

과제중심 수업 1주차 보고서

2021044375 유준상

<https://github.com/ymyoo4258/osw>

1. 현재 테트리스 게임의 배경음악을 주어진 3개의 음악 중 1개가 재생되도록 수정

180줄, 182줄

`pygame.mixer.music.load('빈칸')` 빈칸에 주어진 3개의 음악 파일 중 하나를 작성하여 수정함

2. 상태창 이름을 학번_이름 으로 수정

174줄

`pygame.display.set_caption('빈칸')` 빈칸에 학번_이름을 적어 수정함

3. 게임시작화면의 문구를 MY TETRIS으로 변경

177줄

`showTextScreen('빈칸')` 빈칸에 MY TETRIS를 적어 수정함

4. 게임시작화면의 문구 및 배경색을 노란색으로 변경

38줄, 40줄, 176줄

`TEXTCOLOR` 상수를 `YELLOW` 상수를 사용하여 수정함

`TITLECOLOR = YELLOW` 상수를 추가함

`DISPLAYSURF.fill(TITLECOLOR)`를 추가해 게임시작화면의 배경색을 노란색으로 변경

5. 게임 경과 시간을 초 단위로 표시 (새 게임 시작시 0으로 초기화 되어야 함)

167줄, 170줄, 186줄, 517~525줄

`global start_tick`을 추가함

`start_tick = pygame.time.get_ticks()` 로 초기화 값을 만들어주고

`def drawTime()` 함수에서 시간을 측정하는 변수값을 만들고 `render` 함수로 화면에 표시함

추가로 게임오버 부분에 `start_tick += elapsed_tick`을 추가하여 게임오버 후 새 게임 시작 시 시간이 0으로 초기화 되게 함

6. 7개의 블록이 각각 고유의 색을 갖도록 코드를 수정하거나 추가

158~164줄, 383줄

`PIECES_COLOR` 라는 상수 딕셔너리를 하나 만들어서 `newPiece` 딕셔너리 안의 'color' key값에 `PIECES_COLOR[shape]`로 `shape`에 대응하는 color 값을 출력하게 해주었음

>> 각 함수의 역할

main() 함수: 게임이 켜진 후 메인 화면을 띄워주고 runGame()을 실행시켜준다. 후에 runGame()이 종료되었을 때 Game Over 스크린을 띄워주고 플레이어가 키를 누를 시 게임이 재시작되게 해준다.

runGame() 함수: 테트리스가 작동하는 사실상 제일 메인이 되는 함수이며 while True 로 게임의 루프를 만든 후 떨어지는 블록과 다음 블록을 제어하고, 플레이어가 게임 오버 조건을 만족하였는지, for event in pygame.event.get() 루프를 통해 플레이어의 키 입력을 받아 그에 맞는 동작을 하게 해준다.

makeTextObjs() 함수: text 출력을 위해 필요한 Surface와 Rect 개체를 코드를 작성할 때 편하게 쓰기 위해 만든 함수

terminate() 함수: 게임과 시스템을 끝낼 때 사용하는 함수

checkForKeyPress() 함수: 플레이어가 키를 눌렀는 지 확인하는 함수이다. checkForQuit() 함수로 QUIT 이벤트가 발생하는 지도 확인하여 종료 시켜주기도 한다.

showTextScreen() 함수: 시작 화면, 게임 종료 화면, 중지 화면에 사용될 화면을 만드는 역할을 한다.

checkForQuit() 함수: 프로그램이 종료될 이벤트에 있을 때 종료시켜주는 역할을 한다.

calculateLevelAndFallFreq() 함수: 플레이어가 지운 라인 수를 받아 레벨을 늘리고 레벨에 따른 블록이 떨어지는 속도를 조절하는 역할을 한다.

getNewPiece() 함수: 무작위로 새로운 블록을 생성하는 역할을 한다.

addToBoard() 함수: 블록이 끝까지 내려왔을 때 board에 추가하여 블록이 쌓이게 해주는 역할을 한다.

getBlankBoard() 함수: 새로운 보드를 만드는 함수이다.

isOnBoard() 함수: x ,y 값이 보드를 넘어갔는지 확인하는 함수이다.

isValidPosition() 함수: 블록이 보드에 있는 블록과 비교하여 유효한 위치에 있는지 확인하는 함수이다.

isCompleteLine() 함수: 한 줄이 완성되었는지 확인하는 함수이다.

removeCompleteLines() 함수: 완성된 줄을 없애주는 함수이다.

convertToPixelCoords() 함수: 보드의 박스 x, y 위치를 픽셀 위치로 변환시켜주는 함수이다.

drawBox() 함수: pygame.draw.rect 함수를 이용하여 박스를 그려주는 함수이다.

drawBoard() 함수: 보드를 그려주는 함수이다.

drawStatus() 함수: 현재 상태를 표시해주는 함수이다.

drawPiece() 함수: drawBox() 함수를 이용하여 블록 조각을 그려주는 함수이다.

drawNextPiece() 함수: 다음 블록 조각을 그려주는 함수이다.

+ drawTime() 함수: 시간을 측정하여 화면에 표시해주는 함수이다.

>> 함수의 호출 순서 및 호출 조건

main() → runGame() 후

board = getBlankBoard() 함수로 새 보드를 만들고

fallingPiece = getNewPiece()

nextPiece = getNewPiece() 함수로 새 블록 조각을 생성 후 게임 루프에서

fallingPiece가 None인지 확인 후 None 이면 fallingPiece = nextPiece로 넘기고

nextPiece = getNewPiece() 함수로 새 블록 조각을 받는다.

if not isValidPosition(board, fallingPiece) 조건문으로 fallingPiece가 유효한 위치에 있는 지 확인하고

checkForQuit() 함수로 종료 조건에 있는 지 확인한다.

for event in pygame.event.get() 반복문으로 event에 플레이어의 입력값을 받는다.

입력값이 pause일 때 showTextScreen('Pause') 함수로 중지 화면을 띄운다.

입력값이 ←나 → 일 때 조건문에 isValidPosition()으로 유효한 위치에 있는 지도 추가로 확인해준다.

블록 조각이 아무런 키 입력 없이 자연스럽게 떨어질 때도 isValidPosition()을 확인해주고

유효하다면 addToBoard() 함수로 떨어지는 블록을 board에 추가해주고 score값에

removeCompleteLine() 함수의 리턴값을 더하여 score값을 추가한다.

후에 drawBoard(), drawStatus(), drawNextPiece() drawPiece() 함수를 모두 사용하여 화면에 현재 상태를 모두 출력시켜준다.

checkForKeyPress() 함수엔 checkForQuit() 함수를 사용하여 QUIT 이벤트가 있는 지 확인하고

showTextScreen 함수엔 makeTextObjs() 함수를 사용하여 text 개체를 만드는 데 사용한다.

isValidPosition() 함수엔 isOnBoard() 함수를 이용하여 보드에 있는지 확인하는 조건문을 작성하였고 (if not isOnBoard(): return False)

removeCompleteLines() 함수에는 isCompleteLine() 함수를 사용하여 라인이 지워졌는 지

확인하는 조건문을 작성하였다.

drawPiece() 함수에서는 drawBox() 함수를 이용하여 블록 조각을 만들어내었고

drawNextPiece() 함수에서는 drawPiece() 함수를 이용하여 다음 블록 조각을 만들어내었다.