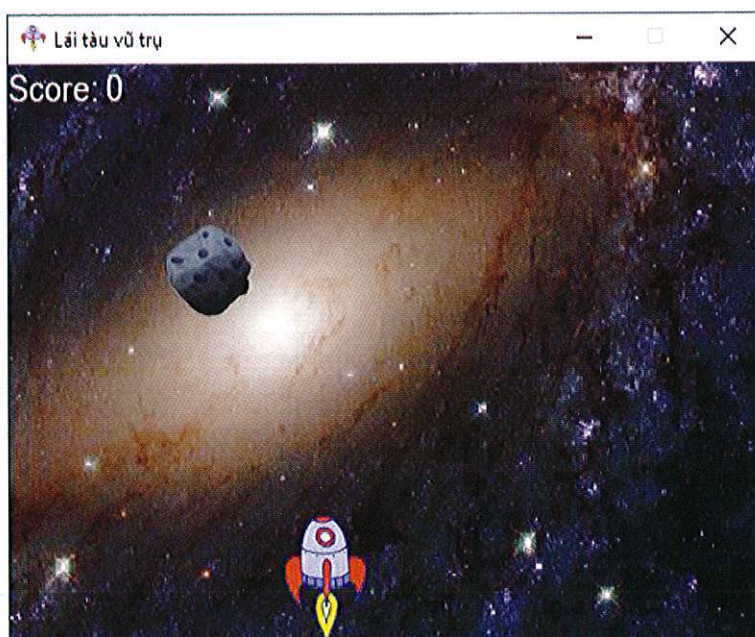


## Trò chơi Lái tàu vũ trụ

Trong chương này, các bạn sẽ được tìm hiểu cách thiết kế và lập một trò chơi trên máy tính có tên “Lái tàu vũ trụ”. Trong trò chơi này, người chơi cần dùng các phím mũi tên trái và phải để điều khiển con tàu di chuyển và tránh các thiên thạch rơi xuống tại vị trí bất kỳ từ bên trên màn hình.

Chúng ta sẽ trải qua các bước sau:

- Thêm hình nền, Tàu vũ trụ, Thiên thạch và vẽ hình ảnh lên màn hình.
- Lập trình điều khiển Tàu vũ trụ bằng các phím mũi tên.
- Lập trình Thiên thạch rơi.
- Lập trình xử lý va chạm giữa Tàu vũ trụ và Thiên thạch.



## Tổng quan chương trình

Chương trình gồm các thành phần chính:

### 1. Hình nền

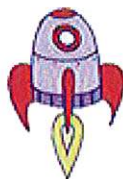


### 2. Thiên thạch



Hoạt động: Thiên thạch xuất hiện tại vị trí ngẫu nhiên bên trên và rơi xuống dưới. Tới cuối màn hình thiên thạch sẽ quay lại lên trên. Nếu thiên thạch rơi trúng Tàu vũ trụ, trò chơi sẽ dừng lại và kết thúc.

### 3. Tàu vũ trụ



Hoạt động: Tàu vũ trụ được điều khiển bằng các phím mũi tên trái, phải để di chuyển tránh thiên thạch. Nếu chạm vào thiên thạch, tàu sẽ biến mất, tiếng tàu nổ được phát ra và chương trình kết thúc.

### 4. Điểm số

Score: 0

Hoạt động: Mỗi lần Tàu vũ trụ tránh được thiên thạch, điểm sẽ được cộng thêm 1.

Tất cả hình ảnh tàu, thiên thạch, hình nền và âm thanh nổ đều có sẵn trong tài liệu của cuốn sách.



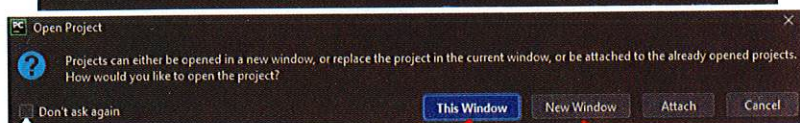
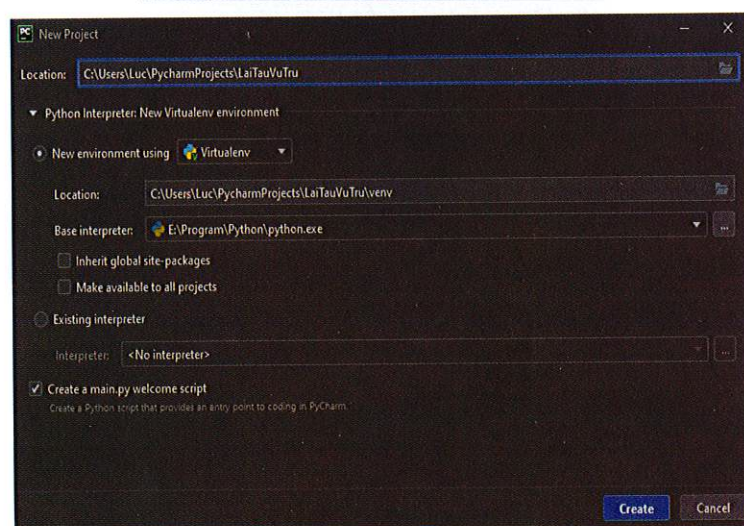
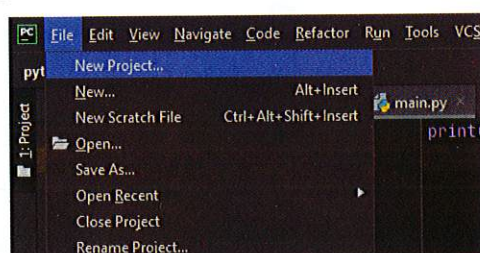
# BÀI 1

## THIẾT LẬP CỬA SỔ HIỂN THỊ

Trong bài học này, chúng ta sẽ tìm hiểu về thư viện pygame, lập trình tạo ra cửa sổ và thiết lập các thông tin cho cửa sổ như tiêu đề, biểu tượng, hình nền.

### 1.1. Tạo dự án và cài đặt thư viện pygame

Khi mở PyCharm, chương trình sẽ mở lại dự án làm việc gần nhất của các bạn. Để tạo dự án mới, các bạn chọn **File ► New Project**. Các bạn đặt tên cho dự án (ví dụ: LaiTauVuTru), kiểm tra đường dẫn và nhấn **Create**. Sau khi nhấn, cửa sổ chọn cách mở sẽ xuất hiện, các bạn tích vào mục **Don't ask again** để những lần tạo dự án mới sau này sẽ không xuất hiện cửa sổ này nữa, tiếp đó nhấn lên lựa chọn **This Window** hoặc **New Window**. Dự án mới sẽ được mở với file **main.py** có sẵn, các bạn xóa những nội dung mặc định của file này.



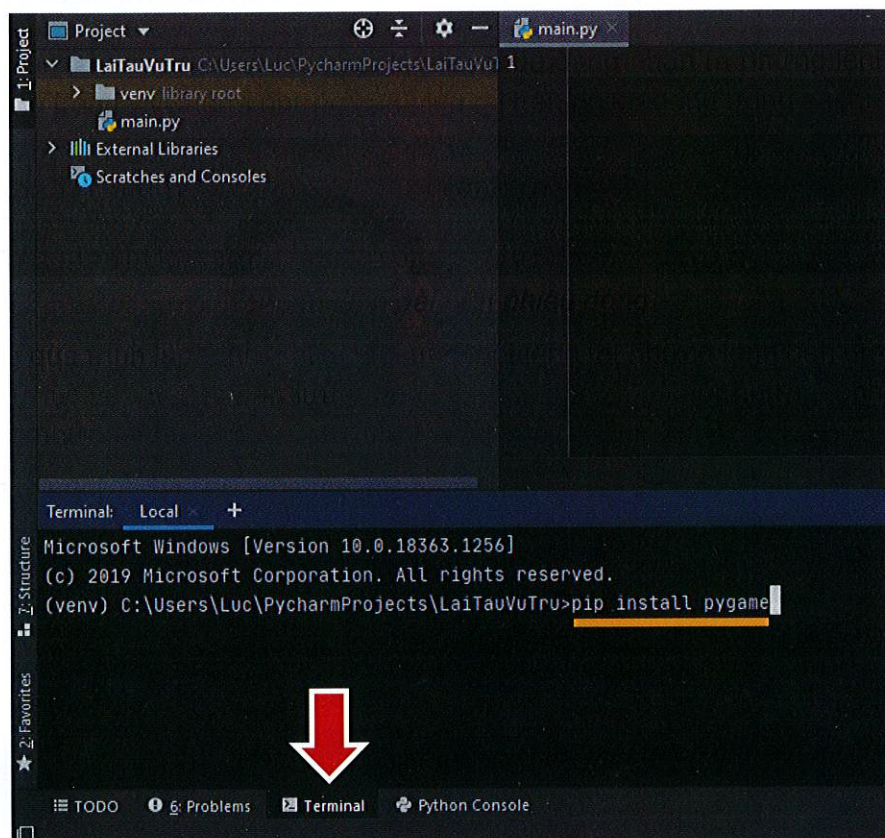
Mở dự án mới trên cửa sổ hiện tại

Mở dự án mới trên cửa sổ mới



Sau khi cài đặt, môi trường phát triển PyCharm hỗ trợ các hàm có chức năng rất cơ bản, trong khi đó chúng ta thường xuyên gặp phải trường hợp cần sử dụng những chức năng phức tạp mà môi trường không có sẵn, ví dụ như tạo giao diện đồ họa, hiển thị hình ảnh, nhập xuất file, các phép toán lượng giác, nhận sự kiện từ bàn phím và chuột... Lúc này, chúng ta cần nhập (import) các thư viện (library) hỗ trợ các chức năng đó để sử dụng. Những thư viện này còn được gọi là các mô-đun (module), gồm một hoặc nhiều file mã nguồn. Một số thư viện được tích hợp sẵn trong quá trình cài đặt Python và PyCharm, chúng ta chỉ cần nhập để sử dụng.

Trong ứng dụng này, chúng ta cần sử dụng thư viện **pygame** hỗ trợ việc lập trình game. Các bạn cài đặt thư viện bằng cách mở thẻ **Terminal** và gõ lệnh **pip install pygame**, sau đó nhấn **Enter**. PyCharm sẽ tiến hành cài đặt thư viện, các bạn lưu ý khi cài đặt cần có kết nối mạng.



## 1.2. Lập trình hiển thị cửa sổ

Sau khi đã cài đặt, các bạn import thư viện vào chương trình bằng lệnh **import pygame** và có thể sử dụng các hàm được cung cấp bởi thư viện này. Bên trong một thư viện sẽ có các hàm hoặc module nhỏ hơn, mỗi hàm hoặc module này sẽ có vai trò nhất định.





Để gọi một hàm, chúng ta cần gõ lệnh theo cú pháp **<module>.<tên hàm>**, trong đó **<module>** là thư viện hoặc module chứa hàm đó. Ví dụ đoạn chương trình dưới đây gọi hàm khởi tạo thư viện và thiết lập cửa sổ.

```
1 import pygame # nhập thư viện
2 pygame.init() # khởi tạo thư viện
3 pygame.display.set_mode((480, 360)) # Thiết lập cửa sổ
```

**pygame.init()**

module

hàm

**pygame.display.set\_mode((480, 360))**

module

module

hàm

Hàm **init()** trong module **pygame** thực hiện việc khởi tạo thư viện, được sử dụng bằng cách gọi lệnh **pygame.init()**. Hàm **set\_mode((480, 360))** thực hiện việc tạo cửa sổ hiển thị và thiết lập cửa sổ có kích thước 480x360 (đơn vị pixel – điểm ảnh trên màn hình). Các bạn có thể thấy hàm này thuộc module **display** bên trong **pygame**, được sử dụng bằng lệnh **pygame.display.set\_mode((480, 360))**.


Hàm **set\_mode()** yêu cầu 1 tham số là cặp giá trị chiều dài và chiều rộng cửa sổ. Các bạn lưu ý với cách gọi **set\_mode((480, 360))**, chúng ta hiểu rằng hàm **set\_mode()** được truyền một tham số đầu vào là cặp giá trị **(480, 360)**, tránh nhầm lẫn với **set\_mode(480, 360)** là truyền 2 tham số 480 và 360. Đến đây chúng ta có thể suy luận được **init()** là hàm không cần tham số nào, trong các bài học sau chúng ta sẽ phân tích kỹ hơn việc truyền tham số cho hàm.

# truyền 1 tham số là cặp giá trị (480, 360) -> cách gọi đúng  
set\_mode((480, 360))

# truyền 2 tham số riêng biệt là 480 và 360 -> cách gọi sai  
set\_mode(480, 360)

Việc thiết lập cửa sổ sẽ trả về kết quả là cửa sổ hiển thị trên màn hình, chúng ta cần gán vào một biến (ví dụ: **screen**) để có thể sử dụng sau này.

```
1 import pygame # nhập thư viện
2 pygame.init() # khởi tạo thư viện
   # gán cửa sổ vào biến screen để có thể sử dụng sau này
3 screen = pygame.display.set_mode((480, 360))
```

Các bạn lập trình đoạn code như trong hình trên và nhấn vào biểu tượng  để chạy thử, các bạn sẽ thấy một cửa sổ xuất hiện và biến mất ngay lập tức.







Trong lập trình giao diện đồ họa với Python, về bản chất, chúng ta cần thực hiện liên tục (lặp đi lặp lại) việc “vẽ” lên màn hình những gì chúng ta muốn hiển thị, gồm có vẽ ra cửa sổ, vẽ hình nền, vẽ con tàu, vẽ thiên thạch... Việc hiển thị cửa sổ được thực hiện bằng lệnh **pygame.display.update()**. Để lệnh này có thể thực hiện liên tục lặp đi lặp lại, chúng ta sử dụng vòng lặp **while** với cú pháp:

```
while <điều kiện>:
    # Các lệnh nằm bên trong vòng lặp while được lặp đi lặp
    # lại, chú ý thụt lề một cấp so với lệnh while
    <các lệnh cần thực hiện>

# Vị trí này là các lệnh bên ngoài vòng lặp while và có cùng
# cấp (cùng mức thụt lề) với vòng lặp while, sẽ được thực hiện
# sau khi vòng lặp kết thúc
```

Vòng lặp sẽ kiểm tra điều kiện, nếu đúng thì thực hiện các lệnh bên dưới, nếu sai sẽ dừng vòng lặp. Những lệnh được lặp lại này cùng cấp với nhau và một nhóm lệnh cùng cấp như vậy được gọi là một **block**. Các ngôn ngữ lập trình khác thường có ký tự để đánh dấu mở đầu và kết thúc một block lệnh (ví dụ trong C/C++ là cặp dấu ngoặc nhọn { }), trong khi đó Python sử dụng mức thụt lề (**indentation** - số lượng dấu cách space và dấu tab tính từ đầu dòng) để quy định các block, những lệnh có cùng mức thụt lề giống nhau được coi là cùng một block. Việc thụt lề trong Python rất quan trọng, các bạn cần thống nhất sử dụng chỉ dấu space hoặc chỉ dấu tab để thụt dòng chứ không dùng lẫn lộn cả space và tab, ngoài ra nếu các câu lệnh thụt lề sai sẽ dẫn đến lỗi ngay lập tức, vì vậy các bạn cần chú ý điều này.

**block 1 nằm bên trong block 2**


```
block 2 | while <điều kiện>:
        | :
        | | dòng lệnh 1
        | | dòng lệnh 2
        | | dòng lệnh 3
```

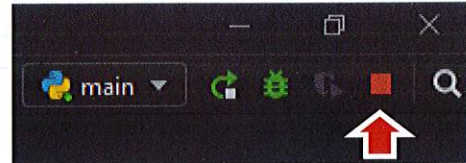
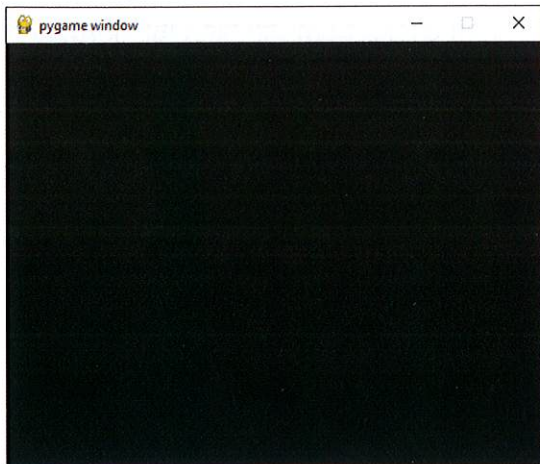
Để vòng lặp **while** cứ thế thực hiện lặp đi lặp lại, chúng ta đặt điều kiện kiểm tra là **True**. Đây là vòng lặp cập nhật trạng thái chính của chương trình (còn gọi là lệnh lặp chính), những việc vẽ ảnh hay cập nhật trạng thái đều thực hiện bên trong vòng lặp này. Các bạn bổ sung thêm các lệnh như hình dưới đây và chạy thử.




```
1 import pygame
2 pygame.init()
3 screen = pygame.display.set_mode((480, 360))
```



```
# bổ. sung vòng lặp
5 while True:
6     pygame.display.update()
```

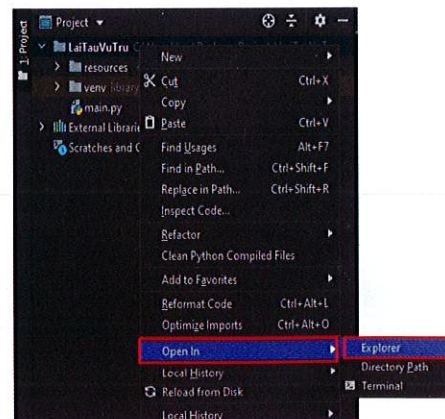
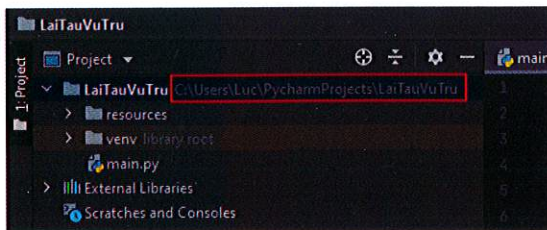
Các bạn sẽ thấy một cửa sổ xuất hiện và không biến mất nữa, nhưng chúng ta nhấn dấu  cũng không tắt được cửa sổ này đi. Hình dưới đây mô tả thành phần và cách tắt cửa sổ này.



Để có thể tắt cửa sổ khi nhấn dấu , chúng ta cần liên tục kiểm tra xem sự kiện người dùng nhấn nút  có xảy ra hay không, việc này còn được gọi là “bắt sự kiện”. Việc lập trình bắt sự kiện nhấn nút  sẽ được hướng dẫn chi tiết trong bài học sau, trong bài học này chúng ta sẽ sử dụng cách tắt thủ công như trên.

### 1.3. Lập trình thiết lập tiêu đề, icon và hình nền

Trước khi lập trình những phần tiếp theo, chúng ta sẽ tải về tài liệu là những hình ảnh và âm thanh sử dụng trong chương trình. Trước tiên, các bạn truy cập vào thư mục của dự án bằng cách thủ công theo đường dẫn tới vị trí lưu dự án (vị trí cạnh tên dự án) hoặc nhấn phải chuột vào tên dự án, chọn **Open in**, chọn tiếp **Explorer**. Mặc định, dự án có các thư mục và tệp tin như mô tả ở hình dưới đây.



PycharmProjects > LaiTauVuTru		
Name	Type	Size
.idea	File folder	
venv	File folder	
main.py	Python File	1 KB

Khi lập trình một dự án bất kỳ, tất cả các tài liệu (hình ảnh, video, âm thanh...) được sử dụng trong dự án nên được để trong một thư mục nhất định nằm bên trong thư mục của dự án đó. Trong dự án Lái tàu vũ trụ này, chúng ta sẽ tạo thư mục (ví dụ: "resources", các bạn có thể đặt tên khác) để chứa tất cả các tài liệu của chương trình. Trong các dự án có nhiều tài liệu, các bạn nên lưu mỗi loại tài liệu (âm thanh, hình ảnh,...) vào một thư mục riêng để dễ dàng phân loại, nhưng trong ví dụ này số lượng tài liệu rất ít nên chúng ta có thể để chung. Các bạn sao chép các tài liệu đã tải về của dự án Lái tàu vũ trụ vào thư mục resources mới tạo

PycharmProjects > LaiTauVuTru		
Name	Type	Size
.idea	File folder	
resources	File folder	
venv	File folder	
main.py	Python File	1 KB

### Tạo thư mục **resources**



### Đưa các tài liệu dự án vào trong thư mục **resources**

Sau khi đã có hình ảnh, chúng ta sẽ lập trình thiết lập tiêu đề, icon và hình nền (icon và hình nền sử dụng hình ảnh tài liệu trên). Những lệnh này được đặt ở vị trí trước vòng lặp **while**.

Đầu tiên, các bạn sử dụng lệnh dưới đây để đặt tiêu đề (ví dụ: "Lái tàu vũ trụ", các bạn có thể đổi tiêu đề khác tùy ý).



```
pygame.display.set_caption("Lái tàu vũ trụ")
```

Tiếp theo là thiết lập icon và hình nền. Trước khi lập trình hiển thị hình ảnh, các bạn cần tải ảnh đó vào trong chương trình bằng lệnh **pygame.image.load(<đường dẫn đến tệp tin ảnh>)** và gán vào một biến, sau đó mới có thể sử dụng, chúng ta có thể hiểu biến này có kiểu dữ liệu là một đối tượng ảnh. Hình dưới đây mô tả lệnh thiết lập icon bằng cách sử dụng biến **ship\_img** để tải ảnh **rocketship.png** trong thư mục **resources**, sau đó sử dụng lệnh **pygame.display.set\_icon(<đối tượng ảnh>)** để thiết lập icon sử dụng biến **ship\_img** trên. Thông tin đường dẫn **"resources/rocketship.png"** là đường dẫn kể từ vị trí tệp tin **main.py**.

```
# tạo biến ship_img để tải ảnh rocketship.png trong thư mục
# resources, biến này có kiểu dữ liệu là đối tượng ảnh
ship_img = pygame.image.load("resources/rocketship.png")

# thiết lập hình ảnh của biến icon cho chương trình
pygame.display.set_icon(ship_img)
```

Việc thêm ảnh nền cũng cần tạo một biến và tải ảnh tương tự như vậy, nhưng thiết lập ảnh đó làm hình nền có chút khác biệt. Về bản chất, sau khi tải ảnh đó, chúng ta sẽ liên tục "vẽ" ảnh đó sao cho kín được toàn bộ màn hình. Việc vẽ này được thực hiện bởi đối tượng cửa sổ **screen** bằng lệnh **screen.blit(<đối tượng ảnh>, <tọa độ vẽ>)**, vị trí ở bên trong lệnh lặp chính (vòng lặp **while** mà chúng ta đã lập trình trước đó). Tọa độ vẽ là một cặp số (x, y) tương đương tọa độ x và tọa độ y của ảnh.

```
# tạo biến bg_img để tải ảnh bg_galaxy.png
bg_img = pygame.image.load("resources/bg_galaxy.png")

while True:
    # bổ sung việc vẽ hình nền vào trong vòng lặp
    screen.blit(bg_img, (0, 0))

    pygame.display.update()
```

Dưới đây là đoạn chương trình sau khi đã thêm các lệnh thiết lập tiêu đề, icon và hình nền.

```
1 import pygame
2 pygame.init()
3 screen = pygame.display.set_mode((480, 360))
4 pygame.display.set_caption("Lái tàu vũ trụ")
5 ship_img = pygame.image.load("resources/rocketship.png")
6 pygame.display.set_icon(ship_img)
7 bg_img = pygame.image.load("resources/bg_galaxy.png")
```





```
8 while True:
9     screen.blit(bg_img, (0, 0))
10    pygame.display.update()
```



### Tóm tắt lý thuyết và bài tập thực hành

Trong bài học đầu tiên, chúng ta đã được tìm hiểu cách tạo dự án với Pycharm, cài đặt thư viện pygame, lập trình hiển thị và thiết lập các thông tin cho cửa sổ như kích thước, tiêu đề, icon.

**Bài tập 1.** Lập trình khởi tạo cửa sổ kích thước 720 x 540 pixel.

**Bài tập 2.** Thêm hình ảnh ngôi sao và lập trình cho ngôi sao xuất hiện tại các vị trí góc trên bên trái, góc trên bên phải, góc dưới bên trái, góc dưới bên phải cửa sổ.

**Bài tập 3:** Thay đổi icon tùy chọn cho cửa sổ.