|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC ĐIỆN LỰC**  **KHOA CÔNG NGHỆ THÔNG TIN**  **C:\Users\VCIS\Desktop\logo-epu-inkythuatso-14-15-47-22.jpg**  **BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**  **ĐỒ HỌA MÁY TÍNH**  **XÂY DỰNG GAME FRUIT NINJA**   |  |  | | --- | --- | | **Giảng viên hướng dẫn** | **: ThS. NGUYỄN ĐÌNH THÁI** | | **Sinh viên thực hiện** | **: TRẦN VIỆT BÁCH**  **LÊ ANH DUY**  **LÊ VĂN TUẤN** | | **Ngành** | **: CÔNG NGHỆ THÔNG TIN** | | **Chuyên ngành** | **: CÔNG NGHỆ PHẦN MỀM** | | **Lớp** | **: D16CNPM1** | | **Khóa** | **: 2021** |   ***Hà Nội, tháng 05 năm 2024*** |

**PHIẾU CHẤM ĐIỂM**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ và tên** | **Nội dung thực hiện** | **Chữ ký** | **Điểm** |
| Lê Văn Tuấn  MSV: 21810310056 |  |  |  |
| Trần Việt Bách  MSV: 21810310039 |  |  |  |
| Lê Anh Duy  MSV: 21810310030 |  |  |  |

**Giảng viên chấm:**

|  |  |  |
| --- | --- | --- |
| **Giảng viên** | **Chữ ký** | **Ghi chú** |
| Giảng viên chấm 1: |  |  |
| Giảng viên chấm 2: |  |  |

LỜI CẢM ƠN

Trong suốt học kỳ vừa qua, chúng em muốn bày tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Đình Thái, giảng viên môn Đồ họa máy tính, vì sự hỗ trợ và sự động viên không ngừng của thầy trong quá trình chúng em nghiên cứu và thực hiện đề tài "Xây dựng game bắt ruồi 2 người chơi".

Dù thời gian và kinh nghiệm của chúng em còn hạn chế, và báo cáo chuyên đề của chúng em chưa thể hoàn thiện như mong đợi, nên chúng em rất mong được nhận được sự phản hồi và đóng góp xây dựng từ thầy cùng các bạn để chúng em có thể rút kinh nghiệm và cải thiện hơn cho những dự án sau này.

Chúng em xin gửi lời cảm ơn chân thành đến thầy và mọi người!

MỤC LỤC

Trang

[MỞ ĐẦU 1](#_Toc164688652)

[Chương 1 - tổng quan 2](#_Toc164688653)

[1.1. Lý do chọn đề tài 2](#_Toc164688654)

[1.2. Mô tả đề tài 2](#_Toc164688655)

[1.2.1. OpenGL là gì? 2](#_Toc164688656)

[1.2.2. Tổng quan về game 3](#_Toc164688657)

[Chương 2 - ứng dụng đồ họa xây dụng game 5](#_Toc164688658)

[2.1. Ý tưởng 5](#_Toc164688659)

[2.2. Thực hiện chương trình 6](#_Toc164688660)

[2.2.1. Tổng quan về chương trình 6](#_Toc164688661)

[2.2.2. Các chức năng chính của trò chơi 6](#_Toc164688662)

[2.2.3. Vấn đề và hướng giải quyết 6](#_Toc164688663)

[2.3. Thiết kế giao diện 6](#_Toc164688664)

[2.3.1. Giao diện chính của trò chơi khi mới bắt đầu 6](#_Toc164688665)

[2.3.2. Giao diện khi người dùng thứ nhất sử dụng phím Space 6](#_Toc164688666)

[2.3.3. Giao diện khi người dùng thứ hai sử dụng phím Enter 7](#_Toc164688667)

[2.3.4. Giao diện khi người chơi bắt được ruồi và ghi điểm 7](#_Toc164688668)

[2.3.5. Giao diện khi ếch nhảy được lên viên gạch trên không trung 8](#_Toc164688669)

[2.3.6. Giao diện khi kết thúc trò chơi gameOver 8](#_Toc164688670)

[2.4. Code chương trình 8](#_Toc164688671)

[KẾT LUẬN 10](#_Toc164688672)

[TÀI LIỆU THAM KHẢO 11](#_Toc164688673)

[PHỤ LỤC 12](#_Toc164688674)

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

MỞ ĐẦU

Đồ họa máy tính góp phần quan trọng làm cho giao tiếp của con người và máy tính trở nên thân thiện gần gũi hơn. Trong lĩnh vực kỹ thuật máy tính, đồ họa máy tính càng ngày càng phát triển mạnh mẽ. Đồ họa máy tính là một lĩnh vực của khoa học máy tính nghiên cứu về ở toán học, các thuật toán cũng như các kĩ thuật để cho phép tạo, hiển thị và điều khiển hình ảnh trên màn hình máy tính. Đồ họa máy tính có liên quan ít nhiều đến một số lĩnh vực như đại số, hình học giải tích, hình học họa hình, quang học,...và kĩ thuật máy tính, đặc biệt là chế tạo phần cứng (các loại màn hình, các thiết bị xuất, nhập, các vỉ mạch đồ họa...).

Là sinh viên khoa Công Nghệ Thông Tin trường Đại học Điện Lực. Em cũng được tiếp xúc với môn học đồ họa máy tính. Với những kiến thức chúng em đã được học và được sự hướng dẫn của thầy, chúng em đã thực hiện đề tài “Xây dựng game bắt ruồi 2 người”.

# giới thiệu về đồ họa máy tính

**1.1. Lịch sử phát triển**

- Graphics những năm 1950-1960:

+ Năm 1959 Thiết bị đồ hoạ đầu tiên là màn hình xuất hiện tại Đức.

+ Năm 1960 - SAGE (Semi-Automatic Ground Environment System) xuất hiện bút sáng thao tác với màn hình.

+ Năm 1960 William Fetter nhà khoa học người Mỹ, ông đang nghiên cứu xây dựng mô hình buồng lái máy bay cho hãng Boeing của Mỹ. Ông đã dựa trên hình ảnh 3 chiều của mô hình ngườiphi công trong buồng lái của máy bay để xây dựng nên một mô hình tối ưu cho buồng lái máy bay. Phương pháp này cho phép các nhà thiết kế quan sát một cách trực quan vị trí của người lái trong khoang. Ông đặt tên cho phương pháp này là đồ hoạ máy tính (Computer Graphics).

+ Màn hình là thiết bị thông dụng nhất trong hệ đồ hoạ, các thao tác của hầu hết các màn hình đều dựa trên thiết kế ống tia âm cực CRT (Cathode ray tube). Khi đó giá để làm tươi màn hình là rất cao, máy tính xử lý chậm, đắt và không chắc chắn (không đáng tin cậy).

- Graphics: 1960-1970

+ Năm 1963 Ivan Sutherland (hội nghị Fall Joint Computer - lần đầu tiên có khả năng tạo mới, hiển thị và thay đổi được thực hiện trong thời gian thực trên màn CRT).

+ Hệ thống này được dùng để thiết kế mạch điện: CRT, LightPen (bút sáng), computer (chứa chương trình xử lý thông tin). Người sử dụng có thể vẽ mạch điện trực tiếp lên màn hình thông qua bút sáng.

- Graphics:1970-1980

+ Raster Graphics (đồ hoạ điểm). Bắt đầu chuẩn đồ hoạ ví dụ như: GKS (Graphics Kernel System): European effort (kết quả của châu âu), Becomes ISO 2D standard.

- Graphics: 1980-1990

+ Mục đích đặc biệt về phần cứng, thiết bị hình học đồ hoạ Silicon. Xuất hiện các chuẩn công nghiệp: PHIGS (Programmers Hierarchical Interactive Graphics Standard) xác định các phương pháp chuẩn cho các mô hình thời gian thực và lập trình hướng đối tượng.

**1.2. Khái niệm đồ họa máy tính**

- Đồ họa máy tính là một ngành khoa học Tin học chuyên nghiên cứu về các phương pháp và kỹ thuật để có thể mô tả và thao tác trên các đối tượng của thế giới thực bằng máy tính

- Về bản chất: đó là một quá trình xây dựng và phát triển các công cụ trên cả hai lĩnh vực phần cứng và phần mềm hổ trợ cho các lập trình viên thiết kế các chương trình có khả năng đồ họa cao.

- Với việc mô tả dữ liệu thông qua các hình ảnh và màu sắc đa dạng của nó, các chương trình đồ họa thường thu hút người sử dụng bởi tính thân thiện, dể dùng, ... kích thích khả năng sáng tạo và nâng cao năng suất làm việc.

- Các khía cạnh chính của đồ họa máy tính bao gồm:

+ Mô hình hóa (Modeling): Quá trình tạo ra các mô hình số học của đối tượng 3D, sử dụng các phương pháp toán học và hình học để biểu diễn hình dạng và cấu trúc của chúng.

+ Kết xuất (Rendering): Quá trình chuyển đổi mô hình 3D thành hình ảnh 2D bằng cách sử dụng các thuật toán và kỹ thuật khác nhau để tính toán ánh sáng, bóng đổ, màu sắc, và các hiệu ứng hình ảnh khác.

+ Hoạt hình (Animation): Quá trình tạo ra chuyển động và biến đổi của các đối tượng theo thời gian, bao gồm việc lập trình các chuyển động, biến dạng, và hiệu ứng tương tác.

+ Xử lý hình ảnh (Image Processing): Liên quan đến việc chỉnh sửa, cải thiện và phân tích các hình ảnh số, bao gồm các kỹ thuật như lọc ảnh, nén ảnh, và nhận dạng đối tượng.

+ Tương tác người-máy (Human-Computer Interaction - HCI): Nghiên cứu cách người dùng tương tác với các hệ thống đồ họa máy tính, bao gồm thiết kế giao diện người dùng, thiết bị nhập liệu, và trải nghiệm người dùng.

* Ứng dụng của đồ họa máy tính rất rộng rãi, bao gồm:

+ Trò chơi điện tử: Sử dụng đồ họa 3D để tạo ra các thế giới ảo sống động và các nhân vật chuyển động thực tế.

+ Phim ảnh và hoạt hình: Sử dụng kỹ xảo đồ họa để tạo ra các hiệu ứng đặc biệt và các cảnh quay mà không thể thực hiện được trong thực tế.

+ Thiết kế công nghiệp và kiến trúc: Sử dụng các mô hình 3D để thiết kế và trình bày các sản phẩm, công trình xây dựng trước khi chúng được thực hiện.

+ Y học: Sử dụng hình ảnh 3D để mô phỏng và phân tích cơ thể con người, hỗ trợ trong việc chẩn đoán và điều trị bệnh.

+ Khoa học và kỹ thuật: Sử dụng đồ họa máy tính để trực quan hóa dữ liệu phức tạp và mô phỏng các hiện tượng vật lý.

Tóm lại, đồ họa máy tính là một lĩnh vực quan trọng và không ngừng phát triển, góp phần làm cho cuộc sống hiện đại trở nên phong phú và đa dạng hơn thông qua các ứng dụng thực tế và nghệ thuật số.

## Lý do chọn đề tài

## Đối với những bạn mới bắt đầu học lập trình căn bản với C/C++ và muốn tiếp cận việc phát triển game, sử dụng Pygame là một lựa chọn rất hợp lý. Pygame là một thư viện phát triển game đơn giản, dễ học và sử dụng, hỗ trợ lập trình bằng Python, ngôn ngữ lập trình phổ biến và dễ tiếp cận. Pygame cho phép bạn phát triển game và ứng dụng đồ họa trên nhiều nền tảng khác nhau, bao gồm Windows, macOS, và Linux.

## Với Pygame, bạn có thể tạo ra và điều khiển các đối tượng đồ họa một cách dễ dàng, từ các hình cơ bản như đường thẳng và hình chữ nhật đến các đối tượng phức tạp hơn như các hình ảnh động. Pygame cung cấp nhiều công cụ mạnh mẽ cho việc xử lý đồ họa, âm thanh và tương tác người dùng, giúp bạn tạo ra các trải nghiệm game đa dạng và chất lượng cao.

## Dần dần nắm vững kiến thức và kỹ năng sử dụng Pygame, bạn có thể tiếp tục phát triển và xây dựng những game và ứng dụng đồ họa phức tạp hơn, đồng thời hiểu rõ hơn về cách thức hoạt động của các thư viện game và công nghệ đồ họa cơ bản. Với Pygame, bạn cũng có thể dễ dàng tích hợp các tính năng như vật lý đơn giản, âm thanh, và tương tác người dùng, giúp tạo ra các sản phẩm game chất lượng cao như Fruit Ninja.

## Trong đề tài này, chúng tôi sẽ tập trung vào việc phát triển một phiên bản đơn giản của game Fruit Ninja bằng Pygame, giúp người học tiếp cận và hiểu rõ hơn về quá trình phát triển game từ ý tưởng đến thực hiện. Qua đó, bạn sẽ học được cách làm việc với Pygame, từ việc tạo dựng các đối tượng game, lập trình logic game, đến việc xử lý các tương tác và hiệu ứng đồ họa, âm thanh.

## Bằng cách lựa chọn Pygame làm công cụ phát triển, chúng tôi mong muốn cung cấp một nền tảng học tập hiệu quả và thân thiện cho những bạn mới bắt đầu, đồng thời khơi dậy sự hứng thú và đam mê với lập trình game và đồ họa máy tính.

## Mô tả đề tài

### Pygame là gì?

### Pygame là một thư viện mạnh mẽ và phổ biến, được sử dụng rộng rãi để phát triển các ứng dụng và trò chơi đồ họa 2D. Pygame là một bộ thư viện đa phương tiện cho Python, cung cấp nhiều công cụ hữu ích để xử lý đồ họa, âm thanh và tương tác người dùng, giúp cho việc phát triển game trở nên trực quan và dễ tiếp cận.

### Pygame ra đời vào năm 2000 bởi Pete Shinners, với mục tiêu ban đầu là giúp đỡ các nhà phát triển indie và những người mới bắt đầu học lập trình game. Qua thời gian, Pygame đã phát triển thành một công cụ toàn diện và được sử dụng bởi cả các nhà phát triển lớn lẫn nhỏ trong ngành công nghiệp game.

### Pygame cung cấp nhiều công cụ và tính năng hữu ích cho việc phát triển game, bao gồm:

### Hỗ Trợ Đa Nền Tảng: Pygame cho phép xuất bản game trên nhiều nền tảng khác nhau như Windows, macOS và Linux mà không cần phải thay đổi mã nguồn nhiều.

### Giao Diện Người Dùng Thân Thiện: Pygame có cấu trúc đơn giản và dễ hiểu, giúp người dùng dễ dàng tạo và quản lý các đối tượng game, xử lý sự kiện và âm thanh.

### Ngôn Ngữ Lập Trình: Pygame sử dụng Python làm ngôn ngữ lập trình chính, một ngôn ngữ mạnh mẽ và dễ học, đặc biệt phù hợp cho những người mới bắt đầu lập trình game.

### Công Cụ Đồ Họa Mạnh Mẽ: Pygame hỗ trợ các công cụ và tính năng đồ họa cơ bản, bao gồm xử lý hình ảnh, vẽ các hình dạng cơ bản và quản lý sprite. Điều này giúp các nhà phát triển tạo ra các game với đồ họa đơn giản và hiệu quả.

### Hệ Thống Âm Thanh: Pygame tích hợp sẵn một hệ thống âm thanh, giúp tạo ra các hiệu ứng âm thanh phong phú và chân thực trong game.

### Cộng Đồng Và Tài Nguyên Phong Phú: Pygame có một cộng đồng lớn và năng động, với nhiều tài liệu, hướng dẫn và ví dụ có sẵn, giúp người dùng dễ dàng học hỏi và phát triển kỹ năng.

### Với những tính năng và ưu điểm trên, Pygame đã trở thành một công cụ không thể thiếu cho những ai muốn bắt đầu hoặc tiếp tục sự nghiệp trong lĩnh vực phát triển game. Pygame không chỉ giúp bạn hiện thực hóa các ý tưởng game của mình một cách dễ dàng mà còn mở ra nhiều cơ hội để bạn phát triển các ứng dụng đồ họa chất lượng cao.

### Giới thiệu Thuật Toán Cohen-Sutherland

Thuật toán Cohen-Sutherland là một thuật toán được sử dụng để cắt một đoạn thẳng bởi một cửa sổ hình chữ nhật (viewport) trong không gian hai chiều. Được phát triển bởi Danny Cohen và Ivan Sutherland vào năm 1967, thuật toán này thường được áp dụng trong đồ họa máy tính để tìm ra phần của một đoạn thẳng nằm trong một cửa sổ hình chữ nhật, giúp tối ưu hóa việc vẽ đồ họa.

Dưới đây là một số khái niệm cơ bản trong thuật toán Cohen-Sutherland:

Đoạn thẳng: Một đoạn thẳng là một phần của một đường thẳng nối hai điểm cuối có thể được xác định bằng cặp tọa độ (x, y).

Cửa sổ hình chữ nhật (viewport): Là khu vực trong không gian hai chiều mà chúng ta muốn hiển thị đoạn thẳng. Cửa sổ này được xác định bởi hai điểm, thường là điểm trên cùng bên trái (xmin, ymin) và điểm dưới cùng bên phải (xmax, ymax).

Mã phần tử (outcode): Mỗi điểm trong không gian hai chiều được gán một mã phần tử, đại diện cho vị trí của điểm đó so với cửa sổ hình chữ nhật. Mã phần tử này thường được biểu diễn dưới dạng một chuỗi bit, trong đó mỗi bit đại diện cho một vị trí tương ứng: trên, dưới, trái, phải.

Thuật toán Cohen-Sutherland hoạt động như sau:

* Gắn mã phần tử cho hai điểm cuối của đoạn thẳng.
* Kiểm tra xem liệu cả hai điểm đó có nằm ngoài cửa sổ không. Nếu có, đoạn thẳng hoàn toàn nằm ngoài cửa sổ và không cần cắt.
* Nếu cả hai điểm đều ở bên trong cửa sổ, thì đoạn thẳng không cần cắt.
* Nếu một trong hai điểm nằm bên ngoài cửa sổ, ta sẽ kiểm tra các trường hợp đặc biệt:
* Xác định điểm giao giữa đoạn thẳng và cạnh của cửa sổ.
* Thay thế điểm nằm ngoài cửa sổ bằng điểm giao đó.
* Lặp lại quá trình cho đến khi cả hai điểm đều ở bên trong cửa sổ hoặc cả hai điểm đều ở bên ngoài cửa sổ.

Thuật toán Cohen-Sutherland có thể được mở rộng để xử lý các tình huống khác như đoạn thẳng nằm hoàn toàn bên trong, bên ngoài hoặc chỉ chạm vào cửa sổ, giúp chúng ta tìm ra phần của đoạn thẳng mà chúng ta cần vẽ lên màn hình.

### Tổng quan về Game

Game Fruit Ninja 2D là phiên bản đơn giản hóa từ trò chơi mobile nổi tiếng cùng tên. Loại hình game này khá được các bạn trẻ yêu thích vì lối chơi đơn giản nhưng thú vị. Người chơi sẽ sử dụng chuột hoặc cảm ứng để cắt các loại trái cây xuất hiện trên màn hình, mục tiêu là đạt được số điểm cao nhất có thể. Game kết thúc khi người chơi để lỡ quá nhiều trái cây hoặc cắt nhầm các quả bom.

**2.2.2.1 Giới thiệu về Game**

Trong game Fruit Ninja 2D, người chơi sẽ điều khiển một lưỡi dao ảo để cắt trái cây. Khi trái cây xuất hiện trên màn hình, người chơi cần phải nhanh chóng cắt chúng trước khi chúng rơi xuống. Nếu để lỡ quá nhiều trái cây hoặc cắt phải bom, trò chơi sẽ kết thúc.

Điều Khiển: Người chơi sử dụng chuột hoặc cảm ứng để điều khiển lưỡi dao cắt trái cây. Di chuyển lưỡi dao qua trái cây để cắt chúng và ghi điểm.

Mục Tiêu: Cắt càng nhiều trái cây càng tốt để đạt điểm cao nhất

**2.2.2.2. Cách Chơi**

Khi game bắt đầu, các loại trái cây sẽ xuất hiện và di chuyển từ dưới lên trên màn hình theo các quỹ đạo ngẫu nhiên. Người chơi sử dụng chuột hoặc cảm ứng để cắt các loại trái cây này. Mỗi lần cắt trúng trái cây, người chơi sẽ ghi điểm. Trái cây sẽ xuất hiện liên tục và người chơi cần phải nhanh tay và tinh mắt để không bỏ lỡ chúng.

**2.2.2.3. Tính Năng Của Game**

Ngôn Ngữ Lập Trình:

Trò chơi được code bằng ngôn ngữ lập trình Python và sử dụng thư viện Pygame để tạo ra trò chơi.

Cắt Trái Cây:

Người chơi điều khiển lưỡi dao cắt trái cây bằng cách di chuyển chuột hoặc di chuyển ngón tay trên màn hình cảm ứng.

Điểm Số:

Mỗi lần cắt trúng trái cây, người chơi sẽ ghi được một số điểm nhất định. Điểm số hiện tại sẽ được hiển thị trên màn hình.

Quỹ Đạo Trái Cây:

Trái cây xuất hiện và di chuyển theo các quỹ đạo ngẫu nhiên để tăng độ khó cho trò chơi.

Game Fruit Ninja 2D không chỉ đơn giản là một trò chơi giải trí mà còn là một công cụ giúp người chơi rèn luyện kỹ năng phản xạ và sự nhanh nhạy. Với lối chơi đơn giản, hiệu ứng đồ họa sinh động và tính tương tác cao, trò chơi này chắc chắn sẽ mang lại những phút giây giải trí thú vị cho người chơi.

# ứng dụng đồ họa xây dụng game

## Ý tưởng

Ý tưởng cơ bản của trò chơi Fruit Ninja 2D là xây dựng một hệ thống mà trái cây xuất hiện liên tục trên màn hình, và người chơi sẽ sử dụng lưỡi dao để cắt chúng. Mục tiêu của game là đạt được số điểm cao nhất bằng cách cắt được nhiều trái cây nhất có thể.

Người chơi sẽ sử dụng chuột hoặc màn hình cảm ứng để điều khiển lưỡi dao cắt trái cây. Người chơi di chuyển lưỡi dao qua các trái cây để cắt chúng và ghi điểm. Trái cây sẽ xuất hiện và di chuyển theo các quỹ đạo ngẫu nhiên, tăng dần độ khó theo thời gian.

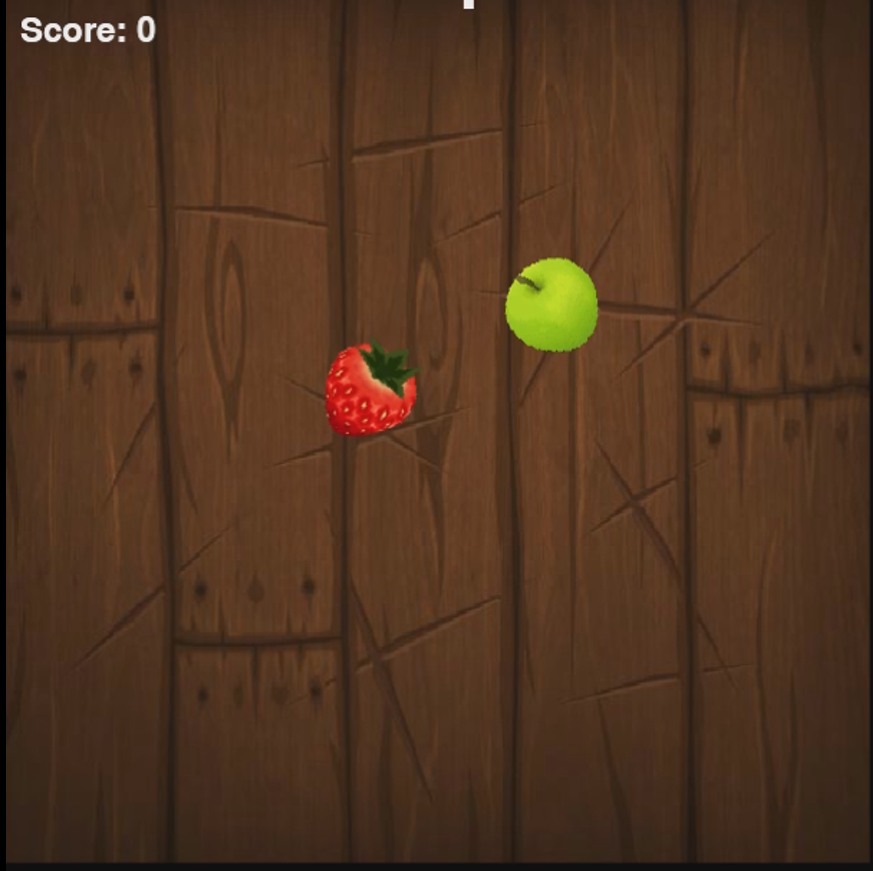
Cách chơi đơn giản nhưng yêu cầu sự nhanh nhạy và phản xạ tốt từ người chơi. Trò chơi mang lại những phút giây giải trí thú vị và giúp người chơi rèn luyện kỹ năng phản xạ và sự tinh mắt.

Hình 2.1. Hình ảnh game khi mới khởi động



## Thiết kế giao diện

### Giao diện chính của trò chơi khi mới bắt đầu

**

*Hình 2.2. Giao diện khi trò chơi mới bắt đầu*

### Giao diện khi người dùng di chuột



*Hình 2.3. Giao diện khi người dùng di chuột*

### Giao diện khi người dùng chém trúng hoa quả và cộng điểm

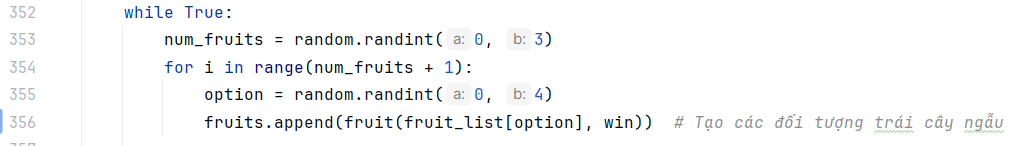


*Hình 2.4. Giao diện khi người dùng chém trúng hoa quả và cộng điểm*

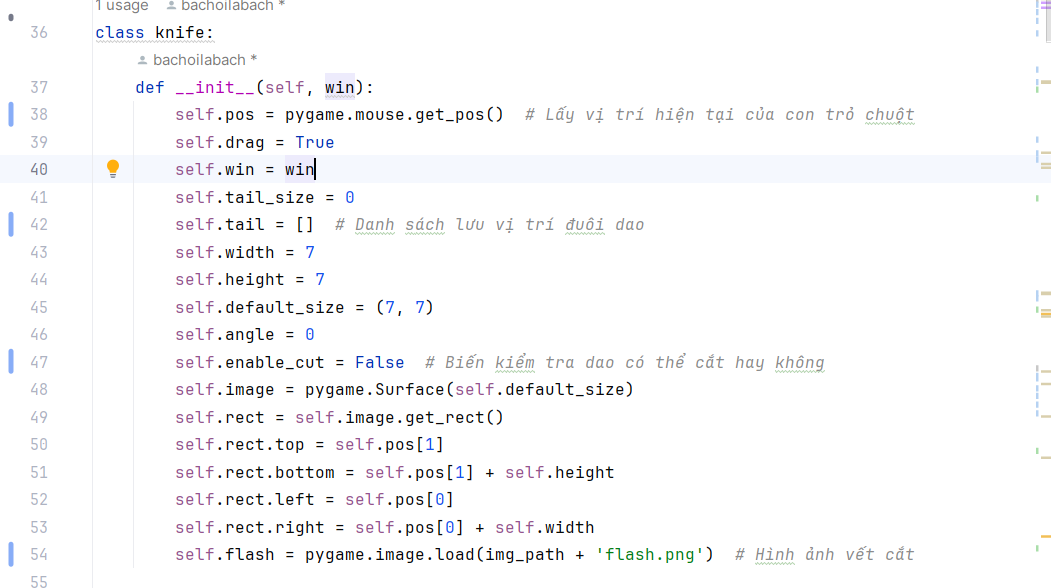
## 3.3 Xây dựng chương trình

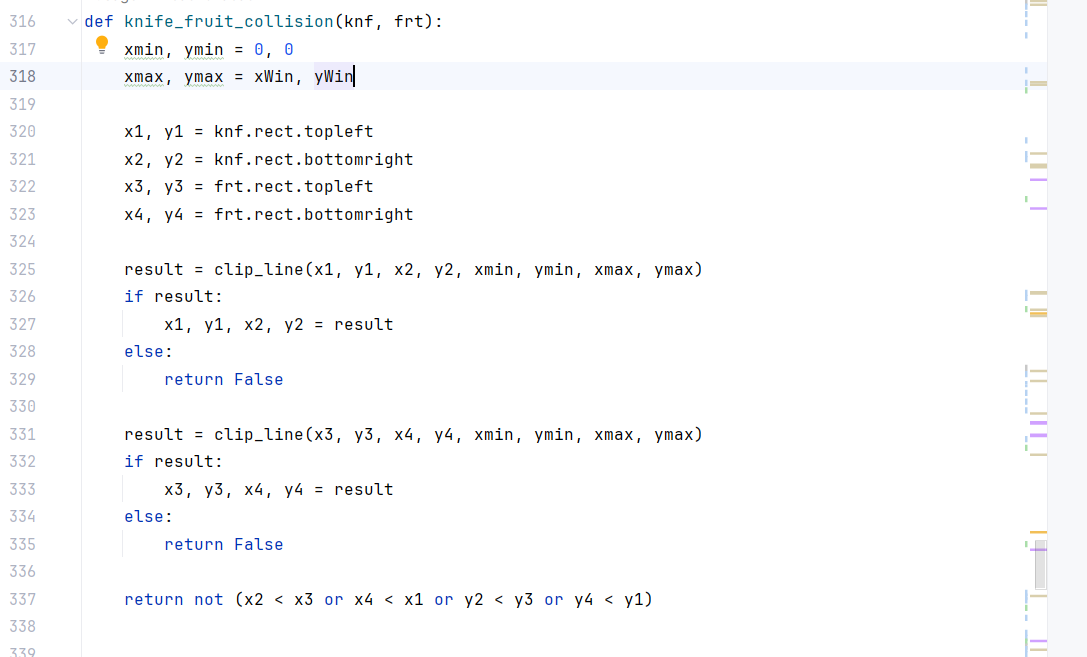
**3.3.1 Xây dựng BackGround** **

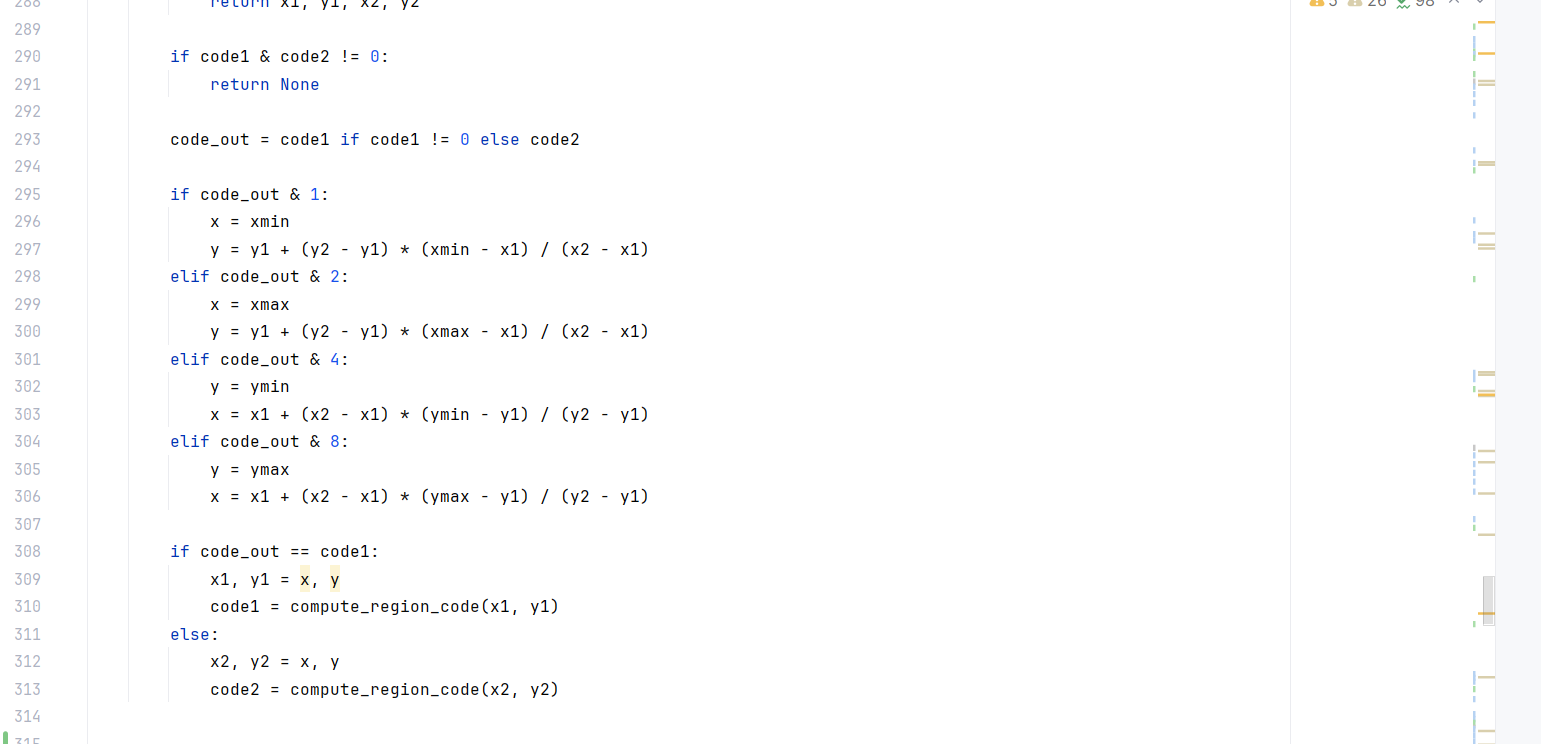
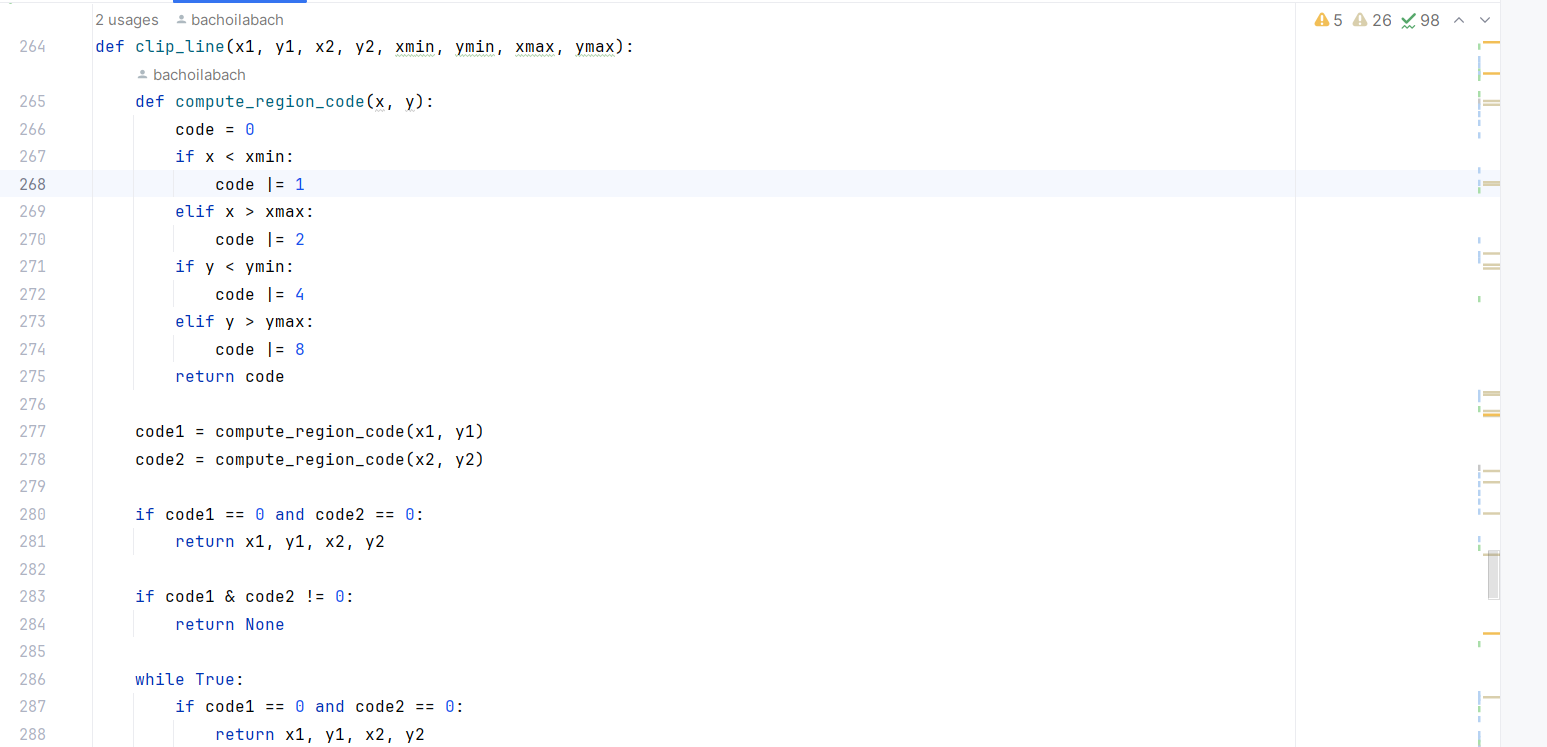
**3.3.4 Xây dựng Spawner(Tự động tạo)**

****

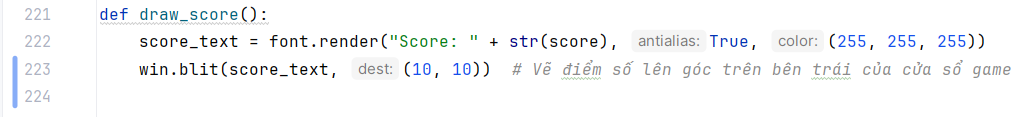
**3.3.5. Xây dựng Blade(lưỡi dao)**



3.3.6 Xây dựng cắt

3.3.7 Xây dựng Score

KẾT LUẬN

TÀI LIỆU THAM KHẢO

PHỤ LỤC