Để thoát chương trình, chúng ta cần import thêm thư viện **sys,** sử dụng 2 lệnh **pygame.quit()** và **sys.exit().** Hình dưới đây mô tả toàn bộ chương trình đến thời điểm hiện tại, lúc này các bạn có thể nhấn nút ☒ để thoát chương trình.

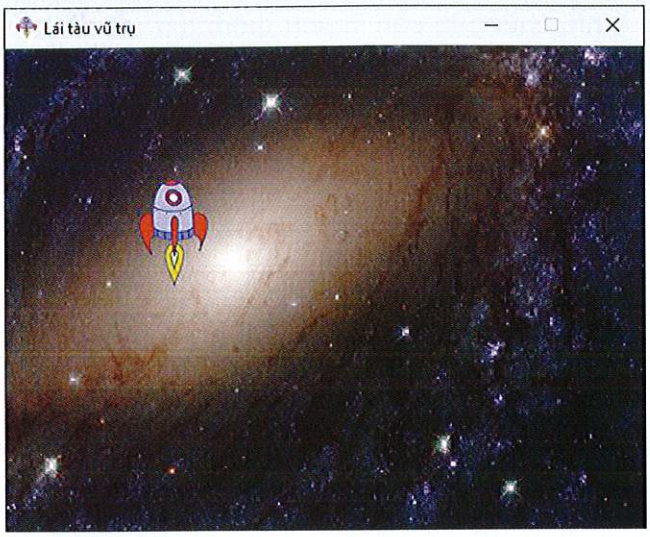
```
1  import pygame
2  import sys     # bổ sung thư viện sys

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

9  while True:
10   for event in pygame.event.get():
11     if event.type == pygame.QUIT:
11       # bổ sung 2 lệnh thoát chương trình
12       pygame.quit()
13       sys.exit()

14   screen.blit(bg_img, (0, 0))
15   pygame.display.update()
```

2.2. Thêm tàu vũ trụ và bắt sự kiện nhấn phím trên bàn phím

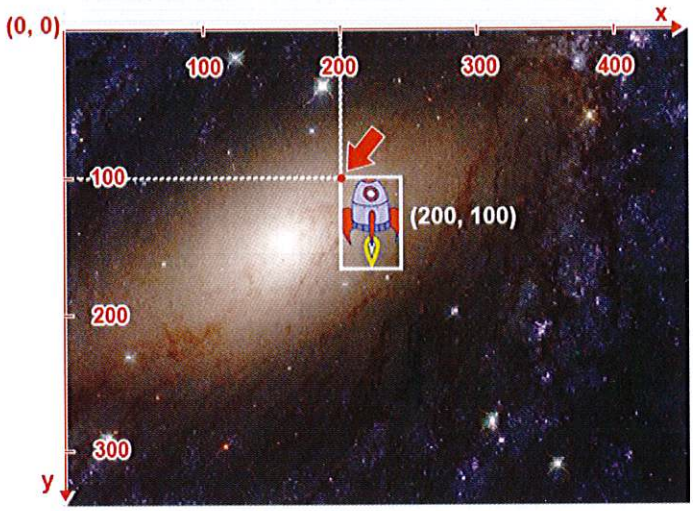
Để thêm con tàu, chúng ta cũng cần thực hiện các bước tải ảnh vào chương trình và vẽ ra màn hình. Chúng ta đã có biến **ship_img** đã chứa dữ liệu ảnh con tàu từ trước, chúng ta chỉ cần bổ sung việc vẽ ảnh con tàu bằng lệnh **screen.blit()** phía dưới lệnh vẽ hình nền. Các bạn bổ sung thêm lệnh vẽ con tàu ở tọa độ (100, 100) và chạy chương trình.

Lưu ý: Hình ảnh các đối tượng được vẽ trên màn hình được sắp xếp đứng trước hay đứng sau theo thứ tự lập trình, các hình ảnh vẽ sau sẽ đè lên trên các hình ảnh được vẽ trước đó, vì vậy lệnh vẽ con tàu cần đặt phía dưới lệnh vẽ hình nền.

```
14   screen.blit(bg_image, (0, 0))
     # bổ sung lệnh vẽ tàu ở vị trí sau lệnh vẽ nền
15   screen.blit(ship_img, (100, 100))
16   pygame.display.update()
```



Hình dưới đây giải thích về tọa độ vẽ các hình. Cửa sổ chương trình là một hệ tọa độ có tâm (0, 0) tại vị trí góc trên bên trái và kích thước được thiết lập bằng lệnh **set_mode()**, giá trị hoành độ x tăng từ trái sang phải, giá trị tung độ y tăng từ trên xuống dưới. Tọa độ của một hình vẽ bất kỳ có vị trí góc trên bên trái hình đó.



Về bản chất, chúng ta sử dụng vòng lặp while để liên tục vẽ các hình ảnh lên màn hình, việc điều khiển một hình ảnh di chuyển chính là việc thay đổi vị trí tọa độ vẽ hình. Các bạn tạo hai biến để lưu vị trí x và y của tàu vũ trụ (ví dụ: "ship_x" và "ship_y"), sau đó thay đổi lệnh vẽ để sử dụng hai biến này làm vị trí vẽ. Giá trị khởi tạo của biến là vị trí bắt đầu của tàu, các bạn tùy ý lựa chọn vị trí này.

```
...     ...
8   bg_img = pygame.image.load("resources/bg_galaxy.png")

9   ship_x = 220
10  ship_y = 280
```

```
11 while True:
…     ...
17    screen.blit(ship_img, (ship_x, ship_y))
18    pygame.display.update()
```

Tiếp theo, chúng ta sẽ lập trình để bắt sự kiện nhấn phím mũi tên trái và phải trên bàn phím. Chúng ta cần kiểm tra kiểu sự kiện là sự kiện có một phím bất kỳ được nhấn (**pygame.KEYDOWN**) trước, sau đó mới kiểm tra phím được nhấn có phải phím mũi tên trái/phải hay không bằng cách so sánh thuộc tính **event.key** với giá trị **pygame.K_LEFT** (phím mũi tên trái) và **pygame.K_RIGHT** (phím mũi tên phải).

```
11 while True:
12    for event in pygame.event.get():
13       if event.type == pygame.QUIT:
14          pygame.quit()
15          sys.exit()

          #bổ sung lệnh kiểm tra sự kiện có một phím bất kỳ được nhấn
16       if event.type == pygame.KEYDOWN:
            #tiếp tục kiểm tra phím được nhấn là phím nào
17          if event.key == pygame.K_LEFT:  #phím mũi tên trái
18             pass
19          if event.key == pygame.K_RIGHT:  #phím mũi tên phải
20             pass
```

pygame.QUIT, **pygame.KEYDOWN**, **pygame.K_LEFT**, **pygame.K_RIGHT** là các sự kiện được quy ước sẵn bởi nhà phát triển thư viện pygame. Để có thể biết được các sự kiện khác (ví dụ như nhấn chuột, nhấn phím khác trên bàn phím…), chúng ta tìm kiếm trên mạng và đọc thêm tài liệu của nhà phát triển, ví dụ đường dẫn sau là tài liệu về sự kiện nhấn các phím: https://www.pygame.org/docs/ref/key.html

Khi mũi tên trái được nhấn, ta sẽ giảm tọa độ x của tàu (tương đương với việc giảm giá trị biến **ship_x**), tương tự như vậy với việc lập trình mũi tên phải.

```
16       if event.type == pygame.KEYDOWN:
17          if event.key == pygame.K_LEFT:
18             ship_x -= 5  # tương đương ship_x = ship_x - 5
19          if event.key == pygame.K_RIGHT:
20             ship_x += 5    # tương đương ship_x = ship_x + 5
```

Khi chạy chương trình và nhấn phím mũi tên sang phải, sang trái trên bàn phím, các bạn sẽ thấy tàu vũ trụ di chuyển sang phải và sang trái tương ứng.

Các bạn có thể thay đổi tốc độ di chuyển của tàu vũ trụ bằng cách tăng hoặc giảm giá trị tọa độ x mỗi khi phím được nhấn.

Với đoạn mã lệnh trên, khi kiểm tra **event.key** nếu có giá trị **pygame.K_LEFT** chương trình sẽ thực hiện việc giảm x của tàu, sau đó tiếp tục kiểm tra **event.key** có giá trị **pygame.K_RIGHT** hay không. Để tránh việc dư thừa như vậy, chúng ta sử dụng lệnh **elif** để kiểm tra các điều kiện theo sau một lệnh **if** đầu tiên (có thể lập trình nhiều lệnh **elif** theo sau một lệnh **if**), nếu điều kiện ở lệnh bên trên đúng thì sẽ không thực hiện kiểm tra những điều kiện bên dưới.

```
16      if event.type == pygame.KEYDOWN:
17        if event.key == pygame.K_LEFT:
18          ship_x -= 5
          # nếu điều kiện ở lệnh if đã đúng thì không thực hiện
          # kiểm tra ở lệnh elif theo sau nữa
19        elif event.key == pygame.K_RIGHT:
20          ship_x += 5
```

Với chương trình trên, chúng ta cần nhấn phím rất nhiều lần để tàu di chuyển. Có nhiều thuật toán để lập trình tàu di chuyển khi nhấn giữ phím và dừng khi nhả phím, dưới đây là một cách được sử dụng trong ví dụ này:

- Tạo một biến lưu giá trị là lượng x thay đổi của tàu (ví dụ: **x_change**) thay vì đặt cố định bằng 5 như trước
- Liên tục thay đổi tọa độ x của tàu theo biến **x_change** bằng lệnh **ship_x = x_change**. Với lệnh này, các bạn có thể thấy tàu sẽ:
  - Di chuyển sang phải nếu **x_change** có giá trị dương
  - Di chuyển sang trái nếu **x_change** có giá trị âm
  - Đứng im nếu **x_change** bằng 0
- Bắt sự kiện nhấn và nhả phím:
  - Nếu phím mũi tên phải được nhấn thì đặt **x_change** một giá trị dương (ví dụ 5)
  - Nếu phím mũi tên trái được nhấn thì đặt **x_change** một giá trị âm (ví dụ -5)
  - Nếu một phím bất kỳ được nhả thì đặt **x_change** bằng 0. Sự kiện một phím bất kỳ được nhả có giá trị **pygame.KEYUP**

```
11 x_change = 0     # tạo biến x_change
12 while True:
13   for event in pygame.event.get():
…       ...
17      if event.type == pygame.KEYDOWN:
18        if event.key == pygame.K_LEFT:
19          x_change = -5  # đặt x_change giá trị âm
20        elif event.key == pygame.K_RIGHT:
21          x_change = 5  # đặt x_change giá trị dương

              # kiểm tra sự kiện phím bất kỳ được nhả
22      if event.type == pygame.KEYUP:
```

```
23          x_change = 0  # đặt x_change giá trị 0

24      ship_x += x_change  # liên tục cập nhật tọa độ x của tàu
25      screen.blit(bg_image, (0, 0))
…       ...
```

Đến đây, các bạn có thể chạy thử chương trình và nhấn phím mũi tên trái/phải để điều khiển tên lửa. Các bạn có thể thay đổi tốc độ di chuyển theo ý thích của mình.

Tóm tắt lý thuyết và bài tập thực hành

Bài tập 1. Lập trình khi nhấn phím đi lên và xuống, tàu sẽ di chuyển lên và xuống tương ứng.

Bài tập 2. Lập trình khi nhấn phím cách, tàu sẽ xuất hiện tại vị trí (0, 0).

Bài tập 3. Lập trình khi nhấn phím A, D, S, W tàu sẽ di chuyển lần lượt sang trái, sang phải, xuống dưới và lên trên.