

BÁO CÁO ĐỒ ÁN MÔN HỌC IE221.N21.CNCL

SINH VIÊN THỰC HIỆN

Mã sinh viên: 19522411

Họ và tên: Phan Thành Bảo Trọng

TÊN ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG GAME NARUTO ĐỐI KHÁNG

CÁC NỘI DUNG CẦN BÁO CÁO:

1. Giới thiệu đồ án
2. Tóm tắt quá trình thực hiện
3. Kết quả đạt được
4. Tài liệu tham khảo
5. Phụ lục 1: Giới thiệu (demo) kết quả
6. Phụ lục 2: docstring

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỒ ÁN	3
1.1. Giới thiệu về đề tài.....	3
1.2. Mục tiêu của đề tài.....	3
CHƯƠNG 2. TÓM TẮT QUÁ TRÌNH THỰC HIỆN	4
2.1. Yêu cầu phần mềm.....	4
2.2. Quy trình phát triển	4
2.2.1. Thiết kế, xây dựng gameplay	4
2.2.2. Lập trình	4
2.2.3. Kiểm tra.....	5
2.3. Mô hình quy trình	5
2.4. Kiến trúc phần mềm.....	5
2.5. Quy trình xây dựng ứng dụng game: Naruto đối kháng	6
2.5.1. Thiết kế giao diện và xử lý.....	6
2.5.2. Lập trình game	7
2.6. Bảng tóm tắt quá trình thực hiện.....	8
CHƯƠNG 3. KẾT QUẢ ĐẠT ĐƯỢC	11
3.1 Kết quả đạt được	11
TÀI LIỆU THAM KHẢO.....	13
PHỤ LỤC 1: GIỚI THIỆU (DEMO) KẾT QUẢ	14
PHỤ LỤC 2: DOCSTRING.....	21

CHƯƠNG 1. GIỚI THIỆU ĐỒ ÁN

1.1. Giới thiệu về đề tài

- Làm một dự án game là một trong những kế hoạch mà em đã dự định sẵn từ khi mới bước chân vào giảng đường đại học những ngày đầu tiên, với môn học này em đã có cơ hội thực hiện được mong muốn đó. Game mà em thực hiện có tên là Naruto đối kháng. Đây là một tựa game thuộc thể loại đối kháng của những nhân vật hoạt hình vô cùng hấp dẫn và lôi cuốn. Game cho phép 2 nhân vật đánh với nhau 3 hiệp liên tiếp, nếu nhân vật nào thắng được 2 trận trước sẽ là người chiến thắng

- Đồ án này sử dụng ngôn ngữ Python và các công cụ hỗ trợ như Pygame [1], Pyvidplayer. Em sử dụng Pygame làm nền tảng cho các module cần thiết cho game như vòng lặp game, xử lý va chạm, rendering và các hàm tiện ích trong game.

- Mã nguồn github: <https://github.com/trong5086/Naruto-Pygame>

1.2. Mục tiêu của đề tài

- Mục tiêu là xây dựng được một game đối kháng dễ chơi, đồ họa ưa nhìn, nhạc không quá ồn. Đáp ứng được những tiêu chí sau:

- Lấy game đối kháng với nhân vật được xây dựng dựa trên phim hoạt hình “Naruto” làm cơ sở
- Được phép chọn nhân vật, địa hình thi đấu
- Có nhiều chế độ chơi như: Người với người, người với máy,...
- Thêm yếu tố âm nhạc
- Thêm yếu tố âm thanh cho nhân vật khi đánh nhau
- Thêm yếu tố tính điểm và phân thắng thua
- Khi kết thúc trận đấu có chức năng chơi lại hoặc trở về menu chính
- Giữa trận đấu nếu muốn có thể trở về menu chính

Trong báo cáo này, em tập trung trình bày 5 nội dung chính: (1) Yêu cầu phần mềm, (2) Quy trình phát triển, (3) Mô hình quy trình, (4) Kiến trúc phần mềm, (5) Quy trình xây dựng ứng dụng game: Naruto đối kháng

CHƯƠNG 2. TÓM TẮT QUÁ TRÌNH THỰC HIỆN

2.1. Yêu cầu phần mềm

- Yêu cầu nghiệp vụ: Chơi, Cài đặt
- Yêu cầu hệ thống: Hạn chế lỗi khi chạy
- Yêu cầu chất lượng: Giao diện dễ nhìn, thân thiện, gameplay đơn giản

2.2. Quy trình phát triển

2.2.1. Thiết kế, xây dựng gameplay

- Khởi đầu game người chơi phải lựa chọn chế độ chơi, tiếp đến là nhân vật và sau cùng là địa hình thi đấu
- Hai người chơi phân định thắng thua bằng cách chiến đấu, gây sát thương tới khi nào người chơi còn lại hết máu đầu tiên thì người chơi còn lại sẽ ghi được một điểm.
- Một lượt chơi có 3 vòng đấu nhân vật nào đạt được 2 trận thắng đầu tiên sẽ giành chiến thắng chung cuộc.
- Khởi đầu trận đấu 2 nhân vật đều có cùng 1 cây máu, tuy nhiên kỹ năng đánh sẽ là khác nhau phụ thuộc vào tùy nhân vật mà mình chọn

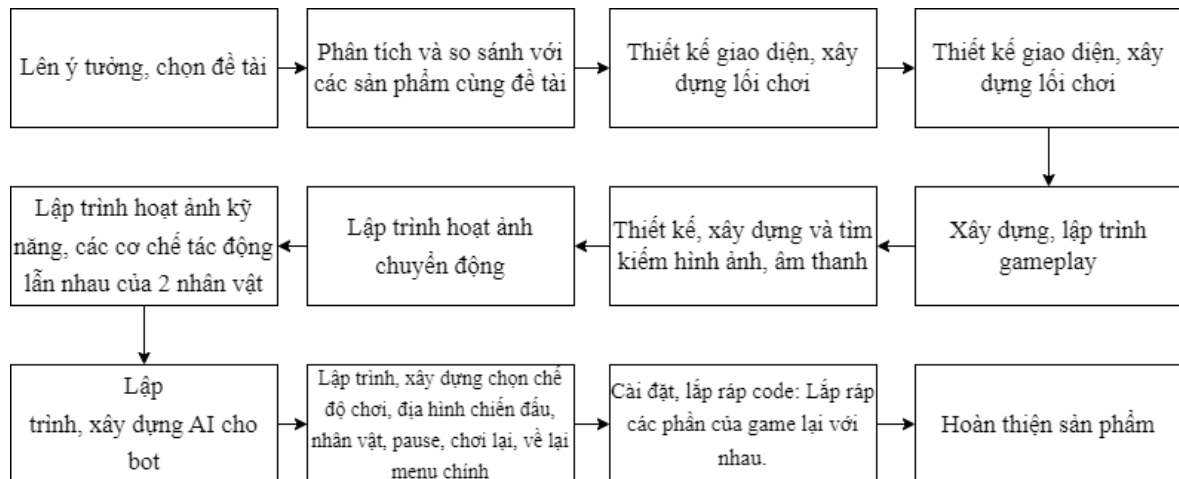
2.2.2. Lập trình

- Lập trình gameplay
- Thiết kế, xây dựng và tìm kiếm hình ảnh, âm thanh
- Lập trình hoạt ảnh chuyển động
- Lập trình hoạt ảnh kỹ năng, các cơ chế tác động lẫn nhau của 2 nhân vật
- Lập trình, xây dựng AI cho bot
- Lập trình, xây dựng chọn chế độ chơi, địa hình chiến đấu, nhân vật
- Lập trình các bước còn lại như: Chơi lại, Trở về menu, pause game
- Cài đặt, lắp ráp code: Lắp ráp các phần của game lại với nhau.

2.2.3. Kiểm tra

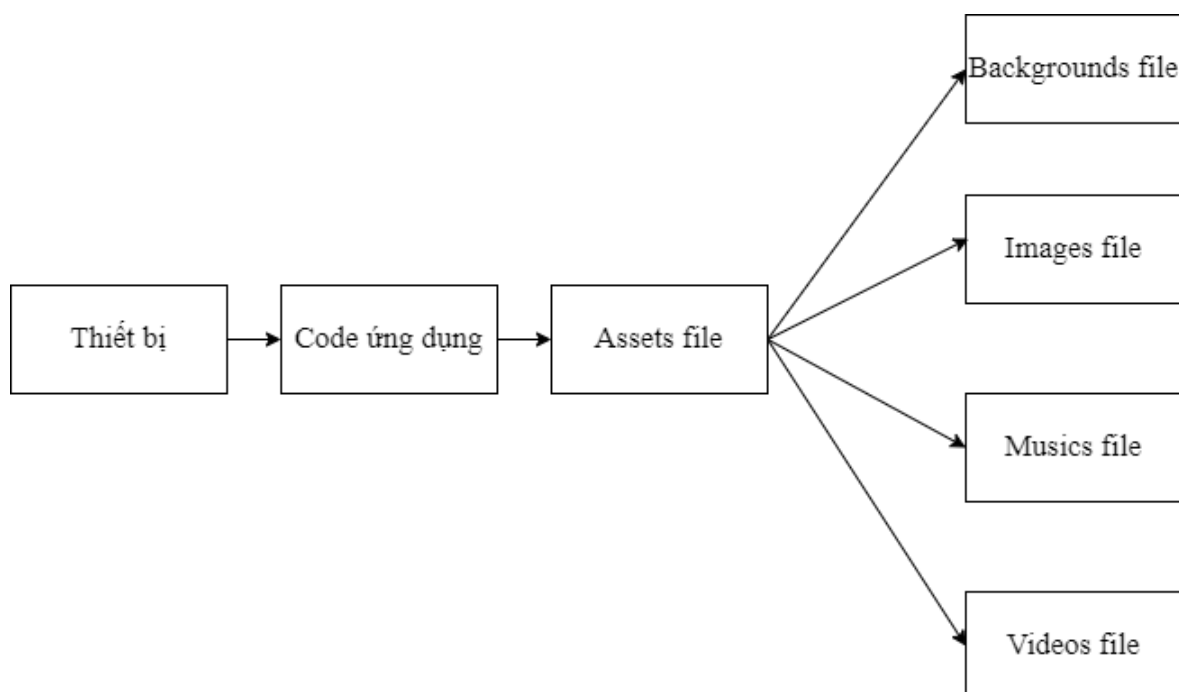
- Kiểm thử và sửa chữa: Kiểm lại các lỗi về phần code, hình ảnh, âm thanh. Đồng thời chơi thử và cân bằng lại game, chỉnh lại gameplay.

2.3. Mô hình quy trình



Ảnh 2.1 Ảnh mô hình quy trình

2.4. Kiến trúc phần mềm

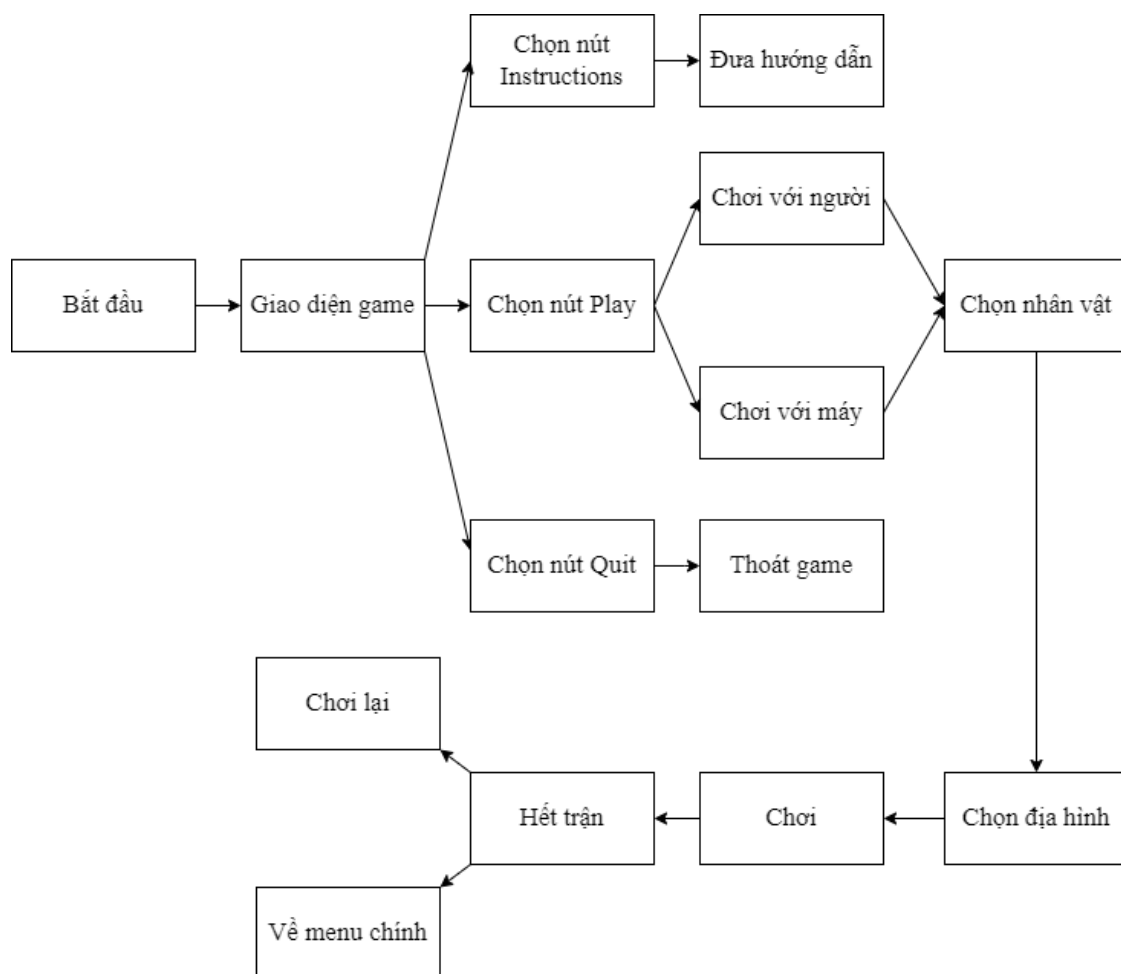


Ảnh 2.2 Ảnh kiến trúc phần mềm

- Code ứng dụng: có vai trò thực hiện chương trình, hiện tương tác với người dùng, có thể truy cập các tài nguyên trong thư mục assets phục vụ cho việc hiển thị và thực thi chương trình.
- Assets file: là nơi lưu trữ các tài nguyên của game.
- Backgrounds file: là nơi lưu trữ hình ảnh nền của game.
- Images file: là nơi lưu trữ hình ảnh nhân vật, thông tin của game.
- Musics file: là nơi lưu trữ các tệp âm thanh và âm nhạc của game.
- Videos file: là nơi lưu trữ các tệp video của game.

2.5. Quy trình xây dựng ứng dụng game: Naruto đối kháng

2.5.1. Thiết kế giao diện và xử lý



Ảnh 2.3 Ảnh thiết kế giao diện

2.5.2. Lập trình game

- Bước đầu tiên cần xác định khung hình của game. Tiếp theo tạo một lớp Player ở lớp này ta xây dựng thông tin, hoạt ảnh đánh nhau, mất máu, chết, di chuyển, bắn phi tiêu,... của nhân vật.

- Tiếp theo bằng thuộc tính của lớp Player ta dễ dàng chia nhân vật thành chế độ người với người nhờ vào việc đánh số 1 và 2 (1 tương ứng với Player 1 và 2 tương ứng với Player 2)

- Tiếp theo ta tiến hành xây dựng lớp AIplayer để AI hóa player 2, hiện thực chế độ người với máy

- Tiếp theo ta xây dựng opening của game bằng module Pyvidplayer

- Tiếp theo ta xây dựng âm thanh cho nhân vật khi bị trúng chiêu,...

- Tiếp đến ta xây dựng nút Play, rồi xây dựng giao diện chọn chế độ chơi, rồi chọn nhân vật, rồi giao diện chọn địa hình chiến đấu.

- Tiếp đến ta xử lý pause game hay chơi lại và trở về menu chính khi kết thúc 1 trận đấu

- Tiếp theo ta xây dựng những hoạt ảnh còn lại: Tính điểm, đếm ngược khi bắt đầu ván đấu thứ 2 trở đi, icon nhân vật,...

- Cuối cùng ta lắp ráp lại và tiến hành chỉnh sửa lỗi, kiểm thử, cân bằng lại các thông số cũng như gameplay để tránh game trở nên quá lỗi về lối chơi.

2.6. Bảng tóm tắt quá trình thực hiện

Mốc thời gian	Công nghệ sử dụng	Mức độ hoàn thành	Mục tiêu thực hiện	Kết quả đạt được
08/05/2023 – 14/05/2023	- Thư viện Pygame, sys	100%	<ul style="list-style-type: none"> - Tìm hiểu sơ bộ về thư viện Pygame - Tạo cửa sổ game, bắt sự kiện nút nhập - Tạo chuyển động khối giả lập: Nhảy, chạy qua trái, chạy qua phải 	<ul style="list-style-type: none"> - Hiểu sơ bộ được các hàm cơ bản cũng như logic của thư viện pygame - Tạo cơ bản được cửa sổ game, các khối có thể chuyển động nhờ vào bấm các nút trên bàn phím
15/05/2023 – 21/05/2023	- Thư viện pygame, sys, module Pyvidplayer	95%	<ul style="list-style-type: none"> - Thêm ảnh nhân vật, thanh máu,... cùng nhiều hoạt ảnh khác - Thêm hiệu ứng âm thanh, tạo opening cho game. - Chỉnh sửa ảnh chứa các hoạt ảnh nhân vật cho phù hợp 	<ul style="list-style-type: none"> - Tạo được nhân vật với một số chuyển động đơn giản tuy nhiên vẫn chưa được đẹp - Tạo được opening của game nhờ vào module Pyvidplayer và âm thanh tương tác

22/05/2023 – 28/05/2023	- Thư viện pygame, sys	100%	<ul style="list-style-type: none"> - Thêm hoạt ảnh chuyển động cho nhân vật như: chạy, phóng phi tiêu, chết, đánh nhau,... - Tạo toàn bộ các màn hình menu cũng như các chức năng pause, hay chơi lại hoặc trở về menu khi trong trận 	<ul style="list-style-type: none"> - Cơ bản tạo được các chuyển động - Hoàn thành hết các màn hình menu - Thêm chức năng pause, chơi lại hay trở về menu - Ngoài ra còn tạo thêm được chữ thông báo người chiến thắng
29/05/2023 – 04/06/2023	- Thư viện pygame, sys	100 %	<ul style="list-style-type: none"> - Hiện thực chức năng người đánh với người, người đánh với máy - Tạo điểm số cho nhân vật trong trận đấu - Hiện thực được từ trận đấu thứ 2 trở đi sẽ đếm ngược 3 giây 	<ul style="list-style-type: none"> - Hoàn thành những thứ còn sót lại của game như: chức năng người với người, máy với máy, tạo được điểm số cũng như thông báo người chiến thắng, tạo đếm ngược trận đấu. Ghép

			<ul style="list-style-type: none"> - Sửa lỗi giao diện, lỗi chức năng còn tồn đọng - Ghép toàn bộ các chức năng rời rạc thành 1 game hoàn chỉnh - Build thành 1 file game exe 	<p>toàn bộ các màn hình rời rạc cũng như các chức năng lại với nhau thành 1 game hoàn chỉnh. Cuối cùng build nó thành 1 file game đuôi exe hoàn chỉnh có ảnh minh họa</p>
--	--	--	--	---

CHƯƠNG 3. KẾT QUẢ ĐẠT ĐƯỢC

3.1 Kết quả đạt được

Công việc đã làm	Mức độ hoàn thành	Khó khăn	Thuận lợi	Mục tiêu tương lai
- Xây dựng được opening cho phần mở đầu game	100%	- Xảy ra nhiều lỗi khi chèn video làm opening vì chưa thực sự hiểu luồng của thư viện pygame	- Tồn tại một vài video trên youtube hướng dẫn cách chèn, hay một số tài liệu liên quan trên internet	- Tự mình xây dựng một video thật đẹp về game lấy ảnh từ game mà tạo nên chính opening của game đó
- Xây dựng được các hoạt ảnh khá ổn, không quá xấu	100%	- Việc tìm ảnh và chỉnh sửa ảnh tương đối khó khăn vì chưa có kinh nghiệm	- Có một số tài liệu tham khảo	- Xây dựng lại giao diện thật đẹp và bắt mắt
- Game có gameplay tạm ổn	100%	- Tính toán các sát thương cũng như lập ra các luồng tương đối khó vì ít tài liệu và vì muốn cá nhân hóa nên một số thứ bắt buộc phải tự sáng tạo. Ít	- Không có thuận lợi	- Chỉnh sửa lại cho gameplay cân bằng hơn

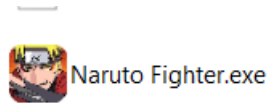
		thời gian tìm hiểu		
- Game dễ cài đặt, dễ chơi, ít lỗi	100%	- Ít thời gian tìm hiểu	- Không có	- Tối ưu hóa code giảm dung lượng của file exe
- Hoạt ảnh chuyển động của game như: đánh nhau, bị gây sát thương, chết và hoạt ảnh khác khá mượt. các màn hình menu khá ổn, bot tạm chấp nhận	100%	- Ít thời gian nghiên cứu vì phải đi thực tập	- Một số tài liệu trên mạng tuy không nhiều nhưng vẫn góp một ít cho việc nghiên cứu và code game	- Chỉnh sửa hoạt ảnh đẹp hơn, mượt hơn, thêm nhiều hoạt ảnh mới, thêm skill, thêm combo ,....

TÀI LIỆU THAM KHẢO

- [1 R. D. P. S. N. D. T. K. Lenard Lindstrom, "Pygame," 28 10 2000. [Online].
] Available: <https://www.pygame.org/news>. [Accessed 08 05 2023].
- [2 B. M. Châu, "Code learn," CodeLEARN, 27 10 2020. [Online]. Available:
] <https://codelearn.io/sharing/lap-trinh-game-co-ban-voi-pygame>. [Accessed 10 05 2023].
- [3 T. Despoudis, "ActiveState," ActiveState, 22 3 2021. [Online]. Available:
] <https://www.activestate.com/blog/how-to-use-pygame-for-game-development/>.
[Accessed 09 05 2023].
- [4 Ankthon, "Geeksforgeeks," Geeksforgeeks, [Online]. Available:
] <https://www.geeksforgeeks.org/python-display-images-with-pygame/>.
[Accessed 22 5 2023].
- [5 Seth Kenlon, Jess Weichler, "OpenSource," OpenSource, 18 12 2017. [Online].
] Available: <https://opensource.com/article/17/12/game-python-moving-player#:~:text=To%20move%20a%20sprite%20in,control%20function%20you%20created%20earlier..> [Accessed 13 05 2023].
- [6 I. ALAM, "makeuseof," makeuseof, 10 2 2023. [Online]. Available:
] <https://www.makeuseof.com/display-images-with-pygame/>. [Accessed 22 05 2023].
- [7 L. Lindstrom. [Online]. Available: <https://www.pygame.org/news>.
]

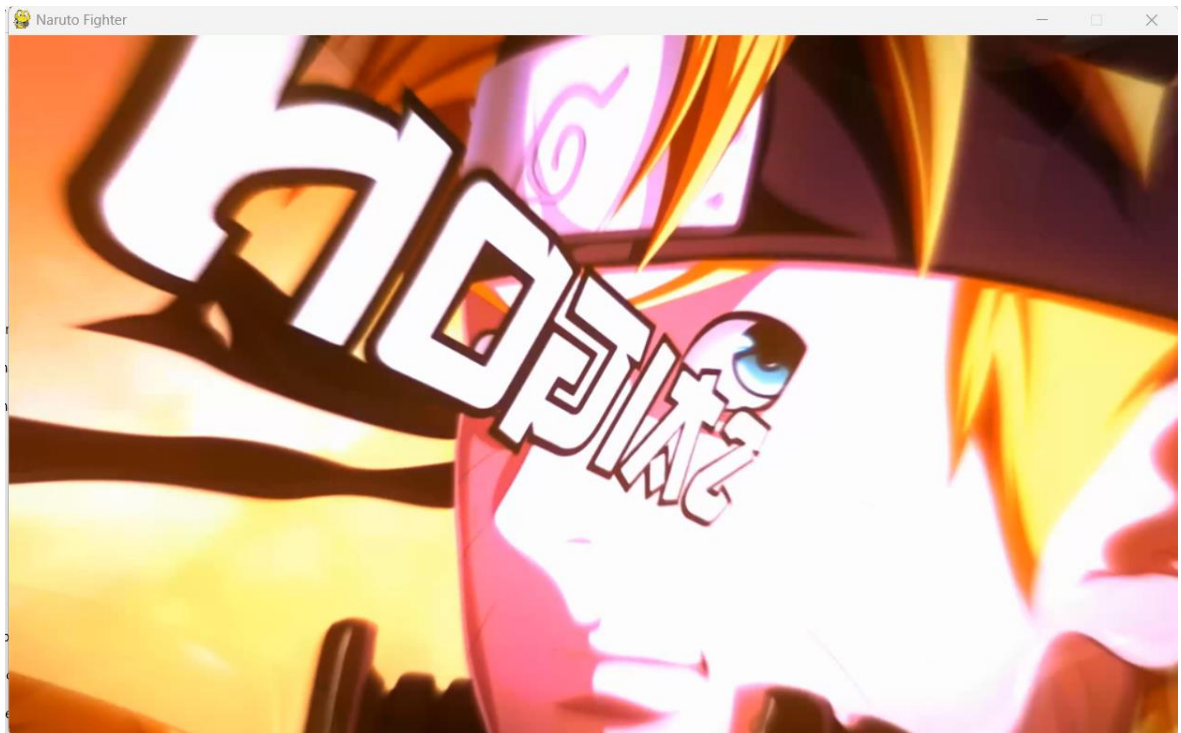
PHỤ LỤC 1: GIỚI THIỆU (DEMO) KẾT QUẢ

- Game sau khi được build thành file exe



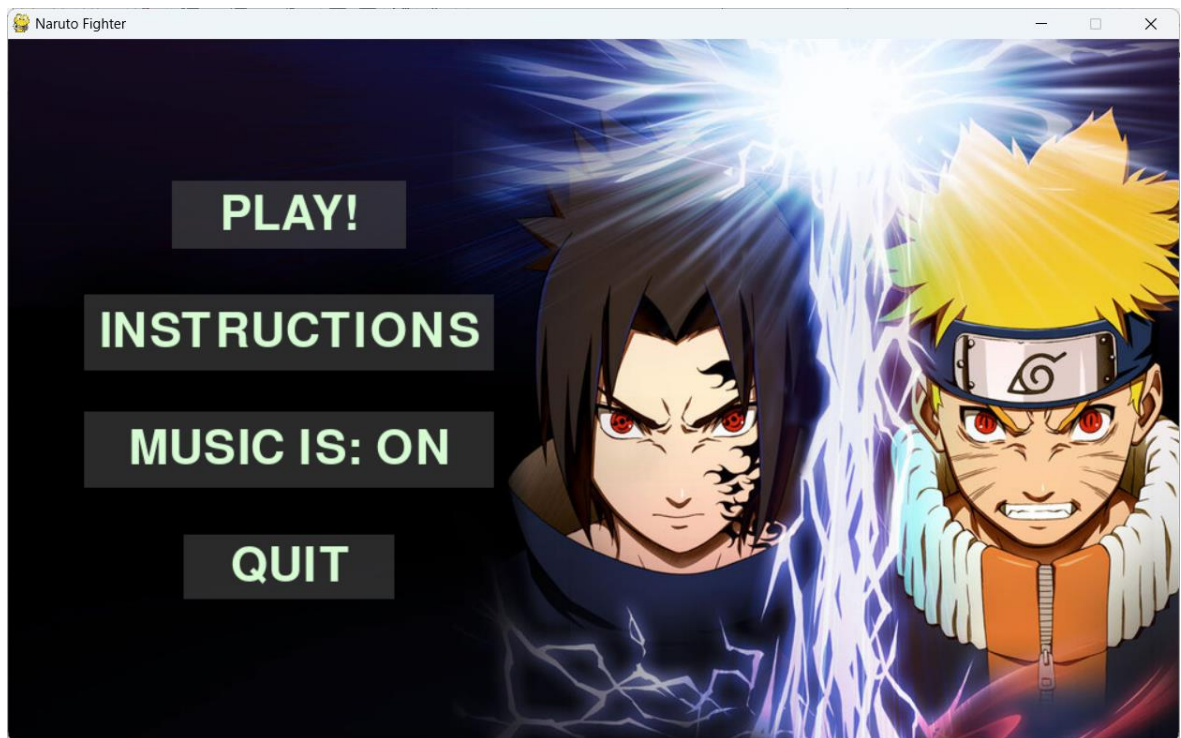
Ảnh file exe của game

- Màn hình opening khi vừa vào game.



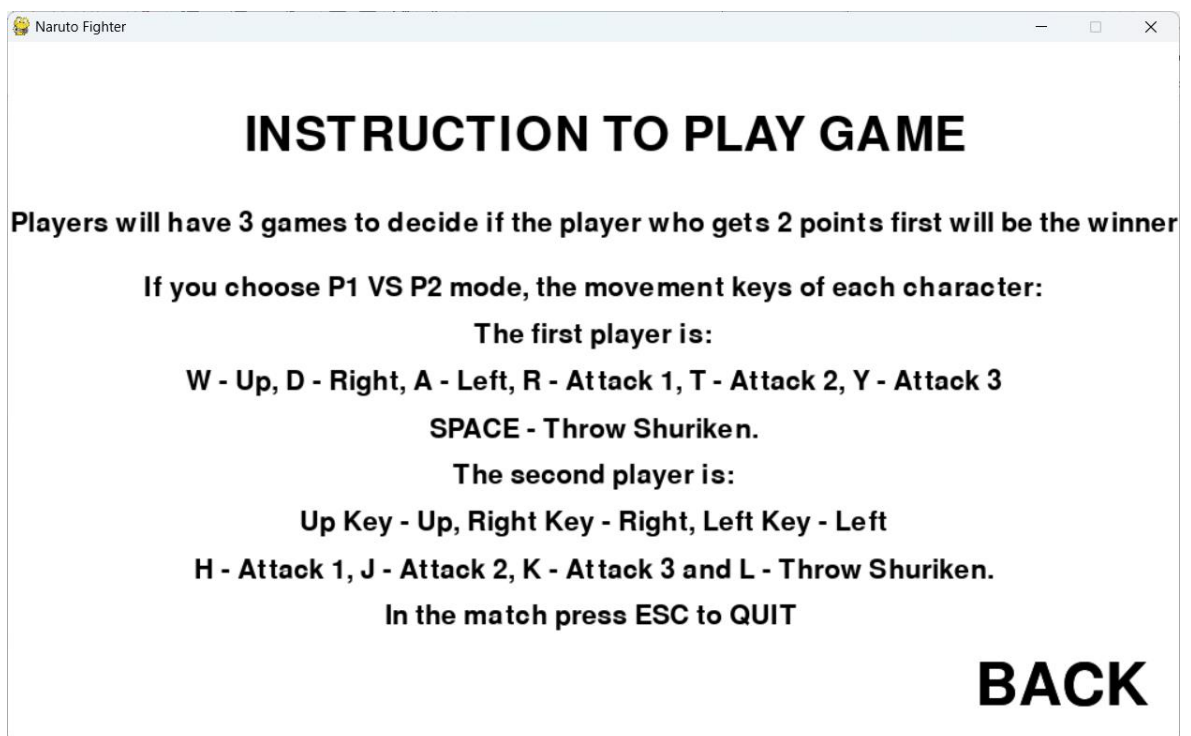
Ảnh opening game

- Màn hình menu chính gồm có 4 nút khi hover vào thì sẽ đổi màu trắng: Play, Instruction, Music is On/Off, Quit. Ta ấn vào nút Music để bật tắt nhạc và phím Quit để thoát game.



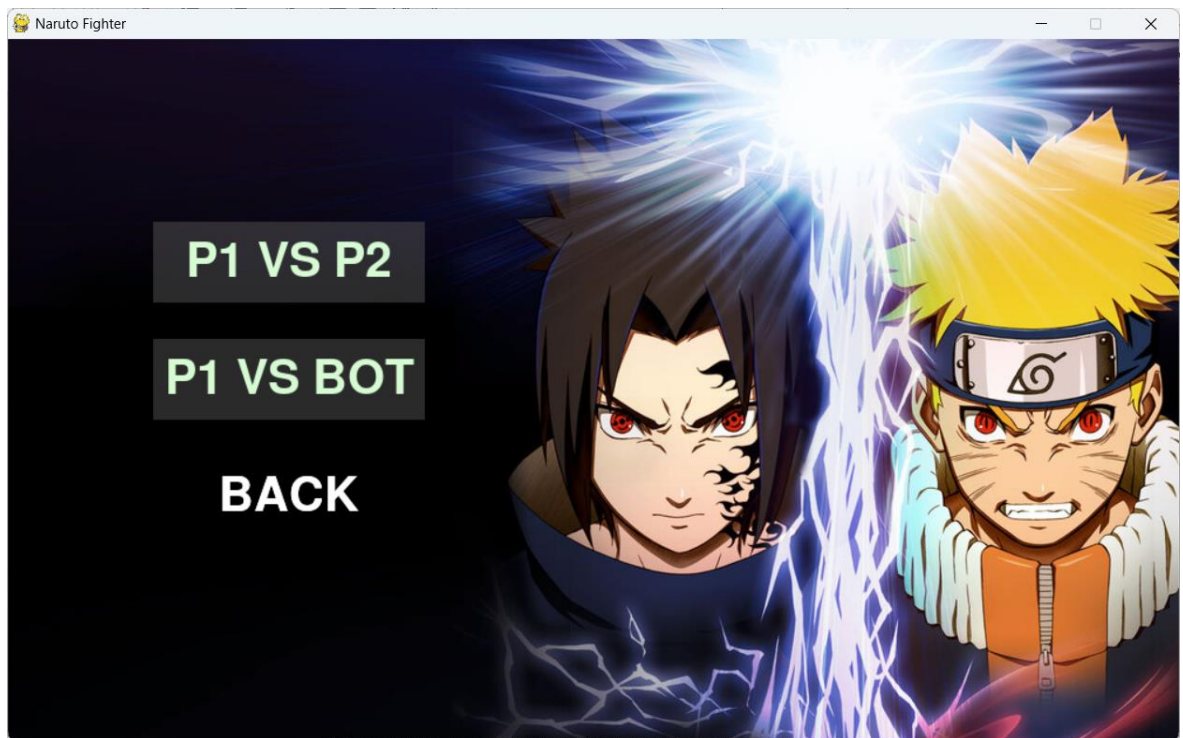
Ảnh menu chính

- Màn hình khi ta ấn vào nút Instructions: Sẽ giới thiệu sơ về gameplay cũng như hướng dẫn các phím để điều khiển nhân vật



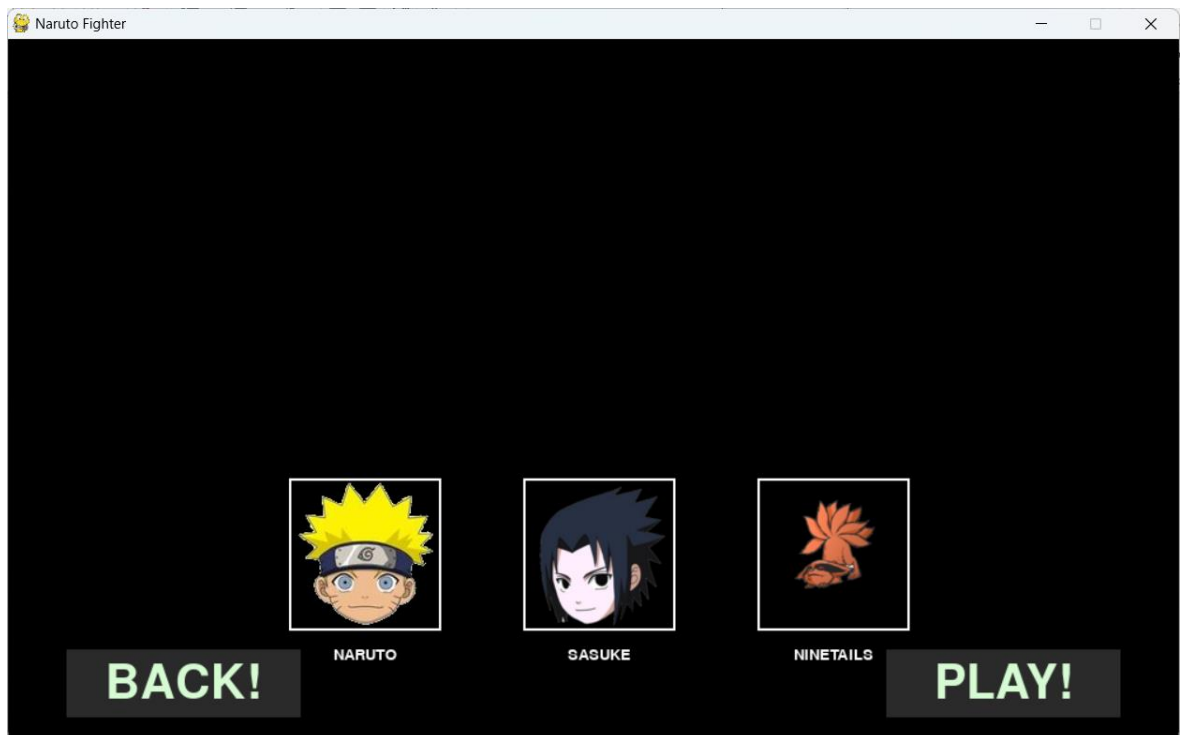
Ảnh hướng dẫn game

- Màn hình khi ta ấn vào nút Play khi đó sẽ xuất hiện 3 phím gồm: P1 vs P2, P1 vs Bot, là các chế độ chơi và phím Back để trở về menu chính

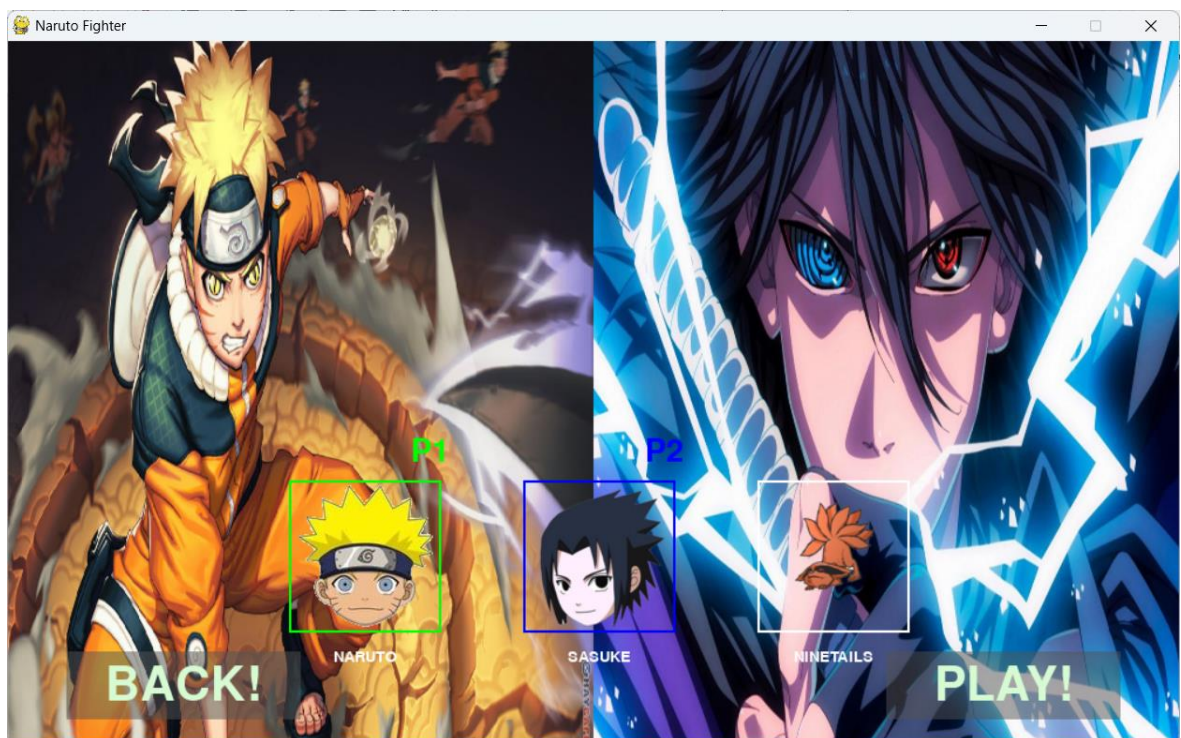


Ảnh chọn chế độ

- Tại màn hình tiếp theo khi ta ấn vào nút P1 vs P2, màn hình chọn nhân vật đã hiển thị với 2 nút xuất hiện là Back và Play khi đó ta cần chọn 2 nhân vật cho người chơi thứ nhất và hai nếu không chọn đủ thì khi nhấn vào nút Play sẽ không có gì xảy ra ngược lại thì ta sẽ chuyển màn hình đến chọn địa hình, khi ấn nút Back thì màn hình sẽ trở về chọn chế độ

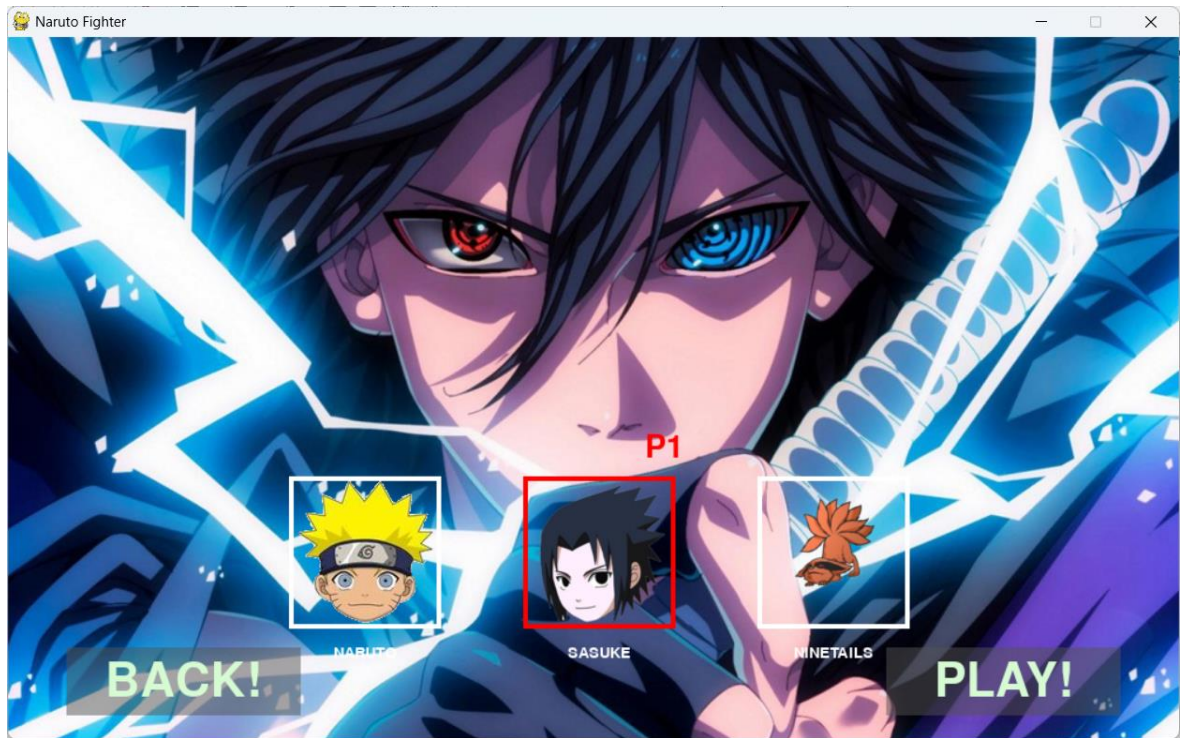


Ảnh chọn nhân vật



Ảnh chọn nhân vật chế độ P1 vs P2

- Tại màn hình P1 vs Bot cũng sẽ tương tự với màn hình P1 vs P2 tuy nhiên chỉ được chọn 1 nhân vật



Ảnh chọn nhân vật chế độ P1 vs Bot

- Tiếp theo là màn hình chọn địa hình, tương tự như màn hình chọn nhân vật gồm 2 nút Play và Back dĩ nhiên khi ta chọn được địa hình thì đấu thì mới chọn nút Play được

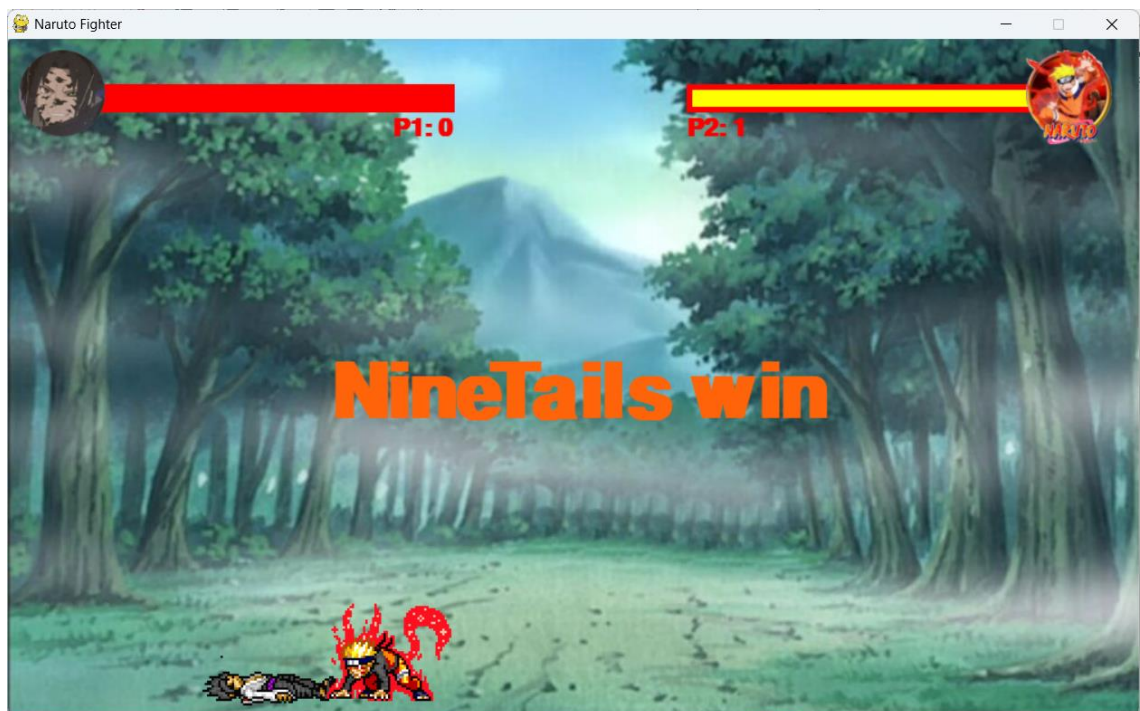


Ảnh chọn địa hình chiến đấu

- Tiếp đến là màn hình thi đấu khi này 2 nhân vật sẽ thi đấu với nhau bên nào hết máu trước sẽ thua và người còn lại được 1 điểm, người chơi nào đạt được 2 điểm trước sẽ chiến thắng

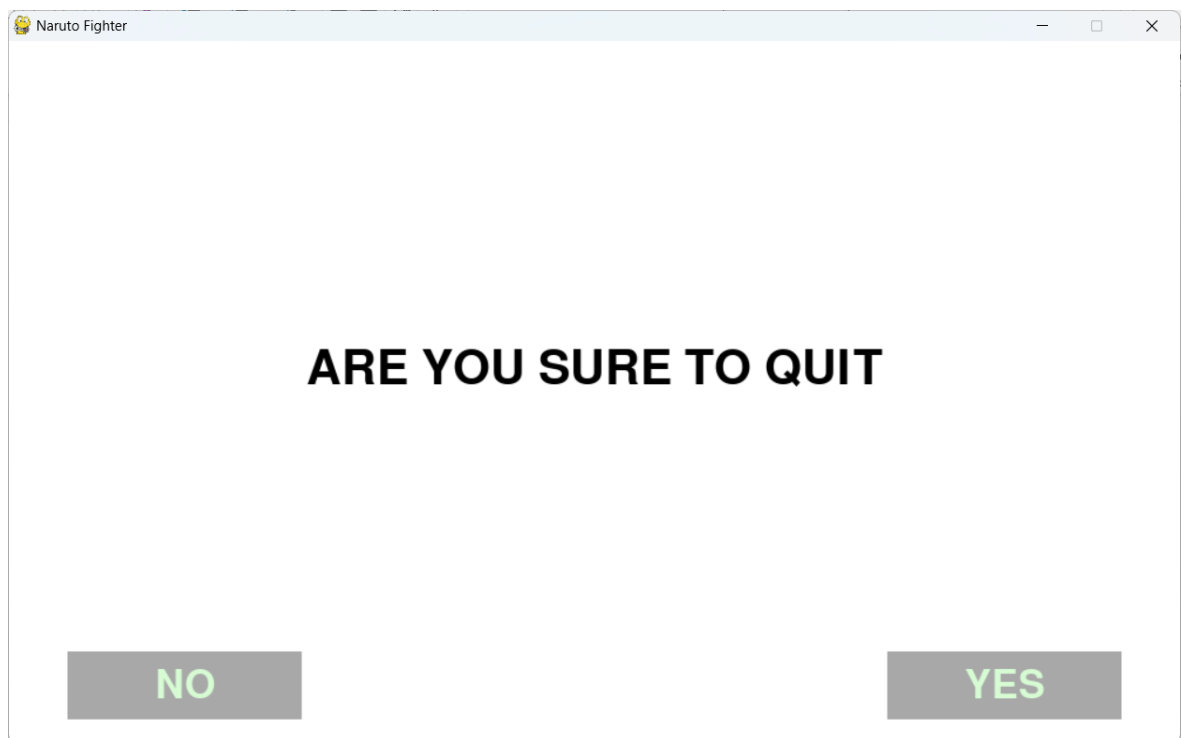


Ảnh vào trận game



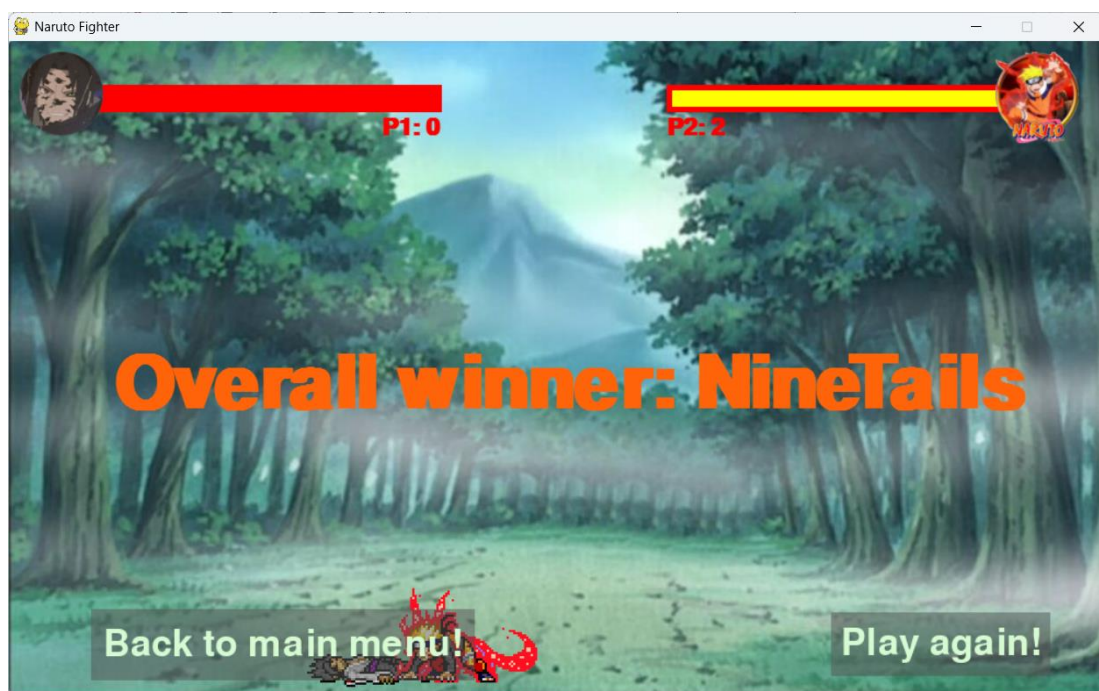
Ảnh P1 chết và P2 thắng

- Màn hình Pause, đây là màn hình khi ta muốn dừng trận đấu hoặc trở về menu chính



Ảnh màn hình Pause

- Màn hình kết thúc trận đấu: Có 2 nút để chơi lại hoặc trở về menu chính



Ảnh P2 chiến thắng 2 trận

PHỤ LỤC 2: DOCSTRING

- Mã nguồn github: <https://github.com/trong5086/Naruto-Pygame>
- File source code gồm:
 - Button.py:
 - Module chứa class Button dùng để tạo các nút trong menu và các nút trong toàn bộ chương trình, gồm 4 phương thức
 - def __init__(self, image, pos, text_input, font, base_color, hovering_color): Phương thức khởi tạo dùng để khởi tạo nút
 - Input: Ảnh nút, vị trí nút, chữ, font, màu nút, màu nút khi rê chuột vào
 - Output: Không có
 - def update(self, screen): Phương thức update cập nhật nút xử lý lên màn hình giải quyết case khi nút có ảnh và không có ảnh
 - Input: Màn hình giao diện game
 - Output: Không có
 - def checkForInput(self, position): Phương thức checkForInput bắt sự kiện khi chuột click vào button
 - Input: Vị trí của chuột
 - Output: True khi click trúng button, False khi không trúng
 - def changeColor(self, position): Phương thức changeColor đổi màu nút khi chuột hover vào
 - Input: Vị trí của chuột
 - Output: Không có
 - main_menu.py
 - Module dùng để tạo toàn bộ menu trong game, gồm 8 hàm
 - Hàm def get_font(size, bold): Để tạo font
 - Input: Size chữ, có in đậm True ngược lại None
 - Output: Trả về font cùng các thuộc tính sau khi khởi tạo
 - Hàm def randomBot(x): Dùng để random bot ngoại trừ nhân vật của người chơi đã chọn
 - Input: Index của nhân vật mà P1 đã chọn
 - Output: Trả về index của bot ngẫu nhiên
 - Hàm def choose_background(screen, main_game, selected_character, mode, character_labels): Vẽ màn hình chọn địa hình thi đấu sau khi đã chọn nhân vật
 - Input: Màn hình game, hàm main_game, danh sách 2 nhân vật đã chọn, danh sách tên 2 nhân vật
 - Output: Không có
 - Hàm def play(screen, main_game): Hàm vẽ màn hình chọn các options: Play P1vsP2, P1vsBot, Back to main menu
 - Input: Màn hình chính game, hàm main_game

- Output: Không có
- Hàm def instructions(screen,main_game): Vẽ màn hình hướng dẫn
 - Input: Màn hình game, hàm main_game
 - Output: Không có
- Hàm def choose_P1_P2(screen, main_game): Vẽ màn hình chọn nhân vật cho chế độ P1 vs P2
 - Input: Màn hình game, hàm main_game
 - Output: Không có
- Hàm def choose_P1_BOT(screen, main_game): Vẽ màn hình chọn nhân vật cho chế độ P1 vs Bot
 - Input: Màn hình game, hàm main_game
 - Output: Không có
- Hàm def main_menu(screen,main_game): Vẽ màn hình main menu sau khi hết opening gồm các options: Play, Instructions, Music, Quit
 - Input: Màn hình game, hàm main_game
 - Output: Không có
- Draw.py
 - Module dùng để vẽ thanh máu, nhân vật, shuriken và chữ cho game, gồm 3 hàm
 - Hàm def draw_health_bar(health, x, y, screen): Dùng để gắn thanh máu 2 nhân vật lên màn hình
 - Input: Máu nhân vật, tọa độ x,y của thanh máu trên màn hình game, màn hình chính của game
 - Output: Không có
 - Hàm def drawGameWithImage(screen, player, enemy, bg, shurikens1, shurikens2): Vẽ nhân vật, vẽ thanh máu, vẽ shurikens
 - Input: Máu nhân vật, tọa độ x,y của thanh máu trên màn hình game, màn hình chính của game
 - Output: Không có
 - Hàm def draw_text(text, font, text_color, bg_color, x, y, screen): Dùng để vẽ chữ
 - Input: Chữ cần vẽ, loại font, màu, màu phông chữ, tọa độ x,y, màn hình chính
 - Output: Không có
- main.py
 - Đây là file chính của game gọi các module khác để tạo nên 1 game hoàn chỉnh, gồm 5 hàm
 - Hàm def intro(): Dùng để gọi opening lúc vừa vào game và gọi menu chính
 - Input: Không có

- Output: Không có
- Hàm def handle_input(p1, p2, ai, throwSpeed1, throwSpeed2, shurikens1, shurikens2): Chủ yếu gắn danh sách các input keys và danh sách shurikens vào P1 và P2, Nếu P2 là AI thì keys sẽ dựa vào random các case bên class Aiplayer
 - Input: P1, P2, Ai, Tốc độ ném phi tiêu P1, Tốc độ ném phi tiêu P2, Danh sách shuriken của P1, Danh sách shuriken của P2
 - Output: Không có
- Hàm def unpause(): Xử lý ngưng pause:
 - Input: Không có
 - Output: Không có
- Hàm def ispause(): Xử lý pause:
 - Input: Không có
 - Output: Không có
- Hàm def main_game(characters, mode, character_labels, background): Chủ yếu gọi phần chính của game: gồm khởi tạo nhân vật, hoạt ảnh đánh nhau, chuyển động... Đây là hàm chủ đạo liên kết các module khác nhau lại thành 1 game hoàn chỉnh (hàm quan trọng)
 - Input: Danh sách nhân vật tham gia trận chiến, chế độ chơi, danh sách tên nhân vật, địa hình thi đấu
 - Output: Không có
- weapons.py
 - Đây là module chứa class Weapons dùng để xử lý các vấn đề liên quan tới shurikens như: vẽ, hitbox shuriken gồm 2 phương thức
 - Phương thức khởi tạo def __init__(self, X = 50, Y = 400, W = 40, H = 60, F = 1): Dùng để khởi tạo shurikens
 - Input: Tọa độ x,y, Kích thước WxH của shuriken, phía mà shuriken sẽ bay tới
 - Output: Không có
 - Phương thức def draw(self, DISPLAYSURF): Dùng để vẽ shuriken.
 - Input: Màn hình chính của game
 - Output: Không có
- Aiplayer.py
 - Đây là module chứa class Player (AI) dùng để xử lý các vấn đề liên quan tới Bot, gồm 10 phương thức
 - Phương thức khởi tạo def __init__(self, input_dict, player1, player2, ai_scheme = 'heuristic'): Dùng để khởi tạo Bot:
 - Input: Danh sách phím nhập, P1, P2 (bot), giải thuật sử dụng là heuristic

- Output: Không có
- Phương thức `def heuristics(self)`: Giúp random input các keys nhập từ bàn phím dựa vào các case từ phương thức `choose_heuristic`, nếu key trả về thì value tăng lên 1 (Bot sẽ hoạt động dựa vào key có value bằng 1)
 - Input: Không có
 - Output: Trả về danh sách các input keys với 1 key có value bằng 1
- Phương thức `def choose_heuristic(self)`: Trả về sequence dựa vào các case từ khoảng cách
 - Input: Không có
 - Output: Trả về sequence chứa các danh sách keys input tùy vào các case ở bên dưới
- Phương thức `def is_left(self)`: Phương thức giúp kiểm tra bot có bên trái P1 hay không?
 - Input: Không có
 - Output: Nếu tọa độ x (tính từ giữa) của bot > P1 thì return True, ngược lại return False
- Phương thức `def is_right(self)`: Kiểm tra bot có bên phải P1 hay không?
 - Input: Không có
 - Output: Nếu tọa độ x (tính từ giữa) của bot < P1 thì return True, ngược lại return False
- Phương thức `def is_far(self, distance = 160)`: Kiểm tra bot có đang ở xa người chơi hay không?
 - Input: Khoảng cách để xác định
 - Output: Nếu tọa độ x (tính từ giữa) của P1 - Bot mà lớn hơn hàm `scale` (tính toán khoảng cách của nhân vật P1 dựa trên tỷ lệ màn hình thực tế và màn hình game) thì return True, ngược lại return False
- Phương thức `def is_medium(self, low_distance = 100, high_distance = 160)`: kiểm tra bot có đang ở không xa không gần người chơi hay không?
 - Input: Khoảng cách xa để xác định, khoảng cách gần để xác định
 - Output: Nếu không xa, không gần thì return True, ngược lại return False
- Phương thức `def is_close(self, distance = 100)`: Kiểm tra bot có đang ở gần người chơi hay không?
 - Input: Khoảng cách để xác định
 - Output: Nếu tọa độ x (tính từ giữa) của P1 - Bot mà bé hơn hàm `scale` (tính toán khoảng cách của nhân vật P1 dựa

- trên tỷ lệ màn hình thực tế và màn hình game) thì return True, ngược lại return False
- Phương thức `def is_on_top(self)`: Kiểm tra bot có đang ở phía trên P1 hay không?
 - Input: Không có
 - Output: Nếu tọa độ y (tính từ giữa) của Bot bé hơn P1 và P1 vs Bot gần nhau với khoảng cách 20 sẽ return True, ngược lại False
- Phương thức `def is_under(self)`: Kiểm tra bot có đang ở phía dưới P1 hay không?
 - Input: Không có
 - Output: Nếu tọa độ y (tính từ giữa) của Bot lớn hơn P1 và P1 vs Bot gần nhau với khoảng cách 50 sẽ return True, ngược lại False
- `player.py`
 - Đây là module chứa class Player (người chơi) dùng để xử lý các vấn đề liên quan tới người chơi, gồm 10 phương thức
 - Phương thức khởi tạo `def __init__(self, PLAYER, X, Y, FLIP, DATA, SPRITES_SHEET, ANIMATION_STEPS, AI, NAME, ICON)`: Dùng để khởi tạo nhân vật
 - Input: Loại player, tọa độ, hướng quay, kích thước, phần bù trừ tọa độ, ảnh, số bước di chuyển của mỗi action, có phải bot hay không, tên nhân vật, icon nhân vật
 - Output: Không có
 - Phương thức `def draw(self, DISPLAYSURF)`: Dùng vẽ icon, vẽ ảnh nhân vật
 - Input: Màn hình chính
 - Output: Không có
 - Phương thức `def move(self, DISPLAYSURF, target, round_over, keys)`: Dùng để xử lý, cập nhật hoạt ảnh của nhân vật
 - Input: Màn hình chính, target của nhân vật, cờ kết thúc 1 round, các keys input
 - Output: Không có
 - Phương thức `def shuriken(self, throwSpeed, shurikens, target, keys)`: Dùng để xử lý, cập nhật liên quan tới hoạt ảnh phóng shuriken
 - Input: Tốc độ phóng, danh sách các shurikens, đối thủ, danh sách keys người dùng nhập
 - Output: Không có

- Phương thức `def loading(self, sprites_sheet, animation_steps)`: Xử lý, cắt sprites sheet thành những ảnh nhỏ liên tiếp đưa vào trong một danh sách
 - Input: `sprites_sheet`, các bước di hoạt động của 1 action
 - Output: Trả lại danh sách ảnh đã cắt
- Phương thức `def scale(self, val)`: Dùng để tỷ lệ hóa giá trị đầu vào dựa trên tỷ lệ màn hình hiện tại và tỷ lệ màn hình mong muốn
 - Input: Giá trị cần tỷ lệ
 - Output: Trả lại tỉ lệ đã tính toán
- Phương thức `def attack(self, DISPLAYSURF, target, x, y)`: Xử lý hitbox đánh nhau, trừ máu
 - Input: Target, tọa độ x,y
 - Output: Không có
- Phương thức `def update (self)`: Cập nhật hoạt ảnh: chết, chạy, ra chiêu, trúng chiêu, đứng,....
 - Input: Target, tọa độ x,y
 - Output: Không có
- Phương thức `def update_action(self, new_action)`: Cập nhật action
 - Input: Action mới
 - Output: Không có
- Phương thức `def Hit(self)`: Xử lý trừ máu khi dính shuriken
 - Input: Không có
 - Output: Không có
- `pyvidplayer.py` (module bên ngoài hỗ trợ việc phát video ở đầu game làm opening): <https://github.com/ree1261/pyvidplayer>
- File assets: Chứa thông tin ảnh nhân vật, nhạc nền, âm thanh, video, ảnh nền