

코드쌤: 지혜야, 슬기야! 지난 시간에 우리 게임 무대에 알록달록 색깔도 칠하고, 여러 가지 도형도 그려봤지?

지혜: 네! 저는 빨간 동그라미랑 파란 네모를 엄청 많이 그려봤어요! 창이 짝 찼어요! 😊

슬기: 저는 거대한 노란색 태양을 그리고, 그 옆에 초록색 작은 외계인 로봇도 상상하면서 그려봤어요! 도형으로 만드니까 좀 각저 보였지만요. 🤖

코드쌤: 와, 정말 멋진 상상력인데! 슬기의 외계인 로봇처럼, 우리가 직접 그린 그림 말고, 이미 만들어진 멋진 캐릭터 그림을 게임에 등장시키고 싶을 때도 있겠지? 예를 들면... 귀여운 몬스터라든지!

지혜: 네! 포켓몬스터 같은 거요? 아니면 제가 그린 그림을 사진 찍어서 넣을 수도 있어요?

코드쌤: 물론이지! 오늘 배울 내용이 바로 그거야. **이미지 파일을 불러와서 우리 게임 화면에 “짠!” 하고 보여주는 방법**을 배울 거란다. 마치 스티커를 붙이듯이!

(코드 조각 1: 이미지 파일 준비 및 로딩)

```
import pygame

# Pygame 초기화
pygame.init()

# 창 크기 설정
WINDOW_WIDTH = 600
WINDOW_HEIGHT = 300 # 창 높이를 조금 줄여볼까?

# 디스플레이 표면 생성 및 창 제목 설정
window_surface = pygame.display.set_mode((WINDOW_WIDTH, WINDOW_HEIGHT))
pygame.display.set_caption("몬스터 등장! Blit 마법!")

# 배경 색상 설정 (하늘색 같은 청록색!)
window_surface.fill((0, 255, 255)) # Cyan

# 몬스터 이미지를 불러오자! (파일 이름이 'monster.png'이고, 'data' 폴더 안에 있다고 가정!)
monster_img = pygame.image.load("data/monster.png")
# print(type(monster_img)) # <class 'pygame.Surface'> 라고 나올 거야! 이미지도 하나의 '표면'이거든!

# 앗! 이미지를 불러왔는데, 어디에 그릴지 아직 안 정했네?
# 이미지의 '자리표'를 만들어주자!
monster_rect = monster_img.get_rect()
# print(monster_rect) # <rect(0, 0, 64, 64)> 이렇게 나올 수 있어. (이미지 크기에 따라 다름)
# 이건 "이미지의 왼쪽 위 모서리(topleft)가 (0,0)에 있고, 가로 64, 세로 64 크기야" 라는 뜻!

# 몬스터를 화면의 왼쪽 맨 위에 놓아볼까?
```

```
monster_rect.topleft = (0, 0)
```

```
# ... (게임 루프 코드는 아래에) ...
```

슬기: pygame.image.load("data/monster.png")요? 이건 “monster.png”라는 그림 파일을 “불러와!” 하는 주문인가요?

코드쌤: 맞아, 슬기! pygame.image.load() 함수는 마치 마법 주문 같아. 우리가 지정한 그림 파일을 컴퓨터 메모리로 “뽕!” 하고 불러와서, Pygame이 다룰 수 있는 이미지 데이터(우리는 monster_img라고 이름 붙였지?)로 만들어줘. 이 이미지 데이터도 사실은 Surface라는 특별한 종류의 ‘그림판’이야. 우리가 그림 그렸던 window_surface처럼!

지혜: monster_img.get_rect()는 뭐예요? rect는 사각형이라는 뜻이었는데... 몬스터 그림의 사각형을 가져온다는 건가?

코드쌤: 정확해, 지혜! 모든 이미지는 네모난 영역을 차지하잖아? get_rect()는 그 이미지의 “**겉 테두리 사각형 정보**”를 가져오는 거야. 이 사각형 정보(monster_rect)에는 이미지의 크기 (가로 몇 픽셀, 세로 몇 픽셀)와 화면 어디에 위치할지 (x좌표, y좌표) 정보가 담겨있어. 마치 이미지에게 이름표와 주소를 붙여주는 것과 같아!

슬기: monster_rect.topleft = (0, 0)은 그럼 그 몬스터 이름표에 “너는 화면 왼쪽 맨 위에 있어!” 라고 주소를 적어주는 거네요?

코드쌤: 빙고! topleft는 사각형의 ‘왼쪽 위 꼭짓점’을 의미해. 그래서 (0,0)으로 지정하면 이미지가 화면의 가장 왼쪽, 가장 위쪽에 놓이게 된단다. center, bottomright 등 다른 위치를 기준으로 정할 수도 있어!

코드쌤: 자, 이제 이미지를 불러왔고, 어디에 놓을지도 정했어. 그럼 이걸 어떻게 화면에 보여줄까? 바로 “**블릿(blit)**”이라는 마법을 사용할 거야!

(코드 조각 2: 이미지 화면에 그리기 - Blit 마법!)

```
# ... (이전 코드는 그대로) ...
```

```
# 게임 루프
```

```
running = True
```

```
while running:
```

```
    for event in pygame.event.get():
```

```
        if event.type == pygame.QUIT:
```

```
            running = False
```

```
    # 자, 이제 마법 주문을 외울 시간!
```

```
    # window_surface (어디에다가?) monster_img (무엇을?) monster_rect (어느 위치에?)
```

```
    # "도화지(window_surface)야, 몬스터 그림(monster_img)을 몬스터 자리표(monster_rect)에 그려줘!"
```

```
    window_surface.blit(monster_img, monster_rect)
```

```
    # 보너스! 화면에 가로선도 한번 그어볼까?
```

```

pygame.draw.line(window_surface, (255, 255, 255), (0, 75), (WINDOW_WIDTH, 75), 5) # 하얀색 선

# 더 신기한 거! 불러온 몬스터 이미지 위에도 직접 그림을 그릴 수 있어!
# monster_img (몬스터 그림 위에다가!) (255,255,255)(하얀색으로) (0,0)(시작점) (64,63)(끝점) 3(굵기)
# 몬스터 왼쪽 위에서 오른쪽 아래로 대각선! (몬스터 크기가 64x64라고 가정)
pygame.draw.line(monster_img, (255, 255, 255), (0, 0), (63, 63), 3) # 몬스터 이미지에 직접 선 그리기!
# 이 선은 몬스터 이미지 자체에 그려져
# 몬스터가 움직이면 선도 같이 움직여

# 디스플레이 업데이트 (이거 안하면 안보이는거 알지?)
pygame.display.update()

```

Pygame 종료

```
pygame.quit()
```

지혜: window_surface.blit(monster_img, monster_rect)요? “블릿”이라니, 뭔가 팡! 하고 나타날 것 같은데요!

코드쌤: 맞아! “Blit”은 “Block Transfer”의 줄임말로, 쉽게 말하면 “한 그림판(Surface)의 내용을 다른 그림판(Surface)으로 복사해서 붙여넣기!” 하는 강력한 마법이야. 여기서는 monster_img라는 몬스터 그림판의 내용을 window_surface라는 우리 게임 화면 그림판에, monster_rect가 알려주는 위치에 “착!” 하고 붙여넣는 거지. 마치 스티커를 딱 붙이는 것처럼!

슬기: 와! 진짜 몬스터 그림이 화면에 나왔어요! 신기하다! 그런데 선생님, pygame.draw.line(monster_img, ...) 이건 뭐예요? window_surface가 아니라 monster_img에 선을 그려요?

코드쌤: 예리한데, 슬기! 맞아. 아까 monster_img도 일종의 ‘그림판(Surface)’이라고 했지? 그래서 우리가 window_surface에 그림을 그릴 수 있었던 것처럼, 불러온 monster_img 위에도 직접 그림을 그릴 수 있어! 지금 코드에서는 몬스터 이미지 자체에 하얀색 대각선을 그린 거야. 그래서 blit으로 몬스터를 화면에 보여줄 때, 그 대각선도 함께 그려진 채로 나타나는 거지.

지혜: 아! 그럼 만약에 몬스터 이미지에 눈을 더 크게 그려주거나, 모자를 씌워줄 수도 있겠네요? 불러온 이미지에다가요!

코드쌤: 정확해! blit을 하기 전에 monster_img에 그림을 그리면, 그 변경된 모습이 화면에 나타나게 돼. 정말 재미있는 기능이지?

슬기: 선생님, 그럼 monster_rect.topleft = (100, 50) 이렇게 바꾸면 몬스터 위치도 바뀌나요?

코드쌤: 당연하지! monster_rect는 이미지의 위치를 담고 있으니까, 그 값을 바꾸고 다시 blit하면 몬스터가 새로운 위치에 그려질 거야. 이걸 게임 루프 안에서 조금씩 바꾸면 어떻게 될까?

지혜/슬기: 몬스터가 움직여요! / 날아다녀요!

코드쌤: 정답! 그게 바로 다음 시간에 우리가 탐험할 애니메이션의 세계란다! 오늘 우리는 이미지를 불러오고, get_rect()로 위치와 크기 정보를 얻고, blit 마법으로 화면에 보여주는 방법을 배웠어. 정말 중요한 기술들을

익힌 거야!

지혜: pygame.image.load()로 그림 부르고, get_rect()로 자리표 만들고, blit()으로 붙이고! 기억하세요!

슬기: 저도 이제 제가 좋아하는 캐릭터 그림 파일 찾아서 게임에 넣어볼래요! 완전 신나요!

코드쌤: 자, 슬기야, 지혜야! 방금 pygame.image.load("data/monster.png") 코드로 몬스터를 멋지게 불러왔지? 그런데 만약에... 컴퓨터가 "data/monster.png" 파일을 못 찾으면 어떻게 될까?

지혜: 음... 몬스터가 안 나타나겠죠? 아니면... 에러가 날까요?

코드쌤: 맞아, 지혜야! 한번 상상해 보자. 슬기가 신나서 자기가 제일 좋아하는 용감한 사자 캐릭터 그림(lion.png)을 다운받아서 바탕화면에 뒀어. 그리고 코드에 이렇게 적었지.

슬기의 (실수하는) 코드

```
brave_lion_img = pygame.image.load("lion.png")
```

슬기: (두근두근) 네! 그렇게 하고 실행 버튼을 딱! 눌렀는데... 어? 선생님! 빨간 글씨로 막 뭐라고 뜨면서 프로그램이 멈춰버렸어요! 🤖 FileNotFoundError: No such file or directory: 'lion.png' 라고 나왔어요! 이게 뭐예요? 몬스터는 잘 나왔는데 왜 사자는 안 나오는 거죠?

(상상 속 슬기의 컴퓨터 화면에 에러 메시지가 뜬다)

코드쌤: 아하! 슬기야, 그건 바로 컴퓨터가 “어흥! 사자 그림 어딴데? 아무리 찾아도 안 보이는데?” 하고 길을 잃었다는 신호야. FileNotFoundError는 말 그대로 “**파일을 찾을 수 없습니다!**” 라는 뜻이지.

지혜: 어? 바탕화면에 있다고 했잖아요! 컴퓨터가 바탕화면을 못 찾는 건가요?

코드쌤: 좋은 질문이야, 지혜야! 우리가 파이썬 코드를 실행하면, 파이썬은 기본적으로 **지금 실행하고 있는 .py 파일이 있는 바로 그 폴더 안에서** 파일을 찾기 시작해. 이걸 ‘**작업 폴더(Working Directory)**’라고 부른다. 마치 우리가 요리할 때, 지금 있는 부엌에서 재료를 찾는 것과 같아.

슬기: 아! 제 파이썬 코드 파일은 ‘내 게임’ 폴더에 있는데, lion.png 파일은 ‘바탕화면’ 폴더에 있으니까... 부엌(내 게임 폴더)에는 사자 그림(재료)이 없었던 거네요!

코드쌤: 정확해, 슬기! 컴퓨터는 우리가 특별히 “저쪽 방 냉장고에 있어!” 하고 알려주지 않으면, 지금 있는 부엌만 뒤져본단다. 그래서 lion.png 파일을 못 찾고 “없어!” 하고 소리친 거지.

지혜: 그럼 어떻게 해야 사자 그림을 찾을 수 있어요?

코드쌤: 두 가지 방법이 있어.

1. **가장 쉬운 방법: 파일을 같은 폴더로 옮기기!** 슬기가 다운받은 lion.png 파일을 **파이썬 코드 파일(my_game.py 같은)이 있는 바로 그 ‘내 게임’ 폴더 안으로 복사하거나 옮겨주면 돼.** 그럼 파이썬이 바로 옆에서 “아, 여기 있었네!” 하고 쉽게 찾을 수 있지.

(상상: 슬기가 바탕화면의 lion.png 파일을 ‘내 게임’ 폴더로 드래그 앤 드롭한다)

2. **조금 더 정확한 방법: 파일의 전체 주소(경로) 알려주기!** 만약 파일을 꼭 다른 폴더에 뒀야 한다면, 컴퓨터에게 “**야야, 그 파일은 ‘바탕화면’ 폴더 안에 있는 ‘lion.png’란다!**” 하고 아주 자세한 주소(경로)를 알려줘야 해. 예를 들면 이렇게 말이지. (컴퓨터마다 경로는 다를 수 있어!)

- 윈도우 예시: `pygame.image.load("C:/Users/슬기/Desktop/lion.png")`
- 맥 예시: `pygame.image.load("/Users/슬기/Desktop/lion.png")` 하지만 이건 좀 복잡하고, 나중에 폴더 위치가 바뀌면 또 수정해야 해서 처음에는 1번 방법이 훨씬 편하단다.

슬기: 아하! 그럼 아까 `monster.png`는 왜 잘 나왔어요? `pygame.image.load("data/monster.png")` 였는데!

코드쌤: 아주 좋은 관찰력이야, 슬기! 그건 우리가 이렇게 약속했기 때문이야. “파이썬 파일아, 네가 있는 곳에서 ‘data’라는 이름의 폴더를 먼저 찾아봐. 그리고 그 ‘data’ 폴더 안에 있는 ‘monster.png’를 찾아줘!” 라고 알려준 거지. / 기호가 폴더 안으로 들어가는 문이라고 생각하면 돼.

그래서 우리 프로젝트 폴더 구조가 이렇게 되어 있어야 해:

내_게임_프로젝트/ (여기가 작업 폴더!)

```

├── 나의_게임코드.py (우리가 지금 짜고 있는 파이썬 파일)
├── data/ (이미지들을 모아둘 'data' 라는 이름의 하위 폴더)
│   ├── monster.png (몬스터 이미지 파일!)
│   └── (다른 이미지들...)

```

지혜: 아! 그래서 `data/monster.png`라고 쓰면, 현재 폴더 안에 있는 data 폴더로 들어가서 `monster.png`를 찾는 거군요! 정리 정돈을 잘 해두는 거랑 비슷하네요!

코드쌤: 맞아! 이렇게 폴더를 만들어서 파일을 관리하면 나중에 게임이 커지고 파일이 많아져도 찾기 쉽고 깔끔하단다. 그래서 보통 이미지 파일, 소리 파일 등은 각각 `images`, `sounds` 또는 `assets`, `data` 같은 하위 폴더를 만들어서 그 안에 넣어두는 습관을 들이면 아주 좋아.

슬기: 이제 알겠어요! 파일은 꼭 파이썬 파일이 있는 곳(작업 폴더)이나, 거기서부터 시작하는 정확한 길(경로)을 알려줘야 하는군요! 아까 빨간 글씨 에러 나왔을 때 진짜 당황했는데, 이제 안 무서워요!

코드쌤: 그래, 슬기야! 에러 메시지는 무서운 게 아니라, 컴퓨터가 우리에게 “이것 좀 확인해 줄래?” 하고 보내는 중요한 힌트란다. `FileNotFoundError`가 뜨면 “아! 파일 위치가 잘못됐나 보다!” 하고 침착하게 확인하면 되는 거야.

지혜: 선생님, 오늘 또 중요한 거 배웠어요! 파일 경로랑 작업 폴더 개념, 그리고 에러 메시지와 친해지는 법까지요!

코드쌤: 아주 훌륭해! 이제 너희는 이미지 파일을 불러오다가 문제가 생겨도 스스로 해결할 수 있는 능력을 갖게 된 거야! 자, 그럼 아까 하던 blit 마법 이야기로 다시 돌아가 볼까? (웃음)