

```

is_jumping = False
jump_count = 10
# 점프 키가 눌렸을 때
if keys[pygame.K_SPACE] and not is_jumping:
    is_jumping = True
    jump_count = 10
# 플레이어 점프 처리
if is_jumping:
    if jump_count >= -10:
        neg = 1
        if jump_count < 0:
            neg = -1
        character_y -= (jump_count ** 2) * 0.5 * neg
        jump_count -= 1
    else:
        is_jumping = False

```

#### 1. 플레이어 점프 관련 변수 초기화:

- is\_jumping 변수를 False로 설정하여 플레이어가 점프 중인지 여부를 추적합니다.
- jump\_count 변수를 10으로 설정하여 점프 동작의 높이를 조절합니다.

#### 2. 점프 키 입력 확인:

- 키보드 입력을 확인하여 스페이스바가 눌렸는지 확인합니다.
- keys[pygame.K\_SPACE]를 사용하여 스페이스바 키 입력 여부를 확인합니다.
- not is\_jumping을 추가하여 이미 점프 중인 상태에서는 추가적인 점프 입력을 무시합니다.

#### 3. 점프 동작 활성화:

- 점프 키가 눌렸고, 이미 점프 중이지 않은 상태일 때, 점프 동작을 활성화합니다.
- is\_jumping 변수를 True로 설정하여 플레이어가 점프 중임을 나타냅니다.
- jump\_count를 10으로 설정하여 점프의 최대 높이를 지정합니다.

#### 4. 플레이어 점프 처리:

- is\_jumping 변수가 True인 동안 점프 동작을 처리합니다.
- jump\_count가 -10보다 크거나 같은 동안 점프 동작을 수행합니다.
- neg 변수를 1로 초기화하여 플레이어가 상승할 때의 방향을 지정합니다.
- jump\_count가 0보다 작을 경우 neg 변수를 -1로 설정하여 플레이어를 하강시킵니다.
- 플레이어의 Y 좌표인 character\_y를  $(\text{jump\_count}^2) * 0.5 * \text{neg}$  만큼 변경하여 점프 동작을 구현합니다.
- jump\_count를 1씩 감소시켜 플레이어의 점프 높이를 조절합니다.

#### 5. 점프 완료 처리:

- jump\_count가 -10보다 작은 경우, 즉, 점프 동작이 완료된 경우, is\_jumping 변수를 False로 설정하여 점프 동작을 종료합니다.
- jump\_count를 다시 10으로 초기화하여 다음 점프를 준비합니다.