

## 오픈소스를 활용한 게임 만들기

2023051458 이민섭

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

1-1. 주어진 3가지 음악 중 하나가 재생되도록 하기 위해 main()의 코드 일부를 다음과 같이 수정함.

(수정 전)

```
while True: # game loop
    if random.randint(0, 1) == 0:
        pygame.mixer.music.load('tetrisb.mid')
    else:
        pygame.mixer.music.load('tetrisc.mid')
```

(수정 후)

```
while True: # game loop
    i = random.randint(0, 2)
    if i == 0:
        pygame.mixer.music.load('Hover.mp3')
    elif i == 1:
        pygame.mixer.music.load('Our_lives_Past.mp3')
    else:
        pygame.mixer.music.load('Platform_9.mp3')
```

1-2,3. 상태창 이름과 게임 시작 화면의 문구를 변경하기 위해 main()의 다음 코드를 수정함.

(수정 전)

```
pygame.display.set_caption('Tetromino')
```

```
showTextScreen('Tetromino')
```

(수정 후)

```
pygame.display.set_caption('2023051458_이민섭')
```

```
showTextScreen('Tetromino')
```

1-4. 게임 시작 화면의 문구 및 배경색을 변경하기 위해 다음 코드를 수정함.

(수정 전)

```
TEXTCOLOR = WHITE
TEXTSHADOWCOLOR = GRAY
```

(수정 후)

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

1-5. 게임 경과 시간을 표시하기 위해 다음과 같이 코드를 수정 또는 추가함

1-5-1. runGame()에 다음 코드를 추가함

```
startTime = time.time()
```

1-5-2. runGame()의 drawStatus() 호출 코드의 인자를 수정함.

(수정 전)

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

(수정 후)

```
drawStatus(score, level, time.time() - startTime)
```

### 1-5-3. drawStatus()의 인자를 수정함

(수정 전)

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

(수정 후)

```
drawStatus(score, level, elapsedTime)
```

### 1-5-4. drawStatus()에 경과 시간을 표시하는 코드를 추가함

```
timeSurf = BASICFONT.render('Time: %s' % int(elapsedTime), True, TEXTCOLOR)
timeRect = timeSurf.get_rect()
timeRect.topleft = (WINDOWWIDTH - 600, 20)
DISPLAYSURF.blit(timeSurf, timeRect)
```

1-6. 각각의 테트리스 조각에 고유한 색을 지정하기 위해 다음과 같이 코드를 수정 또는 추가함. (runtime error가 계속 발생하여 수정, 추가한 코드를 ipynb 파일에 반영하지 않음)

#### 1-6-1. PIECES를 다음과 같이 수정함.

(수정 전)

```
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}
```

(수정 후)

```
PIECES = {'S': (LIGHTGREEN, S_SHAPE_TEMPLATE),
          'Z': (RED, Z_SHAPE_TEMPLATE),
          'J': (LIGHTYELLOW, J_SHAPE_TEMPLATE),
          'L': (BLUE, L_SHAPE_TEMPLATE),
          'I': (LIGHTBLUE, I_SHAPE_TEMPLATE),
          'O': (YELLOW, O_SHAPE_TEMPLATE),
          'T': (LIGHTRED, T_SHAPE_TEMPLATE)}
```

#### 1-6-2. drawPiece()를 다음과 같이 수정함.

(수정 전)

```
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))
```

(수정 후)

```
shapeToDraw = PIECES[piece['shape']][1]
colorToDraw = PIECES[piece['shape']][0]
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[piece['rotation']][y][x] != BLANK:
            drawBox(None, None, colorToDraw, pixelx + (x * BOXSIZE), pixely + (y * BOXSIZE))
```

1-6-3. drawBox()를 다음과 같이 수정함.

(수정 전)

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

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

(수정 후)

```
pygame.draw.rect(DISPLAYSURF, color, (pixelx + 1, pixely + 1, BOXSIZE - 1, BOXSIZE - 1))
```

1-6-4. addToBoard()와 isValidPosition()의 코드 중 일부를 다음과 같이 수정함.

(수정 전)

```
PIECES[piece['shape']][piece['rotation']][y][x]
```

(수정 후)

```
PIECES[piece['shape']][1][piece['rotation']][y][x]
```

## 2. 각 함수에 대한 설명

main(): 게임을 초기화하고 루프 시작, showTextScreen()을 호출하여 게임 시작 화면을 표시, runGame() 함수를 호출하여 실제 게임 루프를 실행, 게임이 종료되면 showTextScreen()을 호출하여 게임 종료 화면 표시

runGame(): 게임 루프 시작, 게임 보드를 초기화하고 현재 조작 중인 블록과 다음 블록 설정, 게임 중 사용자의 입력을 처리하고 블록의 이동 및 회전을 관리, 블록이 떨어지는 동작을 처리하고 블록이 땅에 닿았을 때 보드에 추가, 보드에서 완전한 줄을 제거하고 점수 계산, 화면에 게임 상태를 업데이트

makeTextObjs(): 주어진 텍스트, 글꼴, 색상으로 텍스트 서피스 생성

terminate(): 게임, 프로그램 종료

checkForKeyPress(): 사용자가 키를 눌렀는지 떴는지 확인

showTextScreen(): makeTextObjs()를 호출하여 화면 중앙에 텍스트 표시, 사용자의 키 입력 대기

checkForQuit(): QUIT, ESCAPE 키 이벤트가 발생하면 terminate()를 호출

calculateLevelAndFallFreq(): 현재 점수를 기반으로 플레이어의 레벨과 블록이 떨어지는 빈도를 계산

getNewPiece(): 랜덤한 모양, 색상의 새로운 블록을 생성

addToBoard(): 보드에 현재 블록 추가

getBlankBoard(): 빈 보드를 생성하여 반환

isOnBoard(): 주어진 좌표가 보드에 있는지 확인  
isValidPosition(): 현재 블록의 위치가 유효한지 확인  
isCompleteLine(): 주어진 좌표의 줄이 완전히 채워졌는지 확인  
removeCompleteLines(): 보드에서 완전한 줄을 제거하고 제거된 줄 수를 반환  
convertToPixelCoords(): 보드의 좌표를 화면 픽셀의 좌표로 변환  
drawBox(): 화면에 하나의 블록을 그림  
drawBoard(): 게임 보드를 화면에 그림  
drawStatus(): 점수, 레벨, 경과 시간을 화면에 그림  
drawPiece(): 현재 블록을 화면에 그림  
drawNextPiece(): 다음 블록을 화면에 그림, “Next:” 텍스트 표시