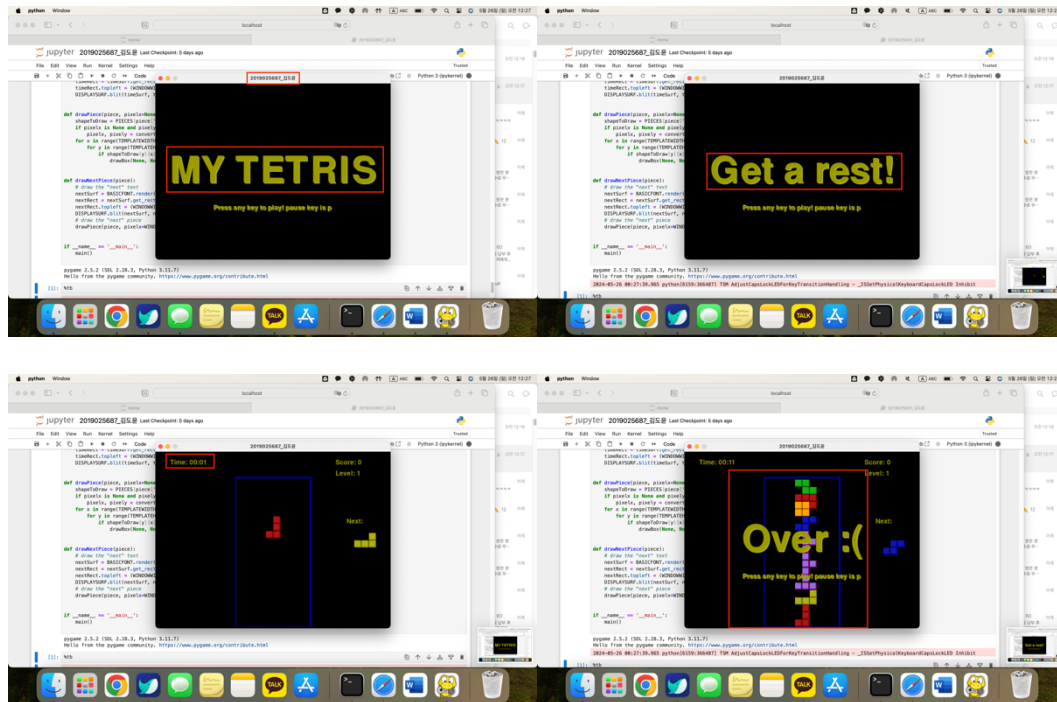


## 1. 수정 결과 - 이미지



## 2. 코드 수정

### 2-1) 현재 테트리스 게임의 배경음악을 주어진 3 개의 음악 중 1 개가 재생되도록 수정

```
while True: # game loop
    songs = ['Hover.mp3', 'Our_Lives_Past.mp3', 'Platform_9.mp3']
    selected_song = random.choice(songs)
    pygame.mixer.music.load(selected_song)
    pygame.mixer.music.play(-1, 0.0)

    runGame()

    pygame.mixer.music.stop()
    showTextScreen('Over :(')
```

- > 기존 코드는 0또는 1중 하나가 랜덤하게 선택되고 이에 해당하는 음원이 재생되는 방식
- > 음원의 수를 3개로 늘리기 위해 songs 라는 리스트를 작성
- > random.choice(songs) 를 통해 selected\_song 을 선택 후 이를 재생

## 2-2) 상태창 이름을 학번\_이름 으로 수정

```
pygame.display.set_caption('2019025687_김도윤')
```

-> pygame.display.set\_caption('')을 변경

## 2-3) 게임시작화면의 문구를 MY TETRIS 으로 변경

```
showTextScreen('MY TETRIS')
```

-> showTextScreen('')을 변경

## 2-4) 게임시작화면의 문구 및 배경색을 노란색으로 변경

```
TEXTCOLOR = YELLOW  
TEXTSHADOWCOLOR = LIGHTYELLOW
```

-> Textcolor 와 TextShadowcolor 를 각각 Yellow, LightYellow 로 변경

```
def makeTextObjs(text, font, color, shadowColor):  
    surf = font.render(text, True, color)  
    shadowSurf = font.render(text, True, shadowColor)  
    return surf, surf.get_rect(), shadowSurf, shadowSurf.get_rect()
```

-> makeTextObjs 함수에 shadowSurf 를 추가

```
def showTextScreen(text):  
    # This function displays large text in the center of the screen until a key is pressed.  
    titleSurf, titleRect, shadowSurf, shadowRect = makeTextObjs(text, BIGFONT, TEXTCOLOR, TEXTSHADOWCOLOR)  
    titleRect.center = (int(WINDOWWIDTH / 2), int(WINDOWHEIGHT / 2))  
    shadowRect.center = (int(WINDOWWIDTH / 2) + 3, int(WINDOWHEIGHT / 2) + 3)  
    DISPLAYSURF.blit(shadowSurf, shadowRect)  
    DISPLAYSURF.blit(titleSurf, titleRect)  
  
    # Draw the additional "Press a key to play." text.  
    pressKeySurf, pressKeyRect, pressKeyShadowSurf, pressKeyShadowRect = makeTextObjs('Press any key to play! pause key is p', BASICFONT, TEXTCOLOR, TEXTSHADOWCOLOR)  
    pressKeyRect.center = (int(WINDOWWIDTH / 2), int(WINDOWHEIGHT / 2) + 100)  
    pressKeyShadowRect.center = (int(WINDOWWIDTH / 2) + 1, int(WINDOWHEIGHT / 2) + 101)  
    DISPLAYSURF.blit(pressKeyShadowSurf, pressKeyShadowRect)  
    DISPLAYSURF.blit(pressKeySurf, pressKeyRect)
```

-> showTextScreen 함수에도 모두 shadowsurf와 관련된 코드를 작성하여 글자들에 그림자가 생길 수 있도록 함

-> shadowRect를 titleRect 보다 먼저 작성하여 shadow위에 text를 배치

-> shadow의 위치는 title의 경우 Text보다 +3, +3 , pressKey의 경우 +1, +101 (+100 + 1)로 크기에 따른 차이를 둠

2- 5) 게임 경과 시간을 초 단위로 표시 ( 새 게임 시작시 0 으로 초기화 되어야 함)

```
def runGame():  
    # setup variables for the start of the game  
    startTime = time.time()
```

-> 게임 시작시 startTime 을 기록

```
def drawTime(startTime):  
    elapsedTime = int(time.time() - startTime)  
    minutes = elapsedTime // 60  
    seconds = elapsedTime % 60  
    timeSurf = BASICFONT.render(f'Time: {minutes:02}:{seconds:02}', True, TEXTCOLOR)  
    timeRect = timeSurf.get_rect()  
    timeRect.topleft = (WINDOWWIDTH - 600, 20)  
    DISPLAYSURF.blit(timeSurf, timeRect)
```

-> elapsedTime으로 시작시간으로부터 경과된 시간을 계산함  
-> minutes, seconds로 elapsedTime을 분, 초로 변환  
-> timeSurf로 경과 시간을 text화  
-> timeRect, timeRect.topleft로 경과 시간의 표시 위치를 결정

```
# drawing everything on the screen  
DISPLAYSURF.fill(BGCOLOR)  
drawBoard(board)  
drawStatus(score, level)  
drawTime(startTime)  
drawNextPiece(nextPiece)  
if fallingPiece != None:  
    drawPiece(fallingPiece)
```

-> runGame 함수에서 drawTime 함수 호출, 경과 시간 표시

2 - 6) 7 개의 블록이 각각 고유의 색을 갖도록 코드를 수정하거나 추가

```
#           R   G   B
WHITE = (255, 255, 255)
GRAY = (185, 185, 185)
BLACK = (0, 0, 0)
RED = (155, 0, 0)
LIGHTRED = (175, 20, 20)
GREEN = (0, 155, 0)
LIGHTGREEN = (20, 175, 20)
BLUE = (0, 0, 155)
LIGHTBLUE = (20, 20, 175)
NAVY = (0, 0, 128)
LIGHTNAVY = (173, 216, 230)
YELLOW = (155, 155, 0)
LIGHTYELLOW = (175, 175, 20)
ORANGE = (255, 140, 0)
LIGHTORANGE = (255, 165, 0)
PURPLE = (128, 0, 128)
LIGHTPURPLE = (147, 112, 219)

BORDERCOLOR = BLUE
BGCOLOR = BLACK
TEXTCOLOR = YELLOW
TEXTSHADOWCOLOR = LIGHTYELLOW
COLORS = (BLUE, GREEN, RED, YELLOW, NAVY, ORANGE, PURPLE)
LIGHTCOLORS = (LIGHTBLUE, LIGHTGREEN, LIGHTRED, LIGHTYELLOW, LIGHTNAVY, LIGHTORANGE, LIGHTPURPLE)
```

-> NAVY,ORANGE,PURPLE과 각각에 매치되는 LIGHT 색상의 RGB코드를 입력

-> 해당 color를 각 COLORS, LIGHTCOLORS에 추가

```
PIECE_COLORS = {
    'S': 0,
    'Z': 1,
    'J': 2,
    'L': 3,
    'I': 4,
    'O': 5,
    'T': 6
}
```

-> PIECE\_COLORS 딕셔너리를 작성, 각 piece의 모양에 고유한 color 딕셔너리의 인덱스 번호를 부여

```
def getNewPiece():
    # return a new piece with a random shape and rotation
    shape = random.choice(list(PIECES.keys()))
    color = PIECE_COLORS[shape]
    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': color # assign color based on shape
    }
    return newPiece
```

-> getNewPiece 함수에서 새로운 piece를 생성. Shape에서 랜덤하게 모양이 결정되면 color는 선택된 모양에 해당하는 색상을 PIECE\_COLORS 딕셔너리에서 불러옴

-> newPiece라는 새로운 딕셔너리 생성, 'color'를 통해 piece에 맞는 색상인덱스 설정

```
def drawPiece(piece, pixelx=None, pixely=None):
    shapeToDraw = PIECES[piece['shape']][piece['rotation']]
    if pixelx is None and pixely is None:
        pixelx, pixely = convertToPixelCoords(piece['x'], piece['y'])
    for x in range(TEMPLATEWIDTH):
        for y in range(TEMPLATEHEIGHT):
            if shapeToDraw[y][x] != BLANK:
                drawBox(None, None, PIECE_COLORS[piece['shape']], pixelx + (x * BOXSIZE), pixely + (y * BOXSIZE))
```

-> drawBox 함수를 호출하여 piece를 그릴 때, PIECE\_COLORS를 통해 piece의 색상을 획득

### 3. 각 함수의 역할

#### 3-1) def main():

- 게임을 초기화하고 메인 게임 루프를 실행
- Pygame을 초기화하고 화면 설정
- 글꼴 로드
- 창 제목 설정
- 시작 화면 표시
- 무한 루프 및 게임 실행
- 음악 랜덤 선택 및 재생
- def runGame() 호출
- 게임 종료시 음악 정지 및 종료 화면 표시

#### 3-2) def runGame():

- 게임의 주요 루프를 실행
- 게임 상태 업데이트 및 화면 표시
- 게임보드 초기화, 새 조각 생성, 사용자 입력 처리, 피스 이동, 충돌 검사, 점수 및 레벨 업데이트

수행

- 게임 오버 상태 감지
- 게임 화면 업데이트

### 3-3) def getNewPiece( ):

- 새로운 테트리스 피스 생성 및 반환
- 피스 모양, 회전 상태, 시작 위치 및 색상 설정
- 피스는 랜덤하게 선택
- 피스의 초기 위치 설정

### 3-4) def isValidPosition( ):

- 피스가 보드의 유효 위치 내에 있는지 확인
- 피스가 보드 경계를 벗어나지 않고, 다른 피스와 겹치지 않는지를 검사
- 피스가 보드의 유효한 위치에 있으면 True, 그렇지 않으면 False를 반환

## 4. 함수의 호출 순서 및 호출 조건에 대한 설명

게임 시작

-> main()을 호출하여 초기 설정을 수행,

-> showTextScreen()을 통해 시작 화면 재생

-> main() 함수 내에서 무한 루프가 실행

-> runGame()이 호출

    -> 게임 보드와 피스를 초기화,

    -> 게임 루프 시작

        -> 사용자 입력을 처리

        -> 피스를 이동 및 회전

        -> 화면을 갱신

        -> 피스 착지

            -> addToBoard()와 removeCompleteLines() 호출, 보드 업데이트

-> 매 프레임마다 보드, 피스, 점수, 시간을 화면에 표시

## 5. GitHub Repository 주소

- <https://github.com/Kakillaa/osw>