

코드쌤: 지혜야, 슬기야! 지난 시간에는 키보드 방향키로 드래곤을 멋지게 조종했었지?

지혜: 네! 제가 누르는 대로 드래곤이 쓱쓱 움직여서 정말 신났어요! 🐉🖱️

슬기: 맞아요! 근데 선생님, 게임 중에는 마우스로 뭔가를 클릭하거나 드래그해서 옮기는 것도 많잖아요. 그것도 할 수 있어요?

코드쌤: 오호! 아주 좋은 질문이야, 슬기! 오늘은 바로 그 **마우스의 움직임과 클릭을 감지해서 드래곤을 조종하는 방법**을 배울 거란다. 그리고 마우스 이벤트를 자세히 들여다보면 아주 재미있는 '정보 꾸러미'를 발견할 수 있는데, 그 비밀을 푸는 열쇠가 바로 '딕셔너리'라는 신기한 보물 상자야!

(코드 조각 1: 마우스 이벤트 감지 준비)

```
import pygame

pygame.init()

WINDOW_WIDTH = 600
WINDOW_HEIGHT = 300 # 오타 수정! WINDOW_HEIGHT 여야겠죠? (설명은 WINDOW_HEIGHT로 할게요)

display_surface = pygame.display.set_mode((WINDOW_WIDTH, WINDOW_HEIGHT))
pygame.display.set_caption("마우스로 드래곤을 움직여볼까?") # 캡션 변경!

dragon_img = pygame.image.load("data/dragon_right.png")
dragon_rect = dragon_img.get_rect()
dragon_rect.topleft = (25, 25) # 드래곤 처음 위치는 왼쪽 위!

running = True
while running:
    for event in pygame.event.get():
        print(event) # <-- 어떤 이벤트가 발생하는지 여기서 다 보여줘! (중요!)
        if event.type == pygame.QUIT:
            running = False

    # --- 여기가 오늘의 핵심! 마우스 이벤트 감지! ---

    # 1. 마우스 버튼이 "딸깍!" 눌렀을 때 (MOUSEBUTTONDOWN)
    if event.type == pygame.MOUSEBUTTONDOWN:
        # print(event) # 여기서 이벤트 정보를 자세히 볼 수 있지!
        mouse_x = event.pos[0] # 클릭한 x좌표
        mouse_y = event.pos[1] # 클릭한 y좌표
        dragon_rect.centerx = mouse_x # 드래곤 중심을 마우스 x좌표로!
```

```

dragon_rect.centery = mouse_y # 드래곤 중심을 마우스 y좌표로!

# 2. 마우스 버튼을 누른 채로 "속~" 움직일 때 (MOUSEMOTION)
# event.buttons[0] == 1 은 "왼쪽 버튼이 눌린 상태인가?" 를 확인하는 거야!
if event.type == pygame.MOUSEMOTION and event.buttons[0] == 1:
    # print(event) # 여기서도 이벤트 정보를 자세히!
    mouse_x = event.pos[0] # 현재 마우스 x좌표
    mouse_y = event.pos[1] # 현재 마우스 y좌표
    dragon_rect.centerx = mouse_x
    dragon_rect.centery = mouse_y

display_surface.fill((0, 0, 0))
display_surface.blit(dragon_img, dragon_rect)

pygame.display.update()

pygame.quit()

```

지혜: WONDOW_HEIGHT는 WINDOW_HEIGHT 오타인 것 같아요! 😊 그리고 print(event)가 맨 처음에 있네요! 이걸로 마우스 움직일 때 어떤 이벤트가 나오는지 다 볼 수 있겠어요!

코드쌤: 오, 예리한 지혜! 오타도 발견하고, print(event)의 역할도 정확히 파악했네! 맞아. 저 print(event)는 우리가 마우스를 움직이거나 클릭할 때마다 “지금 이런 일이 일어났어요!” 하고 컴퓨터가 우리에게 모든 정보를 보여주는 아주 중요한 창문이야.

(상상: 코드쌤이 마우스를 클릭하고 움직이자 터미널에 이벤트 정보가 차르륵 출력된다) <Event(5-MouseButtonDown {'button': 1, 'pos': (123, 456)})> <Event(4-MouseMotion {'pos': (234, 567), 'rel': (10, -5), 'buttons': (0, 0, 0)})> <Event(6-MouseButtonUp {'button': 1, 'pos': (789, 321)})>

슬기: 와! print(event) 하니까 진짜 뭔가 많이 나와요! <Event(5-MouseButtonDown ... 이런 거요! 저 중괄호 {} 안에 있는 건 뭐예요? button: 1, pos: (123, 456) 이렇게요!

코드쌤: 아주 좋은 질문이야, 슬기! 저 중괄호 {}로 묶인 부분이 바로 오늘 우리가 탐험할 ‘딕셔너리(Dictionary)’라는 보물 상자란다! 딕셔너리는 마치 **이름표(Key)**가 붙은 **물건(Value)**들을 넣어두는 상자와 같아.

코드쌤: (딕셔너리 보물 상자 그림을 보여주며) “자, 이 상자에는 'button'이라는 이름표가 붙은 물건이 있고, 그 물건은 숫자 1이야. 또 'pos'라는 이름표가 붙은 물건이 있고, 그건 (123, 456)이라는 좌표 꾸러미지!”

지혜: 아! 그럼 MouseButtonDown 이벤트가 발생하면, 그 이벤트 정보 안에는 'button'이라는 이름표로 “어떤 버튼이 눌렸는지” (1은 왼쪽 버튼!), 'pos'라는 이름표로 “어디를 클릭했는지” 좌표값이 들어있는 거네요!

코드쌤: 정확해! 딕셔너리는 이렇게 **이름표(Key)**와 **실제 값(Value)**을 한 쌍으로 묶어서 정보를 저장해. 그래서 우리가 “이름표 'pos'에 뭐가 들어있어?” 하고 물어보면 좌표값을 꺼내 쓸 수 있는 거야.

예시: 딕셔너리 사용법 (파이썬 셸에서 간단히 보여주는 상상)

이벤트 정보를 가짜로 만들어볼게!

```
event_info = {'button': 1, 'pos': (100, 200), 'window': None} # 이런 딕셔너리가 있다고 치자!
```

'pos' 이름표에 해당하는 값을 꺼내려면?

```
클릭_좌표 = event_info['pos']
```

```
print(클릭_좌표) # (100, 200) 이렇게 나오겠지?
```

클릭_좌표는 (100, 200) 이라는 튜플이니까, x좌표만 꺼내려면?

```
클릭_x = 클릭_좌표[0] # 튜플의 0번째 값!
```

```
print(클릭_x) # 100
```

코드샐: 우리 코드에서 event.pos가 바로 저 딕셔너리에서 'pos'라는 이름표로 값을 꺼내온 거야. 그리고 event.pos[0]은 그 값 (좌표 튜플)에서 첫 번째 항목(x좌표)을, event.pos[1]은 두 번째 항목(y좌표)을 꺼내서 mouse_x, mouse_y 변수에 저장한 거지.

슬기: 와! 딕셔너리 신기하다! 그럼 my_dict = {"name": "왕새우", "score": (100, 95, 80)} 이것도 딕셔너리예요? "name"이라는 이름표에는 "왕새우"가, "score"라는 이름표에는 (100, 95, 80) 점수 꾸러미가 들어있는 거네요!

코드샐: 맞아! 슬기가 완벽하게 이해했네! my_dict["name"] 하면 "왕새우"가 나오고, my_dict["score"][0] 하면 100점이 나오겠지? 딕셔너리는 이렇게 관련된 정보들을 이름표를 붙여서 깔끔하게 정리할 수 있어서 아주 유용하단다. 그리고 my_dict["job"] = "선생님"처럼 새로운 이름표와 값을 추가할 수도 있어!

지혜: 그럼 다시 우리 코드로 돌아와서, if event.type == pygame.MOUSEBUTTONDOWN: 이 부분은 마우스 버튼을 "딸깍" 눌렀을 때 실행되는 거죠? 그때 마우스 커서가 있던 위치의 x, y 좌표를 가져와서 드래곤의 중심(centerx, centery)을 그 위치로 옮기는 거네요!

코드샐: 정확해! 그래서 마우스를 클릭하면 드래곤이 "뿡!" 하고 그 위치로 순간이동하는 것처럼 보일 거야. dragon_rect.x나 dragon_rect.y로 위치를 바꾸는 것과 dragon_rect.centerx, dragon_rect.centery로 바꾸는 건 드래곤의 어느 부분을 기준으로 위치를 맞추는지 차이가 있어. centerx, centery는 이미지의 정중앙을 기준으로 하니 더 자연스럽게 보이겠지?

슬기: 그럼 if event.type == pygame.MOUSEMOTION and event.buttons[0] == 1: 이건 뭐예요? MOUSEMOTION은 마우스가 움직일 때인데, event.buttons[0] == 1은 또 뭐죠?

코드샐: 좋은 질문이야! MOUSEMOTION 이벤트 정보 안에는 buttons라는 이름표로 현재 마우스 버튼들의 상태가 (왼쪽, 가운데, 오른쪽) 순서로 들어있어. (0, 0, 0)이면 아무것도 안 눌린 상태, (1, 0, 0)이면 왼쪽 버튼만 눌린 상태를 의미해. 그래서 event.buttons[0] == 1은 "왼쪽 마우스 버튼이 눌린 채로 마우스가 움직였니?" 라고 물어보는 거란다.

지혜: 아! 그럼 이건 마우스 왼쪽 버튼을 누른 상태로 드래그하는 거네요! 그때마다 마우스 위치를 가져와서 드래곤의 중심을 계속 업데이트하니까, 드래곤이 마우스 따라서 쭉~ 끌려오겠어요!

코드샐: 맞아! 바로 '드래그 앤 드롭' 기능의 핵심 원리 중 하나지! 오늘 우리는 마우스 이벤트를 감지하고, 그

안에 담긴 중요한 정보(딕셔너리!)를 꺼내 쓰는 방법을 배웠어. 그리고 그 정보를 이용해서 드래곤을 마우스로 조종하는 마법까지 익혔네!

슬기: 딕셔너리라는 보물 상자, 정말 신기하고 유용해요! 마우스 이벤트 정보도 이제 무섭지 않아요! pos 이름표만 찾으면 좌표를 알 수 있으니까요!

지혜: 키보드 조작이랑 마우스 조작을 둘 다 할 수 있게 되니까 게임이 훨씬 더 풍부해질 것 같아요! 다음엔 뭘 배우나요, 선생님?

코드쌤: 하하, 너희들의 호기심은 정말 끝이 없구나! 다음 시간에는 오늘 배운 마우스 조작을 좀 더 다듬고, 어쩌면 드래곤이 화면 밖으로 나가지 않도록 하는 방법도 다시 한번 생각해 볼 수 있겠지? 기대해도 좋아!