**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**HỌC PHẦN: LẬP TRÌNH ỨNG DỤNG VỚI PYTHON**

**MÃ ĐỀ THI: 42**

**TÊN ĐỀ TÀI: XÂY DỰNG TRÒ CHƠI BẮN GÀ**

**LỚP TÍN CHỈ: LTUDP.03.K13.08.LH.C04.1\_LT**

**Giảng viên hướng dẫn: Nguyễn Viết Hùng**

**Danh sách sinh viên thực hiện: Nhóm 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20223709 | Hoàng Hải Đăng | DCCNTT.13.10.19 |
| **2** | 20223713 | Phạm Thị Phương Thảo | DCCNTT.13.10.19 |
| **3** | 20223631 | Nguyễn Thị Phương Nguyên | DCCNTT.13.10.19 |
| **4** | 20223433 | Hoàng Minh Thanh | DCCNTT.13.10.19 |
| **5** | 20223017 | Phạm Tùng Lâm | DCCNTT.13.10.19 |

**Bắc Ninh - 2025**

MỤC LỤC

[PHẦN MỞ ĐẦU 1](#_Toc192887843)

[1.1. Giới thiệu môn học 1](#_Toc192887844)

[1.2. Giới thiệu đề tài 1](#_Toc192887845)

[*1.2.1. Lý do chọn đề tài* 1](#_Toc192887846)

[*1.2.2. Mục tiêu của đề tài* 2](#_Toc192887847)

[*1.2.3. Ý nghĩa của đề tài* 3](#_Toc192887848)

[1.3. Phân công nhiệm vụ 4](#_Toc192887849)

[CHƯƠNG II. CƠ SỞ LÝ THUYẾT 6](#_Toc192887850)

[2.1. Giới thiệu các công cụ được sử dụng để xây dựng hệ thống 6](#_Toc192887851)

[*2.1.1. Python* 6](#_Toc192887852)

[*2.1.2. Pygame* 6](#_Toc192887853)

[*2.1.3. SQLite* 7](#_Toc192887854)

[*2.1.4. Công cụ thiết kế đồ họa* 7](#_Toc192887855)

[2.2. Giới thiệu các module/thư viện Python được sử dụng trong đề tài 8](#_Toc192887856)

[*2.2.1. Pygame* 8](#_Toc192887857)

[*2.2.2. Sys* 8](#_Toc192887858)

[*2.2.3. Random* 9](#_Toc192887859)

[*2.2.4. Math* 9](#_Toc192887860)

[*2.2.5. Sqlite3* 9](#_Toc192887861)

[CHƯƠNG III. XÂY DỰNG CHƯƠNG TRÌNH 11](#_Toc192887862)

[3.1. Mô tả cấu trúc dữ liệu của chương trình 11](#_Toc192887863)

[*3.1.1. Dữ liệu trong bộ nhớ (In-memory Data)* 11](#_Toc192887864)

[*3.1.2. Dữ liệu lưu trữ vĩnh viễn (Database)* 11](#_Toc192887865)

[*3.1.3. File dữ liệu (Tài nguyên hình ảnh)* 12](#_Toc192887866)

[3.2. Mô tả cấu trúc mã nguồn chương trình và giải thích ý nghĩa các file mã nguồn 15](#_Toc192887867)

[*3.2.1. Phần khai báo biến toàn cục* 15](#_Toc192887868)

[*3.2.2. Phần định nghĩa lớp* 15](#_Toc192887869)

[*3.2.3. Phần hàm hỗ trợ* 22](#_Toc192887870)

[*3.2.4. Phần vòng lặp chính (run\_game)* 26](#_Toc192887871)

[3.3. Mô tả các giao diện (GUI) của chương trình đã xây dựng 30](#_Toc192887872)

[*3.3.1. Menu chính (show\_menu)* 31](#_Toc192887873)

[*3.3.2. Màn hình nhập tên (get\_player\_name)* 32](#_Toc192887874)

[*3.3.3. Màn hình trò chơi (run\_game)* 32](#_Toc192887875)

[*3.3.4. Màn hình Game Over (game\_over)* 33](#_Toc192887876)

[*3.3.5. Màn hình bảng xếp hạng (show\_scores)* 34](#_Toc192887877)

[CHƯƠNG IV. KẾT QUẢ THỰC NGHIỆM CHƯƠNG TRÌNH 35](#_Toc192887878)

[4.1. Mô tả kết quả kiểm thử các chức năng chương trình với tối thiểu 2 mẫu dữ liệu đầu vào 35](#_Toc192887879)

[*4.1.1. Kịch bản 1: Người chơi mới, chơi đến Level 3 và thua* 35](#_Toc192887880)

[*4.1.2. Kịch bản 2: Người chơi có kinh nghiệm, chơi đến Level 5 và thua* 36](#_Toc192887881)

[*4.1.3. Tổng hợp kết quả* 38](#_Toc192887882)

[CHƯƠNG V. KẾT LUẬN 39](#_Toc192887883)

[5.1. Các nội dung đã đạt được 39](#_Toc192887884)

[*5.1.1. Hoàn thiện một trò chơi hoàn chỉnh:* 39](#_Toc192887885)

[*5.1.2. Ứng dụng thành công kiến thức từ môn học:* 39](#_Toc192887886)

[*5.1.3. Giao diện và trải nghiệm người dùng:* 39](#_Toc192887887)

[*5.1.4. Kiểm thử và hiệu suất:* 39](#_Toc192887888)

[*5.1.5. Kỹ năng làm việc nhóm:* 39](#_Toc192887889)

[5.2. Các nội dung chưa đạt được 40](#_Toc192887890)

[*5.2.1. Thiếu âm thanh:* 40](#_Toc192887891)

[*5.2.2. Chưa có hệ thống power-up hoặc vật phẩm hỗ trợ:* 40](#_Toc192887892)

[*5.2.3. Hiệu suất chưa tối ưu ở cấp độ cao:* 40](#_Toc192887893)

[*5.2.4. Thiếu màn chơi riêng hoặc boss cuối:* 40](#_Toc192887894)

[*5.2.5. Giao diện đơn giản:* 40](#_Toc192887895)

[5.3. Dự kiến phát triển 41](#_Toc192887896)

[*5.3.1. Tích hợp âm thanh:* 41](#_Toc192887897)

[*5.3.2. Thêm hệ thống power-up:* 41](#_Toc192887898)

[*5.3.3. Tối ưu hóa hiệu suất:* 41](#_Toc192887899)

[*5.3.4. Phát triển màn chơi và boss cuối:* 41](#_Toc192887900)

[*5.3.5. Nâng cấp giao diện:* 41](#_Toc192887901)

[*5.3.6. Mở rộng tính năng multiplayer:* 42](#_Toc192887902)

[*5.3.7. Phát hành và chia sẻ:* 42](#_Toc192887903)

[TÀI LIỆU THAM KHẢO 43](#_Toc192887904)

**PHẦN MỞ ĐẦU**

**1.1. Giới thiệu môn học**

Môn học "Lập trình ứng dụng với Python" là một môn học quan trọng trong chương trình đào tạo ngành Công nghệ Thông tin, được thiết kế nhằm cung cấp cho sinh viên những kiến thức và kỹ năng cần thiết để phát triển các ứng dụng thực tế bằng ngôn ngữ lập trình Python. Với sự phổ biến ngày càng tăng của Python trong cộng đồng lập trình viên toàn cầu, môn học này không chỉ tập trung vào việc giảng dạy cú pháp cơ bản mà còn hướng đến việc ứng dụng Python vào các dự án thực tiễn, từ phát triển phần mềm đơn giản đến các ứng dụng phức tạp như trò chơi, hệ thống quản lý dữ liệu, và các công cụ tự động hóa.

Môn học được xây dựng với mục tiêu giúp sinh viên hiểu rõ cách sử dụng Python – một ngôn ngữ lập trình bậc cao, dễ học, và có tính linh hoạt cao – để giải quyết các bài toán thực tế. Trong quá trình học, sinh viên được làm quen với các thư viện mạnh mẽ của Python, chẳng hạn như Pygame (dành cho phát triển trò chơi 2D), SQLite (quản lý cơ sở dữ liệu nhẹ), và các module chuẩn như random, math, sys. Những công cụ này cho phép sinh viên không chỉ viết mã nguồn mà còn tích hợp các yếu tố đồ họa, sự kiện người dùng, và lưu trữ dữ liệu vào trong ứng dụng của mình.

Ngoài việc cung cấp kiến thức lý thuyết, môn "Lập trình ứng dụng với Python" còn chú trọng đến thực hành thông qua các bài tập và dự án nhóm. Các dự án này yêu cầu sinh viên áp dụng tư duy logic, kỹ năng lập trình hướng đối tượng, và khả năng làm việc nhóm để tạo ra những sản phẩm hoàn chỉnh. Dự án phát triển trò chơi "Bắn Gà" (Shooting Chicken) mà nhóm chúng tôi thực hiện là một ví dụ tiêu biểu, phản ánh sự kết hợp giữa lý thuyết và thực hành, đồng thời thể hiện những gì chúng tôi đã học được từ môn học này. Qua đó, sinh viên không chỉ nắm vững cách viết mã mà còn rèn luyện kỹ năng phân tích vấn đề, thiết kế hệ thống, và tối ưu hóa hiệu suất – những yếu tố quan trọng để chuẩn bị cho sự nghiệp lập trình viên trong tương lai.

**1.2. Giới thiệu đề tài**

*1.2.1. Lý do chọn đề tài*

Việc lựa chọn đề tài "Xây dựng trò chơi Bắn Gà bằng Python và Pygame" trong môn "Lập trình ứng dụng với Python" xuất phát từ nhiều lý do mang tính học thuật, thực tiễn và cá nhân. Thứ nhất, trò chơi điện tử từ lâu đã trở thành một lĩnh vực quen thuộc và hấp dẫn trong đời sống giải trí, đặc biệt với giới trẻ. Những trò chơi kinh điển như "Chicken Invaders" hay "Galaga" không chỉ mang lại niềm vui mà còn khơi dậy sự tò mò về cách chúng được lập trình và vận hành. Điều này đã thôi thúc nhóm chúng tôi chọn phát triển một trò chơi tương tự để khám phá sâu hơn về lập trình ứng dụng, đặc biệt là trong lĩnh vực đồ họa 2D.

Thứ hai, Python – ngôn ngữ chính được sử dụng trong môn học – là một công cụ lý tưởng cho việc phát triển ứng dụng nhờ tính đơn giản, dễ tiếp cận và hệ sinh thái thư viện phong phú. Thư viện Pygame, được giới thiệu trong môn "Lập trình ứng dụng với Python", cung cấp các công cụ mạnh mẽ để xử lý đồ họa, sự kiện người dùng, và vòng lặp trò chơi, giúp nhóm chúng tôi có thể biến ý tưởng thành hiện thực một cách hiệu quả. Việc chọn đề tài này không chỉ giúp củng cố kiến thức về Python mà còn mở rộng hiểu biết về cách xây dựng một ứng dụng tương tác thời gian thực – một kỹ năng quan trọng trong ngành công nghệ thông tin.

Thứ ba, đề tài "Bắn Gà" có độ phức tạp phù hợp với trình độ của nhóm và thời gian thực hiện dự án trong khuôn khổ môn học. Trò chơi yêu cầu xử lý nhiều khía cạnh như điều khiển phi thuyền, bắn đạn, tạo kẻ thù với các đặc điểm khác nhau, kiểm tra va chạm, và lưu trữ điểm số – tất cả đều là những bài toán thực tế mà sinh viên được khuyến khích giải quyết trong môn "Lập trình ứng dụng với Python". Đây là một thách thức vừa sức nhưng cũng đầy thú vị, cho phép nhóm học hỏi và hoàn thiện kỹ năng lập trình.

Cuối cùng, lý do cá nhân cũng đóng vai trò quan trọng trong việc chọn đề tài. Các thành viên trong nhóm đều có niềm yêu thích với trò chơi điện tử và mong muốn tự tay tạo ra một sản phẩm mang dấu ấn riêng. Việc phát triển "Bắn Gà" không chỉ đáp ứng yêu cầu của môn học mà còn mang lại sự hứng khởi, động lực học tập, và niềm tự hào khi hoàn thành một ứng dụng có thể chia sẻ với bạn bè, người thân. Đây cũng là cơ hội để nhóm trải nghiệm quy trình phát triển phần mềm thực tế, từ lên ý tưởng, lập kế hoạch, đến triển khai và kiểm thử.

*1.2.2. Mục tiêu của đề tài*

Đề tài "Xây dựng trò chơi Bắn Gà bằng Python và Pygame" được thực hiện với các mục tiêu cụ thể như sau:

Mục tiêu học thuật: Áp dụng các kiến thức từ môn "Lập trình ứng dụng với Python", bao gồm lập trình hướng đối tượng, sử dụng thư viện Pygame để xử lý đồ họa, và tích hợp SQLite để quản lý dữ liệu, nhằm xây dựng một ứng dụng hoàn chỉnh.

Mục tiêu kỹ thuật: Phát triển một trò chơi hoạt động mượt mà với các tính năng cơ bản như điều khiển phi thuyền bằng chuột, bắn đạn để tiêu diệt kẻ thù, tăng cấp độ theo thời gian, và lưu trữ điểm số cao của người chơi trong cơ sở dữ liệu.

Mục tiêu sáng tạo: Thiết kế các loại kẻ thù đa dạng (gà và các quái vật) với đặc điểm riêng biệt về tốc độ, loại đạn, và độ khó, nhằm tạo nên một trò chơi hấp dẫn và thử thách.

Mục tiêu nhóm: Rèn luyện kỹ năng làm việc nhóm thông qua việc phân công nhiệm vụ rõ ràng, phối hợp giữa các thành viên để tích hợp mã nguồn, và giải quyết các vấn đề kỹ thuật phát sinh trong quá trình phát triển.

Mục tiêu dài hạn: Đặt nền tảng cho việc nghiên cứu và phát triển các dự án lập trình phức tạp hơn trong tương lai, đồng thời tích lũy kinh nghiệm thực tế để áp dụng vào các ứng dụng khác ngoài trò chơi.

*1.2.3. Ý nghĩa của đề tài*

Đề tài "Xây dựng trò chơi Bắn Gà bằng Python và Pygame" mang lại nhiều ý nghĩa quan trọng:

Ý nghĩa học tập: Dự án là cơ hội để nhóm áp dụng kiến thức từ môn "Lập trình ứng dụng với Python" vào thực tế, từ đó hiểu sâu hơn về cách sử dụng Python và các thư viện liên quan. Việc phát triển trò chơi giúp cải thiện kỹ năng lập trình, tư duy logic, và khả năng tổ chức mã nguồn một cách khoa học.

Ý nghĩa thực tiễn: "Bắn Gà" không chỉ là một bài tập học thuật mà còn là một sản phẩm giải trí có thể sử dụng thực tế. Ứng dụng này có thể được chia sẻ với bạn bè, người thân hoặc cộng đồng, góp phần nâng cao nhận thức về khả năng của Python trong việc phát triển các ứng dụng tương tác.

Ý nghĩa cá nhân: Đối với các thành viên trong nhóm, việc hoàn thành dự án là một trải nghiệm quý báu, giúp phát triển kỹ năng mềm như giao tiếp, quản lý thời gian, và làm việc dưới áp lực. Đây cũng là một kỷ niệm đáng nhớ trong quá trình học tập tại trường.

Ý nghĩa xã hội: Dù là một trò chơi đơn giản, dự án này thể hiện tiềm năng sáng tạo của sinh viên trong việc sử dụng công nghệ để tạo ra các sản phẩm hữu ích. Điều này góp phần thúc đẩy phong trào học lập trình và phát triển phần mềm tại Việt Nam, đồng thời khuyến khích các thế hệ sinh viên tiếp theo tham gia vào lĩnh vực công nghệ thông tin.

**1.3. Phân công nhiệm vụ**

Dự án "Xây dựng trò chơi Bắn Gà bằng Python và Pygame" được thực hiện bởi nhóm 5 thành viên. Nhóm trưởng chịu trách nhiệm chính với khối lượng công việc lớn, trong khi các thành viên khác hỗ trợ các nhiệm vụ nhỏ hơn. Phân công cụ thể như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **Mã Sinh Viên** | **Phân công nhiệm vụ** |
| 1 | Hoàng Hải Đăng | 20223709 | - Thiết kế cấu trúc trò chơi, lập trình các lớp chính (Ship, Bullet, Chicken, Monster1-4), viết logic chính (run\_game), tối ưu hiệu suất, tích hợp mã nguồn, và sửa lỗi.  - Đảm bảo chất lượng toàn bộ dự án. |
| 2 | Phạm Thị Phương Thảo | 20223713 | - Chuẩn bị hình ảnh phi thuyền, gà, và nền; kiểm tra giao diện đồ họa.  - Hỗ trợ phần hình ảnh cơ bản. |
| 3 | Nguyễn Thị Phương Nguyên | 20223631 | - Xây dựng menu chính (show\_menu) và kiểm tra điều hướng.  - Đảm bảo menu hoạt động ổn định. |
| 4 | Hoàng Minh Thanh | 20223433 | - Tích hợp SQLite (init\_db, save\_score) và kiểm tra lưu điểm.  - Hỗ trợ quản lý điểm số. |
| 5 | Phạm Tùng Lâm | 20223017 | - Thiết kế màn hình nhập tên (get\_player\_name) và kiểm tra nhập liệu.  - Đảm bảo nhập tên dễ dàng. |

Các thành viên trong nhóm đều tham gia thảo luận, đóng góp ý tưởng, và hỗ trợ lẫn nhau trong suốt quá trình thực hiện. Sự phối hợp chặt chẽ giữa các thành viên là yếu tố then chốt giúp nhóm hoàn thành dự án đúng thời hạn và đạt được các mục tiêu đề ra trong môn "Lập trình ứng dụng với Python".

**CHƯƠNG II. CƠ SỞ LÝ THUYẾT**

**2.1. Giới thiệu các công cụ được sử dụng để xây dựng hệ thống**

Để phát triển trò chơi "Bắn Gà" trong môn "Lập trình ứng dụng với Python", nhóm chúng tôi đã sử dụng một số công cụ quan trọng, từ ngôn ngữ lập trình, thư viện đồ họa, đến hệ quản trị cơ sở dữ liệu. Các công cụ này không chỉ hỗ trợ quá trình lập trình mà còn đảm bảo ứng dụng hoạt động ổn định, hiệu quả và đáp ứng được các yêu cầu của đề tài. Dưới đây là mô tả chi tiết về từng công cụ:

*2.1.1. Python*

Giới thiệu: Python là một ngôn ngữ lập trình bậc cao, mã nguồn mở, được phát triển bởi Guido van Rossum và ra mắt lần đầu vào năm 1991. Với cú pháp đơn giản, dễ đọc và khả năng hỗ trợ đa nền tảng, Python đã trở thành một trong những ngôn ngữ phổ biến nhất thế giới, được sử dụng trong nhiều lĩnh vực như phát triển web, phân tích dữ liệu, trí tuệ nhân tạo, và phát triển trò chơi. Trong môn "Lập trình ứng dụng với Python", Python được chọn làm ngôn ngữ chính nhờ tính linh hoạt và hệ sinh thái thư viện phong phú.

Vai trò trong đề tài: Python là nền tảng để nhóm viết toàn bộ mã nguồn của trò chơi "Bắn Gà". Từ việc định nghĩa các lớp đối tượng (như Ship, Chicken, Monster), xử lý logic trò chơi, đến tích hợp các thư viện khác, Python cung cấp môi trường lập trình thuận tiện và hiệu quả. Phiên bản Python 3.x được sử dụng trong dự án để tận dụng các tính năng mới nhất và đảm bảo tương thích với các thư viện như Pygame.

Công cụ hỗ trợ: Nhóm sử dụng Visual Studio Code – một trình soạn thảo mã nguồn mạnh mẽ, hỗ trợ gỡ lỗi, gợi ý cú pháp, và tích hợp terminal – để viết và chạy mã Python.

*2.1.2. Pygame*

Giới thiệu: Pygame là một thư viện mã nguồn mở được thiết kế dành riêng cho việc phát triển trò chơi 2D trên Python. Được xây dựng dựa trên thư viện SDL (Simple DirectMedia Layer), Pygame cung cấp các công cụ để xử lý đồ họa, âm thanh, và sự kiện đầu vào từ người dùng (chuột, bàn phím). Thư viện này được giới thiệu trong môn "Lập trình ứng dụng với Python" như một công cụ lý tưởng để sinh viên thực hành lập trình đồ họa và tương tác thời gian thực.

Vai trò trong đề tài: Pygame là trái tim của hệ thống đồ họa trong trò chơi "Bắn Gà". Nó được sử dụng để:

Tạo cửa sổ trò chơi với kích thước 1500x800 pixel (pygame.display.set\_mode()).

Tải và hiển thị các hình ảnh như phi thuyền, kẻ thù, đạn, và nền (pygame.image.load(), DISPLAYSURF.blit()).

Xử lý sự kiện người dùng như di chuyển chuột để điều khiển phi thuyền và nhấp chuột để bắn đạn (pygame.event.get(), MOUSEBUTTONDOWN).

Điều khiển vòng lặp trò chơi và giới hạn tốc độ khung hình ở 60 FPS (pygame.time.Clock().tick(60)).

Ưu điểm: Pygame dễ sử dụng, phù hợp với trình độ sinh viên, và không yêu cầu cấu hình phần cứng cao, giúp nhóm triển khai dự án trên các máy tính thông thường.

*2.1.3. SQLite*

Giới thiệu: SQLite là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) nhẹ, không cần máy chủ riêng, được lưu trữ dưới dạng một tệp duy nhất trên đĩa. Được phát triển bởi D. Richard Hipp vào năm 2000, SQLite nổi bật nhờ tính đơn giản, hiệu quả, và khả năng tích hợp dễ dàng với Python thông qua module sqlite3. Đây là công cụ lý tưởng cho các ứng dụng nhỏ như trò chơi hoặc dự án học thuật.

Vai trò trong đề tài: SQLite được sử dụng để lưu trữ điểm số cao của người chơi trong trò chơi "Bắn Gà". Cụ thể:

Tạo tệp cơ sở dữ liệu scores.db để lưu thông tin người chơi (tên và điểm số).

Quản lý bảng scores với các cột id, name, và score, cho phép lưu và truy xuất dữ liệu để hiển thị bảng xếp hạng (show\_scores).

Đảm bảo dữ liệu được lưu vĩnh viễn, ngay cả khi trò chơi tắt, giúp người chơi theo dõi thành tích của mình qua nhiều lần chơi.

Ưu điểm: SQLite không yêu cầu cài đặt phức tạp, phù hợp với quy mô nhỏ của dự án, và dễ dàng tích hợp với Python.

*2.1.4. Công cụ thiết kế đồ họa*

Giới thiệu: Để tạo tài nguyên hình ảnh cho trò chơi, nhóm sử dụng các công cụ thiết kế đồ họa như Adobe Photoshop và các nguồn tài nguyên miễn phí từ internet như OpenGameArt.org. Các hình ảnh bao gồm phi thuyền, gà, quái vật, đạn, và nền được chỉnh sửa để phù hợp với kích thước và phong cách của trò chơi.

Vai trò trong đề tài: Các công cụ này hỗ trợ:

Tạo và chỉnh sửa hình ảnh như phithuyen.png, chicken.png, monster1-4.png, laser.png, v.v.

Điều chỉnh kích thước hình ảnh (ví dụ: phi thuyền 100x100, gà 50x50) để phù hợp với giao diện trò chơi.

Đảm bảo tính thẩm mỹ và đồng nhất về mặt hình ảnh, nâng cao trải nghiệm người chơi.

Lợi ích: Giúp nhóm tự do sáng tạo hoặc tận dụng tài nguyên có sẵn, tiết kiệm thời gian mà vẫn đảm bảo chất lượng đồ họa.

**2.2. Giới thiệu các module/thư viện Python được sử dụng trong đề tài**

Trò chơi "Bắn Gà" sử dụng một số module và thư viện Python để giải quyết các nhiệm vụ cụ thể trong quá trình phát triển. Dưới đây là mô tả chi tiết về từng module/thư viện, kèm theo cách chúng được áp dụng trong dự án:

*2.2.1. Pygame*

Giới thiệu: Đây là module chính để phát triển trò chơi 2D, được cài đặt thông qua lệnh pip install pygame. Pygame cung cấp các lớp và hàm để quản lý cửa sổ, đồ họa, sự kiện, và thời gian trong trò chơi.

Ứng dụng trong đề tài:

Tạo giao diện: pygame.display.set\_mode((WINDOWWIDTH, WINDOWHEIGHT)) tạo cửa sổ trò chơi 1500x800 pixel.

Hiển thị đồ họa: pygame.image.load() tải hình ảnh (phi thuyền, kẻ thù, đạn), DISPLAYSURF.blit() vẽ chúng lên màn hình.

Xử lý sự kiện: pygame.event.get() nhận diện hành động của người chơi (nhấn chuột, thoát trò chơi).

Điều khiển tốc độ: pygame.time.Clock().tick(60) giới hạn trò chơi ở 60 khung hình/giây, đảm bảo hiệu suất ổn định.

Quản lý văn bản: pygame.font.Font() hiển thị điểm số và cấp độ trên màn hình.

Nhiệm vụ giải quyết: Pygame chịu trách nhiệm cho toàn bộ phần đồ họa và tương tác người dùng, là nền tảng để xây dựng trải nghiệm trò chơi.

*2.2.2. Sys*

Giới thiệu: Module sys là một module chuẩn của Python, cung cấp các hàm và biến để tương tác với hệ thống, chẳng hạn như thoát chương trình hoặc truy cập tham số dòng lệnh.

Ứng dụng trong đề tài:

Thoát trò chơi: sys.exit() được gọi khi người chơi đóng cửa sổ hoặc chọn "Quit" từ menu, đảm bảo chương trình kết thúc an toàn.

Nhiệm vụ giải quyết: Đảm bảo trò chơi có thể thoát một cách sạch sẽ khi không còn cần thiết, tránh lãng phí tài nguyên hệ thống.

*2.2.3. Random*

Giới thiệu: Module random là module chuẩn của Python, cung cấp các hàm để tạo số ngẫu nhiên, rất hữu ích trong các ứng dụng cần yếu tố bất ngờ như trò chơi.

Ứng dụng trong đề tài:

Vị trí kẻ thù: random.randint(0, WINDOWWIDTH - width) tạo vị trí ngẫu nhiên cho gà và quái vật khi xuất hiện trên màn hình.

Tần suất bắn: random.randint(1, 1000) > 995 xác định xác suất kẻ thù bắn đạn (ví dụ: gà bắn trứng, quái bắn đạn đặc biệt).

Nhiệm vụ giải quyết: Tạo sự ngẫu nhiên và đa dạng trong hành vi của kẻ thù, tăng tính thú vị và khó đoán cho trò chơi.

*2.2.4. Math*

Giới thiệu: Module math cung cấp các hàm toán học cơ bản như sin, cos, logarit, v.v., được sử dụng để tính toán trong lập trình.

Ứng dụng trong đề tài:

Chuyển động zigzag: math.sin(self.angle) \* 2 trong lớp Bullet3 tạo đường đi zigzag cho đạn của Monster3.

Chuyển động sóng: self.speed\_y \* math.sin(self.x / 50) trong lớp Monster3 tạo hiệu ứng di chuyển lên xuống dạng sóng.

Nhiệm vụ giải quyết: Tăng độ phức tạp và độc đáo cho chuyển động của đạn và kẻ thù, nâng cao trải nghiệm chơi.

*2.2.5. Sqlite3*

Giới thiệu: Module sqlite3 là module chuẩn của Python, cho phép tương tác với cơ sở dữ liệu SQLite mà không cần cài đặt thêm phần mềm bên ngoài.

Ứng dụng trong đề tài:

Khởi tạo cơ sở dữ liệu: sqlite3.connect("scores.db") kết nối tới tệp scores.db, c.execute() tạo bảng scores với các cột id, name, score.

Lưu điểm số: c.execute("INSERT INTO scores ...") thêm tên và điểm của người chơi sau mỗi lần chơi.

Truy xuất điểm số: c.execute("SELECT ... ORDER BY score DESC LIMIT 10") lấy top 10 điểm cao để hiển thị bảng xếp hạng.

Nhiệm vụ giải quyết: Quản lý và lưu trữ điểm số vĩnh viễn, cho phép người chơi so sánh thành tích và tăng tính cạnh tranh.

**CHƯƠNG III. XÂY DỰNG CHƯƠNG TRÌNH**

**3.1. Mô tả cấu trúc dữ liệu của chương trình**

Cấu trúc dữ liệu của trò chơi "Bắn Gà" được thiết kế để hỗ trợ các tính năng chính như quản lý đối tượng trong trò chơi, lưu trữ điểm số, và đảm bảo hiệu suất hoạt động. Dữ liệu được chia thành hai loại chính: dữ liệu trong bộ nhớ (in-memory) để xử lý thời gian thực và dữ liệu lưu trữ vĩnh viễn trong cơ sở dữ liệu SQLite. Dưới đây là mô tả chi tiết:

*3.1.1. Dữ liệu trong bộ nhớ (In-memory Data)*

Mô tả: Đây là dữ liệu được lưu trữ tạm thời trong RAM khi trò chơi đang chạy, bao gồm các đối tượng và trạng thái của trò chơi.

Cấu trúc:

Danh sách đối tượng:

ship.bullets: Danh sách các đối tượng Bullet (đạn của phi thuyền), mỗi đối tượng chứa tọa độ (x, y), kích thước (width, height), và tốc độ (speed).

chicken.eggs: Danh sách các đối tượng Egg (trứng của gà), chứa tọa độ và kích thước.

monstersX.eggs (X từ 1 đến 4): Danh sách đạn của các quái vật (Bullet1, Bullet2, Bullet3, Bullet4), mỗi loại có đặc điểm riêng (tốc độ, chuyển động zigzag, kích thước).

chickens, monsters1, monsters2, monsters3, monsters4: Danh sách các đối tượng kẻ thù, mỗi đối tượng chứa tọa độ, kích thước, tốc độ, hướng di chuyển (direction), và máu (health với các quái vật).

Biến trạng thái:

score: Số nguyên lưu điểm số của người chơi, tăng khi tiêu diệt kẻ thù (gà: +1, Monster1: +2, Monster2: +3, Monster3: +4, Monster4: +10).

level: Số nguyên biểu thị cấp độ, tăng sau mỗi 10 giây.

bg\_y1, bg\_y2: Số nguyên lưu vị trí của hai hình nền để tạo hiệu ứng cuộn liên tục.

Vai trò: Dữ liệu trong bộ nhớ được sử dụng để quản lý trạng thái tức thời của trò chơi, như vị trí phi thuyền, đạn, kẻ thù, và các thông số hiển thị trên màn hình.

*3.1.2. Dữ liệu lưu trữ vĩnh viễn (Database)*

Mô tả: Dữ liệu này được lưu trong tệp cơ sở dữ liệu SQLite để ghi lại điểm số cao của người chơi, cho phép truy xuất và hiển thị bảng xếp hạng.

Cấu trúc:

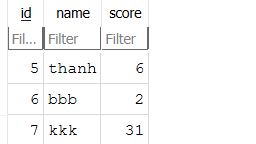
Tệp: scores.db – tệp cơ sở dữ liệu SQLite.

Bảng: scores với các cột:

id (INTEGER PRIMARY KEY AUTOINCREMENT): Khóa chính tự tăng, đánh số thứ tự mỗi bản ghi.

name (TEXT): Tên người chơi, tối đa 255 ký tự.

score (INTEGER): Điểm số đạt được trong một lần chơi.



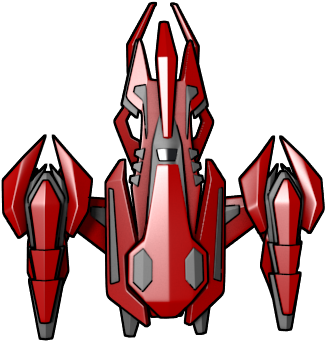
Vai trò: Lưu trữ vĩnh viễn điểm số sau mỗi lần chơi, cho phép hiển thị top 10 người chơi có điểm cao nhất trong mục "View achievements". SQLite được chọn vì tính nhẹ, không cần máy chủ, và phù hợp với quy mô nhỏ của trò chơi.

*3.1.3. File dữ liệu (Tài nguyên hình ảnh)*

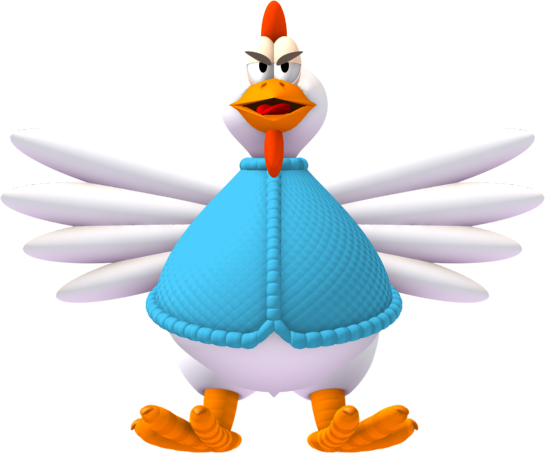
Mô tả: Các tệp hình ảnh được lưu trong thư mục image để hiển thị đồ họa trong trò chơi.

Cấu trúc:

phithuyen.png: Hình ảnh phi thuyền (100x100 pixel):

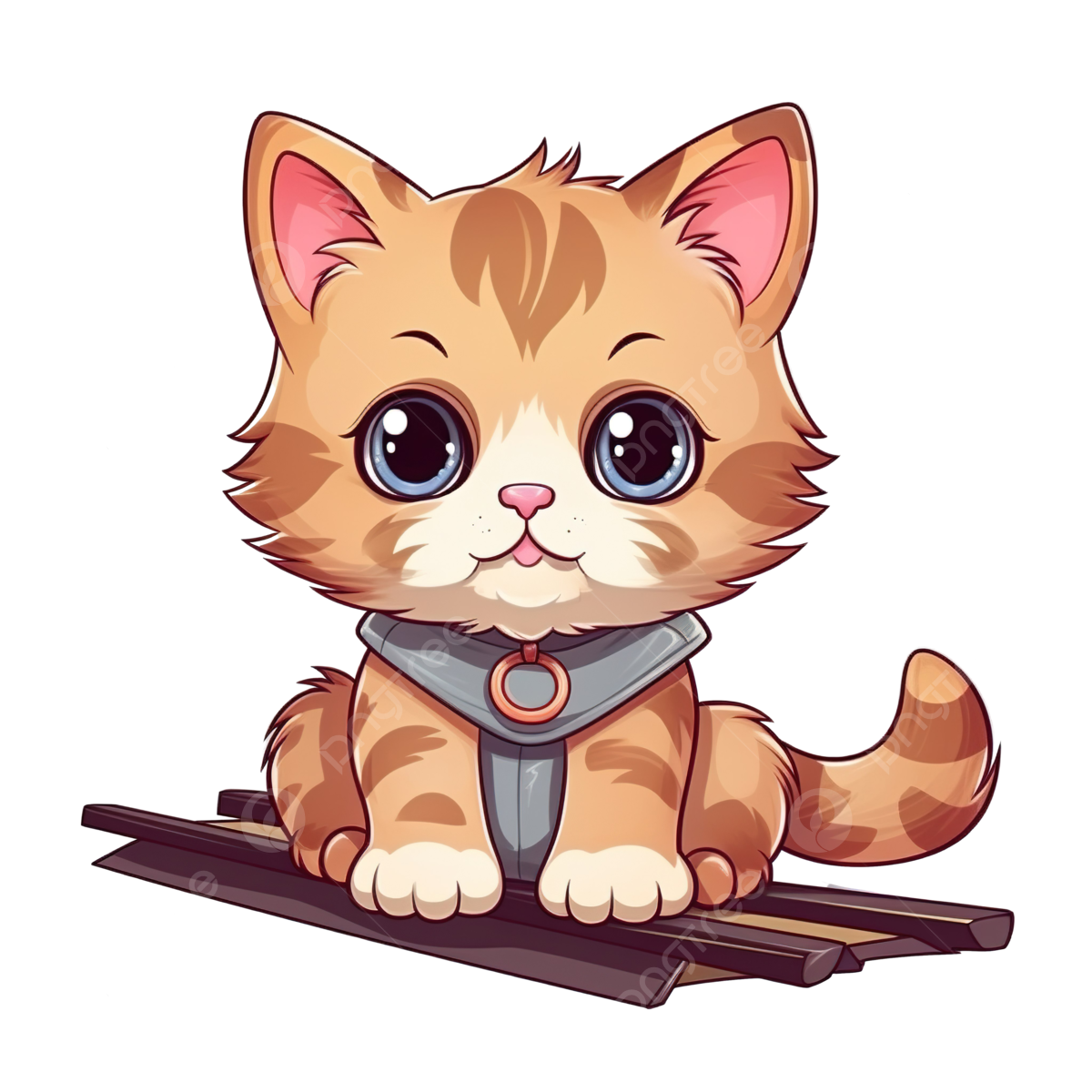


chicken.png: Hình ảnh gà (50x50 pixel):



sau.png, meo.png, cho.png, rong.png: Hình ảnh các quái vật (kích thước lần lượt 60x60, 80x80, 50x50, 100x100):



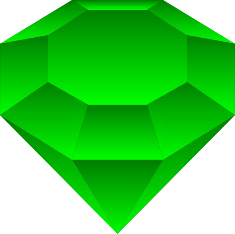


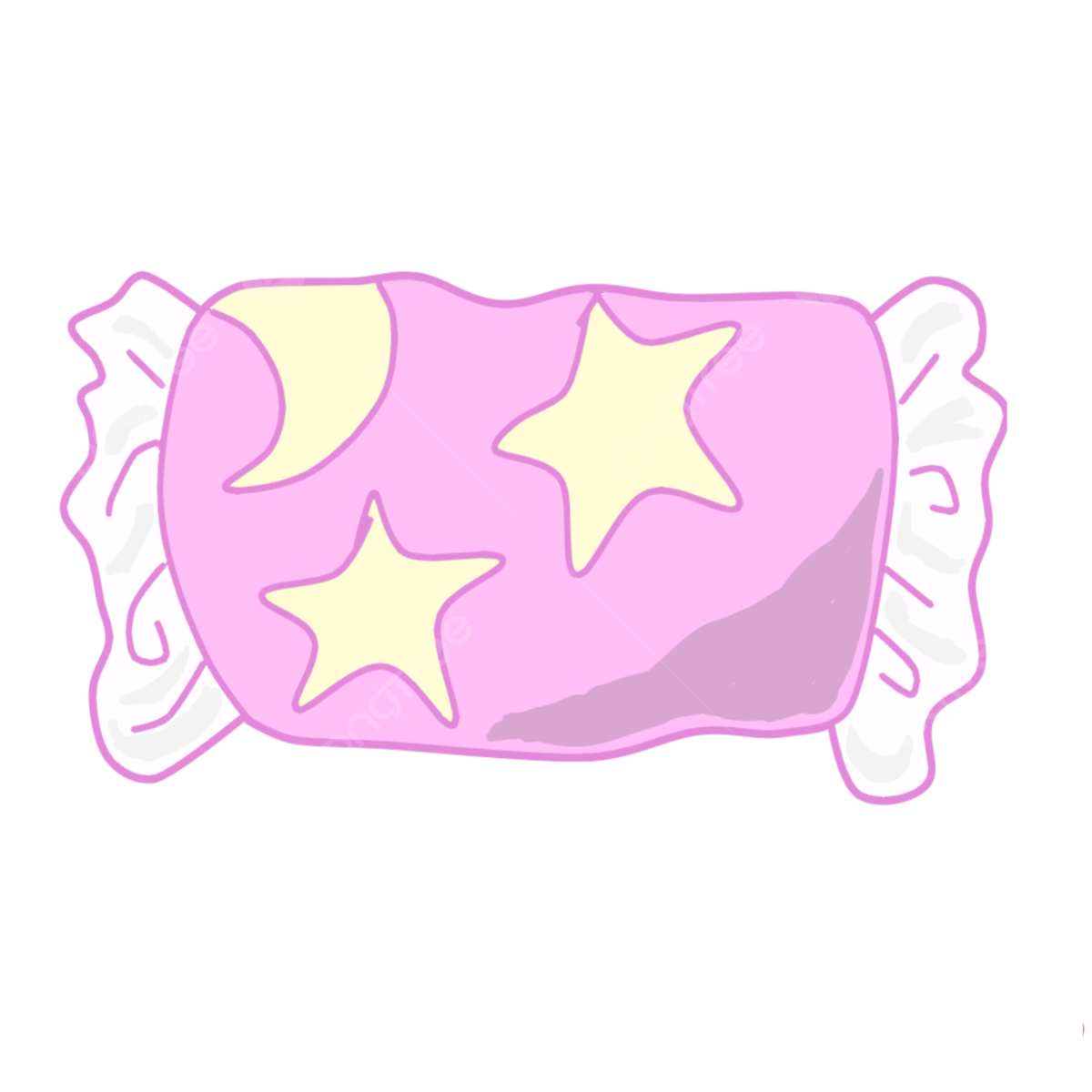


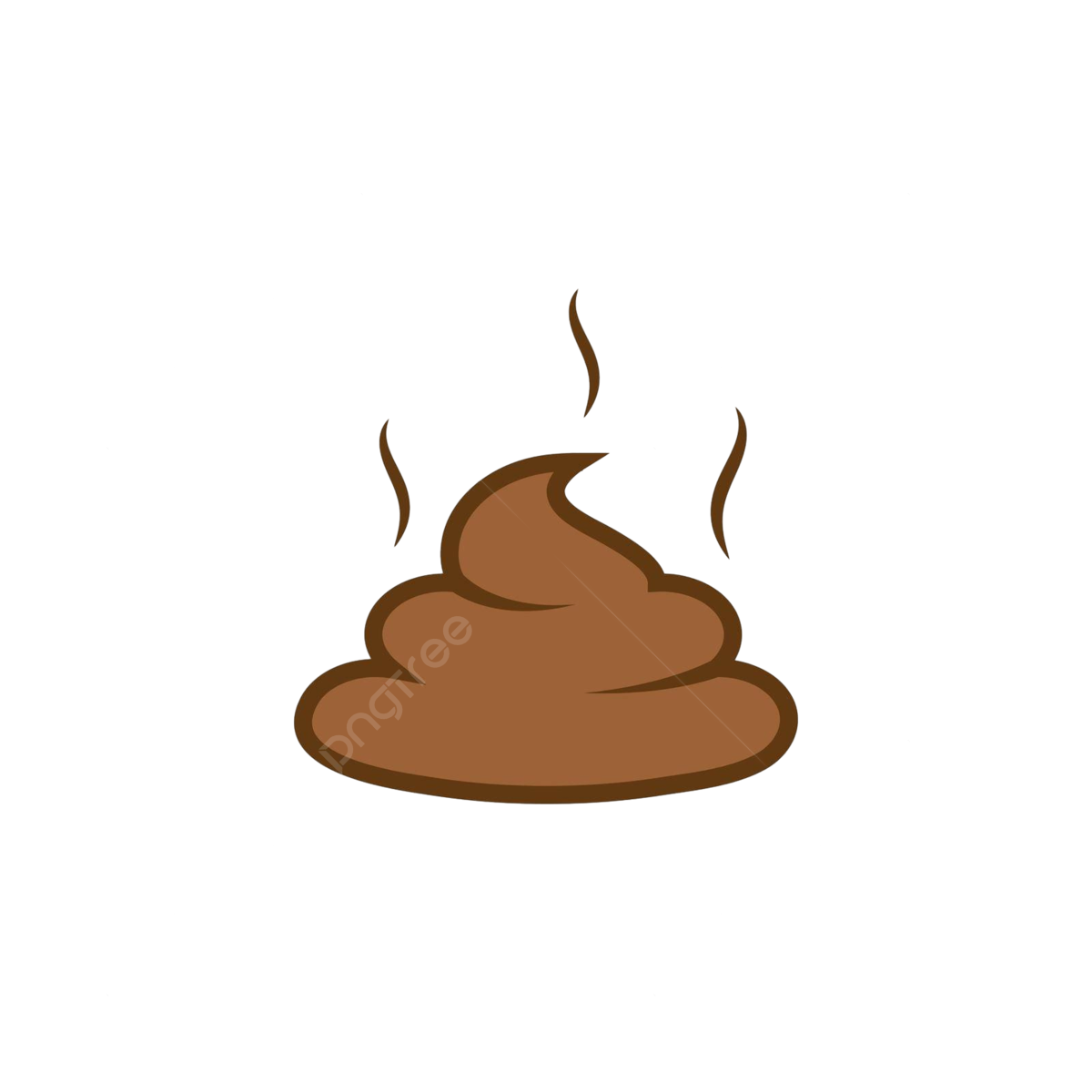


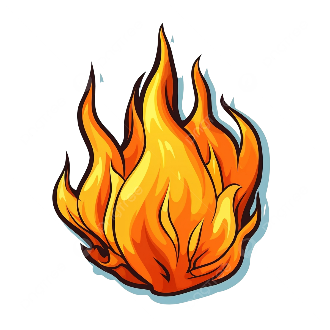
laser.png, ss.png, goi.png, phancho.png, lua.png: Hình ảnh các loại đạn (kích thước từ 10x10 đến 20x20).



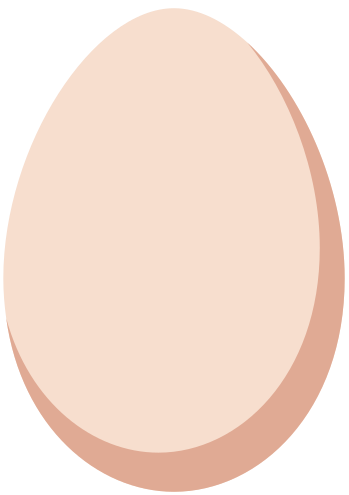








egg.png: Hình ảnh trứng (10x10 pixel):



bg2.jpg: Hình ảnh nền (1500x800 pixel sau khi điều chỉnh):



Vai trò: Cung cấp tài nguyên đồ họa để vẽ các đối tượng và nền, tạo trải nghiệm trực quan cho người chơi.

**3.2. Mô tả cấu trúc mã nguồn chương trình và giải thích ý nghĩa các file mã nguồn**

Mã nguồn của trò chơi "Bắn Gà" được tổ chức trong một tệp Python duy nhất shooting\_chicken.py để đơn giản hóa việc quản lý trong khuôn khổ dự án học thuật. Tệp này được chia thành các phần logic rõ ràng, từ khai báo biến, định nghĩa lớp, đến triển khai vòng lặp trò chơi. Dưới đây là mô tả cấu trúc và ý nghĩa:

*3.2.1. Phần khai báo biến toàn cục*

Nội dung:

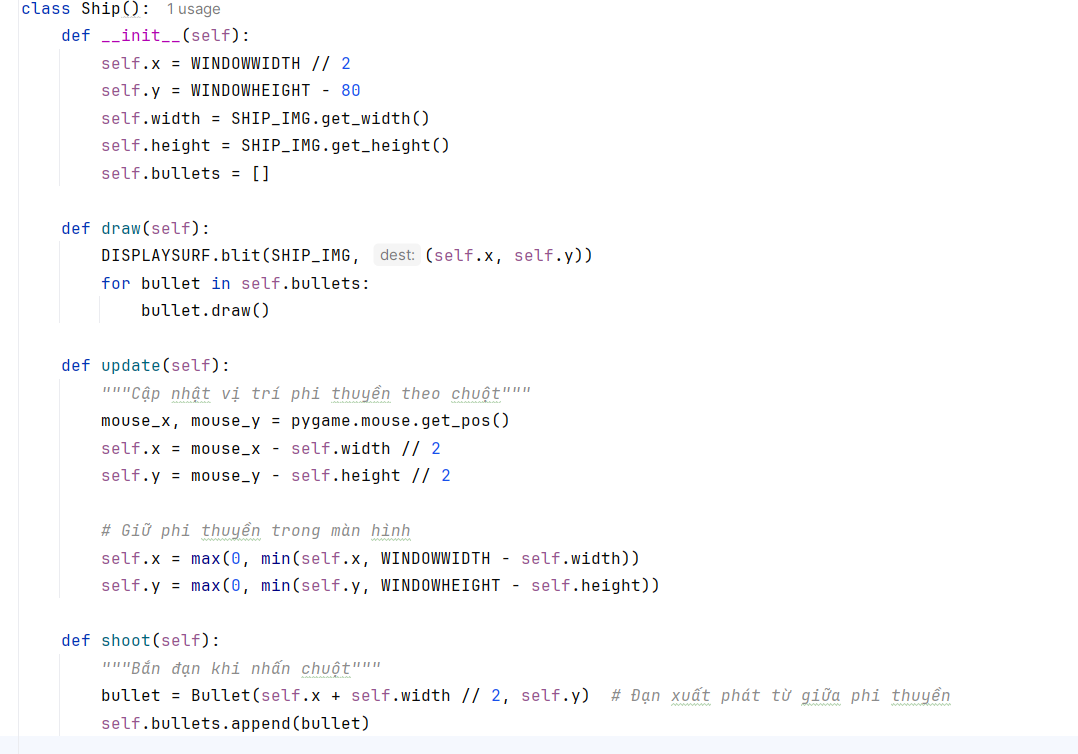
Các hằng số: WINDOWWIDTH = 1500, WINDOWHEIGHT = 800, SHIP\_SPEED = 5, BULLET\_SPEED = 7, CHICKEN\_SPEED = 2, EGG\_SPEED = 2, BG\_SCROLL\_SPEED = 1.

Tải hình ảnh: SHIP\_IMG = pygame.image.load('image\\phithuyen.png'), CHICKEN\_IMG, MONSTER1\_IMG, v.v., và điều chỉnh kích thước bằng pygame.transform.scale().

Ý nghĩa: Định nghĩa các tham số cơ bản của trò chơi (kích thước cửa sổ, tốc độ) và chuẩn bị tài nguyên đồ họa để sử dụng trong các lớp đối tượng.

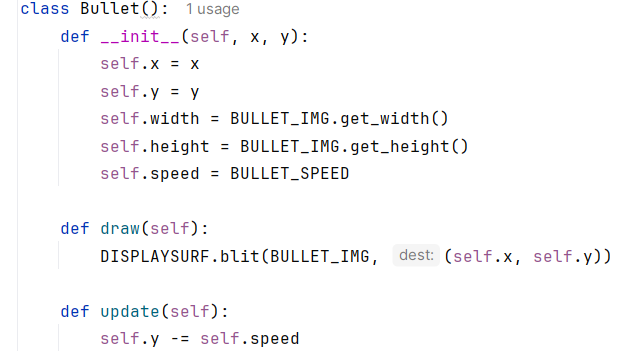
*3.2.2. Phần định nghĩa lớp*

Nội dung:

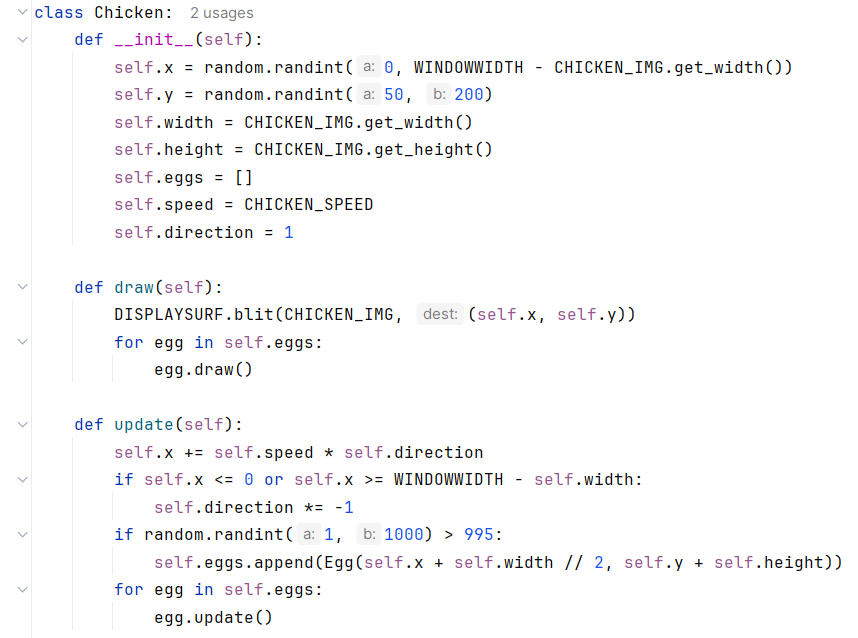
Ship: 

Lớp phi thuyền với thuộc tính (x, y, width, height, bullets) và phương thức update() (di chuyển theo chuột), shoot() (bắn đạn), draw() (vẽ).

Bullet:

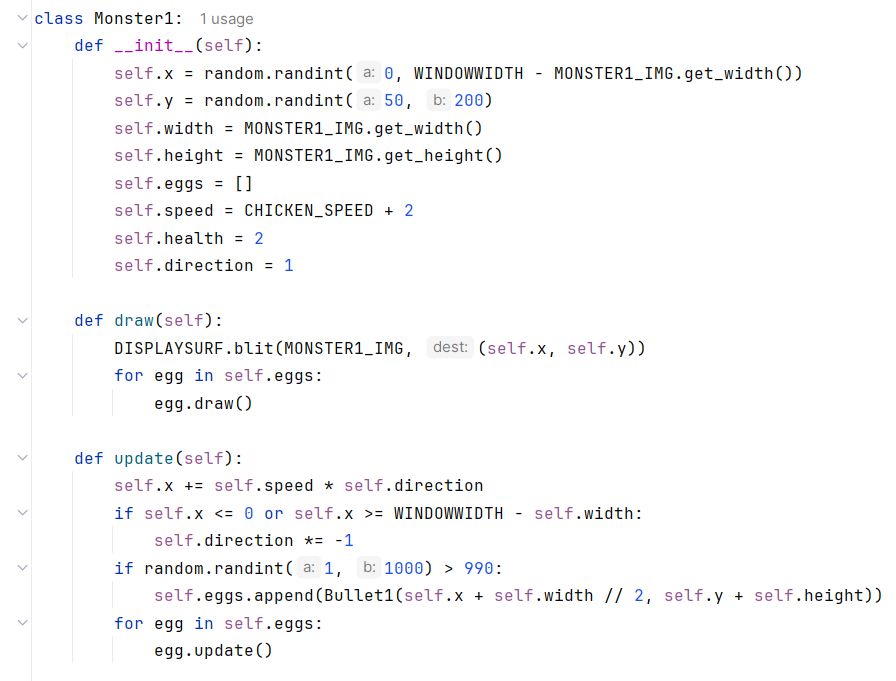


Lớp đạn của phi thuyền với thuộc tính (x, y, speed) và phương thức update() (di chuyển lên), draw().

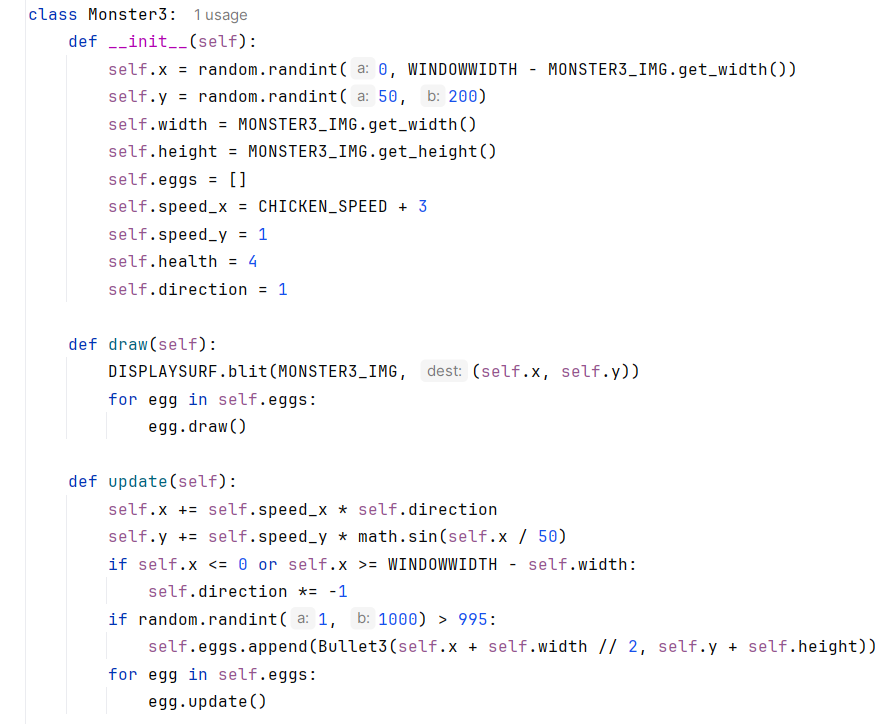
Chicken:

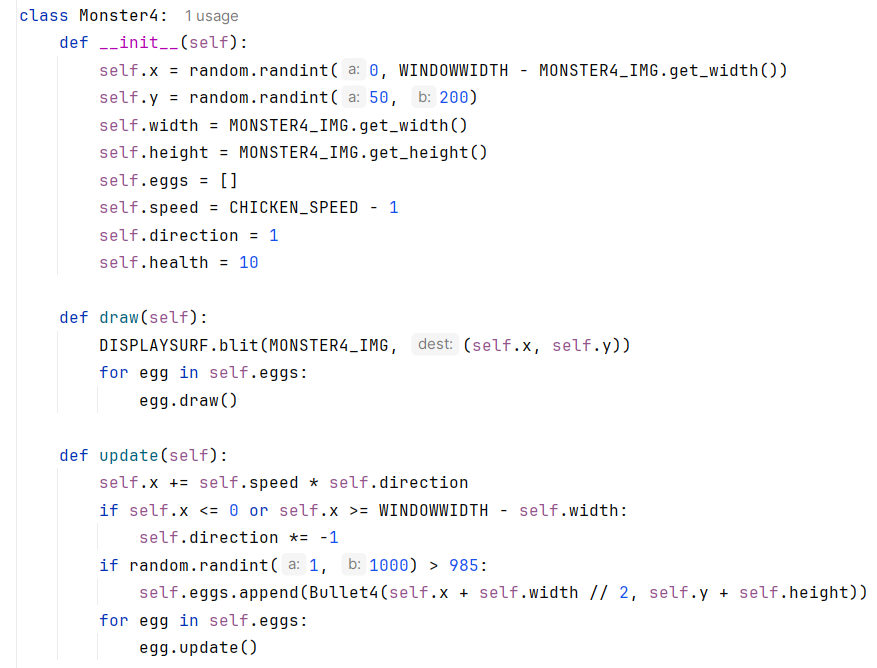
Lớp gà với thuộc tính (x, y, width, height, eggs, speed, direction) và phương thức update() (di chuyển ngang, bắn trứng), draw().

Monster1, Monster2, Monster3, Monster4:





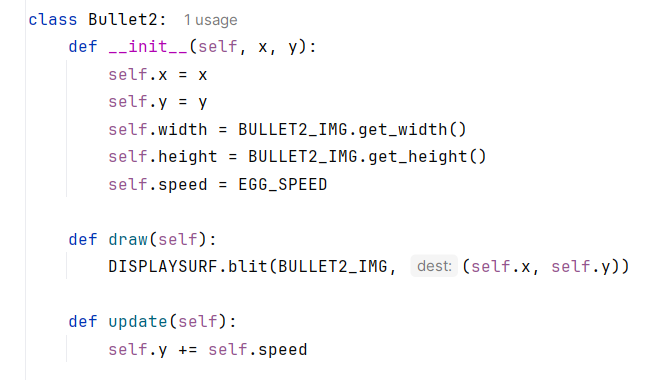


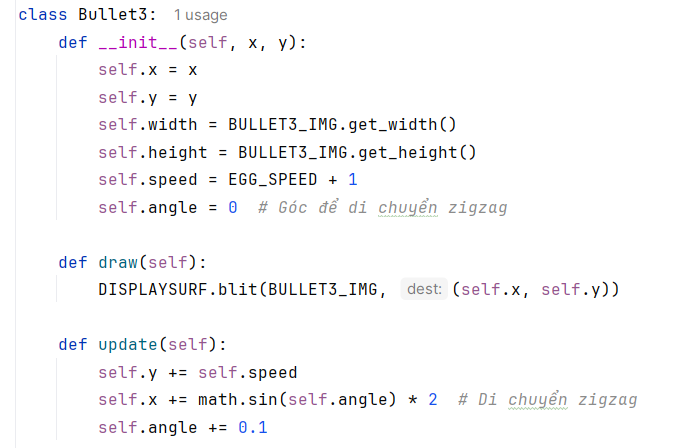


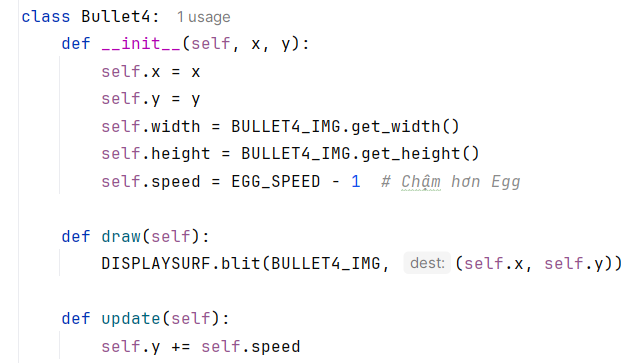
Lớp quái vật với thuộc tính bổ sung health (máu) và đạn riêng (Bullet1-4), phương thức update() có chuyển động đặc biệt (ví dụ: Monster3 di chuyển dạng sóng).

Egg, Bullet1, Bullet2, Bullet3, Bullet4:









Lớp đạn của kẻ thù với các đặc điểm riêng (tốc độ nhanh, zigzag, kích thước lớn).

Ý nghĩa: Xây dựng các đối tượng cơ bản của trò chơi, sử dụng lập trình hướng đối tượng để quản lý hành vi và trạng thái của phi thuyền, kẻ thù, và đạn.

*3.2.3. Phần hàm hỗ trợ*

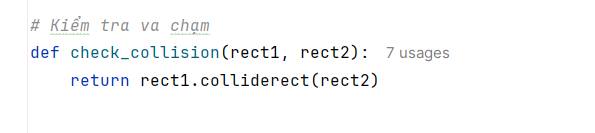
Nội dung:

scroll\_background():



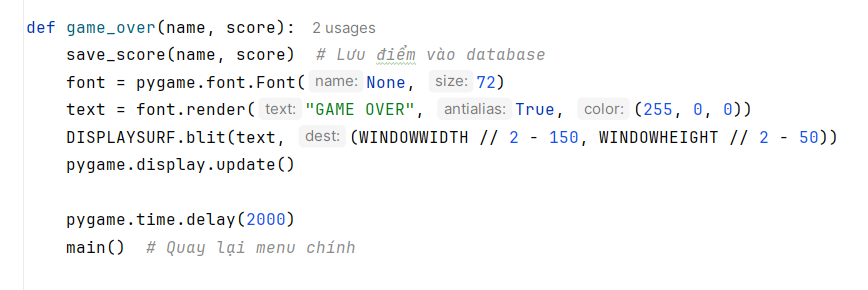
Cuộn nền liên tục bằng cách cập nhật bg\_y1, bg\_y2.

check\_collision():



Kiểm tra va chạm giữa hai hình chữ nhật (pygame.Rect).

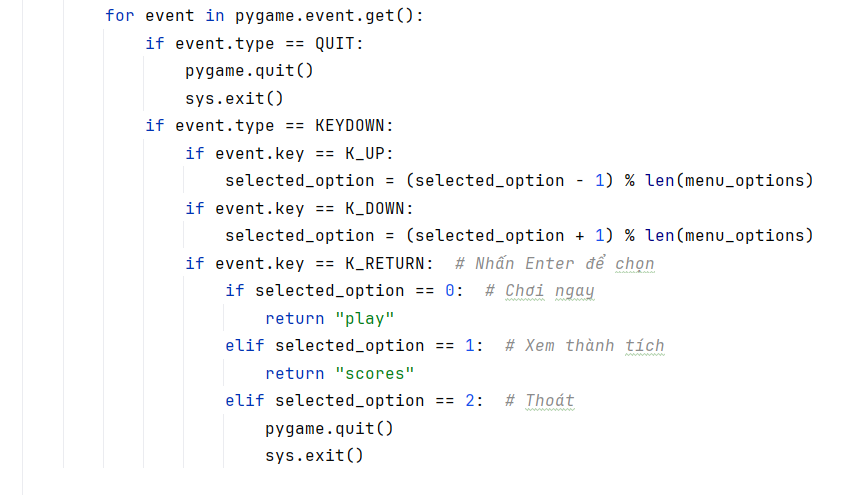
game\_over():

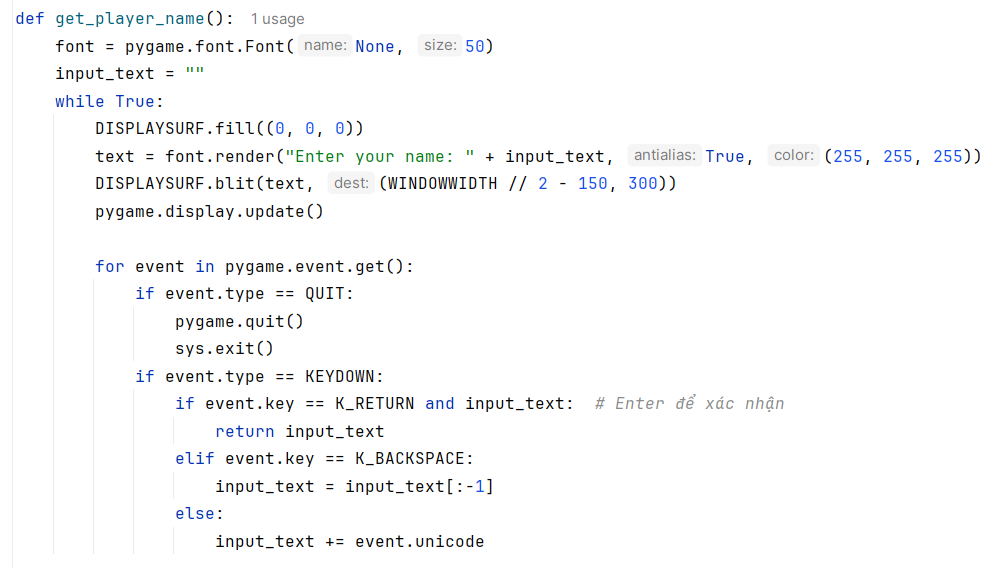


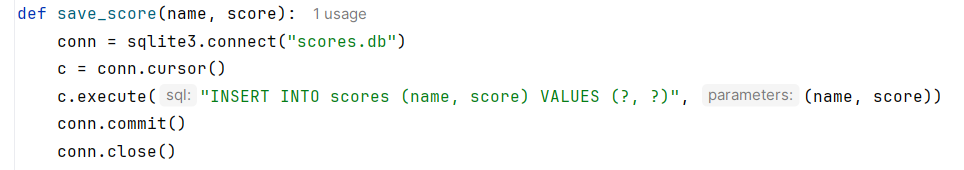
Hiển thị "GAME OVER" và lưu điểm.

show\_menu(), get\_player\_name(), show\_scores():



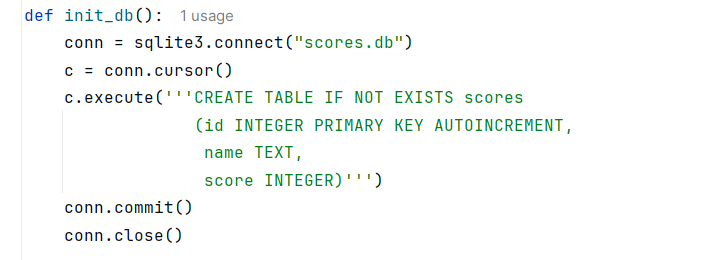


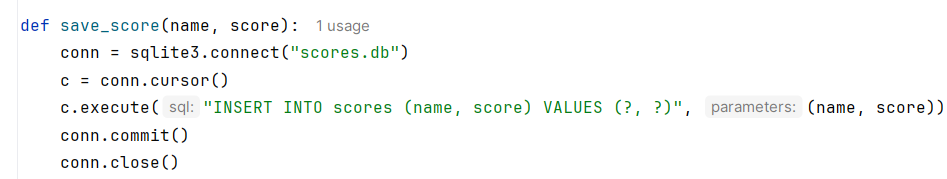




Quản lý giao diện menu, nhập tên, và bảng xếp hạng.

init\_db(), save\_score():



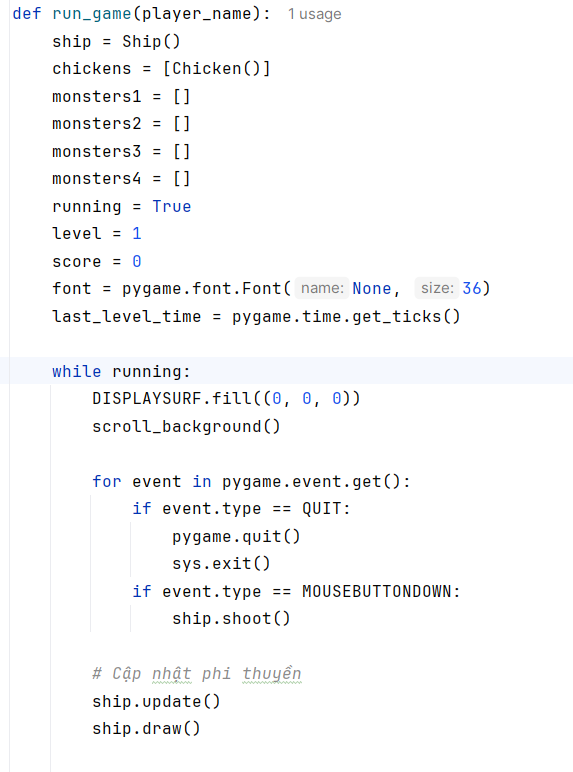


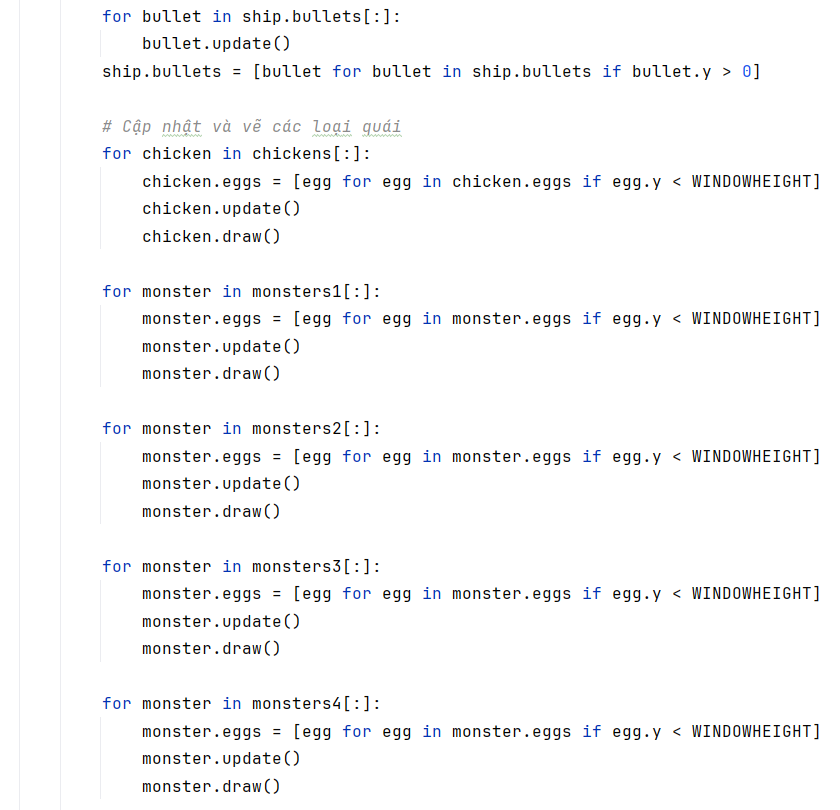
Khởi tạo và lưu điểm vào SQLite.

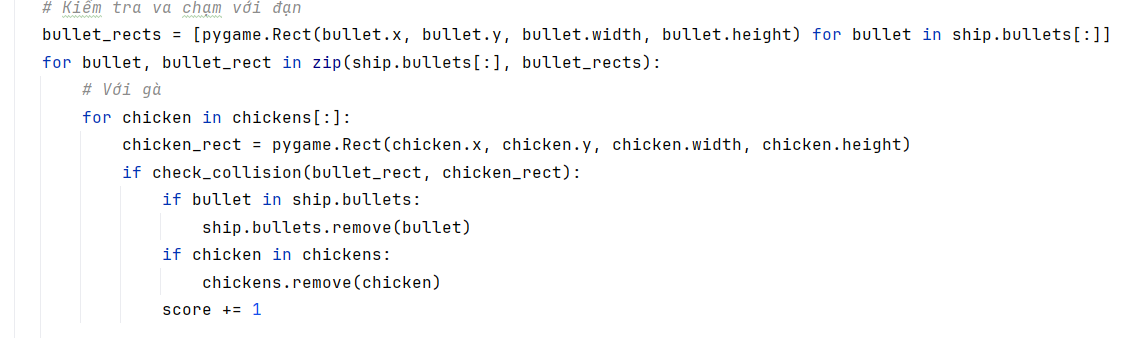
Ý nghĩa: Cung cấp các hàm tiện ích để xử lý logic phụ, giao diện, và dữ liệu, hỗ trợ vòng lặp chính hoạt động trơn tru.

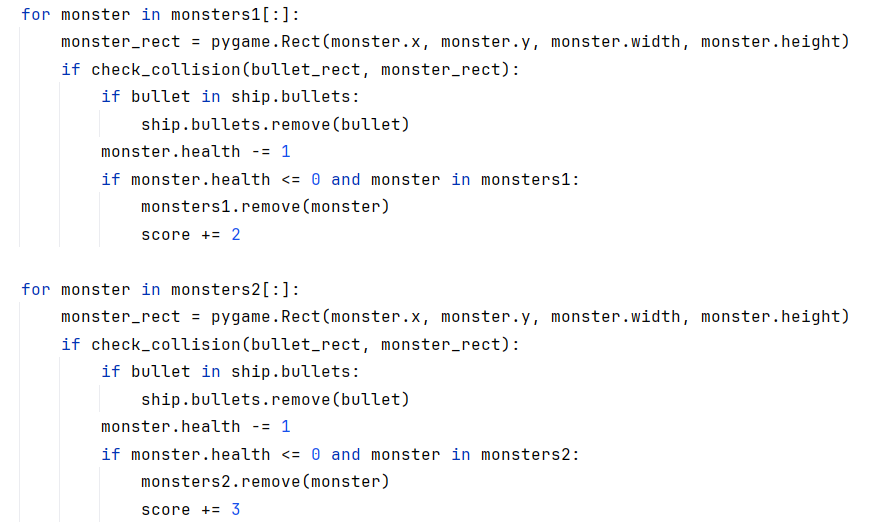
*3.2.4. Phần vòng lặp chính (run\_game)*

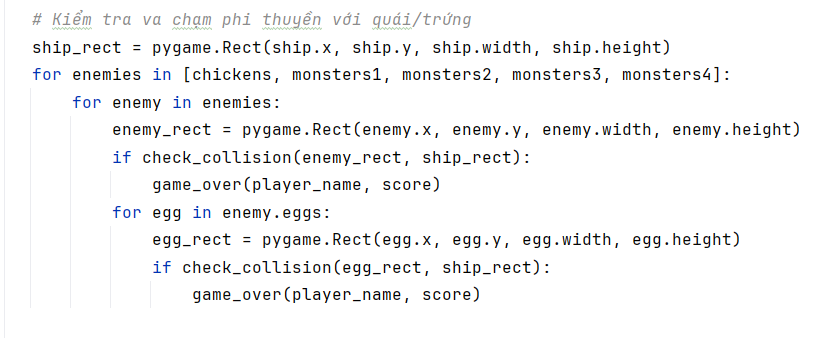
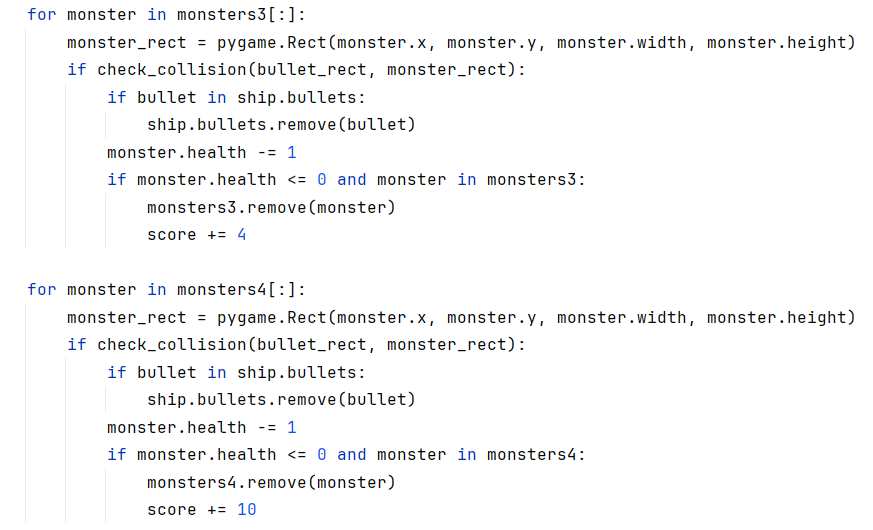
Nội dung:

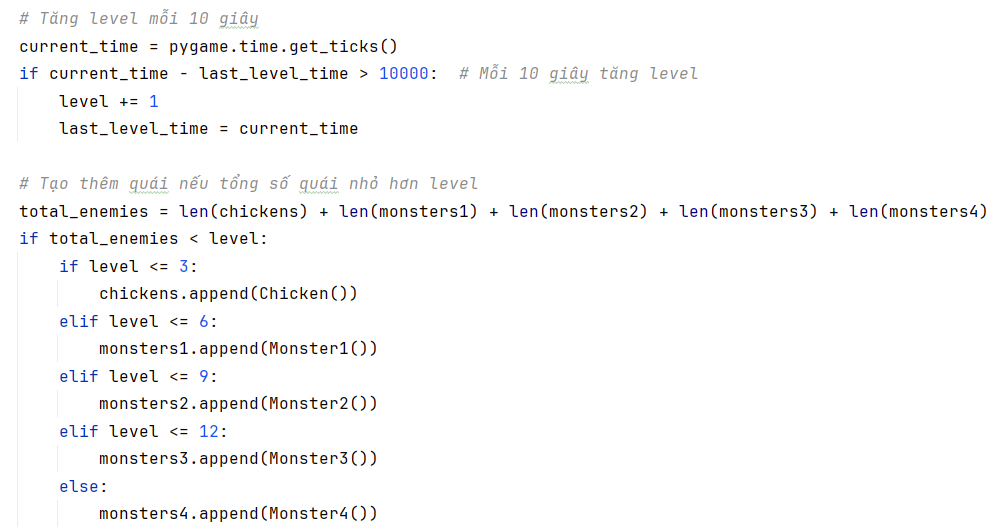


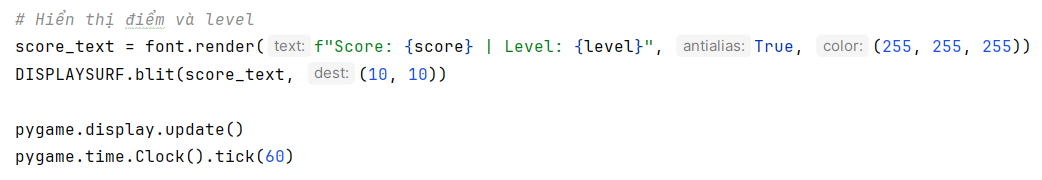












Khởi tạo đối tượng: ship, danh sách kẻ thù (chickens, monsters1-4).

Xử lý sự kiện: Nhấn chuột để bắn, thoát trò chơi.

Cập nhật trạng thái: Di chuyển phi thuyền, đạn, kẻ thù; kiểm tra va chạm; tăng cấp độ mỗi 10 giây.

Vẽ màn hình: Nền, phi thuyền, kẻ thù, đạn, điểm số.

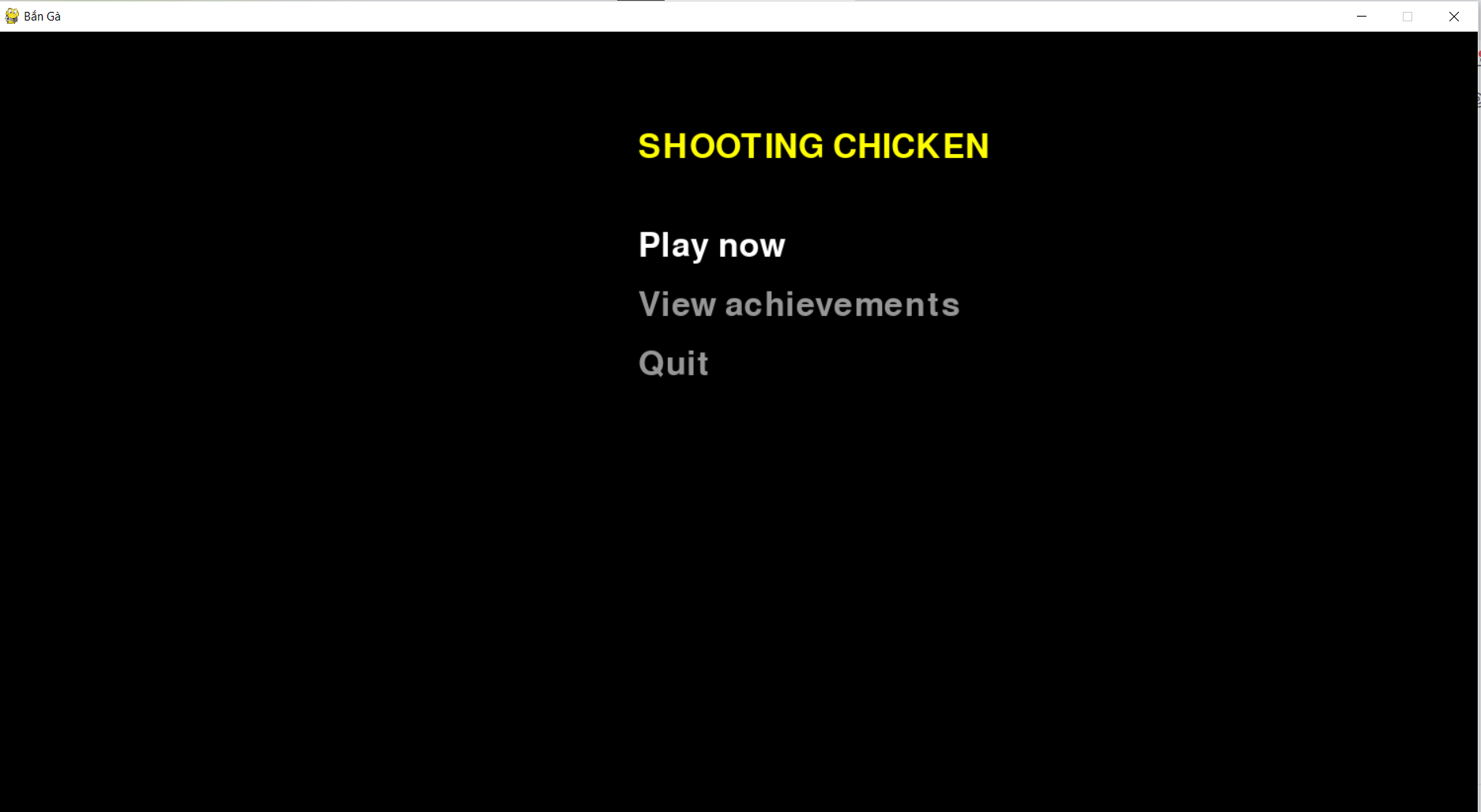
Ý nghĩa: Là trung tâm điều khiển trò chơi, kết nối tất cả các thành phần để tạo ra trải nghiệm chơi hoàn chỉnh.

Tệp mã nguồn: Chỉ có một tệp shooting\_chicken.py chứa toàn bộ mã, phù hợp với quy mô nhỏ của dự án. Nếu mở rộng, có thể tách thành các tệp riêng (ví dụ: classes.py cho lớp, game\_logic.py cho logic).

**3.3. Mô tả các giao diện (GUI) của chương trình đã xây dựng**

Trò chơi "Bắn Gà" sử dụng Pygame để xây dựng các giao diện đồ họa người dùng (GUI) đơn giản nhưng trực quan, bao gồm các màn hình chính sau:

*3.3.1. Menu chính (show\_menu)*

 Mô tả:

Nền màu đen (DISPLAYSURF.fill((0, 0, 0))).

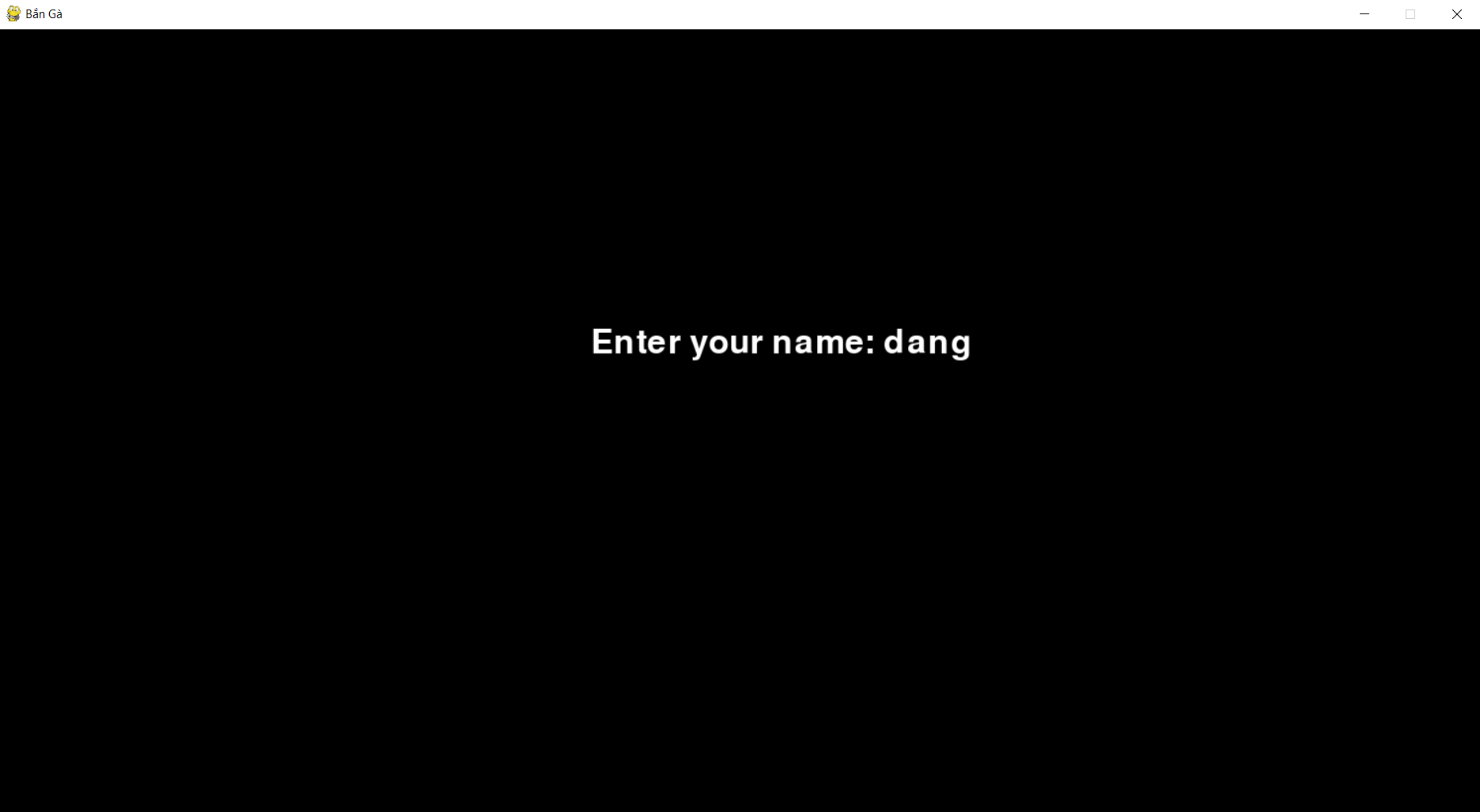
Tiêu đề "SHOOTTING CHIKEN" màu vàng (pygame.font.Font(None, 50)), đặt ở giữa trên cùng.

Ba tùy chọn: "Play now", "View achievements", "Quit", hiển thị dọc, màu trắng khi được chọn, xám khi không chọn.

Tương tác: Dùng phím lên/xuống để điều hướng, phím Enter để chọn.

Vai trò: Điểm khởi đầu của trò chơi, cho phép người chơi bắt đầu, xem điểm, hoặc thoát.

*3.3.2. Màn hình nhập tên (get\_player\_name)*



Mô tả:

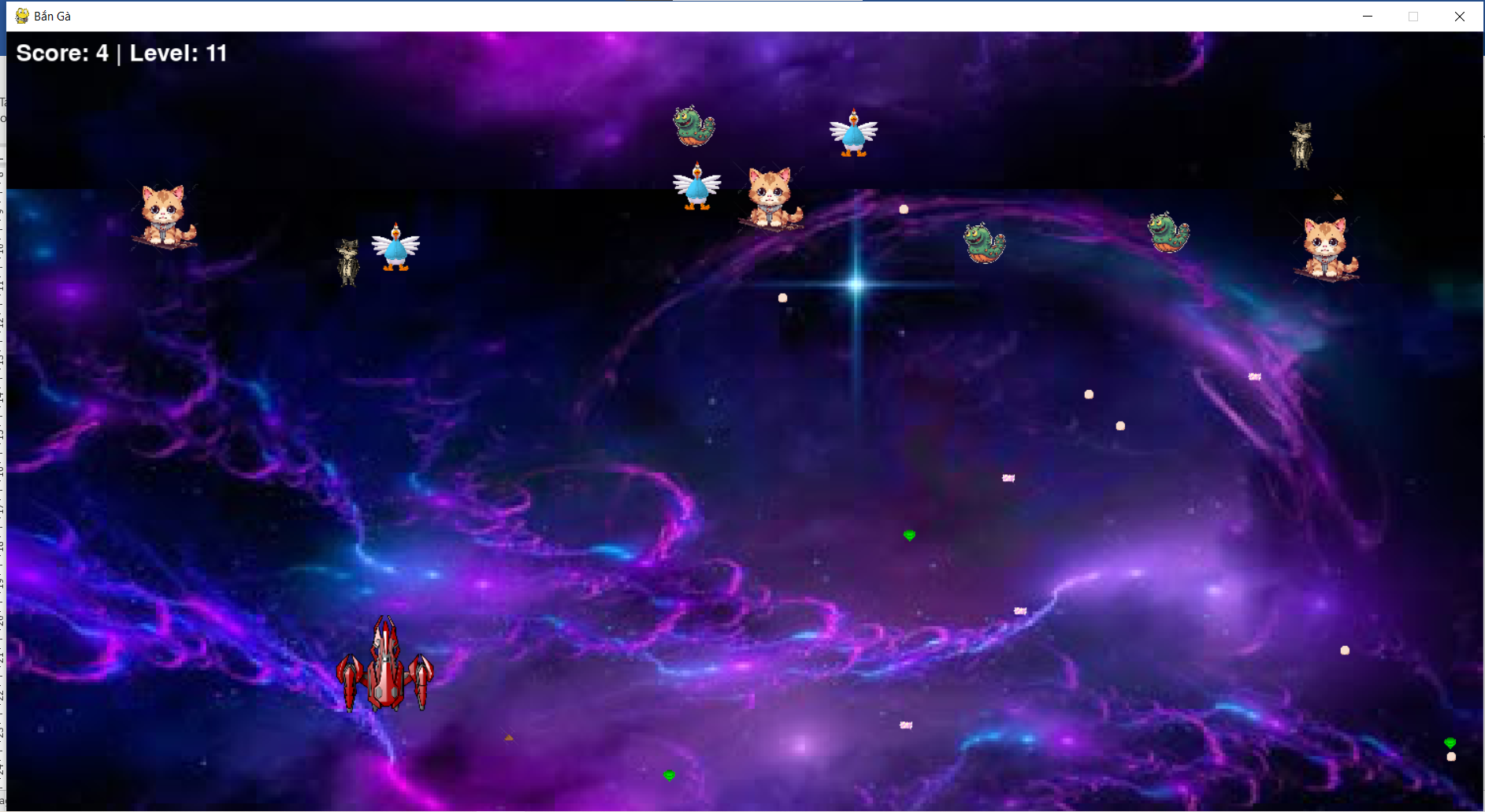
Nền màu đen.

Dòng chữ "Enter your name: " kèm theo tên đang nhập (màu trắng), hiển thị ở giữa màn hình.

Tương tác: Nhập ký tự bằng bàn phím, xóa bằng Backspace, xác nhận bằng Enter.

Vai trò: Thu thập tên người chơi để lưu điểm số.

*3.3.3. Màn hình trò chơi (run\_game)*



Mô tả:

Nền: Hình ảnh bg2.jpg cuộn liên tục từ dưới lên (scroll\_background()).

Phi thuyền: Hình ảnh phithuyen.png di chuyển theo chuột.

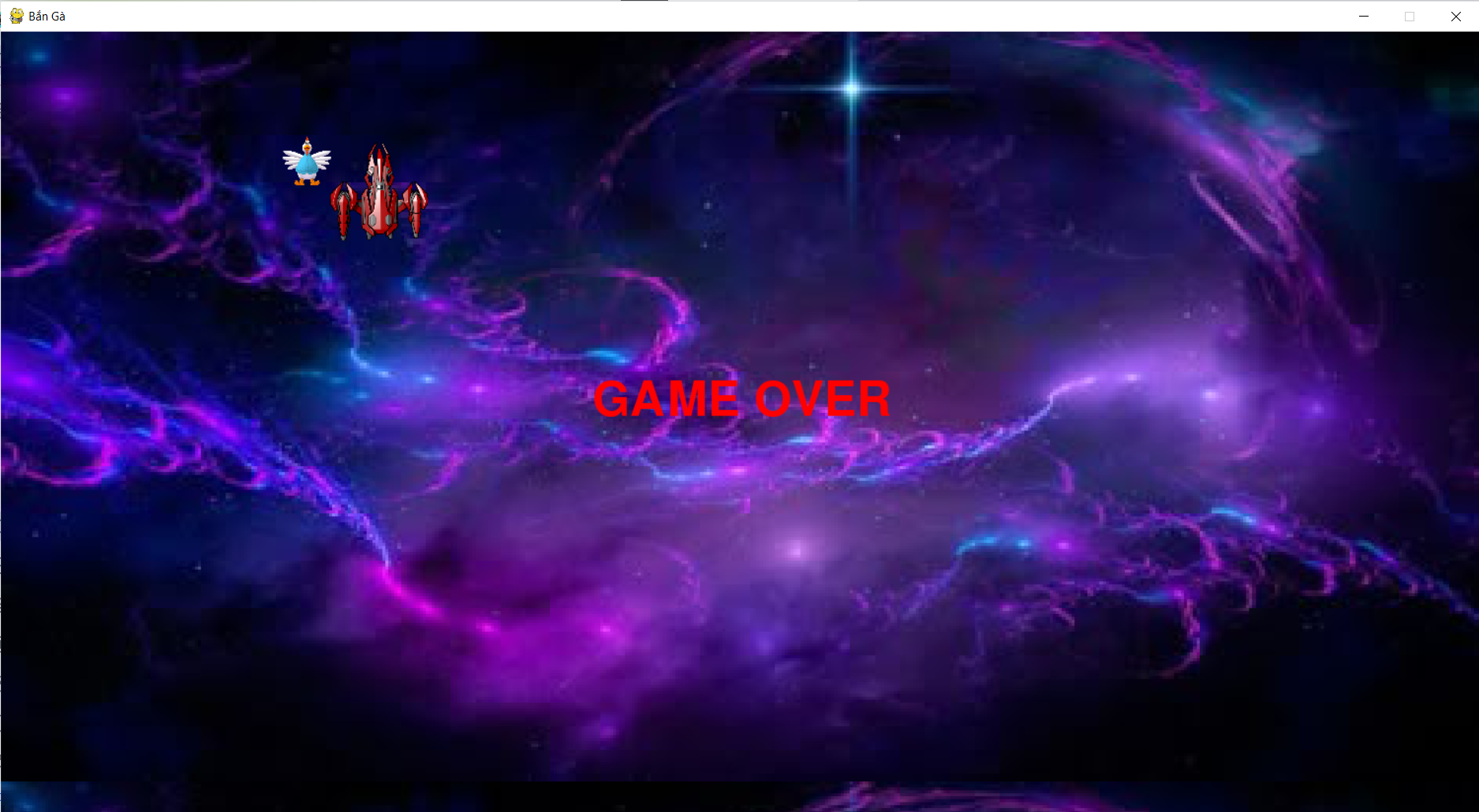
Kẻ thù: Gà (chicken.png) và quái vật (sau.png, meo.png, cho.png, rong.png) xuất hiện từ trên xuống, bắn đạn (egg.png, ss.png, v.v.).

Điểm số và cấp độ: Văn bản "Score: X | Level: Y" màu trắng, góc trên bên trái.

Tương tác: Di chuột để điều khiển phi thuyền, nhấp chuột để bắn.

Vai trò: Màn hình chính nơi diễn ra gameplay, hiển thị mọi hoạt động của trò chơi.

*3.3.4. Màn hình Game Over (game\_over)*

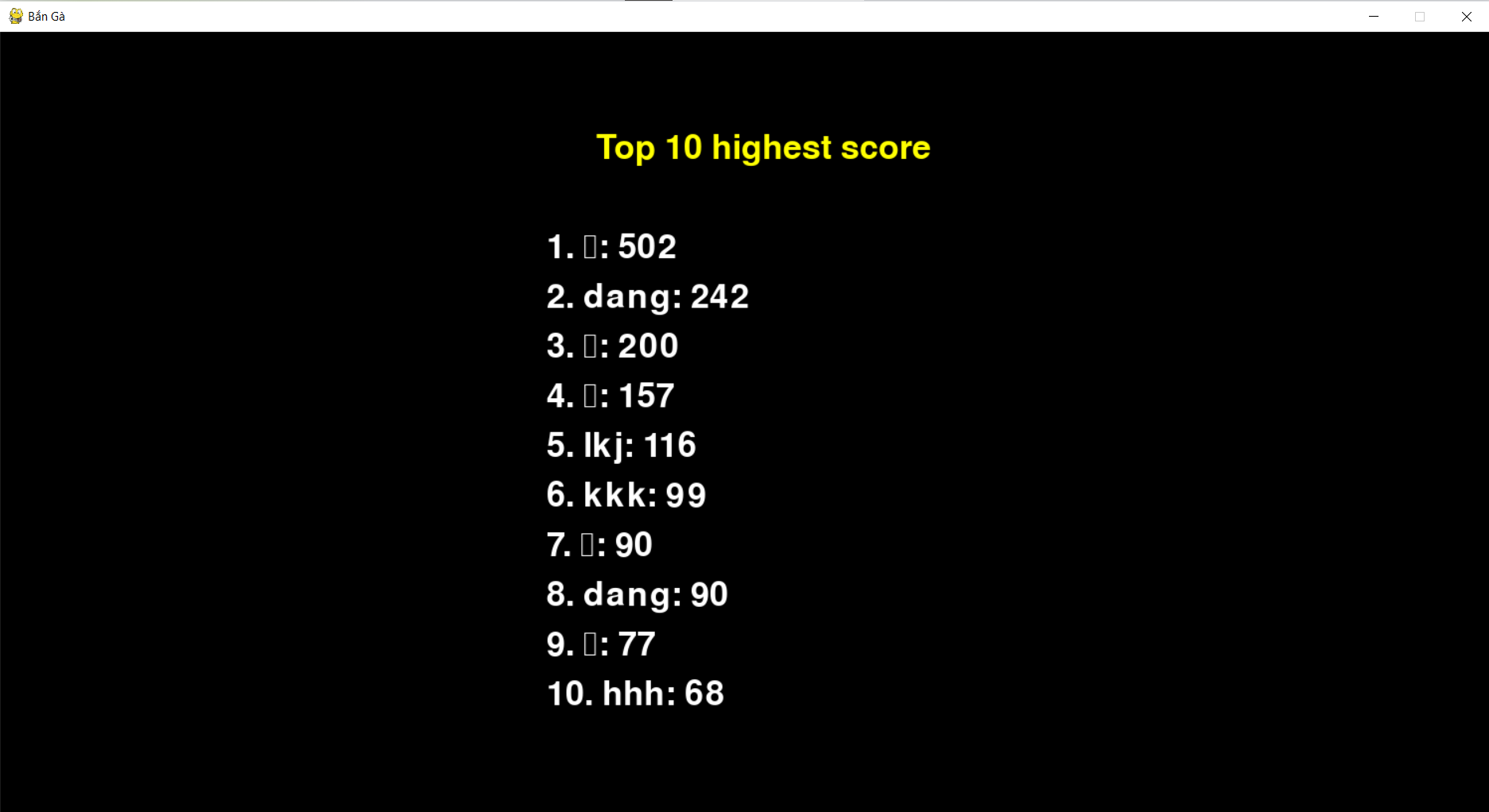
 Mô tả:

Văn bản "GAME OVER" màu đỏ (pygame.font.Font(None, 72)), hiển thị giữa màn hình trong 2 giây.

Tương tác: Tự động quay lại menu chính sau khi hiển thị.

Vai trò: Thông báo kết thúc trò chơi và lưu điểm.

*3.3.5. Màn hình bảng xếp hạng (show\_scores)*



Mô tả:

Nền màu đen.

Tiêu đề "Top 10 highest score" màu vàng, phía trên.

Danh sách 10 người chơi có điểm cao nhất, màu trắng, xếp dọc.

Tương tác: Nhấn Enter để quay lại menu.

Vai trò: Hiển thị thành tích, tăng tính cạnh tranh.

**CHƯƠNG IV. KẾT QUẢ THỰC NGHIỆM CHƯƠNG TRÌNH**

4.1. Mô tả kết quả kiểm thử các chức năng chương trình với tối thiểu 2 mẫu dữ liệu đầu vào

Để đánh giá hiệu quả và tính ổn định của trò chơi "Bắn Gà" được phát triển trong môn "Lập trình ứng dụng với Python", nhóm chúng tôi đã tiến hành kiểm thử các chức năng chính của chương trình với hai kịch bản thực nghiệm khác nhau. Mỗi kịch bản sử dụng một mẫu dữ liệu đầu vào (tên người chơi và hành vi chơi) để kiểm tra toàn bộ các tính năng, từ giao diện, điều khiển, logic trò chơi, đến lưu trữ điểm số. Dưới đây là mô tả chi tiết quá trình kiểm thử và kết quả thu được:

*4.1.1. Kịch bản 1: Người chơi mới, chơi đến Level 3 và thua*

Mẫu dữ liệu đầu vào:

Tên người chơi: "Player1".

Hành vi: Chơi cơ bản, tập trung tiêu diệt gà và một số quái vật cấp thấp, thua khi bị trúng đạn của kẻ thù.

Quy trình kiểm thử:

Khởi động chương trình: Chạy tệp shooting\_chicken.py, màn hình menu chính xuất hiện với các tùy chọn "Play now", "View achievements", và "Quit".

Chọn "Play now": Nhấn phím Enter trên tùy chọn "Play now", chuyển sang màn hình nhập tên.

Nhập tên: Gõ "Player1" bằng bàn phím, nhấn Enter để xác nhận, chuyển sang màn hình trò chơi.

Chơi trò chơi:

Điều khiển phi thuyền bằng chuột, bắn đạn bằng cách nhấp chuột trái.

Tiêu diệt 5 con gà (Chicken) trong Level 1, mỗi con cộng 1 điểm, tổng cộng 5 điểm.

Sau 10 giây, trò chơi tự động tăng lên Level 2, xuất hiện Monster1. Tiêu diệt 3 Monster1 (mỗi con 2 điểm), tổng cộng thêm 6 điểm, nâng điểm số lên 11.

Sau 10 giây tiếp theo, lên Level 3, Monster1 bắn đạn nhanh hơn (Bullet1). Phi thuyền bị trúng một viên đạn từ Monster1.

Kết thúc: Màn hình "GAME OVER" xuất hiện trong 2 giây, sau đó quay lại menu chính.

Kiểm tra bảng xếp hạng: Chọn "View achievements" để xem điểm số đã lưu.

Kết quả kiểm thử:

Giao diện:

Menu chính hiển thị đúng, điều hướng bằng phím lên/xuống và Enter hoạt động ổn định.

Màn hình nhập tên nhận diện chính xác ký tự nhập vào, chuyển sang trò chơi sau khi nhấn Enter.

Màn hình trò chơi hiển thị nền cuộn mượt mà, phi thuyền, kẻ thù, và đạn được vẽ đúng vị trí.

"GAME OVER" hiển thị rõ ràng, tự động quay lại menu sau 2 giây.

Điều khiển:

Phi thuyền di chuyển theo chuột chính xác, không vượt ra ngoài biên màn hình.

Nhấp chuột bắn đạn đều đặn, đạn di chuyển lên với tốc độ BULLET\_SPEED = 7.

Logic trò chơi:

Điểm số tăng đúng: 5 gà (+5), 3 Monster1 (+6), tổng cộng 11 điểm.

Cấp độ tăng đúng sau mỗi 10 giây (Level 1 → Level 2 → Level 3).

Va chạm được phát hiện chính xác: Trúng đạn của Monster1 dẫn đến kết thúc trò chơi.

Lưu trữ điểm số:

Điểm 11 của "Player1" được lưu vào scores.db.

Bảng xếp hạng hiển thị: "1. Player1: 11" ở vị trí đầu tiên (do là lần chơi đầu tiên).

Hiệu suất: Chạy ổn định ở 60 FPS, không giật lag khi số lượng đối tượng dưới 10.

Đánh giá: Kịch bản này kiểm tra thành công các chức năng cơ bản như giao diện menu, nhập tên, điều khiển phi thuyền, bắn đạn, tăng điểm, tăng cấp độ, và lưu điểm. Trò chơi hoạt động đúng như thiết kế, phù hợp với người chơi mới.

*4.1.2. Kịch bản 2: Người chơi có kinh nghiệm, chơi đến Level 5 và thua*

Mẫu dữ liệu đầu vào:

Tên người chơi: "Player2".

Hành vi: Chơi tích cực, tiêu diệt nhiều kẻ thù qua nhiều cấp độ, thua khi bị trúng kẻ thù trực tiếp.

Quy trình kiểm thử:

Khởi động chương trình: Tương tự kịch bản 1, vào menu chính và chọn "Play now".

Nhập tên: Gõ "Player2", nhấn Enter để bắt đầu.

Chơi trò chơi:

Level 1: Tiêu diệt 10 con gà (+10 điểm).

Level 2: Sau 10 giây, tiêu diệt 5 Monster1 (+10 điểm), tổng 20 điểm.

Level 3: Sau 10 giây, tiêu diệt 3 Monster2 (+9 điểm), tổng 29 điểm.

Level 4: Sau 10 giây, tiêu diệt 2 Monster3 (+8 điểm), tổng 37 điểm.

Level 5: Sau 10 giây, xuất hiện Monster4 (máu 10), phi thuyền va chạm trực tiếp với Monster4 và thua.

Kết thúc: Màn hình "GAME OVER" hiển thị, quay lại menu sau 2 giây.

Kiểm tra bảng xếp hạng: Chọn "View achievements" để xem điểm của "Player2" so với "Player1".

Kết quả kiểm thử:

Giao diện:

Màn hình trò chơi hiển thị đúng các loại kẻ thù mới (Monster2, Monster3, Monster4) với hình ảnh và đạn đặc trưng (zigzag với Bullet3, chậm to với Bullet4).

Điểm số và cấp độ hiển thị chính xác ở góc trên trái: "Score: 37 | Level: 5" trước khi thua.

Điều khiển:

Phi thuyền né đạn thành công qua 4 cấp độ, chứng minh khả năng điều khiển mượt mà.

Bắn liên tục tiêu diệt kẻ thù không gặp lỗi chồng đạn.

Logic trò chơi:

Điểm số tăng đúng: 10 gà (+10), 5 Monster1 (+10), 3 Monster2 (+9), 2 Monster3 (+8), tổng 37 điểm.

Cấp độ tăng đúng qua 40 giây (4 lần 10 giây), xuất hiện kẻ thù mới theo cấp độ:

Level 1: Chỉ có gà.

Level 2-3: Thêm Monster1, Monster2.

Level 4-5: Thêm Monster3, Monster4.

Va chạm trực tiếp với Monster4 dẫn đến thua ngay lập tức, đúng logic thiết kế.

Lưu trữ điểm số:

Điểm 37 của "Player2" được lưu vào scores.db.

Bảng xếp hạng cập nhật:

1. Player2: 37

2. Player1: 11

Hiệu suất: FPS ổn định ở 60 khi có dưới 15 đối tượng, hơi giảm nhẹ (khoảng 55 FPS) khi Monster4 xuất hiện do kích thước lớn và đạn Bullet4 phức tạp.

Đánh giá: Kịch bản này kiểm tra thành công các chức năng nâng cao như xử lý nhiều loại kẻ thù, đạn đặc biệt, tăng cấp độ liên tục, và cập nhật bảng xếp hạng. Trò chơi đáp ứng tốt với người chơi có kinh nghiệm, dù hiệu suất cần tối ưu hơn khi số lượng đối tượng tăng.

*4.1.3. Tổng hợp kết quả*

Chức năng kiểm tra:

Giao diện: Menu, nhập tên, trò chơi, Game Over, bảng xếp hạng – tất cả hoạt động đúng.

Điều khiển: Di chuyển, bắn đạn chính xác, không lỗi.

Logic: Điểm số, cấp độ, va chạm được tính toán chính xác.

Lưu trữ: Điểm số lưu và hiển thị đúng trong SQLite.

Hạn chế: Hiệu suất giảm nhẹ ở cấp độ cao (Level 5), cần tối ưu thêm khi số lượng đối tượng lớn.

Kết luận: Trò chơi "Bắn Gà" hoạt động ổn định với cả người chơi mới và có kinh nghiệm, đáp ứng đầy đủ yêu cầu của môn "Lập trình ứng dụng với Python".

**CHƯƠNG V. KẾT LUẬN**

**5.1. Các nội dung đã đạt được**

Dự án "Xây dựng trò chơi Bắn Gà bằng Python và Pygame" trong môn "Lập trình ứng dụng với Python" đã hoàn thành với nhiều thành tựu đáng ghi nhận. Nhóm chúng tôi đã đạt được các mục tiêu đề ra cả về mặt học thuật lẫn kỹ thuật, cụ thể như sau:

Hoàn thiện một trò chơi hoàn chỉnh:

Trò chơi "Bắn Gà" đã được xây dựng với đầy đủ các tính năng cơ bản của một trò chơi bắn súng 2D: điều khiển phi thuyền bằng chuột, bắn đạn để tiêu diệt kẻ thù, tăng cấp độ theo thời gian, và hiển thị điểm số trên màn hình. Các loại kẻ thù (gà và 4 loại quái vật) được thiết kế với đặc điểm riêng biệt (tốc độ, máu, loại đạn), tạo nên sự đa dạng và thử thách cho người chơi.

Ứng dụng thành công kiến thức từ môn học:

Sử dụng Python và thư viện Pygame để xử lý đồ họa, sự kiện người dùng, và vòng lặp trò chơi, thể hiện rõ khả năng lập trình hướng đối tượng và quản lý tài nguyên đồ họa. Tích hợp SQLite để lưu trữ điểm số, cho phép hiển thị bảng xếp hạng top 10, minh chứng cho việc áp dụng quản lý dữ liệu trong ứng dụng thực tế.

Giao diện và trải nghiệm người dùng:

Giao diện đồ họa (GUI) được thiết kế trực quan với menu chính, màn hình nhập tên, màn hình trò chơi, màn hình "Game Over", và bảng xếp hạng. Nền cuộn liên tục và các hình ảnh (phi thuyền, kẻ thù, đạn) được hiển thị mượt mà, tạo cảm giác chuyên nghiệp. Điều khiển bằng chuột nhạy bén, phù hợp với cả người chơi mới và có kinh nghiệm.

Kiểm thử và hiệu suất:

Qua hai kịch bản kiểm thử (người chơi mới đến Level 3 và người chơi kinh nghiệm đến Level 5), trò chơi hoạt động ổn định ở 60 FPS khi số lượng đối tượng vừa phải, đáp ứng đúng các yêu cầu về điểm số, cấp độ, và va chạm. Điểm số được lưu trữ chính xác trong cơ sở dữ liệu scores.db và hiển thị đúng thứ tự trong bảng xếp hạng.

Kỹ năng làm việc nhóm:

Nhóm đã phối hợp hiệu quả trong việc phân công nhiệm vụ, tích hợp mã nguồn, và giải quyết vấn đề. Nhóm trưởng đảm nhận phần lớn công việc kỹ thuật, trong khi các thành viên khác hỗ trợ tốt ở các khía cạnh phụ, giúp hoàn thành dự án đúng thời hạn. Nhìn chung, dự án không chỉ đáp ứng yêu cầu của môn học mà còn mang lại một sản phẩm có thể sử dụng thực tế, thể hiện khả năng sáng tạo và kỹ thuật của nhóm trong việc phát triển ứng dụng bằng Python.

**5.2. Các nội dung chưa đạt được**

Mặc dù dự án đã đạt được nhiều thành công, vẫn còn một số hạn chế và nội dung chưa thể hoàn thiện do giới hạn về thời gian, kinh nghiệm, và nguồn lực. Các nội dung chưa đạt được bao gồm:

Thiếu âm thanh:

Trò chơi hiện chưa có nhạc nền, hiệu ứng âm thanh khi bắn đạn, tiêu diệt kẻ thù, hay khi thua (Game Over). Điều này làm giảm trải nghiệm cảm xúc của người chơi so với các trò chơi thương mại như "Chicken Invaders".

Lý do: Nhóm chưa có đủ thời gian để tìm hiểu và tích hợp thư viện âm thanh của Pygame (pygame.mixer).

Chưa có hệ thống power-up hoặc vật phẩm hỗ trợ:

Trò chơi thiếu các tính năng nâng cao như vật phẩm tăng tốc độ bắn, thêm mạng sống, hoặc vũ khí mạnh hơn. Những yếu tố này có thể làm tăng tính hấp dẫn và chiến lược cho người chơi.

Lý do: Thiết kế và lập trình power-up đòi hỏi thêm logic phức tạp, vượt quá thời gian thực hiện dự án.

Hiệu suất chưa tối ưu ở cấp độ cao:

Khi số lượng kẻ thù và đạn tăng lên (đặc biệt từ Level 5 với Monster4), FPS giảm nhẹ từ 60 xuống khoảng 55, gây cảm giác giật nhẹ trên một số máy tính cấu hình thấp.

Lý do: Nhóm chưa triển khai các kỹ thuật tối ưu hóa như giảm số lần vẽ đối tượng hoặc sử dụng cấu trúc dữ liệu hiệu quả hơn.

Thiếu màn chơi riêng hoặc boss cuối:

Trò chơi hiện chỉ tăng cấp độ bằng cách thêm kẻ thù mới mà chưa có màn chơi đặc biệt hoặc boss cuối với thiết kế độc đáo, điều thường thấy trong các trò chơi bắn súng.

Lý do: Thiết kế boss đòi hỏi thêm tài nguyên đồ họa và logic riêng, vượt quá phạm vi dự án học thuật.

Giao diện đơn giản:

Mặc dù giao diện đã đáp ứng yêu cầu cơ bản, nhưng menu và bảng xếp hạng còn đơn điệu (chỉ có chữ và nền đen), thiếu hiệu ứng hoặc màu sắc sinh động để thu hút người chơi.

Lý do: Nhóm tập trung chủ yếu vào logic trò chơi hơn là nâng cấp giao diện.

Những hạn chế này không ảnh hưởng nghiêm trọng đến chức năng chính của trò chơi, nhưng nếu được khắc phục, sản phẩm sẽ hoàn thiện và hấp dẫn hơn.

**5.3. Dự kiến phát triển**

Dựa trên những thành tựu đã đạt được và các hạn chế còn tồn tại, nhóm chúng tôi lên kế hoạch phát triển thêm cho trò chơi "Bắn Gà" trong tương lai để nâng cao chất lượng và trải nghiệm người chơi. Các hướng phát triển dự kiến bao gồm:

Tích hợp âm thanh:

Thêm nhạc nền liên tục trong trò chơi và hiệu ứng âm thanh (bắn đạn, nổ khi tiêu diệt kẻ thù, tiếng thông báo khi thua) bằng cách sử dụng pygame.mixer. Điều này sẽ làm tăng sự sống động và cuốn hút của trò chơi.

Thêm hệ thống power-up:

Thiết kế các vật phẩm xuất hiện ngẫu nhiên trên màn hình (ví dụ: tăng tốc độ bắn, thêm mạng, đạn mạnh hơn). Mỗi vật phẩm sẽ có hình ảnh riêng và ảnh hưởng đến thuộc tính của phi thuyền (SHIP\_SPEED, số lượng đạn bắn cùng lúc). Điều này giúp người chơi có thêm chiến lược và tăng tính tương tác.

Tối ưu hóa hiệu suất:

Áp dụng các kỹ thuật như sử dụng sprite groups trong Pygame để quản lý đối tượng hiệu quả hơn, giảm tải cho CPU khi số lượng kẻ thù và đạn tăng. Thử nghiệm trên nhiều cấu hình máy tính để đảm bảo FPS ổn định ngay cả ở cấp độ cao.

Phát triển màn chơi và boss cuối:

Thiết kế các màn chơi riêng biệt (ví dụ: màn 1 chỉ có gà, màn 2 có quái vật nhỏ, màn 3 có boss). Tạo một boss cuối với kích thước lớn, nhiều máu, và đạn phức tạp (ví dụ: bắn theo hình tròn hoặc lưới), xuất hiện ở Level 10 để làm điểm nhấn cho trò chơi.

Nâng cấp giao diện:

Thêm hiệu ứng chuyển động cho menu (ví dụ: chữ nhấp nháy), sử dụng hình nền đẹp hơn thay vì màu đen đơn điệu. Cải thiện bảng xếp hạng bằng cách thêm màu sắc, biểu tượng, hoặc hình ảnh minh họa cho từng vị trí.

Mở rộng tính năng multiplayer:

Nghiên cứu tích hợp chế độ chơi hai người (local multiplayer) trên cùng một máy bằng cách chia đôi màn hình hoặc dùng bàn phím cho người thứ hai. Điều này sẽ tăng tính cạnh tranh và giải trí.

Phát hành và chia sẻ:

Đóng gói trò chơi thành tệp thực thi (.exe) bằng công cụ như PyInstaller để chia sẻ với bạn bè hoặc cộng đồng. Đăng tải mã nguồn lên GitHub để nhận phản hồi và cải tiến từ cộng đồng lập trình viên. Những kế hoạch này không chỉ giúp khắc phục các hạn chế hiện tại mà còn mở ra tiềm năng biến "Bắn Gà" thành một trò chơi có giá trị thực tiễn cao hơn, đồng thời là cơ hội để nhóm tiếp tục học hỏi và phát triển kỹ năng lập trình trong tương lai.

**TÀI LIỆU THAM KHẢO**

[1] Pygame Documentation, "Pygame Official Documentation".

[2] SQLite Documentation, "SQLite Official Documentation”

[3] Python Documentation, "Python 3.x Official Documentation”

[4] Real Python, "Game Development with Pygame"