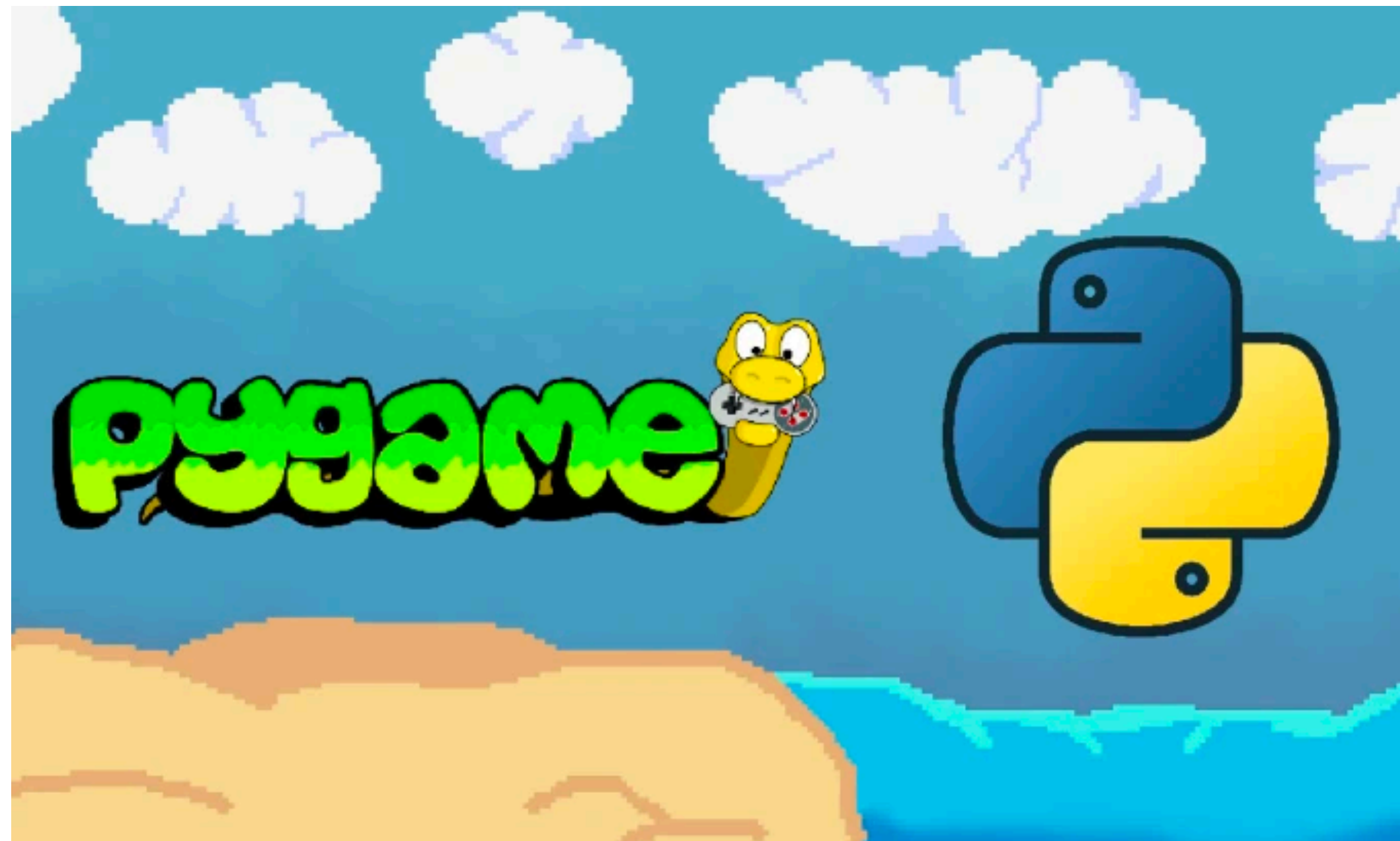


What is PyGame ?



- Pygame là một thư viện đa nền tảng được viết bằng ngôn ngữ lập trình Python, dùng để phát triển trò chơi và các ứng dụng đồ họa. Được phát hành dưới giấy phép LGPL, Pygame mở ra cơ hội cho cả những người mới bắt đầu và các nhà phát triển chuyên nghiệp để tạo ra trò chơi và ứng dụng đồ họa một cách nhanh chóng và hiệu quả
- Pygame thích hợp cho việc phát triển các trò chơi 2D và có thể được sử dụng để tạo ra một loạt các thể loại trò chơi, từ trò chơi giáo dục đơn giản đến trò chơi hành động phức tạp hơn. Nó cũng thường được sử dụng trong giáo dục để giảng dạy lập trình và khoa học máy tính.

Cài đặt Pygame !

```
pip install pygame
```

OR

```
pip3 install pygame
```

Kiểm tra Pygame !

```
pip list
```

```
macbookpro@MacBook-Pro-cua-Macbook-2 pygame % pip3 list
```

Package	Version
certifi	2023.7.22
charset-normalizer	3.2.0
customtkinter	5.2.1
darkdetect	0.8.0
idna	3.4
packaging	23.2
Pillow	10.1.0
pip	23.2.1
pygame	2.5.2
requests	2.31.0
setuptools	65.5.0
urllib3	2.0.5

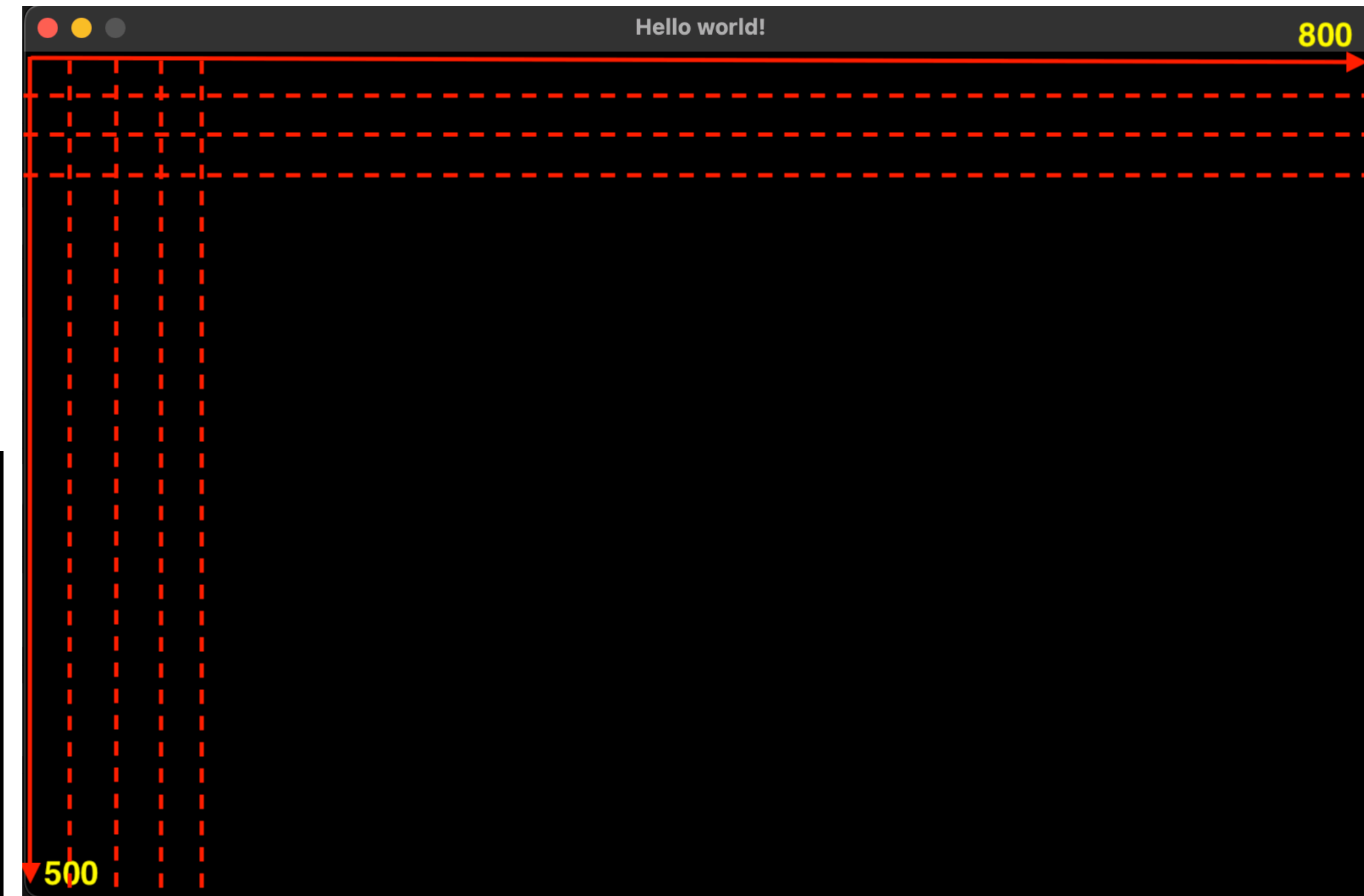
```
[notice] A new release of pip is available: 23.2.1 -> 23.3.2
[notice] To update, run: pip3 install --upgrade pip
macbookpro@MacBook-Pro-cua-Macbook-2 pygame %
```

Khởi tạo giao diện Pygame !

```
# import thư viện
import pygame, sys
from pygame.locals import *
```

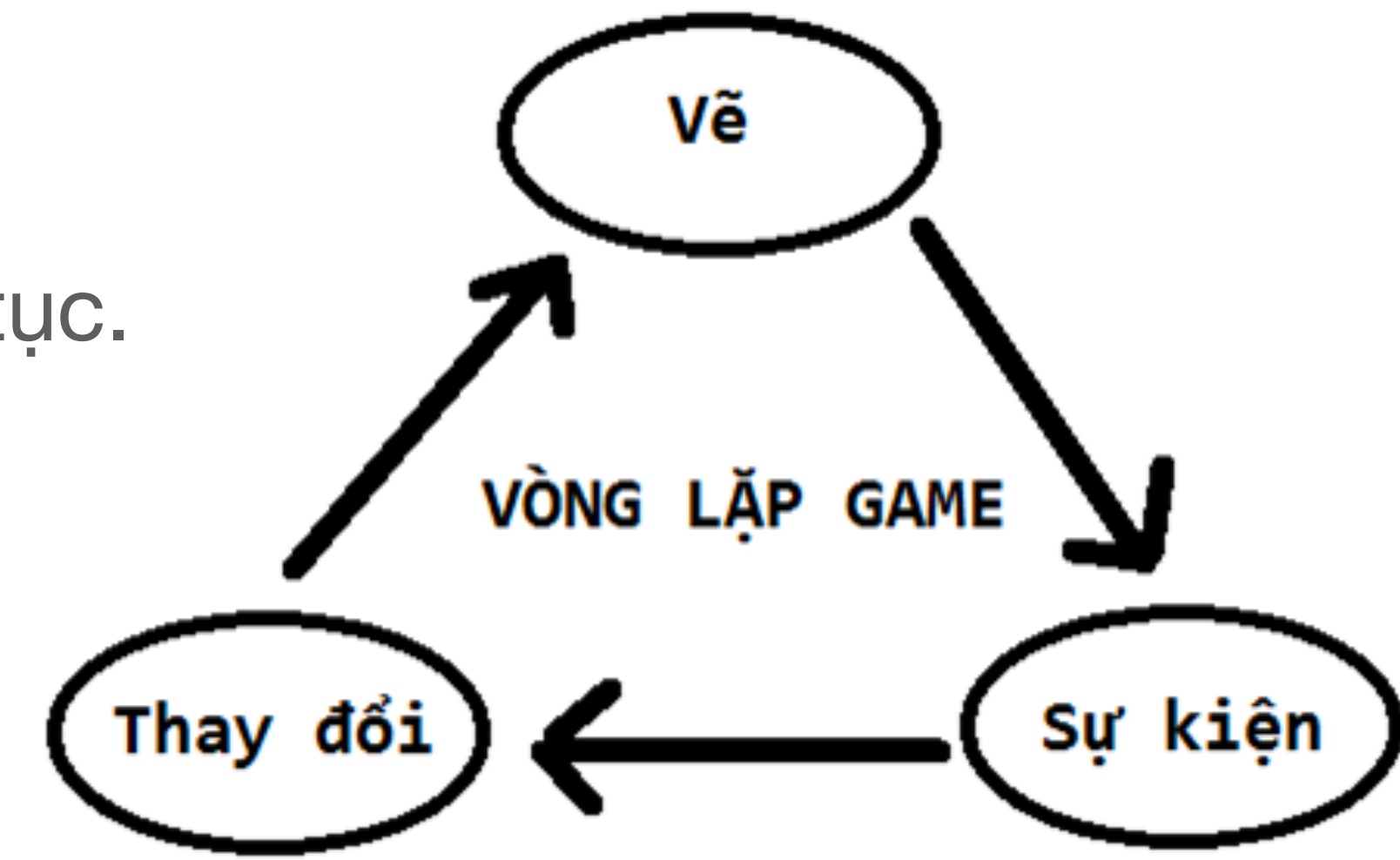
```
# khởi tạo game
pygame.init()
# gán Width và Height cho cửa sổ game. Đồng
thời gán biến khung cha
DISPLAYSURF = pygame.display.set_mode((800,
500))
```

```
# gán tiêu đề của sổ game
pygame.display.set_caption('Hello world!')
```



Khởi tạo giao diện Pygame !

=> Các hình ảnh trong game luôn luôn lặp lại để tạo các hành động liên tục. Giống như video là tập hợp của nhiều ảnh tạo thành, nên ta có cảm giác các hình ảnh chuyển động. Các trò chơi cũng thể hiện y như vậy. Được gọi là vòng lặp game



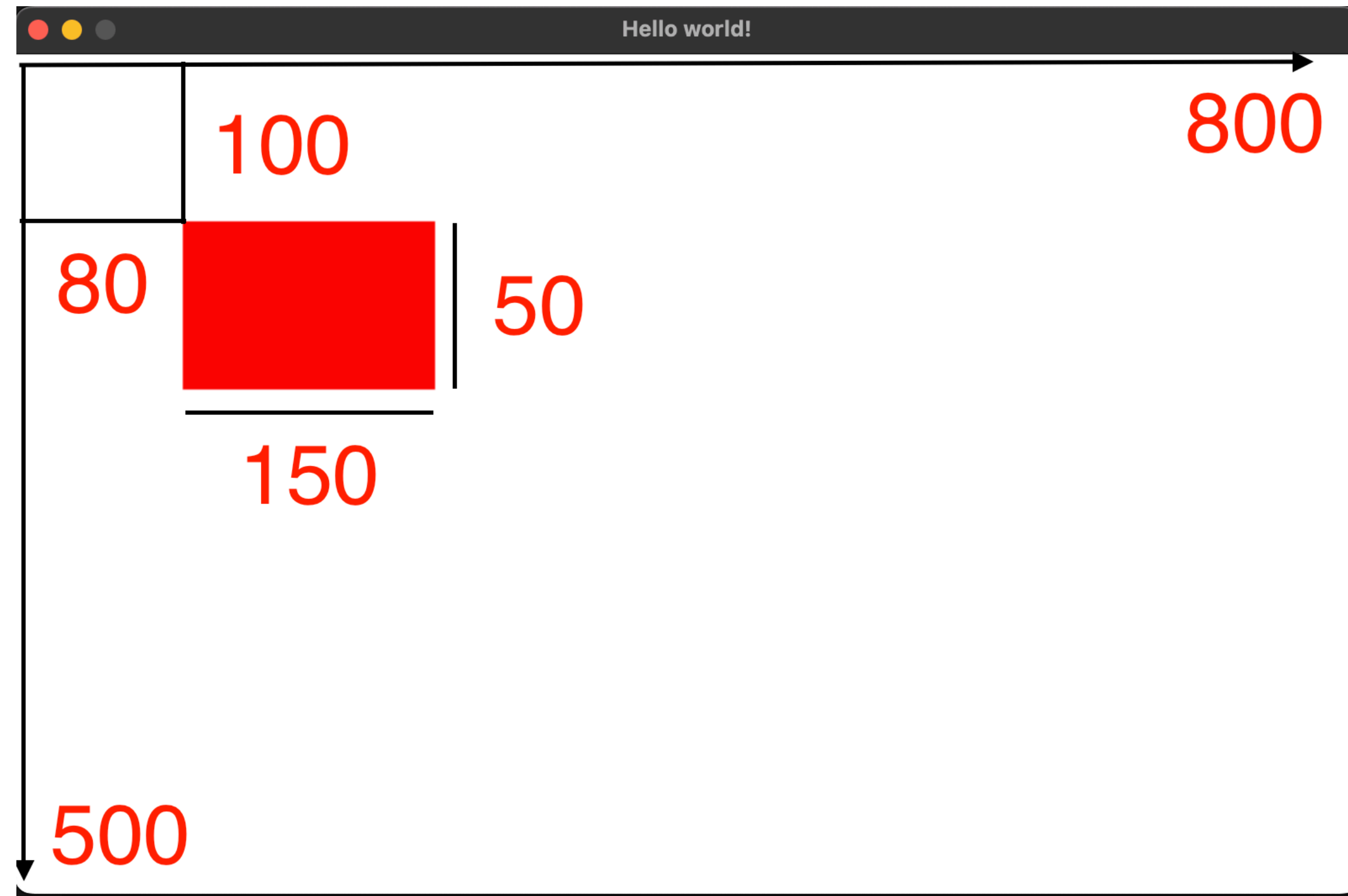
```
# vòng lặp game chính
while True:
    # vòng lặp khởi tạo các event sự kiện trong game: nhấn phím, chuột,...
    for event in pygame.event.get():
        # sự kiện khi người dùng nhấn dấu X để thoát game
        if event.type == QUIT:
            pygame.quit()

    # hàm cập nhật lại các trạng thái mới trong game (quan trọng)
    pygame.display.update()
    # tốc độ quét khung hình (FPS) khi chạy vòng lặp game
    pygame.time.Clock().tick(60)
```

Vẽ vài thứ lên Pygame !

```
# đổi màu nền game
DISPLAYSURF.fill((255, 255, 255))

# vẽ một hình tứ giác màu đỏ
pygame.draw.rect(DISPLAYSURF, (255, 0, 0), (100, 80, 150, 50))
```



Tìm hiểu Surface !

=> Surface giống như một frame khung chứa các thông tin của chính nó, hoặc có thể chứa một thông tin surface khác.

```
# Khởi tạo một surface quản lý riêng với width và height
rectSurface = pygame.Surface((150, 50))
# thêm màu cho surface
rectSurface.fill((255,0,0))
# Hàm blit dùng để vẽ 1 surface lên 1 surface khác. Cụ thể ở đây là vẽ surface
lên DISPLAYSURF, (x,y)
DISPLAYSURF.blit(rectSurface, (200, 50))
```

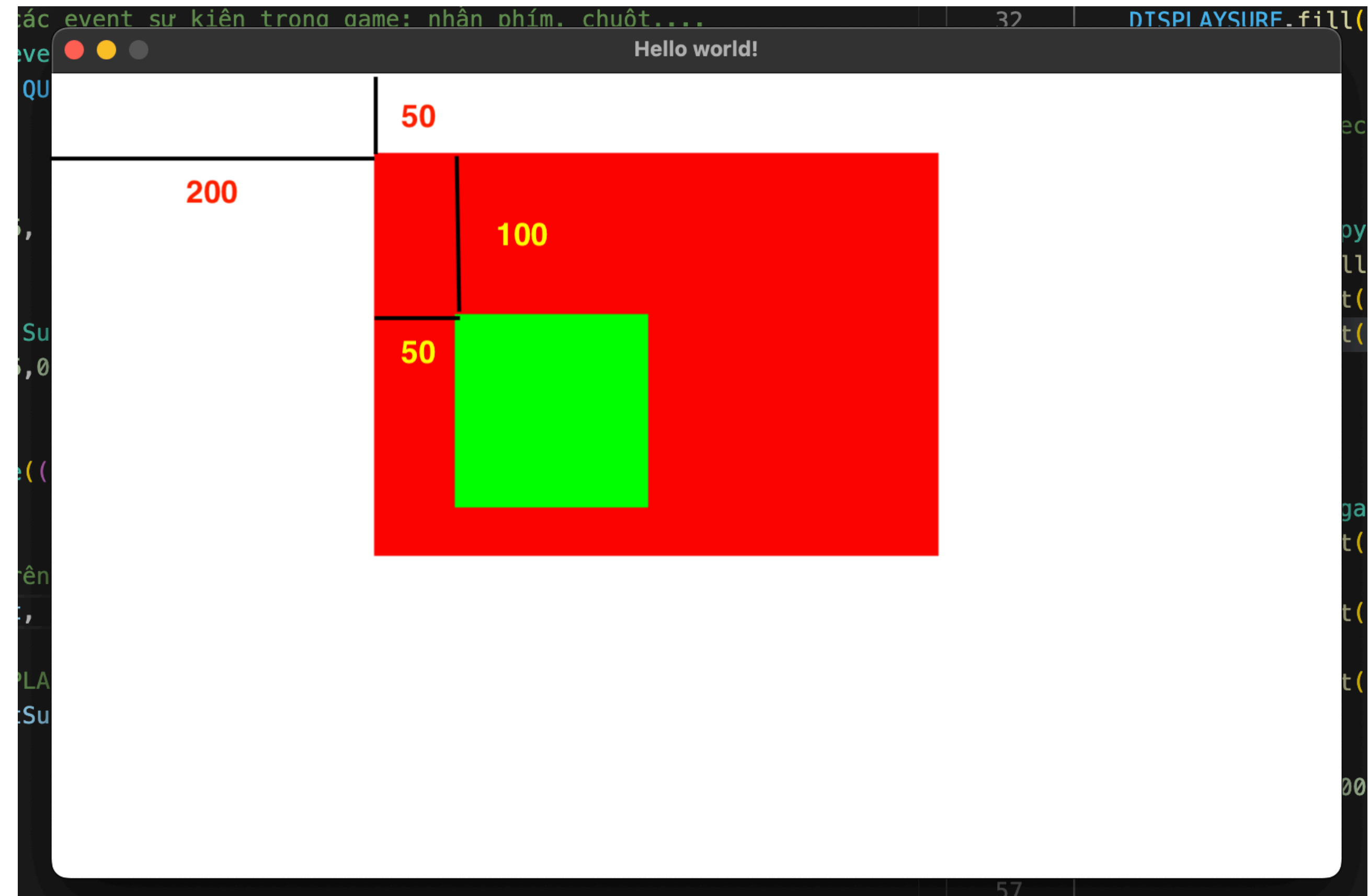

Tìm hiểu Surface !

```
# surface 1
rectSurface = pygame.Surface((350, 250))
rectSurface.fill((255,0,0))
```

```
# surface thứ 2
rect = pygame.Surface((120, 120))
rect.fill((0,255,0))
```

```
# vẽ surface 2 lên trên surface 1
rectSurface.blit(rect, (50,100))
```

```
# vẽ surface lên DISPLAYSURF, (x,y)
DISPLAYSURF.blit(rectSurface, (200, 50))
```



Animation !

```
x = 0

# vòng lặp game chính
while True:
```

```
# surface 1
surface1 = pygame.Surface((120, 120))
surface1.fill((255,0,0))
```

```
# surface 2
Surface2 = pygame.Surface((120, 120))
surface2.fill((0,255,0))
```

```
# tăng X lên mỗi lần lên 2
x+=2
```

```
# vẽ surface 1 và 2 lên DISPLAYSURF, (x,y)
DISPLAYSURF.blit(surface1, (300, 50))
```

```
# giá trị trục X sẽ được tăng dần khi vẽ lại => tạo nên chuyển động
DISPLAYSURF.blit(surface2, (x, 50))
```


Va chạm !

- Trong **Pygame**, hàm **get_rect()** là một phương thức được sử dụng với các đối tượng Surface để lấy một đối tượng Rect (hình chữ nhật) mô tả kích thước và vị trí của Surface.
- Rect là một cấu trúc dữ liệu quan trọng trong Pygame, được sử dụng rộng rãi để xử lý vị trí, kích thước, và va chạm của các đối tượng đồ họa.

```
# surface 1
surface1 = pygame.Surface((120, 120))
surface1.fill((255,0,0))

# surface 2
surface2 = pygame.Surface((120, 120))
surface2.fill((0,255,0))

# hàm get_rect()
rect1 = surface1.get_rect(topleft=(300, 50))
rect2 = surface2.get_rect(topleft=(0, 50))

# vòng lặp game chính
while True:
```

```
# tăng X lên mỗi lần lên 2
rect2.x+=2
# vẽ surface 1 và 2 lên DISPLAYSURF, (x,y)
DISPLAYSURF.blit(surface1, rect1)
# giá trị trục X sẽ được tăng dần khi vẽ lại => tạo nên chuyển động
DISPLAYSURF.blit(surface2, rect2)

if rect2.colliderect(rect1):
    surface2.fill((255,0,255))
else:
    surface2.fill((0,255,0))
```

Vẽ image & sprite image !

```
DISPLAYSURF.blit(pygame.image.load('kyo.png'), (0, 0))
```

```
surfaceIMG.blit(pygame.image.load('kyo.png'), (0, 0, 150, 105))
```

```
DISPLAYSURF.blit(surfaceIMG, (0, 0))
```

=> Trong lĩnh vực phát triển trò chơi, "sprite" là một thuật ngữ được sử dụng để mô tả một hình ảnh hoặc một nhóm hình ảnh được sử dụng để đại diện cho một đối tượng trong trò chơi, thường là nhân vật, quái vật, hoặc các đối tượng tương tác khác. Sprite thường được sử dụng trong các trò chơi 2D, nhưng cũng có thể xuất hiện trong một số trò chơi 3D, đặc biệt là trong các hiệu ứng hình ảnh



Go game !

