



오픈소스SW 과제중심수업 1주차 보고서

ICT융합학부 2023033572 김규리

<https://github.com/cuuu2/osw>



목차

1. 코드 수정 부분
2. 함수의 역할
3. 함수의 호출 순서 및 조건

1. 코드 수정 부분

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

```
if random.randint(0, 1) == 0:
    pygame.mixer.music.load('Hover.mp3')
else:
    pygame.mixer.music.load('Platform_9.mp3')
```

기존 코드에서 같은 파일 내에 있는 'Hover.mp3' 혹은 'Platform_9.mp3' 중 하나가 재생되도록 변경

(2) 상태창 이름을 학번_이름으로 수정

```
pygame.display.set_caption('2023033572_KimGyuri')
```

기존 코드에서 상태창 위치에 뜨는 문구를 '2023033572_KimGyuri'로 수정

(3) 게임 시작 화면의 문구를 MY TETRIS로 변경

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

기존 시작 문구에서 'MY TETRIS'로 변경

(4) 게임 시작 화면의 문구 및 배경색을 노란색으로 변경

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

TEXTCOLOR와 TEXTSHADOWCOLOR을 기존 색에서 'YELLOW'로 변경

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

```
startTime = time.time()

elapsedTime = time.time() - startTime

drawStatus(score, level, elapsedTime)
```

함수 'runGame'에 변수 'startTime', 'elapsedTime' 추가

함수 'drawStatus'에 전달 인자 'elapsedTime' 추가

```
def drawStatus(score, level, elapsedTime):
    # draw the score text
    scoreSurf = BASICFONT.render('Score: %s' % score, True, TEXTCOLOR)
    scoreRect = scoreSurf.get_rect()
    scoreRect.topleft = (WINDOWWIDTH - 150, 20)
    DISPLAYSURF.blit(scoreSurf, scoreRect)

    # draw the level text
    levelSurf = BASICFONT.render('Level: %s' % level, True, TEXTCOLOR)
    levelRect = levelSurf.get_rect()
    levelRect.topleft = (WINDOWWIDTH - 150, 50)
    DISPLAYSURF.blit(levelSurf, levelRect)

    timeSurf = BASICFONT.render('Time: %s sec' % int(elapsedTime), True, TEXTCOLOR)
    timeRect = timeSurf.get_rect()
    timeRect.topleft = (WINDOWWIDTH - 490, 20)
    DISPLAYSURF.blit(timeSurf, timeRect)
```

함수 'drawStatus'에 받는 인자 'elapsedTime' 추가

텍스트를 담은 표면 객체 'timeSurf', 메서드에 의해 생성된 사각형 객체 'timeRect' 추가

텍스트 'timeSurf'를 지정된 위치 'timeRect'에 그림

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

```
#           R     G     B
WHITE      = (255, 255, 255)
LIGHTWHITE = (254, 254, 254)
GRAY       = (185, 185, 185)
LIGHTGRAY  = (205, 205, 205)
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)
YELLOW     = (155, 155,   0)
LIGHTYELLOW = (175, 175,  20)
PURPLE     = (77,  0, 77)
LIGHTPURPLE = (97, 20, 97)

BORDERCOLOR = BLUE
BGCOLOR = BLACK
TEXTCOLOR = YELLOW
TEXTSHADOWCOLOR = YELLOW
COLORS = (BLUE, GREEN, RED, YELLOW, WHITE, GRAY, PURPLE)
LIGHTCOLORS = (LIGHTBLUE, LIGHTGREEN, LIGHTRED, LIGHTYELLOW, LIGHTWHITE, LIGHTGRAY, LIGHTPURPLE)
assert len(COLORS) == len(LIGHTCOLORS) # each color must have light color
```

기존에 존재하는 COLORS는 4가지 색뿐이므로 블록당 색깔 지정이 어려워 WHITE, GRAY, 기존에 존재하지 않던 PURPLE 추가

COLORS의 색만큼 LIGHTCOLORS 또한 필요해서 LIGHTWHITE, LIGHTGRAY, LIGHTPURPLE 추가

```
PIECES_COLORS = {'S': WHITE,
                  'Z': YELLOW,
                  'J': GRAY,
                  'L': BLUE,
                  'I': RED,
                  'O': PURPLE,
                  'T': GREEN}
```

각 모양에 대한 색상 정의

```
'color': PIECES_COLORS[shape]}
```

기존 코드 'color': random.randint(0, len(COLORS)-1)에서 랜덤 요소를 없애고 지정된 인덱스를 따라가도록 수정

```
pygame.draw.rect(DISPLAYSURF, color, (pixelx + 1, pixely + 1, BOXSIZE - 1, BOXSIZE - 1))
pygame.draw.rect(DISPLAYSURF, LIGHTCOLORS[COLORS.index(color)], (pixelx + 1, pixely + 1, BOXSIZE - 4, BOXSIZE - 4))
```

color가 튜플이어야 하는데, 다른 튜플의 인덱스로 사용되고 있어 문제가 발생, color를 직접 COLORS의 인덱스로 사용하지 않고 COLORS 목록에서 color의 인덱스를 찾아 LIGHTCOLORS에 전달하는 것으로 수정

2. 함수의 역할

(1) main(): 시작과 동시에 실행을 요구받는 함수

FPSCLOCK, DISPLAYSURF, BASICFONT, BIGFONT를 전역 함수로 받음
DISPLAYSURF의 크기를 정해진 WINDOWWIDTH, WINDOWHEIGHT의 크기로 수정
BASICFONT를 주어진 폰트와 크기로 수정
BIGFONT를 주어진 폰트와 크기로 수정
상태창 이름을 수정
showTextScreen 함수 실행을 통해 화면에 'MY TETRIS' 문구 띄움
게임이 실행되는 동안 무한 반복문 실행
랜덤 함수를 통해 재생될 음악을 'Hover.mp3' 혹은 'Platform_9.mp3'로 설정
음악 재생 시작
runGame 함수 실행을 통해 게임 시작
게임이 종료될 시 음악 멈춤
showTextScreen 함수 실행을 통해 'Game Over' 문구 띄움

(2) runGame(): main 함수 내에서 실행을 명령받는 함수

getBlankBoard 함수 실행을 통해 board 변수 선언
lastMoveDownTime, lastMoveSidewaysTime, lastFall 변수를 time 모듈 내 time() 함수를 통해 선언
movingDown, movingLeft, movingRight 변수를 초기 값 False로 선언
score 변수를 초기 값 0으로 선언
level, fallFreq 변수를 calculateLevelAndFallFreq 함수 실행을 통해 선언
startTime 변수를 time 모듈 내 time() 함수를 통해 선언
fallingPiece, nextPiece 변수를 getNewPiece 함수를 통해 선언
게임이 실행되는 동안 계속 새로운 피스 생성
돌리거나 움직이는 것도 조작
score 계산

(3) getNewPiece(): 새로운 피스를 만드는 함수

shape 변수를 PIECES의 keys 중 하나를 임의로 선택해 선언
모양, 크기, 회전, 색이 정해진 newPiece 생성
newPiece를 반환

3. 함수의 호출 순서 및 조건

- (1) 실행 시 main() 함수 호출
- (2) showTextScreen 호출로 'MY TETRIS' 문구 띄움
- (3) runGame 호출로 게임 시작
- (4) getBlankBoard 호출로 빈 보드 생성
- (5) calculateLevelAndFallFreq 호출로 level과 fallFreq 선언
- (6) getNewPiece 호출로 fallingPiece 선언
- (7) getNewPiece 호출로 nextPiece 선언
- (8) 게임이 실행되는 동안 만약 fallingPiece가 없다면 fallingPiece에 nextPiece 할당
- (9) getNewPiece 호출로 nextPiece를 새로 선언
- (10) isValidPosition 호출로 보드 검사 후 새 피스가 들어올 수 없을 시 반복문 탈출, 게임 오버
- (10-1) isOnBoard 호출로 정해진 보드 크기 반환
- (11) checkForQuit 호출로 만약 창을 닫거나 ESC 클릭 시 게임이 꺼지게 함
- (11-1) terminate 호출로 게임 끄
- (12) p를 누르면 showTextScreen 호출로 'Paused' 문구 띄우고 멈춤
- (13) a, d, s 혹은 화살표를 누르면 블록 회전
- (14) lastFallTime과 fallFreq의 비교를 통해 score 계산
- (15) drawBoard 호출로 보드 자리 생성
- (15-1) drawBox 호출로 지정된 자리에 상자 그림
- (16) drawStatus 호출로 score, level, slapsedTime 화면에 표시
- (17) drawNextPiece 호출로 다음에 나올 블록 띄움
- (17-1) drawPiece 호출로 좌표 지정
- (17-2) drawBox 호출로 지정된 자리에 블록 그림
- (18) fallingPiece가 비어 있지 않다면 drawPiece 호출로 띄움