

PYGAME으로 핑 퐁게임 만들기

10318 최유준

목차

- 스크린 세팅 및 화면 업데이트
- 사용할 변수 선언과 공 이동 구문
- 키 입력 관련 함수
- 플레이어 이동 구문
- 충돌감지
- 점수판 세팅 및 점수계산

스크린 세팅

```
import sys
import pygame
from pygame.locals import *
# import math
# import random
pygame.init() # 파이게임 초기화

frame = pygame.time.Clock()
screen_width = 640

screen_height = 440

BLUE = (0,0,255)
RED = (255,0,0)
GREEN = (0,255,0)
BLACK = (0,0,0)
WHITE = (255,255,255)

screen = pygame.display.set_mode((screen_width,screen_height))
screen.fill(BLACK)
pygame.display.set_caption("pingpong")
```

게임을 만들고 실행시키기 위해 가장 먼저 보이는 화면을 만들어 주었습니다

`frame.tick(fps)`

화면을 일정 간격으로 보여주도록 초당 프레임을 설정하였습니다

`pygame.display.update()`

화면에 작성한것을 계속 업데이트하기위해 사용하였습니다

변수 선언과 공 이동 구문

```
p1 = pygame.draw.rect(screen, RED, [0, 165, 20, 90]) # 플1
p2 = pygame.draw.rect(screen, BLUE, [620, 165, 20, 90]) # 플2
boll = pygame.draw.circle(screen, WHITE, [325, 220], 10) #5
redcolor = RED
bluecolor = BLUE
blackcolor = BLACK
fps = 60
bsd = 10
bx = 325
by = 220
y1 = 160
y2 = 160
length = 50
score1 = 0
score2 = 0
t_color = (255, 255, 255)
font = pygame.font.SysFont("arial", 30, True, True)
```

핑퐁게임을 만들기위해 여러 변수들을 선언하고
플레이어 1,2 와 공의 모양을 잡아줬습니다.

```
deleteB = pygame.draw.rect(screen, blackcolor, boll)
boll = deleteB
del boll
bx += bsd
boll = pygame.draw.circle(screen, WHITE, [bx, by], 10)
```

공이 정해진 속도대로 움직일 수 있게
하였습니다

키 입력 관련 함수들

```
for event in pygame.event.get():  
    if event.type == QUIT:  
        ing = False  
        pygame.quit()  
        sys.exit()
```

이벤트를 처리하는 부분입니다
창을 닫으려고 하는 이벤트가 감지되면
즉시 창이 사라집니다

```
if not hasattr(event, 'key'):  
    continue
```

키 관련 이벤트가 아니면 건너뛰는 함수입니다

오래 누르면 누른만큼 이동할수 있게
반복해주는 함수입니다
플레이어 이동 부분에서 반복해주는 함수를
알지 못하였어서 조금의 시간을 들여
구글링을 통해 알게 되었습니다

```
39 pygame.key.set_repeat(30, 30)
```

플레이어 이동 구문

```
if event.type == KEYDOWN:
    if event.key == K_W or event.key == K_S:
        if event.key == K_W and boll.colliderect(p1):
            by -= length
        elif event.key == K_S and boll.colliderect(p1):
            by += length
        deleteW = pygame.draw.rect(screen, blackcolor, p1)
        p1 = deleteW
        del p1
        if event.key == K_W:
            y1 = y1 - 20
        else:
            y1 = y1 + 20
        p1 = pygame.draw.rect(screen, redcolor, [0, y1, 20, 90])

    if event.key == K_UP or event.key == K_DOWN:
        if event.key == K_UP and boll.colliderect(p2):
            by -= length
        elif event.key == K_DOWN and boll.colliderect(p2):
            by += length
        DeleteUp = pygame.draw.rect(screen, blackcolor, p2)
        p2 = DeleteUp
        del p2
        if event.key == K_UP:
            y2 -= 20
        else:
            y2 += 20
        p2 = pygame.draw.rect(screen, bluecolor, [620, y2, 20, 90])
```

키보드 입력을 받는 함수를
사용해서 만들었습니다.
플레이어의 좌표를 옮기는 방법을
생각했는데 생각이 잘 나지않았어서
떠올리기까지 오래걸렸었습니다.

충돌감지

```
if y1 <= 0 or y1 >= 350:  
    deleteW = pygame.draw.rect(screen, blackcolor, p1)  
    p1 = deleteW  
    del p1  
    if y1 <= 0:  
        y1 = 0  
        p1 = pygame.draw.rect(screen, redcolor, [0, y1, 20, 90])  
    else:  
        y1 = 350  
        p1 = pygame.draw.rect(screen, redcolor, [0, y1, 20, 90])
```

```
if y2 <= 0 or y2 >= 350:  
    deleteW = pygame.draw.rect(screen, blackcolor, p2)  
    p2 = deleteW  
    del p2  
    if y2 >= 350:  
        y2 = 350  
        p2 = pygame.draw.rect(screen, bluecolor, [620, y2, 20, 90])  
    else:  
        y2 = 0  
        p2 = pygame.draw.rect(screen, bluecolor, [620, y2, 20, 90])
```

플레이어가 스크린 밖으로 벗어나지않게
Y값을 잡아주었습니다

```
if by <= 0 or by >= 440:  
    bsd *= -1
```

공이 y축 양쪽 끝에 닿으면 공의 속도에-1을 곱해주어
반대로 돌아가게 하였습니다

```
if boll.colliderect(p1):  
    bsd *= -1  
if boll.colliderect(p2):  
    bsd *= -1
```

Colliderct라는 함수를 사용하여
플레이어가 공에 닿으면 똑같이 공의 속도에
-1을 곱하여 반대로 돌아가게 하였습니다

점수판 구성 및 점수계산

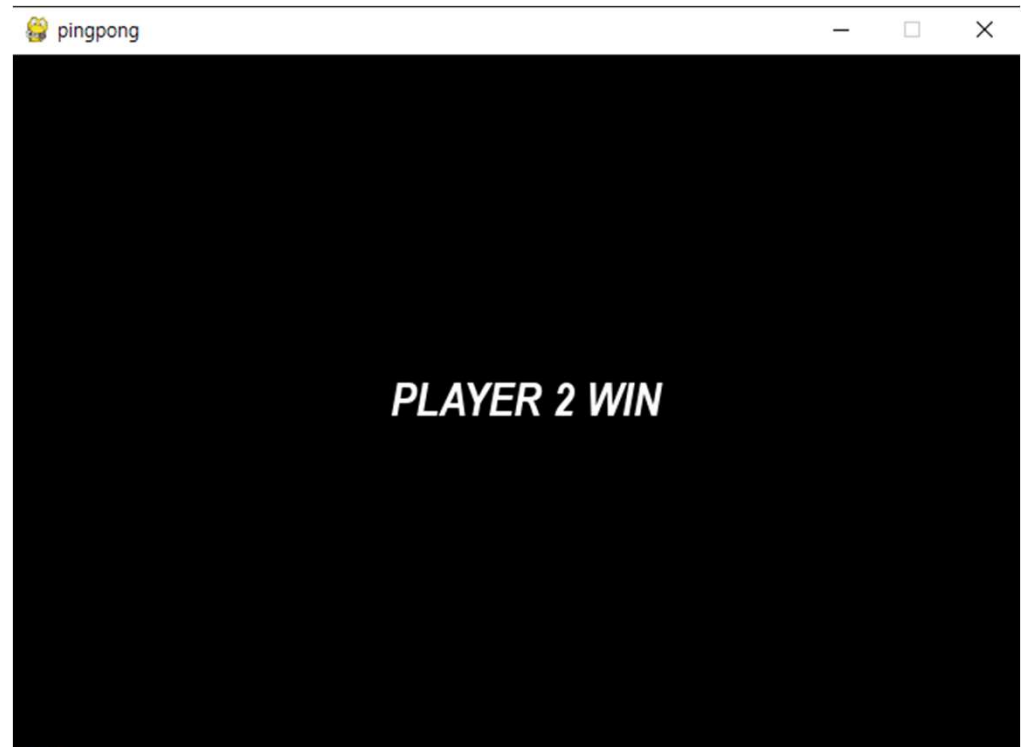
```
text = font.render(f"{score1} : {score2}", True, t_color)
screen.blit(text, (300, 100))
```

처음에 선언한 텍스트 색과 폰트로 구성된 점수를 화면에 띄웠습니다

```
if score1 == 7 or score2 == 7:
    screen.fill(blackcolor)
    redcolor = BLACK
    bluecolor = BLACK
    bsd = 0
    if score1 == 7:
        end_text_p1Win = font.render("PLAYER 1 WIN", True, t_color)
        screen.blit(end_text_p1Win, (240, 200))
    else:
        end_text_p2Win = font.render("PLAYER 2 WIN", True, t_color)
        screen.blit(end_text_p2Win, (240, 200))
```

플레이어 1과 2중에 아무나 먼저 7점을 얻는다면 7점을 먼저 얻은사람이 승리했다는 문구가 나오고 게임이 끝나게 됩니다.

승리화면



감사합니다