

실습 과제 #7 - 게임 프레임워크에 랭킹 상태의 추가

■ 랭킹 표시 상태(ranking_state)

- 메인 게임 상태(main_state)에서, 'q' 버튼을 누를 경우, 랭킹 표시 화면으로 이동
- 캐릭터 위치와 시간 정보를 화면에 출력(폰트사용)
- 플레이 시간 순서에 따라, 랭킹이 정해짐. 많으면 랭킹이 높음.
- 상위 플레이 10개만 표시.
- 플레이어 이름은 Play01, Player02, 이런 식으로 출력.
- 랭킹 표시 상태에서 'ESC'를 누르면, 다시 맨처음의 타이틀 표시 상태로 이동함.

■ 매 플레이를 마칠 때 마다, 플레이 시간을 측정할 수 있어야 함.

■ 랭킹은, json 형태의 파일 형식으로 기록함.

Serialization(직렬화)

- 프로그램 내의 변수를 외부에 저장 또는 내보내는 행위
- 나중에 다시 복구할 수 있어야 함.
- 게임 플레이 상황을 저장(Save)하고 로드(load)하는 것으로 직렬화로 볼 수 있음.

```
json.dump( XX , f)
```

변수 데이터 XX 를 파일 f로 출력해줌.

```
xx = json.load(f)
```

파일 f로부터 변수 데이터 xx를 채워줌.

json serialization을 이용한 게임 Save 및 Load

```
Class Hero:
```

```
    x = 100
```

```
    y = 200
```

```
    hp = 30.0
```

```
import json
```

```
f = open('save.txt', 'w')
```

```
json.dump([Hero.x, Hero.y, Hero.hp], f)
```

```
f.close()
```

```
[Hero.x, Hero.y, Hero.z] = [0, 0, 0.0]
```

```
f = open('save.txt', 'r')
```

```
[Hero.x, Hero.y, Hero.z] = json.load(f)
```

```
f.close()
```

```
print([Hero.x, Hero.y, Hero.hp])
```

폰트를 사용한 텍스트 출력(일단 영문만)

```
font = load_font('ENCR10B.TTF', size)
```

```
font.draw(x, y, 'Your Text')
```

```
font.draw(x, y, 'Your Text', (128, 128, 128))
```

r	the red component in the range 0-255
g	the green component in the range 0-255
b	the blue component in the range 0-255

후면 버퍼(back buffer)에 그리기 때문에,
나중에 update_canvas() 할 때, 표시됨.

이미지의 반투명화

`image.opacify(o)`

o: 0~1까지의 값. 0이면 완전 투명, 1이면 불투명