

코드쌤: 지혜야, 슬기야! 우리가 지금까지 게임 화면에 멋진 그림도 그리고, 글자도 쓰고, 신나는 소리랑 음악도 넣어봤지?

지혜: 네! 제 아기 용이 화면 중앙에 딱하니 나타나고, 배경음악도 웅장하게 흘러나왔어요! 🎵

슬기: 맞아요! 근데... 선생님, 제 아기 용이 가만히 있기만 하니깐 좀 심심해요. 제가 직접 키보드로 막 움직이게 하고 싶은데, 그건 어떻게 해요? 슈팅 게임처럼 막 피하고 공격하게요!

코드쌤: 오호! 드디어 게임의 핵심 재미 중 하나인 '조작'에 대한 궁금증이 생겼구나! 아주 좋아! 오늘은 바로 키보드의 방향키를 눌러서 우리 귀여운 드래곤을 화면 이곳저곳으로 움직이게 만드는 마법을 배울 거란다. 마치 드래곤 조종사가 되는 것처럼!

(코드 조각 1: 캐릭터 준비 및 초기 위치 설정 - 있는 코드 그대로!)

```
import pygame

pygame.init()

# 창의 너비와 높이
WINDOW_WIDTH = 600
WINDOW_HEIGHT = 300

# 드래곤의 이동 속도 (한 번 누를 때마다 얼마나 움직일까?)
VELOCITY = 30 # 원래 코드에서는 30이었네요! 10으로 수정하지 않을게요.

# 화면 표시를 위한 서피스 설정
display_surface = pygame.display.set_mode((WINDOW_WIDTH, WINDOW_HEIGHT))
pygame.display.set_caption("Keyboard moving") # 원래 코드의 캡션으로!

# 드래곤 이미지 로드
# (dragon_right.png 파일이 'data' 폴더 안에 있다고 가정해요!)
# 만약 파일이 없으면 여기서 에러가 날 수 있어요!
dragon_img = pygame.image.load("data/dragon_right.png")

# 드래곤 이미지의 사각형 정보를 가져옴
dragon_img_rec = dragon_img.get_rect() # 원래 변수명으로!

# 드래곤의 초기 위치를 창의 x축 중앙과 창의 하단으로 설정
dragon_img_rec.centerx = WINDOW_WIDTH // 2
dragon_img_rec.bottom = WINDOW_HEIGHT

# (게임 루프 코드는 원래 코드 그대로 이어집니다)
```

지혜: VELOCITY = 30이요? 이걸 '속도'라는 뜻인데, 드래곤이 한 번 움직일 때 30픽셀씩 움직인다는 건가요?

코드쌤: 정확해, 지혜! VELOCITY는 드래곤의 이동 속도를 정하는 변수야. 이 숫자를 크게 하면 드래곤이 훨씬 빠르게 움직이고, 작게 하면 조금씩 천천히 움직이겠지? 지금은 30으로 설정되어 있네.

슬기: dragon_img_rec.centerx = WINDOW_WIDTH // 2랑 dragon_img_rec.bottom = WINDOW_HEIGHT는요? 아! rect 객체에는 topleft 말고도 centerx (가운데 x좌표), bottom (아래쪽 y좌표) 같은 여러 가지 위치 속성이 있나 봐요! 이걸로 처음 시작 위치를 화면 아래쪽 가운데로 정한 거네요!

코드쌤: 똑똑한데, 슬기! 맞아. Rect 객체는 정말 다양한 방법으로 위치를 지정할 수 있어. centerx는 사각형의 가로 중심을, bottom은 사각형의 맨 아래쪽 가장자리의 y좌표를 의미해. 그래서 이렇게 설정하면 드래곤이 화면 바닥에 딱 발을 딛고 중앙에서 게임을 시작하게 된단다. 여기서도 dragon_right.png 파일이 data 폴더 안에 정확히 있어야 컴퓨터가 이미지를 잘 불러올 수 있어.

코드쌤: 자, 이제 드래곤도 준비됐고, 초기 위치도 정했어. 그럼 어떻게 키보드 입력을 받아서 드래곤을 움직일까? 바로 **이벤트 루프** 안에서 키보드 이벤트를 확인하는 거야!

(코드 조각 2: 키보드 입력(KEYDOWN) 이벤트 처리 - 있는 코드 그대로!)

```
# ... (이전 코드는 그대로) ...
```

```
running = True
while running:
    for event in pygame.event.get(): # 어떤 사건이 일어났나?
        if event.type == pygame.QUIT: # 창 닫기 버튼?
            running = False

    # --- 여기가 오늘의 핵심! 키보드 눌림 감지! ---
    # 만약 발생한 이벤트의 종류가 "키보드 키가 눌렸다!" (KEYDOWN) 라면?
    if event.type == pygame.KEYDOWN:
        # print(f"키 눌림! 어떤 키? {event.key}") # 주석 처리된 print는 생략할게요.

        # 방향키가 눌렸는지 확인하고 드래곤을 이동시킴
        if event.key == pygame.K_LEFT:
            dragon_img_rec.x -= VELOCITY
        elif event.key == pygame.K_RIGHT:
            dragon_img_rec.x += VELOCITY
        elif event.key == pygame.K_UP:
            dragon_img_rec.y -= VELOCITY
        elif event.key == pygame.K_DOWN:
            dragon_img_rec.y += VELOCITY

    # 서피스를 검은색으로 채움
```

```
display_surface.fill((0, 0, 0))

# 드래곤 이미지를 현재 위치에 그림
display_surface.blit(dragon_img, dragon_img_rec)

# 화면 업데이트
pygame.display.update()
```

게임 종료 및 창 닫기

```
pygame.quit()
```

지혜: event.type == pygame.KEYDOWN이요! 지난번에 pygame.QUIT은 창 닫기 이벤트였는데, KEYDOWN은 “키보드의 어떤 키든 눌렀다!”라는 신호군요!

코드쌤: 맞아! KEYDOWN 이벤트가 발생하면, 우리는 event.key라는 속성을 통해 “정확히 어떤 키가 눌렸는지” 알 수 있어. Pygame은 각 키마다 고유한 이름표(상수)를 가지고 있는데, 예를 들어 왼쪽 방향키는 pygame.K_LEFT, 오른쪽 방향키는 pygame.K_RIGHT 이런 식이야. K는 'Key'의 약자지.

슬기: 와! 그래서 if event.key == pygame.K_LEFT: 이렇게 해서 “만약 눌린 키가 왼쪽 방향키라면?” 하고 확인하는 거네요! 그리고 dragon_img_rec.x -= VELOCITY는 드래곤의 x좌표를 VELOCITY만큼 빼서 왼쪽으로 움직이게 하는 거고요!

코드쌤: 정확해!

- 왼쪽으로 가려면 x좌표를 줄이고 (-=)
- 오른쪽으로 가려면 x좌표를 늘리고 (+=)
- 위로 가려면 y좌표를 줄이고 (-=) - Pygame에서는 화면 위쪽이 y좌표 0, 아래로 갈수록 y값이 커져!
- 아래로 가려면 y좌표를 늘리는 (+=) 거지.

지혜: 아! y좌표는 아래로 갈수록 커지는군요! 수학시간에 배운 좌표랑은 위아래가 반대네요! 신기하다!

코드쌤: 맞아. 컴퓨터 그래픽에서는 보통 화면 왼쪽 위를 (0,0)으로 기준 삼아서 그렇게 된단다. 자, 이제 이 코드를 실행하고 방향키를 눌러보면 어떻게 될까?

(상상: 지혜와 슬기가 신나게 방향키를 누르자 화면 속 드래곤이 움직인다!)

슬기: 와! 움직여요! 드래곤이 제 손가락 따라서 막 움직여요! 왼쪽, 오른쪽, 위, 아래! 너무 신기해요! 제가 진짜 드래곤 조종사가 된 것 같아요! 😄

지혜: 선생님, 근데 드래곤이 화면 밖으로 나가버리기도 하는데요? 벽에 부딪혀서 멈추게 할 수는 없나요?

코드쌤: 오, 아주 좋은 질문이야, 지혜! 지금은 드래곤이 화면 경계를 신경 쓰지 않고 그냥 뚫고 나가버리지? 다음 시간에는 바로 그 **화면 경계 처리**와, 또 다른 중요한 키보드 이벤트인 **KEYUP(키에서 손을 뗄 때)** 이벤트에 대해서도 배워볼 거야. 그러면 더 섬세한 조작이 가능해지겠지?

슬기: 네! 드래곤이 화면 밖으로 도망가지 않게 하고 싶어요! 그리고 스페이스바 누르면 불 뿜는 것도 만들 수 있어요? 🔥

코드샐: 하하, 물론이지! 스페이스바도 `pygame.K_SPACE`라는 이름표가 있단다. 오늘 배운 KEYDOWN 이벤트를 활용하면 얼마든지 만들 수 있어! 너희들의 상상력이 게임을 더욱 재미있게 만드는 거야!

지혜: 키보드 입력으로 캐릭터를 움직이니깐 진짜 게임하는 느낌이 확 살아요! KEYDOWN 이벤트랑 `event.key`, 그리고 Rect 객체의 좌표를 변경하는 방법! 정말 중요한 걸 배웠어요!

코드샐: 훌륭해! 오늘 너희는 플레이어와 게임이 직접 소통하는 가장 기본적인 방법을 익힌 거야. 이제 이 원리를 응용하면 점프도 시키고, 공격도 하게 만들 수 있겠지? 다음 시간에 더 재미있는 조작의 세계로 떠나보자!