

14

파이썬으로 게임 만들기 프로젝트1



염 희 균

목차

- 환경 설정 및 라이브러리 설치(pygame)
 - pip install pygame

1. 게임 뼈대 만들기 (1_create_frame.py)

- 객체 지향 중심의 게임 프로그램을 만들기 위한 큰 뼈대 작성

1. `import pygame`

2. 화면 크기 설정

- 가로: 480, 세로 :640

3. 화면 타이틀 설정

- `pygame.display.set_caption()`함수사용

4. 창이 꺼지지 않도록 이벤트 루프 대기 (반복문 사용)

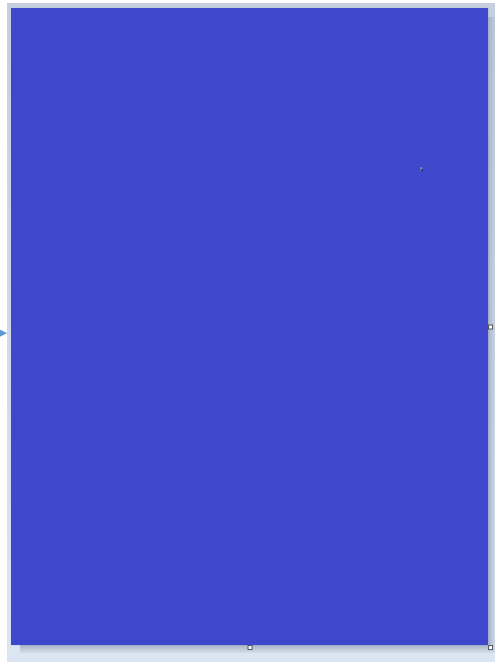
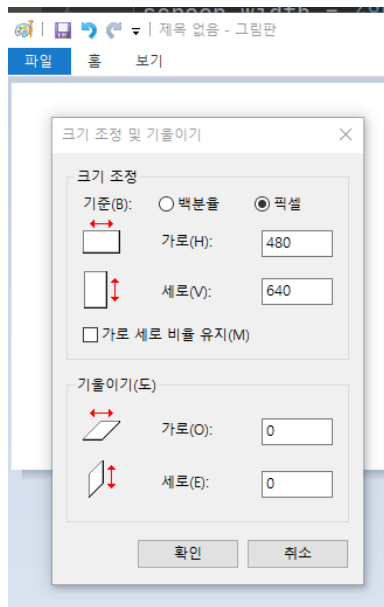
- `running = True` # 게임 진행중인 상태
- `while running :`

5. Pygame 종료

- `pygame.quit()`

2. 배경설정 (2_background.py)

- 그림판으로 배경 이미지 만들기
 - 픽셀-가로세로(480*640)-가로세로 비율유지 해지
 - 파랑으로 채우기
 - pygame_basic - background.png 로저장



2-1. 이미지 불러오기

- `background = pygame.image.load()`
- `load()`에서 이미지 경로에서 이미지 불러오기
 - 이미지 더블클릭 후 이미지이름 탭에서 우 클릭 - copypath 후 코드에 붙여 넣기
 - ₩이를 그냥 /(슬래시)로 변경하기

```
# 배경이미지 불러오기
```

```
background = pygame.image.load("C:/Users/HeeGyun/PycharmProjects/HelloPycharm/pygame_basic/background.png")
```

2-2. 배경 이미지 그리기

1. 게임이 진행중이면 배경 그리기

- `screen.blit(background, (x좌표, y좌표))`

2. 게임 화면 다시 그리기

- `pygame.display.update()`

3. 창닫기 이벤트 처리

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

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

```
        running = False
```

3. 캐릭터 설정 (3_main_sprite.py)

- 그림판으로 캐릭터 이미지 만들기
 - 픽셀-가로세로(70*70)-가로세로 비율유지 해지
 - 빨강으로 채우기
 - pygame_basic - character.png 로저장



3-1. 캐릭터 불러오기

- `character = pygame.image.load()`
- `load()`에서 이미지 경로에서 이미지 불러오기
 - 이미지 더블클릭 후 이미지이름 탭에서 우클릭 - `copypath` 후 코드에 붙여넣기
 - ₩이를 그냥 /(슬래시)로 변경하기

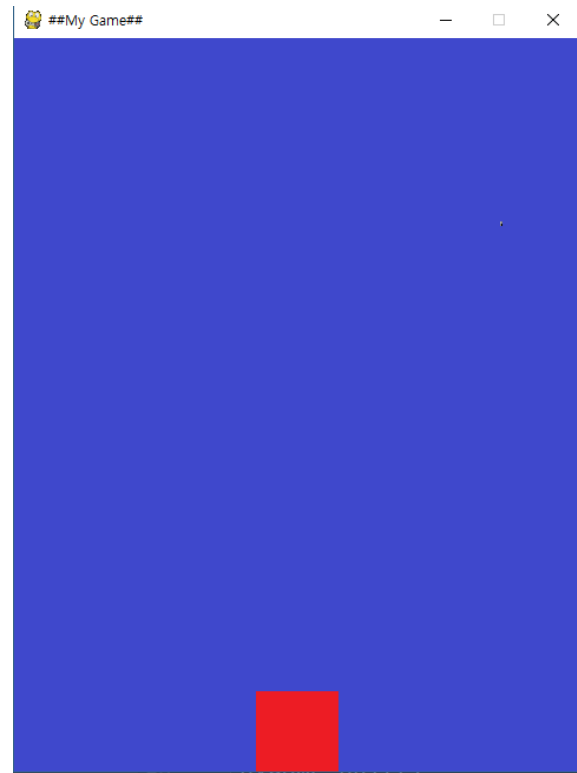
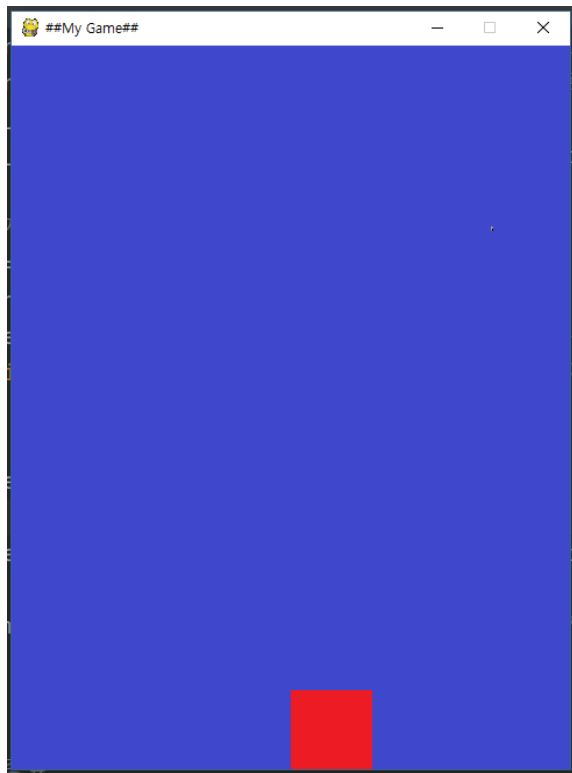
```
# 스프라이트(캐릭터) 불러오기
character = pygame.image.load("C:/Users/HeeGyun/PycharmProjects/HelloPycharm/pygame_basic/character.png")
```


3-2. 캐릭터 크기 설정

- `character_size=character.get_rect().size` # 이미지 전체 크기를 구해 옴
- `character_width = character_size[0]`
- `character_height = character_size[1]`

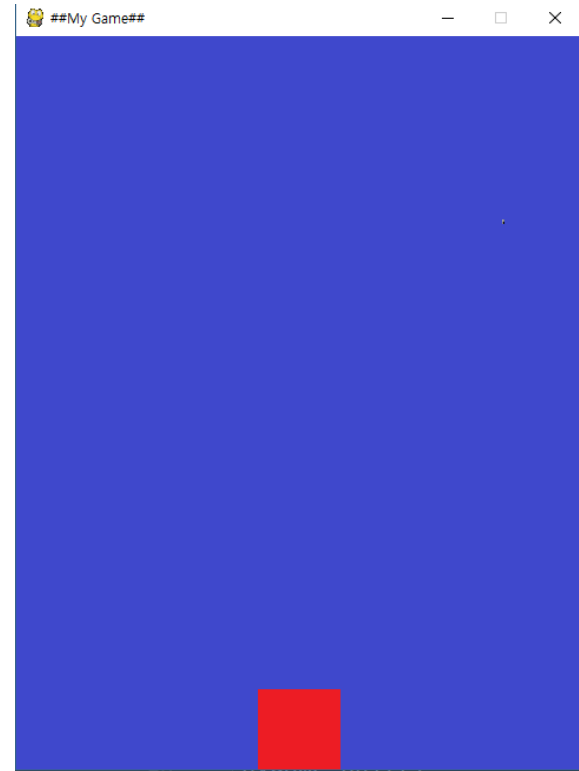
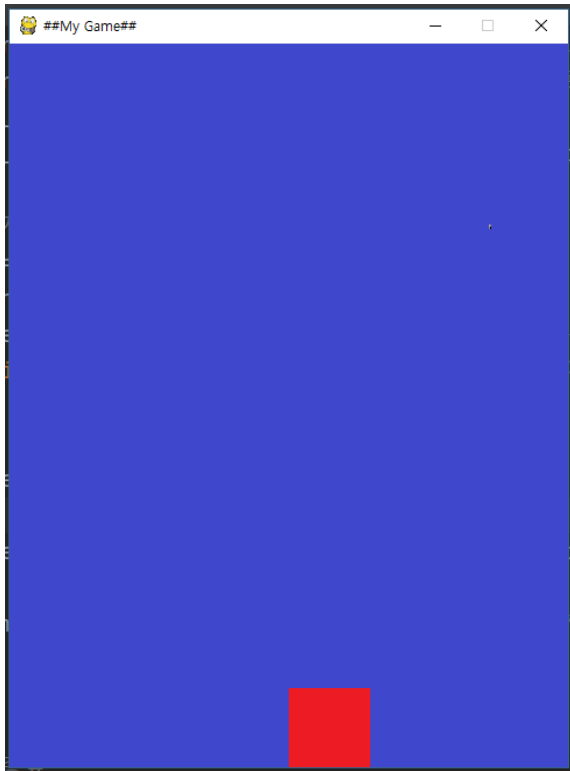
3-3. 캐릭터 위치 설정

- 세로 위치는 화면의 세로 크기에서 캐릭터의 세로 크기를 뺀 위치
- 그런데 가로 위치가 약간 오른쪽으로 쏠려 있다. 이를 중앙으로 옮기기 위해
 - 화면 절반 크기 - 캐릭터의 절반을 빼 준다.



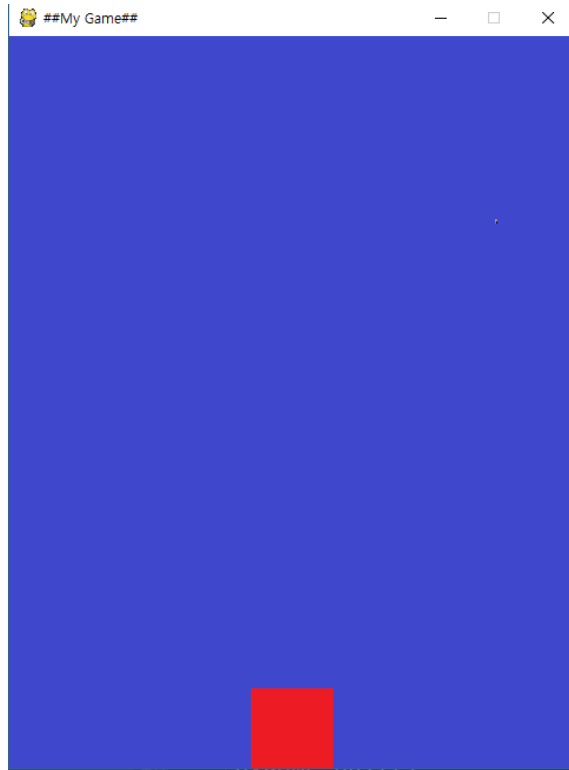
3-3. 캐릭터 위치 설정

- `character_x_pos = (screen_width / 2) - (character_width/2)` # 가로 위치
- `character_y_pos = screen_height - character_height` # 세로 위치



3-4. 캐릭터 그리기

- `screen.blit(character, (character_x_pos, character_y_pos))`



4. 키보드 이벤트 처리 (4_keyboard_event.py)

- 캐릭터를 왼쪽, 오른쪽, 위, 아래로 이동하는 키보드 이벤트 처리
- 변수 선언

```
24     # 이동할 좌표
25     to_x = 0
26     to_y = 0
```

1. 먼저 키가 눌려졌는지 확인 후 눌려 졌다면,
 - 왼쪽 키보드가 눌려졌다면, 왼쪽으로 5 만큼 이동
 - 오른쪽 키보드가 눌려졌다면, 오른쪽으로 5 만큼 이동
 - 위쪽 키보드가 눌려졌다면, 위쪽으로 5 만큼 이동
 - 아래쪽 키보드가 눌려졌다면, 아래쪽으로 5 만큼 이동
2. 만약 방향키를 눌렀다 떼면 현재 캐릭터의 좌우 위치는 0으로 처리

4-1. 키보드 이벤트 처리

```
35     if event.type == pygame.KEYDOWN: # 먼저 키가 눌려졌는지 확인
36         if event.key == pygame.K_LEFT: # 왼쪽 키보드가 눌려졌다면 캐릭터를 왼쪽으로 이동
37             to_x -= 5
38         elif event.key == pygame.K_RIGHT: # 오른쪽 키보드가 눌려졌다면 캐릭터를 오른쪽으로 이동
39             to_x += 5
40         elif event.key == pygame.K_UP: # 위쪽 키보드가 눌려졌다면 캐릭터를 위쪽으로 이동
41             to_y -= 5
42         elif event.key == pygame.K_DOWN: # 아래쪽 키보드가 눌려졌다면 캐릭터를 아래쪽으로 이동
43             to_y += 5
44     if event.type == pygame.KEYUP: # 방향키를 떼면 멈춤
45         if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
46             to_x = 0
47         elif event.key == pygame.K_UP or event.key == pygame.K_DOWN:
48             to_y = 0
49
50     character_x_pos += to_x
51     character_y_pos += to_y
```

4-2 .키보드 이벤트 처리

- 캐릭터 화면 밖으로 벗어나는 것 처리 (가로, 세로 경계값 처리)
 - 캐릭터 위치가 화면내에 위치하도록 하면 됨

```
53      # 가로 경계값 처리(화면 밖으로 벗어난 경우)
54      if character_x_pos < 0 :
55          character_x_pos = 0
56      elif character_x_pos > screen_width - character_width:
57          character_x_pos = screen_width-character_width
58
59      # 세로 경계값 처리(화면 밖으로 벗어난 경우)
60      if character_y_pos < 0:
61          character_y_pos = 0
62      elif character_y_pos > screen_height - character_height:
63          character_y_pos = screen_height - character_height
64
```

5. FPS: Frame Per Second (5_frame_per_second.py)

- FPS(초당 프레임 수): 프레임수가 높으면 캐릭터나 화면이 부드럽게 움직이고, 낮으면 끊기는 현상이 있다) 즉,이동 속도는 일정하게 유지 시켜 줘야 한다.

```
13 # FPS
14 clock = pygame.time.Clock()
```

- 이벤트 처리 시 이동 값 5 로 명시한 부분을 변수 처리한다.

```
31 # 이동 속도
32 character_speed = 0.6
```

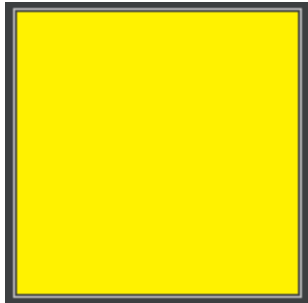
- `dt = clock.tick(10)` # 게임 화면의 초당 프레임 수 를 설정

5. FPS: Frame Per Second (5_frame_per_second.py)

```
49     if event.type == pygame.KEYDOWN: # 먼저 키가 눌려졌는지 확인
50         if event.key == pygame.K_LEFT: # 왼쪽 키보드가 눌려졌다면 캐릭터를 왼쪽으로 이동
51             to_x -= character_speed
52         elif event.key == pygame.K_RIGHT: # 오른쪽 키보드가 눌려졌다면 캐릭터를 오른쪽으로 이동
53             to_x += character_speed
54         elif event.key == pygame.K_UP: # 위쪽 키보드가 눌려졌다면 캐릭터를 위쪽으로 이동
55             to_y -= character_speed
56         elif event.key == pygame.K_DOWN: # 아래쪽 키보드가 눌려졌다면 캐릭터를 아래쪽으로 이동
57             to_y += character_speed
58     if event.type == pygame.KEYUP: # 방향키를 떼면 멈춤
59         if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
60             to_x = 0
61         elif event.key == pygame.K_UP or event.key == pygame.K_DOWN:
62             to_y = 0
63
64     character_x_pos += to_x * dt
65     character_y_pos += to_y * dt
```

6. 적 캐릭터 추가 및 충돌 처리(6_collesion.py)

- 그림판으로 적 이미지 만들기
 - 픽셀-가로세로(70*70)-가로세로 비율유지 해지
 - 노랑으로 채우기
 - pygame_basic - enemy.png 로 저장



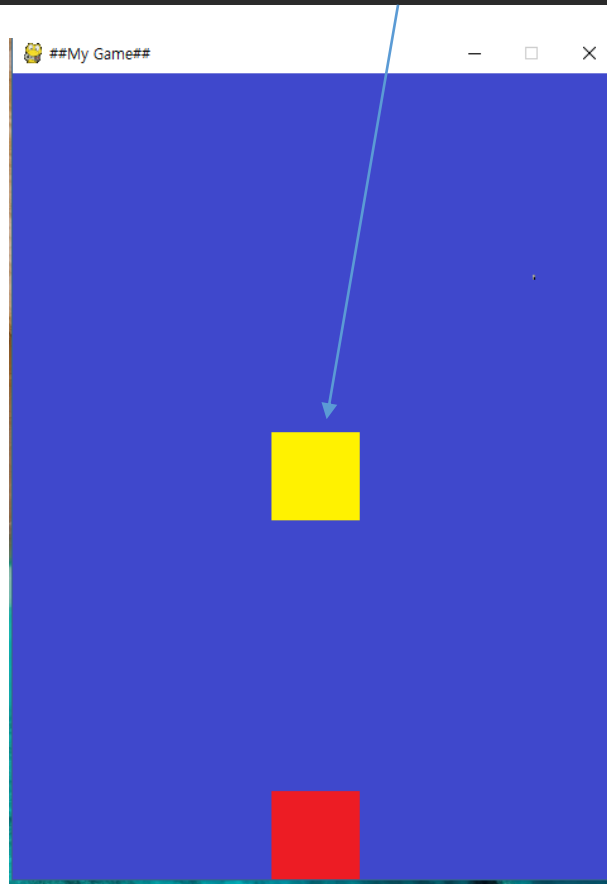
6-1. 적 캐릭터 불러오기

- `enemy = pygame.image.load()`
- `load()`에서 이미지 경로에서 이미지 불러오기
 - 이미지 더블클릭 후 이미지이름 탭에서 우클릭 - `copypath` 후 코드에 붙여넣기
 - ₩이를 그냥 `/`(슬래시)로 변경하기

```
34 # 적 enemy 캐릭터
35 enemy = pygame.image.load("C:/Users/HeeGyun/PycharmProjects/HelloPycharm/pygame_basic/enemy.png")
```

6-2. 적 캐릭터 위치 설정

```
36 enemy_size = enemy.get_rect().size # 이미지의 크기를 구해옴
37 enemy_width = enemy_size[0] # 캐릭터의 가로크기
38 enemy_height = enemy_size[1] # 캐릭터의 세로크기
39 enemy_x_pos = (screen_width / 2) - (enemy_width / 2) # 화면 가로의 절반 크기에 해당하는 곳에 위치(가로 위치)-캐릭터의 절반만큼을 빼기
40 enemy_y_pos = (screen_height / 2) - (enemy_height / 2) # 화면 세로 크기 가장 아래에 해당하는 곳에 위치(세로 위치)
```



6-3. 적 캐릭터 그리기

- `screen.blit(enemy, (enemy_x_pos, enemy_y_pos))`

```
102     screen.blit(background, (0, 0)) # 배경 그리기
103     screen.blit(character, (character_x_pos, character_y_pos)) # 캐릭터 그리기
104     screen.blit(enemy, (enemy_x_pos, enemy_y_pos)) # 적 enemy 그리기
```

6-4. 충돌 처리

- 캐릭터와 적의 충돌 처리를 위해 rect 정보를 얻어온 후, 그 값을 업데이트 함

```
88  # 충돌 처리를 위한 rect 정보 업데이트
89  character_rect = character.get_rect()
90  character_rect.left = character_x_pos
91  character_rect.top = character_y_pos
92
93  enemy_rect = enemy.get_rect() # 적의 위치 rect 정보 업데이트
94  enemy_rect.left = enemy_x_pos
95  enemy_rect.top = enemy_y_pos
```

6-5. 충돌 체크

- 충돌체크 함수 colliderect()사용 : 사각형 기준으로 충돌 체크
- 충돌 시, 게임 상태를 False로 변경

```
97      # 충돌 체크
98      if character_rect.colliderect(enemy_rect): # 사각형 기준으로 충돌체크
99          print("충돌했어요")
100         running = False
```

7. 텍스트 표시 (7_text.py)

1. 타이머 표시 폰트 정의 -> `game_font=pygame.font.Font(None, 40)`
2. 총 시간 설정
3. 시작 시간 : 현재 tick을 받아 와 시작시간으로 설정
-> `start_ticks= pygame.time.get_ticks()`
4. 경과 시간 정보 -> 현재 tick을 받아와서 시작시간을 빼줌
-> `elapsed_time = (pygame.time.get_ticks() - start_ticks) / 1000` (초단위로 표시 위해)

7-1. 타이머 텍스트 표시 (7_text.py)

```
42 # 폰트 정의
43 game_font = pygame.font.Font(None, 40) # 폰트 객체 생성 (폰트, 크기)
44
45 # 총 시간
46 total_time = 10
47
48 # 시작 시간 정보
49 start_ticks = pygame.time.get_ticks() # 현재 tick을 받아옴
50
51
115 # 타이머 집어 넣기
116 # 경과 시간 계산
117 elapsed_time = (pygame.time.get_ticks() - start_ticks) / 1000
118 # 경과시간(ms)을 1000으로 나누어서 초(s) 단위로 표시
119
120 timer = game_font.render(str(int(total_time - elapsed_time)), True, (255, 255, 255))
121 # (출력할 글자, True, 글자 색상)
122 screen.blit(timer, (10, 10))
123
124 # 만약 시간이 0 이하이면 게임 종료
125 if total_time - elapsed_time <= 0:
126     print("타임아웃")
127     running = False
```

7-2. 타이머 텍스트 표시 (7_text.py)

5. 잠시 대기

```
131     # 잠시 대기
132     pygame.time.delay(2000) # 2초 정도 대기(ms)
```

8. 게임 프레임 설정 (8_frame.py)

- 나만의 게임 프레임 파일 설정하기
 1. 기본 초기화 부분
 2. 사용자 게임 초기화 부분
 - 배경화면, 게임 이미지, 좌표, 속도, 폰트등 설정
 3. 이벤트 처리 (키보드, 마우스등)
 4. 게임 캐릭터 위치 정의
 5. 충돌처리
 6. 화면에 그리기
 7. 게임 종료

8. 게임 프레임 설정 (8_frame.py)

```
1  import pygame
2  #####
3  # 기본 초기화 (반드시 해야 하는 것들)
4  pygame.init()
5
6  # 화면 크기 설정
7  screen_width = 480 # 가로 크기
8  screen_height = 640 # 세로 크기
9  screen = pygame.display.set_mode((screen_width, screen_height))
10
11 # 화면 타이틀 설정
12 pygame.display.set_caption("게임 이름") # 게임이름 타이틀 지정
13
14 # FPS
15 clock = pygame.time.Clock()
16 #####
```

8. 게임 프레임 설정 (8_frame.py)

```
18 # 1. 사용자 게임 초기화( 배경화면, 게임이미지, 좌표, 속도, 폰트 등 설정)
19
20 running = True
21 while running:
22     dt = clock.tick(30)
23
24     # 2. 이벤트 처리(키보드, 마우스등)
25     for event in pygame.event.get():
26         if event.type == pygame.QUIT:
27             running = False
28
29     # 3. 게임 캐릭터 위치 정의
30
31     # 4. 충돌 처리
32
33     # 5. 화면에 그리기
34     pygame.display.update()
35
36 # pygame 종료
37 pygame.quit()
```

Quiz: 똥피하기 게임 만들기

[게임 조건]

1. 캐릭터는 화면 가장 아래에 위치, 좌우로만 이동 가능
2. 똥은 화면 가장 위에서 떨어짐. x 좌표는 매번 랜덤으로 설정
3. 캐릭터가 똥을 피하면 다음 똥이 다시 떨어짐
4. 캐릭터가 똥과 충돌하면 게임 종료
5. FPS는 30으로 고정

[게임 이미지]

1. 배경 : 640 * 480 (세로 가로) - background.png
2. 캐릭터 : 70 * 70 - character.png
3. 똥 : 70 * 70 - enemy.png