2019058668 영미언어문화학과 남윤서

Github 주소 :

https://github.com/ysnam0123/2019058668

1) 1~6번의 과제를 코드의 어느 부분을 어떻게 수정했는지 설명

```
def main():
    global FPSCLOCK, DISPLAYSURF, BASICFONT, BIGFONT
    pygame.init()
    FPSCLOCK = pygame.time.Clock()
    DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
    BASICFONT = pygame.font.Font('freesansbold.ttf', 18)
    BIGFONT = pygame.font.Font('freesansbold.ttf', 100)
    pygame.display.set_caption('2019058668_남윤서')
```

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

1. 음악 수정 (Hover.mp3, Our_Lives_Past.mp3)
2. pygame.display.set_caption('2019058668_남윤서') 를 통해서 상태창 이름 수정
3. showTextScreen('MY TETRIS') 를 통해 시작 문구 수정
4. (TEXTCOLOR = YELLOW) 으로 색 변경

```
    startTime = time.time()
    elapsedTime = int(time.time() - startTime)
        drawElapsedTime(elapsedTime)
```

```
def drawElapsedTime(elapsedTime):
    timeSurf = BASICFONT.render(f'Play time: {elapsedTime:01} sec',
True, TEXTCOLOR)
    timeRect = timeSurf.get_rect()
    timeRect.topleft = (10, 10)
    DISPLAYSURF.blit(timeSurf, timeRect)
```

으로 게임 경과 시간 설정. —(5)
시작 시간 저장, 경과시간 업데이트 , 화면에 표시(좌측 상단)

(6) 블록들이 각각 고유의 색을 갖도록 하기

```
SHAPE_COLORS = {
    'S': GREEN,
    'Z': RED,
    'J': BLUE,
    'L': LIGHTBLUE,
    'I': LIGHTGREEN,
    'O': YELLOW,
    'T': LIGHTRED
}

def getNewPiece():
    # return a random new piece in a random rotation and color
    shape = random.choice(list(PIECES.keys()))
    color = SHAPE_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}
    return newPiece
```

으로 수정 후,

```
def drawBox(boxx, boxy, color, pixelx=None, pixely=None):
    # draw a single box (each tetromino piece has four boxes)
    # at xy coordinates on the board. Or, if pixelx & pixely
    # are specified, draw to the pixel coordinates stored in
    # pixelx & pixely (this is used for the "Next" piece).
    if color == BLANK:
        return
    if pixelx == None and pixely == None:
        pixelx, pixely = convertToPixelCoords(boxx, boxy)
    pygame.draw.rect(DISPLAYSURF, color, (pixelx + 1, pixely + 1,
BOXSIZE - 1, BOXSIZE - 1))
```

drawbox 코드 부분 수정

---

2) 각 함수의 역할 3군데 정도 주요하다고 생각되는 코드를 설명해주기

1) 게임 시작 및 종료 함수

```
def runGame():
```

```
    board = getBlankBoard()
    lastMoveDownTime = time.time()
    lastMoveSidewaysTime = time.time()
    lastFallTime = time.time()
    startTime = time.time()
    movingDown = False
    movingLeft = False
    movingRight = False
    score = 0
    level, fallFreq = calculateLevelAndFallFreq(score)

    fallingPiece = getNewPiece()
    nextPiece = getNewPiece()

    while True:
       if fallingPiece == None:
            fallingPiece = nextPiece
            nextPiece = getNewPiece()
            lastFallTime = time.time()
```

이 부분에서 떨어지고 있는 조각이 없으면 새로운 조각을 생성하고,
마지막 떨어짐 시간 초기화하고 새 조각을 불러온다.

```
            if not isValidPosition(board, fallingPiece):
                return
```

마지막으로, 새 조각을 놓을 공간이 없으면 게임을 종료한다.

2) 새로운 테트리스 조각을 생성하는 함수

```
def getNewPiece():
    # return a random new piece in a random rotation and color
    shape = random.choice(list(PIECES.keys()))
    color = SHAPE_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}
    return newPiece
```

Shape -> PIECES 딕셔너리에서 무작위로 선택함
Color -> SHAPE_COLORS 딕셔너리에서 모양의 고유한 색상을 가져온다.
newPiece 딕셔너리 생성 -> 모양, 회전, 시작좌표,색상 설정
Return -> 새로운 조각 생성

3) 그리기 함수

```
def drawBoard(board):
    # draw the border around the board
    pygame.draw.rect(DISPLAYSURF, BORDERCOLOR, (XMARGIN - 3,
TOPMARGIN - 7, (BOARDWIDTH * BOXSIZE) + 8, (BOARDHEIGHT * BOXSIZE)
+ 8), 5)

    # fill the background of the board
    pygame.draw.rect(DISPLAYSURF, BGCOLOR, (XMARGIN, TOPMARGIN,
BOXSIZE * BOARDWIDTH, BOXSIZE * BOARDHEIGHT))
    # draw the individual boxes on the board
    for x in range(BOARDWIDTH):
        for y in range(BOARDHEIGHT):
            drawBox(x, y, board[x][y])

def drawPiece(piece, pixelx=None, pixely=None):
    shapeToDraw = PIECES[piece['shape']][piece['rotation']]
    if pixelx == None and pixely == None:
        # if pixelx & pixely hasn't been specified, use the
location stored in the piece data structure
        pixelx, pixely = convertToPixelCoords(piece['x'],
piece['y'])

    # draw each of the boxes that make up the piece
    for x in range(TEMPLATEWIDTH):
        for y in range(TEMPLATEHEIGHT):
            if shapeToDraw[y][x] != BLANK:
                drawBox(None, None, piece['color'], pixelx + (x *
BOXSIZE), pixely + (y * BOXSIZE))


def drawNextPiece(piece):
    # draw the "next" text
    nextSurf = BASICFONT.render('Next:', True, TEXTCOLOR)
    nextRect = nextSurf.get_rect()
    nextRect.topleft = (WINDOWWIDTH - 120, 80)
    DISPLAYSURF.blit(nextSurf, nextRect)
    # draw the "next" piece
    drawPiece(piece, pixelx=WINDOWWIDTH-120, pixely=100)
```

-> 게임 보드의 테두리와 배경, 각 칸을 그려준다
drawPiece 를 통해 현재 떨어지고 있는 블록을 그려준다.
convertToPixelCoords 함수는 게임 보드 좌표를 실제 화면 픽셀 좌표로 변환해준다.

3) 함수의 호출 순서 및 호출 조건에 대한 설명 (예를들어서 p를누르면 멈춘다 같은 설명)

1. 도형 이동

```
                if (event.key == K_LEFT or event.key == K_a) and
isValidPosition(board, fallingPiece, adjX=-1):
                    fallingPiece['x'] -= 1
                    movingLeft = True
                    movingRight = False
                    lastMoveSidewaysTime = time.time()
```

화살표뿐 아니라, asd 를 누르게 되면 도형이 움직인다.

```
                elif (event.key == K_q):  # rotate the other
direction
                    fallingPiece['rotation'] =
(fallingPiece['rotation'] - 1) %
len(PIECES[fallingPiece['shape']])
                    if not isValidPosition(board, fallingPiece):
                        fallingPiece['rotation'] =
(fallingPiece['rotation'] + 1) %
len(PIECES[fallingPiece['shape']])
```

q를 누르면 떨어지는 조각을 반시계방향으로 회전
W 를 누르면 떨어지는 조각을 시계방향으로 회전시킨다.