

오픈소스SW 과제 중심 수업1 보고서 (Tetromino)

2021063763 강홍구

```

170 def main():
171     global FPSLOCK, DISPLAYSURF, BASICFONT, BIGFONT
172     pygame.init()
173     FPSLOCK = pygame.time.Clock()
174     DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
175     BASICFONT = pygame.font.Font('freesansbold.ttf', 18)
176     BIGFONT = pygame.font.Font('freesansbold.ttf', 100)
177     pygame.display.set_caption('2021063763_강홍구')
178
179     showTextScreen('MY TETRIS')
180     while True: # game loop
181         if random.randint(0, 1) == 0:
182             pygame.mixer.music.load('Hover.mp3')
183         else:
184             pygame.mixer.music.load('Hover.mp3')
185         pygame.mixer.music.play(-1, 0.0)
186         runGame()
187         pygame.mixer.music.stop()
188         showTextScreen('Over :(')

```

1. #182, 184번째 줄을 'Hover.mp3'로 수정하여 'Hover' 배경음악 1개가 재생됩니다.
2. #177번째 줄에서 상태창 이름을 학번_이름으로 수정했습니다.
3. #179번째 줄에서 게임시작화면의 문구를 MY TETRIS로 변경했습니다.

```

36 BORDERCOLOR = BLUE
37 BGCOLOR = BLACK
38 TEXTCOLOR = YELLOW
39 TEXTSHADOWCOLOR = YELLOW
40 COLORS = (BLUE, GREEN, RED, YELLOW)
41 LIGHTCOLORS = (LIGHTBLUE, LIGHTGREEN, LIGHTRED, LIGHTYELLOW)
42 assert len(COLORS) == len(LIGHTCOLORS) # each color must have light color

```

4. #38번째 줄에서 TEXTCOLOR = YELLOW로, #318번째 줄에서 "Yellow"로 변경했습니다.

```

353 pressKeySurf, pressKeyRect = makeTextObjs('Press any key to play! pause key is p', BASICFONT, TEXTCOLOR)
354 pressKeyRect.center = (int(WINDOWWIDTH / 2), int(WINDOWHEIGHT / 2) + 100)
355 DISPLAYSURF.blit(pressKeySurf, pressKeyRect)
219 checkForQuit()
220 for event in pygame.event.get(): # event handling loop
221     if event.type == KEYUP:
222         if (event.key == K_p):
223             # Pausing the game
224             DISPLAYSURF.fill(BGCOLOR)
225             pygame.mixer.music.stop()
226             showTextScreen('Get a reset!') # pause until a key press
227             pygame.mixer.music.play(-1, 0.0)
228             lastFallTime = time.time()
229             lastMoveDownTime = time.time()
230             lastMoveSidewaysTime = time.time()

```

+ 슬라이드에 나온대로 문구 바꾸기

게임오버화면 #188 'Over :('

일시정지화면 #353 'Press any key to play! pause key is p'. #226 'Get a reset!'

```

191 def runGame():
192     global start_time
193     # setup variables for the start of the game
194     board = getBlankBoard()
195     lastMoveDownTime = time.time()
196     lastMoveSidewaysTime = time.time()
197     lastFallTime = time.time()
198     movingDown = False # note: there is no movingUp variable
199     movingLeft = False
200     movingRight = False
201     score = 0
202     level, fallFreq = calculateLevelAndFallFreq(score)
203     start_time = time.time()
204
497     current_time = int(time.time() - start_time)
498     timeSurf = BASICFONT.render('Play Time: %s sec' % current_time, True, TEXTCOLOR)
499     timeRect = timeSurf.get_rect()
500     timeRect.topleft = (WINDOWWIDTH - 600, 20)
501     DISPLAYSURF.blit(timeSurf, timeRect)

```

5. #192번째 줄에서 start_time을 global로 선언한 후
#203번째 줄에서 start_time = time.time()을 작성하여 start_time을 시작시간으로 만들었습니다. 그 후

#497 ~ 501번째 줄을 새로 추가했습니다. current_time = int(time.time() - start_time)으로 작성하여 현재시간에서 시작시간을 빼서 경과된 시간을 current_time으로 하였습니다. 그 후 위 코드를 참고하여 BASICFONT.render 메서드를 사용하여 텍스트를 pygame로 렌더링 후 timeRect 변수에 저장하여 왼쪽 위 모서리에 나타나도록 했습니다.

```

158 PIECE_COLORS = {
159     'S': LIGHTBLUE,
160     'Z': LIGHTRED,
161     'J': LIGHTGREEN,
162     'L': RED,
163     'I': BLUE,
164     'O': YELLOW,
165     'T': GREEN
166 }

```

6. #158~166번째 줄에 PIECE_COLORS를 새로 생성하고 각 블록별로 색상을 지정했습니다. 3개 블록은 lightcolor를 사용했습니다.

```

378 def getNewPiece():
379     # return a random new piece in a random rotation and color
380     shape = random.choice(list(PIECES.keys()))
381     color = PIECE_COLORS[shape]
382     newPiece = {'shape': shape,
383                 'rotation': random.randint(0, len(PIECES[shape]) - 1),
384                 'x': int(BOARDWIDTH / 2) - int(TEMPLATEWIDTH / 2),
385                 'y': -2, # start it above the board (i.e. less than 0)
386                 'color': color}
387     return newPiece

```

```

459 def drawBox(boxx, boxy, color, pixelx=None, pixely=None):
460     # draw a single box (each tetromino piece has four boxes)
461     # at xy coordinates on the board. Or, if pixelx & pixely
462     # are specified, draw to the pixel coordinates stored in
463     # pixelx & pixely (this is used for the "Next" piece).
464     if color == BLANK:
465         return
466     if pixelx == None and pixely == None:
467         pixelx, pixely = convertToPixelCoords(boxx, boxy)
468     pygame.draw.rect(DISPLAYSURF, color, (pixelx + 1, pixely + 1, BOXSIZE - 1, BOXSIZE - 1))
469     pygame.draw.rect(DISPLAYSURF, color, (pixelx + 1, pixely + 1, BOXSIZE - 4, BOXSIZE - 4))

```

6. 그 후 getNewPiece()함수에 color부분에 PIECE_COLORS[shape]을 저장하여 사용했습니다. 또 drawBox 함수에 #468,469번째 줄에 COLORS[color], LIGHTCOLORS[color] 부분을 color로 수정하였습니다.

각 함수의 역할

main():

게임을 초기화하고, 메인 게임 루프를 실행하며, 게임 종료 후 다시 시작할 수 있도록 합니다.

runGame():

실제 게임이 진행되는 함수입니다.

게임 보드와 현재 떨어지는 조각을 초기화하고, 게임 루프를 실행합니다.

사용자 입력을 처리하고, 게임 로직을 업데이트하며, 화면을 다시 그립니다.

makeTextObjs(text, font, color):

텍스트를 렌더링하고 해당 텍스트의 표면과 사각형 객체를 반환합니다.

terminate():

게임을 종료하는 함수로 Pygame과 프로그램을 종료합니다.

checkForKeyPress():

키 입력을 확인하는 함수입니다. 게임 일시 정지나 재시작 시에 사용됩니다.

showTextScreen(text):

화면 중앙에 큰 텍스트를 표시하고, 사용자가 키 입력을 기다리는 함수입니다.

게임 시작, 일시 정지, 게임 종료 시에 사용됩니다.

checkForQuit():

QUIT 이벤트와 ESC 키 입력을 확인하고, 게임을 종료합니다.

calculateLevelAndFallFreq(score):

점수에 따라 현재 레벨과 블록이 떨어지는 속도를 계산합니다..

getNewPiece():

새로운 테트리스 블록을 생성하여 반환합니다.

랜덤한 모양과 모양별로 지정된 색상으로 조각을 만듭니다.

addToBoard(board, piece):

현재 보드에 블록을 추가합니다.

조각의 위치와 모양에 따라 보드의 해당 위치를 채웁니다.

getBlankBoard():

게임 시작 시 빈 게임 보드를 생성합니다.

isOnBoard(x, y):

주어진 좌표가 보드 내에 있는지 확인합니다.

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

블록이 현재 위치 또는 이동한 위치에서 유효한지 확인합니다.

보드 경계와 충돌하지 않으며 다른 블록과 충돌하지 않는지 확인합니다.

isCompleteLine(board, y):

주어진 행이 완전히 채워졌는지 확인합니다.

removeCompleteLines(board):

완전히 채워진 모든 행을 제거하고, 위의 행들을 아래로 이동시킵니다.

제거된 행의 수를 반환합니다.

convertToPixelCoords(boxx, boxy):

보드 좌표를 화면 좌표로 변환합니다.

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

개별 블록을 화면에 그립니다.

조각을 구성하는 각 블록을 그리는 데 사용됩니다.

drawBoard(board):

게임 보드를 화면에 그립니다.

보드의 경계와 각 블록을 그립니다.

drawStatus(score, level):

현재 점수, 레벨, 시간(초)를 화면에 그립니다.

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

현재 조각을 화면에 그립니다.

drawNextPiece(piece):

다음에 떨어질 블록을 화면 오른쪽 상단에 그립니다.

함수 호출 순서 및 호출 조건

1. 프로그램 시작 및 초기화

main() 호출

showTextScreen('TETRIS') 호출

checkForKeyPress() 호출

runGame() 호출

2. 메인 게임 루프 (runGame())

getBlankBoard() 호출

getNewPiece() 호출

calculateLevelAndFallFreq(score) 호출

fallingPiece = getNewPiece() 호출

nextPiece = getNewPiece() 호출

while True: 게임 루프 시작

if fallingPiece == None: 블록이 없는 경우 새로운 블록 생성합니다.

nextPiece = getNewPiece() 호출

if not isValidPosition(board, fallingPiece): (새 블록을 놓을 곳이 없으면 게임 종료)

checkForQuit() 호출 (종료 이벤트 확인)

showTextScreen('Get a reset!') (중단 키가 눌렸을 때 'Get a reset!' 문구를 띄웁니다.)

여러 가지 elif 문들은 사용자 키 입력에 따라 블록이 움직이거나 회전을 하도록 합니다.

그때 isValidPosition(board, piece, adjX, adjY) 함수가 호출되어 사용됩니다.

if time.time() - lastFallTime > fallFreq:

블록이 다음 떨어지는 시간이 되었을 때 다음 if문이 실행됩니다.

if not isValidPosition(board, fallingPiece, adjY=1): 블록이 바닥에 닿았을 때:

addToBoard(board, fallingPiece): 함수를 호출하여 블록을 추가합니다.

score += removeCompleteLines(board): 채워진 행을 제거 후 점수를 업데이트합니다.

level, fallFreq = calculateLevelAndFallFreq(score): 점수에 따라 레벨과 블록이 떨어지는 속도를 다시 계산합니다.

화면 그리는 부분입니다.

drawBoard(board) 호출

drawBox(boxx, boxy, color, pixelx, pixely) drawBoard 내에서 호출

drawStatus(score, level) 호출

drawNextPiece(piece) 호출

if fallingPiece != None: 떨어지는 블록이 있다면:

drawPiece(piece, pixelx, pixely) 호출

3. 게임 종료 및 재시작

showTextScreen('Game Over') 호출 (게임 종료 화면 표시)

checkForKeyPress() 호출 (사용자 입력 대기)

main() 재시작

4. 프로그램 종료

checkForQuit() 호출 (종료 이벤트 확인)

terminate() 호출 (프로그램 종료)