오픈소스SW 과제중심수업 보고서

ICT융합학부 컬처테크 전공

2019020855 김 빈

GitHub repository 주소 : https://github.com/samsam1568/osw

# 각 함수들의 역할

1~16줄: 게임에서 사용되는 몇가지 상수 변수를 정의하고 게임의 토대가 되는 셀의 너비와 높이를

CELLSIZE에 저장한다. 그리고 13,14행의 assert문으로 셀이 창에 맞는 지 확인한다.

17~41줄: 밑에서 쓸 색깔들을 지정해서 선언하고 방향키와 worm머리 등을 선언한다.

Def main() : 코드의 주요부분을 runGame()함수에 맡기고 main함수는 이를 호출하는 방식으로

사용한다. showStartScreen()를 호출하여 시작화면 'OSW Game'을 표시하고 runGame을 호출하여 'OSW Game'을 실행한다.

OSW Game은 플레이어의 웜이 자체 충돌하거나 벽에 충돌하면 게임이 끝날 떄 반환된다.

그러면 그 떄 showGameOverScreen()을 호출하여 다른 화면을 보여주고 함수 호출이

반환되면 루프의 시작으로 다시 돌아가서 rungame()을 다시 호출한다.

while루프는 프로그램이 종료될 때 까지 계속 루프한다.

47 – 폰트는 파일이 저장되어 있는 위치를 직접 지정해 불러왔다. (체크리스트 7)

52 – Wormy를 학번과 이름으로 바꿔 저장했다(체크리스트 1)

55,56 – runGame()를 호출하고 GameOver시 showGameOverScreen()을 수행한다.

Def runGame() : Wormy게임을 가장 직접적으로 구현하는 함수

게임을 시작할 때 웜이 임의의 위치에서 시작되어 이 위치를 startx starty에 좌표를

저장한다. 웜의 몸체는 하나의 부분 당 하나의 사전 값이 할당되어 있다. 사전에는

해당 본문의 XY좌표에 대한 키 ‘x’ ‘y’가 있는데 머리는 startx starty에 있어야 한다.

웜의 머리는 wormCoords[0]이어야하고 이를 쉽게 식별하기 위해

wormCoord[HEAD]로 대신 사용할 수 있게 만들었다.

84라인은 메인 게임 루프의 시작이고 라인 85은 이벤트 핸들링 루프의 시작이다.

QUIT이벤트의 경우 terminate()를 호출한다. KEYDOWN이벤트인 경우 누른 키가

화살표 키인지 WASD 키인지 확인한다. 항상 방향 변수의 현재 값에 대해

검사함으로써 웜을 충돌시키는 경우 키 입력을 무시하게 한다.

머리가 그리드 가장자리에서 이동하거나 머리가 이미 다른 몸체 세그멘트를 차지하고

있는 셀로 이동할 때 웜이 충돌한다.

헤드의 X 좌표 가 -1 인지 또는 CELLWIDTH와 같은지 확인하여 헤드가 그리드의

가장자리에서 이동했는지 확인 할 수 있다.

머리의 Y 좌표가 -1이거나 CELLHEIGHT인 경우 헤드가 격자에서 벗어난 것을 알 수

있다. 게임을 끝내기 위해서 할 일은 runGame에서 돌아오는 것이다. runGame()이

main 함수 호출로 돌아오면 showGameOverScreen()을 호출해 ‘Game Over’가 화면에

출력된다. 112라인은 머리 뒤 wormCoords의 모든 신체 세그멘트를 통해 반복한다.

몸체의 세그멘트 값과 머리의 세그멘트 x y값이 모두 동일하면 runGame()을 즉시

종료하도록 한다.

우리는 웜의 머리와 사과의 XY 좌표 사이에서 유사한 충돌 감지 검사를 수행한다.

일치하면 사과의 좌표를 임의의 새 위치로 설정한다. (getRandomLocation()의 반환

값에서 가져온다.)

머리가 사과와 충돌하지 않은 경우 wormCoords 목록에서 마지막 신체 세그먼트를

삭제한다. 인덱스에 대한 음수 정수는 목록 끝에서 계산된다.

따라서 0은 목록의 첫 번째 항목의 인덱스이고 1은 두 번째 항목에 대한 인덱스이지만

-1은 목록의 마지막 항목에 대한 것이고 -2는 두 번째 항목에서 마지막 항목이다.

124~132 행의 코드 : 웜이 이동하는 방향으로 새로운 몸체 세그먼트 (머리에 대한)를 추가한다. 이렇게 하면 웜이 한 세그먼트 더 길어진다. 웜이 사과를 먹을 때 마지막 신체 세그먼트를 삭제하지 않으면 웜의 전체 길이가 하나씩 증가한다. 그러나 라인(89)이 마지막 바디 세그먼트를 삭제할 때, 새로운 헤드 세그먼트가 바로 이후에 추가되기 때문에 크기는 동일하게 유지된다.

웜을 이동하려면 웜Coords 목록의 시작 부분에 새 본문 세그먼트를 추가한다. 본문

세그먼트가 목록의 시작 부분에 추가되기 때문에 새 헤드가 된다. 새 머리의 좌표는

이전 머리의 좌표 바로 옆에 있다. X 또는 Y 좌표에서 1을 더하거나 뺄 지 여부는

웜이 가는 방향에 따라 다르다. 이 새로운 헤드 세그먼트는 100행의 insert() list

메서드를 사용하여 wormCoords에 추가된다.

전체 디스플레이 표면을 배경색으로 채운다. 라인들은 그리드, 웜, 사과, 및 스코어를

디스플레이 표면에 그린다. 그런 다음 pygame.display.update() 를 호출하면

디스플레이 Surface가 실제 컴퓨터 화면으로 그려진다.

60, 61 – START\_TICKS를 전역변수로 선언,현재 TICKS를 START\_TICKS에 대입 – 나중에 현재 TICKS와 START\_TICKS의 차를 구해서 한 판 당의 Playtime을 구하기 위해서이다.

72 – 음악을 랜덤으로 설정하기 위해 1-3의 값을 무작위로 생성하는

Music\_num 생성

73~79 – 난수에 따라서 서로 다른 음악 파일을 실행한다. (체크리스트 5)

81, 82 – 음악 로딩 후 음악을 재생한다.

90~95 – p키를 입력 받으면 배경을 검은색으로 바꾸고 음악을 멈추고 pause()를 호출한 후 화면이 적용될 수 있도록 화면 업데이트를 한다. (체크리스트 6)

110, 114 – 게임 오버될 시 음악을 멈춘다.

139 – drawTime함수를 호출한다 – playTime을 계산해서 화면에 출력한다.

(체크리스트 4)

Def drawPressKeyMsg() : ‘Press a key to play’라는 문구를 화면에 띄운다.(시작 화면, 게임 오버 화면)

시작 화면 애니메이션이 재생되거나 화면을 통한 게임이 표시되는 동안 오른쪽 하단 모서리에 " Press a key to play"라는 작은 텍스트가 있다. showStartScreen() 및 showGameOverScreen() 모두에 코드를 입력하는 대신, 이 별도의 함수에 코드를 넣고 showStartScreen() 및 showGameOverScreen()에서 함수를 호출하면 된다.

Def drawPressKeyMsg\_pause() : ‘Press a key to play’라는 문구를 화면에 띄운다.(pause화면)

Def checkForKeyPress() : 키를 누르고 있는 중인지 아닌 지를 체크하는 함수이다.

먼저 이벤트 큐에 QUIT 이벤트가 있는지 확인한다. 행 160에서 pygame.event.get() 에 대한 호출은 이벤트 큐의 모든 QUIT 이벤트 목록을 반환한다. (QUIT를 인수로 전달하기 때문에). 이벤트 큐에 QUIT 이벤트가 없으면 pygame.event.get() 이 반환하는 목록이 빈 목록이 된다.

117 행의 len() 호출은 pygame.event.get() 이 빈 목록을 반환하면 0을 반환한다. pygame.event.get() 에 의해 반환 된 목록에 0 개 이상의 항목이 terminate() 함수는 161 행에서 호출되고 프로그램은 종료된다.

그런 다음 pygame.event.get() 호출은 이벤트 대기열에있는 모든 KEYUP 이벤트 목록을 가져온다. 키 이벤트가 Esc 키에 대한 것이라면 이 경우에도 프로그램이 종료된다. 그렇지 않으면 pygame.event.get()에 의해 반환된 목록의 첫 번째 키 이벤트 개체가 이 checkForKeyPress() 함수에서 반환된다.

Def showStartScreen() : 게임 시작 시의 화면을 구성하는 함수이다.

Wormy 게임 프로그램이 처음 실행되기 시작하면 플레이어는 자동으로 게임 플레이를 시작하지 않는다. 대신, 플레이어에게 실행 중인 프로그램을 알려주는 시작 화면이 나타난다. 시작 화면은 또한 플레이어에게 게임을 시작할 준비를 할 수 있는 기회를 제공한다.

Wormy 시작 화면에는 "OSW Game!" 텍스트가 화면에 나타난다.

그리고 OSW Game! 이라는 문구에 회전 효과를 부여한다.

155, 156 – Wormy를 OSW Game으로 바꿨다. (체크리스트 2)

165, 166 – 시작화면 시 문구 및 배경 색을 노랑색으로 변경

(체크리스트 3)

Def terminate() : terminate() 함수는 pygame.quit() 및 sys.exit()를 호출하여 게임이 올바르게 종료되도록 한다.

Def getRandomLocation() : 수 많은 픽셀들 중 픽셀을 랜덤으로 반환하여 돌려주는 함수이다. getRandomLocation() 함수는 사과의 새로운 좌표가 필요할 때마다 호출된다. 이 함수는 키가 'x' 및 'y'인 사전을 반환하며, 값은 임의의 XY 좌표로 설정된다.

Def showGameOverScreen() : 게임 오버 시 ‘GAME OVER’라는 문구를 띄워주며 아무 키나 누르면 다시 새 게임을 실행시켜주는 함수이다.

화면 위의 게임은 애니메이션이 없다는 점을 제외하고는 시작 화면과 비슷하다. "Game"과 "Over"라는 단어는 두 개의 Surface 객체로 렌더링된 다음 화면에 그려진다.

Game Over 텍스트는 플레이어가 키를 누를 때까지 화면에 남아 있다. 플레이어가 실수로 키를 너무 빨리 누르지 않도록 하기 위해 228 행에서 pygame.time.wait() 호출로 반 초 동안 일시 중지한다.

그런 다음 checkForKeyPress()가 호출되어 showGameOverScreen() 함수가 시작된 이후 만들어진 모든 주요 이벤트가 무시된다. 주요 이벤트의 일시 중지 및 삭제는 다음과 같은 상황을 방지하기위한 것이다.

202~204 Game Over에 폰트를 적용한다. (체크리스트 7)

Def pause() : 일시정지 기능을 수행하는 함수이다.

224~225 : Paused에 폰트를 적용한다.(체크리스트 7)

230 : pause화면에서의 'Press a key to play.'를 출력한다

233~240 : 게임을 이어서 진행하고 음악을 재생한다. 그리고 화면을 업데이트 한다.

Def drawScore(score) : score를 구현하고 화면에 나타내는 함수이다.

drawScore() 함수는 표시 Surface 객체의 점수 매개 변수에 전달된 점수의

텍스트를 렌더링하고 그립니다.

244 : score에 폰트를 적용한다.(체크리스트 7)

247 : score를 화면에 출력한다.

Def drawTime(time) : Playtime을 구현하고 화면에 나타내는 함수이다.

drawTime() 함수는 표시 Surface 객체의 시간 매개 변수에 전달된 시간의

텍스트를 렌더링하고 그립니다.

251 : Playtime에 폰트를 적용한다.(체크리스트 7)

253 : Playtime를 화면에 출력한다.

Def drawWorm(wormCoords) : worm을 구현한다.

drawWorm() 함수는 웜 몸체의 각 세그먼트에 대해 녹색 상자를 그린다. 세그먼트는 wormCoords 매개 변수로 전달되며, 각 사전에는 'x'키와 'y'키가 있는 사전 목록이 있다. 196행의 for 루프는 wormCoords의 각 사전 값을 통해 루프된다. 그리드 좌표가 전체 창을 차지하고 0, 0 픽셀을 시작하기 때문에 그리드 좌표에서 픽셀 좌표로 변환하는 것은 매우 쉽다. 라인 273, 274에서 좌표 ['x'] 및 좌표 ['y'] 좌표에 CELLSIZE를 곱한다.

라인 275는 라인 276의 pygame.draw.rect() 함수로 전달될 웜 세그먼트에 대한 Rect 오브젝트를 생성한다. 라인 276은 세그먼트에 대해 짙은 녹색 사각형을 그린다. 그러면 위에 더 작은 밝은 녹색 사각형이 그려진다. 안쪽 밝은 녹색 사각형은 오른쪽으로 4픽셀, 셀의 왼쪽 위 모서리 아래 4픽셀로 시작한다. 이 사각형의 너비와 높이는 셀 크기보다 8 픽셀 작기 때문에 오른쪽과 아래쪽에도 4 픽셀 여백이 있다.

Def drawApple(coord) : Apple을 구현한다.

drawApple() 함수는 drawWorm()과 매우 유사하지만, 빨간색 사과는 셀을 채우는 단일 사각형이기 때문에 모든 기능은 픽셀 좌표로 변환하고 사과의 위치와 크기로 Rect 객체를 만든다. 그런 다음이 Rect 객체를 pygame.draw.rect() 함수에 전달한다.

Def drawGrid() : 배경이 되는 격자 무늬를 만든다.

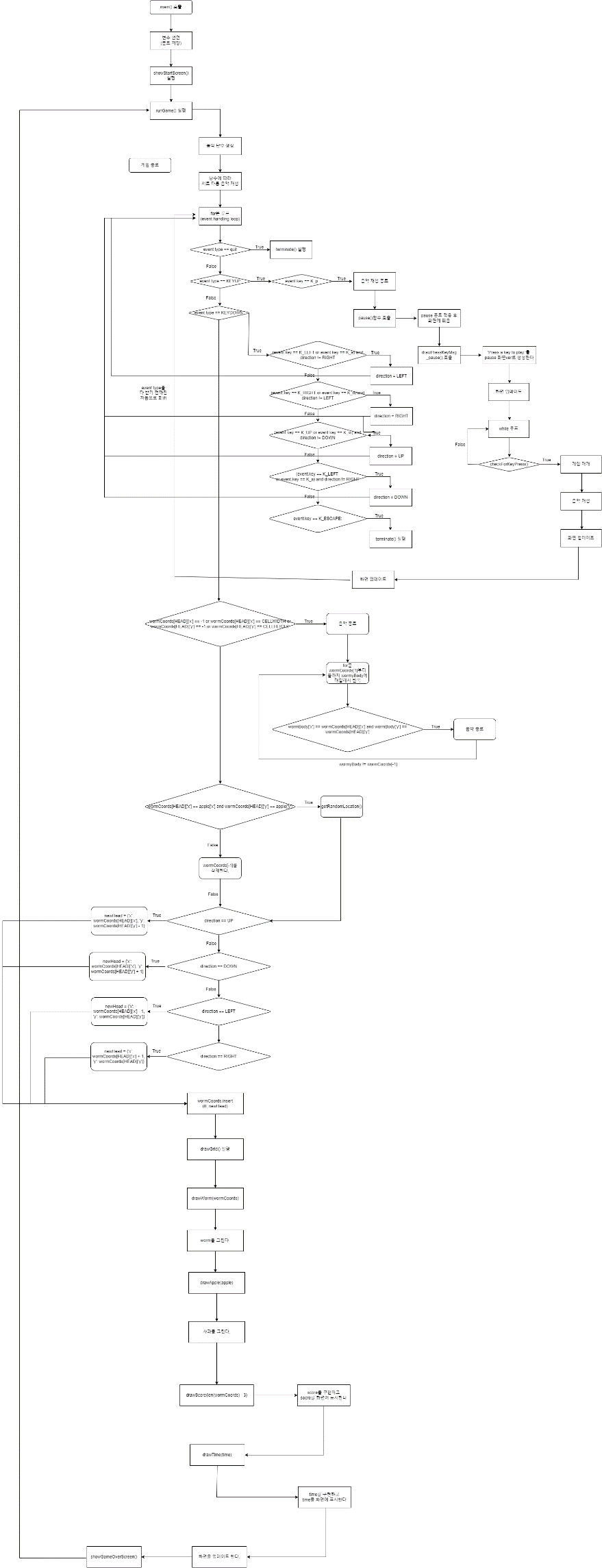
셀 그리드를 더 쉽게 시각화 할 수 있도록 pygame.draw.line() 을 호출하여 그리드의 수직 및 수평 선을 각각 그린다.

일반적으로 필요한 32 개의 세로선을 그리려면

다음 좌표로 pygame.draw.line() 에 대한32 개의 호출이 필요하다.

# 함수의 호출 순서 또는 호출 조건

(별도 png파일이 파일에 있습니다)

****