình C++



Hình 1 – Giao diện mới vào khi chạy chương trình

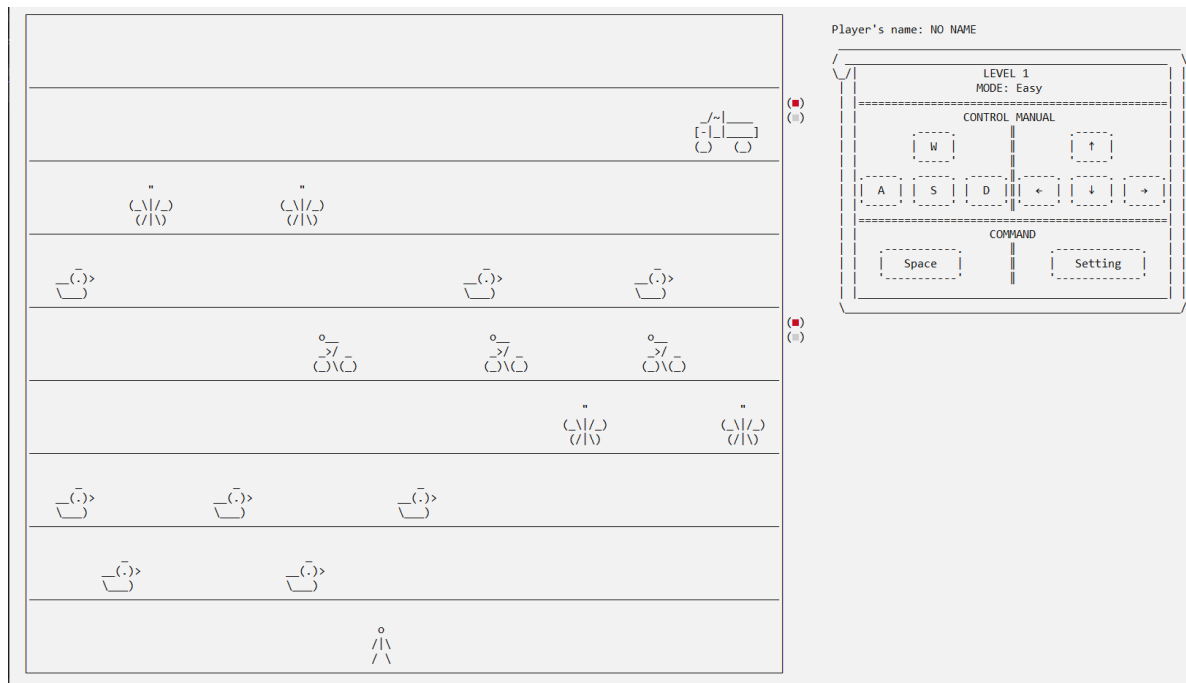


Hình 2 - Giao diện khi load game ở bên ngoài menu



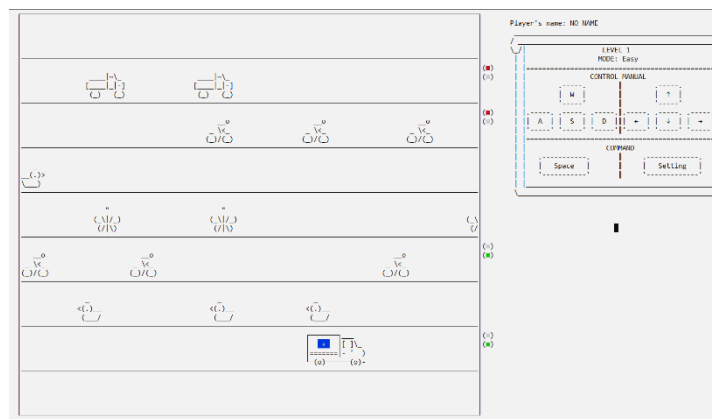
Hình 3 - Giao diện setting ở menu game

- Trước khi vào trò chơi, người chơi có thể ấn LOAD GAME mà mình đã lưu từ trước đó, và chọn tên người chơi mà mình đã lưu trước đó
- Ngoài ra, có thể ấn vào SETTINGS ở hình 1 để có thể chọn chế độ chơi, bật/ tắt âm nhạc ở hình 2

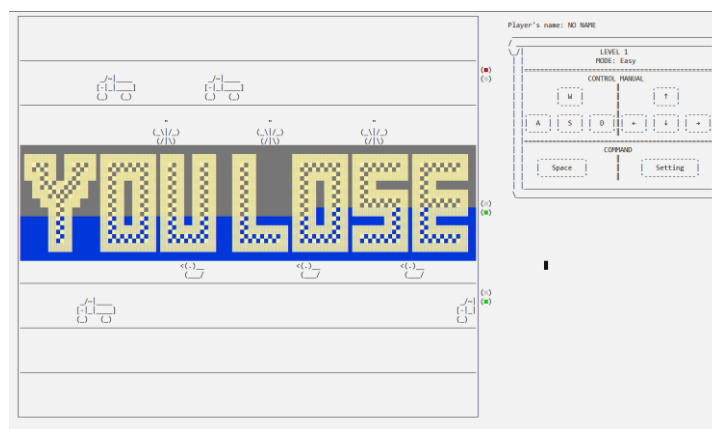


Hình 4 - Giao diện khi tham gia game

- Người chơi sẽ điều khiển nhân vật thông qua ấn các phím “w” – lên, “s” – xuống, “a” – trái, “d” – phải. Người chơi có nhiệm vụ đưa nhân vật vượt qua hết tất cả các làn đường để về đến vạch đích. Tùy vào vật thể ở mỗi làn đường, nếu như làn đường có xe hơi hay người đạp xe thì sẽ có tín hiệu đèn giao thông hiện ra.
- Cứ mỗi lần vượt qua các chướng ngại vật về đích, trò chơi sẽ nâng độ khó bằng cách tăng số lượng vật thể và tốc độ ở trên mỗi làn đường



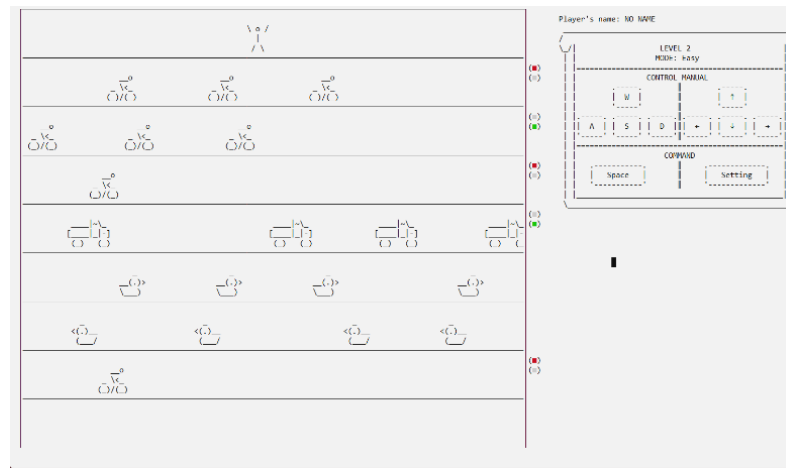
Hình 5



Hình 6

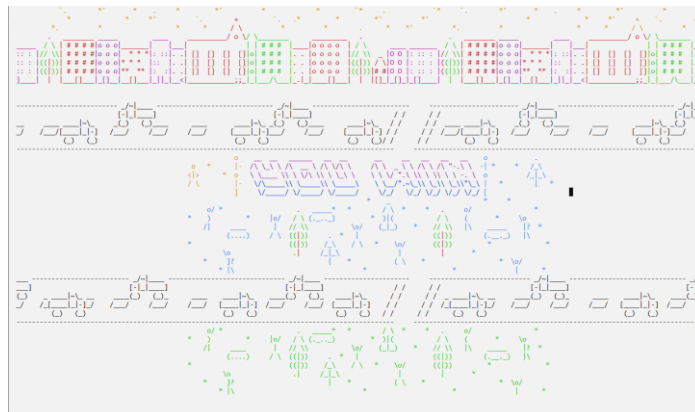
- Trong quá trình tham gia, người chơi đụng vật thể sẽ bị thua. Hiệu ứng sẽ như hình 5, hình 6 và còn có hiệu ứng âm thanh đi kèm.

Nếu như thua ở bất kì cấp độ nào cũng sẽ trở lại ban đầu và thoát ra màn hình chính.



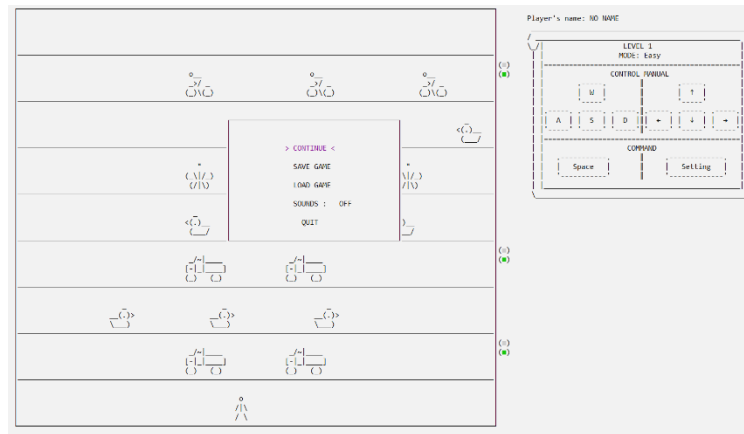
Hình 7 - Giao diện khi nhân vật đến đích

- Khi về đến vạch đích, hiệu ứng thắng mỗi chế độ, người chơi sẽ vỗ tay ăn mừng.



Hình 8 - Giao diện khi thắng hết các cấp độ

- Khi người chơi thắng hết tất cả các cấp độ, giao diện chúc mừng người chơi kèm với hiệu ứng chuyển màu từ “YOU WIN” và hiệu ứng âm thanh đi kèm.



Hình 9 - Settings trong game

- Trong quá trình tham gia trò chơi, người chơi có thể tạm dừng game lại bất kì lúc nào mà mình muốn. Khi tạm dừng game, người chơi ấn phím “space” trên bàn phím để dừng lại, màn hình sẽ hiện ra các cài đặt cho người chơi lựa chọn.

III. Các lớp đối tượng được xây dựng trong game

1. Lớp CPOSITION:

- Trò chơi cần lớp đối tượng là vị trí của đối tượng chơi, và những vật thể cản trở trên những làn đường

2. Lớp CPEOPLE:

- Trò chơi sẽ có đối tượng để tham gia trò chơi, lớp đối tượng CPEOPLE bao gồm:
 - o Các thuộc tính: Vị trí của người chơi, hình dạng của đối tượng tham gia trò chơi và trạng thái sống chết
 - o Các phương thức:
 - Các phương thức tên Up, Down, Left, Right sẽ cập nhật vị trí người chơi khi di chuyển trên những làn đường
 - Phương thức tên checkDead sẽ kiểm tra trạng thái sống chết của đối tượng chơi
 - Ngoài ra còn có những phương thức get của đối tượng để lấy những thuộc tính

3. Lớp COBSTACLE, CBIRD, CBUTTERFLY, CCAR, CDUCK

- Trong quá trình người chơi băng qua đường, các vật thể cản trở sẽ chạy, với mỗi làn sẽ có những vật thể khác nhau. Lớp đối tượng COBSTACLE được áp dụng đa hình.
- COBSTACLE là lớp cha và các lớp con là: CBIRD, CBUTTERFLY, CCAR, CDUCK.
- COBSTACLE bao gồm:
 - Thuộc tính là vị trí của những vật thể chạy trên đường, hình dạng của vật thể và hướng di chuyển của vật
 - Có những phương thức ảo lấy hình dạng của vật thể, âm thanh và loại vật thể, bên cạnh đó có các phương thức cập nhật vị trí của những vật thể
- Các lớp con sẽ có những hình dạng và âm thanh khác nhau, kế thừa tất cả các thuộc tính của lớp cha, không tạo thêm các thuộc tính nào trong từng lớp con.

4. Lớp CONELANE, CLANE

- Lớp CONELANE được xây dựng để chứa những vật thể và làm cho các vật thể chạy trên cùng một làn đường
- Lớp CONELANE chứa:
 - Các thuộc tính: danh sách vật thể có trong một làn đường và tốc độ di chuyển, vị trí đèn giao thông, hiệu ứng đèn giao thông, và hướng di chuyển.
 - Các phương thức:
 - pushObs: thêm vật cản vào trong danh sách trên cùng một làn
 - drawObs: vẽ từng vật thể trên làn đường và printAllObs: vẽ tất cả các vật thể và đèn giao thông trên làn đường
 - eraseObs: xóa từng vật thể và trong phương thức eraseAllObs xóa tất cả các vật thể trên làn đường và cập nhật hiệu ứng đèn giao thông
 - changeRedLight: thay đổi đèn giao thông
 - Ngoài ra còn có những phương thức get để lấy thông tin của những thuộc tính
- Lớp CLANE được xây dựng để chứa danh sách của tất cả các làn đường (cụ thể là danh sách CONELANE) và những phương thức vẽ, xóa và thêm vật thể vào trong danh sách của làn đường bất kì sẽ gọi lại hàm từ lớp đối tượng CONELANE.

5. Lớp CLEVEL

- Lớp đối tượng CLEVEL được xây dựng để tạo cho trò chơi đa dạng hơn về tốc độ di chuyển của vật thể dựa trên thời gian, và các cấp độ, chế độ chơi.
- Lớp bao gồm:
 - o Thuộc tính sẽ bao gồm cấp độ, chế độ trò chơi (Dễ, Trung Bình hay Khó và tốc độ di chuyển của vật thể (thời gian)
 - o Phương thức chooseTypeObs sẽ chọn vật thể được tạo ra trong tất cả các loại vật thể
 - o Ngoài ra còn có những phương thức set và get để cập nhật và lấy giá trị của các thuộc tính

6. Lớp CMAP

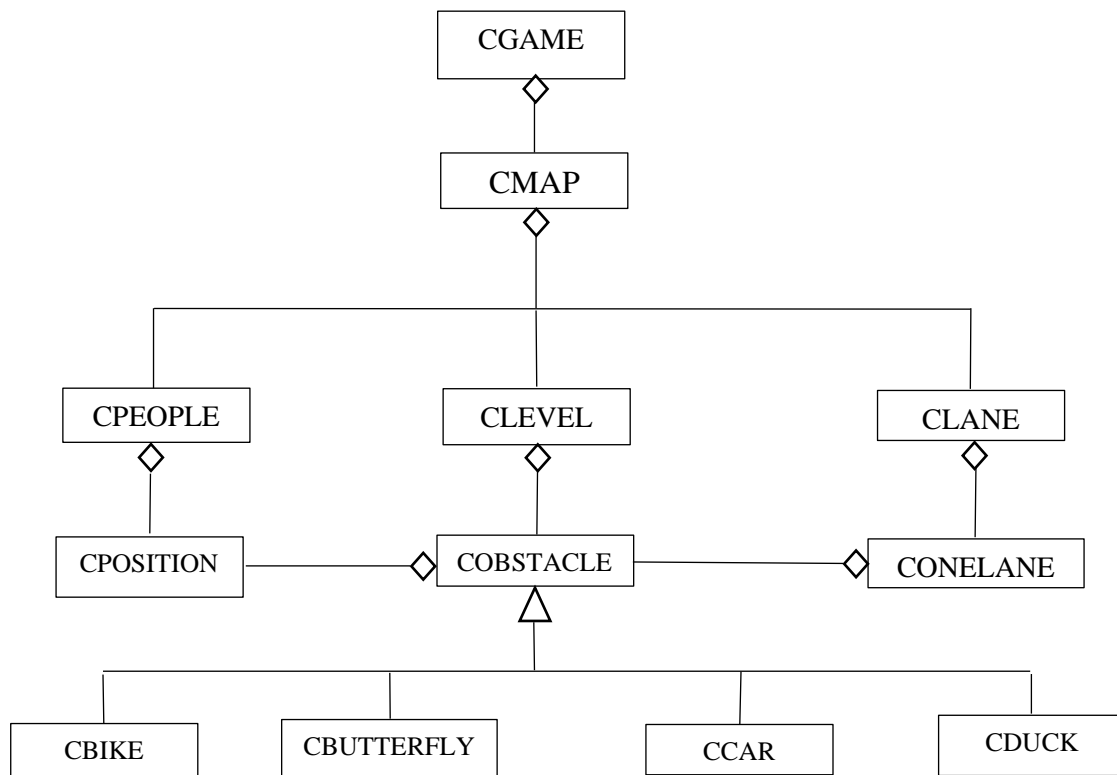
- Được tạo ra để chứa tất cả các thông tin hiện có của trò chơi:
- CMAP sẽ chứa thông tin của người chơi (CPEOPLE), thông tin của các vật thể trên các làn đường (CLANE) và những cấp độ, chế độ trò chơi (CLEVEL)
- Các phương thức có trong đối tượng:
 - o Vẽ tất cả giao diện từ bên ngoài và vào trong trò chơi, hiệu ứng khi người chơi thắng và thua: printMap, printFirstScreen, printSettings, printLoadGame.
 - o Vẽ, xóa và cập nhật vị trí, kiểm tra tình trạng của người chơi khi đã có dữ liệu của tất cả vật thể: drawPeople, erasePeople và Collisions
 - o Khởi tạo những thuộc tính của trò chơi (xóa thông tin cũ và tạo ra dữ liệu mới), cập nhật dữ liệu của trò chơi: initializeNewMap, updateLaneData
 - o Viết các phương thức cấp độ trò chơi để gọi các hàm trong CLEVEL
 - o Các phương thức kiểm tra tình trạng trò chơi đã kết thúc chưa: isEnd, isWin
 - o Ngoài ra có những phương thức get để lấy thông tin của những dữ liệu và phương thức đọc/ lưu tập tin

7. Lớp CGAME:

- Các đối tượng trong trò chơi đã được tạo đầy đủ, Lớp CGAME đóng vai trò trung tâm điều phối và sử dụng các đối tượng trong trò chơi thông qua lớp CMAP.

- Lớp CGAME: thuộc tính sẽ chứa thông tin của CMAP và những biến bool để hỗ trợ kĩ thuật đa tiêu tình và những chức năng lưu trữ và tải trạng thái trò chơi
- Các phương thức trong CGAME:
 - o printGame: in ra hiệu ứng di chuyển người chơi, các vật thể chạy trên các làn đường, tên người chơi
 - o controlGame: phương thức được xây dựng để tạo thêm luồng bên cạnh luồng chính, để cho người ấn các phím con người và những cài đặt có trong trò chơi
 - o establishLoadGame, establishSettings, gameSettings, askPlayerBeforeLoadingGame, loadGameInGameSettings là các phương thức cài đặt những chức năng của trò chơi trước và trong khi vào tham gia
 - o Ngoài ra còn có những phương thức để lưu lại từng thông tin khi người chơi muốn lưu lại.
 - o Phương thức chính được gọi trong hàm main: menuGame sẽ phối hợp và gọi những phương thức đã viết ở trên.

8. Sơ đồ lớp đối tượng tham gia trò chơi:



IV. Các kỹ thuật xử lý trò chơi

- Kỹ thuật vẽ các vật thể để tạo chuyển động:
 - Để làm vật thể chuyển động mượt hiện trên màn hình console, các đối tượng sau khi được in ra sẽ có 1 khoảng thời gian ngắn chờ, sau đó sẽ cập nhật vị trí mới của các đối tượng, và sau đó xóa đối tượng ở vị trí cũ và in ra ở vị trí mới.
 - Ví dụ: khi một chiếc xe di chuyển về phía trước, sau khi in xe sẽ có thời gian ngắn chờ, cập nhật vị trí dựa vào độ dời của xe, sau đó vẽ vị trí mới và xóa phần đuôi xe bị thừa ra ở vị trí cũ, quá trình diễn ra nhanh và liên tục
 - Sử dụng hàm gotoXY, để có thể đến vị trí bất kì trên màn hình console để vẽ các đối tượng
- Kỹ thuật xử lý đa tiêu trình:
 - Để điều khiển nhân vật trong trò chơi, người chơi sẽ phải ấn các phím “w” – lên, “s” – xuống, “a” – trái, “d” – phải. Tuy nhiên, khi chờ ấn các phím di chuyển, chương trình không thể chạy tiếp, các vật thể không di chuyển mà đứng chờ ở dòng lệnh `_getch()`, vì vậy để cho trò chơi có thể tiếp tục, vật thể có thể chạy đồng thời vẫn có nơi chờ dòng lệnh `_getch()` để người chơi có thể điều khiển nhân vật, chúng ta cần sử dụng đến kỹ thuật đa tiêu trình
 - Đề án của nhóm 4 sẽ gồm có 2 luồng:
 - Luồng chính xử lý các trò chơi sẽ chạy xuyên suốt chương trình, sẽ in và thể hiện các chức năng của trò chơi như ấn chọn vào trò chơi mới, tải dữ liệu có sẵn của người chơi, bật/ tắt âm thanh, vẽ các hiệu ứng, và các thông tin khác.
 - Luồng thứ 2 là khi người chơi ấn vào tham gia trò chơi, luồng sẽ mở ra để người chơi có thể điều khiển các nhân vật di chuyển, ngoài ra luồng này còn có chức năng tạm dừng trò chơi
- Kỹ thuật xử lý tập tin (chức năng lưu / tải trò chơi):
 - Người chơi muốn lưu trữ lại trạng thái hiện tại, chương trình sẽ tạo ra một tập tin với tên là tên của người chơi. Trong tập tin sẽ lưu trữ vị trí của người chơi; cấp độ và mức độ của trò chơi; tất cả các vị trí của vật thể hiện tại đang có, vậy khi tải

trò chơi, chương trình sẽ lấy tập tin để làm dữ liệu in các vật thể trên màn hình console.

- Trong quá trình đang chơi, khi lưu lại trò chơi, nếu như người chơi chưa có tên, chương trình sẽ tạo ra một tập tin tương ứng với tên người chơi, nếu đã có trong dữ liệu của chương trình, tập tin tương ứng sẽ tự động cập nhật lại trạng thái mới
- Khi tải trò chơi trong quá trình đang chơi, trên màn hình console sẽ hiện ra khung giao diện sẽ hỏi người chơi về việc lưu trò chơi trước khi tải.
- Kỹ thuật xử lí màn hình console:
 - Trong quá trình tham gia trò chơi, tùy vào mỗi máy, khi chạy chương trình sẽ hiện ra màn hình console kích thước khác nhau, ngoài ra người dùng cũng có thể thay đổi kích thước của màn hình dẫn đến các kích thước của vật thể, các tính toán trước đó trong các lớp đối tượng bị ảnh hưởng.
 - Vì vậy trong mã nguồn, đã xây dựng các hàm dựa vào thư viện Windows.h, trước tiên ta sẽ thay đổi kích thước màn hình console phù hợp với kích thước đồ họa trong trò chơi, sau đó gọi hàm để cố định lại kích thước, khóa lại không cho người dùng thay đổi kích thước:
 - Ngoài ra tạo ra hàm ẩn hiện con nháy ở trên màn hình console, khi sử dụng hàm gotoXY để đến với vị trí bất kì con nháy sẽ hiện ra ở vị trí muốn đến. Vì vậy sử dụng hàm để tạo hiệu ứng trông mượt mà hơn, không gây khó chịu cho người nhìn
- Kỹ thuật chèn âm thanh: Thư viện mmsystem.h, dùng hàm PlaySound có từ trong thư viện để chạy nhạc mà mình đã đặt chung với mã nguồn

V. Mã giả

- Đoạn mã giả phần tiểu trình chạy song song: có chức năng điều khiển người chơi và dùng màn hình:

```

while IsRunning == true: // IsRunning kiểm tra trò chơi có đang chạy
    // điều kiện để tạm dừng phần đa tiểu trình, isEnd kiểm tra việc
    người
    // chơi có thua không
    while IsPausing == false and isEnd == false and isWin == false:
        if not _kbhit():
            key = _getch()
            if key == 32:
                IsPausing = true
            else if key == 'a' or key == 's' or key == 'w' or key ==
or:
                play_sound (move)
                IsPausing = false
                updatePosPlayer(key)
            else if key == 224
                key = _getch()
        end_while
    end_while
end_while

```

- Đoạn mã giải phần thực hiện menu trò chơi và được gọi ở trong hàm main

```

setListFile () // lưu những dữ liệu có sẵn vào mảng lưu trữ tên
người chơi từ //file
printFirstScreen()
pos = 0
changeInput = true
while 1 :
    if changeInput == true:
        key = InputKey()
        swicth key:
            case 'w':
                pos --
                pos = (pos + 4) % 4
                break
            case 's':
                pos --
                pos = (pos + 4) % 4
                break
            case 13:
                swicth pos:
                    case 0:
                        changeInput = true
                        IsPausing = false

```

```

do:
    printGame()
    gameSettings()
    if isWin:
        check = nextLevel()
        if check == false
            break
        else isLoaded = false
    while IsPausing == false and IsRunning == true
and isEnd() == true
    end_while
    resetLevel()
    printFirstScreen()
    break
case 1:
    changeInput = true
    type = establishLoadGame()
    if type == 1
        changeInput = false
        key = 13
        pos = 0
    break
case 2:
    changeInput = true
    establishSettings()
    break
case 3:
    IsRunning = false
    return
end_while

```

VI. Nguồn tài liệu tham khảo

- Hiệu ứng nhạc:
 - o <https://kienthuc24h.com/playsound-trong-visual-studio-c/>
- Kiến thức đa tiến trình:
 - o <https://codelearn.io/sharing/multithreading-trong-cpp-la-gi>
 - o <https://www.geeksforgeeks.org/multithreading-in-cpp/>
- Hiệu ứng chữ và màu:
 - o <https://www.asciiart.eu/>

