

# 오픈소스SW 과제주심수업\_1주차 보고서

체크리스트

1, 현재 테트리스 게임의 배경음악을 주어진 3개의 음악 중 1개가 재생되도록 수정

```
while True: # game loop
    if random.randint(0, 1) == 0:
        pygame.mixer.music.load('Hover.mp3')
    else:
        pygame.mixer.music.load('Hover.mp3')
    pygame.mixer.music.play(-1, 0.0)
    runGame()
    pygame.mixer.music.stop()
    showTextScreen('Game Over')
```

메인 함수 속 반복문의 음악 로드 파일을 같은 디렉토리에 넣은 후 이름 'Hover.mp3' 지정 후 해결

2, 상태창 이름을 학번\_이름 으로 수정

```
pygame.display.set_caption('2021070228_KIMSEUNGYUN')
```

set\_caption속 스트링을 '2021070228\_KIMSEUNGYUN'로 수정하여 해결

3, 게임시작화면의 문구를 MY TETRIS으로 변경

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

메인 함수 속 첫번째 showTextScreen함수의 매개변수 text 값을 'MY TETRIS'로 변경하여 해결

4, 게임시작화면의 문구 및 배경색을 노란색으로 변경

```
BORDERCOLOR = BLUE
```

```
BGCOLOR = BLACK
```

```
TEXTCOLOR = YELLOW
```

```
TEXTSHADOWCOLOR = YELLOW
```

TEXTCOLOR와 TEXTSHADOWCOLOR의 값을 YELLOW 값으로 변경하여 해결

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

```
def runGame():
```

```
    # setup variables for the start of the game
    board = getBlankBoard()
    lastMoveDownTime = time.time()
    lastMoveSidewaysTime = time.time()
    lastFallTime = time.time()
    movingDown = False # note: there is no movingUp variable
    movingLeft = False
    movingRight = False
    score = 0
    level, fallFreq = calculateLevelAndFallFreq(score)
    startTime = time.time()
```

```
-----
playTime = int((time.time() - startTime))
```

```
    drawStatus(score, level, playTime)
```

```
-----
def drawStatus(score, level, playTime):
```

```
    scoreSurf = BASICFONT.render('Score: %s' % score, True, TEXTCOLOR)
    scoreRect = scoreSurf.get_rect()
    scoreRect.topleft = (WINDOWWIDTH - 150, 20)
    DISPLAYSURF.blit(scoreSurf, scoreRect)
```

```
levelSurf = BASICFONT.render('Level: %s' % level, True, TEXTCOLOR)
```

```
    levelRect = levelSurf.get_rect()
    levelRect.topleft = (WINDOWWIDTH - 150, 50)
    DISPLAYSURF.blit(levelSurf, levelRect)
```

```
timeSurf = BASICFONT.render('Play Time : %s sec' % playTime, True, TEXTCOLOR)
```

```
    timeRect = timeSurf.get_rect()
    timeRect.topright = (WINDOWWIDTH - 470, 50)
    DISPLAYSURF.blit(timeSurf, timeRect)
```

1. runGame함수 속에 startTime 변수 추가하여 게임 시작 시 시간으로 초기화
2. while 문 속에 playTime 변수 추가 후 (현재 시간 - 게임 시작 시간)하여 게임 시작 경과 시간 구한 후 int값으로 초기화 하여 저장
3. drawStatus 매개변수에 playtime 추가하여 화면에 경과 시간 출력하는 부분 추가 후 해결

```
timeSurf = BASICFONT.render('Play Time : %s sec' % playTime, True, TEXTCOLOR)
```

```
    timeRect = timeSurf.get_rect()
```

```
timeRect.topright = (WINDOWWIDTH - 470, 50)
```

```
DISPLAYSURF.blit(timeSurf, timeRect)
```

6, 7개의 블록이 각각 고유의 색을 갖도록 코드를 수정하거나 추가

```
PIECESCOLORS = {'S': 0,  
                 'Z': 1,  
                 'J': 2,  
                 'L': 3,  
                 'I': 0,  
                 'O': 1,  
                 'T': 2}
```

-----  
def getNewPiece():

```
    # return a random new piece in a random rotation and color
```

```
    shape = random.choice(list(PIECES.keys()))
```

```
    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': PIECESCOLORS[shape]}
```

```
    return newPiece
```

1. 각 피스의 고유 색을 지정하는 PIECESCOLORS 딕셔너리 생성
2. getNewPiece 함수에서 newPiece 딕셔너리에 'color'값을 PIECESCOLORS의 랜덤으로 생성된 모양에 해당하는 색으로 지정하여 해결

- 함수 역할 정리
  - 1 drawStatus 함수는 매개변수로 점수와 레벨을 받아와 게임 화면 우측 상단에 띄워주는 함수이다.
  - 2 drawPiece 함수는 piece 오브젝트를 받아서 모양을 확인하고 위치를 할당하고 화면 상 해당 위치에 piece를 그리는 함수이다.
  - 3 getNewPiece 함수는 새로운 피스의 모양과, 로테이션, 색깔을 랜덤으로 할당하여 newPiece오브젝트를 리턴해주는 함수다.
- Main 함수의 호출 과정 및 호출 조건
  - 1 pygame.init()을 호출하여 파이게임 모듈 초기화
  - 2 FPSLOCK = pygame.time.Clock()을 통해 프레임률을 제어할 수 있는 객체 생성
  - 3 DISPLAYSURF = pygame.display.set\_mode((WINDOWWIDTH, WINDOWHEIGHT))을 통해 디스플레이 창 설정
  - 4 BASICFONT와 BIGFONT를 통해 텍스트 폰트를 설정
  - 5 pygame.display.set\_caption('2021070228\_KIMSEUNGYUN')을 호출하여 창의 캡션을 설정
  - 6 showTextScreen('MY TETRIS')을 호출하여 "MY TETRIS"라는 텍스트 화면 제시
  - 7 while 루프를 통해 게임 루프가 시작. 이 루프는 사용자가 게임을 끝낼 때까지 계속 반복
  - 8 if random.randint(0, 1) == 0:을 통해 무작위로 조건을 체크. 두 조건 모두 pygame.mixer.music.load("Hover.mp3")를 실행하도록 설정되어 있어서 동일한 음악 파일을 로드
  - 9 pygame.mixer.music.play(-1, 0.0)을 호출하여 배경 음악을 반복 재생
  - 10 runGame()을 호출하여 실제 게임 플레이를 실행
  - 11 pygame.mixer.music.stop()을 호출하여 배경 음악 정지
  - 12 showTextScreen('Game Over')을 호출하여 게임 오버 텍스트 화면 제시

깃 주소 : <https://github.com/yunsful/osw>