

# 오픈소스 SW 13주차 과제중심수업 보고서

학과 : ICT 융합학부

학번 : 2020096608

이름 : 정설빈

제출일 : 2021. 06. 06

( <https://github.com/jsulbeen/osw.git> )

## #화면 크기 설정 및 기본 설정

```
5
6 import random, time, pygame, sys
7 from pygame.locals import *
8
9 FPS = 25
10 WINDOWWIDTH = 640
11 WINDOWHEIGHT = 480
12 BOXSIZE = 20
13 BOARDWIDTH = 10
14 BOARDHEIGHT = 20
15 BLANK = '.'
16
17 MOVESIDEWAYSFREQ = 0.15
18 MOVEDOWNFREQ = 0.1
19
20 XMARGIN = int((WINDOWWIDTH - BOARDWIDTH * BOXSIZE) / 2)
21 TOPMARGIN = WINDOWHEIGHT - (BOARDHEIGHT * BOXSIZE) - 5
```

게임의 화면 크기를 설정해 주고 후에 나올 코드들을 미리 선언한다 또한 MOVE~를 통해서 키를 누를 때 타이밍을 설정하는 상수를 지정한다.

## # 색상

```
23 #           R   G   B
24 WHITE      = (255, 255, 255)
25 GRAY       = (185, 185, 185)
26 BLACK      = (  0,  0,  0)
27 RED        = (155,  0,  0)
28 LIGHTRED   = (175, 20, 20)
29 GREEN      = (  0, 155,  0)
30 LIGHTGREEN = ( 20, 175, 20)
31 BLUE       = (  0,  0, 155)
32 LIGHTBLUE  = ( 20, 20, 175)
33 YELLOW     = (155, 155,  0)
34 LIGHTYELLOW = (175, 175, 20)
35 PINK       = (255,  0, 255)
36 LIGHTPINK  = (255, 204, 255)
37 ORANGE     = (255, 102,  0)
38 LIGHTORANGE = (255, 153,  0)
39 PURPLE     = (102,  0, 255)
40 LIGHTPURPLE = (204, 153, 255)
41
42
43 BORDERCOLOR = BLUE
44 BGCOLOR = BLACK
45 TEXTCOLOR = YELLOW
46 TEXTSHADOWCOLOR = GRAY
47 COLORS = ( BLUE, GREEN, RED, YELLOW, PINK, ORANGE, PURPLE)
48 LIGHTCOLORS = (LIGHTBLUE, LIGHTGREEN, LIGHTRED, LIGHTYELLOW, LIGHTPINK, LIGHTORANGE, LIGHTPURPLE)
49 assert len(COLORS) == len(LIGHTCOLORS) # each color must have light color
50
```

RGB컬러를 통해 색상을 지정하고 지정한 색상들을 리스트 배열하였다. 과제에서 각 피스들에 색상을 지정해주어야 한다고 나와있어 색상을 추가하여 지정하였다.

## #PIECES

```
49 S_SHAPE_TEMPLATE = [['.', '.', '.', '.'],
50                        ['.00.', '.'],
51                        ['.00.', '.'],
52                        ['.', '.'],
53                        ['.0.', '.'],
54                        ['.00.', '.'],
55                        ['.0.', '.'],
56                        ['.', '.'],
57                        ['.00.', '.'],
58                        ['.0.', '.'],
59                        ['.', '.']]
60
61 Z_SHAPE_TEMPLATE = [['.', '.', '.', '.'],
62                      ['.00.', '.'],
63                      ['.00.', '.'],
64                      ['.', '.'],
65                      ['.', '.'],
66                      ['.0.', '.'],
67                      ['.00.', '.'],
68                      ['.0.', '.'],
69                      ['.', '.']]
70
71 I_SHAPE_TEMPLATE = [['.', '.0.', '.'],
72                     ['.0.', '.'],
73                     ['.0.', '.'],
74                     ['.0.', '.'],
75                     ['.', '.'],
76                     ['.', '.'],
77                     ['.', '.'],
78                     ['.000.', '.'],
79                     ['.', '.']]
80
81 O_SHAPE_TEMPLATE = [['.', '.'],
82                     ['.00.', '.'],
83                     ['.00.', '.'],
84                     ['.', '.']]
85
86
87
88 J_SHAPE_TEMPLATE = [['.', '.'],
89                     ['.0.', '.'],
90                     ['.000.', '.'],
91                     ['.', '.'],
92                     ['.', '.'],
93                     ['.', '.'],
94                     ['.00.', '.'],
95                     ['.0.', '.'],
96                     ['.0.', '.'],
97                     ['.', '.'],
98                     ['.', '.'],
99                     ['.', '.'],
100                     ['.000.', '.'],
101                     ['.0.', '.'],
102                     ['.', '.'],
103                     ['.', '.'],
104                     ['.0.', '.'],
105                     ['.0.', '.'],
106                     ['.0.', '.'],
107                     ['.00.', '.'],
108                     ['.', '.']]
109
110 L_SHAPE_TEMPLATE = [['.', '.'],
111                     ['.0.', '.'],
112                     ['.000.', '.'],
113                     ['.', '.'],
114                     ['.', '.'],
115                     ['.', '.'],
116                     ['.0.', '.'],
117                     ['.0.', '.'],
118                     ['.00.', '.'],
119                     ['.', '.'],
120                     ['.', '.'],
121                     ['.000.', '.'],
122                     ['.0.', '.'],
123                     ['.', '.'],
124                     ['.', '.'],
125                     ['.00.', '.'],
126                     ['.00.', '.'],
127                     ['.0.', '.'],
128                     ['.', '.']]
129
130
131 T_SHAPE_TEMPLATE = [['.', '.'],
132                     ['.0.', '.'],
133                     ['.000.', '.'],
134                     ['.', '.'],
135                     ['.', '.'],
136                     ['.', '.'],
137                     ['.0.', '.'],
138                     ['.00.', '.'],
139                     ['.0.', '.'],
140                     ['.', '.'],
141                     ['.', '.'],
142                     ['.', '.'],
143                     ['.000.', '.'],
144                     ['.0.', '.'],
145                     ['.', '.'],
146                     ['.', '.'],
147                     ['.0.', '.'],
148                     ['.00.', '.'],
149                     ['.0.', '.'],
150                     ['.', '.']]
151
152 PIECES = {'S': S_SHAPE_TEMPLATE,
153          'Z': Z_SHAPE_TEMPLATE,
154          'J': J_SHAPE_TEMPLATE,
155          'L': L_SHAPE_TEMPLATE,
156          'I': I_SHAPE_TEMPLATE,
157          'O': O_SHAPE_TEMPLATE,
158          'T': T_SHAPE_TEMPLATE,}
```

테트리스 게임 중 나오는 피스들의 기본 모양과 회전 시 모양을 나타낸 코드이다. 이 때 o는 채워진 상자를, .은 빈칸을 나타낸다. 이를 PIECES라는 변수에 저장하였다.

## #main

```
160
161 def main():
162     global FPSLOCK, DISPLAYSURF, BASICFONT, BIGFONT
163     pygame.init()
164     FPSLOCK = pygame.time.Clock()
165     DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
166     BASICFONT = pygame.font.Font('freesansbold.ttf', 18)
167     BIGFONT = pygame.font.Font('freesansbold.ttf', 100)
168     pygame.display.set_caption('2020096608 JUNGJULBEEN')
169
170     showTextScreen('MY TETRIS')
171     while True: # game loop
172         if random.randint(0, 2) == 0:
173             pygame.mixer.music.load('Our_Lives_Past.mp3')
174         elif random.randint(0, 2) == 1:
175             pygame.mixer.music.load('Hover.mp3')
176         else:
177             pygame.mixer.music.load('Platform_9.mp3')
178         pygame.mixer.music.play(-1, 0.0)
179         runGame()
180         pygame.mixer.music.stop()
181         showTextScreen('Over :(')
182
```

게임에 대한 설정을 보여준다. 게임 창에 뜨는 폰트, 문구, 색상 등을 지정해주고 배경 음악을 나타낸다. 과제에서 3가지 음악 중 1개가 재생되어지도록 if 조건문을 이용하여 코드를 작성하였다.

## #게임 시작

```
184 def runGame():
185     # setup variables for the start of the game
186     board = getBlankBoard()
187     lastMoveDownTime = time.time()
188     lastMoveSidewaysTime = time.time()
189     lastFallTime = time.time()
190     start_time = time.time()
191     movingDown = False # note: there is no movingUp variable
192     movingLeft = False
193     movingRight = False
194     score = 0
195     level, fallFreq = calculateLevelAndFallFreq(score)
196
197
198     fallingPiece = getNewPiece()
199     nextPiece = getNewPiece()
200
```

게임 시작 전에 모두 초기화를 하였고, fallingPiece는 현재 화면에서 떨어지고 있는 블록을, nextPiece는 현재 블록이 떨어진 후 다음에 떨어지는 블록을 말한다.

## #Game loop

```
while True: # game loop
    if fallingPiece == None:
        # No falling piece in play, so start a new piece at the top
        fallingPiece = nextPiece
        nextPiece = getNewPiece()
        lastFallTime = time.time() # reset lastFallTime

    if not isValidPosition(board, fallingPiece):
        return # can't fit a new piece on the board, so game over
```

떨어지고 있는 피스가 없으면 다음 피스를 떨어뜨리고, 게임상의 화면, 즉 보드가 없을 경우 게임을 종료한다는 것을 의미한다.

## #일시정지

```
checkForQuit()
for event in pygame.event.get(): # event handling loop
    if event.type == KEYUP:
        if (event.key == K_p):
            # Pausing the game
            DISPLAYSURF.fill(BG_COLOR)
            pygame.mixer.music.stop()
            showTextScreen('Get a rest!') # pause until a key press
            pygame.mixer.music.play(-1, 0.0)
            lastFallTime = time.time()
            lastMoveDownTime = time.time()
            lastMoveSidewaysTime = time.time()
        elif (event.key == K_LEFT or event.key == K_a):
            movingLeft = False
        elif (event.key == K_RIGHT or event.key == K_d):
            movingRight = False
        elif (event.key == K_DOWN or event.key == K_s):
            movingDown = False
```

p키를 입력하였을 때 일시정지 되는 게임 화면에 대한 코드이다. 색상과 문구를 지정하였고 if 문을 이용하여 p를 눌렀을 때의 상황의 코드를 작성하였다.

## #LEFT, RIGHT

```
elif event.type == KEYDOWN:
    # moving the piece sideways
    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()

    elif (event.key == K_RIGHT or event.key == K_d) and isValidPosition(board, fallingPiece, adjX=1):
        fallingPiece['x'] += 1
        movingRight = True
        movingLeft = False
        lastMoveSidewaysTime = time.time()
```

방향 키 중 좌,우로 밀었을 때의 상황을 작성한 코드이다. 좌측 키를 눌렀을 때 피스가 x축에서 이동하는 양, 우측 키를 눌렀을 때 블록이 x축에서 이동하는 양을 정해 if 조건문을 활용하여 코드를 작성해 주었다. 방향 조작키 네 개에 대한 함수를 전체적으로 if조건문을 통해 작성한 것이다.

## #UP

```
# rotating the piece (if there is room to rotate)
elif (event.key == K_UP or event.key == K_W):
    fallingPiece['rotation'] = (fallingPiece['rotation'] + 1) % len(PIECES[fallingPiece['shape']])
    if not isValidPosition(board, fallingPiece):
        fallingPiece['rotation'] = (fallingPiece['rotation'] - 1) % len(PIECES[fallingPiece['shape']])
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']])
```

방향 키 중 위쪽으로 향하는 키를 눌렀을 때의 상황을 작성한 코드이다. 전에 지정한 피스들의 모양을 이용하여 이를 rotation과 shape를 통해 받으며 작성하였다. 위에서 이어지는 조건문에 일부이다.

## #피스들의 하강

```
# making the piece fall faster with the down key
elif (event.key == K_DOWN or event.key == K_S):
    movingDown = True
    if isValidPosition(board, fallingPiece, adjY=1):
        fallingPiece['y'] += 1
    lastMoveDownTime = time.time()

# move the current piece all the way down
elif event.key == K_SPACE:
    movingDown = False
    movingLeft = False
    movingRight = False
    for i in range(1, BOARDHEIGHT):
        if not isValidPosition(board, fallingPiece, adjY=i):
            break
    fallingPiece['y'] += i - 1
```

방향 조작키 중 아래 키를 누를 경우 빠르게 떨어지고, space키를 누를 경우 바로 바닥으로 떨어진다는 것을 나타내었다. 이 또한 방향 조작키에 대한 조건문의 일부이며 그 안에서 각각 조건문을 통하여 코드를 작성하였다.

## #키를 누르고 있을 때의 상황

```
# handle moving the piece because of user input
if (movingLeft or movingRight) and time.time() - lastMoveSidewaysTime > MOVESIDEWAYSFREQ:
    if movingLeft and isValidPosition(board, fallingPiece, adjX=-1):
        fallingPiece['x'] -= 1
    elif movingRight and isValidPosition(board, fallingPiece, adjX=1):
        fallingPiece['x'] += 1
    lastMoveSidewaysTime = time.time()

if movingDown and time.time() - lastMoveDownTime > MOVEDOWNFREQ and isValidPosition(board, fallingPiece, adjY=1):
    fallingPiece['y'] += 1
    lastMoveDownTime = time.time()

# let the piece fall if it is time to fall
if time.time() - lastFallTime > fallFreq:
    # see if the piece has landed
    if not isValidPosition(board, fallingPiece, adjY=1):
        # falling piece has landed, set it on the board
        addToBoard(board, fallingPiece)
        score += removeCompleteLines(board)
        level, fallFreq = calculateLevelAndFallFreq(score)
        fallingPiece = None
    else:
        # piece did not land, just move the piece down
        fallingPiece['y'] += 1
        lastFallTime = time.time()
```

방향 조작키를 오래 누를 때의 상황을 작성한 코드이고 자연스럽게 피스가 움직이도록 작성하였다.

## #score , level

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

pygame.display.update()
FPSLOCK.tick(FPS)
```

스크린에 표시되는 score와 level의 색상을 지정해주었다.

## #makeTextObjis

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

Render method를 호출 한 후에 Surface와 Rect객체를 반환하는 코드를 작성하였다.

## #showTextScreen

```
def showTextScreen(text):
    # This function displays large text in the
    # center of the screen until a key is pressed.
    # Draw the text drop shadow
    titleSurf, titleRect = makeTextObjs(text, BIGFONT, TEXTSHADOWCOLOR)
    titleRect.center = (int(WINDOWWIDTH / 2), int(WINDOWHEIGHT / 2))
    DISPLAYSURF.blit(titleSurf, titleRect)

    # Draw the text
    titleSurf, titleRect = makeTextObjs(text, BIGFONT, TEXTCOLOR)
    titleRect.center = (int(WINDOWWIDTH / 2) - 3, int(WINDOWHEIGHT / 2) - 3)
    DISPLAYSURF.blit(titleSurf, titleRect)

    # Draw the additional "Press any key to play! pause key is p" text.
    pressKeySurf, pressKeyRect = makeTextObjs('Press any key to play! pause key is p', BASICFONT, YELLOW)
    pressKeyRect.center = (int(WINDOWWIDTH / 2), int(WINDOWHEIGHT / 2) + 100)
    DISPLAYSURF.blit(pressKeySurf, pressKeyRect)

    while checkForKeyPress() == None:
        pygame.display.update()
        FPSLOCK.tick()
```

위에 언급한 makeTextObjs함수로부터 Surf와 Rect를 받아서 스크린에 표시되어지는 글자를 나타내는 코드를 작성하였다.

## #score

```
366
367 def calculateLevelAndFallFreq(score):
368     # Based on the score, return the level the player is on and
369     # how many seconds pass until a falling piece falls one space.
370     level = int(score / 10) + 1
371     fallFreq = 0.27 - (level * 0.02)
372     return level, fallFreq
373
374
```

점수를 계산하는 함수를 설정했다. 또, 점수에 따라 레벨을 올리고 피스의 속도를 조정해주는 함수이다.



## #NewPiece

```
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': setColor(shape)}  
    return newPiece
```

새로운 피스에 대한 함수를 정의했다. 모양과 위치, 그리고 색상을 정의하였다. 피스의 shape는 앞에서 정의했던 함수를 이용하였고, 이 리스트를 랜덤으로 나오도록 랜덤함수를 이용하였다. Rotation 또한 랜덤함수를 사용하였으며 color는 주어진 과제에서 각 피스당 색상이 지정되도록 하였기에 setColor(shape)라는 함수를 이용하였다.

## #setColor(shape)

```
386 def setColor(shape):  
387  
388     color = 0 #BLUE  
389  
390     if shape == 'S':  
391         color = 0  
392     if shape == 'Z':  
393         color = 1  
394     if shape == 'J':  
395         color = 2  
396     if shape == 'L':  
397         color = 3  
398     if shape == 'I':  
399         color = 4  
400     if shape == "O":  
401         color = 5  
402     if shape == 'T':  
403         color = 6  
404  
405     return color  
406
```

각 피스들에 색상이 지정되도록 앞에서 리스트 한 함수들을 활용하여 코드를 작성하였다.

## #BlankBoard

```
415
416 def getBlankBoard():
417     # create and return a new blank board data structure
418     board = []
419     for i in range(BOARDWIDTH):
420         board.append([BLANK] * BOARDHEIGHT)
421     return board
```

새로운 보드를 반환하도록 설정한 코드이다.

## #score and level text

```
501
502 def drawStatus(score, level):
503     # draw the score text
504     scoreSurf = BASICFONT.render('Score: %s' % score, True, TEXTCOLOR)
505     scoreRect = scoreSurf.get_rect()
506     scoreRect.topleft = (WINDOWWIDTH - 150, 20)
507     DISPLAYSURF.blit(scoreSurf, scoreRect)
508
509     # draw the level text
510     levelSurf = BASICFONT.render('Level: %s' % level, True, TEXTCOLOR)
511     levelRect = levelSurf.get_rect()
512     levelRect.topleft = (WINDOWWIDTH - 150, 50)
513     DISPLAYSURF.blit(levelSurf, levelRect)
514
```

Score text와 level text 또한 Surf 와 Rect를 이용하여 작성되었다.

## #next

```
528
529 def drawNextPiece(piece):
530     # draw the "next" text
531     nextSurf = BASICFONT.render('Next:', True, TEXTCOLOR)
532     nextRect = nextSurf.get_rect()
533     nextRect.topleft = (WINDOWWIDTH - 120, 80)
534     DISPLAYSURF.blit(nextSurf, nextRect)
535     # draw the "next" piece
536     drawPiece(piece, pixelx=WINDOWWIDTH-120, pixely=100)
537
```

Next 또한 surf와 rect를 통해 문구를 보여주고 다음으로 나올 피스를 보여준다.

감사합니다.

-2020096608 정설빈 -