**TRƯỜNG ĐẠI HỌC NAM CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Sinh viên : NGUYỄN TRỌNG NGHĨA**

**MSSV : 211134**

**Khóa : K9**

**ĐỀ TÀI**

**Flatform game bắn UFO**

**BÁO CÁO ĐỒ ÁN CƠ SỞ 02**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Mã số ngành: 7480201**

**Cần Thơ, tháng 12 năm 2024**

**TRƯỜNG ĐẠI HỌC NAM CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN CƠ SỞ 02**

**Ngành: Công Nghệ Thông Tin**

**Mã số ngành: 7480201**

**ĐỀ TÀI**

**FLATFORM GAME BẮN UFO**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn**  **Thầy TRẦN VĂN THIỆN** | **Sinh viên thực hiện**  **NGUYỄN TRỌNG NGHĨA**  **MSSV: 2111134** |

**Cần Thơ, tháng 12 năm 2024**

**LỜI CẢM ƠN**

Sau một thời gian học tập và tích lũy kiến thức chuyên môn từ các môn học, cùng với việc thực hành các kiến thức, nhóm chúng em đã tìm hiểu thêm các kiến thức về lập trình. Điều này giúp chúng em có cái nhìn toàn diện hơn về lĩnh vực này. Chúng em đã áp dụng những kiến thức đã học trên lớp cùng các tài liệu bổ sung để triển khai dự án mang tên “Flatform game bắn UFO”.

Trong quá trình thực hiện đồ án, nhóm chúng em xin gửi lời cảm ơn sâu sắc đến thầy Trần Văn Thiện đã tận tình hướng dẫn, theo dõi sát sao quá trình xây dựng đồ án và hỗ trợ kịp thời, giúp chúng em có thể hoàn thành dự án.

Chúng em cũng xin chân thành cảm ơn quý thầy cô Khoa Công nghệ thông tin Trường Đại học Nam Cần Thơ đã tận tâm giảng dạy và truyền đạt những kiến thức quý báu, giúp chúng em hoàn thiện chương trình học tại trường.

Vì còn hạn chế về kinh nghiệm và kiến thức, mặc dù đã rất nỗ lực, chúng em không thể tránh khỏi những thiếu sót trong quá trình thực hiện đề tài. Chúng em rất mong nhận được những góp ý từ quý thầy cô để có thể khắc phục và hoàn thiện đề tài một cách tốt nhất.

Cuối cùng, chúng em xin kính chúc quý Thầy Cô Khoa Công nghệ Thông tin Trường Đại học Nam Cần Thơ luôn dồi dào sức khỏe và gặt hái nhiều thành công.

*Em xin chân thành cảm ơn thầy cô rất nhiều!*

|  |
| --- |
| **Sinh viên thực hiện** |
| **Nguyễn Trọng Nghĩa** |

**LỜI CAM KẾT**

Chúng em xin cam kết rằng toàn bộ nội dung và kết quả trình bày trong đồ án cơ sở 1 này về "Flatform game bắn UFO" là thành quả từ kiến thức, kinh nghiệm và nỗ lực chung của nhóm. Sản phẩm được thực hiện một cách độc lập, chưa từng sử dụng cho bất kỳ đồ án nào khác và phản ánh quá trình nghiên cứu cũng như làm việc chăm chỉ của chúng em.

Trong suốt quá trình thực hiện, nhóm đã luôn tôn trọng quyền lợi và sự riêng tư của các cá nhân, tổ chức liên quan, đảm bảo không có bất kỳ vi phạm nào xảy ra. Các thông tin, dữ liệu, hình ảnh và tài liệu tham khảo đều được sử dụng một cách có trách nhiệm, được trích dẫn nguồn và chú thích đầy đủ.

Nhóm cũng đã tận dụng các công cụ lập trình phù hợp nhằm tối ưu hóa hiệu quả và chất lượng sản phẩm. Đây là kết quả của sự phối hợp nhịp nhàng giữa các thành viên, với mục tiêu mang đến một tựa game bắn súng phong cách platform giàu tính thử thách và hấp dẫn.

Với mong muốn đóng góp vào sự phát triển của ngành công nghiệp game, nhóm sẽ không ngừng cải thiện và hoàn thiện sản phẩm, mang lại những trải nghiệm giải trí độc đáo cho người chơi.

|  |
| --- |
| **Sinh viên thực hiện**  **Nguyễn Trọng Nghĩa** |
|
|  |

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

*Cần Thơ, ngày .... tháng ..... năm 2024*

**NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN**

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

.................................................................................................................................................................................

*Cần thơ, ngày ..... tháng ..... năm 2024*

|  |  |
| --- | --- |
|  | **Giảng viên phản biện** |

**LỜI NÓI ĐẦU**

Với sự bùng nổ của công nghệ thông tin và nhu cầu giải trí ngày càng cao, các nền tảng game đang trở thành một phần không thể thiếu trong đời sống của nhiều người. Game không chỉ mang lại sự thư giãn, giải trí mà còn mở ra một không gian sáng tạo, nơi người chơi có thể thử thách bản thân và trải nghiệm những câu chuyện độc đáo. Sự phát triển vượt bậc của công nghệ đã thay đổi cách chúng ta tiếp cận và trải nghiệm các trò chơi, góp phần thúc đẩy ngành công nghiệp game ngày càng phát triển mạnh mẽ.

Trong số đó, các trò chơi platform bắn súng đang nhận được sự yêu thích đặc biệt nhờ sự kết hợp giữa lối chơi sáng tạo, thử thách và yếu tố hành động gây cấn. Nhu cầu phát triển các tựa game thuộc thể loại này không ngừng tăng lên, mở ra cơ hội lớn cho các nhà phát triển trong việc mang đến những sản phẩm giải trí chất lượng cao, đáp ứng sở thích đa dạng của người chơi.

Nhận thấy tiềm năng và sức hút của thể loại này, chúng em đã quyết định phát triển một dự án Flatform game bắn UFO. Mục tiêu của dự án không chỉ là tạo ra một tựa game giải trí cuốn hút mà còn nhằm thử nghiệm và áp dụng các kỹ thuật lập trình, thiết kế đồ họa và phát triển tính năng theo chuẩn mực hiện đại. Trò chơi này không chỉ mang lại trải nghiệm thú vị cho người chơi mà còn là cơ hội để nhóm học hỏi và phát triển kỹ năng trong lĩnh vực lập trình game, đồng thời đóng góp vào sự phát triển của ngành công nghiệp giải trí kỹ thuật số.

# MỤC LỤC

[MỤC LỤC vi](#_Toc184428838)

[Danh Mục Hình Ảnh ix](#_Toc184428839)

[CHƯƠNG 1: GIỚI THIỆU 1](#_Toc184428840)

[I. Giới thiệu đề tài 1](#_Toc184428841)

[II. Lý do chọn đề tài 1](#_Toc184428842)

[III. Mục tiêu của đề tài 1](#_Toc184428843)

[IV. Phạm vi ứng dụng 1](#_Toc184428844)

[V. Phương pháp thực hiện 2](#_Toc184428845)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 3](#_Toc184428846)

[I. Lập trình trò chơi 2D 3](#_Toc184428847)

[II. Giới thiệu thư viện Pygame 3](#_Toc184428848)

[III. SDL2 và vai trò trong Pygame 4](#_Toc184428849)

[IV. Lập trình hướng đối tượng (OOP) trong trò chơi 4](#_Toc184428850)

[V. Xử lý va chạm (Collision Detection) 5](#_Toc184428851)

[VI. Tối ưu hóa hiệu suất trong trò chơi 5](#_Toc184428852)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ GAME 6](#_Toc184428853)

[I. Phân tích yêu cầu 6](#_Toc184428854)

[1.1. Yêu cầu tính năng 6](#_Toc184428855)

[1.2. Yêu cầu giao diện 6](#_Toc184428856)

[1.3. Yêu cầu hiệu năng 7](#_Toc184428857)

[II. Giao diện trò chơi 8](#_Toc184428858)

[2.1. Thiết kế menu chính 8](#_Toc184428859)

[2.2. Thiết kế giao diện trong trò chơi 8](#_Toc184428860)

[2.3. Phong cách đồ họa và hiệu ứng âm thanh 9](#_Toc184428861)

[2.4. Thiết kế hệ thống 11](#_Toc184428862)

[CHƯƠNG 4: QUÁ TRÌNH TRIỂN KHAI 13](#_Toc184428863)

[I. Cấu trúc mã nguồn 13](#_Toc184428864)

[1.1. Các tệp mã nguồn chính 13](#_Toc184428865)

[1.2. Các lớp và đối tượng trong mã nguồn 14](#_Toc184428866)

[1.3. Quản lý tài nguyên 15](#_Toc184428867)

[1.4. Kiểm thử và duy trì mã nguồn 16](#_Toc184428868)

[II. Các thành phần chính của hệ thống 16](#_Toc184428869)

[2.1. Quản lý các đối tượng trong trò chơi 16](#_Toc184428870)

[2.2. Cập nhật trạng thái và hành vi của các đối tượng 16](#_Toc184428871)

[2.3. Xử lý sự kiện người dùng 17](#_Toc184428872)

[2.4. Quản lý tài nguyên 17](#_Toc184428873)

[III. Xử lý và quản lý sự kiện trong trò chơi 17](#_Toc184428874)

[3.1. Xử lý sự kiện người dùng 18](#_Toc184428875)

[3.2 Quản lý sự kiện trên menu 18](#_Toc184428876)

[3.3 Cập nhật và phản hồi các sự kiện 18](#_Toc184428877)

[IV. Giao diện trò chơi 19](#_Toc184428878)

[4.1 Giao diện menu 19](#_Toc184428879)

[4.2 Giao diện khởi đầu 19](#_Toc184428880)

[4.3 Giao diện trò chơi 20](#_Toc184428881)

[4.4 Giao diện tạm dừng 20](#_Toc184428882)

[4.5 Giao diện game over 21](#_Toc184428883)

[4.6. Giao diện khi nhân vật chọn buff 21](#_Toc184428884)

[CHƯƠNG 5: KẾT LUẬN 22](#_Toc184428885)

[I. Kết quả đạt được 22](#_Toc184428886)

[II. Ưu điểm của trò chơi 22](#_Toc184428887)

[III. Nhược điểm của trò chơi 22](#_Toc184428888)

[IV. Hướng phát triển trong tương lai 23](#_Toc184428889)

[4.1. Tính năng mới và mở rộng gameplay 23](#_Toc184428890)

[4.2. Tối ưu hóa hiệu suất và khả năng mở rộng 23](#_Toc184428891)

[4.3. Cải tiến đồ họa và âm thanh 24](#_Toc184428892)

[4.4. Hệ thống bảng xếp hạng và chia sẻ điểm 24](#_Toc184428893)

[TÀI LIỆU SỬ DỤNG VÀ THAM KHẢO 25](#_Toc184428894)

# Danh Mục Hình Ảnh

[Hình 3.1 Mô hình phi thuyền 9](#_Toc184181654)

[Hình 3.2 Mô hình đĩa bay 10](#_Toc184181655)

[Hình 3.3 Hiệu ứng đạn bắn 10](#_Toc184181656)

[Hình 3.4 Hình nền 11](#_Toc184181657)

[Hình 4.5 Giao diện menu 19](#_Toc184181658)

[Hình 4.6 Hình giao diện khởi đầu 19](#_Toc184181659)

[Hình 4.7 Giao diện trò chơi 20](#_Toc184181660)

[Hình 4.8 Giao diện tạm dừng 21](#_Toc184181661)

[Hình 4.9 Giao diện game over 21](#_Toc184181662)

[Hình 4.10 Giao diện chọn buff 22](#_Toc184181663)

# CHƯƠNG 1: GIỚI THIỆU

## I. Giới thiệu đề tài

Trò chơi điện tử là một trong những lĩnh vực phát triển nhanh nhất trong ngành công nghiệp công nghệ, không chỉ đáp ứng nhu cầu giải trí mà còn trở thành một nền tảng học tập, sáng tạo và nghiên cứu kỹ thuật số. Đề tài Flatform Game Bắn UFO được lựa chọn với mục tiêu xây dựng một trò chơi 2D đơn giản nhưng hấp dẫn, hoạt động ổn định trên máy tính và có tiềm năng triển khai trên các nền tảng khác như Android. Thông qua đề tài này, nhóm thực hiện mong muốn ứng dụng các kiến thức đã học vào thực tiễn, từ lập trình, xử lý đồ họa, âm thanh, cho đến tối ưu hóa hiệu năng và quản lý dự án.

## II. Lý do chọn đề tài

Đề tài này được lựa chọn xuất phát từ sự kết hợp giữa sở thích cá nhân với mục tiêu phát triển kỹ năng lập trình và sáng tạo. Lập trình trò chơi không chỉ là một cách thực hành hiệu quả mà còn là một lĩnh vực thú vị, đầy thử thách, giúp người thực hiện cải thiện tư duy logic, khả năng giải quyết vấn đề và kỹ năng làm việc nhóm. Ngoài ra, việc tạo ra một sản phẩm giải trí như trò chơi Flatform Game Bắn UFO là cơ hội để nhóm thực hiện khám phá tiềm năng của Python và Pygame trong việc phát triển các ứng dụng thực tế. Đặc biệt, đề tài này còn hướng đến việc xây dựng sản phẩm có giá trị sử dụng, mang tính ứng dụng cao trong môi trường học tập và nghiên cứu.

## III. Mục tiêu của đề tài

Mục tiêu chính của đề tài là xây dựng một trò chơi 2D hoàn chỉnh với giao diện thân thiện, logic gameplay rõ ràng, và hiệu năng ổn định. Trò chơi sẽ bao gồm các tính năng cơ bản như điều khiển nhân vật, bắn đạn tiêu diệt UFO, tích điểm, và hiển thị thông tin trạng thái trò chơi. Đồng thời, dự án cũng hướng đến việc đảm bảo mã nguồn được tổ chức chặt chẽ, dễ bảo trì và có khả năng mở rộng trong tương lai. Một mục tiêu khác là giúp nhóm thực hiện nâng cao kỹ năng lập trình hướng đối tượng (OOP), thiết kế giao diện, xử lý đồ họa và âm thanh, cũng như khả năng đóng gói ứng dụng trên các nền tảng khác nhau.

## IV. Phạm vi ứng dụng

Phạm vi thực hiện của đề tài tập trung vào việc phát triển trò chơi 2D với các tính năng cơ bản, phù hợp với người mới học lập trình game. Trò chơi sẽ được triển khai trên máy tính, sử dụng ngôn ngữ Python và thư viện Pygame. Phần thiết kế bao gồm giao diện menu, giao diện trong trò chơi, và logic gameplay cơ bản. Dự án sẽ không tập trung vào các tính năng phức tạp như nhiều người chơi hoặc tích hợp trực tuyến, nhưng đảm bảo trò chơi hoạt động mượt mà và dễ dàng mở rộng. Ngoài ra, đề tài cũng đặt mục tiêu xây dựng nền tảng để chuyển trò chơi lên thiết bị di động trong tương lai.

## V. Phương pháp thực hiện

Để thực hiện đề tài, nhóm sử dụng phương pháp phát triển phần mềm từ khâu phân tích, thiết kế, triển khai đến kiểm thử và tối ưu hóa. Python và Pygame là công cụ chính, nhờ tính dễ học và hỗ trợ mạnh mẽ cho lập trình game 2D. Phương pháp lập trình hướng đối tượng (OOP) được áp dụng để quản lý các thành phần của trò chơi như nhân vật, kẻ thù, đạn, và logic trò chơi. Ngoài ra, nhóm cũng tham khảo các tài liệu, khóa học trực tuyến về lập trình game, đồng thời thực hiện kiểm thử liên tục để đảm bảo chất lượng và hiệu năng của sản phẩm.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## I. Lập trình trò chơi 2D

Lập trình trò chơi 2D là lĩnh vực kết hợp giữa thiết kế phần mềm và đồ họa, nhằm tạo ra các ứng dụng giải trí mang tính tương tác. Trò chơi 2D có các đặc điểm cơ bản như:

**Đồ họa hai chiều**: Trò chơi sử dụng không gian 2D với chiều rộng và chiều cao để mô phỏng đối tượng. Mọi vật thể trong trò chơi như nhân vật, kẻ thù, đạn, v.v., đều có thể được hiển thị dưới dạng hình ảnh 2D.

**Sự kiện và điều khiển**: Trò chơi yêu cầu sự tương tác giữa người chơi và phần mềm thông qua các sự kiện, chẳng hạn như nhấn chuột, di chuyển chuột, hoặc nhấn phím. Các sự kiện này sẽ tác động đến trạng thái và hành vi của trò chơi.

**Luồng trò chơi (Game Loop)**: Đây là vòng lặp chính trong trò chơi, nơi các sự kiện được kiểm tra, các đối tượng trong trò chơi được cập nhật, và màn hình được vẽ lại. Game loop giúp trò chơi hoạt động liên tục và mượt mà.

Để phát triển trò chơi 2D, các công cụ như **Pygame** có thể hỗ trợ lập trình viên xử lý đồ họa, âm thanh và nhập liệu một cách hiệu quả.

## II. Giới thiệu thư viện Pygame

Pygame, với sự hỗ trợ mạnh mẽ của SDL2, không chỉ là một công cụ để tạo trò chơi 2D đơn giản mà còn là một nền tảng linh hoạt để phát triển các ứng dụng đa phương tiện đa dạng. Từ các trò chơi platform cổ điển, game bắn súng góc nhìn trên cho đến các ứng dụng mô phỏng vật lý, hình ảnh động và giao diện người dùng tùy chỉnh, Pygame đều đáp ứng được. Khả năng tùy biến cao, cộng đồng lớn và tính tương thích đa nền tảng khiến Pygame trở thành một lựa chọn tuyệt vời cho cả những nhà phát triển nghiệp dư và chuyên nghiệp.

Pygame với cú pháp đơn giản, dễ học và cộng đồng người dùng sôi động, Pygame giúp các nhà phát triển, từ người mới bắt đầu cho đến những người có kinh nghiệm, nhanh chóng tạo ra các trò chơi 2D hấp dẫn. Tính tương thích đa nền tảng của Pygame cho phép các trò chơi được chạy mượt mà trên nhiều hệ điều hành khác nhau, mở rộng khả năng tiếp cận của người chơi. Bên cạnh việc phát triển game, Pygame còn có thể ứng dụng trong nhiều lĩnh vực khác như tạo hình ảnh động, mô phỏng vật lý và phát triển giao diện người dùng. Tuy nhiên, để tạo ra các trò chơi 3D phức tạp và đòi hỏi hiệu năng cao, các engine game chuyên dụng như Unity hay Unreal Engine vẫn là lựa chọn phù hợp hơn.

## III. SDL2 và vai trò trong Pygame

SDL2 là một thư viện phần mềm hỗ trợ việc phát triển các ứng dụng đa phương tiện, đặc biệt là các trò chơi điện tử. SDL2 là nền tảng chính mà **Pygame** sử dụng để thực hiện các thao tác đồ họa, âm thanh và nhập liệu. Các tính năng nổi bật của SDL2 bao gồm:

**Xử lý đồ họa 2D**: SDL2 cho phép vẽ các hình ảnh và hình học cơ bản trên màn hình, từ đó giúp trò chơi hiển thị đồ họa mượt mà và nhanh chóng.

**Quản lý âm thanh**: SDL2 hỗ trợ phát nhạc nền và các hiệu ứng âm thanh, giúp tạo không khí cho trò chơi.

**Tương tác với thiết bị nhập liệu**: SDL2 có thể nhận sự kiện từ các thiết bị như chuột, bàn phím và màn hình cảm ứng, hỗ trợ việc tương tác trong trò chơi.

**Tính tương thích cao**: SDL2 hỗ trợ nhiều hệ điều hành và nền tảng khác nhau như Windows, Linux, macOS, Android, và iOS. Điều này giúp các trò chơi sử dụng SDL2 có thể triển khai trên nhiều thiết bị.

Pygame sử dụng SDL2 để đảm bảo rằng các trò chơi được phát triển với Pygame có hiệu suất cao và khả năng tương thích với nhiều nền tảng.

## IV. Lập trình hướng đối tượng (OOP) trong trò chơi

Lập trình hướng đối tượng (OOP) là một phương pháp lập trình rất phổ biến và hiệu quả, đặc biệt trong phát triển trò chơi. OOP cho phép tổ chức mã nguồn theo cách dễ bảo trì và mở rộng. Các khái niệm cơ bản trong OOP bao gồm:

**Lớp (Class)**: Là khuôn mẫu để tạo ra các đối tượng. Trong trò chơi, các đối tượng như nhân vật, kẻ thù, đạn, v.v., đều có thể được định nghĩa dưới dạng các lớp.

**Đối tượng (Object)**: Là thực thể cụ thể của lớp, với các thuộc tính và phương thức riêng biệt. Ví dụ: đối tượng "Player" có thể có thuộc tính như hp (máu) và phương thức như move() (di chuyển).

**Kế thừa (Inheritance)**: Cho phép lớp con kế thừa các thuộc tính và phương thức từ lớp cha. Ví dụ, lớp "Enemy" có thể kế thừa từ lớp "Character" để chia sẻ các đặc điểm chung.

**Đóng gói (Encapsulation)**: Đảm bảo các thuộc tính và phương thức của lớp không bị truy cập trái phép từ bên ngoài, giúp mã nguồn bảo mật hơn.

**Polymorphism (Đa hình)**: Cho phép các đối tượng khác nhau sử dụng các phương thức có tên giống nhau nhưng thực hiện các hành vi khác nhau.

OOP giúp dễ dàng tổ chức và mở rộng mã nguồn trong các trò chơi phức tạp, nơi nhiều đối tượng cần tương tác với nhau.

## V. Xử lý va chạm (Collision Detection)

Xử lý va chạm là một trong những yếu tố quan trọng trong trò chơi 2D, đặc biệt là trong các trò chơi bắn súng như Flatform Game Bắn UFO, nơi các đối tượng như đạn và kẻ thù cần phải tương tác với nhau. Các phương pháp kiểm tra va chạm phổ biến trong Pygame bao gồm:

**pygame.Rect.colliderect()**: Kiểm tra va chạm giữa hai hình chữ nhật. Đây là cách đơn giản và hiệu quả để phát hiện va chạm giữa các đối tượng.

**pygame.sprite.collide\_rect()**: Kiểm tra va chạm giữa hai sprite, giúp xác định khi các đối tượng đồ họa tương tác với nhau.

**pygame.sprite.collide\_circle()**: Kiểm tra va chạm giữa các đối tượng có hình dạng hình tròn.

Xử lý va chạm giúp trò chơi trở nên sống động và tạo cảm giác phản hồi chính xác khi người chơi thực hiện các hành động như bắn đạn hoặc di chuyển qua các vật thể.

## VI. Tối ưu hóa hiệu suất trong trò chơi

Để trò chơi chạy mượt mà, đặc biệt trên các thiết bị di động hoặc phần cứng hạn chế, việc tối ưu hóa hiệu suất là rất quan trọng. Một số kỹ thuật tối ưu hóa hiệu suất bao gồm:

* **Giới hạn FPS (Frames Per Second)**: Duy trì tốc độ khung hình ổn định giúp trò chơi chạy mượt mà. Sử dụng pygame.time.Clock để giới hạn FPS và giúp hệ thống xử lý tốt hơn.
* **Quản lý bộ nhớ**: Cần đảm bảo rằng các tài nguyên như hình ảnh, âm thanh được giải phóng khi không còn sử dụng, tránh tình trạng chiếm dụng bộ nhớ thừa.
* **Thuật toán hiệu quả**: Sử dụng các thuật toán hiệu quả để xử lý va chạm, di chuyển và các hoạt động tính toán khác giúp tiết kiệm tài nguyên hệ thống và giảm độ trễ.

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ GAME

## I. Phân tích yêu cầu

### 1.1. Yêu cầu tính năng

#### a. Điều khiển nhân vật

Nhân vật chính, một phi thuyền vũ trụ nhỏ nhắn với thiết kế [phong cách đồ họa], sẽ được điều khiển linh hoạt bằng các phím mũi tên hoặc nút ảo trên màn hình cảm ứng. Việc di chuyển phải mượt mà, cho phép người chơi thực hiện những pha né tránh ngoạn mục trước các đợt tấn công của UFO. Ngoài khả năng di chuyển cơ bản, nhân vật còn có thể tăng tốc trong thời gian ngắn để vượt qua những chướng ngại vật hoặc đuổi theo kẻ thù.

#### b. Kẻ thù (UFO)

Các UFO sẽ xuất hiện ngẫu nhiên từ mọi phía của màn hình, với hình dạng và kích thước đa dạng. Chúng không chỉ di chuyển theo đường thẳng mà còn có thể thực hiện các manơơ linh hoạt như lượn vòng, đột ngột đổi hướng để gây khó khăn cho người chơi. Để tăng thêm thử thách, một số loại UFO đặc biệt sẽ được trang bị vũ khí mạnh mẽ hoặc khả năng tự phục hồi.

#### c. Tích điểm

Mỗi khi tiêu diệt một UFO, người chơi sẽ nhận được một số điểm tương ứng với cấp độ của kẻ thù đó. Điểm số sẽ được hiển thị rõ ràng ở góc trên bên phải màn hình, cùng với các hiệu ứng âm thanh và hình ảnh sống động để tạo cảm giác phấn khích. Ngoài ra, người chơi có thể nhận được điểm thưởng khi hoàn thành các nhiệm vụ đặc biệt hoặc đạt được những thành tích nhất định.

#### d. Trạng thái trò chơi

Trò chơi sẽ có ba trạng thái chính: màn hình tiêu đề, màn hình chơi và màn hình kết quả. Màn hình tiêu đề sẽ giới thiệu tên trò chơi, các nút điều khiển và tùy chọn cài đặt. Màn hình chơi là nơi diễn ra các trận chiến căng thẳng, với âm nhạc nền sôi động và hiệu ứng âm thanh sống động. Màn hình kết quả sẽ hiển thị điểm số cuối cùng, các thành tích đã đạt được và cung cấp cho người chơi cơ hội chia sẻ kết quả lên mạng xã hội.

### 1.2. Yêu cầu giao diện

#### a. Giao diện menu chính

Giao diện menu chính của trò chơi sẽ được thiết kế với phong cách đơn giản nhưng thu hút, tập trung vào sự sắp xếp khoa học và dễ nhìn của các nút chức năng. Các nút như "Bắt đầu", và "Thoát" sẽ được đặt ở vị trí trung tâm để người chơi dễ dàng thao tác mà không mất thời gian tìm kiếm. Bối cảnh của menu sẽ được thiết kế lấy cảm hứng từ không gian vũ trụ bao la, với hình ảnh những ngôi sao lấp lánh và các hành tinh xa xôi tạo nên một khung cảnh vừa huyền bí, vừa kích thích trí tò mò. Phong cách này không chỉ giúp tăng tính thẩm mỹ mà còn tạo nên sự gắn kết với chủ đề của trò chơi. Hiệu ứng động nhẹ nhàng, chẳng hạn như các ngôi sao chuyển động chậm rãi hoặc ánh sáng nhấp nháy, sẽ được thêm vào để làm cho giao diện trở nên sống động hơn.

#### b. Giao diện trong trò chơi

Giao diện trong trò chơi sẽ được tối ưu để hiển thị đầy đủ các thông tin cần thiết một cách trực quan và dễ nhìn, giúp người chơi dễ dàng theo dõi tình trạng hiện tại mà không bị phân tâm. Thanh máu của nhân vật sẽ nằm ở góc trên cùng, hiển thị trạng thái sinh tồn của người chơi. Điểm số sẽ được đặt ở vị trí nổi bật, thường là góc trên bên phải, để người chơi biết mình đã tiêu diệt được bao nhiêu kẻ thù. Nếu trò chơi có tính năng giới hạn thời gian, một đồng hồ đếm ngược sẽ được thêm vào để hiển thị thời gian còn lại. Ngoài ra, nếu người chơi sở hữu nhiều loại vũ khí, danh sách các vũ khí hiện có sẽ được hiển thị ở một góc cố định để thuận tiện cho việc chọn lựa. Đặc biệt, trong quá trình chơi, màn hình sẽ liên tục xuất hiện các hiệu ứng đặc biệt như tia laser, vụ nổ khi UFO bị tiêu diệt, hoặc hiệu ứng chuyển động của phi thuyền. Những hiệu ứng này không chỉ làm tăng sự hấp dẫn mà còn giúp tạo ra bầu không khí chiến đấu căng thẳng và đầy kịch tính, phù hợp với nhịp độ nhanh của trò chơi.

#### c. Hiệu ứng đồ họa và âm thanh

Trò chơi sẽ sử dụng đồ họa 2D với phong cách hiện đại, màu sắc tươi sáng và đường nét sắc nét. Các đối tượng như phi thuyền, kẻ thù (UFO), đạn, và bối cảnh không gian vũ trụ sẽ được thiết kế sao cho nổi bật nhưng vẫn hài hòa với tổng thể. Đặc biệt, các hiệu ứng chuyển động sẽ được tối ưu để đảm bảo mượt mà và tự nhiên, chẳng hạn như hiệu ứng vụ nổ khi kẻ thù bị tiêu diệt hoặc hiệu ứng ánh sáng từ tia laser. Bên cạnh đó, âm thanh đóng vai trò quan trọng trong việc tạo nên không khí của trò chơi. Nhạc nền sôi động sẽ duy trì năng lượng của người chơi trong suốt thời gian trải nghiệm, trong khi các hiệu ứng âm thanh như tiếng nổ, tiếng bắn đạn, và âm thanh đặc trưng của UFO sẽ làm tăng thêm cảm giác sống động và chân thực. Sự kết hợp giữa đồ họa và âm thanh sẽ mang đến trải nghiệm chơi game hấp dẫn và đầy lôi cuốn.

### 1.3. Yêu cầu hiệu năng

Để trò chơi chạy mượt mà trên nhiều thiết bị khác nhau, việc tối ưu hóa hiệu năng là một yếu tố rất quan trọng. Trò chơi sẽ được thiết kế để duy trì tốc độ khung hình ổn định, lý tưởng ở mức 60 FPS, nhằm đảm bảo trải nghiệm mượt mà cho người chơi. Các thuật toán hiệu quả sẽ được áp dụng để xử lý logic trò chơi, đặc biệt là việc quản lý số lượng đối tượng trên màn hình, nhằm giảm thiểu áp lực tính toán lên CPU và GPU. Chẳng hạn, khi số lượng UFO xuất hiện tăng lên theo thời gian, thuật toán sẽ được điều chỉnh để giới hạn tối đa số lượng đối tượng được hiển thị đồng thời, tránh gây quá tải. Ngoài ra, việc sử dụng các hiệu ứng đồ họa sẽ được kiểm soát chặt chẽ, hạn chế các hiệu ứng quá phức tạp hoặc yêu cầu tài nguyên cao. Quá trình phát triển trò chơi sẽ bao gồm việc kiểm thử trên nhiều loại thiết bị khác nhau, từ máy tính có cấu hình thấp đến các thiết bị hiện đại, để đảm bảo trò chơi hoạt động ổn định và tương thích rộng rãi. Thông qua các biện pháp tối ưu hóa này, trò chơi không chỉ đảm bảo hiệu năng tốt mà còn giữ được chất lượng đồ họa và âm thanh ở mức cao nhất có thể.

## II. Giao diện trò chơi

Giao diện trong trò chơi (in-game interface) là một thành phần quan trọng, cung cấp thông tin cần thiết cho người chơi và tạo trải nghiệm trực quan, hấp dẫn. Trong trò chơi Flatform Game Bắn UFO, giao diện trò chơi được thiết kế để tối ưu hóa sự tương tác giữa người chơi và các yếu tố trong trò chơi.

### 2.1. Thiết kế menu chính

Giao diện menu chính của trò chơi Flatform Game Bắn UFO sẽ được thiết kế đơn giản nhưng bắt mắt, đảm bảo người chơi có thể dễ dàng tiếp cận các chức năng chính ngay từ lần đầu tiên vào trò chơi. Các nút chức năng như "Bắt đầu" và "Thoát" sẽ được sắp xếp khoa học và dễ nhìn, nằm ở vị trí trung tâm của màn hình để thuận tiện cho việc lựa chọn. Bối cảnh của menu sẽ được lấy cảm hứng từ không gian vũ trụ, với những ngôi sao lấp lánh, hành tinh xa xôi và ánh sáng mờ ảo, tạo nên một không gian huyền bí và kích thích sự tò mò của người chơi. Hiệu ứng động nhẹ nhàng như các ngôi sao chuyển động hoặc lấp lánh sẽ được thêm vào để làm cho giao diện trở nên sinh động và hấp dẫn hơn, đồng thời tăng thêm cảm giác thích thú và sự cuốn hút ngay từ màn hình đầu tiên.

### 2.2. Thiết kế giao diện trong trò chơi

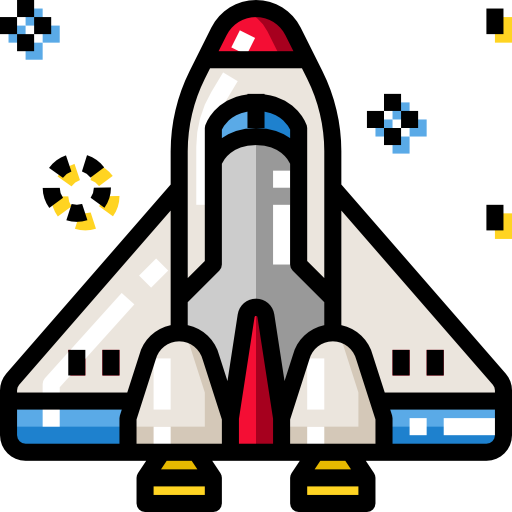
Giao diện trong trò chơi sẽ được tối ưu hóa để hiển thị đầy đủ các thông tin cần thiết một cách trực quan và dễ nhìn. Các yếu tố chính của giao diện bao gồm thanh máu của nhân vật, số điểm, thời gian còn lại (nếu có), và các loại vũ khí đang sở hữu. Thanh máu sẽ được hiển thị ở góc trên bên trái màn hình, giúp người chơi dễ dàng theo dõi tình trạng sức khỏe của nhân vật. Điểm số sẽ được đặt ở góc trên bên phải để người chơi có thể nhận biết số điểm đã đạt được sau mỗi lần tiêu diệt UFO. Nếu trò chơi có giới hạn thời gian, một đồng hồ đếm ngược sẽ được hiển thị ở phía trên, thông báo về thời gian còn lại. Ngoài ra, nếu trò chơi có nhiều loại vũ khí, một danh sách các vũ khí hiện có sẽ được hiển thị ở góc màn hình để người chơi có thể lựa chọn và sử dụng. Các hiệu ứng đồ họa như tia laser, vụ nổ khi UFO bị tiêu diệt, sẽ được sử dụng để tạo nên một không khí chiến đấu căng thẳng và kịch tính. Màn hình chính sẽ hiển thị đầy đủ những yếu tố này một cách rõ ràng, giúp người chơi dễ dàng nắm bắt thông tin và tăng cường sự hứng thú khi tham gia vào các pha hành động.

### 2.3. Phong cách đồ họa và hiệu ứng âm thanh

Đồ họa của trò chơi sẽ sử dụng phong cách 2D hiện đại, với màu sắc tươi sáng và các hiệu ứng chuyển động mượt mà để tạo nên một không gian vũ trụ bao la và sống động. Các nhân vật như phi thuyền và UFO sẽ được thiết kế chi tiết, nổi bật trên nền không gian rộng lớn, giúp người chơi dễ dàng nhận diện. Các hiệu ứng chuyển động, như tia laser khi bắn đạn hoặc ánh sáng từ vụ nổ, sẽ được tối ưu hóa để tạo ra sự chuyển động mượt mà, tăng thêm cảm giác sinh động cho trò chơi. Bên cạnh đó, âm thanh sẽ đóng một vai trò quan trọng trong việc tạo ra không khí cho trò chơi. Nhạc nền sẽ được thiết kế sôi động, giữ cho người chơi luôn có động lực và hứng khởi trong suốt quá trình chơi. Các hiệu ứng âm thanh sống động như tiếng nổ khi UFO bị tiêu diệt, tiếng bắn đạn, và tiếng la hét của kẻ thù sẽ làm tăng tính hấp dẫn và hồi hộp cho trò chơi, tạo ra sự tương tác mạnh mẽ giữa người chơi và môi trường trò chơi. Dưới đây là hỉnh ảnh sơ bộ về các nhân vậy trong game

1. **Nhân vật chính( Player )**:

o Phi thuyền của người chơi được thiết kế đơn giản nhưng dễ nhận diện. Hình dáng phi thuyền được làm nổi bật với hiệu ứng chuyển động mượt mà khi di chuyển.



Hình 3.1 Mô hình phi thuyền

2. **Kẻ thù (UFO)**:

o UFO có nhiều kiểu dáng và kích thước khác nhau, từ UFO nhỏ dễ tiêu diệt đến Boss lớn với hành vi phức tạp.

o Hiệu ứng va chạm khi đạn bắn trúng kẻ thù tạo cảm giác sống động, bao gồm các hoạt ảnh nổ và âm thanh.



Hình 3.2 Mô hình đĩa bay

3. **Hiệu ứng bắn đạn**:

o Khi phi thuyền bắn đạn, mỗi viên đạn được hiển thị với màu sắc và chuyển động rõ ràng, giúp người chơi dễ dàng quan sát và điều chỉnh chiến thuật.



Hình 3.3 Hiệu ứng đạn bắn

4. **Nền trò chơi**:

o Nền vũ trụ động với các ngôi sao hoặc hành tinh chuyển động nhẹ, tạo cảm giác về một không gian rộng lớn và sống động.



Hình 3.4 Hình nền

### 2.4. Thiết kế hệ thống

Trong phần thiết kế hệ thống, mục tiêu là xây dựng một cấu trúc mã nguồn rõ ràng, dễ hiểu và dễ duy trì, đồng thời đảm bảo tính mở rộng và khả năng mở rộng trong tương lai. Hệ thống trò chơi Flatform Game Bắn UFO sẽ được tổ chức theo mô hình lập trình hướng đối tượng (OOP), giúp dễ dàng quản lý các đối tượng trong trò chơi như nhân vật, kẻ thù, đạn, và các thành phần khác. Cấu trúc hệ thống sẽ được chia thành nhiều lớp (classes), mỗi lớp đại diện cho một thành phần trong trò chơi, với các thuộc tính (properties) và phương thức (methods) riêng biệt.

Lớp Player sẽ quản lý các thuộc tính và hành vi của nhân vật người chơi, bao gồm các thuộc tính như sức khỏe (HP), vị trí (x, y), và tốc độ di chuyển. Lớp này cũng sẽ có các phương thức như move() để di chuyển nhân vật và shoot() để bắn đạn. Đạn sẽ được quản lý bởi lớp Bullet, với các thuộc tính như vị trí, tốc độ, và phương thức update() để cập nhật chuyển động của đạn. Kẻ thù, hay UFO, sẽ được quản lý bởi lớp Enemy, với các thuộc tính như vị trí, tốc độ di chuyển, và các phương thức xử lý va chạm với đạn hoặc người chơi.

Lớp Game sẽ là lớp điều phối chính, chịu trách nhiệm quản lý vòng lặp trò chơi và điều khiển các thành phần trong trò chơi. Lớp này sẽ quản lý các đối tượng như người chơi, kẻ thù, đạn, điểm số, và trạng thái trò chơi (chơi, tạm dừng, kết thúc). Nó cũng sẽ xử lý việc khởi tạo các đối tượng, kiểm tra va chạm, và cập nhật màn hình hiển thị. Các lớp như Menu và Settings sẽ xử lý giao diện người dùng và các tùy chọn, bao gồm việc hiển thị menu chính, hướng dẫn, và các thiết lập trò chơi như âm thanh, độ khó.

Với cách tiếp cận này, mã nguồn của trò chơi sẽ có tính tổ chức tốt, dễ dàng bảo trì và mở rộng trong tương lai. Khi cần thêm tính năng hoặc thay đổi bất kỳ thành phần nào trong trò chơi, chúng ta chỉ cần điều chỉnh hoặc thêm các lớp và phương thức phù hợp mà không làm ảnh hưởng đến các phần khác của hệ thống. Ngoài ra, hệ thống sẽ được thiết kế sao cho có thể dễ dàng mở rộng với các chế độ chơi mới, đối tượng mới, hoặc các tính năng khác mà không gặp phải sự cố về tương thích hoặc hiệu suất.

# CHƯƠNG 4: QUÁ TRÌNH TRIỂN KHAI

## I. Cấu trúc mã nguồn

Cấu trúc mã nguồn của trò chơi **Flatform Game Bắn UFO** được tổ chức một cách rõ ràng và hợp lý để đảm bảo tính dễ bảo trì, mở rộng và kiểm thử. Các tệp mã nguồn chính sẽ bao gồm **game.py** và **menu.py**, mỗi tệp đảm nhận một vai trò cụ thể trong việc phát triển trò chơi. Toàn bộ hệ thống trò chơi sẽ được xây dựng dựa trên mô hình lập trình hướng đối tượng (OOP), giúp phân chia các thành phần và tính năng của trò chơi thành các đối tượng dễ dàng quản lý và phát triển.

### 1.1. Các tệp mã nguồn chính

1. **game.py** Đây là tệp chính trong trò chơi, nơi chứa logic trò chơi và vòng lặp chính. Các đối tượng quan trọng như **Player**, **Enemy**, và **Bullet** sẽ được quản lý và cập nhật trong tệp này. Mã nguồn trong **game.py** sẽ thực hiện các công việc sau:

o **Khởi tạo trò chơi**: Tệp này khởi tạo cửa sổ trò chơi, tải tài nguyên (hình ảnh, âm thanh), và thiết lập các đối tượng ban đầu.

o **Vòng lặp trò chơi**: Trong vòng lặp chính của trò chơi, tệp này sẽ kiểm tra các sự kiện (nhấn phím, di chuyển chuột), cập nhật trạng thái của các đối tượng (di chuyển nhân vật, bắn đạn, xuất hiện UFO), và vẽ lại mọi thứ lên màn hình.

o **Kiểm tra va chạm**: Các phương thức trong tệp này sẽ xử lý việc kiểm tra va chạm giữa đạn và UFO, cũng như giữa nhân vật và UFO, qua đó xác định các hành động như giảm máu hoặc tăng điểm.

2. **menu.py** Tệp này xử lý giao diện menu chính của trò chơi. Giao diện menu sẽ bao gồm các tùy chọn như **Bắt đầu**, **Hướng dẫn** và **Thoát**. Tệp này sẽ thực hiện các công việc như:

o **Hiển thị menu**: Giao diện chính sẽ được thiết kế với các nút chức năng đặt ở vị trí trung tâm màn hình, dễ dàng tương tác với người chơi.

o **Xử lý sự kiện**: Khi người chơi nhấp vào các nút trên menu, các sự kiện sẽ được xử lý, chuyển đến màn chơi hoặc thoát trò chơi.

o **Hiệu ứng động**: Các hiệu ứng động, như ánh sáng nhấp nháy của các nút, sẽ được thêm vào để làm cho menu trở nên hấp dẫn và sinh động hơn.

### 1.2. Các lớp và đối tượng trong mã nguồn

Trong trò chơi **Flatform Game Bắn UFO**, việc sử dụng mô hình lập trình hướng đối tượng (OOP) giúp quản lý các thành phần của trò chơi một cách hiệu quả. Các lớp chính trong trò chơi bao gồm **Player**, **Enemy**, **Bullet**, và **Game**. Mỗi lớp có nhiệm vụ riêng biệt và các phương thức để điều khiển hành vi của các đối tượng trong trò chơi.

1. **Lớp Player (Trong game.py)** Lớp **Player** quản lý các thuộc tính và hành vi của nhân vật chính trong trò chơi. Các thuộc tính và phương thức của lớp này bao gồm:

o **Thuộc tính**:

§ hp: Sức khỏe của nhân vật, giảm khi bị UFO va chạm.

§ position: Vị trí hiện tại của nhân vật trên màn hình.

§ speed: Tốc độ di chuyển của nhân vật.

o **Phương thức**:

§ move(): Di chuyển nhân vật theo các hướng trái, phải, lên, xuống.

§ shoot(): Bắn đạn về phía trước.

§ update(): Cập nhật vị trí của nhân vật và kiểm tra va chạm với UFO.

2. **Lớp Enemy (Trong game.py)** Lớp **Enemy** quản lý các kẻ thù trong trò chơi, là các UFO. Các thuộc tính và phương thức của lớp này bao gồm:

o **Thuộc tính**:

* position: Vị trí của UFO trên màn hình.
* speed: Tốc độ di chuyển của UFO.

o **Phương thức**:

* move(): Di chuyển UFO theo hướng ngẫu nhiên hoặc đường thẳng.
* update(): Cập nhật vị trí UFO và kiểm tra va chạm với đạn hoặc người chơi.

3. **Lớp Bullet (Trong game.py)** Lớp **Bullet** quản lý các viên đạn mà nhân vật bắn ra. Các thuộc tính và phương thức của lớp này bao gồm:

o **Thuộc tính**:

* position: Vị trí của viên đạn trên màn hình.
* speed: Tốc độ di chuyển của đạn.

o **Phương thức**:

* update(): Cập nhật vị trí của đạn và kiểm tra va chạm với các UFO.
* draw(): Vẽ viên đạn lên màn hình.

4. **Lớp Game (Trong game.py)** Lớp **Game** điều phối toàn bộ trò chơi. Nó sẽ quản lý vòng lặp trò chơi, khởi tạo các đối tượng và xử lý logic trò chơi. Các phương thức của lớp này bao gồm:

o run(): Quản lý vòng lặp chính của trò chơi, kiểm tra các sự kiện, cập nhật đối tượng, và vẽ lại mọi thứ lên màn hình.

o update(): Cập nhật các đối tượng trong trò chơi, kiểm tra va chạm và thay đổi trạng thái (ví dụ: tăng điểm, giảm máu).

o draw(): Vẽ các đối tượng lên màn hình, bao gồm nhân vật, UFO, đạn, và các hiệu ứng đặc biệt.

### 1.3. Quản lý tài nguyên

Tất cả các tài nguyên như hình ảnh, âm thanh, và hiệu ứng sẽ được lưu trữ trong các thư mục riêng biệt để dễ dàng truy cập và sử dụng trong mã nguồn. Các tài nguyên này sẽ bao gồm:

* **Hình ảnh**: Các tệp hình ảnh cho nhân vật, UFO, đạn, nền trò chơi và các hiệu ứng.
* **Âm thanh**: Các tệp âm thanh cho nhạc nền và hiệu ứng âm thanh như tiếng bắn đạn, vụ nổ, và âm thanh khi UFO bị tiêu diệt.
* Các tệp tài nguyên này sẽ được tải và giải phóng khi cần thiết thông qua các phương thức của các lớp như **Player**, **Enemy**, **Bullet**, giúp tối ưu hóa bộ nhớ và hiệu suất trò chơi.

### 1.4. Kiểm thử và duy trì mã nguồn

Cấu trúc mã nguồn này sẽ giúp việc kiểm thử và bảo trì trở nên dễ dàng hơn. Các phương thức trong các lớp như **Player**, **Enemy**, **Bullet** sẽ được kiểm tra độc lập qua các trường hợp kiểm thử đơn vị (unit tests) để đảm bảo các chức năng như di chuyển nhân vật, bắn đạn và va chạm hoạt động chính xác. Ngoài ra, việc kiểm thử tích hợp sẽ được thực hiện để đảm bảo rằng các thành phần trong trò chơi hoạt động ăn khớp với nhau trong suốt vòng lặp trò chơi. Kiểm thử sẽ được thực hiện cả trên máy tính để bàn và thiết bị di động để đảm bảo tính ổn định và hiệu suất của trò chơi trên nhiều nền tảng khác nhau.

## II. Các thành phần chính của hệ thống

Trong phần này, chúng ta sẽ đi sâu vào các thành phần chính của hệ thống trò chơi Flatform Game Bắn UFO. Mỗi thành phần trong hệ thống sẽ có một vai trò cụ thể và tác động trực tiếp đến hoạt động của trò chơi. Các thành phần này bao gồm quản lý các đối tượng trò chơi, cập nhật trạng thái và hành vi của các đối tượng, xử lý sự kiện người dùng, và quản lý tài nguyên. Việc hiểu rõ cách các thành phần này phối hợp với nhau giúp cho việc phát triển trò chơi trở nên logic và dễ dàng mở rộng.

### 2.1. Quản lý các đối tượng trong trò chơi

Một trong những thành phần quan trọng trong trò chơi là việc quản lý các đối tượng như nhân vật, kẻ thù, đạn, và các hiệu ứng đồ họa. Trong trò chơi Flatform Game Bắn UFO, các đối tượng này sẽ được đại diện bằng các lớp riêng biệt, mỗi lớp có các thuộc tính và phương thức riêng. Lớp Player quản lý nhân vật người chơi, với các thuộc tính như máu, vị trí và tốc độ di chuyển. Lớp Enemy sẽ quản lý các kẻ thù trong trò chơi, mỗi UFO sẽ có các thuộc tính như vị trí, tốc độ di chuyển và phương thức để di chuyển trong không gian trò chơi. Đối với đạn, lớp Bullet sẽ đại diện cho các viên đạn được bắn ra từ nhân vật, với các thuộc tính về vị trí, tốc độ và hướng di chuyển.

Mỗi đối tượng này sẽ được tạo ra, di chuyển, và cập nhật thông qua các phương thức trong các lớp của chúng. Các đối tượng sẽ được theo dõi và quản lý liên tục trong suốt vòng lặp trò chơi, từ khi chúng xuất hiện cho đến khi chúng bị phá hủy hoặc không còn tồn tại nữa. Việc tách biệt các đối tượng này giúp hệ thống trò chơi dễ dàng mở rộng, cho phép thêm nhiều đối tượng khác vào trong trò chơi mà không ảnh hưởng đến các đối tượng hiện tại.

### 2.2. Cập nhật trạng thái và hành vi của các đối tượng

Khi trò chơi diễn ra, tất cả các đối tượng trong trò chơi cần phải được cập nhật liên tục để thể hiện đúng hành vi và trạng thái của chúng. Các đối tượng như nhân vật, kẻ thù và đạn sẽ thay đổi vị trí theo thời gian, và các hành động như bắn đạn, va chạm hoặc tiêu diệt kẻ thù sẽ cần được xử lý.

Lớp Game sẽ chịu trách nhiệm chính trong việc điều phối tất cả các hành động này. Nó sẽ gọi các phương thức update() của từng đối tượng để cập nhật vị trí, kiểm tra va chạm và xử lý các thay đổi trong trạng thái. Ví dụ, mỗi khi nhân vật di chuyển, phương thức move() trong lớp Player sẽ được gọi để thay đổi vị trí của nhân vật trên màn hình. Cùng lúc đó, phương thức update() trong lớp Enemy sẽ được gọi để di chuyển UFO, kiểm tra va chạm với nhân vật hoặc đạn, và cập nhật các trạng thái khác như máu của kẻ thù.

Ngoài việc cập nhật hành vi của các đối tượng, việc kiểm tra va chạm là một phần không thể thiếu trong quá trình cập nhật. Khi đạn bắn ra từ nhân vật, phương thức check\_collision() sẽ được sử dụng để kiểm tra xem đạn có va chạm với UFO hay không. Nếu có, UFO sẽ bị tiêu diệt và người chơi sẽ được cộng điểm.

### 2.3. Xử lý sự kiện người dùng

Xử lý sự kiện người dùng là một phần quan trọng để người chơi có thể tương tác với trò chơi. Các sự kiện này bao gồm nhấn phím, di chuyển chuột hoặc các thao tác trên màn hình cảm ứng, tùy thuộc vào thiết bị mà trò chơi đang chạy.

Trong Flatform Game Bắn UFO, lớp Game sẽ giám sát và xử lý tất cả các sự kiện người dùng. Ví dụ, khi người chơi nhấn phím mũi tên hoặc di chuyển chuột, các sự kiện này sẽ được nhận diện và chuyển thành các hành động trong trò chơi, như di chuyển nhân vật hoặc bắn đạn. Lớp Player sẽ nhận sự kiện từ Game và thay đổi vị trí của nhân vật hoặc thực hiện hành động bắn đạn. Ngoài ra, lớp Menu cũng sẽ xử lý các sự kiện trong menu chính, cho phép người chơi chọn các tùy chọn như bắt đầu trò chơi hoặc thoát ra khỏi trò chơi.

Việc xử lý sự kiện là rất quan trọng để tạo ra sự tương tác giữa người chơi và trò chơi. Mỗi hành động của người chơi phải được phản hồi ngay lập tức trong trò chơi, giúp người chơi cảm thấy như họ đang điều khiển một cách trực tiếp các yếu tố trong trò chơi.

### 2.4. Quản lý tài nguyên

Tài nguyên trong trò chơi bao gồm hình ảnh, âm thanh, và các hiệu ứng đồ họa. Quản lý tài nguyên là một yếu tố quan trọng giúp trò chơi hoạt động mượt mà và hiệu quả. Trong Flatform Game Bắn UFO, các tài nguyên sẽ được lưu trữ trong các thư mục riêng biệt và được tải khi cần thiết trong quá trình trò chơi.

Các hình ảnh như nhân vật, UFO, đạn và nền trò chơi sẽ được lưu trữ dưới dạng các tệp hình ảnh (PNG, JPG, v.v.). Các tài nguyên âm thanh như nhạc nền, tiếng bắn đạn và tiếng nổ của UFO sẽ được lưu trữ dưới dạng các tệp âm thanh (MP3, WAV, v.v.). Mỗi khi trò chơi cần một tài nguyên, phương thức tương ứng sẽ được gọi để tải tài nguyên đó và sử dụng trong trò chơi. Sau khi không cần nữa, tài nguyên sẽ được giải phóng để tiết kiệm bộ nhớ và tối ưu hóa hiệu suất.

Quá trình quản lý tài nguyên cần phải được thực hiện một cách cẩn thận, tránh tình trạng tải quá nhiều tài nguyên cùng một lúc, gây giảm hiệu suất trò chơi. Việc giải phóng tài nguyên khi không còn sử dụng sẽ giúp trò chơi không bị giật lag, đặc biệt khi trò chơi chạy trên các thiết bị có cấu hình thấp hoặc khi sử dụng trên nền tảng di động.

## III. Xử lý và quản lý sự kiện trong trò chơi

Xử lý và quản lý sự kiện là một phần không thể thiếu trong bất kỳ trò chơi nào, giúp trò chơi tương tác được với người chơi thông qua các hành động như nhấn phím, di chuyển chuột, hoặc chạm màn hình. Trong **Flatform Game Bắn UFO**, các sự kiện được xử lý một cách chi tiết để đảm bảo sự tương tác mượt mà và phản hồi nhanh chóng từ phía trò chơi. Mỗi sự kiện sẽ được nhận diện và chuyển thành các hành động cụ thể, như di chuyển nhân vật, bắn đạn, hoặc thực hiện các thao tác trong menu chính.

### 3.1. Xử lý sự kiện người dùng

Trong Flatform Game Bắn UFO, lớp Game sẽ chịu trách nhiệm chính trong việc quản lý và xử lý các sự kiện từ người dùng. Mỗi sự kiện, chẳng hạn như nhấn phím, di chuyển chuột, hay chạm màn hình, sẽ được xử lý một cách tức thời để thay đổi trạng thái của trò chơi. Ví dụ, khi người chơi nhấn các phím mũi tên trên bàn phím hoặc di chuyển chuột, lớp Player sẽ nhận biết sự kiện này và thực hiện hành động tương ứng, như di chuyển nhân vật sang trái, phải, lên hoặc xuống. Nếu người chơi nhấn phím bắn (hoặc nút chuột trái), phương thức shoot() trong lớp Player sẽ được gọi để tạo ra một viên đạn mới và bắn ra từ vị trí của nhân vật.

Việc xử lý sự kiện này cần phải rất nhanh chóng và mượt mà, đảm bảo rằng mọi thao tác của người chơi sẽ được phản ánh ngay lập tức trong trò chơi mà không có độ trễ đáng kể. Mỗi lần người chơi thực hiện hành động, hệ thống phải phản hồi ngay lập tức, tạo cảm giác mượt mà và tự nhiên, giúp nâng cao trải nghiệm chơi game.

### 3.2 Quản lý sự kiện trên menu

Bên cạnh việc xử lý sự kiện trong phần chơi chính của trò chơi, lớp Menu trong menu.py cũng đóng vai trò quan trọng trong việc quản lý các sự kiện trong giao diện menu. Khi người chơi mở trò chơi và truy cập vào menu chính, các sự kiện từ các nút như Bắt đầu, Hướng dẫn, và Thoát sẽ được nhận diện và xử lý. Khi người chơi chọn "Bắt đầu", trò chơi sẽ chuyển sang màn chơi chính. Khi chọn "Hướng dẫn", một cửa sổ hoặc màn hình hướng dẫn sẽ được hiển thị để giải thích cách chơi. Nếu người chơi chọn "Thoát", trò chơi sẽ kết thúc hoặc quay lại màn hình chính. Những sự kiện này sẽ được lớp Menu xử lý thông qua các phương thức kiểm tra sự kiện chuột hoặc phím, và sau đó thực hiện các hành động chuyển đổi màn hình tương ứng. Đảm bảo rằng các sự kiện trong menu được xử lý mượt mà giúp trò chơi không bị gián đoạn và giữ được tính trực quan cho người chơi ngay từ bước đầu tiên khi vào trò chơi.

### 3.3 Cập nhật và phản hồi các sự kiện

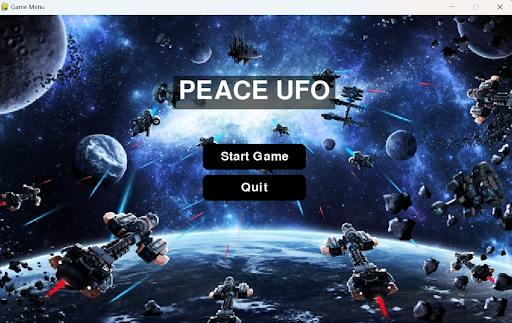
Sau khi các sự kiện được xử lý, trò chơi cần cập nhật trạng thái và hành vi của các đối tượng trong trò chơi. Các sự kiện người dùng sẽ dẫn đến việc thay đổi các thuộc tính của đối tượng, chẳng hạn như thay đổi vị trí của nhân vật, tạo đạn mới hoặc cập nhật các thông số khác như điểm số và máu. Ví dụ, khi nhân vật di chuyển, vị trí của nhân vật sẽ thay đổi và được vẽ lại trên màn hình, đồng thời kiểm tra va chạm giữa đạn và UFO sẽ được thực hiện ngay sau đó.

Các thay đổi này cần được cập nhật liên tục trong mỗi vòng lặp của trò chơi để đảm bảo rằng người chơi luôn thấy được sự phản hồi trực quan đối với hành động của mình. Hệ thống sẽ không chỉ cập nhật các đối tượng mà còn quản lý các yếu tố như điểm số, thời gian chơi, và các hiệu ứng đặc biệt (như vụ nổ hoặc hiệu ứng âm thanh) để mang đến một trải nghiệm chơi game hoàn thiện.

## IV. Giao diện trò chơi

### 4.1 Giao diện menu

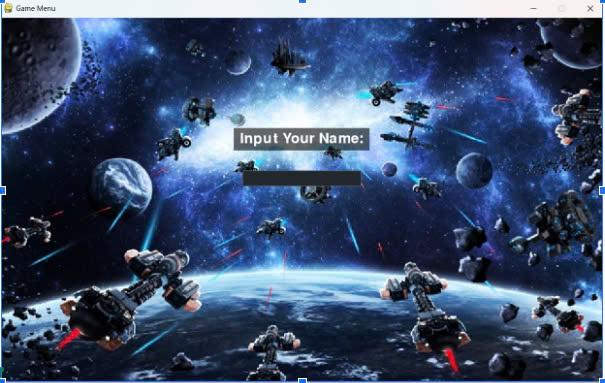
Khi khởi động game sẽ xuất hiện các nút như “start game”, “quit”



Hình 4.5 Giao diện menu

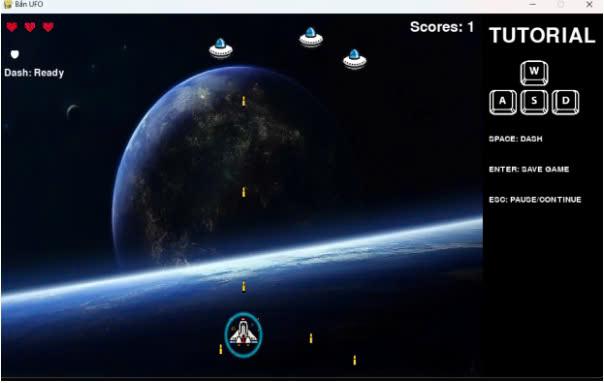
### 4.2 Giao diện khởi đầu

Giao diện này sẽ yêu cầ mọi người nhập tên để có thể lưu tiến trình sau này



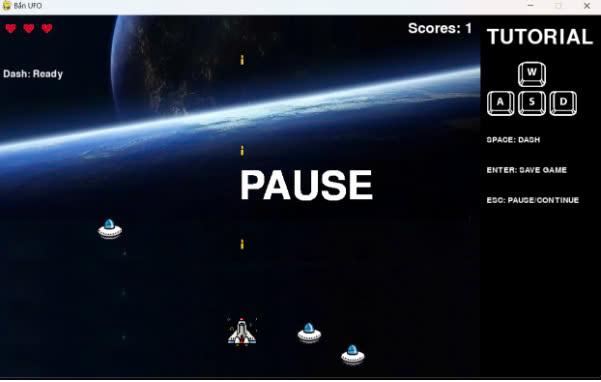
Hình 4.6 Hình giao diện khởi đầu

### 4.3 Giao diện trò chơi



Hình 4.7 Giao diện trò chơi

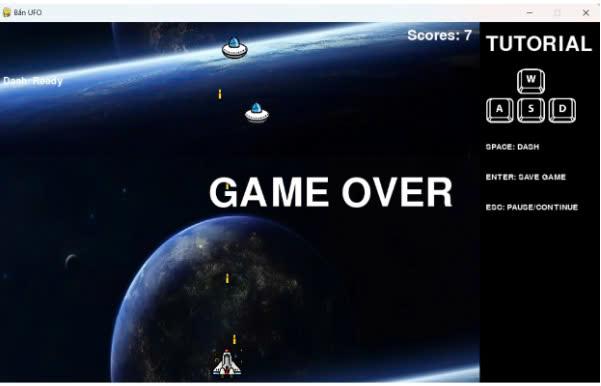
### 4.4 Giao diện tạm dừng



Hình 4.8 Giao diện tạm dừng

### 4.5 Giao diện game over

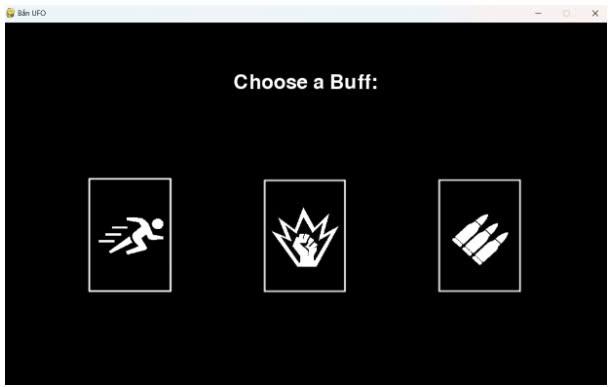
Game Over sẽ hiện ra khi hp của nhân vật trở về 0



Hình 4.9 Giao diện game over

### 4.6. Giao diện khi nhân vật chọn buff

Giao diện chọn các buff để tăng sức mạnh cho nhân vật khi đánh bại boss. có 5 loại buff cơ bản là tăng tốc độ, tăng tấn công, tăng lượng đạn, tăng máu, nhân khiên mỗi khi đánh bại boss sẽ random ra 3\5 buff trên



Hình 4.10 Giao diện chọn buff

# CHƯƠNG 5: KẾT LUẬN

## I. Kết quả đạt được

Trò chơi Flatform Game Bắn UFO đã hoàn thành và đáp ứng được các yêu cầu cơ bản ban đầu, bao gồm giao diện đơn giản nhưng dễ sử dụng, các chức năng điều khiển nhân vật, bắn đạn và tiêu diệt kẻ thù. Các thành phần trong trò chơi, như Player, Enemy, và Bullet, đã được xây dựng và hoạt động chính xác theo các yêu cầu đã đề ra. Nhân vật có thể di chuyển theo bốn hướng, bắn đạn, và tiêu diệt UFO, trong khi các UFO sẽ di chuyển ngẫu nhiên và gia tăng độ khó theo thời gian.

Giao diện của trò chơi đã được thiết kế trực quan và dễ sử dụng, với các thanh hiển thị máu, điểm số và các thông tin cần thiết. Các hiệu ứng đồ họa như tia laser, vụ nổ khi UFO bị tiêu diệt và hiệu ứng âm thanh sống động đã được tích hợp để tăng tính hấp dẫn của trò chơi. Những hiệu ứng này không chỉ giúp tạo ra không khí chiến đấu căng thẳng mà còn làm trò chơi trở nên sinh động hơn.

Điểm đáng chú ý là trò chơi đã hoàn thành các yêu cầu về tối ưu hiệu năng. Trò chơi hoạt động mượt mà trên các thiết bị khác nhau, đặc biệt là khi được kiểm thử trên các nền tảng di động và máy tính với cấu hình phần cứng khác nhau. Nhờ vào việc tối ưu hóa thuật toán và quản lý tài nguyên hợp lý, trò chơi không gặp phải tình trạng giật lag hoặc treo máy khi số lượng đối tượng trên màn hình tăng lên.

## II. Ưu điểm của trò chơi

Một trong những ưu điểm nổi bật của Flatform Game Bắn UFO là việc trò chơi cung cấp một trải nghiệm chơi mượt mà và dễ hiểu, thích hợp cho người mới bắt đầu. Giao diện người dùng đơn giản, dễ thao tác, cùng với các chỉ số rõ ràng như điểm số và thanh máu giúp người chơi dễ dàng theo dõi và tương tác. Các hiệu ứng đồ họa và âm thanh, mặc dù đơn giản nhưng hiệu quả, đã tạo ra một bầu không khí thú vị và hấp dẫn.

Trò chơi cũng khá dễ dàng để mở rộng và thêm các tính năng mới trong tương lai. Với việc sử dụng mô hình lập trình hướng đối tượng (OOP), mã nguồn được tổ chức rõ ràng, giúp việc duy trì và phát triển thêm các chế độ chơi hoặc tính năng mới trở nên thuận tiện. Những lớp như Player, Enemy, và Bullet có thể dễ dàng mở rộng để thêm các hành vi và thuộc tính mới mà không làm phức tạp hệ thống hiện tại.

## III. Nhược điểm của trò chơi

Mặc dù trò chơi đã hoàn thành các yêu cầu cơ bản, nhưng vẫn còn một số hạn chế. Một trong những nhược điểm là tính đa dạng của kẻ thù. Hiện tại, trò chơi chỉ có một loại kẻ thù (UFO), và sự đa dạng trong các loại kẻ thù sẽ giúp tăng tính thú vị và thử thách cho người chơi. Việc thêm nhiều loại UFO với các kỹ năng hoặc hành vi khác nhau có thể tạo ra các pha hành động hấp dẫn và phong phú hơn. Thêm vào đó, một số tính năng chưa được triển khai đầy đủ. Ví dụ, trò chơi không có hệ thống bảng xếp hạng để người chơi có thể so tài điểm số với bạn bè hoặc người chơi khác. Điều này có thể khiến trò chơi thiếu tính cạnh tranh, một yếu tố rất quan trọng đối với các trò chơi bắn súng hoặc hành động. Một vấn đề khác là hiệu suất trên thiết bị di động. Mặc dù trò chơi chạy mượt mà trên các máy tính, nhưng trên một số thiết bị di động với cấu hình phần cứng hạn chế, hiệu suất có thể giảm nếu quá nhiều đối tượng xuất hiện đồng thời trên màn hình.

## IV. Hướng phát triển trong tương lai

Dù trò chơi Flatform Game Bắn UFO đã hoàn thành và đáp ứng được các yêu cầu cơ bản, nhưng vẫn còn nhiều cơ hội để phát triển và mở rộng trò chơi trong tương lai. Các hướng phát triển trong tương lai bao gồm:

### 4.1. Tính năng mới và mở rộng gameplay

Một trong những hướng phát triển quan trọng là tăng tính đa dạng của kẻ thù. Hiện tại, trò chơi chỉ có một loại kẻ thù (UFO), nhưng để làm tăng độ khó và sự hấp dẫn cho người chơi, có thể thêm nhiều loại kẻ thù với hành vi và khả năng đặc biệt khác nhau. Ví dụ, một số UFO có thể di chuyển theo đường chéo hoặc nhanh hơn, trong khi những UFO khác có thể có khả năng bắn trả lại. Điều này không chỉ tăng thêm thử thách mà còn làm trò chơi trở nên phong phú và đa dạng hơn.

Thêm vào đó, các chế độ chơi mới có thể được triển khai để tạo ra sự mới mẻ cho người chơi. Các chế độ như Chế độ sinh tồn, Chế độ thời gian (đoán thời gian sống sót của người chơi) hay Chế độ nhiều người chơi sẽ giúp trò chơi trở nên hấp dẫn và lôi cuốn hơn. Chế độ nhiều người chơi có thể được thiết kế cho phép người chơi thi đấu với nhau hoặc hợp tác để tiêu diệt kẻ thù.

Một tính năng quan trọng có thể bổ sung là hệ thống nâng cấp vũ khí. Hiện tại, người chơi chỉ có thể bắn đạn cơ bản, nhưng với việc thêm các loại vũ khí nâng cấp như súng máy, tên lửa hoặc tia laser mạnh mẽ hơn, người chơi sẽ cảm thấy phấn khích hơn khi có thể nâng cấp sức mạnh của vũ khí trong suốt trò chơi.

### 4.2. Tối ưu hóa hiệu suất và khả năng mở rộng

Mặc dù trò chơi hiện tại hoạt động ổn định, nhưng việc tối ưu hóa hiệu suất trên các thiết bị di động và các thiết bị có cấu hình thấp sẽ là một phần quan trọng trong việc phát triển trò chơi. Việc sử dụng hiệu ứng đồ họa phức tạp và hiển thị nhiều đối tượng trên màn hình có thể khiến trò chơi bị giật lag khi chơi trên các thiết bị có phần cứng hạn chế. Một giải pháp là giảm số lượng đối tượng hoặc tối ưu hóa các thuật toán xử lý va chạm và di chuyển để tiết kiệm tài nguyên hệ thống.

Ngoài ra, việc phát triển trò chơi cho nền tảng di động (Android, iOS) là một bước tiến quan trọng. Trò chơi hiện tại có thể dễ dàng chuyển sang các thiết bị di động nhờ vào việc sử dụng Pygame và các công cụ đóng gói trò chơi. Tuy nhiên, để đảm bảo rằng trò chơi hoạt động mượt mà trên di động, các thao tác điều khiển và giao diện cần được điều chỉnh cho phù hợp với màn hình cảm ứng, thay vì sử dụng chuột hoặc bàn phím.

### 4.3. Cải tiến đồ họa và âm thanh

Để trò chơi hấp dẫn hơn, việc cải thiện đồ họa là một yếu tố quan trọng. Mặc dù trò chơi hiện tại đã sử dụng đồ họa 2D cơ bản nhưng việc áp dụng phong cách đồ họa pixel art hoặc vector hiện đại có thể mang lại một cái nhìn mới mẻ và thu hút người chơi. Các hiệu ứng ánh sáng, chuyển động mượt mà, và các chi tiết nền sẽ giúp tạo ra bầu không khí sâu sắc và hấp dẫn hơn.

Cùng với đó, cải thiện âm thanh là một hướng phát triển cần thiết. Việc bổ sung thêm âm thanh nền phong phú, cũng như các hiệu ứng âm thanh đặc sắc cho từng hành động như bắn đạn, va chạm, và nổ UFO, sẽ giúp trò chơi trở nên sống động và kích thích cảm xúc của người chơi. Ngoài ra, việc sử dụng nhạc nền thay đổi tùy theo các mức độ hoặc chế độ chơi sẽ tăng cường thêm trải nghiệm người chơi.

### 4.4. Hệ thống bảng xếp hạng và chia sẻ điểm

Một tính năng hấp dẫn mà trò chơi có thể bổ sung là hệ thống bảng xếp hạng trực tuyến. Điều này sẽ giúp người chơi so tài với nhau về điểm số, tạo ra sự cạnh tranh và thách thức. Người chơi có thể chia sẻ điểm số của mình lên các bảng xếp hạng trực tuyến hoặc tham gia các sự kiện thi đấu với bạn bè. Việc tích hợp tính năng này sẽ giúp trò chơi trở nên hấp dẫn hơn và kéo dài thời gian chơi của người dùng.

#### 4.5. Tích hợp mạng xã hội và tính năng chia sẻ

Một hướng phát triển tiếp theo là tích hợp tính năng chia sẻ trên các mạng xã hội như Facebook, Twitter, hoặc Instagram. Người chơi có thể chia sẻ thành tích của mình, mời bạn bè tham gia trò chơi, hoặc tham gia các sự kiện đặc biệt. Điều này sẽ không chỉ giúp tăng cường sự kết nối cộng đồng mà còn tạo ra cơ hội quảng bá trò chơi một cách tự nhiên.

# TÀI LIỆU SỬ DỤNG VÀ THAM KHẢO

1. **John R. Hubbard**, *Programming with C++* (4th edition), Prentice Hall, 2006.
2. **Al Sweigart**, *Making Games with Python & Pygame*, 2015.
3. **Pygame Documentation**:<https://www.pygame.org/docs/>
4. **The SDL2 Library**:<https://www.libsdl.org/>
5. **David M. Bourg**, *Python Game Programming by Example*, Packt Publishing, 2016.
6. **M. H. A. Khan**, *Game Development with Pygame*, Apress, 2018.
7. **Richard Lawrence**, *Learn Python by Building Games*, Packt Publishing, 2018.
8. **Game Design Theory: A New Philosophy for Understanding Games** - Keith Burgun, 2012.
9. **Python.org Documentation**:<https://docs.python.org/>
10. **Online Resources and Tutorials**:
    * Real Python - Pygame Tutorials
    * Pygame Wiki - Pygame Basics