

추가로 수정(색이 고유 색상 없이 무작위 색상을 가져서 수정)

수정전)

```
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': random.randint(0, len(COLORS)-1)}
    return newPiece

def drawBox(boxx, boxy, color, pixelx=None, pixely=None):
    # draw a single box (each tetromino piece has four boxes)
    # at xy coordinates on the board. Or, if pixelx & pixely
    # are specified, draw to the pixel coordinates stored in
    # pixelx & pixely (this is used for the "Next" piece).
    if color == BLANK:
        return
    if pixelx == None and pixely == None:
        pixelx, pixely = convertToPixelCoords(boxx, boxy)
    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))

def addToBoard(board, piece):
    # fill in the board based on piece's location, shape, and rotation
    for x in range(TEMPLATEWIDTH):
        for y in range(TEMPLATEHEIGHT):
            if PIECES[piece['shape']][piece['rotation']][y][x] != BLANK:
                board[x + piece['x']][y + piece['y']] = piece['color']

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

수정후

```
# 각 블록의 고유 색상 정의
SHAPE_COLORS = {
    'S': GREEN,
```

```

    'Z': RED,
    'J': BLUE,
    'L': ORANGE,
    'I': CYAN,
    'O': YELLOW,
    'T': MAGENTA
}

# 각 블록의 고유 밝은 색상 정의
SHAPE_LIGHTCOLORS = {
    'S': LIGHTGREEN,
    'Z': LIGHTRED,
    'J': LIGHTBLUE,
    'L': LIGHTORANGE,
    'I': LIGHTCYAN,
    'O': LIGHTYELLOW,
    'T': LIGHTMAGENTA
}

def getNewPiece():
    # 무작위로 새 조각을 반환하되, 고유 색상 부여
    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, # 보드 위에 시작 (즉, 0보다 작음)
                'color': shape}
    return newPiece

def drawBox(boxx, boxy, shape, pixelx=None, pixely=None):
    # 단일 박스를 그림 (각 테트로미노 조각은 네 개의 박스로 구성)
    # 보드의 xy 좌표에 그리거나, pixelx & pixely가 지정된 경우
    # 해당 픽셀 좌표에 그린다 (이것은 "다음" 조각을 위해 사용됨).
    if shape == BLANK:
        return
    color = SHAPE_COLORS[shape]
    lightColor = SHAPE_LIGHTCOLORS[shape]
    if pixelx == None and pixely == None:
        pixelx, pixely = convertToPixelCoords(boxx, boxy)
    pygame.draw.rect(DISPLAYSURF, color, (pixelx + 1, pixely + 1, BOXSIZE - 1,
    BOXSIZE - 1))
    pygame.draw.rect(DISPLAYSURF, lightColor, (pixelx + 1, pixely + 1, BOXSIZE - 4,
    BOXSIZE - 4))

def addToBoard(board, piece):
    # 조각의 위치, 모양 및 회전 정보를 기반으로 보드를 채운다
    for x in range(TEMPLATEWIDTH):
        for y in range(TEMPLATEHEIGHT):
            if PIECES[piece['shape']][piece['rotation']][y][x] != BLANK:
                board[x + piece['x']][y + piece['y']] = piece['shape']

def drawPiece(piece, pixelx=None, pixely=None):
    shapeToDraw = PIECES[piece['shape']][piece['rotation']]
    if pixelx == None and pixely == None:

```

```

# pixelx & pixely가 지정되지 않은 경우, 조각 데이터 구조에 저장된 위치를 사용
pixelx, pixely = convertToPixelCoords(piece['x'], piece['y'])

# 조각을 구성하는 각 박스를 그린다
for x in range(TEMPLATEWIDTH):
    for y in range(TEMPLATEHEIGHT):
        if shapeToDraw[y][x] != BLANK:
            drawBox(None, None, piece['shape'], pixelx + (x * BOXSIZE), pixely
+ (y * BOXSIZE))

```

함수 순서 및 조건

함수 호출 순서 및 조건

1. getNewPiece 함수

역할: 무작위로 새로운 테트로미노 조각을 생성합니다.

호출 조건:

게임 시작 시

현재 조각이 보드에 고정된 후 새로운 조각이 필요할 때

2. drawBox 함수

역할: 단일 박스를 화면에 그립니다. 각 테트로미노 조각은 네 개의 박스로 구성됩니다.

호출 조건:

drawPiece 함수 내에서 각 박스를 그릴 때

보드 상태를 화면에 갱신할 때

3. drawPiece 함수

역할: 현재 조각을 화면에 그립니다.

호출 조건:

현재 조각을 화면에 그릴 필요가 있을 때

게임 루프 내에서 주기적으로 호출

사용자 입력(이동, 회전 등)에 의해 조각의 위치가 변경될 때

4. addToBoard 함수

역할: 현재 조각을 보드에 고정시킵니다.

호출 조건:

현재 조각이 더 이상 움직일 수 없을 때(즉, 바닥에 닿거나 다른 조각과 겹쳤을 때)

전체 게임 흐름에서의 함수 호출 순서

게임 시작 시:

getNewPiece를 호출하여 첫 번째 조각과 다음 조각을 생성합니다.

게임 루프 내에서:

주기적으로 drawPiece를 호출하여 현재 조각을 화면에 그립니다.

사용자 입력에 따라 조각의 위치나 회전 상태가 변경될 때도 drawPiece를 호출합니다.

조각이 더 이상 아래로 이동할 수 없을 때:

addToBoard를 호출하여 현재 조각을 보드에 고정합니다.

새로운 조각을 생성하기 위해 getNewPiece를 호출합니다.