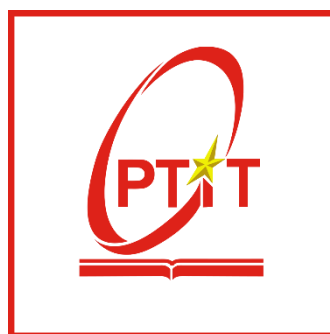


HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



**BÀI TẬP LỚN**  
**MÔN HỌC: LẬP TRÌNH PYTHON**

**Giảng viên: NGUYỄN TRỌNG KHÁNH**

**Nhóm môn học : 10**

**Nhóm bài tập : 4**

**Đề tài : SPACESHIP**

**Thành viên : 1. ĐỖ ĐỨC THIÊN**  
**2. LÊ ĐỨC LONG**  
**3. LÊ MINH QUANG**  
**4. NGUYỄN QUANG MINH**  
**5. LÊ XUÂN BÁCH**

*Hà Nội 2023*



# MỤC LỤC

## I. GIỚI THIỆU

1. Tên đề tài, ý tưởng
2. Công cụ cần thiết

## II. CÀI ĐẶT

1. Mô hình chung
2. Khởi tạo thư viện người dùng các biến và hàm
  - 2.1. Thư viện `init.py`.
  - 2.2. Thư viện `func.py`.
3. Màn hình khởi đầu.
4. Màn hình lựa chọn.
5. Giao diện chính trò chơi.
6. Màn hình kết thúc.

## III. TÀI LIỆU KHAM KHẢO

# I. GIỚI THIỆU

## 1. Tên đề tài, ý tưởng

- Đề tài là một game 2D có tên **Spaceship**.
- **Spaceship** hay có một tên gọi quen thuộc hơn là “Bắn Gà” với luật chơi: người chơi điều khiển phi thuyền chiến đấu đơn độc với chủng tộc gà không gian và cố gắng đạt điểm số cao nhất có thể.
- Có hai chế độ chơi:
  - + Chế độ thứ nhất: các kẻ thù sẽ xuất hiện ngẫu nhiên ở đối diện và di chuyển về phía phi thuyền, nhiệm vụ cụ thể của người chơi là điều khiển phi thuyền tránh va chạm và tiêu diệt được nhiều kẻ thù nhất có thể.
  - + Chế độ thứ hai: các kẻ thù lần này sẽ xuất hiện theo các hàng phía đối diện nhưng sẽ không di chuyển về tới phi thuyền mà di chuyển trái phải, bắn các đường đạn để tấn công phi thuyền, nhiệm vụ của màn này là người chơi né tránh các đường đạn và tiêu diệt hết các kẻ thù.

## 2. Công cụ cần thiết

- Lập Trình Ngôn Ngữ Python Với Thư Viện Pygame:

Pygame là một thư viện (library) và môi trường phát triển cho Python được sử dụng để phát triển trò chơi và ứng dụng đa phương tiện (multimedia) 2D. Pygame là một dự án mã nguồn mở và miễn phí, giúp cho việc phát triển trò chơi trên nền tảng Python trở nên dễ dàng hơn. Thư viện này cung cấp các công cụ và chức năng để vẽ đồ họa, xử lý âm thanh, và quản lý các sự kiện trong trò chơi, chuột và bàn phím cũng được Python hỗ trợ một cách hiệu quả nhất.

Ưu điểm khi lập trình trò chơi bằng Pygame:

Dễ học và dễ sử dụng: Python là một ngôn ngữ lập trình dễ học và dễ đọc, do đó khi sử dụng Pygame, bạn có lợi thế của việc sử dụng một ngôn ngữ ngắn gọn và có cú pháp rõ ràng. Điều này làm cho việc phát triển trò chơi trở nên dễ dàng hơn đối với người mới bắt đầu trong lĩnh vực lập trình game.

Cộng đồng và tài liệu phong phú: Pygame có một cộng đồng mạnh mẽ và nhiều tài liệu học tập, ví dụ và hướng dẫn trực tuyến. Điều này sẽ giúp bạn dễ dàng tìm kiếm giúp đỡ và giải quyết các vấn đề phát triển trò chơi.

Đa nền tảng: Pygame hỗ trợ nhiều hệ điều hành, bao gồm Windows, macOS và nhiều phiên bản của Linux, cho phép bạn phát triển trò chơi một lần và chạy trên nhiều nền tảng khác nhau.

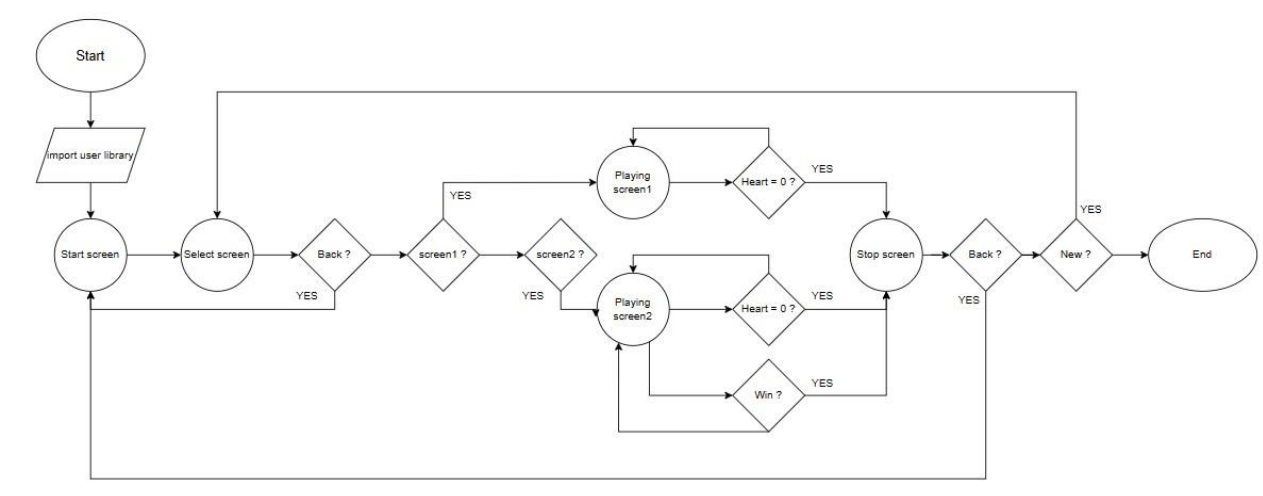
Đồ họa 2D tốt: Pygame cung cấp khả năng vẽ đồ họa 2D một cách dễ dàng, với hỗ trợ cho sprite, animation và hiệu ứng.

- Môi trường phát triển (IDE): Visual Studio Code (VS Code):

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở và miễn phí, phát triển bởi Microsoft. Nó được thiết kế để hỗ trợ lập trình viên trong việc phát triển ứng dụng đa nền tảng với tính năng mở rộng và tùy chỉnh cao.

## II CÀI ĐẶT

### 1. Mô hình chung



Đầu tiên, ta import các thư viện cần thiết và các thư viện người dùng. Một số module quan trọng như:

- random: Cho phép thực hiện các hoạt động liên quan đến biến ngẫu nhiên. Dùng để random các giá trị tạo chick hay chick\_bullet.
- sys: Thư viện liên quan đến các thao tác và thông tin liên quan đến hệ thống.
- init: Thư viện khởi tạo các biến, thông số, sự kiện.

- Func: Thư viện các hàm tạo và quản lý các đối tượng.

Chương trình sẽ bắt đầu với màn hình Start screen. Ở giao diện này người dùng có thể tiến hành bắt đầu trò chơi click vào các nút đang hiển thị trên màn hình để chuyển tới Select screen để chọn chế độ chơi.

Ở chế độ chơi screen1 người chơi bắt đầu với 5 heart và đạt được nhiều điểm nhất có thể, khi heart = 0 màn hình End hiện lên báo trò chơi kết thúc

Ở chế độ chơi screen2 người chơi bắt đầu với 5 heart và chiến thắng 6 màn chơi, khi heart = 0 hay vượt qua 6 màn chơi thì màn hình End hiện lên báo trò chơi kết thúc

## 2. Khởi tạo thư viện người dùng các biến và hàm.

### 2.1 Thư viện biến init.py

Tạo ra các biến và tham số cụ thể nhằm xây dựng tham số trong hàm hoặc dùng làm giá trị trung gian. Các biến quan trọng như :

- Các biến tạo văn bản trên Surface: font,play,new,back.
- Các biến âm thanh: kill, die, level\_up, hades.
- Biến trạng thái game: game.
- Đếm điểm hiện tại: score.
- Kỷ lục điểm cao: hight\_score.
- Mốc giảm tốc độ tạo kẻ địch tiếp theo: speed\_up\_next.
- Máu: heart.
- Các biến sự kiện: spawn\_chick, spawn\_bullet, spawn\_add, spawn\_flap, spawn\_battle

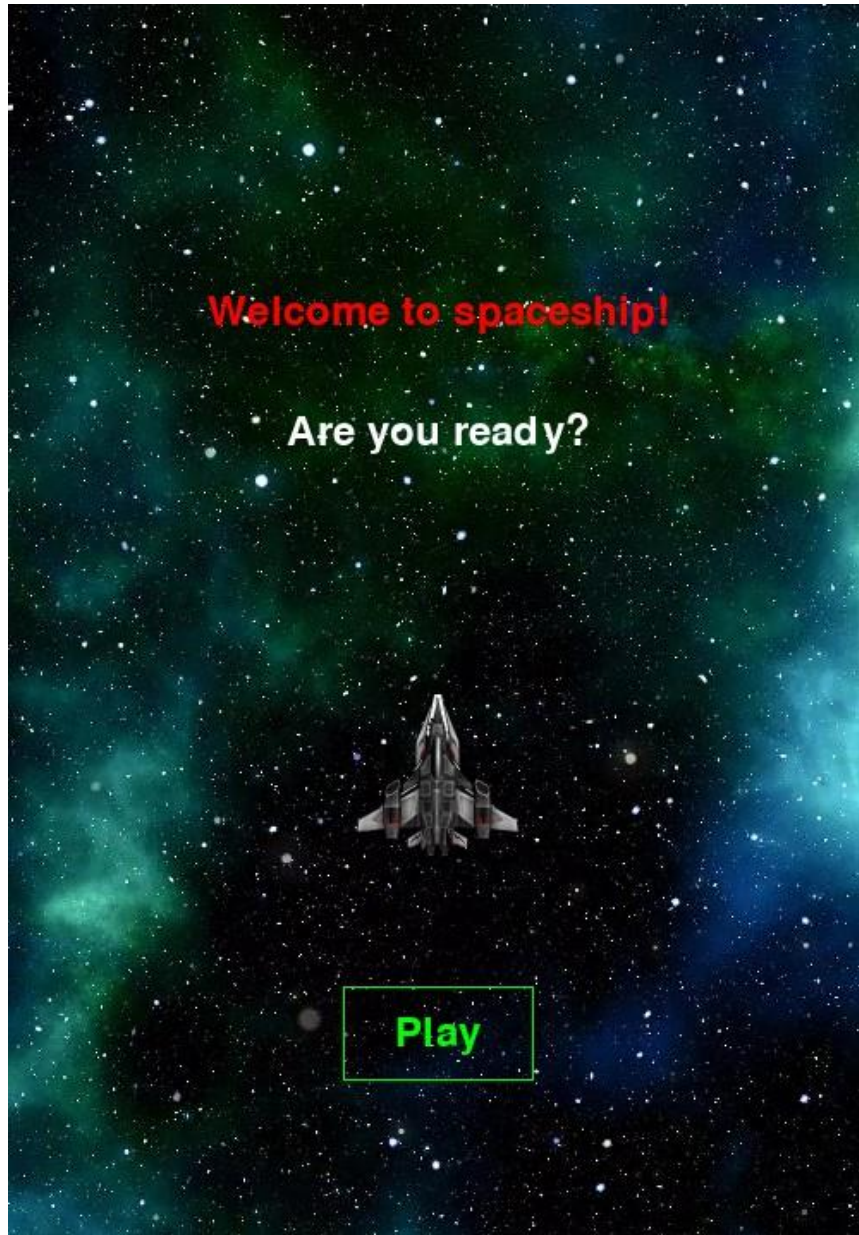
### 2.2 Thư viện hàm func.py

Tạo các quản lý các đối tượng trên Surface

- Nhóm các hàm tạo, di chuyển, hiển thị đối tượng: create\_sp, draw\_chick, create\_bullet,...
- Nhóm hàm tương tác với người dùng: check()...
- Nhóm các hàm tương tác giữa các đối tượng với nhau
  - + Hàm va chạm giữa bullet và chick: cls\_bullet\_chick.
  - + Hàm va chạm giữa sp và chick: cls\_sp\_chick.
  - + Hàm va chạm giữa sp và add: cls\_sp\_add.

- + Hàm va chạm giữa sp và chick\_bullet: cls\_sp\_thigh.
- + Hàm va chạm giữa bullet và boss: cls\_bullet\_boss.

### 3. Màn hình khởi đầu



Giao diện được khởi tạo tại hàm start(), trong hàm sẽ thực hiện lần lượt các bước:

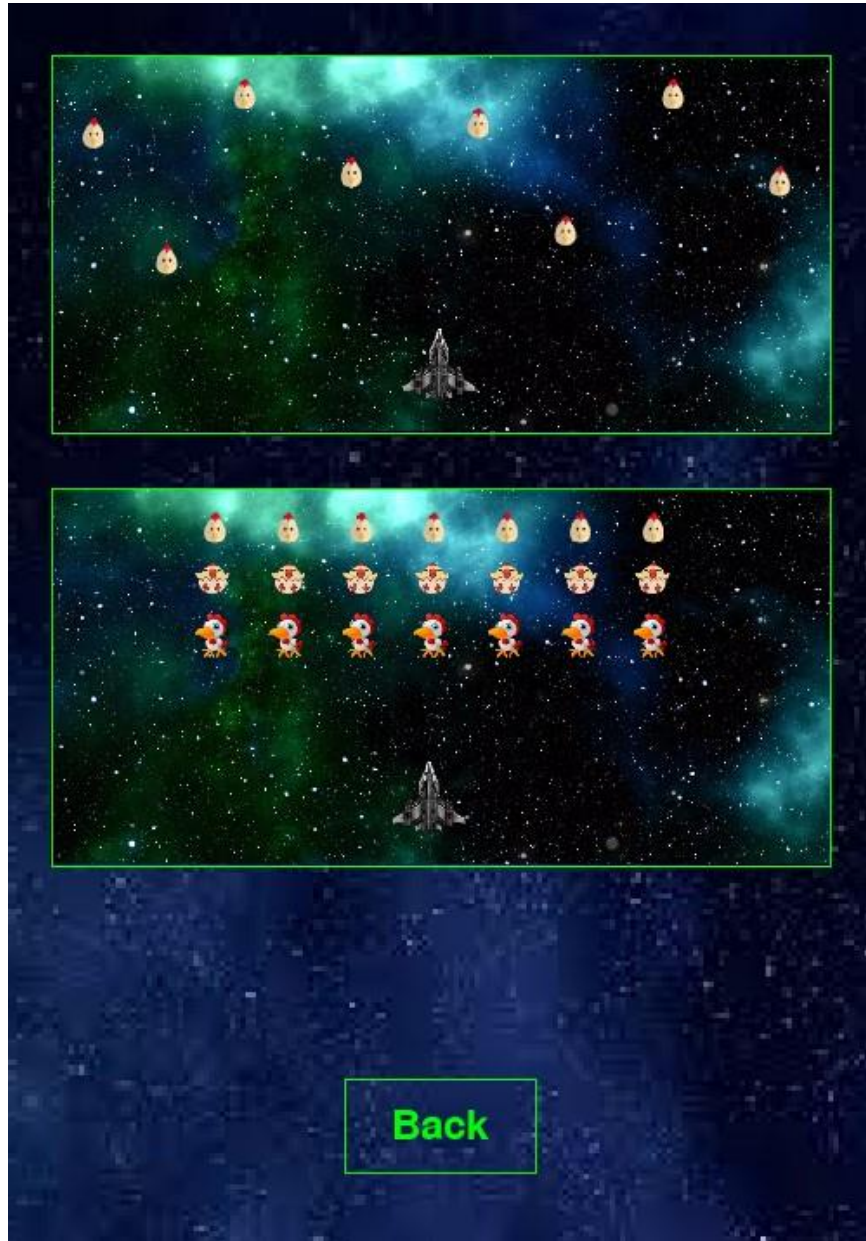
- Hiển thị hình ảnh nền
- Tạo và hiển thị hình ảnh chứa nội dung văn bản: welcome, ready.
- Tạo nút “Play”.

Sau khi đã thực hiện xong việc vẽ đồ họa, chương trình sẽ kiểm tra xem người chơi có click vào nút nào trên màn hình hay không bằng cách lấy tọa độ con trỏ chuột thông qua



phương thức `pygame.mouse.get_pos()`. Ghi nhận sự kiện người dùng click chuột thông qua event `pygame.MOUSEBUTTONDOWN`, kiểm tra tọa độ con trỏ chuột có trùng với tọa độ nút hay không với hàm `check()` và trạng thái game = “start”. Khi người chơi click nút play, hàm `start()` sẽ kết thúc và chuyển sang hàm `select()` để chọn chế độ chơi, cập nhật trạng thái biến game = ‘select’.

#### 4. Màn hình lựa chọn.



*Hình : select screen*

Giao diện được khởi tạo tại hàm `select()` có các ý chính như sau:

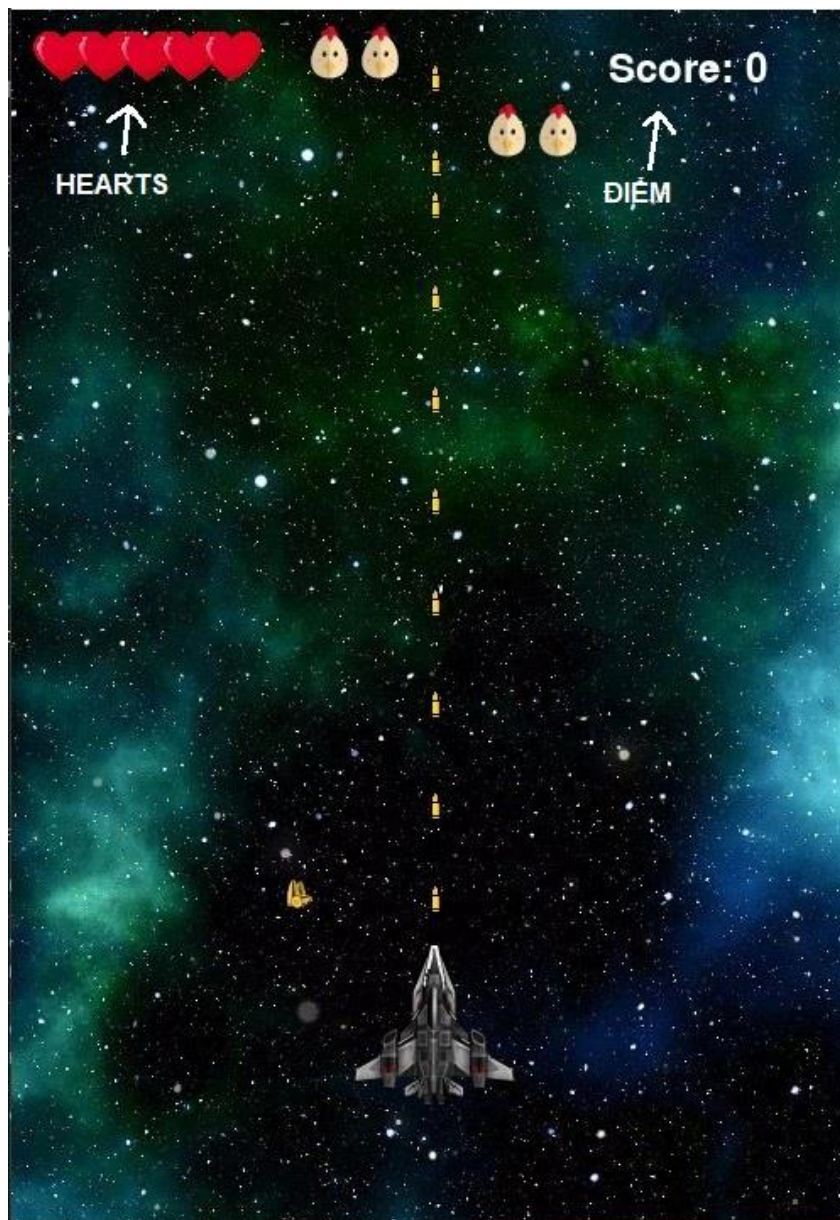
- Hiển thị hình nền.
- Tạo các surface `screen1`, `screen2`.



- Tạo các nút cho screen1, screen2( hình ảnh mô tả) và nút “Back”.

Sau khi đã thực hiện xong việc vẽ đồ hoạ, chương trình ghi nhận sự kiện người dùng click chuột, kiểm tra toạ độ con trỏ chuột có trùng với toạ độ nút hay không thông qua hàm check() và trạng thái game = “select”. Khi người chơi click nút “Back, hàm screen() sẽ kết thúc và quay về hàm start() biến trạng thái game trở thành = “start”, hoặc click vào 1 trong 2 nút “screen1” hay “screen2” để chọn chế độ chơi và chuyển tới màn hình chơi khi đó biến trạng thái game = “screen1” hoặc game = “screen2”.

## 5. Giao diện chính của trò chơi.



Sau khi người chơi chọn chế độ chơi trong Select screen. Chương trình tiến hành vẽ các đối tượng spaceship, chick, bullet,... Ngoài ra, còn có icon heart mô tả số mạng còn lại của người chơi, icon score hiển thị số điểm hiện tại.

Các đối tượng chick, add, bullet được tạo ra bởi các sự kiện được định nghĩa ở trong thư viện init.

Spaceship được người chơi điều khiển thông qua các nút mũi tên và hàm move() để giới hạn Spaceship di chuyển trong màn hình chơi.

Khi bullet và chick được kiểm tra qua hàm cls\_bullet\_chick() nếu vị trí của bullet và chick chạm nhau thì sẽ xóa bullet và chick đó trong list, cập nhật list và score.

Khi spaceship và chick được kiểm tra vị trí qua hàm cls\_sp\_chick () xóa chick đã va chạm và cập nhật list chick, heart hiện tại.

Ở screen1, chick được điều khiển di chuyển từ trên xuống thông qua hàm move\_chick(). Khi score bằng giá trị của biến speed\_up\_next và thời gian xuất hiện sự kiện spawn\_chick >150 chúng ta sẽ giảm thời gian xuất hiện của sự kiện đi 100 và lấy mốc speed\_up\_next cho lần giảm tiếp theo. Ở chế độ chơi này chúng ta sẽ có đối tượng add ( vật phẩm hỗ trợ), khi spaceship chạm vào add hàm cls\_sp\_add() sẽ tăng giới hạn bullet được tạo ra và xóa đối tượng add đó trong list add.

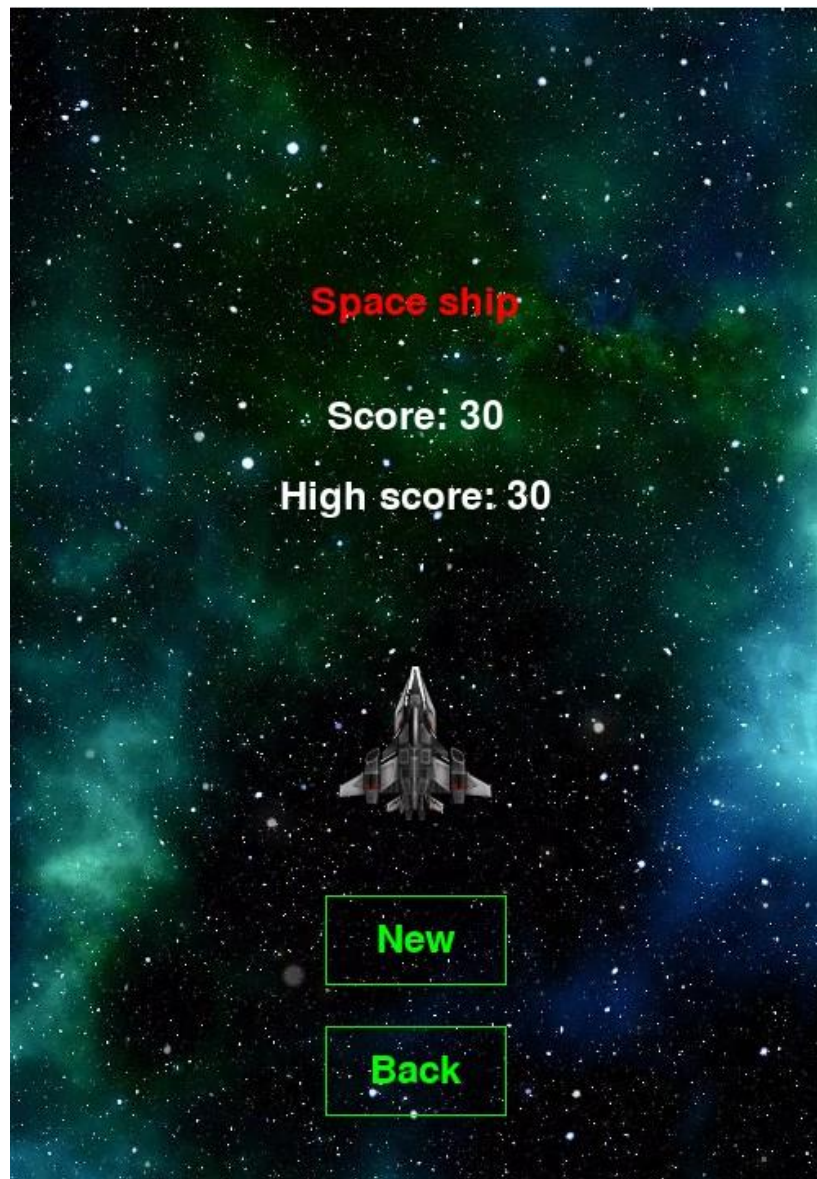
Ở screen2, chick và boss sẽ được điều khiển di chuyển trái phải đối diện spaceship qua hàm move\_chick\_screen2(), chick sẽ xuất hiện 3 hàng và luôn được cập nhật thông qua list battle để cho các đối tượng chick chưa bị xóa bắn đạn. Khi tới màn thứ 3 hoặc 6 sẽ xuất hiện đối tượng boss, đối tượng này được quản lý bởi 1 thanh máu hiển thị cùng, tương tự chúng ta sẽ kiểm tra va chạm giữa boss và bullet để cập nhật giá trị máu của boss. Ở màn chơi này chúng ta sẽ kiểm tra va chạm giữa đạn của chick hay boss với spaceship qua hàm cls\_sp\_thigh() để cập nhật giá trị heart và list thigh.

Khi heart = 0 giá trị của biến trạng thái game = “stop”, gọi biến âm thanh và chuyển tới màn hình kết thúc.



**6. Màn hình kết thúc.**





Giao diện được khởi tạo tại hàm stop(), trong hàm sẽ thực hiện lần lượt các bước:

- Hiển thị hình ảnh nền
- Cập nhật high score.
- Xóa các list đối tượng như chick, bullet, add,..
- Tạo và hiển thị hình ảnh chứa nội dung văn bản: space ship, score, high score.
- Tạo nút “New”, ”Back”.

Sau khi đã thực hiện xong việc vẽ đồ họa và cập nhật giá trị các biến và list, chương trình sẽ kiểm tra xem người chơi có click vào nút nào trên màn hình hay không . Khi người chơi click nút “New” hàm stop kết thúc và chuyển tới select screen biến game = “select”, click nút “Back” để chuyển tới start screen biến game = “start”.

### III TÀI LIỆU THAM KHẢO

Tài liệu hướng dẫn sử dụng thư viện pygame:

[https://www.pygame.org/docs/ref/mixer.html?fbclid=IwAR0lkBBqDpX-XTMKQOx9VZNiNwVt92G2HIAa3XkE4\\_IKHt2lOPnghTsLIc](https://www.pygame.org/docs/ref/mixer.html?fbclid=IwAR0lkBBqDpX-XTMKQOx9VZNiNwVt92G2HIAa3XkE4_IKHt2lOPnghTsLIc)

Họ và tên	Phân công công việc
<b>ĐỖ ĐỨC THIÊN</b>	Lên ý tưởng game, mô tả thuật toán, cài đặt game
<b>LÊ ĐỨC LONG</b>	Cài đặt game
<b>LÊ MINH QUANG</b>	Cài đặt game
<b>NGUYỄN QUANG MINH</b>	Viết báo cáo, kiểm thử game
<b>LÊ XUÂN BÁCH</b>	Tìm đồ họa và âm thanh, kiểm thử game