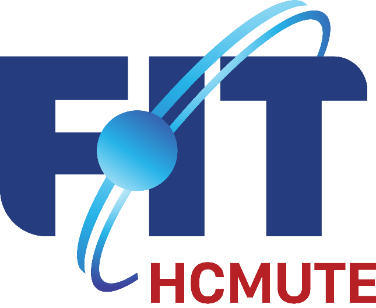
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT**

**THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO TIỂU LUẬN**

**NGÔN NGỮ LẬP TRÌNH PYTHON**

**ỨNG DỤNG LẬP TRÌNH GAME – TRÒ CHƠI CON RẮN**

**Thông tin thành viên:**

23162058 – Nguyễn Nhật Nam – 23162B

23162095 – Vũ Đức Thắng – 23162B

23162100 – Dương Nguyễn Ngọc Thương – 23162C

23162116 – Vũ Hoàng Vũ – 23162B

**Giảng viên:** TS. Phan Thị Thể

**Thành phố Hồ Chí Minh, 11/2024**

# NHẬN XÉT, ĐÁNH GIÁ CỦA GIẢNG VIÊN

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

# LỜI CẢM ƠN

Kính gửi cô Phan Thị Thể,

Chúng em xin gửi lời cảm ơn chân thành đến cô đã tận tình hướng dẫn và hỗ trợ nhóm em trong suốt quá trình thực hiện đồ án cuối kỳ. Những kiến thức quý báu và sự tận tâm của cô đã giúp chúng em vượt qua những khó khăn và hoàn thiện đồ án của mình.

Không chỉ vậy, sự động viên và khích lệ từ cô đã giúp chúng em tự tin hơn, sẵn sàng đối mặt với thử thách và phát triển bản thân. Chúng em rất trân trọng những góp ý, chỉ bảo của cô, điều đó không chỉ giúp nhóm em phát triển kỹ năng chuyên môn mà còn nâng cao khả năng tư duy sáng tạo. Chúng em hy vọng sẽ có nhiều cơ hội học hỏi từ cô hơn nữa trong tương lai.

Một lần nữa, chúng em xin chân thành cảm ơn cô!

Trân trọng,

Nhóm 3

|  |  |
| --- | --- |
| Trường ĐH Sư Phạm Kỹ Thuật TP.HCM  Khoa : CNTT |  |

**ĐỀ CƯƠNG BÁO CÁO ĐỒ ÁN CUỐI KÌ**

|  |  |
| --- | --- |
| Họ và Tên SV thực hiện 1: Nguyễn Nhật Nam | MSSV: 23162058 |
| Họ và Tên SV thực hiện 2: Vũ Đức Thắng | MSSV: 23162095 |
| Họ và Tên SV thực hiện 3: Dương Nguyễn Ngọc Thương | MSSV: 23162100 |
| Họ và Tên SV thực hiện 4: Vũ Hoàng Vũ | MSSV: 23162116 |

Thời gian làm báo cáo : từ : 25/11/2024 Đến : 27/11/2024

Chuyên ngành : An toàn thông tin

Tên luận văn : Ứng dụng lập trình game – Trò chơi con rắn

GV hướng dẫn : TS. Phan Thị Thể

**Nhiệm Vụ Của Báo Cáo :**

1. Giới thiệu đề tài
2. Khảo sát và phân tích yêu cầu
3. Xây dựng mô hình hệ thống
4. Thiết kế giao diện người dùng
5. Thiết kế cơ sở dữ liệu
6. Phát triển và lập trình hệ thống
7. Kiểm thử hệ thống
8. Triển khai hệ thống
9. Đánh giá kết quả đạt được
10. Kết luận và hướng phát triển
11. Tài liệu tham khảo
12. Phụ lục

# KẾ HOẠCH THỰC HIỆN

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thời gian** | **Công việc** | **Ghi chú** |
| **1** | 10/11 - 12/11 | Lập kế hoạch, thiết kế giao diện và kiến trúc | Xác định mục tiêu, vẽ sơ đồ, chia module |
| **2** | 13/11 - 18/11 | Phát triển các chức năng chính | Vòng lặp trò chơi, giao diện, đăng nhập, bảng xếp hạng, âm thanh, hiệu ứng |
| **3** | 19/11 - 21/11 | Hoàn thiện và tối ưu mã nguồn | Tối ưu hiệu suất, thêm tính năng tìm kiếm |
| **4** | 22/11 - 23/11 | Kiểm thử và chuẩn bị báo cáo | Test toàn bộ chức năng, sửa lỗi, viết báo cáo |
| **5** | 26/11 | Kiểm tra lần cuối và nộp đồ án | Chạy thử chương trình, rà soát báo cáo |

# PHÂN CÔNG CÔNG VIỆC

|  |  |  |
| --- | --- | --- |
| **Thành viên** | **Công việc chính** | **Mô tả chi tiết** |
| Vũ Đức Thắng | Phát triển cơ chế game và logic chính | - Tập trung vào các file main.py.  - Kiểm tra và cải tiến logic game chính (game Snake).  - Tối ưu hóa chương trình |
| Nguyễn Nhật Nam | Tương tác người dùng và giao diện | - Tập trung vào các file login.py, search.py.  - Cải tiến hệ thống đăng nhập đảm bảo tương tác mượt mà.  - Nâng cao chức năng tìm kiếm trên bảng xếp hạng.  - Thiết kế các giao diện mẫu (mock-up) hoặc chụp ảnh màn hình để sử dụng trong bài thuyết trình. |
| Vũ Hoàng Vũ | Quản lý người chơi và xử lý dữ liệu | - Tập trung vào các file Player.py, player\_manager.py.  - Kiểm tra chức năng tạo và quản lý người chơi.  - Kiểm tra việc load, lưu và cập nhật điểm số người chơi.  - Ghi lại và cải thiện các điểm mạnh/yếu trong xử lý dữ liệu. |
| Dương Nguyễn Ngọc Thương | Quản lý cơ chế nút bấm thao tác của người chơi | - Tập trung vào file button.py.  - Thử nghiệm và nâng cấp hệ thống tương tác nút bấm.  - Ghi lại các chức năng chính để phục vụ thuyết trình. |

**MỤC LỤC**

[NHẬN XÉT, ĐÁNH GIÁ CỦA GIẢNG VIÊN 2](#_Toc183634971)

[LỜI CẢM ƠN 3](#_Toc183634972)

[KẾ HOẠCH THỰC HIỆN 6](#_Toc183634973)

[PHÂN CÔNG CÔNG VIỆC 7](#_Toc183634974)

[DANH MỤC HÌNH ẢNH 12](#_Toc183634975)

[TÓM TẮT ĐỀ TÀI 14](#_Toc183634976)

[PHẦN MỞ ĐẦU 15](#_Toc183634977)

[1. Tính cấp thiết của đề tài 15](#_Toc183634978)

[2. Mục đích của đề tài 15](#_Toc183634979)

[3. Cách tiếp cận và phương pháp nghiên cứu 15](#_Toc183634980)

[PHẦN NỘI DUNG 17](#_Toc183634981)

[CHƯƠNG 1: TỔNG QUAN VÀ PHÂN TÍCH TRÒ CHƠI 17](#_Toc183634982)

[1.1. Tổng quan về trò chơi 17](#_Toc183634983)

[1.2. Công cụ và ngôn ngữ lập trình 18](#_Toc183634984)

[1.3. Đặc tả yêu cầu 18](#_Toc183634985)

[1.3.1. Yêu cầu chức năng 18](#_Toc183634986)

[1.3.2. Yêu cầu phi chức năng 19](#_Toc183634987)

[1.4. Phân tích yêu cầu hệ thống 20](#_Toc183634988)

[1.4.1. Chức năng 20](#_Toc183634989)

[1.4.2. Luồng hoạt động 21](#_Toc183634990)

[CHƯƠNG 2: TRIỂN KHAI MÃ NGUỒN 23](#_Toc183634991)

[2.1. Lớp Button 23](#_Toc183634992)

[2.1.1. Giới thiệu 23](#_Toc183634993)

[2.1.2. Thuộc tính: 23](#_Toc183634994)

[2.1.3. Các phương thức 24](#_Toc183634995)

[2.1.4. Ứng dụng thực tế 26](#_Toc183634996)

[2.2. Lớp Login 26](#_Toc183634997)

[2.2.1. Giới thiệu 26](#_Toc183634998)

[2.2.2. Phân tích hàm login\_screen 26](#_Toc183634999)

[2.2.3. Hàm phụ trợ: display\_message 30](#_Toc183635000)

[2.3. Lớp Player và PlayerManager 30](#_Toc183635001)

[2.3.1. Giới thiệu 30](#_Toc183635002)

[2.3.2. Cấu trúc lớp Player 30](#_Toc183635003)

[2.3.3. Cấu trúc lớp PlayerManager 31](#_Toc183635004)

[2.4. Lớp Search 35](#_Toc183635005)

[2.4.1. Giới thiệu 35](#_Toc183635006)

[2.4.2. Chức năng hiển thị bảng xếp hạng (superHardScoreBoard) 35](#_Toc183635007)

[2.4.3. Chức năng tìm kiếm trong bảng xếp hạng (search\_player\_in\_scoreboard) 36](#_Toc183635008)

[2.4.4. Chức năng hiển thị thông báo (display\_message) 37](#_Toc183635009)

[2.4.5. Điểm nổi bật chung 37](#_Toc183635010)

[2.5. Lớp main 37](#_Toc183635011)

[2.5.1. Giới thiệu 37](#_Toc183635012)

[2.5.2. Khởi tạo trò chơi 38](#_Toc183635013)

[2.5.3. Màn hình đăng nhập 38](#_Toc183635014)

[2.5.4. Menu chính 39](#_Toc183635015)

[2.5.5. Chọn độ khó 39](#_Toc183635016)

[2.5.6. Vòng lặp trò chơi 39](#_Toc183635017)

[2.5.7. Bảng xếp hạng 41](#_Toc183635018)

[2.5.8. Đặc điểm nổi bật 41](#_Toc183635019)

[CHƯƠNG 3: KIỂM THỬ VÀ ĐÁNH GIÁ 41](#_Toc183635020)

[3.1. Mô tả các trường hợp kiểm thử 41](#_Toc183635021)

[3.1.1. Di chuyển rắn trong các tình huống khác nhau 41](#_Toc183635022)

[3.1.2. Xác minh điểm số khi rắn săn mồi 42](#_Toc183635023)

[3.1.3. Xác minh game kết thúc khi va chạm 42](#_Toc183635024)

[3.2. Kết quả kiểm thử và phân tích các lỗi gặp phải 42](#_Toc183635025)

[3.2.1. Kết quả kiểm thử 42](#_Toc183635026)

[3.2.2. Lỗi gặp phải 43](#_Toc183635027)

[3.2.3. Phân tích 43](#_Toc183635028)

[CHƯƠNG 4: CẢI TIẾN VÀ MỞ RỘNG 44](#_Toc183635029)

[4.1. Đề xuất cải tiến 44](#_Toc183635030)

[4.2. Tích hợp các yếu tố thú vị 44](#_Toc183635031)

[PHẦN KẾT LUẬN VÀ KHUYẾN NGHỊ 45](#_Toc183635032)

[1. Kết luận 45](#_Toc183635033)

[2. Khuyến nghị 45](#_Toc183635034)

[TÀI LIỆU THAM KHẢO 46](#_Toc183635035)

# DANH MỤC HÌNH ẢNH

[Hình 2. 1: Hàm def \_\_init\_\_() thuộc lớp button 24](#_Toc183634943)

[Hình 2. 2: Hàm def update() thuộc lớp button 25](#_Toc183634944)

[Hình 2. 3: Hàm checkForInput() thuộc lớp button 25](#_Toc183634945)

[Hình 2. 4: Hàm def changeColor() thuộc lớp button 25](#_Toc183634946)

[Hình 2. 5: Khởi tạo các thành phần thuộc lớp login 27](#_Toc183634947)

[Hình 2. 6: Vòng lặp chính while True 28](#_Toc183634948)

[Hình 2. 7: Code xử lý sự kiện trong trò chơi 28](#_Toc183634949)

[Hình 2. 8: Hiện thị nội dung văn bản lên màn hình 29](#_Toc183634950)

[Hình 2. 9: Hàm display\_message() thuộc lớp login 30](#_Toc183634951)

[Hình 2. 10: Hàm def \_\_init\_\_() thuộc lớp player 31](#_Toc183634952)

[Hình 2. 11: Hàm def \_\_repr\_\_() thuộc lớp player 31](#_Toc183634953)

[Hình 2. 12: Hàm def \_\_init\_\_() thuộc lớp player manager 32](#_Toc183634954)

[Hình 2. 13: Hàm add\_player() thuộc lớp play manager 32](#_Toc183634955)

[Hình 2. 14: Hàm def load\_players() thuộc lớp player manager 33](#_Toc183634956)

[Hình 2. 15: Hàm def save\_players() thuộc lớp player manager 33](#_Toc183634957)

[Hình 2. 16: Hàm def show\_players() thuộc lớp player manager 34](#_Toc183634958)

[Hình 2. 17: Hàm def sort\_players() thuộc lớp player manager 34](#_Toc183634959)

[Hình 2. 18: Hàm def find\_player() thuộc lớp player manager 34](#_Toc183634960)

[Hình 2. 19: Vòng lặp for thực hiện chức năng hiển thị bảng xếp hạng 36](#_Toc183634961)

[Hình 2. 20: Câu điều kiện if – else kiểm tra 36](#_Toc183634962)

[Hình 2. 21: Hàm thực hiện chức năng hiển thị thông báo 37](#_Toc183634963)

[Hình 2. 22: Menu chính của trò chơi 39](#_Toc183634964)

[Hình 2. 23: Chọn độ khó cho trò chơi 39](#_Toc183634965)

[Hình 2. 24: Di chuyển rắn 40](#_Toc183634966)

[Hình 2. 25: Sinh táo 40](#_Toc183634967)

[Hình 2. 26: Va chạm 40](#_Toc183634968)

[Hình 2. 27: Kết thúc trò chơi 40](#_Toc183634969)

[Hình 2. 28: Bảng xếp hạng 41](#_Toc183634970)

# TÓM TẮT ĐỀ TÀI

Snake Game là một trò chơi kinh điển được xây dựng bằng Python với thư viện Pygame. Mục tiêu chính của trò chơi là điều khiển rắn để ăn thức ăn, làm tăng chiều dài của rắn, đồng thời tránh va chạm vào tường hoặc chính thân mình. Trò chơi được thiết kế với giao diện đơn giản, gồm màn hình hiển thị, thức ăn, và con rắn di chuyển.

Chương trình sử dụng các thành phần cơ bản như vòng lặp while để duy trì trạng thái chơi, các sự kiện từ bàn phím để điều khiển hướng đi của rắn, và cơ chế phát hiện va chạm. Thư viện Pygame giúp tạo đồ họa và âm thanh, mang lại trải nghiệm mượt mà hơn. Trạng thái trò chơi được cập nhật liên tục với logic đơn giản, bao gồm tăng điểm số khi ăn thức ăn và kết thúc trò chơi khi rắn va chạm.

Snake Game bằng Python là một dự án thú vị, phù hợp cho người học lập trình muốn thực hành về logic trò chơi, xử lý sự kiện, và sử dụng thư viện Pygame để phát triển ứng dụng tương tác.

# PHẦN MỞ ĐẦU

# 1. Tính cấp thiết của đề tài

Trò chơi Snake là một trong những game cổ điển phổ biến, mang tính giáo dục và giải trí cao. Việc phát triển trò chơi này bằng Python giúp sinh viên làm quen với lập trình game cơ bản, xử lý đồ họa, và logic lập trình.

Python là ngôn ngữ phổ biến với cú pháp đơn giản và dễ học, phù hợp cho việc phát triển ứng dụng nhỏ như game. Điều này làm cho việc nghiên cứu và phát triển Snake Game trở thành một bước khởi đầu quan trọng trong lĩnh vực lập trình game.

Đề tài giúp củng cố kiến thức về các cấu trúc dữ liệu cơ bản như danh sách, xử lý sự kiện, và vòng lặp trong Python, đồng thời mở rộng ứng dụng lập trình vào lĩnh vực giải trí.

# 2. Mục đích của đề tài

Xây dựng một game Snake hoàn chỉnh với giao diện đơn giản, dễ sử dụng.

Tìm hiểu và áp dụng các thư viện Python (như pygame) để lập trình và xử lý đồ họa.

Phát triển kỹ năng thiết kế thuật toán và logic xử lý trong lập trình game.

Khuyến khích tư duy sáng tạo và cải tiến tính năng, chẳng hạn như tăng độ khó, lưu điểm cao, hoặc bổ sung các yếu tố mới.

# 3. Cách tiếp cận và phương pháp nghiên cứu

Cách tiếp cận: Phân tích các phiên bản Snake Game trước đó để hiểu cách hoạt động, từ đó thiết kế game theo phong cách riêng.

Phương pháp nghiên cứu:

* Nghiên cứu lý thuyết: Tìm hiểu về thư viện pygame và các thuật toán cơ bản để điều khiển rắn, sinh mồi, và tính điểm.
* Thực nghiệm: Viết mã nguồn Python, kiểm thử các chức năng chính, và tối ưu hóa hiệu suất của trò chơi.
* So sánh và đánh giá: So sánh kết quả đạt được với các game mẫu khác để hoàn thiện sản phẩm.

# PHẦN NỘI DUNG

# CHƯƠNG 1: TỔNG QUAN VÀ PHÂN TÍCH TRÒ CHƠI

# 1.1. Tổng quan về trò chơi

Lịch sử và phát triển

* Trò chơi Snake ra đời từ năm 1976 với tên Blockade, nhưng trở nên phổ biến toàn cầu nhờ phiên bản cài đặt sẵn trên điện thoại Nokia 6110 (1997).
* Qua thời gian, Snake được cải tiến với nhiều phiên bản như Snake II (2000) và các bản trực tuyến như Slither.io (2016), đưa trò chơi trở thành biểu tượng trong làng game cổ điển.
* Snake cũng là công cụ giáo dục lập trình hiệu quả, được phát triển trên nhiều ngôn ngữ như Python và JavaScript.

Đặc điểm chính

* Điều khiển rắn: Người chơi dùng phím điều hướng để điều khiển rắn di chuyển, tránh va chạm với tường hoặc thân mình.
* Ăn mồi: Rắn ăn mồi để dài ra, tăng điểm số; mồi được sinh ra ngẫu nhiên trên màn hình.
* Tính điểm: Điểm số tăng theo số lần ăn mồi, một số phiên bản có điểm thưởng đặc biệt.
* Tăng độ khó: Tốc độ rắn tăng dần hoặc xuất hiện thêm chướng ngại vật sau mỗi lần ăn mồi.
* Kết thúc trò chơi: Game over khi rắn va chạm vào tường hoặc chính nó.

# 1.2. Công cụ và ngôn ngữ lập trình

Giới thiệu về Python và thư viện Pygame:

* Python: Ngôn ngữ lập trình bậc cao, dễ học, và phổ biến trong việc phát triển các ứng dụng nhỏ, đặc biệt là game. Python có cú pháp đơn giản, phù hợp cho cả người mới bắt đầu và lập trình viên chuyên nghiệp.
* Pygame: Thư viện mạnh mẽ dành riêng cho phát triển game 2D, cung cấp các công cụ để xử lý đồ họa, âm thanh, và sự kiện tương tác một cách hiệu quả. Đây là lựa chọn phổ biến khi viết game bằng Python.

Mô tả môi trường phát triển:

* IDE: Sử dụng các công cụ như PyCharm, Visual Studio Code, hoặc thậm chí IDLE để viết và gỡ lỗi mã Python.
* Hệ điều hành: Game có thể phát triển và chạy trên nhiều nền tảng như Windows, macOS, hoặc Linux nhờ tính đa nền tảng của Python và Pygame.
* Thư viện hỗ trợ:
  + Pygame: Xử lý đồ họa và âm thanh.
  + Random: Sinh số ngẫu nhiên cho vị trí mồi.
  + Sys: Hỗ trợ thoát chương trình khi kết thúc trò chơi hoặc xử lý các lỗi nghiêm trọng.

# 1.3. Đặc tả yêu cầu

## 1.3.1. Yêu cầu chức năng

Điều khiển rắn:

* Người chơi có thể điều khiển rắn di chuyển lên, xuống, trái, hoặc phải thông qua bàn phím.
* Rắn không thể quay đầu ngược lại ngay lập tức (ví dụ: từ trái sang phải).

Quản lý thức ăn:

* Thức ăn được sinh ra ngẫu nhiên trên màn hình.
* Đảm bảo vị trí thức ăn không trùng với thân rắn hoặc chướng ngại vật.
* Thức ăn có hai loại: bình thường (+1 điểm) và đặc biệt (+3 điểm).

Điểm số và tốc độ:

* Điểm hiện tại tăng khi rắn ăn thức ăn.
* Điểm cao nhất được lưu trữ và hiển thị qua các lần chơi.
* Tốc độ di chuyển của rắn tăng theo điểm số.

Kiểm tra va chạm và game over:

* Game over xảy ra khi rắn va chạm:
  + Với thân mình.
  + Với chướng ngại vật.
* Hiển thị thông báo và cho phép người chơi chọn chơi lại hoặc thoát.

Hiển thị giao diện:

* Hiển thị rắn, thức ăn, và chướng ngại vật trên màn hình.
* Hiển thị điểm số hiện tại và điểm cao nhất.

## 1.3.2. Yêu cầu phi chức năng

Hiệu năng: Game phải hoạt động mượt mà, tốc độ khung hình tối thiểu 30 FPS.

Tính ổn định: Đảm bảo lưu trữ và đọc điểm cao từ tệp hoạt động đúng cách.

Khả năng mở rộng: Cấu trúc mã dễ bảo trì, hỗ trợ thêm các tính năng mới như chế độ chơi hoặc đồ họa nâng cao.

# 1.4. Phân tích yêu cầu hệ thống

## 1.4.1. Chức năng

Quản lý rắn:

* Lưu trữ vị trí thân rắn dưới dạng danh sách tọa độ.
* Cập nhật vị trí đầu rắn khi di chuyển.
* Xóa tọa độ đuôi để duy trì chiều dài nếu không ăn mồi.

Quản lý thức ăn:

* Sinh tọa độ ngẫu nhiên cho thức ăn.
* Hiển thị thức ăn trên màn hình với màu sắc khác nhau tùy loại.

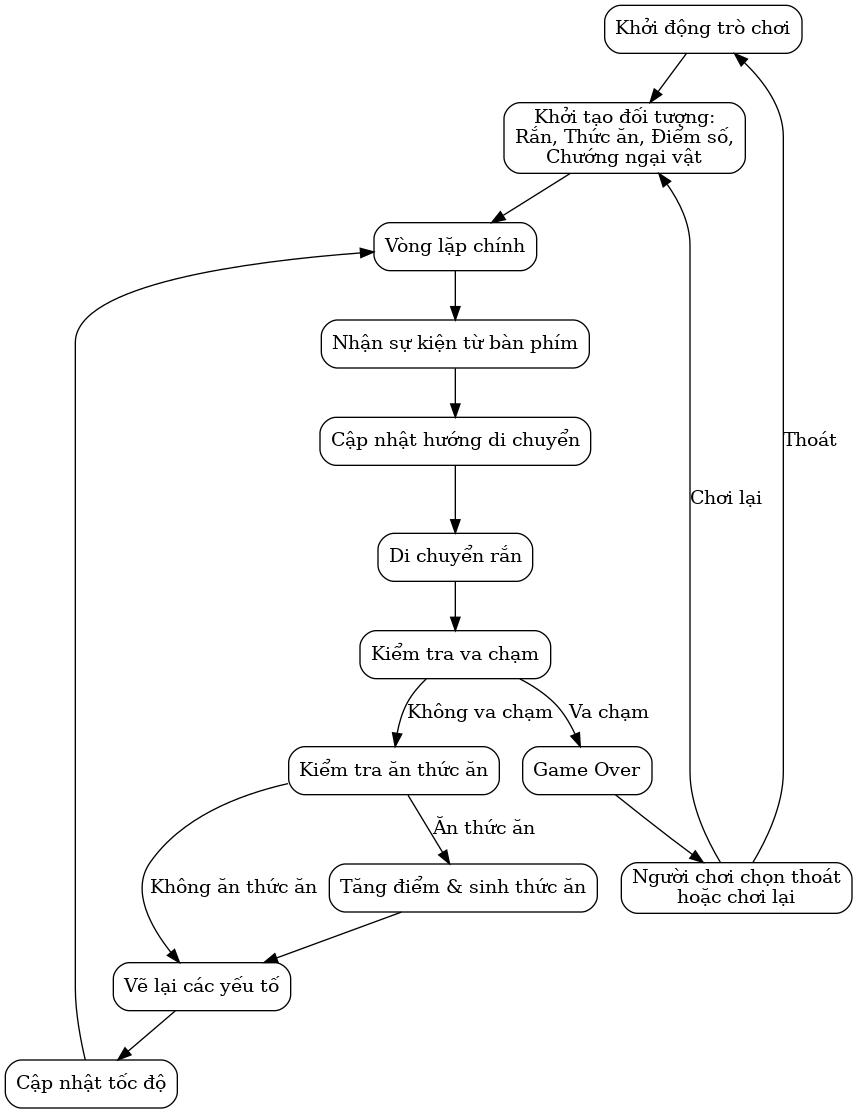
Quản lý điểm số:

* Tăng điểm hiện tại khi ăn thức ăn.
* So sánh điểm hiện tại với điểm cao nhất để cập nhật nếu cần.

Kiểm tra va chạm: Kiểm tra đầu rắn có chạm vào thân, chướng ngại vật, hoặc tường không.

Tạo giao diện: Hiển thị rắn, thức ăn, chướng ngại vật, và điểm số trên màn hình.

## 1.4.2. Luồng hoạt động



Khởi tạo trò chơi:

* Tạo cửa sổ trò chơi với kích thước cố định.
* Khởi tạo rắn, thức ăn, điểm số, và chướng ngại vật.

Vòng lặp chính:

* Nhận sự kiện từ bàn phím để thay đổi hướng di chuyển.
* Di chuyển rắn theo hướng đã chỉ định.

Kiểm tra các điều kiện sau:

* Nếu rắn ăn thức ăn, tăng điểm, sinh thức ăn mới, và cập nhật chiều dài rắn.
* Nếu rắn va chạm (thân, chướng ngại vật), dừng trò chơi và hiển thị thông báo game over.
* Vẽ lại các yếu tố trò chơi trên màn hình.
* Cập nhật tốc độ trò chơi dựa trên điểm số.

Kết thúc trò chơi:

* Khi game over, cho phép người chơi chọn chơi lại hoặc thoát trò chơi.
* Nếu chọn chơi lại, khởi tạo lại các đối tượng trò chơi.

# CHƯƠNG 2: TRIỂN KHAI MÃ NGUỒN

Chi tiết mã nguồn Python của game Snake, bao gồm:

# 2.1. Lớp Button

## 2.1.1. Giới thiệu

Lớp Button được thiết kế để tạo các nút giao diện trong trò chơi với khả năng tương tác như:

* Hiển thị văn bản hoặc hình ảnh trên nút.
* Phản hồi với hành động của người dùng, ví dụ: đổi màu khi di chuột qua.
* Phát hiện khi nút được nhấn.

## 2.1.2. Thuộc tính:

image:

* Hình ảnh hiển thị trên nút.
* Nếu không có hình ảnh, nút sẽ chỉ hiển thị văn bản.

pos:

* Vị trí (tọa độ x, y) của nút trên màn hình.

font:

* Font chữ để hiển thị văn bản trên nút.
* base\_color và hovering\_color:
* Màu sắc mặc định và màu khi chuột di chuyển qua nút.

text\_input:

* Văn bản hiển thị trên nút.

text:

* Văn bản đã được render để hiển thị.

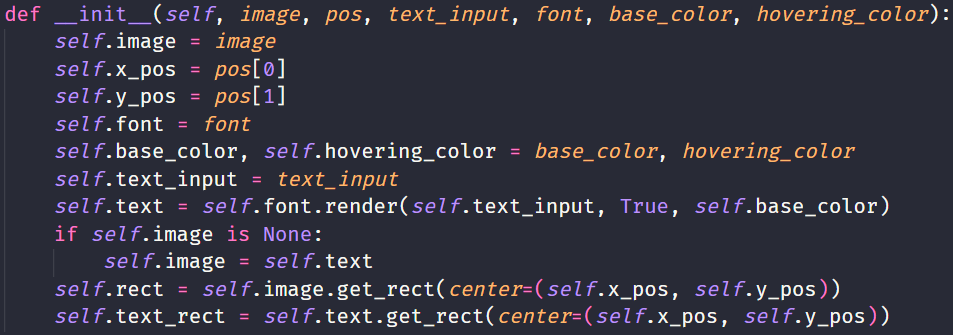
rect và text\_rect:

* Các đối tượng Rect đại diện cho vùng bao của hình ảnh và văn bản.

## 2.1.3. Các phương thức

Khởi tạo \_\_init\_\_()

* Chức năng:
* Thiết lập các thuộc tính của nút khi được khởi tạo.
* Xác định hình ảnh hoặc văn bản hiển thị trên nút.

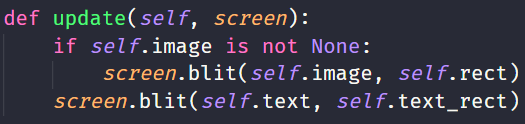


Hình 2. 1: Hàm def \_\_init\_\_() thuộc lớp button

* Kết quả:
* Khi tạo nút, hình ảnh hoặc văn bản được định vị chính xác theo tọa độ pos.
* Nếu không có hình ảnh, văn bản được sử dụng làm đối tượng hiển thị chính.

Phương thức update():

* Chức năng:Vẽ hình ảnh và văn bản của nút lên màn hình.

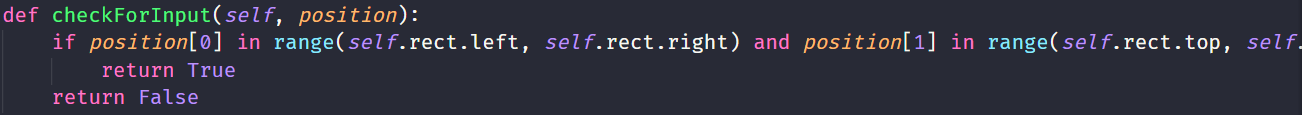


Hình 2. 2: Hàm def update() thuộc lớp button

* Kết quả: Nút được hiển thị đầy đủ trên giao diện người dùng.

Phương thức checkForInput():

* Chức năng: Kiểm tra xem người dùng có nhấn vào vùng nút hay không.

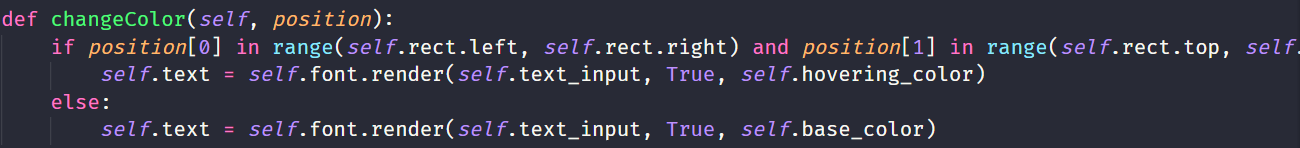


Hình 2. 3: Hàm checkForInput() thuộc lớp button

* Kết quả: Trả về True nếu vị trí chuột nằm trong vùng của nút.

Phương thức changeColor():

* Chức năng: Đổi màu văn bản khi chuột di chuyển qua nút.



Hình 2. 4: Hàm def changeColor() thuộc lớp button

* Kết quả:
* Khi chuột di chuyển qua, văn bản trên nút đổi sang màu hovering\_color.
* Khi chuột rời khỏi, văn bản trở về màu base\_color.

## 2.1.4. Ứng dụng thực tế

Lớp Button được sử dụng để tạo các nút cho các màn hình như:

Menu chính: Các nút: "Play Game", "Scoreboard", "Quit".

Màn hình chọn độ khó: Các nút: "Normal", "Hard", "Super Hard".

# 2.2. Lớp Login

## 2.2.1. Giới thiệu

Hàm login\_screen đảm nhận vai trò hiển thị màn hình đăng nhập cho người chơi trước khi bắt đầu trò chơi.

Tính năng chính:

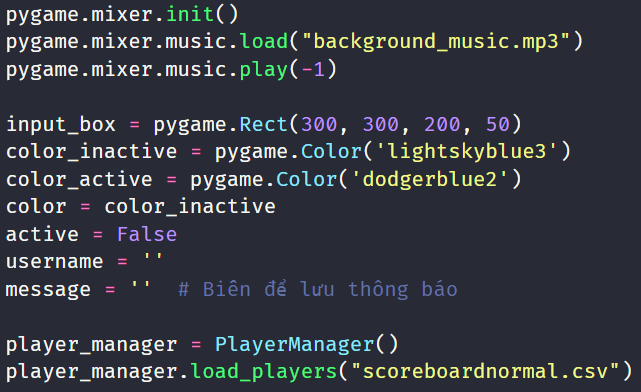
* Hiển thị giao diện nhập tên người chơi.
* Kiểm tra thông tin đầu vào (trống hoặc trùng lặp).
* Lưu thông tin người chơi mới vào hệ thống.

## 2.2.2. Phân tích hàm login\_screen

Tham số đầu vào

* screen: Đối tượng màn hình Pygame.
* bg: Ảnh nền của màn hình đăng nhập.
* get\_font: Hàm để lấy font chữ.
* clock: Đối tượng đồng hồ để giới hạn FPS.
* framerate: Số khung hình hiển thị mỗi giây (FPS).

Khởi tạo các thành phần:

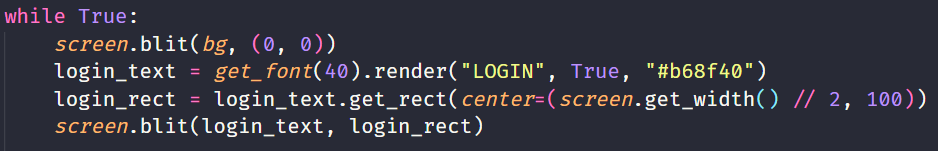


Hình 2. 5: Khởi tạo các thành phần thuộc lớp login

Chức năng:

* Âm nhạc: Phát nhạc nền trong màn hình đăng nhập.
* Hộp nhập liệu: Sử dụng pygame.Rect để định dạng vùng nhập liệu.
* Màu sắc: Định nghĩa màu khi hộp nhập liệu đang được chọn (color\_active) và không được chọn (color\_inactive).
* Quản lý người chơi: Sử dụng PlayerManager để tải thông tin người chơi từ file.

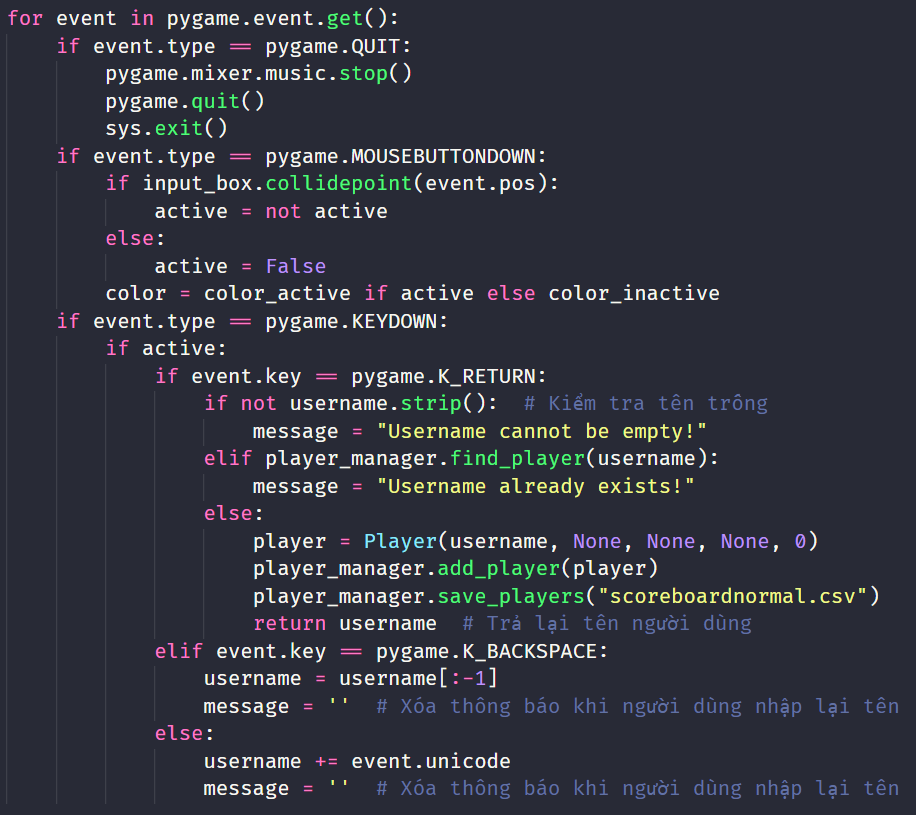
Vòng lặp chính:



Hình 2. 6: Vòng lặp chính while True

Chức năng: Hiển thị ảnh nền và tiêu đề "LOGIN" ở giữa màn hình.

Xử lý sự kiện:

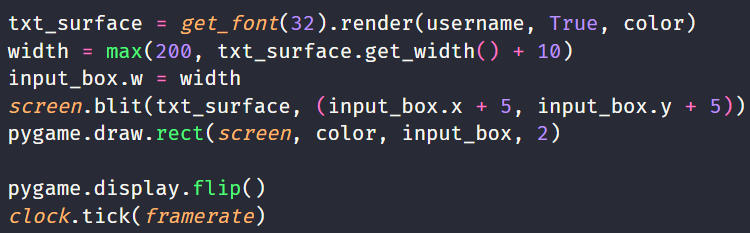


Hình 2. 7: Code xử lý sự kiện trong trò chơi

Chức năng:

* Đóng chương trình: Nhấn nút "X" sẽ dừng nhạc nền và thoát khỏi trò chơi.
* Xử lý chuột: Nhấn chuột vào hộp nhập liệu để kích hoạt trạng thái nhập tên.
* Xử lý bàn phím:
  + Enter: Kiểm tra thông tin người chơi:
    - Báo lỗi nếu tên trống hoặc trùng.
    - Thêm người chơi mới nếu hợp lệ.
  + Backspace: Xóa ký tự cuối trong chuỗi nhập.

Hiển thị văn bản:

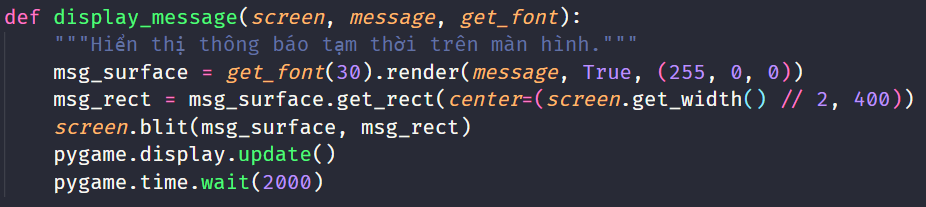


Hình 2. 8: Hiện thị nội dung văn bản lên màn hình

Chức năng:

* Hiển thị nội dung nhập tên của người chơi trong hộp nhập liệu.
* Điều chỉnh chiều rộng của hộp nhập liệu theo nội dung.

## 2.2.3. Hàm phụ trợ: display\_message



Hình 2. 9: Hàm display\_message() thuộc lớp login

Chức năng: Hiển thị thông báo lỗi hoặc thông tin tạm thời trên màn hình.

# 2.3. Lớp Player và PlayerManager

## 2.3.1. Giới thiệu

Lớp Player và PlayerManager được thiết kế để quản lý thông tin và thao tác dữ liệu người chơi:

* Lớp Player: Đại diện cho một người chơi với các thông tin chi tiết như tên, số đĩa, số lượt di chuyển, thời gian chơi và điểm số.
* Lớp PlayerManager: Chịu trách nhiệm quản lý danh sách người chơi, bao gồm thêm, tải, lưu, sắp xếp, và tìm kiếm.

## 2.3.2. Cấu trúc lớp Player

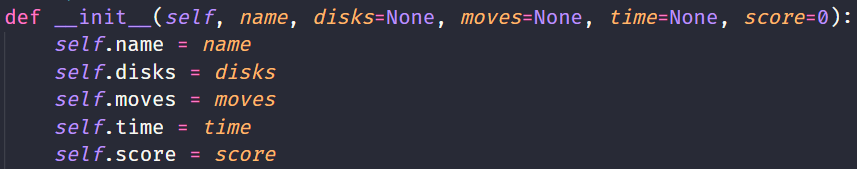
Thuộc tính:

Lớp Player lưu trữ thông tin sau:

* name: Tên người chơi.
* disks: Số đĩa trong trò chơi (có thể liên quan đến chế độ hoặc vòng chơi).
* moves: Số lượt di chuyển.
* time: Thời gian hoàn thành trò chơi.
* score: Điểm số của người chơi.

Các phương thức:

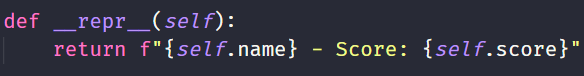
Khởi tạo \_\_int\_\_():



Hình 2. 10: Hàm def \_\_init\_\_() thuộc lớp player

Chức năng: Khởi tạo đối tượng người chơi với thông tin cụ thể.

Phương thức \_\_repr\_\_():



Hình 2. 11: Hàm def \_\_repr\_\_() thuộc lớp player

Chức năng: Cung cấp chuỗi biểu diễn của đối tượng Player khi in ra màn hình.

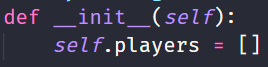
## 2.3.3. Cấu trúc lớp PlayerManager

Thuộc tính:

players: Danh sách các đối tượng Player.

Các phương thức:

Khởi tạo \_\_init\_\_():



Hình 2. 12: Hàm def \_\_init\_\_() thuộc lớp player manager

Chức năng: Khởi tạo danh sách người chơi rỗng.

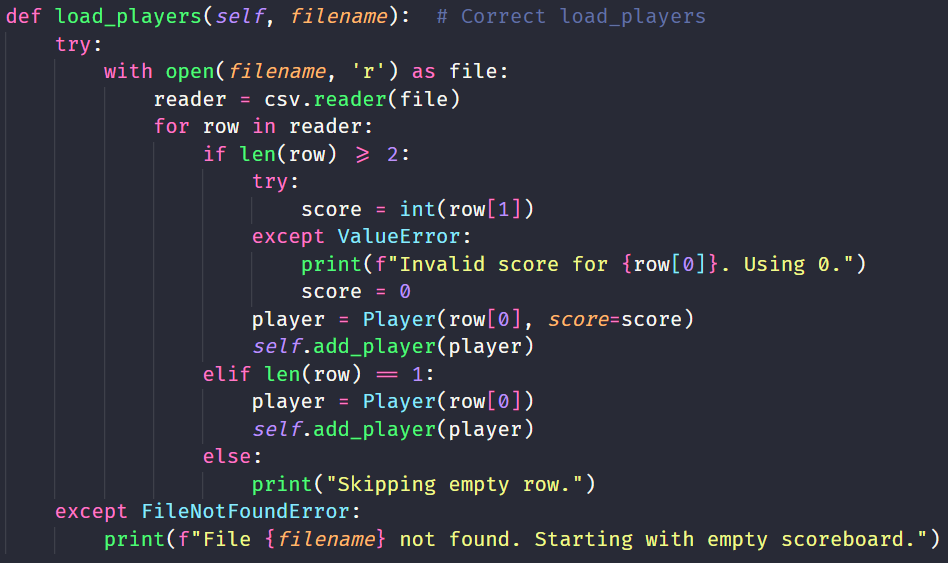
Phương thức add\_player():



Hình 2. 13: Hàm add\_player() thuộc lớp play manager

Chức năng: Thêm một người chơi vào danh sách.

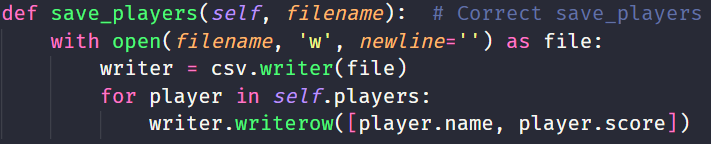
Phương thức load\_players():



Hình 2. 14: Hàm def load\_players() thuộc lớp player manager

Chức năng: Tải thông tin người chơi từ tệp CSV.

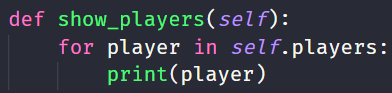
Phương thức save\_players():



Hình 2. 15: Hàm def save\_players() thuộc lớp player manager

Chức năng: Lưu danh sách người chơi vào tệp CSV.

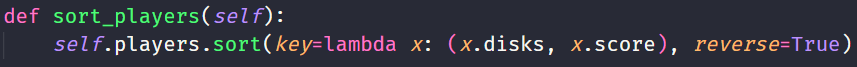
Phương thức show\_players():



Hình 2. 16: Hàm def show\_players() thuộc lớp player manager

Chức năng: Hiển thị thông tin tất cả người chơi.

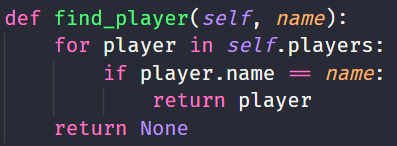
Phương thức sort\_players():



Hình 2. 17: Hàm def sort\_players() thuộc lớp player manager

Chức năng: Sắp xếp danh sách người chơi theo số đĩa giảm dần, sau đó là điểm số.

Phương thức find\_player():



Hình 2. 18: Hàm def find\_player() thuộc lớp player manager

Chức năng: Tìm kiếm người chơi theo tên.

# 2.4. Lớp Search

## 2.4.1. Giới thiệu

Mã nguồn được tổ chức để hỗ trợ tìm kiếm người chơi và hiển thị bảng xếp hạng cho trò chơi Snake. Các tính năng chính bao gồm tìm kiếm theo tên người chơi, sắp xếp điểm số, và hiển thị bảng xếp hạng với giao diện người dùng trực quan.

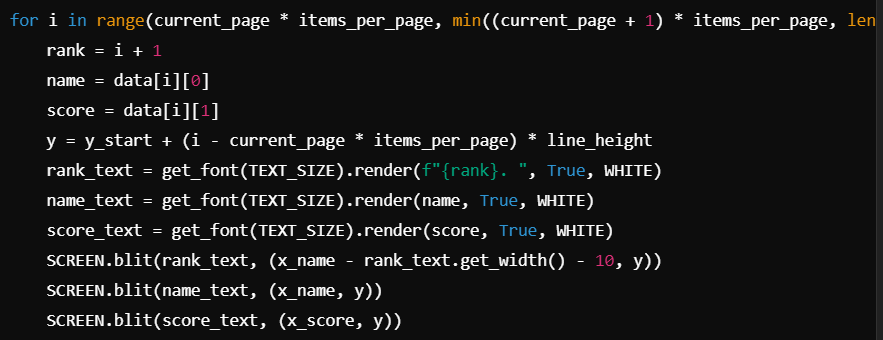
## 2.4.2. Chức năng hiển thị bảng xếp hạng (superHardScoreBoard)

Chức năng chính:

* Đọc dữ liệu từ tệp CSV chứa danh sách người chơi và điểm số.
* Hiển thị bảng xếp hạng theo thứ tự điểm giảm dần.
* Cung cấp nút điều hướng:
* Next: Chuyển đến trang kế tiếp.
* Previous: Quay lại trang trước.
* Search: Tìm kiếm thông tin người chơi.

Điểm nổi bật:

* Sắp xếp dữ liệu: Điểm số được sắp xếp giảm dần.
* Phân trang: Hiển thị số lượng người chơi cố định trên mỗi trang.
* Nút điều hướng: Dễ dàng thao tác qua các trang.



Hình 2. 19: Vòng lặp for thực hiện chức năng hiển thị bảng xếp hạng

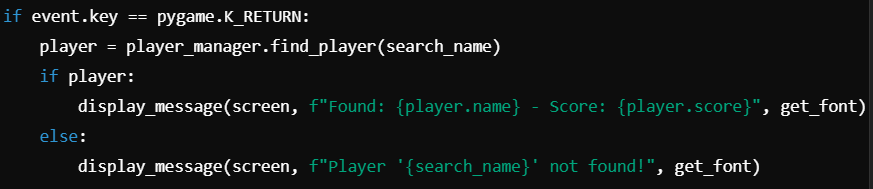
## 2.4.3. Chức năng tìm kiếm trong bảng xếp hạng (search\_player\_in\_scoreboard)

Chức năng chính:

* Người dùng nhập tên người chơi cần tìm.
* Kết quả hiển thị nếu tìm thấy người chơi.
* Nếu không tìm thấy, thông báo lỗi được hiển thị.

Điểm nổi bật:

* Hộp tìm kiếm: Giao diện trực quan, dễ thao tác.
* Thông báo: Hiển thị thông báo tạm thời (2 giây) khi không tìm thấy người chơi.



Hình 2. 20: Câu điều kiện if – else kiểm tra

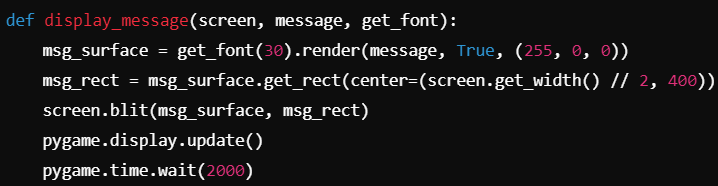
## 2.4.4. Chức năng hiển thị thông báo (display\_message)

Chức năng chính:

* Hiển thị thông báo tạm thời trên màn hình.
* Thời gian hiển thị: 2 giây.

Điểm nổi bật:

* Thẩm mỹ: Vị trí thông báo được căn chỉnh giữa màn hình.
* Thời gian tự động: Không yêu cầu người dùng đóng thông báo.



Hình 2. 21: Hàm thực hiện chức năng hiển thị thông báo

## 2.4.5. Điểm nổi bật chung

Tương tác người dùng: Hộp tìm kiếm và các nút được thiết kế trực quan.

Chức năng mạnh mẽ: Tìm kiếm, sắp xếp và phân trang bảng xếp hạng.

Hiệu năng: Dữ liệu được xử lý hiệu quả với tệp CSV.

# 2.5. Lớp main

## 2.5.1. Giới thiệu

Mã nguồn được tổ chức để phát triển trò chơi Snake Game với các tính năng phong phú bao gồm đăng nhập, chọn độ khó, hiệu ứng pháo hoa, và quản lý bảng xếp hạng.

## 2.5.2. Khởi tạo trò chơi

Tệp tin quan trọng được nhập:

* pygame: Thư viện chính để xây dựng trò chơi.
* button: Tạo và quản lý các nút giao diện.
* PhaoHoa: Hiệu ứng pháo hoa khi trò chơi kết thúc.
* login: Màn hình đăng nhập người chơi.
* search: Tìm kiếm thông tin trong bảng xếp hạng.
* player\_manager: Quản lý thông tin người chơi.

Cài đặt ban đầu:

* Khởi tạo màn hình với kích thước 800x600.
* Tải ảnh nền, âm thanh và cài đặt tốc độ khung hình (framerate = 60).
* Định nghĩa các màu sắc chính như RED, WHITE, GOLD, và các thông số cơ bản của trò chơi (kích thước rắn, tốc độ).

## 2.5.3. Màn hình đăng nhập

Chức năng:

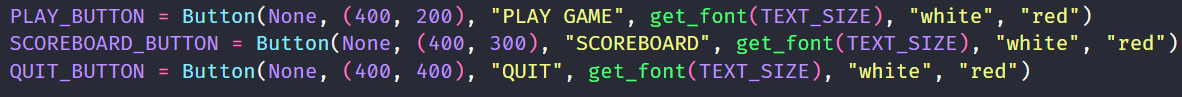
* Người chơi nhập tên để bắt đầu.
* Kiểm tra tên trống hoặc trùng lặp với tên đã có trong bảng xếp hạng.

Đoạn mã nổi bật: USERNAME = login\_screen(SCREEN, BG\_GAME, get\_font, clock, framerate)

Kết quả:

* Lưu tên người chơi toàn cục trong biến USERNAME.
* Sau khi đăng nhập thành công, chuyển đến menu chính.

## 2.5.4. Menu chính



Hình 2. 22: Menu chính của trò chơi

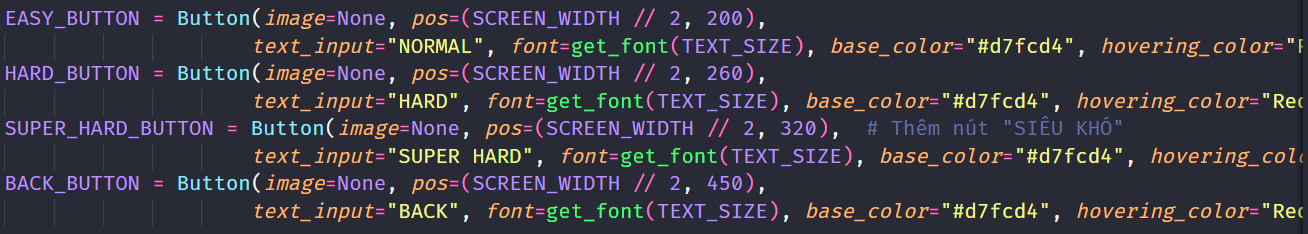
Chức năng:

* Hiển thị tên người chơi.
* Cung cấp các tùy chọn:
  + Play Game: Chọn độ khó và bắt đầu trò chơi.
  + Scoreboard: Xem bảng xếp hạng.
  + Quit: Thoát trò chơi.

## 2.5.5. Chọn độ khó

Chức năng:

* Người chơi chọn một trong ba chế độ:
* Normal: Tốc độ và không có chướng ngại vật.
* Hard: Tốc độ nhanh hơn và có chướng ngại vật.
* Super Hard: Chướng ngại vật thay đổi theo thời gian.



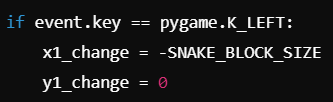
Hình 2. 23: Chọn độ khó cho trò chơi

## 2.5.6. Vòng lặp trò chơi

Chức năng:

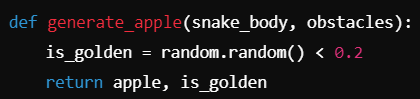
* Điều khiển rắn di chuyển, ăn mồi, và kiểm tra va chạm.
* Sinh ngẫu nhiên chướng ngại vật và táo (với xác suất 20% là táo vàng).
* Tăng chiều dài rắn và điểm số khi ăn táo.

Di chuyển rắn:



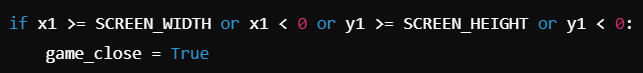
Hình 2. 24: Di chuyển rắn

Sinh táo:



Hình 2. 25: Sinh táo

Va chạm: Với tường, chướng ngại vật hoặc chính cơ thể rắn



Hình 2. 26: Va chạm

Kết thúc trò chơi: Hiển thị pháo hoa và lưu điểm số.

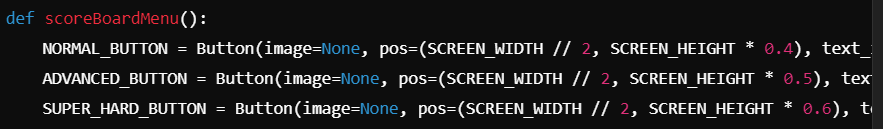


Hình 2. 27: Kết thúc trò chơi

## 2.5.7. Bảng xếp hạng

Chức năng:

* Hiển thị danh sách người chơi theo điểm số giảm dần.
* Cung cấp các nút Next, Previous để chuyển trang và Search để tìm kiếm người chơi.



Hình 2. 28: Bảng xếp hạng

## 2.5.8. Đặc điểm nổi bật

Tính năng phong phú:

* Hỗ trợ ba chế độ chơi.
* Hiệu ứng pháo hoa sinh động.
* Quản lý điểm số và tìm kiếm người chơi trong bảng xếp hạng.

Giao diện trực quan: Các nút tương tác thay đổi màu sắc khi người chơi di chuột qua.

Tích hợp âm thanh: Âm thanh khi ăn táo, thua cuộc, hoặc nhấn nút.

# CHƯƠNG 3: KIỂM THỬ VÀ ĐÁNH GIÁ

# 3.1. Mô tả các trường hợp kiểm thử

## 3.1.1. Di chuyển rắn trong các tình huống khác nhau

Kiểm tra phản hồi điều khiển: Xác minh rắn di chuyển đúng hướng khi nhấn các phím mũi tên (trái, phải, lên, xuống).

Tình huống biên: Kiểm tra khi rắn di chuyển đến ranh giới của màn hình để đảm bảo cơ chế xuyên tường hoạt động chính xác.

Kiểm tra logic quay đầu: Đảm bảo rắn không thể quay ngược đầu (ví dụ: từ trái sang phải ngay lập tức).

## 3.1.2. Xác minh điểm số khi rắn săn mồi

Điểm số đúng: Đảm bảo điểm số tăng +1 khi ăn thức ăn thường và +3 khi ăn thức ăn đặc biệt.

Cập nhật vị trí thức ăn: Kiểm tra vị trí thức ăn mới không trùng với thân rắn sau khi mồi cũ bị ăn.

Cập nhật điểm cao: Xác minh điểm cao được cập nhật khi điểm hiện tại vượt qua điểm cao cũ.

## 3.1.3. Xác minh game kết thúc khi va chạm

Va chạm với thân rắn: Kiểm tra game kết thúc khi đầu rắn chạm vào bất kỳ phần nào của thân.

Va chạm với chướng ngại vật: Đảm bảo trò chơi kết thúc nếu đầu rắn va chạm vào các block chướng ngại.

Hiển thị thông báo kết thúc: Xác minh thông báo "Bạn đã thua!" được hiển thị đúng khi game over.

# 3.2. Kết quả kiểm thử và phân tích các lỗi gặp phải

## 3.2.1. Kết quả kiểm thử

Rắn di chuyển đúng theo phím điều khiển, không thể quay ngược đầu.

Điểm số cập nhật chính xác khi ăn thức ăn, điểm cao được ghi lại và lưu trữ vào tệp.

Game over hoạt động đúng khi có va chạm với thân rắn hoặc chướng ngại vật.

## 3.2.2. Lỗi gặp phải

Vị trí thức ăn trùng với thân rắn: Thỉnh thoảng thức ăn được sinh ở vị trí trùng với thân rắn do logic chưa tối ưu trong kiểm tra.

Tốc độ không đều: Khi điểm số cao, tốc độ tăng quá nhanh, gây khó khăn không mong muốn cho người chơi.

Lỗi tệp lưu trữ điểm cao: Nếu tệp high\_score.txt bị xóa hoặc hỏng, trò chơi không thể khởi tạo điểm cao đúng cách.

## 3.2.3. Phân tích

Các lỗi chủ yếu do logic kiểm tra chưa đầy đủ hoặc thiếu cơ chế xử lý ngoại lệ (ví dụ: trường hợp không có tệp lưu trữ).

Phần tốc độ cần được kiểm soát để giữ cân bằng giữa thử thách và trải nghiệm người chơi.

# CHƯƠNG 4: CẢI TIẾN VÀ MỞ RỘNG

# 4.1. Đề xuất cải tiến

Thêm chế độ chơi mới:

* Chế độ dễ: Tốc độ rắn không tăng, thích hợp cho người mới chơi.
* Chế độ khó: Rắn dài hơn ngay từ đầu hoặc tốc độ tăng nhanh hơn.
* Chế độ thử thách: Thêm yếu tố thời gian giới hạn hoặc nhiệm vụ cụ thể như ăn đủ thức ăn trong thời gian quy định.

Nâng cao đồ họa:

* Thay thế rắn và chướng ngại vật bằng hình ảnh hoặc sprite chi tiết.
* Thêm hiệu ứng chuyển động và âm thanh đặc biệt khi ăn mồi hoặc game over.

Lưu điểm cao trên nền tảng đám mây:

* Đồng bộ hóa điểm cao trực tuyến để người chơi có thể cạnh tranh với bạn bè hoặc toàn cầu.

# 4.2. Tích hợp các yếu tố thú vị

Chướng ngại vật động: Thêm chướng ngại vật di chuyển để tăng độ khó và tính bất ngờ.

Tăng tốc độ rắn theo thời gian: Thay vì tăng tốc sau mỗi lần ăn mồi, tốc độ tăng dần theo thời gian chơi, tạo áp lực liên tục.

Nhiều loại thức ăn: Thêm các loại thức ăn khác nhau với hiệu ứng đặc biệt (ví dụ: làm rắn thu nhỏ, đảo ngược hướng điều khiển).

# PHẦN KẾT LUẬN VÀ KHUYẾN NGHỊ

# 1. Kết luận

Đề tài Snake Game bằng Python đã đạt được mục tiêu xây dựng một trò chơi cơ bản, hoàn chỉnh và đáp ứng yêu cầu đề ra.

Quá trình nghiên cứu và phát triển giúp cải thiện kỹ năng lập trình Python, đặc biệt trong việc sử dụng thư viện pygame và áp dụng thuật toán.

Snake Game là một bước khởi đầu hữu ích trong việc học lập trình game, mở ra nhiều cơ hội phát triển các sản phẩm phức tạp hơn.

# 2. Khuyến nghị

Đối với sinh viên: Tiếp tục nghiên cứu thêm về các kỹ thuật đồ họa, âm thanh và cách tối ưu hóa game.

Đối với các dự án tiếp theo: Tập trung vào cải tiến tính năng của trò chơi hoặc phát triển các game mới dựa trên nền tảng kiến thức đã học.

Gợi ý áp dụng: Tích hợp game vào các ứng dụng giáo dục hoặc xây dựng thư viện mã nguồn mở để chia sẻ với cộng đồng.

# TÀI LIỆU THAM KHẢO

[1] Mark Lutz, *Learning Python*, 5th edition, the United States of America, 2013.

[2] A. Karpathy, *Deep Reinforcement Learning: Pong from Pixels*, Deep Learning Course, 2016.

[3] V. Mnih et al., *Human-level control through deep reinforcement learning*, Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[4] Pygame Community, *Pygame Documentation*, Pygame, 2024

[5] M. Bhatti, M. A. Khan, và M. A. Khan, *Exploration of Reinforcement Learning to Play Snake Game*, 2020 International Conference on Engineering and Emerging Technologies (ICEET), Lahore, Pakistan, 2020, tr. 1–6

[6] M. Bhatti, M. A. Khan, và M. A. Khan, *A Deep Q-Learning based approach applied to the Snake game*, 2021 International Conference on Artificial Intelligence (ICAI), Islamabad, Pakistan, 2021, tr. 1–6

[7] T. Ruscica, *Snake Game Python Tutorial*, freeCodeCamp.org, 2018

[8] *How to Make a Snake Game in Python*, The Python Code, 2024

[9] *Snake Game in Python – Using Pygame module*, GeeksforGeeks, 2024

[10] *Python - Snake Game Using pygame Module*, Tutorials Point, 2024