

오픈소스SW 과제중심수업 1주차 Pygame

2021012951 김선명

목차

1. 체크리스트 1~6번의 수행 방법 설명
2. 각 함수의 역할 설명
3. 함수의 호출 순서 및 호출 조건 설명

Github: <https://github.com/romantic02/osw-repository>

#1. 체크리스트 1~6번의 수행 방법

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

```
158 def main():
159     global FPCLOCK, DISPLAYSURF, BASICFONT, BIGFONT
160     pygame.init()
161     FPCLOCK = pygame.time.Clock()
162     DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
163     BASICFONT = pygame.font.Font('freesansbold.ttf', 18)
164     BIGFONT = pygame.font.Font('freesansbold.ttf', 100)
165     pygame.display.set_caption('Tetromino')
166
167     showTextScreen('Tetromino')
168     while True: # game loop
169         if random.randint(0, 1) == 0:
170             pygame.mixer.music.load('tetrisb.mid')
171         else:
172             pygame.mixer.music.load('tetrisc.mid')
173         pygame.mixer.music.play(-1, 0.0)
174         runGame()
175         pygame.mixer.music.stop()
176         showTextScreen('Game Over')
```

```
167 def main():
168     global FPCLOCK, DISPLAYSURF, BASICFONT, BIGFONT
169     pygame.init()
170     FPCLOCK = pygame.time.Clock()
171     DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
172     BASICFONT = pygame.font.Font('freesansbold.ttf', 18)
173     BIGFONT = pygame.font.Font('freesansbold.ttf', 100)
174     pygame.display.set_caption('2021012951_김선명')
175
176     showTextScreen('MY TETRIS')
177     while True: # game loop
178         k = random.randint(0, 2)
179         if k == 0:
180             pygame.mixer.music.load('Hover.mp3')
181         elif k == 1:
182             pygame.mixer.music.load('Our_Lives_Past.mp3')
183         else:
184             pygame.mixer.music.load('Platform_9.mp3')
185         pygame.mixer.music.play(-1, 0.0)
186         runGame()
187         pygame.mixer.music.stop()
188         showTextScreen('Game Over')
```

pygame 원본 코드(위)와 수정한 코드(아래).

mp3파일인 Hover.mp3, Our_Lives_Past.mp3, Platform_9.mp3를 작업중인 디렉토리에 추가하고 아래 코드의 178~184줄을 수정함으로써 게임 실행시 3개의 음악 중 1개가 재생된다.

2) 상태창 이름을 학번_이름으로 수정

1)의 아래 코드에서 174번 줄의 코드를 수정함으로써 상태창 이름을 2021012951_김선명으로 수정한다.

3) 게임시작화면의 문구를 MY TETRIS로 변경

1)의 아래 코드에서 176번 줄의 코드를 수정함으로써 게임시작화면의 문구를 MY TETRIS로 변경하였다.

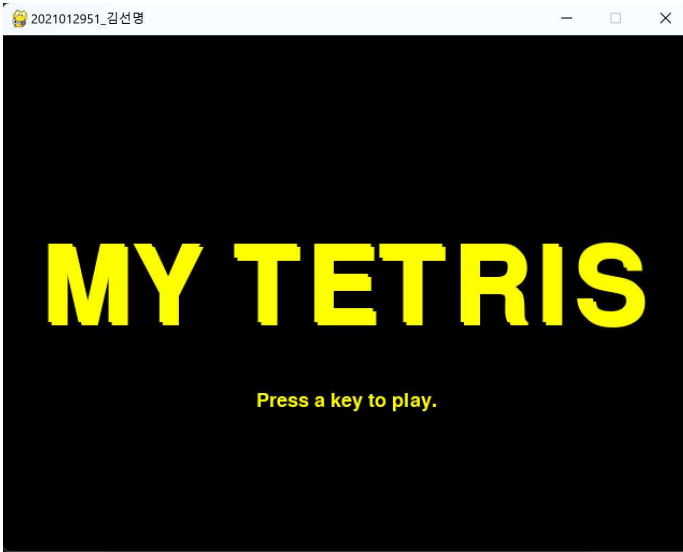
4) 게임시작화면의 문구 및 배경색을 노란색으로 변경

```
302     def makeTextObjs(text, font, color):
303         surf = font.render(text, True, color)
304         return surf, surf.get_rect()
```

```
317     def makeTextObjs(text, font, color):
318         surf = font.render(text, True, "Yellow")
319         return surf, surf.get_rect()
```

pygame 원본 코드(위)와 수정한 코드(아래).

아래의 코드에서 318번 줄의 코드를 수정함으로써 게임시작화면의 문구를 노란색으로 변경했다.



2), 3), 4)의 변경사항이 잘 반영된 것을 확인할 수 있다.

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

```
468 def drawStatus(score, level):
469     # draw the score text
470     scoreSurf = BASICFONT.render('Score: %s' % score, True, TEXTCOLOR)
471     scoreRect = scoreSurf.get_rect()
472     scoreRect.topleft = (WINDOWWIDTH - 150, 20)
473     DISPLAYSURF.blit(scoreSurf, scoreRect)
474
475     # draw the level text
476     levelSurf = BASICFONT.render('Level: %s' % level, True, TEXTCOLOR)
477     levelRect = levelSurf.get_rect()
478     levelRect.topleft = (WINDOWWIDTH - 150, 50)
479     DISPLAYSURF.blit(levelSurf, levelRect)
```

```
483 def drawStatus(score, level, time):
484     # draw the score text
485     scoreSurf = BASICFONT.render('Score: %s' % score, True, TEXTCOLOR)
486     scoreRect = scoreSurf.get_rect()
487     scoreRect.topleft = (WINDOWWIDTH - 150, 20)
488     DISPLAYSURF.blit(scoreSurf, scoreRect)
489
490     # draw the level text
491     levelSurf = BASICFONT.render('Level: %s' % level, True, TEXTCOLOR)
492     levelRect = levelSurf.get_rect()
493     levelRect.topleft = (WINDOWWIDTH - 150, 50)
494     DISPLAYSURF.blit(levelSurf, levelRect)
495
496
497     #draw the timer text
498     timerSurf = BASICFONT.render('Play Time: %s' % time, True, TEXTCOLOR)
499     timerRect = timerSurf.get_rect()
500     timerRect.topleft = (WINDOWWIDTH - 600, 20)
501     DISPLAYSURF.blit(timerSurf, timerRect)
```

pygame 원본 코드(위1)와 수정한 코드(아래1).

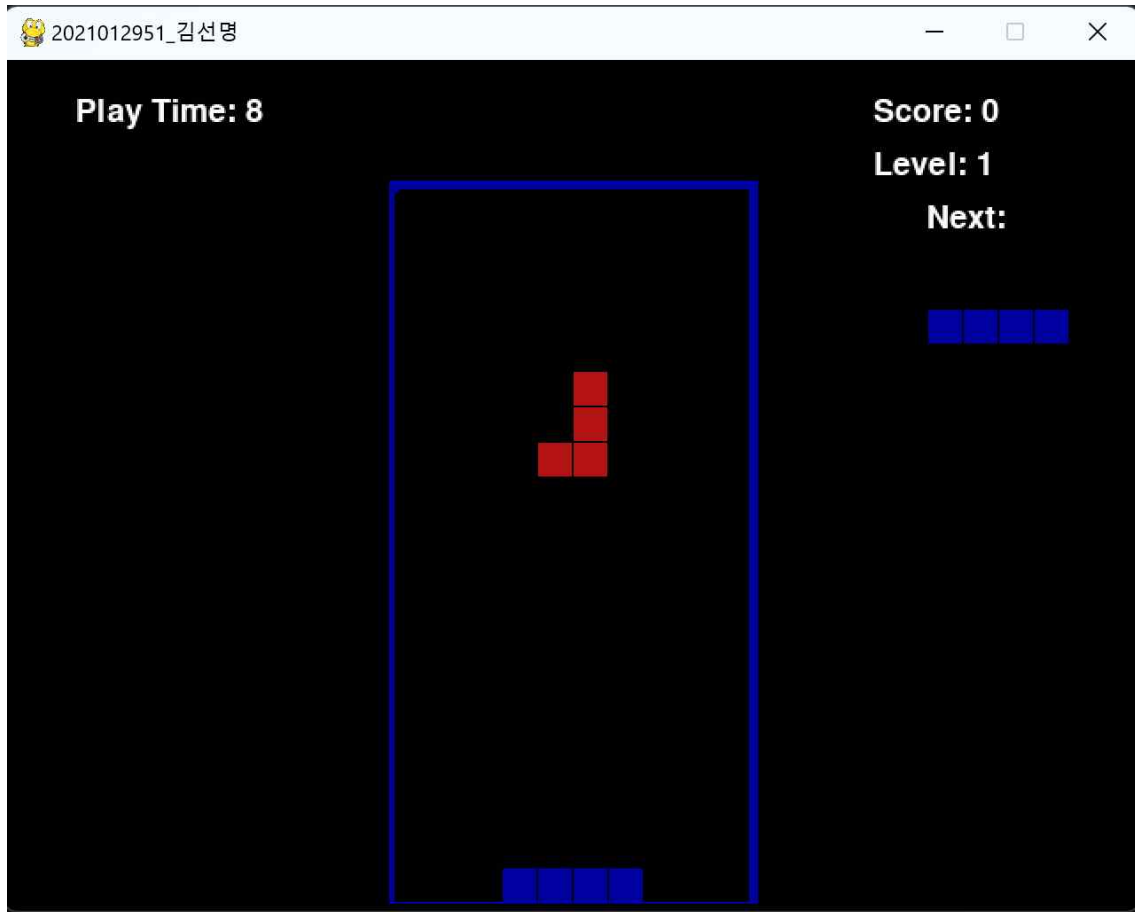
drawStatus 함수에 time 인자를 추가로 전달하고 (아래1) 코드의 498~501줄에 경과시간을 게임화면 좌측상단에 표시하는 코드를 추가했다.

```
osw > timer.py > runGame
179 def runGame():
180     # let the piece fall if it is time to fall
277     if time.time() - lastFallTime > fallFreq:
278         # see if the piece has landed
279         if not isValidPosition(board, fallingPiece, adjY=1):
280             # falling piece has landed, set it on the board
281             addToBoard(board, fallingPiece)
282             score += removeCompletelines(board)
283             level, fallFreq = calculateLevelAndFallFreq(score)
284             fallingPiece = None
285         else:
286             # piece did not land, just move the piece down
287             fallingPiece['y'] += 1
288             lastFallTime = time.time()
289
290     # drawing everything on the screen
291     DISPLAYSURF.fill(BGCOLOR)
292     drawBoard(board)
293     drawStatus(score, level)
294     drawNextPiece(nextPiece)
295     if fallingPiece != None:
296         drawPiece(fallingPiece)
297
298     pygame.display.update()
299     FPSCLOCK.tick(FPS)
300
301
302 def makeTextObjs(text, font, color):
303     surf = font.render(text, True, color)
304     return surf, surf.get_rect()
```

```
osw > 2021012951_김선명.py > runGame
191 def runGame():
289     # let the piece fall if it is time to fall
290     if time.time() - lastFallTime > fallFreq:
291         # see if the piece has landed
292         if not isValidPosition(board, fallingPiece, adjY=1):
293             # falling piece has landed, set it on the board
294             addToBoard(board, fallingPiece)
295             score += removeCompletelines(board)
296             level, fallFreq = calculateLevelAndFallFreq(score)
297             fallingPiece = None
298         else:
299             # piece did not land, just move the piece down
300             fallingPiece['y'] += 1
301             lastFallTime = time.time()
302
303     # drawing everything on the screen
304     DISPLAYSURF.fill(BGCOLOR)
305     elapsed_time = (pygame.time.get_ticks() // 1000) - start_time
306     drawBoard(board)
307     drawStatus(score, level, elapsed_time)
308     drawNextPiece(nextPiece)
309
310     if fallingPiece != None:
311         drawPiece(fallingPiece)
312
313     pygame.display.update()
314     FPSCLOCK.tick(FPS)
315
316
317 def makeTextObjs(text, font, color):
```

pygame 원본 코드(위2)와 수정한 코드(아래2).

게임 실행시 화면 좌측상단에 경과시간을 표시하기 위해 runGame 함수에서 수정된 drawStatus 함수를 호출하고 elapsed_time 인자를 추가로 전달한다. start_time은 runGame 함수에 start_time = pygame.time.get_ticks() // 1000으로 초기화되어있다.



게임화면 좌측 상단에 경과시간이 표시되는 걸 확인할 수 있다.

6) 7개의 블록이 각각 고유의 색을 갖도록 코드를 수정하거나 추가

```
osw> timer.py > ...  
140 |         |         |         |         |         |         |         |         |         |         |  
141 |         |         |         |         |         |         |         |         |         |         |  
142 |         |         |         |         |         |         |         |         |         |         |  
143 |         |         |         |         |         |         |         |         |         |         |  
144 |         |         |         |         |         |         |         |         |         |         |  
145 |         |         |         |         |         |         |         |         |         |         |  
146 |         |         |         |         |         |         |         |         |         |         |  
147 |         |         |         |         |         |         |         |         |         |         |  
148 |         |         |         |         |         |         |         |         |         |         |  
149 |     PIECES = { 'S': S_SHAPE_TEMPLATE,  
150 |               'Z': Z_SHAPE_TEMPLATE,  
151 |               'J': J_SHAPE_TEMPLATE,  
152 |               'L': L_SHAPE_TEMPLATE,  
153 |               'I': I_SHAPE_TEMPLATE,  
154 |               'O': O_SHAPE_TEMPLATE,  
155 |               'T': T_SHAPE_TEMPLATE}  
156 |  
157 |  
158 | def main():  
159 |     global FPSCLOCK, DISPLAYSURF, BASICFONT, BIGFONT  
160 |     pygame.init()  
161 |     FPSCLOCK = pygame.time.Clock()  
162 |     DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))  
163 |     BASICFONT = pygame.font.Font('freesansbold.ttf', 18)  
164 |     BIGFONT = pygame.font.Font('freesansbold.ttf', 100)  
165 |     pygame.display.set_caption('Tetromino')  
166 |  
167 |     showTextScreen('Tetromino')  
168 |     while True: # game loop  
169 |         if random.randint(0, 1) == 0:
```

```
osw > 2021012951_김선영.py > ...
140         '...00.',
141         '...0..',
142         '.....'],
143     ['.....',
144         '...0..',
145         '..00..',
146         '..0...',
147         '.....']]
148
149 PIECES = {'S': S_SHAPE_TEMPLATE,
150           'Z': Z_SHAPE_TEMPLATE,
151           'J': J_SHAPE_TEMPLATE,
152           'L': L_SHAPE_TEMPLATE,
153           'I': I_SHAPE_TEMPLATE,
154           'O': O_SHAPE_TEMPLATE,
155           'T': T_SHAPE_TEMPLATE}
156
157 PIECES_COLORS = {'S': LIGHTBLUE,
158                  'Z': LIGHTGREEN,
159                  'J': LIGHTRED,
160                  'L': LIGHTYELLOW,
161                  'I': BLUE,
162                  'O': GREEN,
163                  'T': RED}
164
165 }
166
167 def main():
168     global FPSLOCK, DISPLAYSURF, BASICFONT, BIGFONT
169     pygame.init()
```

pygame 원본 코드(위1)와 수정한 코드(아래1).


```

osw > timer.py > ...
346
347 def checkForQuit():
348     for event in pygame.event.get(QUIT): # get all the QUIT events
349         terminate() # terminate if any QUIT events are present
350     for event in pygame.event.get(KEYUP): # get all the KEYUP events
351         if event.key == K_ESCAPE:
352             terminate() # terminate if the KEYUP event was for the Esc key
353         pygame.event.post(event) # put the other KEYUP event objects back
354
355
356 def calculateLevelAndFallFreq(score):
357     # Based on the score, return the level the player is on and
358     # how many seconds pass until a falling piece falls one space.
359     level = int(score / 10) + 1
360     fallFreq = 0.27 - (level * 0.02)
361     return level, fallFreq
362
363 def getNewPiece():
364     # return a random new piece in a random rotation and color
365     shape = random.choice(list(PIECES.keys()))
366     newPiece = {'shape': shape,
367                 'rotation': random.randint(0, len(PIECES[shape]) - 1),
368                 'x': int(BOARDWIDTH / 2) - int(TEMPLATEWIDTH / 2),
369                 'y': -2, # start it above the board (i.e. less than 0)
370                 'color': random.randint(0, len(COLORS)-1)}
371     return newPiece
372
373
374 def addToBoard(board, piece):
375     # fill in the board based on piece's location, shape, and rotation

```

```

osw > 2021012951_김선명.py > ...
360
361
362 def checkForQuit():
363     for event in pygame.event.get(QUIT): # get all the QUIT events
364         terminate() # terminate if any QUIT events are present
365     for event in pygame.event.get(KEYUP): # get all the KEYUP events
366         if event.key == K_ESCAPE:
367             terminate() # terminate if the KEYUP event was for the Esc key
368         pygame.event.post(event) # put the other KEYUP event objects back
369
370
371 def calculateLevelAndFallFreq(score):
372     # Based on the score, return the level the player is on and
373     # how many seconds pass until a falling piece falls one space.
374     level = int(score / 10) + 1
375     fallFreq = 0.27 - (level * 0.02)
376     return level, fallFreq
377
378 def getNewPiece():
379     # return a random new piece in a random rotation and color
380     shape = random.choice(list(PIECES.keys()))
381     newPiece = {'shape': shape,
382                 'rotation': random.randint(0, len(PIECES[shape]) - 1),
383                 'x': int(BOARDWIDTH / 2) - int(TEMPLATEWIDTH / 2),
384                 'y': -2, # start it above the board (i.e. less than 0)
385                 'color': PIECES_COLORS[shape]}
386     return newPiece
387
388
389 def addToBoard(board, piece):

```

pygame 원본 코드(위2)와 수정한 코드(아래2).


```

osw > timer.py > ...
437 def convertToPixelCoords(boxx, boxy):
438     return (XMARGIN + (boxx * BOXSIZE)), (TOPMARGIN + (boxy * BOXSIZE))
441
442
443 def drawBox(boxx, boxy, color, pixelx=None, pixely=None):
444     # draw a single box (each tetromino piece has four boxes)
445     # at xy coordinates on the board. Or, if pixelx & pixely
446     # are specified, draw to the pixel coordinates stored in
447     # pixelx & pixely (this is used for the "Next" piece).
448     if color == BLANK:
449         return
450     if pixelx == None and pixely == None:
451         pixelx, pixely = convertToPixelCoords(boxx, boxy)
452     pygame.draw.rect(DISPLAYSURF, COLORS[color], (pixelx + 1, pixely + 1, BOXSIZE - 1, BOXSIZE - 1))
453     pygame.draw.rect(DISPLAYSURF, LIGHTCOLORS[color], (pixelx + 1, pixely + 1, BOXSIZE - 4, BOXSIZE - 4))
454
455
456 def drawBoard(board):
457     # draw the border around the board
458     pygame.draw.rect(DISPLAYSURF, BORDERCOLOR, (XMARGIN - 3, TOPMARGIN - 7, (BOARDWIDTH * BOXSIZE) + 8, (BOARDHEIGHT * BOXSIZE) + 8), 5)
459
460     # fill the background of the board
461     pygame.draw.rect(DISPLAYSURF, BGCOLOR, (XMARGIN, TOPMARGIN, BOXSIZE * BOARDWIDTH, BOXSIZE * BOARDHEIGHT))
462     # draw the individual boxes on the board
463     for x in range(BOARDWIDTH):
464         for y in range(BOARDHEIGHT):
465             drawBox(x, y, board[x][y])
466
467
468 def drawStatus(score, level):

```

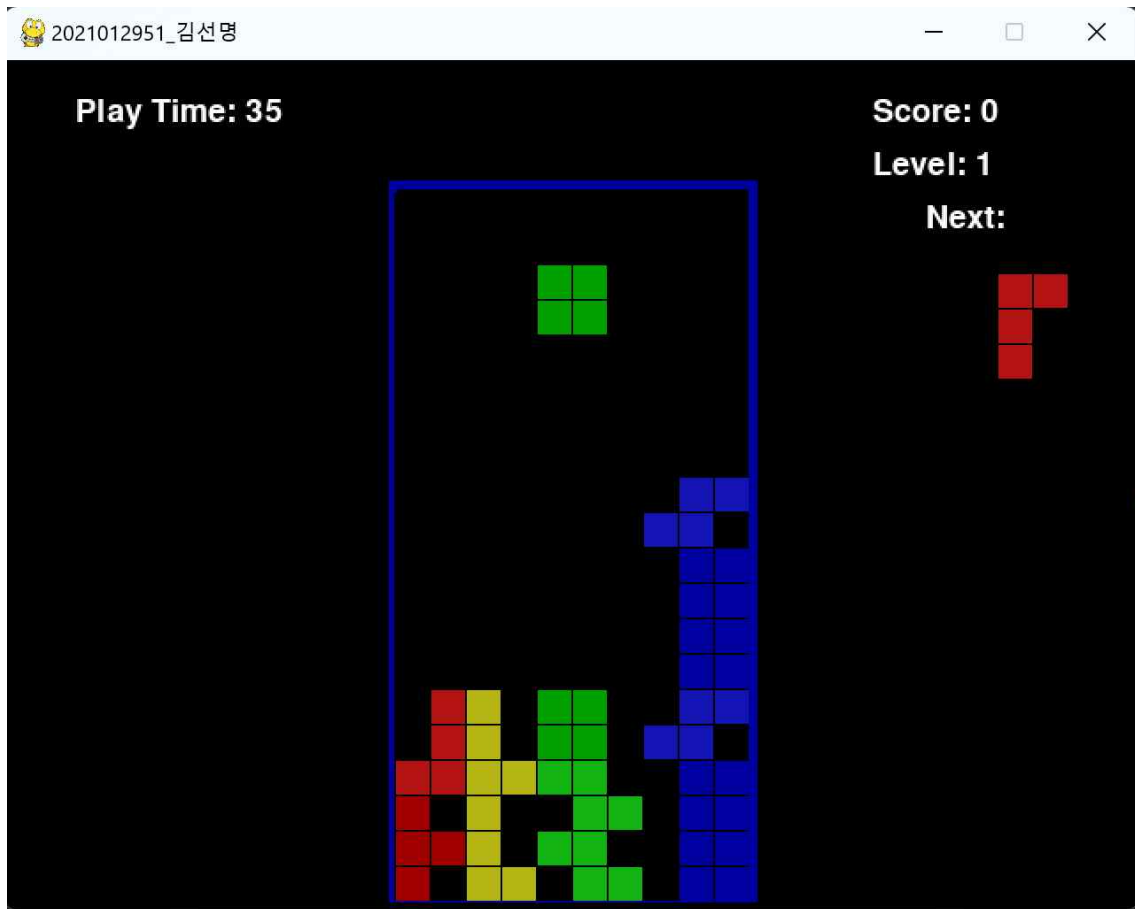
```

osw > 2021012851_김선명.py > ...
452 def convertToPixelCoords(boxx, boxy):
453     return (XMARGIN + (boxx * BOXSIZE)), (TOPMARGIN + (boxy * BOXSIZE))
456
457
458 def drawBox(boxx, boxy, color, pixelx=None, pixely=None):
459     # draw a single box (each tetromino piece has four boxes)
460     # at xy coordinates on the board. Or, if pixelx & pixely
461     # are specified, draw to the pixel coordinates stored in
462     # pixelx & pixely (this is used for the "Next" piece).
463     if color == BLANK:
464         return
465     if pixelx == None and pixely == None:
466         pixelx, pixely = convertToPixelCoords(boxx, boxy)
467     pygame.draw.rect(DISPLAYSURF, color, (pixelx + 1, pixely + 1, BOXSIZE - 1, BOXSIZE - 1))
468     pygame.draw.rect(DISPLAYSURF, color, (pixelx + 1, pixely + 1, BOXSIZE - 4, BOXSIZE - 4))
469
470
471 def drawBoard(board):
472     # draw the border around the board
473     pygame.draw.rect(DISPLAYSURF, BORDERCOLOR, (XMARGIN - 3, TOPMARGIN - 7, (BOARDWIDTH * BOXSIZE) + 8, (BOARDHEIGHT * BOXSIZE) + 8), 5)
474
475     # fill the background of the board
476     pygame.draw.rect(DISPLAYSURF, BGCOLOR, (XMARGIN, TOPMARGIN, BOXSIZE * BOARDWIDTH, BOXSIZE * BOARDHEIGHT))
477     # draw the individual boxes on the board
478     for x in range(BOARDWIDTH):
479         for y in range(BOARDHEIGHT):
480             drawBox(x, y, board[x][y])
481
482
483 def drawStatus(score, level, time):

```

pygame 원본 코드(위3)와 수정한 코드(아래3).

(아래1)의 157~163줄에 각 도형의 고유한 색을 부여하는 코드를 추가하였다.
(아래2)의 getNewPiece 함수에서 newPiece가 색을 무작위가 아닌 도형에 부여된 색을 갖도록 수정하였다. (아래3)의 467, 468줄을 오류가 나지 않도록 수정하였다.



서로 다른 모양의 블록이 서로 다른 색을 갖는 것을 확인할 수 있다. 원색과 LIGHT색이 비슷해 보이지만 다르다. 확실하게 색을 구별하기 위해서 색의 RGB값을 변경할 수도 있다.

이렇게 1~6번의 과제를 모두 성공적으로 수행했다.

실습 강의 PPT 자료의 과제도 수행해보겠다.

시작화면의 상태창에 학번_이름을 출력하는 것은 이미 완료했다.

```

osw > timer.py > ...
312 def checkForKeyPress():
321     return None
322
323
324 def showTextScreen(text):
325     # This function displays large text in the
326     # center of the screen until a key is pressed.
327     # Draw the text drop shadow
328     titleSurf, titleRect = makeTextObjs(text, BIGFONT, TEXTSHADOWCOLOR)
329     titleRect.center = (int(WINDOWWIDTH / 2), int(WINDOWHEIGHT / 2))
330     DISPLAYSURF.blit(titleSurf, titleRect)
331
332     # Draw the text
333     titleSurf, titleRect = makeTextObjs(text, BIGFONT, TEXTCOLOR)
334     titleRect.center = (int(WINDOWWIDTH / 2) - 3, int(WINDOWHEIGHT / 2) - 3)
335     DISPLAYSURF.blit(titleSurf, titleRect)
336
337     # Draw the additional "Press a key to play." text.
338     pressKeySurf, pressKeyRect = makeTextObjs('Press a key to play.', BASICFONT, TEXTCOLOR)
339     pressKeyRect.center = (int(WINDOWWIDTH / 2), int(WINDOWHEIGHT / 2) + 100)
340     DISPLAYSURF.blit(pressKeySurf, pressKeyRect)
341
342     while checkForKeyPress() == None:
343         pygame.display.update()
344         FPSCLOCK.tick()
345
346
347 def checkForQuit():
348     for event in pygame.event.get(QUIT): # get all the QUIT events
349         terminate() # terminate if any QUIT events are present

```

```

osw > 2021012951 김선명.py > ...
327 def checkForKeyPress():
336     return None
337
338
339 def showTextScreen(text):
340     # This function displays large text in the
341     # center of the screen until a key is pressed.
342     # Draw the text drop shadow
343     titleSurf, titleRect = makeTextObjs(text, BIGFONT, TEXTSHADOWCOLOR)
344     titleRect.center = (int(WINDOWWIDTH / 2), int(WINDOWHEIGHT / 2))
345     DISPLAYSURF.blit(titleSurf, titleRect)
346
347     # Draw the text
348     titleSurf, titleRect = makeTextObjs(text, BIGFONT, TEXTCOLOR)
349     titleRect.center = (int(WINDOWWIDTH / 2) - 3, int(WINDOWHEIGHT / 2) - 3)
350     DISPLAYSURF.blit(titleSurf, titleRect)
351
352     # Draw the additional "Press a key to play." text.
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)
356
357     while checkForKeyPress() == None:
358         pygame.display.update()
359         FPSCLOCK.tick()
360
361
362 def checkForQuit():
363     for event in pygame.event.get(QUIT): # get all the QUIT events
364         terminate() # terminate if any QUIT events are present

```

pygame 원본 코드(위)와 수정한 코드(아래).

(아래)의 353번 줄을 수정하였다.

```

osw > timer.py > runGame
179 def runGame():
192     nextPiece = getNewPiece()
193
194     while True: # game loop
195         if fallingPiece == None:
196             # No falling piece in play, so start a new piece at the top
197             fallingPiece = nextPiece
198             nextPiece = getNewPiece()
199             lastFallTime = time.time() # reset lastFallTime
200
201             if not isValidPosition(board, fallingPiece):
202                 return # can't fit a new piece on the board, so game over
203
204         checkForQuit()
205         for event in pygame.event.get(): # event handling loop
206             if event.type == KEYUP:
207                 if (event.key == K_p):
208                     # Pausing the game
209                     DISPLAYSURF.fill(BG_COLOR)
210                     pygame.mixer.music.stop()
211                     showTextScreen('Paused') # pause until a key press
212                     pygame.mixer.music.play(-1, 0.0)
213                     lastFallTime = time.time()
214                     lastMoveDownTime = time.time()
215                     lastMoveSidewaysTime = time.time()
216                 elif (event.key == K_LEFT or event.key == K_a):
217                     movingLeft = False
218                 elif (event.key == K_RIGHT or event.key == K_d):
219                     movingRight = False
220                 elif (event.key == K_DOWN or event.key == K_s):

```

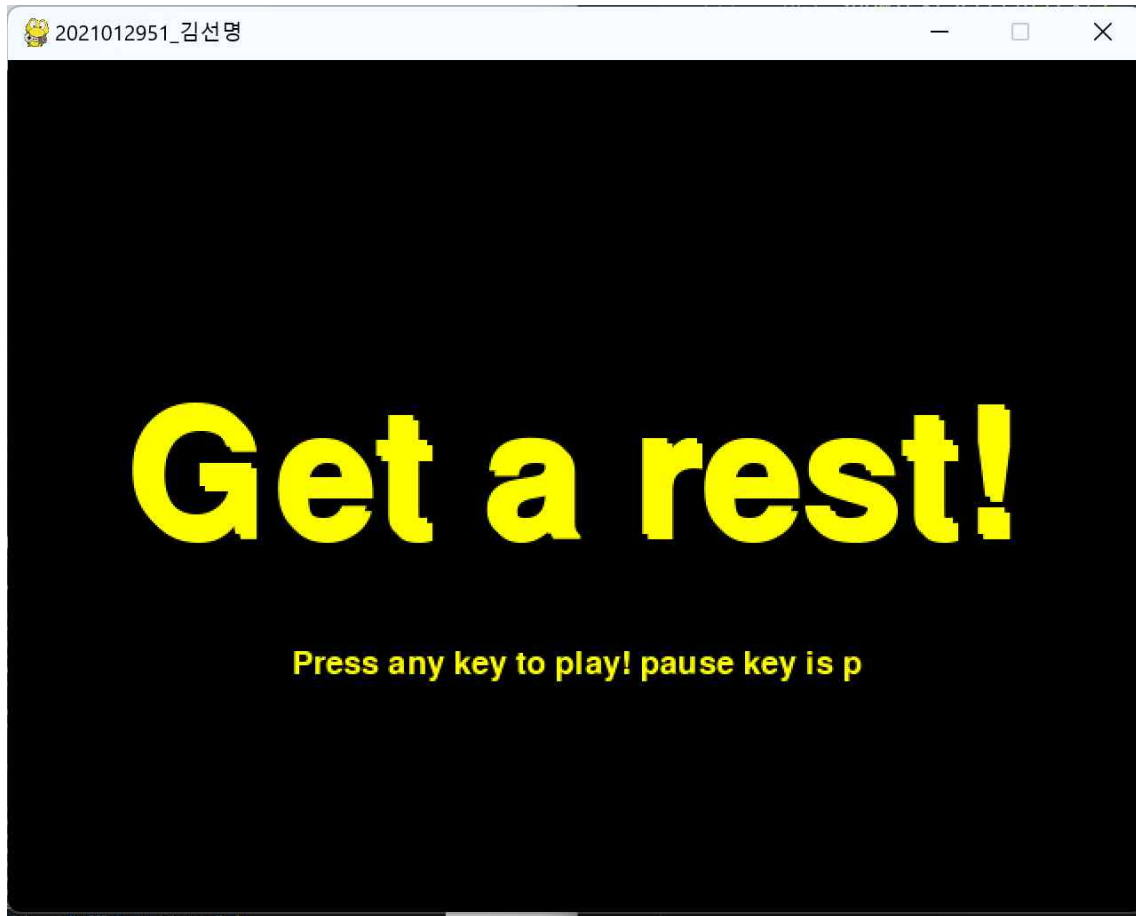
```

osw > 2021012951_김선명.py > runGame
191 def runGame():
205     nextPiece = getNewPiece()
206
207     while True: # game loop
208         if fallingPiece == None:
209             # No falling piece in play, so start a new piece at the top
210             fallingPiece = nextPiece
211             nextPiece = getNewPiece()
212             lastFallTime = time.time() # reset lastFallTime
213
214             if not isValidPosition(board, fallingPiece):
215                 return # can't fit a new piece on the board, so game over
216
217         checkForQuit()
218         for event in pygame.event.get(): # event handling loop
219             if event.type == KEYUP:
220                 if (event.key == K_p):
221                     # Pausing the game
222                     DISPLAYSURF.fill(BG_COLOR)
223                     pygame.mixer.music.stop()
224                     showTextScreen('Get a rest!') # pause until a key press
225                     pygame.mixer.music.play(-1, 0.0)
226                     lastFallTime = time.time()
227                     lastMoveDownTime = time.time()
228                     lastMoveSidewaysTime = time.time()
229                 elif (event.key == K_LEFT or event.key == K_a):
230                     movingLeft = False
231                 elif (event.key == K_RIGHT or event.key == K_d):
232                     movingRight = False
233                 elif (event.key == K_DOWN or event.key == K_s):

```

pygame 원본 코드(위)와 수정한 코드(아래).

(아래)의 224줄 코드를 수정하여 퍼즈를 걸었을 때 출력되는 문구를 수정하였다.



성공적으로 변경된 것을 확인할 수 있다.

```

osw > timer.py > ...
156
157
158 def main():
159     global FPSCLOCK, DISPLAYSURF, BASICFONT, BIGFONT
160     pygame.init()
161     FPSCLOCK = pygame.time.Clock()
162     DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
163     BASICFONT = pygame.font.Font('freesansbold.ttf', 18)
164     BIGFONT = pygame.font.Font('freesansbold.ttf', 100)
165     pygame.display.set_caption('Tetromino')
166
167     showTextScreen('Tetromino')
168     while True: # game loop
169         if random.randint(0, 1) == 0:
170             pygame.mixer.music.load('tetrisb.mid')
171         else:
172             pygame.mixer.music.load('tetrisc.mid')
173         pygame.mixer.music.play(-1, 0.0)
174         runGame()
175         pygame.mixer.music.stop()
176         showTextScreen('Game Over')
177
178
179 def runGame():
180     # setup variables for the start of the game
181     board = getBlankBoard()
182     lastMoveDownTime = time.time()
183     lastMoveSidewaysTime = time.time()
184     lastFallTime = time.time()
185     movingDown = False # note: there is no movingUp variable

```

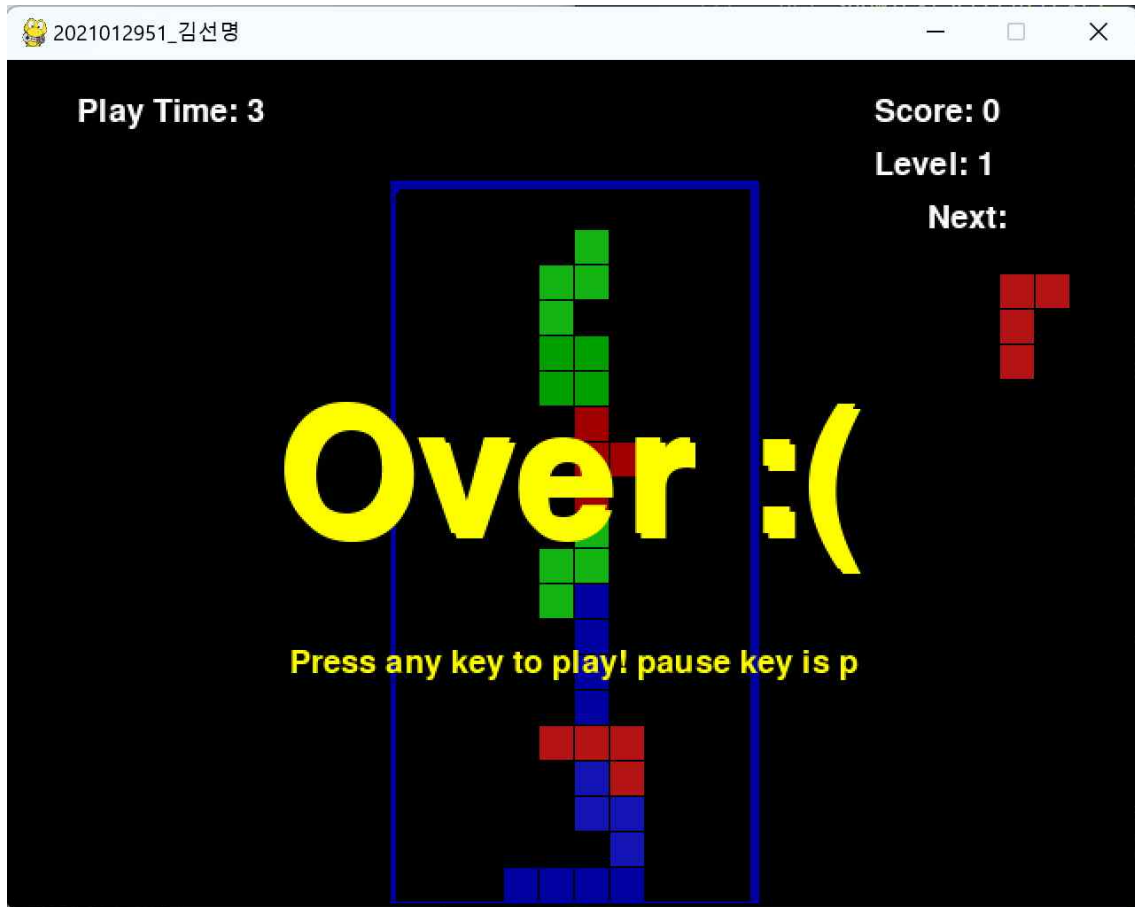
```

osw > 2021012951_김선명.py > ...
165
166
167 def main():
168     global FPSCLOCK, DISPLAYSURF, BASICFONT, BIGFONT
169     pygame.init()
170     FPSCLOCK = pygame.time.Clock()
171     DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
172     BASICFONT = pygame.font.Font('freesansbold.ttf', 18)
173     BIGFONT = pygame.font.Font('freesansbold.ttf', 100)
174     pygame.display.set_caption('2021012951_김선명')
175
176     showTextScreen('MY TETRIS')
177     while True: # game loop
178         k = random.randint(0, 2)
179         if k == 0:
180             pygame.mixer.music.load('Hover.mp3')
181         elif k == 1:
182             pygame.mixer.music.load('Our_Lives_Past.mp3')
183         else:
184             pygame.mixer.music.load('Platform_9.mp3')
185         pygame.mixer.music.play(-1, 0.0)
186         runGame()
187         pygame.mixer.music.stop()
188         showTextScreen('Over :(')
189
190
191 def runGame():
192     # setup variables for the start of the game
193     board = getBlankBoard()
194     lastMoveDownTime = time.time()

```

pygame 원본 코드(위)와 수정한 코드(아래).

(아래)의 188번 줄을 수정하여 게임이 종료되었을 때 출력되는 문구를 수정하였다.



게임 종료시 출력되는 문구가 변경되었다.

이렇게 실습 강의 PPT 자료의 과제도 모두 수행하였다.

#2. 각 함수들의 역할에 대한 설명

1. main()

게임의 주요 요소들을 설정하고 게임이 종료될때까지 메인루프를 실행시킨다. runGame()을 호출해 실제 게임을 실행시킨다.

2. runGame()

실제 게임을 실행하는 함수이다. 게임의 요소들을 설정하고 입력을 관리한다. 블록을 생성시키고 떨어뜨린다. 다음 블록의 자리가 없을 경우 게임을 종료시킨다. p가 입력될 경우 퍼즈를 건다. 방향키나 스페이스바가 입력될 경우 블록을 이동시키거나 블록을 회전시킨다. 완성된 줄을 제거하고 점수와 레벨에 반영시킨다. 화면에 보드, 점수, 단계, 그리고 경과시간을 표시한다.

3. makeTextObjs(text, font, color)

텍스트와 폰트, 색상을 설정한다.

4. terminate()

게임을 종료한다.

5. checkForKeyPress()

키 입력을 확인한다.

6. showTextScreen(text)

사용자가 키를 입력하기 전까지 화면 중앙에 텍스트를 표시한다.

7. checkForQuit()

종료조건이 만족되면 종료시킨다.

8. calculateLevelAndFallFreq(score)

점수를 기반으로 단계를 계산하고 블록이 떨어지는 속도를 조정한다.

9. getNewPiece()

무작위 모양의 블록을 생성하고 모양의 고유한 색을 부여한다.

10. addToBoard(board, piece)

블록의 위치, 모양에 따라 게임보드를 최신화한다.

11. getBlankBoard()

빈 보드를 생성한다.

12. isOnBoard(x, y)

주어진 좌표가 보드를 벗어나는지 확인한다.

13. isValidPosition(board, piece, adjX=0, adjY=0)

블록이 보드의 위치에 존재할 수 있는지 확인한다.

14. isCompleteLine(board, y)

줄이 완성되었는지 확인한다.

15. removeCompleteLines(board)

줄이 완성되었으면 삭제하고 위의 블록들을 내려서 칸을 채운다.

16. convertToPixelCoords(boxx, boxy)

좌표를 보드상의 좌표로 변환한다.

17. drawBox(boxx, boxy, color, pixelx=None, pixely=None)

블록의 한 칸을 이루는 정사각형을 그린다.

18. drawBoard(board)

보드의 테두리를 그린다. 보드의 각 칸에 정사각형을 그린다.

19. drawStatus(score, level, time)

게임의 상태를 표시한다. 화면의 좌측상단에 경과시간, 우측상단에 점수와 단계
계를 표시한다.

20. drawPiece(piece, pixelx=None, pixely=None)

drawBox()를 호출해 블록을 그린다.

21. drawNextPiece(piece)

화면 우측상단에 다음 블록을 표시한다.

#3. 함수의 호출 순서 및 호출 조건에 대한 설명

프로그램을 실행시키면

```
if __name__ == '__main__':  
    main()
```

을 통해 main() 함수를 호출한다. main() 함수는 게임을 초기화하고 showTextScreen('MY TETRIS')를 불러 게임 화면에 MY TETRIS라는 문구를 생성하고 입력이 있을때까지 기다린다. 입력이 확인되면 무작위로 세 개의 음악파일중 하나를 실행하고 runGame() 함수 호출을 반복한다. runGame() 함수는 실제 플레이되는 게임의 요소들을 설정하고 다른 함수들로 입력을 처리한다. getNewPiece() 함수로 새로운 블록을 생성하고 반복문으로 계속 떨어트린다. 게임실행중 checkForQuit() 함수로 게임을 종료하는 이벤트가 발생하는지 확인한다. p키가 입력될 경우 showTextScreen('Get a rest!')를 호출해 다음 입력이 있을때까지 퍼즈를 건다. 방향키나 스페이스바가 입력될 경우 블록의 위치를 옮기거나 회전시킨다. drawBoard()로 게임 보드를 생성, drawStatus()로 게임의 진행상황을 표시하고 drawNextPiece()로 다음 블록을 표시한다. isValidPositoin()으로 게임 종료이 종료되면 pygame.mixer.music.stop()을 호출해 음악을 멈추고 showTextScreen('Over :(')로 화면에 게임이 종료되었다는 문구를 표시한다. 입력이 확인되면 반복문을 다시 실행한다.