

오픈소스SW 과제중심수업 보고서

ICT융합학부 미디어테크 전공
2020051430 김상민

GitHub repository 주소 : <https://github.com/verdandi144/osw>

1. 기본 세팅

#화면 크기 조정

```
FPS = 25
WINDOWWIDTH = 640
WINDOWHEIGHT = 480
BOXSIZE = 20
BOARDWIDTH = 10
BOARDHEIGHT = 20
BLANK = '.'
```

```
MOVESIDEWAYSFREQ = 0.15
MOVEDOWNFREQ = 0.1
```

```
XMARGIN = int((WINDOWWIDTH - BOARDWIDTH * BOXSIZE) / 2)
TOPMARGIN = WINDOWHEIGHT - (BOARDHEIGHT * BOXSIZE) - 5
```

#색상(7가지 블록 + 배경색상)

```
#      R   G   B
WHITE   = (255, 255, 255)
GRAY    = (185, 185, 185)
BLACK   = (  0,  0,  0)
RED     = (155,  0,  0)
LIGHTRED = (175, 20, 20)
ORANGE  = (255, 153,  0)
LIGHTORANGE = (255, 202, 125)
GREEN   = (  0, 155,  0)
LIGHTGREEN = ( 20, 175, 20)
BLUE    = (  0,  0, 155)
LIGHTBLUE = ( 20, 20, 175)
PURPLE  = (98, 0, 163)
LIGHTPURPLE = (202, 125, 255)
PINK    = (255, 51, 255)
LIGHTPINK = (255, 151, 255)
YELLOW  = (155, 155,  0)
LIGHTYELLOW = (175, 175, 20)
```

```
BORDERCOLOR = BLUE
BGCOLOR = BLACK
TEXTCOLOR = YELLOW
TEXTSHADOWCOLOR = LIGHTYELLOW
COLORS = (RED, ORANGE, YELLOW, GREEN, BLUE, PURPLE, PINK)
```

```
LIGHTCOLORS = (LIGHTRED, LIGHTORANGE, LIGHTYELLOW, LIGHTGREEN, LIGHTBLUE,
LIGHTPURPLE, LIGHTPINK)
assert len(COLORS) == len(LIGHTCOLORS) # each color must have light color
```

```
TEMPLATEWIDTH = 5
TEMPLATEHEIGHT = 5
```

#블록 모양 지정해주기

```
S_SHAPE_TEMPLATE = [['.....',
    '.....',
    '..OO..',
    '..OO..',
    '.....'],
    [['.....',
    '..O...',
    '..OO..',
    '..O...',
    '.....']]]
```

```
Z_SHAPE_TEMPLATE = [['.....',
    '.....',
    '..OO..',
    '..OO..',
    '.....'],
    [['.....',
    '..O...',
    '..OO..',
    '..O...',
    '.....']]]
```

```
I_SHAPE_TEMPLATE = [['..O..',
    '..O..',
    '..O..',
    '.....'],
    [['.....',
    '.....',
    '.....',
    'OOOO..',
    '.....',
    '.....']]]
```

```
O_SHAPE_TEMPLATE = [['.....',
    '.....',
    '..OO..',
    '..OO..',
    '.....']]]
```

```
J_SHAPE_TEMPLATE = [['.....',
    '.....',
    '.....',
    '.....'],
    [['.....',
    '.....',
    '..OO..',
    '..O..',
    '..O..',
    '.....'],
    [['.....',
    '.....',
    '.....',
    '.....',
    '.....',
    '.....']]]
```

```

    '...O.',
    '....'],
    ['....',
     '..O..',
     '..O..',
     '..OO..',
     '....']]

```

```
L_SHAPE_TEMPLATE = [['.....',
```

```

    '...O.',
    '..OOO.',
    '....',
    '....'],
    ['....',
     '..O..',
     '..O..',
     '..OO.',
     '....'],
    ['....',
     '....',
     '..OOO.',
     '..O..',
     '....'],
    ['....',
     '..OO.',
     '..O..',
     '..O..',
     '....']]

```

```
T_SHAPE_TEMPLATE = [['.....',
```

```

    '...O.',
    '..OOO.',
    '....',
    '....'],
    ['....',
     '..O..',
     '..OO.',
     '..O..',
     '....'],
    ['....',
     '....',
     '..OOO.',
     '..O..',
     '....'],
    ['....',
     '..O..',
     '..OO.',
     '..O..',
     '....']]

```

#블록들 이름 지정

```

PIECES = {'S': S_SHAPE_TEMPLATE,
          'Z': Z_SHAPE_TEMPLATE,
          'J': J_SHAPE_TEMPLATE,
          'L': L_SHAPE_TEMPLATE,
          'I': I_SHAPE_TEMPLATE,
          'O': O_SHAPE_TEMPLATE,
          'T': T_SHAPE_TEMPLATE}

```

#컬러 지정 리스트

```
shape_color = ['S', 'Z', 'J', 'L', 'I', 'O', 'T']
```

2. 각 함수들의 역할

메인 함수

```
def main()
```

게임 실행 시 함수

```
def runGame()
```

프로그램 종료 시 함수

```
def terminate():
```

KeyUp / KeyDown 이벤트 큐 함수

```
def checkForKeyPress():
```

키가 눌리기 전까지 화면 중앙에 큰 텍스트를 표시하는 함수

```
def showTextScreen(text):
```

Score를 통해 Level(레벨)과 fallFreq(하강속도)를 반환하는 함수

```
def calculateLevelAndFallFreq(score):
```

특정한 색상을 지닌 블록을 종류는 랜덤하게 새로 생성하는 함수

```
def getNewPiece():
```

우측 상황판에 블록의 위치, 모양(회전한 모양) 표시하는 함수

```
def addToBoard(board, piece):
```

새로운 빈 상황판 생성 함수

```
def getBlankBoard():
```

상황판 너비와 높이 반환 함수

```
def isOnBoard(x, y):  
    return x >= 0 and x < BOARDWIDTH and y < BOARDHEIGHT
```

만약 블록이 게임판 안에 있고 충돌 안 한다면 TRUE 반환하는 함수

```
def isValidPosition(board, piece, adjX=0, adjY=0):
```

만약 블록이 라인 전체를 채우면 TRUE 반환하는 함수

```
def isCompleteLine(board, y):
```

#전체 한 줄이 다 채워졌다면 그 줄 없애고 위의 줄 아래로 내리는 함수

```
def removeCompleteLines(board):
```

#박스 위치 결정 함수

```
def convertToPixelCoords(boxx, boxy):
```

#블록을 구성하는 4개의 박스 함수

```
def drawBox(boxx, boxy, color, pixelx=None, pixely=None):
```

#배경, 테두리 색깔

```
def drawBoard(board):
```

#플레이 시간 , 스코어, 레벨 상태 함수

```
def drawStatus(playtime, score, level):
```

#블록을 생성하는 조각들 생성

```
def drawPiece(piece, pixelx=None, pixely=None):
```

#next 텍스트 생성

```
def drawNextPiece(piece):
```