

<https://github.com/ZJEKFKS/OSW>

1. main 함수의 game loop 의 while 반복문에서 기존의 음악을 정하는 함수를 수정해 randint 를 생성해 무작위로 세 가지 음악 중 하나를 플레이하도록 했다.

```
while True: # game loop
    rand = random.randint(0, 2)
    if rand == 0:
        pygame.mixer.music.load('Hover.mp3')
    elif rand == 1:
        pygame.mixer.music.load('Our_Lives_Past.mp3')
    else:
        pygame.mixer.music.load('Platform_9.mp3')
    pygame.mixer.music.play(-1, 0.0)
```

2. main 함수를 수정해서 윈도우 타이틀을 바꿔 출력했다.

```
pygame.display.set_caption('2023056926 권영진') # 윈도우 타이틀
```

3. main 함수에서 showTextScreen 함수로 전달하는 인자의 값을 바꿨다.

```
showTextScreen('MY TETRIS') # 타이틀 스크린
```

4. 기본 초기화 단계에서 TEXTCOLOR 와 TEXTSHADOWCOLOR 의 값을 변경했다.

```
TEXTCOLOR = YELLOW # 텍스트 색 변경
TEXTSHADOWCOLOR = (155, 155, 0) # 텍스트 쉐도우 색 변경
```

5. runGame 함수에서 게임 시작 시간을 기록해 gameloop 의 while 반복문에서 elapsedTime 변수에 지나간 시간을 저장하고, drawStatus 함수에서 기존의 다른 정보를 디스플레이 하는 코드를 참고해 시간을 표기하는 부분을 만들었다.

```
startTime = time.time() # 게임 시작 시간을 기록
```

```
while True: # game loop
    elapsedTime = int(time.time() - startTime) # 경과 시간을 초 단위로 계산
```

```
# 시간 표시
timeSurf = BASICFONT.render('Play Time: %s sec' % elapsedTime, True, TEXTCOLOR)
timeRect = timeSurf.get_rect()
timeRect.topleft = (10, 20)
DISPLAYSURF.blit(timeSurf, timeRect)
```

6. 기존에 지정된 색에 더해서 색을 추가하고 색상 리스트에 추가하고, `getNewPiece` 함수에 색상 인덱스를 추가해 각 피스마다 고유의 색을 지정했다, `drawBox` 함수에서 `LIGHTCOLORS` 도 사용하기때문에 추가했다.

```
# 색 추가
ORANGE = (255, 165, 0)
LIGHTORANGE = (255, 200, 0)
CYAN = (0, 255, 255)
LIGHTCYAN = (150, 255, 255)
PURPLE = (128, 0, 128)
LIGHTPURPLE = (200, 0, 200)
```

```
# 추가된 색 추가
COLORS = (BLUE, GREEN, RED, YELLOW, ORANGE, CYAN, PURPLE)
LIGHTCOLORS = (LIGHTBLUE, LIGHTGREEN, LIGHTRED, LIGHTYELLOW, LIGHTORANGE, LIGHTCYAN, LIGHTPURPLE)
```

```
def getNewPiece():
    # return a random new piece in a random rotation and color
    shape = random.choice(list(PIECES.keys()))
    colorIndex = list(PIECES.keys()).index(shape) # 각 블록에 고유의 색상 할당
```

```
newPiece = {'shape': shape,
            'rotation': random.randint(0, len(PIECES[shape]) - 1),
            'x': int(BOARDWIDTH / 2) - int(TEMPLATEWIDTH / 2),
            'y': -2, # start it above the board (i.e. less than 0)
            'color': colorIndex}
```