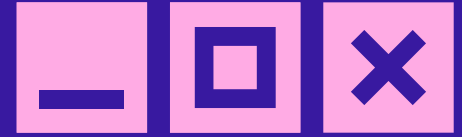


피하기 게임



2023 PYTHON WEEK

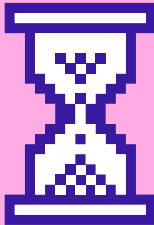
[인생이 게임 팀 - 김민성, 박광석]



PRESENTATION

OVERVIEW

- 목표
- 파이썬 코드 설명
- 게임 플레이



모티브 게임

피하기 게임은
'세상에서 가장 어려운 게임' 을 변형
시켜서 만들었다





```
import pygame
import sys

pygame.init() # 초기화

# 화면 크기 설정
screen_width = 1280 # 가로 크기
screen_height = 720 # 세로 크기
screen = pygame.display.set_mode((screen_width, screen_height))

# 화면 타이틀 설정
pygame.display.set_caption("피하기 게임")

# FPS
clock = pygame.time.Clock()

# 배경 이미지 불러오기
start_background = pygame.image.load("D:\\세종대학교\\1학년 여름방학\\PYTHON WEEK\\1_PYTHON WEEK\\background\\start_bg.png")
dead_background = pygame.image.load("D:\\세종대학교\\1학년 여름방학\\PYTHON WEEK\\1_PYTHON WEEK\\background\\dead_bg.png")

# 시작버튼 불러오기
start_button = pygame.image.load("D:\\세종대학교\\1학년 여름방학\\PYTHON WEEK\\1_PYTHON WEEK\\button\\start_btn.png")
start_button_size = start_button.get_rect().size # 시작버튼 이미지의 크기를 구해옴
start_button_width = start_button_size[0] # 시작버튼의 가로 크기
start_button_height = start_button_size[1] # 시작버튼의 세로 크기

# 시작버튼의 위치
start_button_x_pos = (screen_width / 2) - (start_button_width / 2) - 300
start_button_y_pos = (screen_height / 2) - 100

# 폰트 정의
game_font = pygame.font.Font(None, 80) # 폰트 객체 생성 (폰트, 크기)
```



기본 설정

pygame을 이용

화면 크기 = 1280x720

배경 이미지와 시작 버튼, 시작 버튼 위치 불러오기

```
# 이벤트 루프
running = True # 게임이 진행중
while running: # 게임이 진행중 일 때
    dt = clock.tick(60) # FPS : 60

    for event in pygame.event.get(): # 어떤 이벤트가 발생하는가
        if event.type == pygame.QUIT: # 창이 닫히는 이벤트가 발생하였을때
            running = False # 게임이 진행중이 아님

        if event.type == pygame.KEYDOWN: # 키가 눌려졌는지 확인
            if event.key == pygame.K_LEFT: # 캐릭터를 왼쪽으로
                character_to_x -= character_speed
            elif event.key == pygame.K_RIGHT: # 캐릭터를 오른쪽으로
                character_to_x += character_speed
            elif event.key == pygame.K_UP: # 캐릭터를 위로
                character_to_y -= character_speed
            elif event.key == pygame.K_DOWN: # 캐릭터를 아래로
                character_to_y += character_speed

        if event.type == pygame.KEYUP: # 방향키를 떼면 멈춤
            if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
                character_to_x = 0
            if event.key == pygame.K_UP or event.key == pygame.K_DOWN:
                character_to_y = 0
```



메인 캐릭터 움직이기

while 문을 이용하여 게임이 계속 실행 되도록 만들기
dt 는 프레임(FPS)을 위한 변수
if 를 사용하여 키보드의 방향키가 눌려졌는지 확인
눌려져있다면 이동할 만큼의 위치를 더해줌
떼면 이동할 위치를 0 으로 만든다



```
# 시작화면
if background_type == 0:
    # 시작화면에 그리기
    screen.blit(start_background, (0, 0))
    screen.blit(character, (character_x_pos, character_y_pos))
    screen.blit(start_button, (start_button_x_pos, start_button_y_pos))

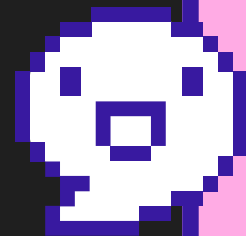
    # 버튼 활성화가 True라면 시작버튼 창에 띄우기
    if start_button_enabled:
        screen.blit(start_button, (start_button_x_pos, start_button_y_pos))

# 시작 버튼 충돌 후 비활성화 및 스테이지 시작
if start_button_enabled and character_rect.colliderect(start_button_rect):
    start_button_enabled = False
    background_type = 2
    character_x_pos, character_y_pos = (block1_1_pos[0], block1_1_pos[1])

# 죽었을때 화면
elif background_type == 1:
    # 죽었을 때 화면 나타내기
    screen.blit(dead_background, (0,0))
    screen.blit(character, (character_x_pos, character_y_pos))
    screen.blit(return1, (return1_x_pos, return1_y_pos))

    # 다시하기 버튼 측량
    return1_rect = return1.get_rect()
    return1_rect.left = return1_x_pos
    return1_rect.top = return1_y_pos

    # 다시하기 버튼을 누르면 현 스테이지로 이동
    if character_rect.colliderect(return1_rect):
        map_type = saved_map_type
        background_type = 2
        if map_type == 1:
            character_x_pos, character_y_pos = block1_1_pos
        if map_type == 2:
            character_x_pos, character_y_pos = block2_1_pos
        if map_type == 3:
            character_x_pos, character_y_pos = block3_1_pos
        if map_type == 4:
            character_x_pos, character_y_pos = block4_1_pos
        if map_type == 5:
            character_x_pos, character_y_pos = block5_1_pos
```



화면 전환

화면의 타임

background_type = 0 # 0 -> 기본 시작 화면

1 -> 재시작 화면

2 -> 게임 화면

맵 5개 만들기

map_type = 1 # 1 -> 맵 1

2 -> 맵 2

3 -> 맵 3

4 -> 맵 4

5 -> 맵 5

왼쪽과 같이 화면의 타임을 나누었다.

처음 시작화면에서 캐릭터가가

시작 버튼에 충돌하면

게임 화면으로 넘어가지게 된다.

그리고 만약 장애물에 부딪쳐 죽으면

GAMEOVER이 뜨고

다시하기 버튼이 생긴다

캐릭터가 다시하기 버튼에 충돌하면

원래 화면으로 돌아가지게 된다.



```
# 블록에서 벗어나면 처음 위치로 이동하도록 하기
blocks1 = [block1_1_rect, block1_2_rect, block1_3_rect, block1_4_rect]
original_image = pygame.image.load("D:\\세종대학교\\1학년 여름방학\\PYTHON WEEK\\1_PYTHONWEEK\\IMAGE\\
if not any(character_rect.colliderect(block) for block in blocks1):

    character_to_x = 0
    character_to_y = 0

    character_width -= 1 # 가로 크기를 줄입니다.
    character_height -= 1 # 세로 크기를 줄입니다.

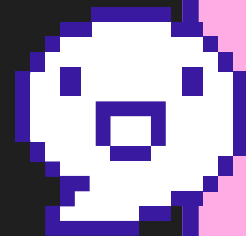
    character_x_pos += 0.5
    character_y_pos += 0.5

    character = pygame.transform.scale(original_image, (character_width, character_height)) # 원본

    if character_width <= 0 or character_height <= 0:
        character_width = character_size[0] # 가로 크기를 줄입니다.
        character_height = character_size[1] # 세로 크기를 줄입니다.
        character = pygame.transform.scale(original_image, (character_width, character_height))
        character_x_pos, character_y_pos = (block1_1_pos[0], block1_1_pos[1])

# block1-4에 닿이면 다음 스테이지로 이동
if character_rect.colliderect(block1_4_rect):
    map_type += 1
    character_x_pos, character_y_pos = (block2_1_pos[0], block2_1_pos[1])

# 2스테이지 일 때 화면 나타내기
if map_type == 2:
```



죽음, 성공 코드

변수 **block** 은 게임 스테이지에 있는
맵을 뜻한다.

만약 맵 밖으로 캐릭터가 나간다면
떨어지는 효과를 주면서 죽는다

그리고 원래 리스폰 지점으로

다시 생성되게 된다.

만약 성공하게 되는 블록으로 들어가면

다음 맵으로 넘어가지게 된다.

```
# enemy1 움직이기
if enemy1_y_1_pos <= 240:
    enemy1_1_move_type = 0
elif enemy1_y_1_pos >= 240 + 100 - enemy1_height:
    enemy1_1_move_type = 1

if enemy1_1_move_type == 0:
    enemy1_y_1_to_y = enemy1_1_speed
elif enemy1_1_move_type == 1:
    enemy1_y_1_to_y = -enemy1_1_speed

enemy1_y_1_pos += enemy1_y_1_to_y * dt
# enemy2, 4, 6 움직이기
if enemy2_y_1_pos <= 240:
    enemy1_2_move_type = 0
elif enemy2_y_1_pos >= 240 + 360 - enemy1_height:
    enemy1_2_move_type = 1

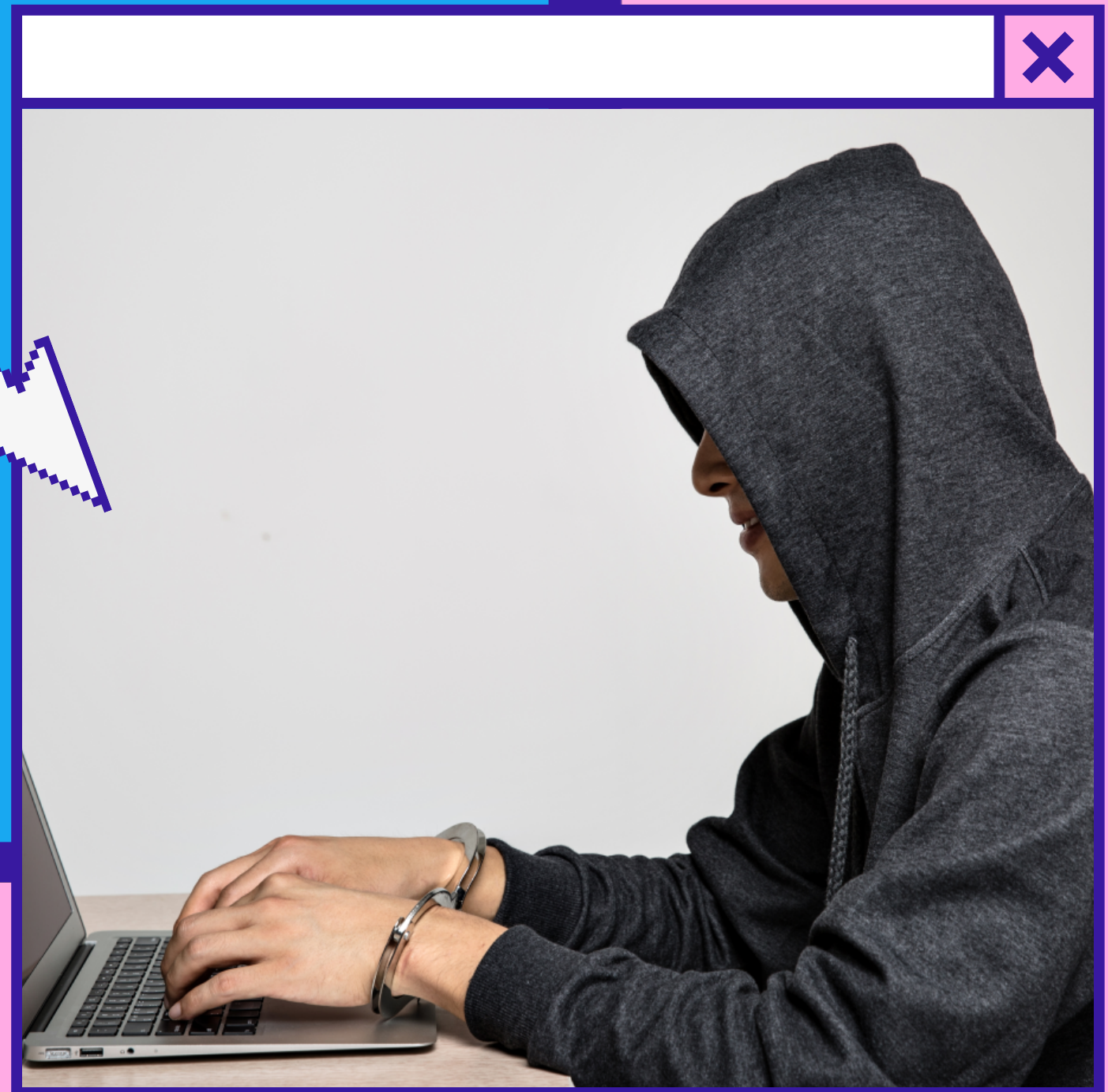
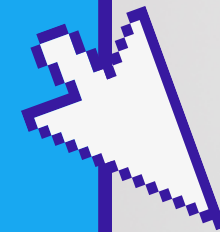
if enemy1_2_move_type == 0:
    enemy2_y_1_to_y = enemy1_2_speed
elif enemy1_2_move_type == 1:
    enemy2_y_1_to_y = -enemy1_2_speed

enemy2_y_1_pos += enemy2_y_1_to_y * dt
```

장애물 움직이기

이러한 코드로
장애물을 상하좌우로 움직이거나
회전하는 등의 코드를 만들었다.

피하기 게임 시연



감사합니다!