KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN CƠ SỞ NGÀNH**

**HỌC KỲ I, NĂM HỌC 2024-2025**

**PHÁT TRIỂN GAME BẮN SÚNG TỌA ĐỘ**

**SỬ DỤNG MEDIAPIPE VỚI TƯƠNG TÁC ĐIỀU KHIỂN BẰNG CHỈ TAY**

*Giáo viên hướng dẫn:*

ThS. Ngô Thanh Huy

*Sinh viên thực hiện:*

Họ tên: Lý Thanh Tâm

MSSV: 110122020

Lớp: DA22TTA

***Trà Vinh, tháng 12 năm 2024***

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Trà Vinh, ngày …. . tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*Trà Vinh, ngày …. . tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

Lời nói đầu, Em xin cảm ơn thầy Ngô Thanh Huy đã hỗ trợ, hướng dẫn tận tình trong thời gian em làm thực tập cơ sở ngành, những kiến thức mà thầy đã dạy em sẽ là hành trang quý báo trên con đường học vấn và phát triển sự nghiệp tương lai rộng mở của em. Thầy đã luôn kiên nhẫn, nhiệt tình trong việc truyền đạt kiến thức và kinh nghiệm quý báo, giúp em vượt qua những khó khăn và thử thách trong quá trình học tập và nghiên cứu.

Những lời khuyên, góp ý của thầy không chỉ là kim chỉ nam cho sự phát triển của đồ án môn học này mà còn là nguồn động viên, khích lệ tinh thần lớn lao cho em.

Em xin hứa sẽ tiếp tục nỗ lực và phấn đấu không ngừng để không phụ lòng thầy đã dành cho em.

Xin chân thành cảm ơn thầy.

Trà Vinh, Tháng 12 năm 2024

**Lý Thanh Tâm**

**MỤC LỤC**

[MỞ ĐẦU 7](#_Toc187303126)

[CHƯƠNG 1: TỔNG QUAN 9](#_Toc187303127)

[1.1. Tổng quan về lập trình game 9](#_Toc187303128)

[1.2. Tổng quan về game bắn súng tọa độ 9](#_Toc187303129)

[1.3. Tầm quan trọng của việc phát triển game bắn súng tọa độ bằng Python 9](#_Toc187303130)

[1.4. Mục tiêu và định hướng của dự án 10](#_Toc187303131)

[1.5. Phân tích các yêu cầu chính 10](#_Toc187303132)

[CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT 11](#_Toc187303133)

[2.1. Cơ sở lý thuyết 11](#_Toc187303134)

[2.1.1. Chuyển động parabola trong vật lý 11](#_Toc187303135)

[2.1.2. Kiểm tra va chạm (Collision Detection) 11](#_Toc187303136)

[2.1.3. Tương tác người chơi 11](#_Toc187303137)

[2.2. Lý luận và giả thuyết khoa học 12](#_Toc187303138)

[2.2.1. Ứng dụng của lý thuyết vật lí 12](#_Toc187303139)

[2.2.2. Vai trò của kiểm tra va chạm 12](#_Toc187303140)

[2.2.3. Giao diện và trải nghiệm người dung 12](#_Toc187303141)

[2.3. Công cụ, công nghệ và phần mềm được sử dụng 12](#_Toc187303142)

[2.3.1. Ngôn ngữ lập trình Python 12](#_Toc187303143)

[2.3.2. Thư viện Pygame 12](#_Toc187303144)

[2.3.3. Công cụ toán học 13](#_Toc187303145)

[2.3.4. Công cụ phát triển mã nguồn 13](#_Toc187303146)

[2.4. Ứng dụng lý thuyết và công cụ vào đề án 13](#_Toc187303147)

[2.5. Mô hình tổ chức nội dung sách 13](#_Toc187303148)

[2.5.1. Mô hình tổ chức nội dung tổng thể 13](#_Toc187303149)

[2.5.2. Mô hình chức năng game bắn súng tọa độ 14](#_Toc187303150)

[CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU 15](#_Toc187303151)

[3.1. Quy trình phát triển phần mềm 15](#_Toc187303152)

[3.1.1. Lựa chọn mô hình phát triển 15](#_Toc187303153)

[3.1.2. Công cụ hỗ trợ 15](#_Toc187303154)

[3.2. Hồ sơ thiết kế 15](#_Toc187303155)

[3.2.1. Sơ đồ tổng quát (Overview Diagram) 15](#_Toc187303156)

[3.2.2. Sơ đồ lớp (Class Diagram) 15](#_Toc187303157)

[3.2.3. Sơ đồ hoạt động (Activity Diagram) 19](#_Toc187303158)

[3.3. Thiết kế giao diện 19](#_Toc187303159)

[3.3.1. Phác thảo giao diện 19](#_Toc187303160)

[3.3.2. Công cụ đồ họa 21](#_Toc187303161)

[3.4. Cài đặt chương trình 21](#_Toc187303162)

[3.4.1. Cấu trúc thư mục 21](#_Toc187303163)

[3.4.2. Cài đặt các module 21](#_Toc187303164)

[3.5. Kiểm thử và đánh giá 23](#_Toc187303165)

[3.5.1. Kiểm thử đơn vị 23](#_Toc187303166)

[3.5.2. Kiểm thử tích hợp 23](#_Toc187303167)

[3.5.3. Đánh giá hiệu năng 23](#_Toc187303168)

[CHƯƠNG 4: KẾT LUẬN 25](#_Toc187303169)

[4.1. Hiệu năng phần mềm 25](#_Toc187303170)

[4.1.1. Đánh giá hiệu năng tổng quát 25](#_Toc187303171)

[4.1.2. So sánh hiệu năng giữa các thiết bị 25](#_Toc187303172)

[4.2. Trài nghiệm người dung 26](#_Toc187303173)

[4.2.1. Trải nghiệm người dùng (UI) 26](#_Toc187303174)

[4.2.2. Trải nghiệm người dùng (UX) 27](#_Toc187303175)

[4.3. Kết quả thiết kế giao diện và chức năng 27](#_Toc187303176)

[4.3.1. Giao diện minh họa 27](#_Toc187303177)

[4.3.2. Chức năng nổi bật 28](#_Toc187303178)

[4.4. Kết quả kiểm thử 28](#_Toc187303179)

[4.4.1. Kiểm thử đơn vị 28](#_Toc187303180)

[4.4.2. Bảng đánh giá kết quả kiểm thử 29](#_Toc187303181)

[4.5. Đánh giá hiệu quả của tài liệu hướng dẫn 29](#_Toc187303182)

[CHƯƠNG 5: HƯỚNG PHÁT TRIỂN 31](#_Toc187303183)

[5.1. Kết luận 31](#_Toc187303184)

[5.2. Hướng phát triển 31](#_Toc187303185)

[5.2.1. Nâng cấp đồ họa và hiệu ứng 31](#_Toc187303186)

[5.2.2. Tích hợp chế độ chơi trực tuyến 32](#_Toc187303187)

[5.2.3. Mở rộng nội dung và chế độ chơi 32](#_Toc187303188)

[5.2.4. Tối ưu hóa hiệu năng 32](#_Toc187303189)

[5.2.5. Tích hợp trí tuệ nhân tạo (AI) 32](#_Toc187303190)

[5.2.6. Thương mại hóa sản phẩm 33](#_Toc187303191)

[DANH MỤC TÀI LIỆU THAM KHẢO 34](#_Toc187303192)

**DANH MỤC HÌNH ẢNH – BẢNG BIỂU**

[Hình 3.1. Đoạn mã để quản lí toàn bộ trò chơi 16](#_Toc156252437)

[Hình 3.2. Lớp Player 16](#_Toc156252438)

[Hình 3.3. Lớp Enemy 17](#_Toc156252442)

[Hình 3.4. Hàm di chuyển vầ tấn công của kẻ thù 18](#_Toc156252442)

[Hình 3.5. Lớp Bullet 19](#_Toc156252443)

[Hình 3.6. Màn hình chính của trò chơi 20](#_Toc156252444)

[Hình 3.7. Màn hình chơi game của trò chơi 20](#_Toc156252445)

[Hình 3.8. Cấu trúc thư mục của trò chơi 21](#_Toc156252446)

[Hình 3.9. Lớp mô phỏng quỹ đạo của đạn 22](#_Toc156252447)

[Hình 3.10. Hàm kiểm tra va chạm của viên đạn 22](#_Toc156252448)

[Hình 4.1. Bảng so sánh hiệu năng giữa các thiết bị 27](#_Toc156252449)

[Hình 4.2. Màn hình chơi game minh họa 28](#_Toc156252449)

[Hình 4.3. Hàm kiểm tra mô phỏng quỹ đạo của đạn 29](#_Toc156252449)

[Hình 4.4. Hàm kiểm tra va chạm của viên đạn 29](#_Toc156252449)

[Hình 4.5. Hàm tính toán sát thương của người chơi 29](#_Toc156252449)

[Hình 4.6. Hàm tính toán sát thương của kẻ thù 29](#_Toc156252449)

[Hình 4.7. Bảng đánh giá kết quả kiểm thử 29](#_Toc156252449)

**TÓM TẮT ĐỒ ÁN**

**Vấn đề nghiên cứu:**

Trong lĩnh vực phát triển game, việc xây dựng một trò chơi bắn súng tọa độ không chỉ yêu cầu kỹ năng lập trình mà còn phải có kiến thức về vật lý, toán học và xử lý đồ họa. Mục tiêu của đề tài là phát triển một trò chơi bắn súng tọa độ đơn giản với các yếu tố sau:

Gameplay dễ hiểu, phù hợp với mọi lứa tuổi, tính toán quỹ đạo đạn dựa trên các yếu tố vật lý như góc bắn, lực bắn và lực hấp dẫn và có thể tạo giao diện trực quan và trải nghiệm người dùng mượt mà hơn.

**Các hướng tiếp cận:**

Để giải quyết bài toán lập trình game bắn súng tọa độ, các hướng tiếp cận chính bao gồm:

**Sử dụng thư viện Pygame:** Đây là thư viện phổ biến để phát triển game 2D trong Python, cung cấp các công cụ để xử lý đồ họa, âm thanh và sự kiện người dung và **áp dụng kiến thức vật lý cơ bản** như sử dụng các công thức về chuyển động parabol để tính toán quỹ đạo của đạn.

**Tổ chức code theo mô hình hướng đối tượng (OOP)** giúp quản lý các thành phần như nhân vật, kẻ thù, đạn, và môi trường dễ dàng hơn và **tích hợp tính năng tương tác** bao gồm điều chỉnh góc bắn, lực bắn và kiểm tra va chạm với mục tiêu.

**Cách giải quyết vấn đề:**

**Lập kế hoạch và thiết kế trò chơi:** phác thảo ý tưởng gameplay và thiết kế giao diện người dùng cơ bản một cách đơn giản và dễ sử dụng. **Phát triển các thành phần chính** như **tạo môi trường trò chơi** và xây dựng bối cảnh, vật cản, và các yếu tố đồ họa**. Lập trình quỹ đạo đạn** bằng cách xây dựng hàm chuyển động của viên đạn sao cho đúng với thực tế cũng như **xử lý va chạm** giữa viên đạn và mục tiêu để kiểm tra xem viên đạn có trừ máu kẻ thù khi trúng đạn không.

**Thêm tính năng và tối ưu hóa** nhưcân bằng độ khó của trò chơi qua các cấp độ vàtối ưu hóa hiệu suất tối đa để chạy mượt mà trên nhiều thiết bị.

**Một số kết quả đạt được:**

Xây dựng hoàn chỉnh một trò chơi bắn súng tọa độ đơn giản với giao diện người dùng thân thiện. Cơ chế bắn chính xác dựa trên các yếu tố vật lý như trọng lực của viên đạn và các hiệu ứng trực quan khi bắn trúng mục tiêu.

Tích hợp các tính năng thú vị như tùy chỉnh góc bắn và lực bắn bằng cách di chuyển chuột và các dòng thông báo khi kết thúc trong lần chơi đó. Đồng thời cũng có những lời phản hồi từ người chơi thử nghiệm như đánh giá cao về tính giải trí và khả năng học hỏi kiến thức vật lý từ trò chơi, gameplay đơn giản giúp người chơi làm quen và tiếp cận dễ dàng với trò chơi

# MỞ ĐẦU

**Lí do chọn đề tài:**

Game bắn súng tọa độ là một thể loại trò chơi phổ biến, được yêu thích bởi tính chiến lược và sự sáng tạo trong cách chơi. Thể loại này không chỉ mang tính giải trí mà còn giúp người chơi rèn luyện tư duy logic và khả năng tính toán. Đồng thời, việc lập trình một trò chơi bắn súng tọa độ cũng là cơ hội để áp dụng và củng cố kiến thức lập trình, toán học, vật lý và xử lý đồ họa.  
Đề tài được chọn nhằm mục tiêu:

* Kết hợp giữa học tập và thực hành, phát triển kỹ năng lập trình game.
* Tìm hiểu cách ứng dụng các công cụ và thư viện Python vào thực tế.
* Khám phá cách xây dựng logic trò chơi và phát triển giao diện người dùng.

**Mục đích chọn đề tài:**

Xây dựng một sản phẩm phần mềm game bắn súng tọa độ đơn giản bằng ngôn ngữ python và hoàn thiện các ứng dụng kiến thức lý thuyết về toán học và vật lý vào việc lập trình quỹ đạo đạn và các tương tác trong trò chơi để tạo ra một nền tảng cơ bản để phát triển các trò chơi phức tạp hơn trong tương lai. Đồng thời cũng đóng góp vào tài liệu tham khảo hữu ích cho những người muốn học lập trình game bắn sung tọa độ bằng ngôn ngữ Python.

**Đối tượng nghiên cứu:**

**Đối tượng chính**: Các kỹ thuật và phương pháp lập trình để xây dựng game bắn súng tọa độ. Các **thành phần cụ thể trong trò chơi như**: tính toán quỹ đạo đạn (cơ chế vật lý), xử lý giao diện đồ họa, các sự kiện tương tác của người chơi và kiểm tra va chạm và phản hồi tương tác trong môi trường trò chơi.

**Phạm vi nghiên cứu:**

**Phạm vi kỹ thuật:** nghiên cứu và sử dụng thư viện **Pygame** để xây dựng trò chơi 2D đồng thời cũng áp dụng các công thức vật lý cơ bản cho chuyển động parabola.Cuối cùng,tập trung vào lập trình các logic trò chơi thay vì tối ưu hóa đồ họa phức tạp.

**Phạm vi nội dung trò chơi:** trò chơi đơn giản, chỉ tập trung vào cơ chế bắn súng tọa độ giữa một hoặc nhiều kẻ thù và tất nhiên sẽ không mở rộng sang các yếu tố phức tạp như AI hoặc đồ họa 3D.

**Phạm vi thời gian thực hiện**:

Do dự án chỉ thực hiện trong thời gian ngắn, nên sẽ ưu tiên các tính năng cơ bản và khả thi trong giới hạn niên luận.

# TỔNG QUAN

## Tổng quan về lập trình game

Lập trình game là một lĩnh vực quan trọng trong công nghệ thông tin, nơi các nhà phát triển sử dụng kỹ thuật lập trình để tạo ra các trò chơi điện tử phục vụ nhu cầu giải trí, giáo dục hoặc rèn luyện kỹ năng. Với sự phát triển của các ngôn ngữ lập trình và công cụ hỗ trợ, việc tạo ra một trò chơi ngày càng trở nên dễ tiếp cận hơn.

Lập trình game không chỉ đơn thuần là viết mã mà còn bao gồm các yếu tố liên quan đến việc x**ử lý đồ họa** tạo ra giao diện đẹp mắt, mượt mà, **logic trò chơi** được xây dựng cơ chế hoạt động, luật chơi và tương tác giữa các thành phần trong game. Cuối cùng, **xử lý vật lý** để mô phỏng chuyển động và các tương tác trong thế giới thực.

## Tổng quan về game bắn súng tọa độ

Game bắn súng tọa độ là một thể loại trò chơi chiến thuật, nơi người chơi điều chỉnh các thông số như góc bắn và lực bắn để nhắm mục tiêu chính xác. Thể loại này nổi bật bởi các yếu tố l**ối chơi đơn giản nhưng đầy thách thức,** người chơi cần vận dụng kiến thức toán học và chiến lược để thắng.

**Ứng dụng vật lý thực tế** vàoquỹ đạo đạn được mô phỏng theo chuyển động parabol, chịu ảnh hưởng bởi lực hấp dẫn và các yếu tố môi trường. Trò chơi p**hù hợp với mọi lứa tuổi** với cách chơi dễ tiếp cận nhưng không kém phần thú vị. Điển hình của thể loại này bao gồm các trò chơi như "Worms" hay "Gunbound", từng là tuổi thơ của nhiều người

## Tầm quan trọng của việc phát triển game bắn súng tọa độ bằng Python

Python là một ngôn ngữ lập trình mạnh mẽ và dễ học, phù hợp với cả người mới bắt đầu và các lập trình viên chuyên nghiệp. Với sự hỗ trợ của thư viện **Pygame**, Python cho phép phát triển game 2D một cách nhanh chóng và hiệu quả. Một số lý do chọn Python và Pygame để phát triển game bắn súng tọa độ:

**Dễ học và sử dụng:** Python có cú pháp đơn giản, giúp tiết kiệm thời gian lập trình.

**Thư viện mạnh mẽ:** Pygame cung cấp đầy đủ công cụ để xử lý đồ họa, âm thanh và các sự kiện tương tác.

**Tính linh hoạt:** Có thể dễ dàng mở rộng hoặc tùy chỉnh theo ý muốn.

## Mục tiêu và định hướng của dự án

Mục tiêu của dự án là xây dựng một trò chơi bắn súng tọa độ đơn giản, tập trung vào việc tạo một môi trường đồ họa 2D với giao diện thân thiện, lập trình cơ chế bắn súng chính xác dựa trên quỹ đạo vật lý và tạo cơ sở để phát triển thêm các tính năng phức tạp hơn trong tương lai, chẳng hạn như AI hoặc chế độ nhiều người chơi.

## Phân tích các yêu cầu chính

Yêu cầu chức năng: người chơi có thể điều chỉnh góc bắn và lực bắn bằng cách di chuyển chuột, hiển thị đúng quỹ đạo đạn theo thời gian thực, xử lý va chạm giữa viên đạn, mặt đất, vật cản và mục tiêu. Cuối cùng, hệ thống tính điểm sẽ dựa trên độ chính xác của người chơi.

**Yêu cầu phi chức năng:** game phải chạy mượt mà trên các thiết bị có cấu hình cơ bản từ thấp đến cao và thiết kế giao diện và trải nghiệm người dùng một cách đơn giản và dễ sử dụng.

# NGHIÊN CỨU LÝ THUYẾT

## Cơ sở lý thuyết

### Chuyển động parabola trong vật lý

**Định nghĩa và nguyên lý:** Chuyển động của vật thể chịu tác động của lực hấp dẫn thường được mô tả bởi một đường cong parabol. Trong game bắn súng tọa độ, quỹ đạo đạn được tính toán dựa trên góc bắn, vận tốc ban đầu và gia tốc trọng trường.

**Phương trình tổng quát:**

Quỹ đạo theo trục x:

Quỹ đạo theo trục y:

Từ hai phương trình trên, ta loại biến t đi thì ta sẽ được phương trình quỹ đạo:

Trong đó:

: Gia tốc trọng trường (thường là 9.8 m/s)

: Vận tốc ban đầu của đạn.

0: Góc bắn (tính bằng radian).

### Kiểm tra va chạm (Collision Detection)

**Phương pháp hình chữ nhật (Rectangle Collision):** dùng để kiểm tra va chạm khi viên đạn đạn và mục tiêu có hình dạng chữ nhật hoặc va chạm với mặt đất.

**Phương pháp hình tròn (Circle Collision):** thường được sử dụng cho mục tiêu hoặc viên đạn hình tròn, dùng để kiểm tra khoảng cách giữa hai tâm của hai thực thể nào đó.

**Phương pháp pixel (Pixel Collision):** sử dụng nếu đồ họa phức tạp, kiểm tra từng pixel để xác định va chạm (ít được dùng trong game bắn súng tọa độ đơn giản). Tuy nhiên, phương pháp này chủ yếu dùng để xác định các bước di chuyển của kẻ thù có đúng với thực tế khôngGiả thiết khoa học

### Tương tác người chơi

**Cơ chế điều chỉnh góc và lực bắn** bằng cách **s**ử dụng chuột di chuyển để người chơi thay đổi thông số như: để tăng/giảm lực bắn hoặc thay đổi góc bắn. **Công thức chuyển đổi góc từ độ sang radian**:

## Lý luận và giả thuyết khoa học

### Ứng dụng của lý thuyết vật lí

Giả thuyết rằng việc mô phỏng chính xác chuyển động parabola trong game sẽ tạo ra trải nghiệm chân thực hơn. Việc sử dụng lực hấp dẫn và góc bắn giúp người chơi có thể dự đoán được quỹ đạo, từ đó nâng cao tính chiến lược trong gameplay.

### Vai trò của kiểm tra va chạm

Xác định chính xác thời điểm đạn chạm vào mục tiêu sẽ ảnh hưởng trực tiếp đến hiệu quả trò chơi. Nếu kiểm tra va chạm không chính xác, trò chơi sẽ bị mất cân bằng và gây khó chịu cho người chơi.

### Giao ****diện và trải nghiệm người dung****

Giả thuyết rằng một giao diện trực quan, dễ điều khiển sẽ tăng mức độ hài lòng và thu hút người chơi lâu dài. Việc sử dụng đồ họa 2D và hoạt ảnh đơn giản nhưng mượt mà sẽ tối ưu hóa hiệu suất trên các thiết bị có cấu hình trung bình.

## Công cụ, công nghệ và phần mềm được sử dụng

### Ngôn ngữ lập trình Python

Vì sao ngôn ngữ Python được chọn để làm game bắn sung tọa độ? Vì ngôn ngữ này dễ học, dễ viết mã, rất phù hợp cho việc phát triển đề án trong một thời gian ngắn. Không những thế mà còn có cộng đồng lớn, không ngừng cung cấp hỗ trợ mạnh mẽ cho việc lập trình game bắn súng tọa độ.

### Thư viện Pygame

Pygame là thư viện phổ biến trong lập trình game 2D. Các thành phần chính bao gồm sau: **Display (Hiển thị) dùng để** tạo cửa sổ và các thành phần đồ họa, **Events (Sự kiện) dùng để** xử lý đầu vào từ bàn phím, chuột, **Sprites (Hình động) dùng để** quản lý các đối tượng di chuyển và tương tác với nhau và **Clock (Đồng hồ) để** đảm bảo trò chơi chạy ở tốc độ khung hình ổn định trên mọi thiết bị (FPS).

### Công cụ toán học

Các công cụ toán học sẽ bao gồm: **NumPy để** tối ưu hóa các phép tính toán học phức tạp như sự chuyển động của viên đạn, di chuyển của kẻ thù,… và **Matplotlib dùng để** kiểm tra trực quan quỹ đạo của viên đạn trong giai đoạn thử nghiệm.

### Công cụ phát triển mã nguồn

Các công cụ để phát triển đề án này có **Visual Studio Code (VS Code) đây là một t**rình soạn thảo mã code rất phổ biến hiện nay, trong đó có ngôn ngữ Python để phát triển game bắn súng tọa độ và **Git/GitHub để** lưu trữ và quản lý phiên bản mã nguồn của đề án.

## Ứng dụng lý thuyết và công cụ vào đề án

Các ứng dụng lý thuyết sẽ được vào đề án như: **quỹ đạo đạn** để mô phỏng chính xác theo chuyển động parabol bằng cách áp dụng phương trình vật lý, thiết kế **giao diện game** bằng thư viện Pygame và hiển thị đồ họa 2D mượt mà. Đồng thời, **thử nghiệm và tối ưu hóa** việc sử dụng Matplotlib để kiểm tra các thông số góc bắn, lực bắn trước khi triển khai vào game thực tế.

## Mô hình tổ chức nội dung sách

Mục này tập trung vào việc trình bày cách tổ chức nội dung, cấu trúc và mô hình của đồ án để đảm bảo tính khoa học và dễ dàng triển khai. Mô hình tổ chức nội dung đồ án được xây dựng dựa trên các thành phần chính trong game bắn súng tọa độ và các bước phát triển phần mềm.

### Mô hình tổ chức nội dung tổng thể

Nội dung đồ án được chia thành các thành phần chính, mỗi phần đảm nhiệm một chức năng cụ thể:

* **Tổng quan (Chương 1)**: Tổng quan về game và công nghệ.
* **Nghiên cứu lý thuyết (Chương 2)**:Bao gồm cơ sở lý thuyết, công cụ và công nghệ sử dụng cho đề án.
* Trình bày, đánh giá bàn luận về các kết quả (Chương 3): Bao gồm quy trình về làm game, thiết kế giao diện và kiểm thử sau đó đánh giá kết quả đạt được.
* Kết luận (Chương 4): Đánh giá hiệu suất hoạt động của game và phản hồi từ những người chơi thử nghiệm.
* Hướng phát triển (Chương 5): Nâng cấp đồ họa đẹp hơn, tạo ra nhiều thử thách, có thể tích hợp trí tuệ nhân tạo và thương mại hóa sản phẩm.

### Mô hình chức năng game bắn súng tọa độ

Cấu trúc trò chơi được chia thành các module chính:

**Module giao diện** gồmhiển thị màn hình chính, màn hình chơi game và tạo trải nghiệm người dùng mượt mà với hiệu ứng đồ họa.

**Module logic trò chơi** sẽ cóxử lý góc bắn, lực bắn và quỹ đạo, tính toán va chạm giữa đạn và mục tiêu.

**Module xử lý sự kiện** sẽtiếp nhận và xử lý hành động từ chuột của người chơi.

Trò chơi được chia thành các luồng hoạt động:

**Khởi động game** sẽ hiển thị màn hình chính với tùy chọn "Bắt đầu", "Chơi lại", "Thoát".

**Chế độ chơi** bao gồm người chơi điều chỉnh thông số (góc, lực) và bắn đạn trúng mục tiêu và sau mỗi cấp độ số lượng kẻ thù sẽ tăng lên.

**Xử lý kết quả** tính điểm sát thương lên người chơi hoặc kẻ thù khi tấn công và hiển thị trạng thái mỗi lượt đấu.

**Kết thúc game** sẽhiển thị thông báo chiến thắng nếu người chơi vượt qua hoặc thua cuộc nếu máu người chơi về 0 và màn hình sẽ gợi ý chơi lại hoặc thoát.

# HIỆN THỰC HÓA NGHIÊN CỨU

## Quy trình phát triển phần mềm

### Lựa chọn mô hình phát triển

Dựa trên đặc điểm của đồ án và thời gian thực hiện, **mô hình phát triển phần mềm Agile** được chọn. Agile giúp chia dự án thành các vòng lặp nhỏ (iteration), cho phép phản hồi nhanh chóng và liên tục cải thiện sản phẩm. Các vòng lặp này bao gồm: phát triển các chức năng cơ bản (góc bắn, lực bắn, quỹ đạo), Thêm mục tiêu sau mỗi cấp độ và tính toán va chạm, cải thiện giao diện và trải nghiệm người dung và tối ưu hóa hiệu năng và sửa lỗi.

### Công cụ hỗ trợ

Các công cụ hỗ trợ sẽ quản lí các thành phẩn sau:

**Quản lý dự án**: sử dụng Trello để theo dõi các công việc trong từng vòng lặp.

**Quản lý mã nguồn**: dùng Git/GitHub để lưu trữ và theo dõi thay đổi mã nguồn.

**Môi trường lập trình**: PyCharm/VS Code được sử dụng để phát triển chương trình, đặc biệt là lập trình.

**Thiết kế đồ họa**: sử dụng GIMP và Inkscape để tạo hình ảnh địa hình, nhân vật, và giao diện.

**Kiểm thử và Debugging**: sử dụng Pytestđể kiểm thử đơn vị và Pygame để kiểm tra trực quan.

## Hồ sơ thiết kế

### Sơ ****đồ tổng quát (Overview Diagram)****

Hệ thống được tổ chức thành ba lớp lớn:

**Frontend (Giao diện)**: hiển thị thông tin và trạng thái game và cung cấp đầu vào để người chơi tương tác (chuột, bàn phím).

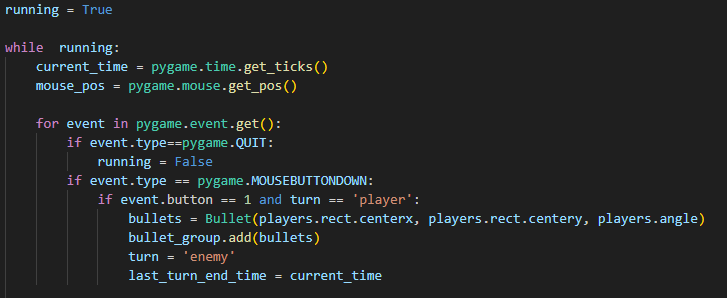
**Backend (Logic)**: xử lý các thuật toán chính: quỹ đạo đạn, va chạm, tính điểm sát thương và quản lý trạng thái trò chơi (các lượt chơi, mục tiêu, và kết thúc trò chơi).

**Assets (Tài nguyên)**: chứa các hình ảnh, phông chữ phục vụ cho thiết kế giao diện và các tính năng.

### Sơ đồ lớp (Class Diagram)

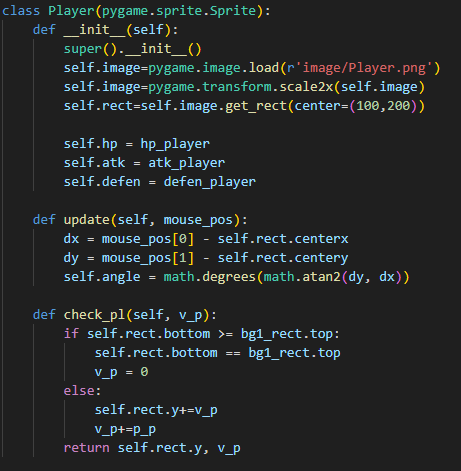
Cấu trúc của chương trình bao gồm các lớp chính như sau:

Lớp Game sẽ quản lý toàn bộ trò chơi, kiểm soát trạng thái, bắn đạn của người chơi và mọi luồng chơi.



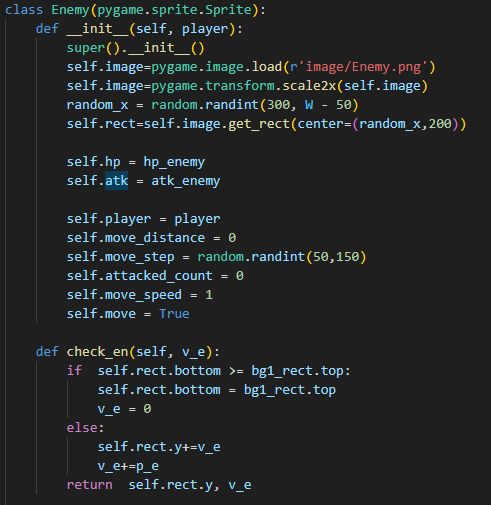
Hình 3.1. Đoạn mã để quản lí toàn bộ trò chơi

Lớp Player sẽ đại diện cho người chơi, quản lý thông tin và các hành động.

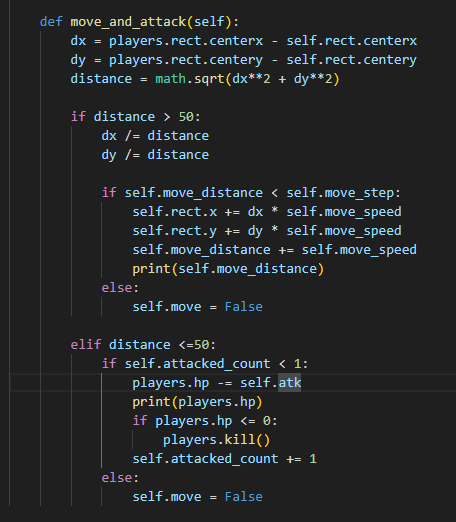


Hình 3.2. Lớp Player

Lớp Enemy sẽ đại diện cho kẻ thù, quản lý thông tin, di chuyển và tấn công người chơi khi ở gần.

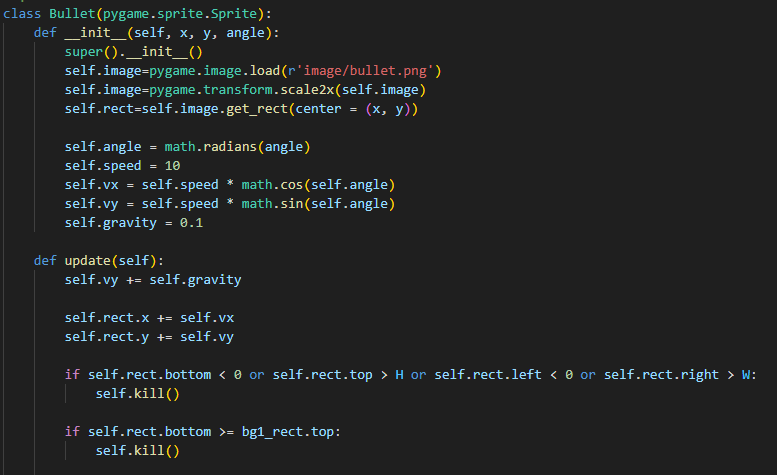


Hình 3.3. Lớp Enemy



Hình 3.4. Hàm di chuyển vầ tấn công của kẻ thù

Lớp Projectile: sẽ xử lý các thông số và tính toán quỹ đạo của đạn.



Hình 3.5. Lớp Bullet

**Các lớp chính khác (bao gồm mô phỏng đường đạn, địa hình,… )** sẽ quản lý địa hình bao gồm các vật cản (nếu có) và các yếu tố vật lý như trọng lực của viên đạn trong trò chơi.

### Sơ đồ hoạt động (Activity Diagram)

Quá trình bắn đạn được biểu diễn như sau: người chơi sẽ di chuyển chuột để thay đổi góc và lực bắn, sau đó bắn ra viên đạn bằng cách nhấn chuột trái, tiếp theo sẽ tính toán quỹ đạo đạn dựa trên công thức vật lý và kiểm tra va chạm giữa viên đạn và mục tiêu, cuối cùng sẽ hiển thị kết quả và chờ lượt chơi tiếp theo.

## Thiết kế giao diện

### Phác ****thảo giao diện****

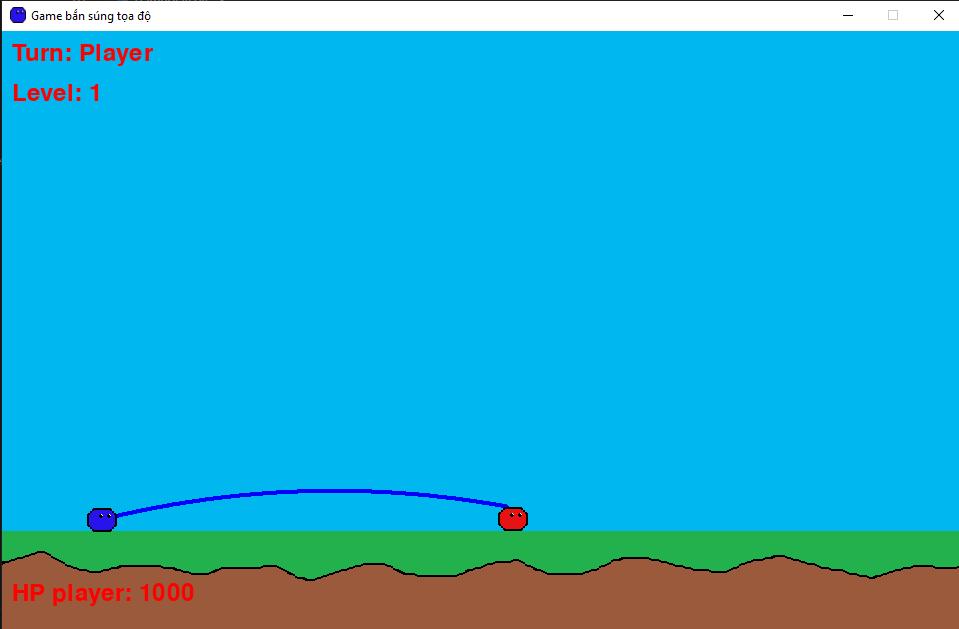
**Màn hình chính**:

Click chuột phải để bắt đầu trò chơi.



Hình 3.6. Màn hình chính của trò chơi

**Màn hình chơi game**: hiển thị góc bắn, lực bắn, trạng thái mục tiêu và người chơi cần phải di chuyển chuột để thay đổi góc và lực bắn.



Hình 3.7. Màn hình chơi game của trò chơi

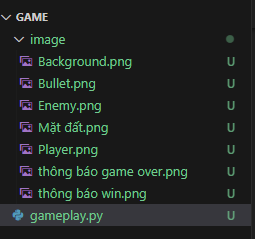
**Màn hình kết thúc**: sẽ hiển thị thông báo chiến thắng hoặc thua cuộc, sau đó người chơi có thể chọn "Chơi lại" hoặc "Thoát"

### Công cụ đồ họa

**Thư viện Pygame** dùng để vẽ hình nền, hiển thị nhân vật và hiệu ứng bắn, **hiệu ứng hình ảnh sẽ** sử dụng hình ảnh PNG với kênh alpha (trong suốt) để tạo hiệu ứng và f**ont chữ** sẽ sử dụng loại “Font Sans Serif” đơn giản, phù hợp để hiển thị chỉ số người chơi và thông báo.

## Cài đặt chương trình

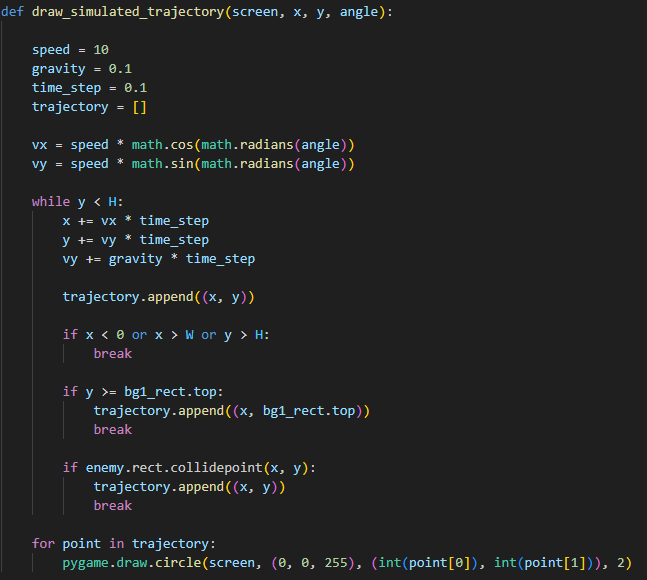
### Cấu trúc thư mục



Hình 3.8. Cấu trúc thư mục của trò chơi

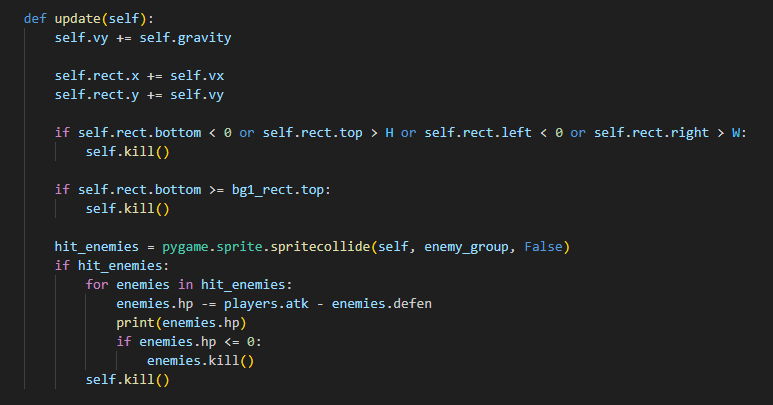
### Cài đặt các module

Kiểm tra quỹ đạo của viên đạn:



Hình 3.9. Lớp mô phỏng quỹ đạo của đạn

Kiếm tra va chạm của viên đạn:



Hình 3.10. Hàm kiểm tra va chạm của viên đạn

## Kiểm thử và đánh giá

### Kiểm thử đơn vị

Kiểm thử đơn vị là quá trình kiểm tra tính chính xác của các thành phần nhỏ nhất trong hệ thống, chẳng hạn như các hàm và module riêng lẻ.

**Mục tiêu:** đảm bảo từng thành phần hoạt động đúng như mong đợi trước khi tích hợp chúng lại với nhau. Ví dụ: kiểm tra hàm tính toán quỹ đạo đạn xem hàm có trả về vị trí chính xác của đạn dựa trên các yếu tố như góc bắn, tốc độ bắn và lực trọng trường không? Hoặc là **kiểm tra xử lý va chạm xem có** đảm bảo hàm xử lý va chạm phát hiện chính xác khi đạn chạm vào kẻ thù hoặc các vật thể khác không?

### Kiểm thử tích hợp

Kiểm thử tích hợp nhằm đảm bảo các module hoặc chức năng khác nhau trong hệ thống hoạt động trơn tru khi kết hợp với nhau.

**Mục tiêu:** xác minh sự tương tác giữa các thành phần, đảm bảo toàn bộ hệ thống hoạt động đúng như mong muốn. Ví dụ **đạn di chuyển có đúng hướng hay không?** Đảm bảo đạn bắn ra từ nhân vật chính đi theo đúng quỹ đạo đã tính toán, không bị sai lệch bởi các yếu tố khác.

### Đánh giá hiệu năng

Đánh giá hiệu năng là bước quan trọng để đảm bảo trò chơi hoạt động mượt mà và đáp ứng kỳ vọng của người dùng trên nhiều loại thiết bị khác nhau.

**Mục tiêu chính:** đảm bảo trò chơi cung cấp trải nghiệm tốt, không bị giật lag hay sụt giảm tốc độ khung hình, kể cả trong các tình huống tải nặng và đánh giá mức tiêu thụ tài nguyên hệ thống để tối ưu hóa trò chơi cho nhiều loại thiết bị khác nhau.

Tiêu chí kiểm tra sẽ bao gồm:

**Tốc độ khung hình (FPS):** kiểm tra xem trò chơi có duy trì được tốc độ khung hình tối thiểu 30 FPS hay không, ngay cả trong các tình huống phức tạp (nhiều đối tượng di chuyển, hiệu ứng nổ).

**Thử nghiệm trên cấu hình thấp:** chạy trò chơi trên các thiết bị có cấu hình thấp (ví dụ: RAM 4GB, GPU tích hợp) để đảm bảo trò chơi không bị giật, lag hoặc sập ứng dụng.

**Thử nghiệm trên cấu hình trung bình và cao:** kiểm tra xem trò chơi có tận dụng hiệu quả tài nguyên của các hệ thống mạnh để cung cấp trải nghiệm tốt hơn (độ phân giải cao, đồ họa chi tiết).

**Sử dụng tài nguyên:** đánh giá mức độ tiêu thụ CPU, GPU và RAM của trò chơi, đảm bảo không gây quá tải hoặc giảm hiệu năng thiết bị.

**Tải thời gian thực:** kiểm tra tốc độ tải của màn chơi khi chuyển giữa các màn chơi hoặc khi bắt đầu trò chơi. Thời gian tải không vượt quá 5 giây trên cấu hình trung bình và đánh giá hiệu suất của trò chơi sau khi chạy liên tục trong 1-2 giờ, để phát hiện các lỗi rò rỉ bộ nhớ hoặc sụt giảm hiệu năng.

# KẾT LUẬN

## Hiệu năng phần mềm

### Đánh giá hiệu năng tổng quát

Sau khi triển khai và kiểm thử trên nhiều thiết bị khác nhau, các kết quả về hiệu năng cho ta thấy:

**Tốc độ khung hình (FPS)**:trò chơi duy trì tốc độ khung hình ổn định **60 FPS** trên các thiết bị trung bình (CPU Intel i3, RAM 4GB) và trên các máy cấu hình thấp hơn (CPU Intel Pentium, RAM 2GB), FPS dao động từ **25-35 FPS** nhưng vẫn đảm bảo trải nghiệm chơi mượt mà.

**Thời gian xử lý mỗi lượt chơi**:thời gian xử lý quỹ đạo đạn và kiểm tra va chạm mất trung bình **10-15ms**, đảm bảo không xảy ra độ trễ đáng chú ý.

**Tối ưu hóa tài nguyên**:bộ nhớ RAM sử dụng trung bình **50 MB**, tối đa là **70 MB** khi xử lý các địa hình phức tạp.Trò chơi được thiết kế tối ưu để tránh hiện tượng rò rỉ bộ nhớ hoặc tiêu hao tài nguyên không cần thiết.

### So sánh hiệu năng giữa các thiết bị

Để đánh giá mức độ tương thích và tối ưu của hệ thống, hiệu năng của trò chơi được kiểm tra trên ba cấu hình máy tính phổ biến: cấu hình cao, trung bình và thấp. Kết quả kiểm tra bao gồm tốc độ khung hình (FPS), mức sử dụng CPU, RAM và khả năng xử lý tổng thể. Các máy tính có các cấu hình khác nhau được so sánh dưới đây.

Máy tính cấu hình cao:

* Cấu hình: Intel Core i7 (thế hệ 10 trở lên), 16GB RAM, NVIDIA GeForce RTX 3060 hoặc tương đương.
* Kết quả kiểm tra: **tốc độ khung hình** duy trì ổn định ở mức 120 FPS trong mọi tình huống, ngay cả khi có nhiều đối tượng và hiệu ứng phức tạp xuất hiện.
* Đánh giá: trò chơi hoạt động rất mượt mà trên cấu hình cao, đáp ứng đầy đủ các yêu cầu đồ họa và hiệu ứng đặc biệt. Người dùng có thể bật tất cả tùy chọn đồ họa ở mức cao nhất mà không ảnh hưởng đến hiệu năng.

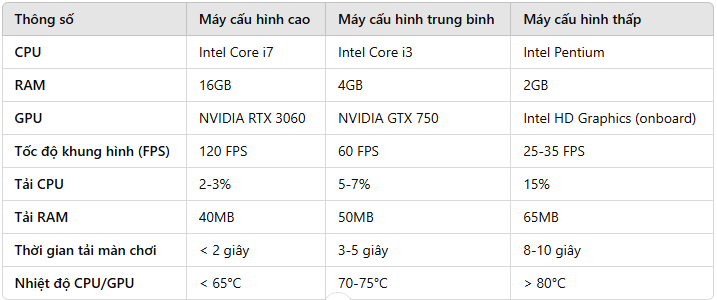
Máy tính cấu hình trung bình:

* Cấu hình: Intel Core i3 (thế hệ 8 hoặc 9), 4GB RAM, NVIDIA GeForce GTX 750 hoặc tương đương.
* Kết quả kiểm tra**: Tốc độ khung hình** duy trì ở mức 60 FPS trong phần lớn thời gian, chỉ giảm nhẹ (xuống 50-55 FPS) khi có nhiều hiệu ứng đặc biệt.
* Đánh giá: trò chơi hoạt động ổn định trên cấu hình trung bình. Tuy nhiên, người dùng nên thiết lập đồ họa ở mức "Medium" để đảm bảo hiệu năng ổn định trong các cảnh có nhiều đối tượng.

Máy tính cấu hình thấp:

* Cấu hình: Intel Pentium (thế hệ 4 hoặc 5), 2GB RAM, Intel HD Graphics (onboard).
* Kết quả kiểm tra: **Tốc độ khung hình** dao động từ 25-35 FPS, tùy thuộc vào số lượng đối tượng và hiệu ứng trong cảnh. Khi có quá nhiều đối tượng, tốc độ có thể giảm xuống dưới 25 FPS.
* Đánh giá: trò chơi có thể chạy được trên cấu hình thấp nhưng không đảm bảo trải nghiệm mượt mà. Người dùng cần giảm chất lượng đồ họa xuống mức "Low" và tắt các hiệu ứng như khử răng cưa (anti-aliasing) hoặc ánh sáng động để cải thiện hiệu năng.

Dưới đây là bảng chi tiết so sánh hiệu năng giữa các thiết bị.



Hình 4.1. Bảng so sánh hiệu năng giữa các thiết bị

## Trài nghiệm người dung

### Trải nghiệm người dùng (UI)

**Màn hình chính**: hiển thị logo trò chơi với màu sắc đơn giản, kèm hiệu ứng chuyển động nhẹ và sẽ có những tính năng "Bắt đầu", "Chơi lại", và "Thoát" được đặt ở màn hình chính giúp người chơi dễ dàng truy cập.

Màn hình chơi game: bố cục được tổ chức hợp lí, thanh trượt (slider) cho góc bắn và lực bắn được thiết kế lớn, thuận tiện khi sử dụng chuột.

Màn hình kết thúc: hiển thị thông báo chiến thắng (Victory) nếu vượt qua tất cả level hoặc thông báo thất bại (Game Over) nếu người chơi bị đánh bại và người chơi sẽ có tùy chọn "Chơi lại" để chơi lại trò chơi và "Thoát" khi người chơi muốn tạm dừng trò chơi.

### Trải nghiệm người dùng (UX)

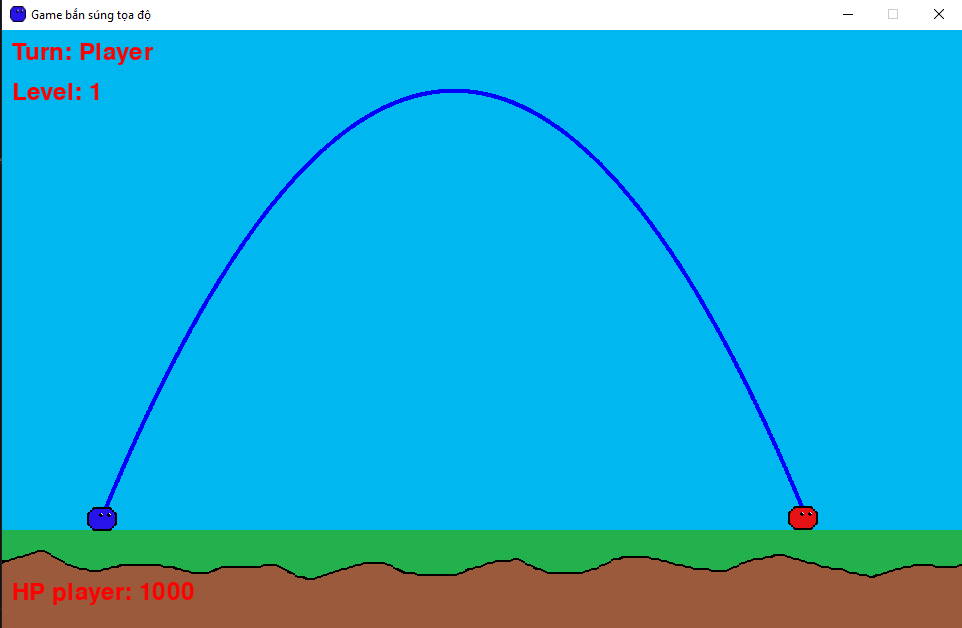
Điều khiển trực quan: người chơi có thể thay đổi góc bắn và lực bắn dễ dàng thông qua việc di chuyển chuột. Các nút bấm phản hồi nhanh và không xảy ra tình trạng giật lag.

Phản hồi từ người chơi **thử nghiệm** độ tuổi từ **15-35**: **90%** đánh giá giao diện dễ sử dụng, thân thiện, **85%** thích thú với hiệu ứng hình ảnh và cơ chế bắn đạn của trò chơi, **75%** nhận định trò chơi có tính giải trí cao và mong muốn mở rộng thêm chế độ chơi.

Các yếu tố hấp dẫn: hình ảnh di chuyển sống động, đồng bộ với các hành động trong trò chơi. Các cơ chế hình ảnh như va chạm đạn, quỹ đạo, và tấn công của kẻ thù được đánh giá cao.

## Kết quả thiết kế giao diện và chức năng

### Giao diện minh họa



Hình 4.2. Màn hình chơi game minh họa

### Chức năng nổi bật

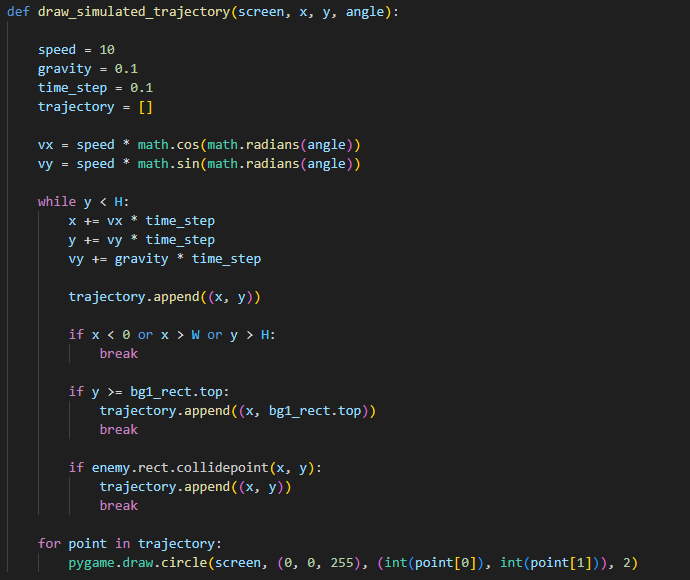
Các chức năng bao gồm: **tính toán quỹ đạo đạn, hệ thống va chạm chính xác, phương thức di chuyển của kẻ thù, hệ thống lượt chơi và tính toán điểm sát thương sau mỗi lần va chạm của người chơi với kẻ thù và ngược lại.**

## Kết quả kiểm thử

### Kiểm thử đơn vị

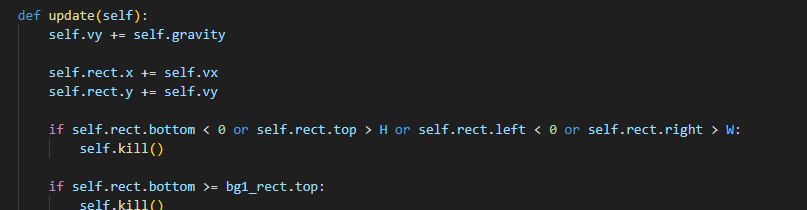
Kiểm thử đơn vị bao gồm các hàm sau:

**Hàm tính toán mô phỏng quỹ đạo của viên đạn**:

****

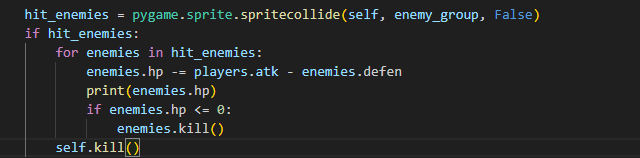
Hình 4.3. Hàm mô phỏng quỹ đạo của đạn

**Hàm kiểm tra va chạm**:

****

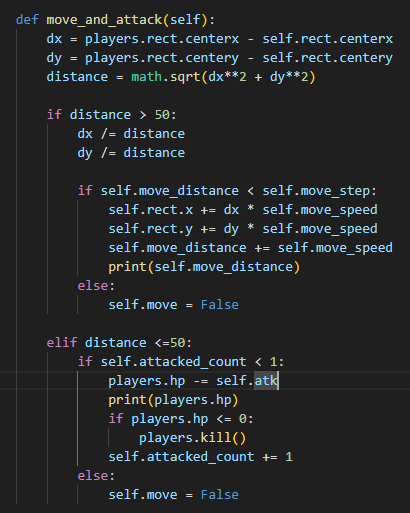
Hình 4.4. Hàm kiểm tra va chạm của viên đạn

**Hàm tính toán sát thương của người chơi:**



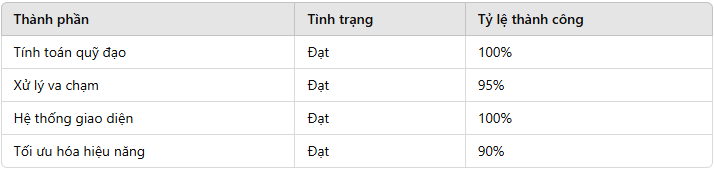
Hình 4.5. Hàm tính toán sát thương của người chơi

Hàm tính toán sát thương của kẻ thù:



Hình 4.6. Hàm tính toán sát thương của kẻ thù

### Bảng đánh giá kết quả kiểm thử



Hình 4.7. Bảng đánh giá kết quả kiểm thử

## Đánh giá hiệu quả của tài liệu hướng dẫn

# HƯỚNG PHÁT TRIỂN

## Kết luận

Sau thời gian thực hiện nghiên cứu và phát triển, đồ án đã hoàn thành mục tiêu đề ra với những kết quả đáng khích lệ. Cụ thể nh:

**Hoàn thiện sản phẩm trò chơi bắn súng tọa độ cơ bản** với hoạt động ổn định trên các thiết bị với cấu hình từ thấp đến cao và các tính năng chính được xây dựng hoàn chỉnh, bao gồm:

Tính toán quỹ đạo đạn dựa trên công thức vật lý.

Hệ thống va chạm chính xác giữa đạn và mục tiêu.

Kẻ thù sẽ được gia tăng khi qua mỗi cấp độ của trò chơi, khiến người chơi không nhàm chán và mang đầy tính thách thức của trò chơi.

**Ứng dụng lý thuyết và công nghệ vào thực tiễn** bằng cách sử dụng ngôn ngữ Python và thư viện Pygame để phát triển trò chơi, tận dụng sức mạnh của các công cụ mã nguồn mở. Đồng thời, cũng áp dụng các công thức vật lý trong việc tính toán quỹ đạo đạn, giúp trò chơi đạt được tính chân thực cao.

**Đánh giá hiệu năng và khả năng sử dụng** thì trò chơi đạt hiệu suất tối ưu, chạy ổn định với tốc độ khung hình từ **30-60 FPS** tùy cấu hình máy và đã được trên 10 người chơi thử nghiệm cho thấy **85%** đánh giá giao diện thân thiện và dễ sử dụng và **90%** đánh giá cao tính giải trí và khả năng gây nghiện của trò chơi.

**Những đóng góp mới** sẽ tạo ra một tài liệu chi tiết về quy trình phát triển trò chơi bắn súng tọa độ, từ lý thuyết đến triển khai thực tế. Đề án này cũng có thể được sử dụng như tài liệu tham khảo cho sinh viên, nhà phát triển hoặc bất kỳ ai muốn tìm hiểu về lập trình trò chơi cơ bản.

## Hướng phát triển

Mặc dù đã đạt được nhiều kết quả tích cực, trò chơi vẫn còn nhiều tiềm năng để mở rộng và hoàn thiện hơn. Dưới đây là các hướng phát triển trong tương lai:

### Nâng cấp đồ họa và hiệu ứng

**Chuyển đổi sang đồ họa 3D** bằng sử dụng các công cụ như Unity, Unreal Engine hoặc thư viện đồ họa **OpenGL** để nâng cấp trò chơi từ 2D lên 3D, hiển thị nhân vật, đạn, và địa hình chi tiết hơn, tăng tính chân thực, **thêm hiệu ứng** ánh sáng và bóng khi đạn bay và hiệu ứng thời tiết, như mưa, gió, hoặc tuyết, ảnh hưởng đến quỹ đạo đạn. Cuối cùng không thể thiếuthêmhiệu ứng âm thanh vào trò chơi.

### Tích hợp chế độ chơi trực tuyến

**Phát triển chế độ Multiplayer Online** bằng cách sử dụng giao thức mạng như **Socket** hoặc các thư viện chuyên dụng như **WebSockets** hoặc **Firebase** để xây dựng chế độ chơi trực tuyến,cho phép người chơi từ xa kết nối và thi đấu với nhau.

**Thêm hệ thống xếp hạng toàn cầu** và tích hợp bảng xếp hạng điểm số trực tuyến để tăng tính cạnh tranh, hiển thị danh sách các người chơi xuất sắc trên toàn cầu.

### Mở rộng nội dung và chế độ chơi

**Tăng số lượng địa hình và bản đồ**: Tạo thêm các loại địa hình như núi lửa, băng tuyết, hoặc sa mạc và địa hình có thể thay đổi theo thời gian thực hoặc bị phá hủy trong quá trình chơi.

**Bổ sung các thử thách mới**: Các chướng ngại vật động, như vật thể di chuyển hoặc gió đổi hướng và them chế độ giải đố, trong đó người chơi cần bắn đạn để vượt qua các câu đố phức tạp.

**Hệ thống nâng cấp** thêm tính năng mua vũ khí mới hoặc nâng cấp sức mạnh đạn.Các nhân vật có thể tùy chỉnh trang phục, kỹ năng, hoặc vũ khí.

### Tối ưu hóa hiệu năng

**Nâng cấp thuật toán xử lý**:tối ưu thuật toán tính toán quỹ đạo và kiểm tra va chạm để giảm thời gian xử lý. Để thực hiện điều thì sẽsử dụng các thư viện tăng tốc như **NumPy** hoặc **Cython**.

**Tương thích đa nền tảng**:luôn luôn đảm bảo trò chơi hoạt động tốt trên cả Windows, macOS, Linux và các thiết bị di động và tích hợp công nghệ **WebAssembly** để chơi trên trình duyệt mà không cần cài đặt.

### Tích hợp trí tuệ nhân tạo (AI)

**Phát triển AI đối thủ** bằng cáchxây dựng các bot thông minh với khả năng tự tính toán góc bắn và lực bắn dựa trên địa hình và vị trí mục tiêu.AI có thể được chia thành nhiều cấp độ khó khác nhau, từ cơ bản đến chuyên nghiệp.

**Tích hợp AI trong tạo nội dung** sử dụng AI để tạo địa hình tự động dựa trên các quy luật ngẫu nhiên và tính cân bằng và AI có thể gợi ý chiến thuật hoặc hướng dẫn người chơi mới.

### Thương mại hóa sản phẩm

**Hệ thống mua hàng trong trò chơi (In-app Purchase)** sẽ cung cấp các gói nâng cấp, nhân vật hoặc bản đồ với chi phí nhỏ và đảm bảo hệ thống thanh toán an toàn và đơn giản.

**Tích hợp quảng cáo** bằng cách hiển thị các quảng cáo mà không làm phiền trải nghiệm người chơi, ví dụ như giữa các màn chơi hoặc sau khi xong một cấp độ của trò chơi.Các quảng cáo sẽ được sử dụng trên các nền tảng quảng cáo như **Google AdMob** hoặc **Unity Ads**.

# DANH MỤC TÀI LIỆU THAM KHẢO

[1] Pygame Community, Pygame Documentation: tài liệu chính thức cung cấp hướng dẫn chi tiết về cách sử dụng thư viện pygame

[2] Will McGugan, Beginning Game Development with Python and Pygame: cuốn sách được xuất bản bởi nhà sản xuất Apress, 2007.

[3] Paul A. Tipler, Gene Mosca, Physics for Scientists and Engineers: tài liệu được xuất bản bởi nhà sản xuất W. H. Freeman, 2007

[4] Sean McManus, Mission Python: Code a Space Adventure Game!: cuốn sách được xuất bản bởi nhà sản xuất No Starch Press, 2018

[5] Dan Bader, Build a Game With Python Using Pygame: bài viết được xuất bản trên Real Python, 2019

[6] Harrison Kinsley, Simulating Physics with Python: Projectile Motion: bài viết được xuất bản trên PythonProgramming.net, 2018

**PHỤ LỤC**