

BÀI 3 LẬP TRÌNH THIÊN THẠCH RƠI NGẪU NHIÊN

Trong bài học này, chúng ta sẽ lập trình để thiên thạch rơi liên tục từ phía trên màn hình xuống phía dưới ở các vị trí ngẫu nhiên khác nhau.

3.1. Thêm hình ảnh thiên thạch và lập trình thiên thạch rơi

Để thêm thiên thạch vào chương trình, chúng ta cũng cần các bước tải hình ảnh và vẽ hình ảnh giống với tàu vũ trụ. Các bạn tạo thêm các biến cần thiết để lưu hình ảnh và tọa độ thiên thạch, sau đó lập trình vẽ hình ảnh tại tọa độ tùy ý ở phía trên màn hình.

```
...
...
# bổ sung các biến lưu hình ảnh và tọa độ thiên thạch
11 meteor_img = pygame.image.load("resources/meteoroid.png")
12 meteor_x = 220
13 meteor_y = 0

14 x_change = 0
15 while True:
...
...
# bổ sung lệnh vẽ thiên thạch
30 screen.blit(meteor_img, (meteor_x, meteor_y))
31 pygame.display.update()
```

Thiên thạch sẽ rơi liên tục từ trên xuống dưới. Việc di chuyển theo chiều dọc từ trên xuống dưới tương đương với việc tăng tọa độ y liên tục. Các bạn bổ sung thêm lệnh **meteor_y += 2** ở trong lệnh lặp chính, khi chạy chương trình sẽ thấy thiên thạch rơi một lần từ trên xuống dưới.

```
27 ship_x += x_change
   # liên tục cập nhật tọa độ y của thiên thạch
28 meteor_y += 2
29 screen.blit(bg_image, (0, 0))
...
...
```

Để thiên thạch có thể rơi liên tục, chúng ta cần liên tục kiểm tra tọa độ y của thiên thạch, nếu lớn hơn chiều cao của cửa sổ (lớn hơn 360) thì sẽ được đặt lại giá trị bằng 0 ở phía trên.

```
...
...
28 meteor_y += 2
   # kiểm tra tọa độ y của thiên thạch
29 if meteor_y > 360:
30     meteor_y = 0
```



Khi chạy chương trình, ta thấy thiên thạch đã rơi liên tục tại cùng một vị trí. Chúng ta sẽ bổ sung thư viện **random** bằng lệnh **import random**, sau đó thiết lập tọa độ x nhận giá trị ngẫu nhiên từ 0 đến 480 bằng lệnh **random.randint(0, 480)**. Lúc này thiên thạch đã rơi được ở những vị trí ngẫu nhiên.

```
3 import random
...
16 while True:
...
29 if meteor_y > 360:
30     meteor_y = 0
# đặt meteor_x giá trị ngẫu nhiên từ 0 đến 480
31     meteor_x = random.randint(0, 480)
```

3.2. Lập trình chức năng tính điểm

Điểm là một đoạn văn bản được vẽ trên màn hình. Chúng ta cần tạo hai biến, một biến lưu giá trị điểm (ví dụ: **score**, dòng 16) và một biến có kiểu văn bản (ví dụ: **score_text**, dòng 17) có thể chuyển đổi thành ảnh và vẽ được trên màn hình. Biến có kiểu văn bản được khởi tạo bằng lệnh **pygame.font.SysFont("<tên font chữ>", <cỡ chữ>)**.

```
# tạo biến score để lưu điểm
16 score = 0
# tạo biến score_text là đối tượng văn bản có thể chuyển
# đổi thành ảnh và vẽ lên màn hình
17 score_text = pygame.font.SysFont("Arial", 24)
18 while True:
```

Trong trò chơi, điểm sẽ được tăng thêm 1 khi tàu vũ trụ tránh được 1 thiên thạch. Vậy chúng ta lập trình tăng điểm thêm 1 sau khi thiên thạch đã rơi hết màn hình và xuất hiện lại bên trên.

```
31 if meteor_y > 360:
31     meteor_y = 0
33     meteor_x = random.randint(0, 480)
# tăng điểm thêm 1
34     score += 1
```

Ta sử dụng lệnh **score_text.render(<đoạn văn bản cần vẽ>, True, <mã màu RGB>)** sử dụng để chuyển đổi biến có kiểu văn bản bên trên (biến **score_text**) trở thành thành ảnh bằng, qua đó ta có thể hiển thị lên màn hình. Trong đó:

- Tham số đầu tiên là đoạn văn bản cần chuyển, gồm chữ "Score: " ghép với biến **score** (cần chuyển **score** thành kiểu dữ liệu string) thành biểu thức **"Score: " + str(score)**.
- Tham số thứ 2 quy định có hiệu ứng khử răng cưa (antialias) hay không, giá trị **True** tương ứng với có hiệu ứng, **False** là ngược lại.