**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**NIÊN LUẬN CƠ SỞ CÔNG NGHỆ THÔNG TIN**

**MÃ HP CT271-01, HKI 2023-2024**

**ĐỀ TÀI**

**GAME RẮN SĂN MỒI**

**SNAKE GAME**

**GV hướng dẫn: Sinh viên thực hiện:**

**PGS.TS Trần Cao Đệ Họ tên: Phạm Hửu Trí**

**MSSV: B2014624**

*Cần Thơ, 11/2023*

MỤC LỤC

[MỤC LỤC 2](#_Toc150694144)

[DANH MỤC BẢNG 4](#_Toc150694145)

[DANH MỤC HÌNH ẢNH 5](#_Toc150694146)

[LỜI MỞ ĐẦU 6](#_Toc150694147)

[PHẦN TỔNG QUAN 7](#_Toc150694148)

[I- MỤC TIÊU ĐỀ TÀI: 7](#_Toc150694149)

[II- ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU: 7](#_Toc150694150)

[III- PHƯƠNG PHÁP NGHIÊN CỨU VÀ PHÂN BỐ NỘI DUNG: 7](#_Toc150694151)

[PHẦN NỘI DUNG 8](#_Toc150694152)

[I- ĐẶC TẢ YÊU CẦU: 8](#_Toc150694153)

[1- Chức năng chính của game “Snake”: 8](#_Toc150694154)

[2- Các chức năng phụ trợ: 9](#_Toc150694155)

[3- Yêu cầu về giao diện hình ảnh và âm thanh: 9](#_Toc150694156)

[II. THIẾT KẾ GIẢI PHÁP: 9](#_Toc150694157)

[1- Kiến trúc chính của phần mềm: 9](#_Toc150694158)

[2- Các chức năng phụ trợ và giao diện hỗ trợ các chức năng này: 11](#_Toc150694159)

[3- Giao diện và chức năng trò chơi chính: 14](#_Toc150694160)

[4- Tìm hiểu công nghệ Pygame: 19](#_Toc150694161)

[III- CÀI ĐẶT PHẦN MỀM: 20](#_Toc150694162)

[1- Thiết kế bản đồ, “Rắn” và “Mồi”: 20](#_Toc150694163)

[2- Thiết kế kiểm tra va chạm, kiểm tra sự kiện eat, tạo food ngẫu nhiên và chuyển động của rắn: 21](#_Toc150694164)

[3- Cài đặt hiển thị: 23](#_Toc150694165)

[4- Kiểm tra, xử lý sự kiện: 24](#_Toc150694166)

[IV- KIỂM THỬ VÀ ĐÁNH GIÁ: 27](#_Toc150694167)

[1- Thông tin cấu hình máy tính: 27](#_Toc150694168)

[2- Kiểm thử chức năng và đánh giá: 27](#_Toc150694169)

[PHẦN TỔNG KẾT 28](#_Toc150694170)

[I- KẾT QUẢ ĐẠT ĐƯỢC: 28](#_Toc150694171)

[1- Ưu nhược điểm của phần mềm: 28](#_Toc150694172)

[2- Kiến thức thu được: 28](#_Toc150694173)

[3- Phân phối được phần mềm: 28](#_Toc150694174)

[II- HƯỚNG PHÁT TRIỂN: 28](#_Toc150694175)

[III- HƯỚNG DẪN CÀI ĐẶT: 28](#_Toc150694176)

DANH MỤC BẢNG

Bảng. Cấu hình máy tính...........................................................................................27

Bảng. Kết quả đanh giá kiểm thử..............................................................................27

DANH MỤC HÌNH ẢNH

[Hình. Trục tọa độ của bản đồ 8](#_Toc150694831)

[Hình. Sơ đồ kịch bản chuyển đổi giao diện 9](#_Toc150694832)

[Hình. Chú thích Sơ đồ kịch bản chuyển đổi giao diện 10](#_Toc150694833)

[Hình. Sơ đồ Cây thư mục 10](#_Toc150694834)

[Hình.Giao diện MAIN\_MENU 11](#_Toc150694835)

[Hình. Các chức năng được hỗ trợ trong giao diện MAIN\_MENU 11](#_Toc150694836)

[Hình. Nút option khi rê chuột vào 12](#_Toc150694837)

[Hình.Giao diện OPTION 13](#_Toc150694838)

[Hình. Giao diện trò chơi chính mode option 14](#_Toc150694839)

[Hình. Giao diện trò chơi chính mode level 15](#_Toc150694840)

[Hình. Giao diện GAME\_OVER của mode level 16](#_Toc150694841)

[Hình. Giao diện PAUSE của mode level 17](#_Toc150694842)

[Hình. Giao diện CONGRATULATION 18](#_Toc150694843)

[Hình. Mã khởi tạo bản đồ 20](#_Toc150694844)

[Hình. Hàm kiểm tra va chạm 21](#_Toc150694845)

[Hình. Hàm kiểm tra ăn mồi 21](#_Toc150694846)

[Hình. Random food 22](#_Toc150694847)

[Hình. Di chuyển rắn lên trên 22](#_Toc150694848)

[Hình. Hàm eat() 22](#_Toc150694849)

[Hình. Tạo các Surface chính 23](#_Toc150694850)

[Hình. Lớp hỗ trợ hiển thị MAIN\_MENU 23](#_Toc150694851)

[Hình. Hàm hiển thị MAIN\_MENU 23](#_Toc150694852)

[Hình. Hàm mouse\_Move\_In 24](#_Toc150694853)

[Hình. Vòng lặp game tóm gọn 24](#_Toc150694854)

[Hình. Mã HightLight nút khi rê chuột vào trên MAIN\_MENU 25](#_Toc150694855)

[Hình. Mã xử lý sự kiện click chuột trên các nút trong MAIN\_MENU 25](#_Toc150694856)

[Hình. Xử lý sự kiện ấn nút mũi tên lên trong GAME\_PLAY 26](#_Toc150694857)

[Hình. Xử lý sự kiện auto\_move() 26](#_Toc150694858)

[Hình. Trick tính điểm 26](#_Toc150694859)

LỜI MỞ ĐẦU

Xã hội càng phát triển, người ta càng ngày càng bận rộn, thời gian nghỉ ngơi giữa những giờ làm việc ngày càng bị rút ngắn. Trong bối cảnh đó, việc chơi những game đơn giản, tốn ít thời gian để thư giản, giải trí sau những giờ làm việc căn thẳng là một lựa chọn không tồi. Nhắc đến những trò chơi đơn giản, không thể quên đi trò chơi huyền thoại “Snake”, mà người Việt hay gọi là “Rắn săn mồi”. Xuất hiện lần đầu trên chiếc điện thoại Nokia 6110 vào năm 1997, game “Snake” thời sơ khai được hoạt động trên màn hình đen trắng, tạo hình con rắn và các vật phẩm trong game đều ở dạng các khối vuông 8 bit. Tuy đơn giản như vậy, nhưng những phiên bản của game “Snake” đã thành công một cách vang dội, nó đã được cài đặt sẵn như một tiêu chuẩn cần phải có trên mọi chiếc điện thoại. Cho đến năm 2005, “Snake” đã phát hành được hơn 350 triệu bản trên toàn thế giới. Trong thời buổi CNTT ngày càng phát triển, càng có nhiều công nghệ được con người tạo ra để lập trình phần mềm, với những giao diện đồ họa được thiết kế đẹp hơn, bắt mắt hơn, trong đó có Pygame. Pygame là một bộ module [Python](https://vi.wikipedia.org/wiki/Python_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) [đa nền tảng](https://vi.wikipedia.org/wiki/%C4%90a_n%E1%BB%81n_t%E1%BA%A3ng) được thiết kế để viết [trò chơi điện tử](https://vi.wikipedia.org/wiki/Tr%C3%B2_ch%C6%A1i_%C4%91i%E1%BB%87n_t%E1%BB%AD). Nó bao gồm đồ họa máy tính và thư viện âm thanh được thiết kế để sử dụng với ngôn ngữ lập trình [Python](https://vi.wikipedia.org/wiki/Python_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)). Niên luận này nghiên cứu việc sử dụng Pygame để phát triển một phiên bản của game “Snake” với đồ họa đẹp hơn so với phiên bản gốc, với tiêu chí là đơn giản, dễ sử dụng và thân thiện với người dùng.

PHẦN TỔNG QUAN

# MỤC TIÊU ĐỀ TÀI:

**Phát triển phần mềm game “Snake”**, đáp ứng được nhu cầu giải trí của khách hàng trong thời gian nghỉ ngắn giữa giờ làm việc.

Nhu cầu giải trí của một bộ phận khách hàng ngày càng tăng cao, nhưng thời gian của họ ngày càng bị rút ngắn. Đồng thời, trước sự cạnh tranh giữa các sản phẩm phần mềm ngày càng khốc liệt, lập trình viên cần cải thiện sản phẩm của mình, hướng tới sự đơn giản, phân phối nhanh, dễ sử dụng, nhưng phải đáp ứng về mặt hình ảnh, chức năng. Do đó, những game nhỏ như “Snake” cần phải đáp ứng những yêu cầu này.

Vấn đề đặt ra là việc cân bằng giữa sự phân phối nhanh và sự hoàn thiện của sản phẩm, do đó, việc thực hiện một game dễ dàng mở rộng là một phương pháp hữu hiệu.

Vấn đề trọng tâm: game “Snake” phải đáp ứng

-Có giao diện đồ họa đẹp

-Kết hợp hình ảnh và âm thanh

-Giao diện dễ sử dụng

-Dễ dàng mở rộng, nâng cấp phiên bản

-Dễ phân phối và cài đặt

# ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU:

**Game “Snake”:**

Nghiên cứu yêu cầu chức năng và đưa ra giải pháp, giải thuật để vận hành một game “Snake”, từ đó áp dụng vào sản phẩm.

**Ngôn ngữ lập trình Python và module Pygame:**

Nghiên cứu cách vận hành của một chương trình Pygame, các công cụ hỗ trợ về hình ảnh và âm thanh của module Pygame, để phát triển, hoàn thiện sản phẩm.

# PHƯƠNG PHÁP NGHIÊN CỨU VÀ PHÂN BỐ NỘI DUNG:

**Phương pháp nghiên cứu:**

Sử dụng tài liệu có sẵn được tìm kiếm trên Internet

**Phân bố nội dung công việc:**

-Đăng ký đề tài: tuần 1-2

-Nghiên cứu, mô tả yêu cầu, tìm hiểu công nghệ: tuần 2-4

-Cài đặt phần mềm demo: tuần 4-8

-Testing và fix lỗi cho bản demo đầu tiên: tuần 8-10

-Tiếp tục hoàn thiện sản phẩm, viết báo cáo version 1: tuần 10-14

-Testing và hoàn thiện báo cáo cuối cùng: các tuần còn lại

PHẦN NỘI DUNG

# ĐẶC TẢ YÊU CẦU:

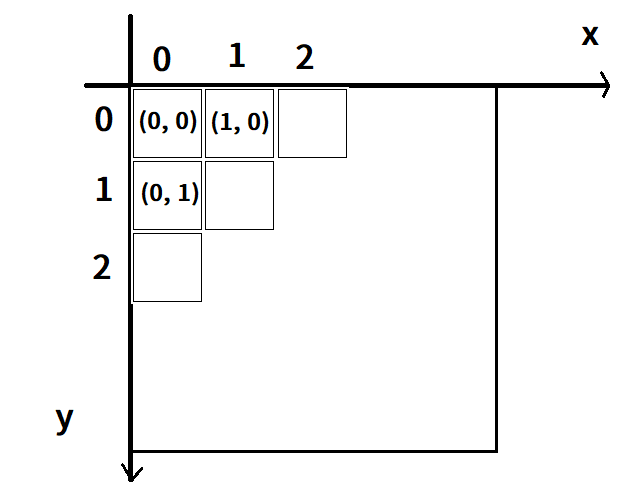
## Chức năng chính của game “Snake”:

**Bản đồ:**

Bản đồ có dạng lưới ô vuông chiều ngang là x, chiều dọc là y, tại các vị trí ô, sẽ có 1 trong những đối tượng “Rắn”, “Mồi” hoặc “Tường”.

Bản đồ khác nhau sẽ có cách bố trí “Tường” khác nhau và cố định theo từng bản đồ.

Trục tọa độ của bản đồ như sau:



Hình. Trục tọa độ của bản đồ

**Đối tượng “Rắn”:**

“Rắn” là một tập hợp những ô vuông có thể di chuyển nối đuôi (ô sau di chuyển vào vị trí của ô trước) nhau, theo 4 hướng: up, down, left, right.

“Rắn” được điều khiển hướng di chuyển bằng phím điều hướng mũi tên trên bàn phím. Sau một khoảng thời gian (tùy theo độ khó) không được điều hướng, rắn sẽ tự động di chuyển (auto\_move()) về hướng đầu rắn đang hướng đến.

**Đối tượng “Mồi”:**

“Mồi” là một vật hình tròn được hiện ngẫu nhiên trên những vị trí không có “Rắn” hoặc “Tường”, mục tiêu trò chơi là điều khiển cho “Rắn” ăn “Mồi” này.

Sau khi “Mồi” bị ăn, “Mồi” mới sẽ hiển thị ngẫu nhiên tại vị trí khác.

**Đối tượng “Tường”:**

“Tường” là những ô vuông, cố định trên bản đồ.

**Ăn mồi:**

Khi phần đầu “Rắn” va chạm vào mồi, “Rắn” dài ra thêm 1 ô, điểm được cộng thêm.

**Game over:**

Khi phần đầu “Rắn” va chạm vào tường hoặc chính thân rắn, màn chơi sẽ kết thúc

## Các chức năng phụ trợ:

-Chọn hình ảnh của “Rắn” hiển thị: dễ dàng thay đổi hình ảnh hiển thị của “Rắn” cho phù hợp với sở thích của người chơi.

-Cài đặt phát nhạc, tắt nhạc: Bật tắt nhạc dễ dàng, có thể thay đổi âm lượng, hoặc thay đổi bản nhạc theo sở thích.

-Chọn chế độ chơi:

+Chế độ option: người chơi chủ động lựa chọn bản đồ muốn chơi, tốc độ auto\_move();

+Chế độ level: hệ thống được thiết kế theo màn chơi, khi người chơi đạt được điểm tối đa của một màn chơi, sẽ tự động được chuyển đến màn chơi kế tiếp (khó hơn).

-Hỗ trợ việc tạm dừng, tiếp tục khi đang chơi.

-Hỗ trợ lưu điểm khi đạt điểm cao trong danh sách “High\_score”.

## Yêu cầu về giao diện hình ảnh và âm thanh:

-Giao diện người dùng thân thiện, dễ sử dụng: có thể dễ dàng chơi bằng 1 tay.

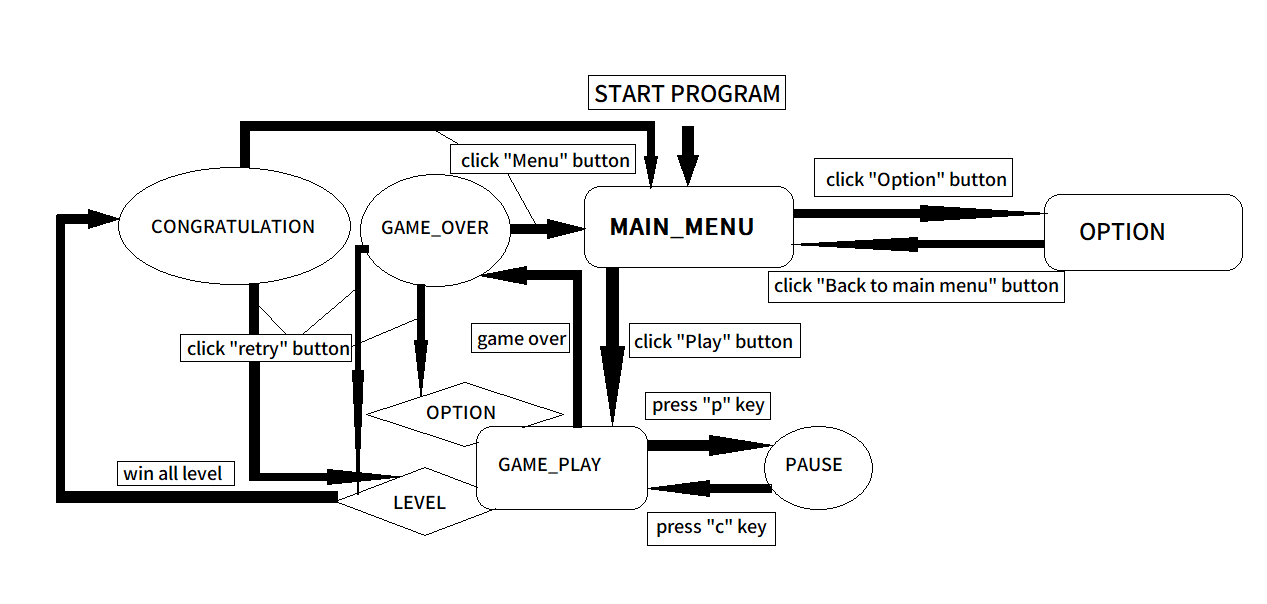
-Hình ảnh hiển thị tươi sáng bắt mắt.

-Âm thanh sống động, âm nhạc nhẹ nhàng thư giãn.

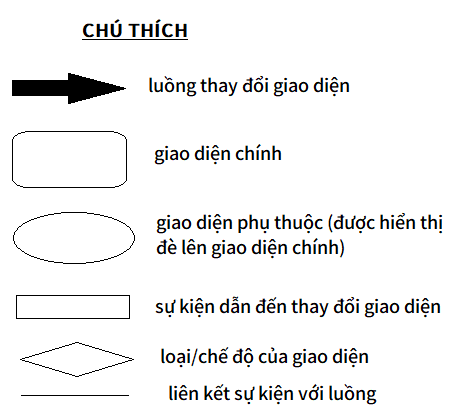
# THIẾT KẾ GIẢI PHÁP:

## Kiến trúc chính của phần mềm:

**Sơ đồ kịch bản chuyển đổi giao diện:**



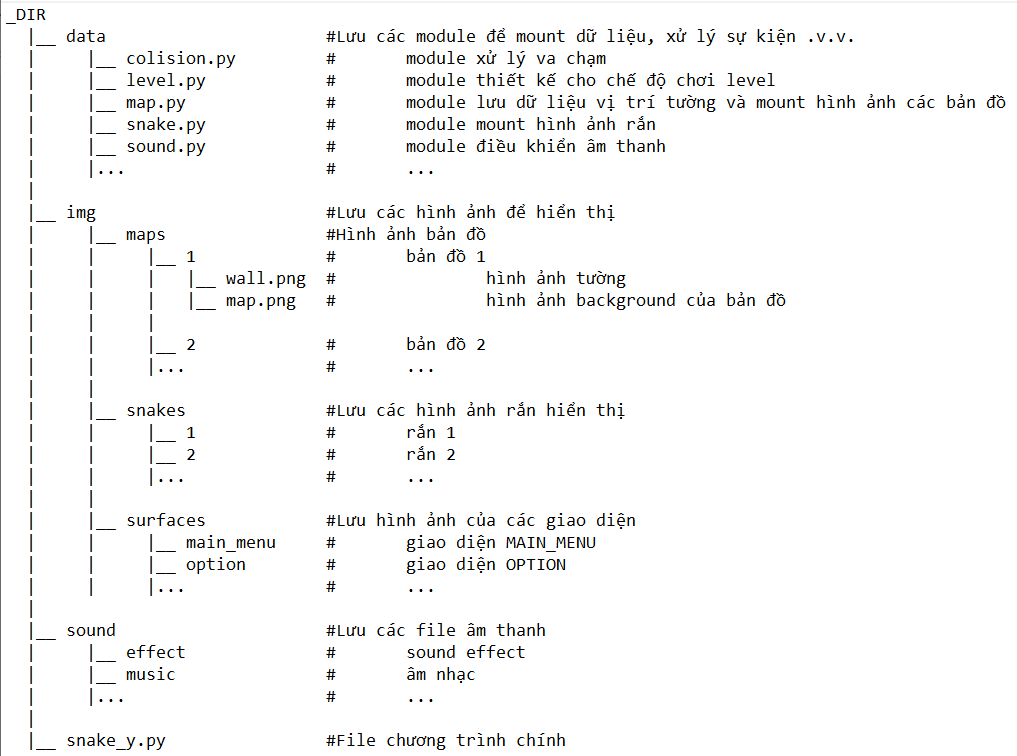
Hình. Sơ đồ kịch bản chuyển đổi giao diện



Hình. Chú thích Sơ đồ kịch bản chuyển đổi giao diện

**Sơ đồ lưu trữ dữ liệu:**

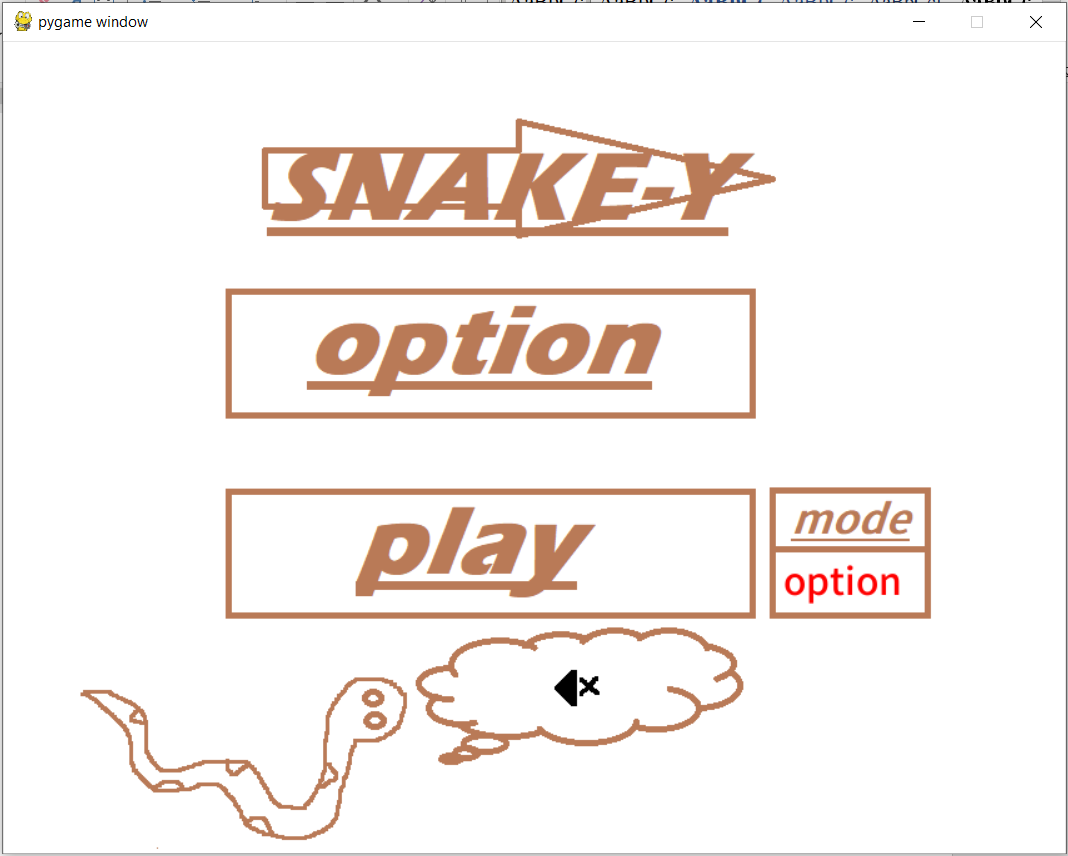
Do dữ liệu tương đối đơn giản, phần mềm được lưu và phân phối dưới dạng một Folder. Cây thư mục:

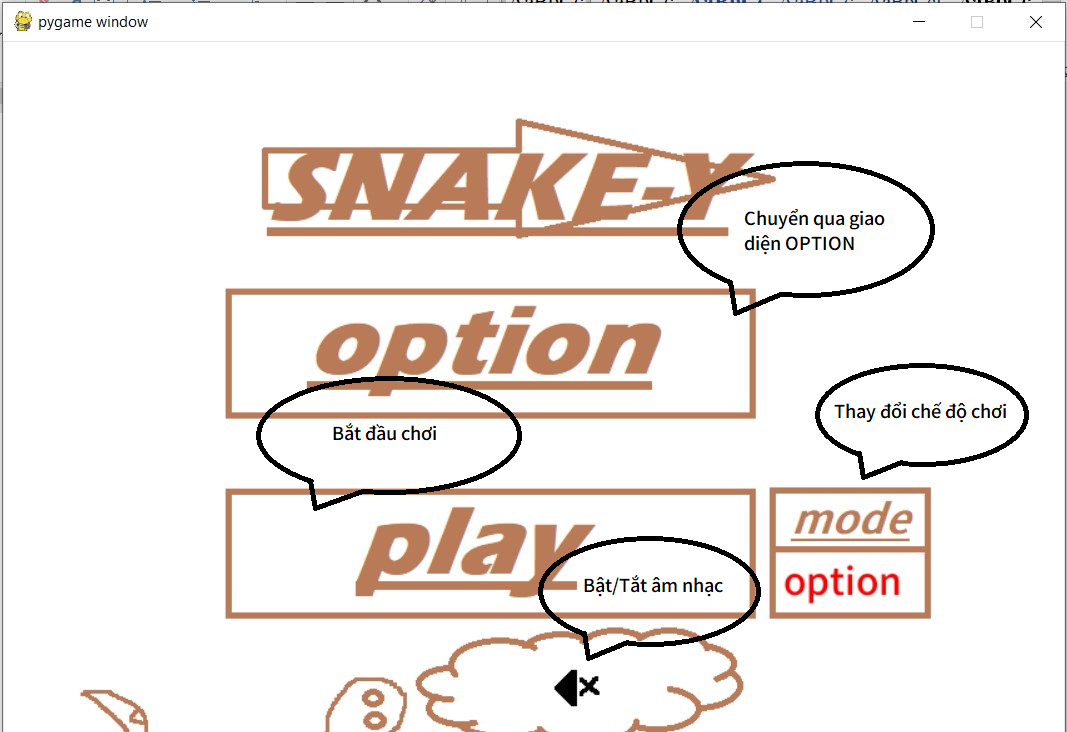


Hình. Sơ đồ Cây thư mục

## Các chức năng phụ trợ và giao diện hỗ trợ các chức năng này:

**Giao diện MAIN\_MENU:**

****

Hình.Giao diện MAIN\_MENU

Hình. Các chức năng được hỗ trợ trong giao diện MAIN\_MENU

Các chức năng hỗ trợ trong giao diện này:

-Click vào ô *mode* để chuyển đổi giữa 2 chế độ chơi option/level trước khi chơi (bằng cách click vào *play*).

-Click vào hình loa  để bật, tắt nhạc nền.

-Chuyển đổi giao diện:

+Ấn vào *option* để chuyển qua giao diện OPTION

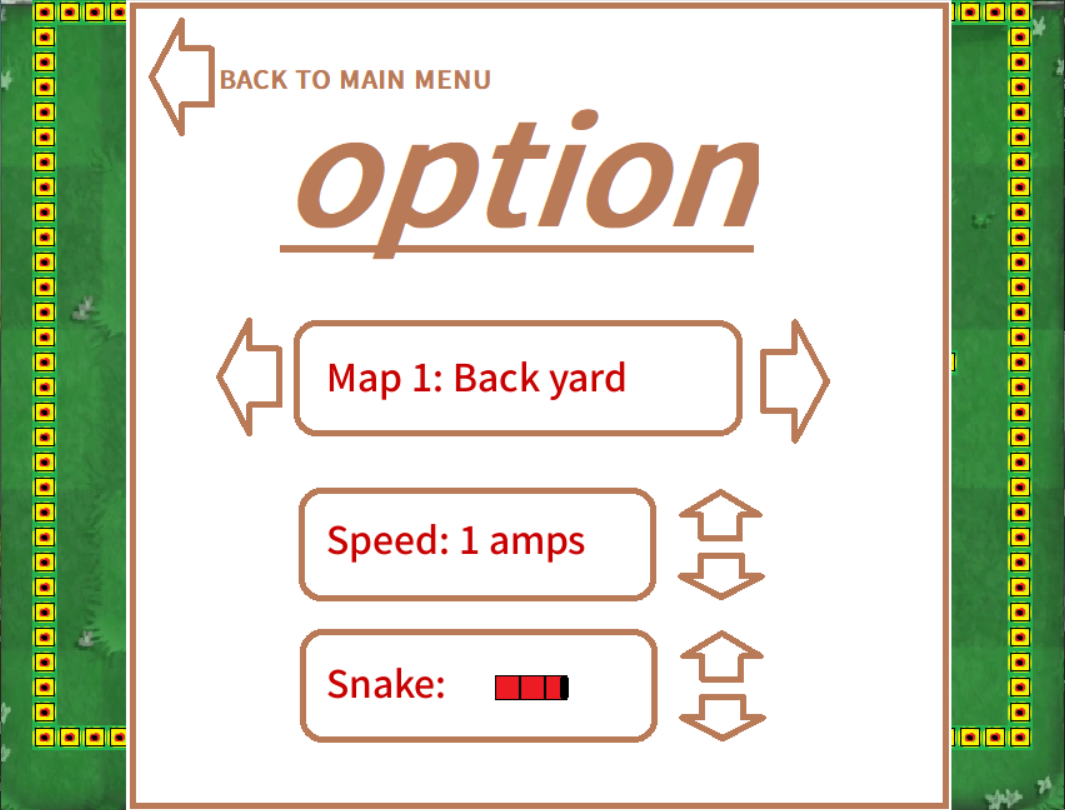
+Ấn vào *play* để bắt đầu màn chơi

Hỗ trợ hiển thị nút: khi rê chuột vào phạm vi *option* hoặc *play* thì chữ trong nút này sẽ đổi màu:



Hình. Nút option khi rê chuột vào

**Giao diện OPTION:**

****

Hình.Giao diện OPTION

Các chức năng được hỗ trợ trong giao diện này:

-Thay đổi bản đồ (mode option): ấn vào  hoặc  ở khung hiển thị Map để thay đổi bản đồ lựa chọn trong chế độ chơi option. Khi thay đổi, bản đồ được chọn hiển thị ở phần nền của bảng chọn.

-Thay đổi độ khó (mode option): ấn vào hai mũi tên  ở khung hiển thị Speed để thay đổi tốc độ tự động di chuyển (amps: auto\_move() per sec), sẽ có 6 tốc độ từ 1-6 amps để phù hợp với FPS 60 (60 chia hết cho từ 1-6).

-Thay đổi hình ảnh Rắn hiển thị (mode option và mode level): ấn vào hai mũi tên  ở khung hiển thị Snake để thay đổi hình ảnh của Rắn được hiển thị khi chơi. Hình ảnh Rắn được chọn sẽ hiển thị trong khung này.

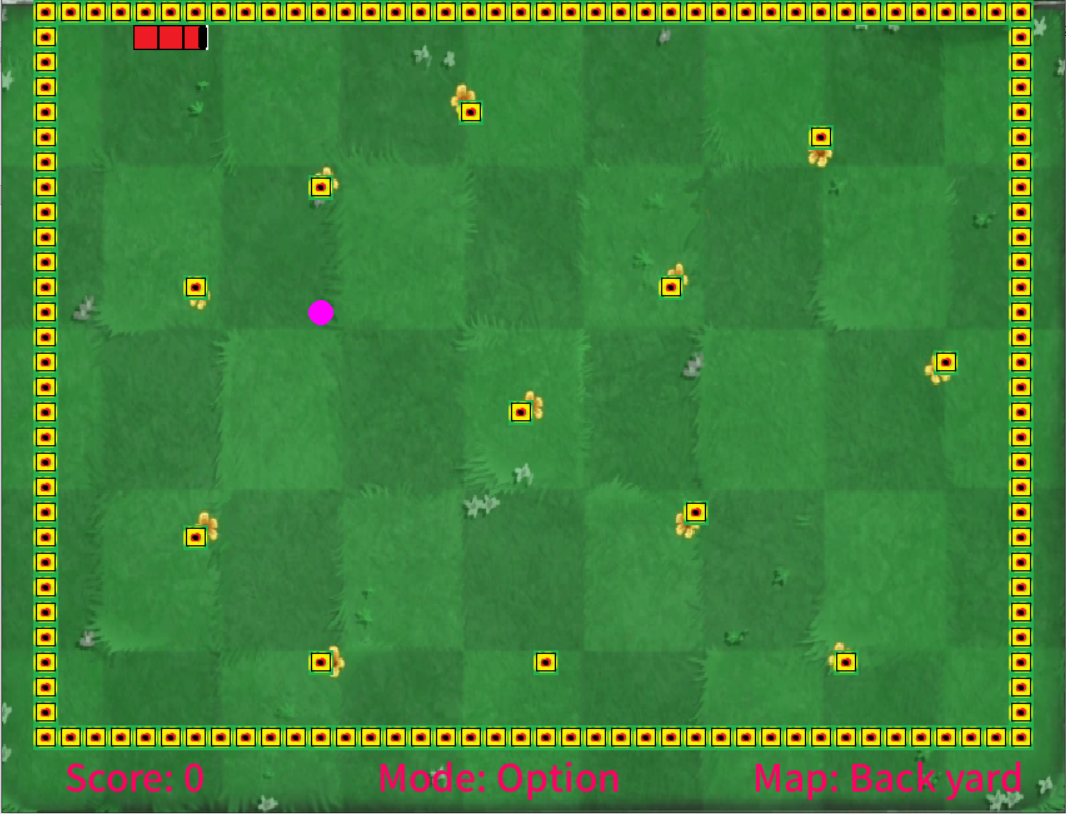
-Chuyển đổi giao diện: Ấn vào  cạnh “BACK TO MAIN MENU” để trở về Giao diện MENU

Hỗ trợ hiển thị nút: các nút mũi tên sẽ được đổi màu giống  khi rê chuột lên

## Giao diện và chức năng trò chơi chính:

**Mode option:**

Hình minh họa: bản đồ và hình ảnh Rắn, cũng như tốc độ có thể thay đổi trong giao diện OPTION trước khi ấn *play*.

****

Hình. Giao diện trò chơi chính mode option

Kịch bản trò chơi:

-Rắn được điều khiển bằng các phím điều hướng mũi tên trên bàn phím.

-Khi ăn được mồi, Rắn sẽ dài ra thêm 1 đơn vị, điểm cũng được tăng lên bằng với tốc độ đã chọn (VD: ở tốc độ 1 amps thì ăn mồi một lần sẽ được +1 điểm, 2 amps thì được +2 điểm, ...), mồi cũng được hiện lên một vị trí ngẫu nhiên mới.

-Khi ấn phím “p” trên bàn phím, trò chơi tạm dừng.

-Khi va chạm vào tường hoặc thân Rắn sẽ bị game over.

-Trò chơi sẽ chạy cho tới khi bị game over.

**Mode level:**

****

Hình. Giao diện trò chơi chính mode level

Kịch bản trò chơi:

-Rắn được điều khiển bằng các phím điều hướng mũi tên trên bàn phím.

-Khi ăn được mồi, Rắn sẽ dài ra thêm 1 đơn vị, điểm cũng được tăng lên bằng với tốc độ hiện tại (VD: ở tốc độ 1 amps thì ăn mồi một lần sẽ được +1 điểm, 2 amps thì được +2 điểm, ...), mồi cũng được hiện lên một vị trí ngẫu nhiên mới.

-Khi đạt đủ một số điểm được quy định sẵn của một level, sẽ được đưa đến level mới (điểm được bảo lưu giữa các level). Bản đồ, tốc độ, số điểm cần đạt ở mỗi level được thiết kế sẵn.

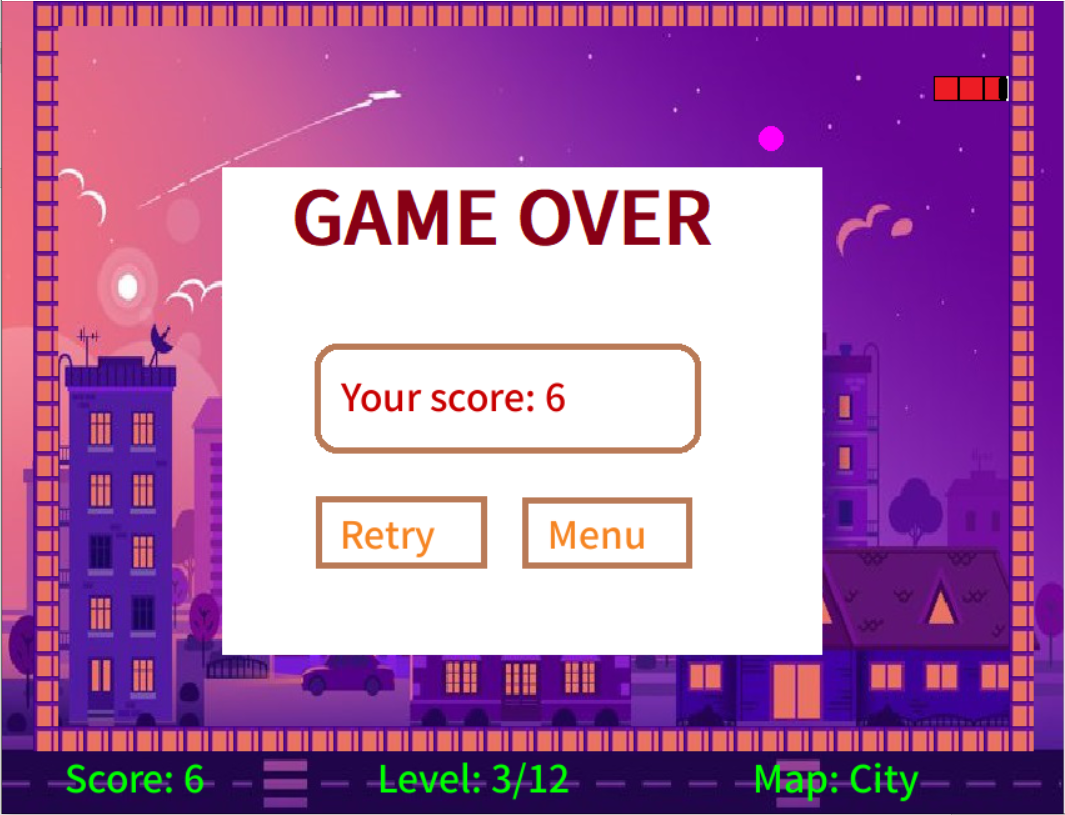
-Khi ấn phím “p” trên bàn phím, trò chơi tạm dừng.

-Khi va chạm vào tường hoặc thân Rắn sẽ bị game over.

-Trò chơi sẽ chạy cho tới khi bị game over hoặc đạt được điểm tối đa của màn chơi cuối cùng được thiết kế (hiển thị giao diện CONGRATULATION).

**Các giao diện phụ thuộc:**

GAME\_OVER (mode option, mode level):



Hình. Giao diện GAME\_OVER của mode level

Các chức năng được hỗ trợ trong giao diện này:

-Lưu điểm nếu diểm cao

-Chuyển đổi giao diện:

+Click vào Retry để bắt đầu lại (từ level 1 nếu là mode level, cùng bản đồ và tốc độ nếu là mode option)

+Click vào Menu để trở về MAIN\_MENU

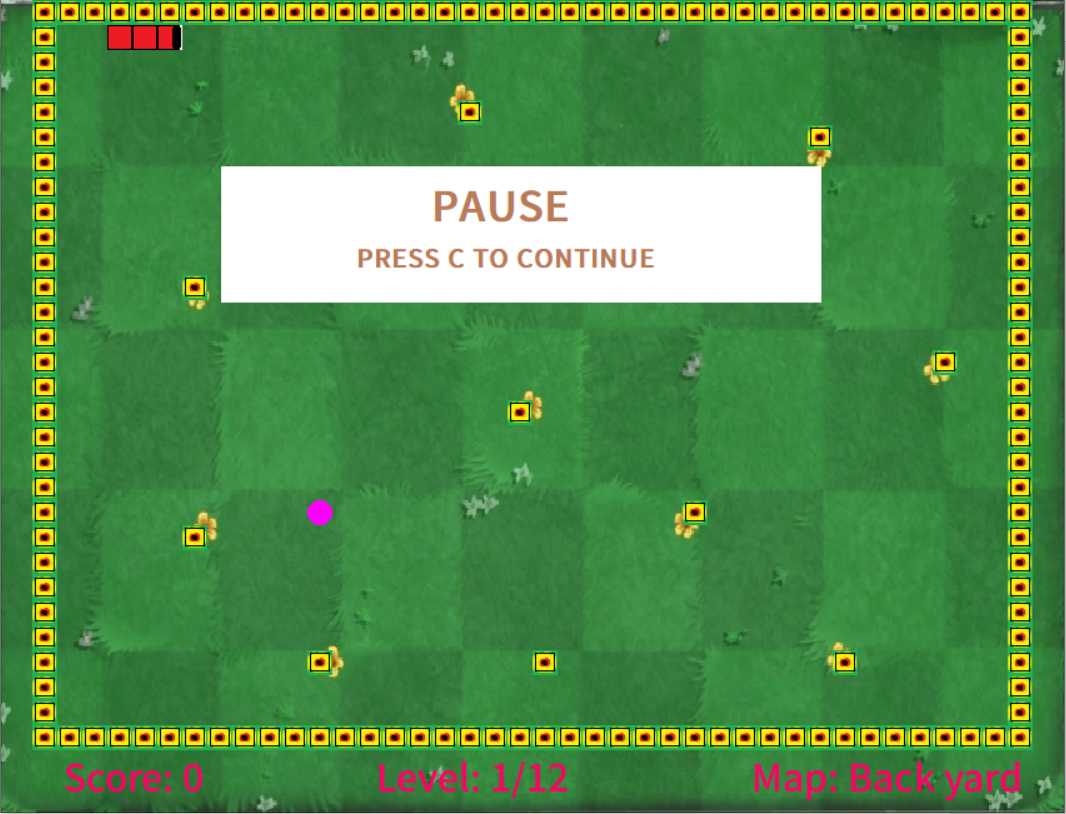
-Hỗ trợ hiển thị nút: chữ Retry và Menu sẽ đổi màu khi rê chuột vào nút này:



*Hình. Nút Retry đổi màu khi rê chuột vào*

PAUSE (mode option, mode level):

Hiển thị khi ấn nút “p” trên bàn phím khi đang chơi:

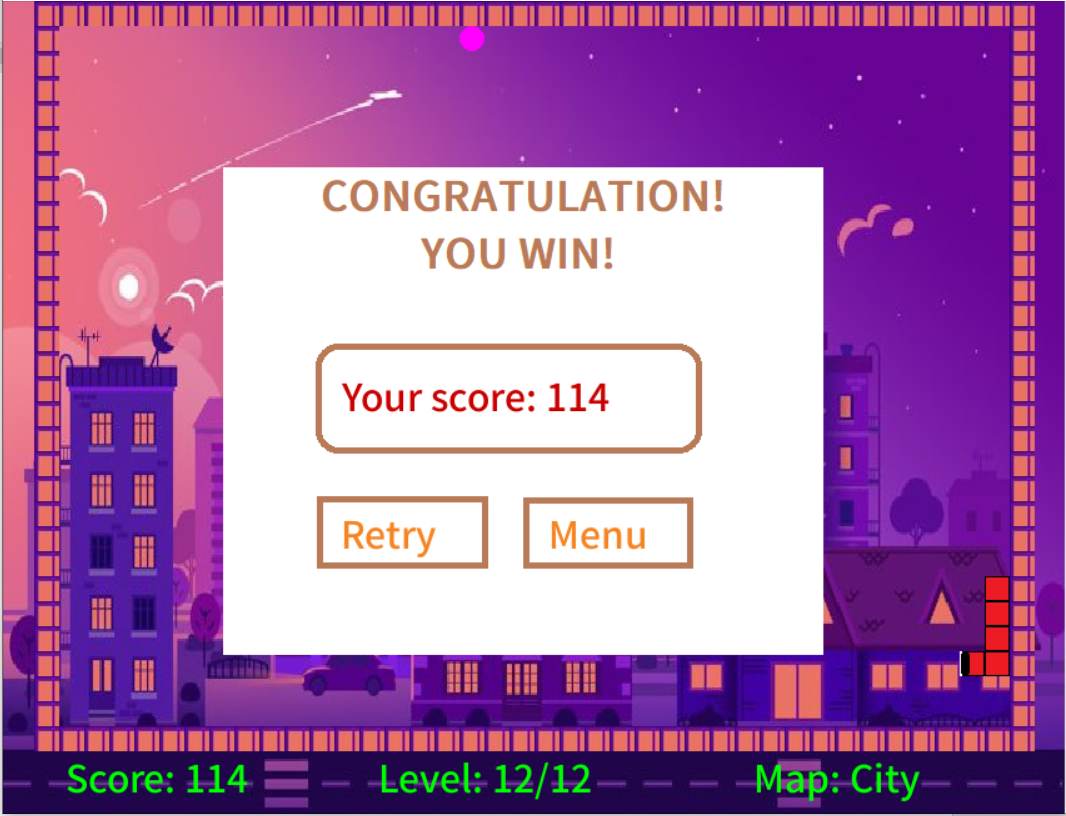


Hình. Giao diện PAUSE của mode level

Các chức năng được hỗ trợ:

-Ấn nút “c” trên bàn phím để tiếp tục chơi

CONGRATULATION (mode level):



Hình. Giao diện CONGRATULATION

Các chức năng được hỗ trợ trong giao diện này:

-Lưu điểm nếu diểm cao

-Chuyển đổi giao diện:

+Click vào Retry để bắt đầu lại từ level 1.

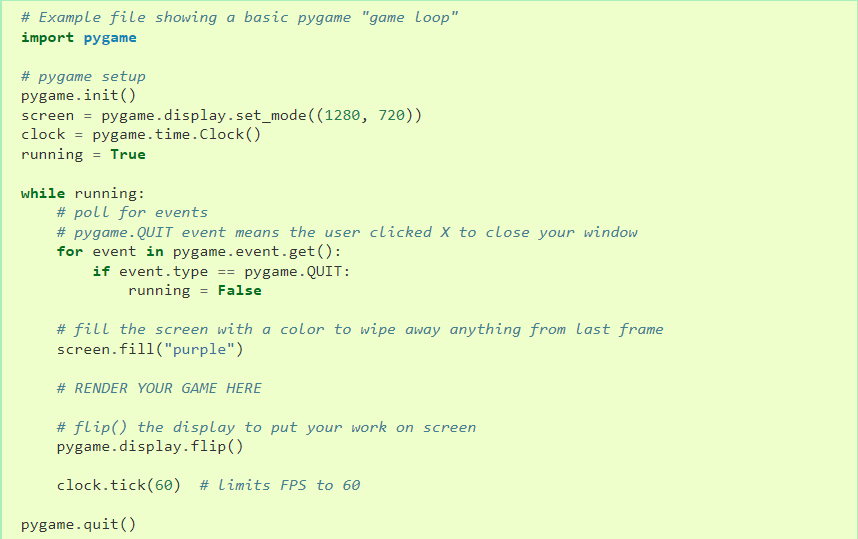
+Click vào Menu để trở về MAIN\_MENU

-Hỗ trợ hiển thị nút: chữ Retry và Menu sẽ đổi màu khi rê chuột vào nút này

## Tìm hiểu công nghệ Pygame:

Pygame là một thư viện mã nguồn mở, miễn phí, được thiết kế để lập trình video game bằng ngôn ngữ Python. Chương trình Pygame chạy trên những “vòng lặp” game. Ở mỗi vòng lặp, Pygame sẽ hiển thị một hình ảnh, những hình ảnh này hiển thị liên tiếp (số hình ảnh hiển thị mỗi giây gần bằng với FPS đã được cài đặt trong chương trình), sẽ tạo thành cảm giác như một video. Ngoài ra, các module phát âm thanh cũng được chạy song song với việc hiển thị vòng lặp game.

Ví dụ chương trình Pygame:



Không như nhiều phần mềm/module hỗ trợ làm game khác, Pygame cung cấp cho người lập trình toàn quyền kiểm soát việc thực thi chương trình của mình, tuy nhiên, người lập trình cũng không cần tìm hiểu mã của Pygame, chỉ cần sử dụng các module hình ảnh, âm thanh của Pygame là hoàn toàn có thể tạo nên một chương trình hoàn thiện.

Đối với module hiển thị hình ảnh:

Pygame hiển thị hình ảnh dưới dạng các đối tượng bề mặt (Surface). Các Surface có nhiều loại: Surface (một Surface thông thường), image.load (ảnh có sẵn), Font.render (chữ), .... Những Surface có thể được đặt chồng lên nhau (bằng lệnh blit()) để hiển thị một hình ảnh trong một vòng lặp. Surface display.set\_mode là một Surface đặt biệt, nó đại diện cho cửa sổ màn hình và luôn nằm bên dưới cùng, một Surface muốn hiển thị phải được blit() lên trên bề mặt này.

Sau khi một Surface bị thay đổi nội dung, nó cần được cập nhật những thay đổi này thành hình ảnh hiển thị với lệnh display.update(Surface) hoặc display.flip() (lệnh này cập nhật toàn bộ Surface, tương đương với lệnh display.update()).

Đối với module phát âm thanh:

Để phát âm thanh,trước tiên cần phải load file âm thanh dưới dạng một đối tượng âm thanh mixer.Sound. Bắt đầu phát âm bằng lệnh <tên đối tượng âm thanh>.play(parameter). Đối số int parameter của lệnh này dùng để thiết lập số lần lặp lại khi phát âm thanh: -1 : phát lặp lại liên tục cho đến khi gặp lệnh <tên đối tượng âm thanh>.stop(); 0 : phát 1 lần; 1 : phát 2 lần;...

Những âm thanh khác nhau có thể phát và ngừng phát độc lập và song song với nhau.

Hoạt động kiểm tra sự kiện (event) ấn chuột, ấn nút...:

Trong mỗi vòng lặp game, mảng event.get() sẽ lưu lại thông tin các sự kiện này. Dùng vòng lặp for kiểm tra các event này và thay đổi hình ảnh, âm thanh,... trong vòng lặp. Vòng lặp for này nên được đặt mỗi khi bắt đầu một vòng lặp game.

# CÀI ĐẶT PHẦN MỀM:

## Thiết kế bản đồ, “Rắn” và “Mồi”:

**Bản đồ và “Tường”:**

Bản đồ được lưu như một mảng 2 chiều (có kích thước 40x30), với tất cả các giá trị khởi đầu là 0 (không có tường tại đây), khi hiển thị, một ô bản đồ có kích thước 20px x 20px.



Hình. Mã khởi tạo bản đồ

“Tường” của mỗi đối tượng lưu trữ thông tin bản đồ (MAP() object) được lưu dưới dạng một mảng, mỗi phần tử trong mảng chứa thông tin vị trí (x từ 0-39,y từ 0-29) của một ô “Tường”. Khi đối tượng MAP() này được chọn, sẽ load thay đổi những vị trí này trong MAP\_2D\_MAZE thành 1 (có tường tại đây).

Ngoài ra, đối tượng MAP() còn lưu thông tin về hình ảnh nền và hình ảnh tường của bản đồ hiển thị.

**“Rắn”:**

Thông tin về những vị trí ô “Rắn” được lưu trong một đối tượng SNAKE() object, dưới dạng một thuộc tính mảng (phần tử cuối trong mảng là đầu rắn), chứa thông tin vị trí tương tự như “Tường” trong trong đối tượng MAP().

Ngoài ra, đối tượng SNAKE() còn lưu thông tin về hình ảnh mà “Rắn” hiển thị.

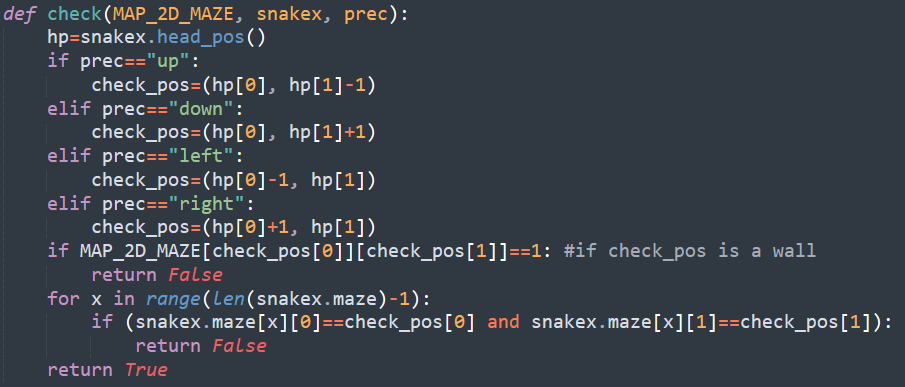
**“Mồi”:**

Mồi được thiết kế hiển thị như một hình tròn có đường kính bằng một ô bản đồ (20px), được lưu vị trí bằng một turple (x, y)

## Thiết kế kiểm tra va chạm, kiểm tra sự kiện eat, tạo food ngẫu nhiên và chuyển động của rắn:

**Kiểm tra va chạm:**

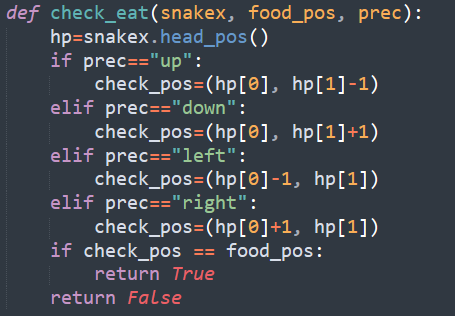
Hàm kiểm tra va chạm:



Hình. Hàm kiểm tra va chạm

Hàm nhận vào MAP\_2D\_MAZE (mảng lưu bảng đồ hiện tại), đối tượng SNAKE(), hướng kiểm tra (so với đầu rắn). Nếu kiểm tra có tường ở hướng kiểm tra, trả về False (bị va chạm); nếu kiểm tra có thân rắn ở hướng kiểm tra, trả về False; nếu không, trả về True (không va chạm).

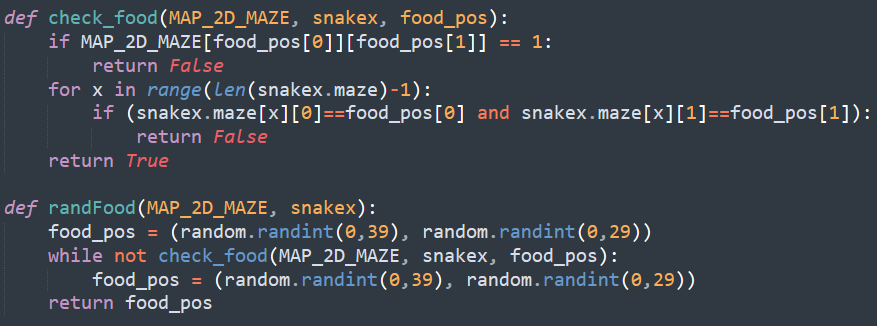
**Kiểm tra ăn mồi:**



Hình. Hàm kiểm tra ăn mồi

Hàm nhận vào đối tượng SNAKE(), vị trí food hiện tại, hướng kiểm tra. Nếu kiểm tra vị trí food trùng với vị trí kiểm tra, trả về True (đã ăn); Nếu không, trả về False (không ăn).

**Random food:**

****

Hình. Random food

Tạo food ngẫu nhiên và kiểm tra vị trí không trùng với bất kỳ ô “Rắn” hoặc “Tường” nào, nếu không thì tạo lại.

**Chuyển động của rắn:**

Di chuyển về một hướng:

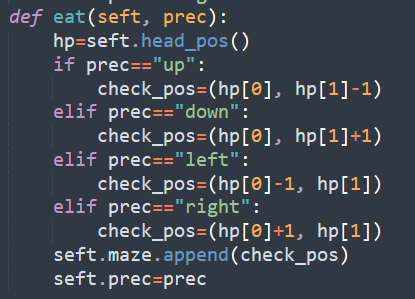


Hình. Di chuyển rắn lên trên

Di chuyển vị trí được lưu trong phần tử từ 0 đến n-2 (áp chót) trong mảng SNAKE.maze thành vị trí của phần tử sau đó. Di chuyển phần tử thứ n-1 (đầu rắn) về hướng cần di chuyển. Thay đổi hướng của rắn (để hỗ trợ cho hiển thị và auto\_move()).

Ăn:

Thêm vào cuối mảng một phần tử tại hướng mà rắn ăn được mồi.



Hình. Hàm eat()

## Cài đặt hiển thị:

**Tạo các Surface** cần thiết và blit() lên screen khi cần thiết:







Hình. Tạo các Surface chính

**Hiển thị MAIN\_MENU:**

****

Hình. Lớp hỗ trợ hiển thị MAIN\_MENU

Tạo một hàm hiển thị:



Hình. Hàm hiển thị MAIN\_MENU

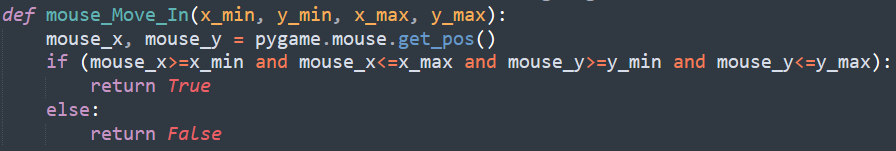
Sử dụng thông tin được lưu trong đối tượng Display\_MAIN\_MENU object, để hiển thị hình ảnh trên Surface MAIN\_MENU. HightLight: đổi màu, khi rê chuột lên nút chỉ định.

**Hiển thị OPTION, GAME\_PLAY:**

Giải pháp tương tự như hiển thị MAIN\_MENU, tạo đối tượng Display\_OPTION\_hander và Display\_GAME\_hander, load vào hàm hiển thị để hiển thị hình ảnh lên Surface.

## Kiểm tra, xử lý sự kiện:

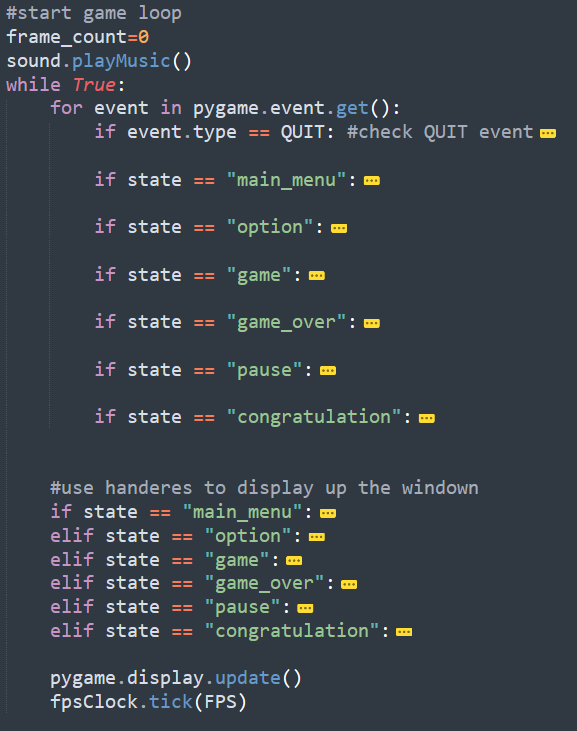
**Hàm mouse\_move\_in:**

****

Hình. Hàm mouse\_Move\_In

Hàm kiểm tra vị trí chuột có nằm bên trên nút hay không, sử dụng để HightLight nút và kiểm tra sự kiện click chuột.

**Vòng lặp game (tóm gọn):**

****

Hình. Vòng lặp game tóm gọn

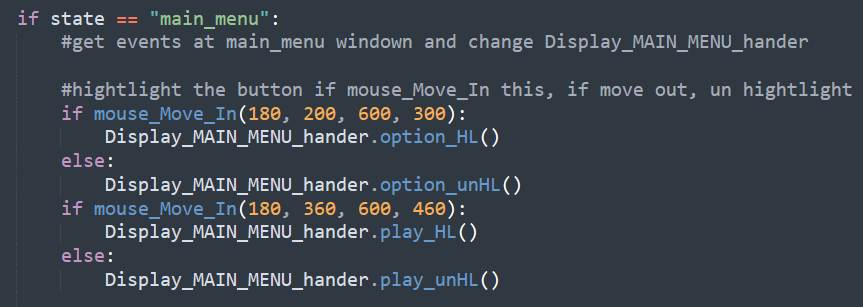
Trong mỗi vòng lặp:

-Kiểm tra và xử lý sự kiện: thay đổi các hander dựa trên các sự kiện bắt được, dựa trên state (giao diện) hiện đang được hiển thị, thay đổi state nếu cần

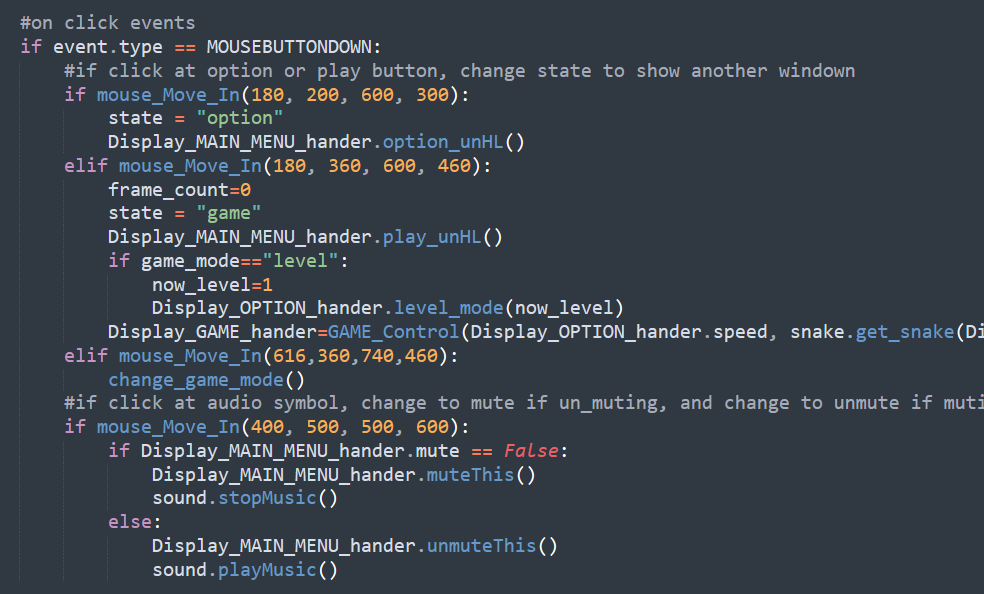
-Dùng hander đã được thay đổi để hiển thị lên các Surface cần thiết, dựa trên state sau khi thay đổi.

-update cửa sổ hiển thị.

**Ví dụ: xử lý sự kiện trong MAIN\_MENU:**

****

Hình. Mã HightLight nút khi rê chuột vào trên MAIN\_MENU



Hình. Mã xử lý sự kiện click chuột trên các nút trong MAIN\_MENU

**Ví dụ: Xử lý sự kiện trong trò chơi chính:**

**Khi ấn điều khiển:**

Nếu hướng điều khiển ngược lại với hướng đầu rắn hiện tại: bỏ qua;

Nếu không:

-Kiểm tra va chạm, nếu va chạm: đổi state thành game\_over

-Kiểm tra ăn, nếu ăn:

+thực hiện ăn

+tạo food ngẫu nhiên

+phát âm thanh khi ăn

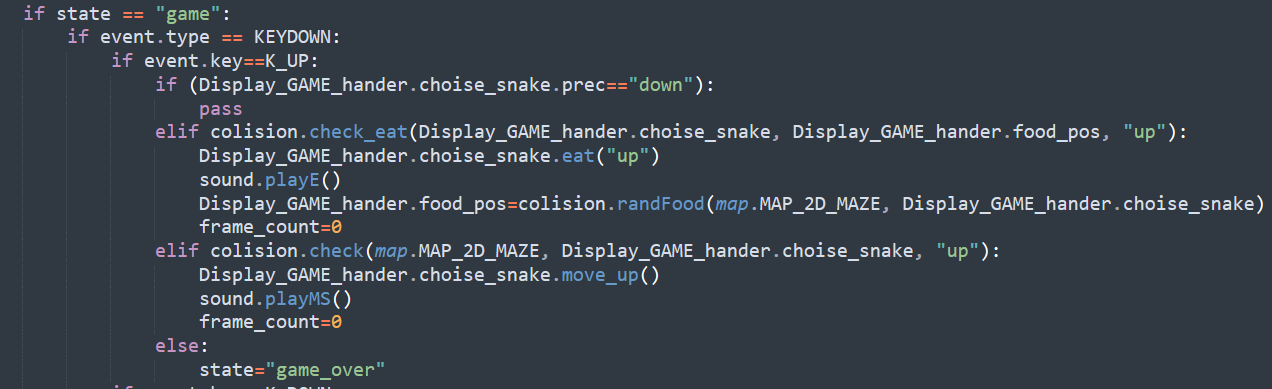
+đặt lại bộ đếm auto\_move

Nếu không ăn:

+thực hiện hành động di chuyển

+phát âm thanh di chuyển

+đặt lại bộ đếm auto move



Hình. Xử lý sự kiện ấn nút mũi tên lên trong GAME\_PLAY

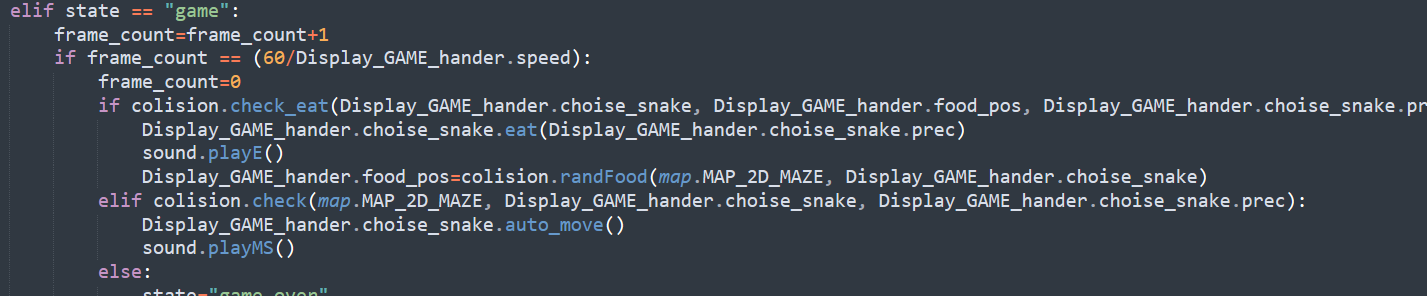
**Khi không ấn điều khiển:**

**-**Tăng bộ đếm auto\_move (frame\_count) lên 1 frame;

**-**Nếu (frame\_count >= 60/speed) thì:

**+**Kiểm tra va chạm và ăn trước auto\_move

+Đặt lại bộ đếm



Hình. Xử lý sự kiện auto\_move()

**Tính điểm:**

Điểm được tính bằng trick:

Điểm = now\_score + speed\*(len(“Rắn”)-3)

Trong mode level, mỗi khi qua bàn mới, ta lại gán now\_score = store\_score (mức lưu điểm gần nhất)



Hình. Trick tính điểm

# KIỂM THỬ VÀ ĐÁNH GIÁ:

## Thông tin cấu hình máy tính:

|  |  |
| --- | --- |
| **OS** | Windown 10 |
| **RAM** | 8 Gb |
| **CPU** | AMD Ryzen 5 5600H with Radeon Graphics 3.30 GHz |
| **GPU** | NVIDIA GeForce GTX 1650 |

*Bảng. Cấu hình máy tính*

## Kiểm thử chức năng và đánh giá:

|  |  |
| --- | --- |
| **Chức năng** | **Đánh giá** |
| Trò chơi chính | Hoạt động tốt, nhưng chưa có cảm giác mượt mà |
| Lựa chọn bản đồ: mode option | Hoạt động tốt |
| Lựa chọn tốc độ rắn | Hoạt động tốt |
| Bật tắt âm nhạc | Hoạt động tốt |
| Lựa chọn hình ảnh rắn | Hoạt động tốt |
| Chuyển đổi giữa các giao diện | Hoạt động tốt |
| Chế độ chơi level | Hoạt động tốt |
| Lưu điểm cao | Hoạt động tốt |
| Khả năng mở rộng | Mở rộng về chiều dọc khá tốt\*, nhưng độ phức tạp khá cao (khó mở rộng thêm chức năng) |

*Bảng. Kết quả đánh giá kiểm thử*

*\*chỉ cần thêm, thay đổi các file hình ảnh/âm thanh, đồng thời thay đổi các module trong data\ là dễ dàng nâng cấp phần mềm*

PHẦN TỔNG KẾT

# KẾT QUẢ ĐẠT ĐƯỢC:

## Ưu nhược điểm của phần mềm:

**Ưu diểm:**

-Phần mềm nhỏ gọn, dễ phân phối.

-Đáp ứng cơ bản đa số những yêu cầu đề ra.

-Dễ dàng mở rộng về chiều dọc.

**Nhược điểm:**

**-**Đồ họa chưa đẹp

**-**Code chưa được tái cấu trúc tốt

## Kiến thức thu được:

Nắm được cơ sở việc lập trình một game bằng thư viện Pygame. Rèn luyện kỹ năng lập trình (hướng đối tượng và hướng thủ tục).

## Phân phối được phần mềm:

Link github repo: <https://github.com/trihtm8/snake_y.git>

# HƯỚNG PHÁT TRIỂN:

-Thêm chức năng lưu điểm.

-Mở rộng về lượng dữ liệu, số màn chơi, ...

-Tái cấu trúc code, mở rộng thêm các chức năng mới

-Nâng cấp đồ họa

-Combine file thành file .exe

# HƯỚNG DẪN CÀI ĐẶT:

-Tải Python.

-Download thư viện Pygame: trên cmd: pip install pygame

-Vào đường link trên để tải về.

-Trên cmd: trỏ đến thư mục chứa file snake\_y.py

python snake\_y.py