스도쿠 게임 단위 테스트 및 통합 테스트

소프트웨어프로젝트II 6조

1. 단위 테스트 (Unit Test)

1) testGrid.py

Grid의 createGrid와 createBlank가 정상적으로 동작하는지 확인하기 위한 코드
createGrid()를 하여 grid가 정상적으로 만들어졌는지 text로 출력하고 sudoku.finalCheck를 이용하여 True가 나오는지 확인.
createBlank(1), createBlank(2)를 사용하여 난이도에 ᄄᆞ른 빈칸 개수를 확인. 또한, sudoku.finalCheck를 호출하여 False가 나오는 것을 확인함.

```python
import sys
from grid import Grid
from sudoku import SudokuCheck

grid = Grid()
sudoku = SudokuCheck()

# 그리드 생성되는 것 확인
grid.createGrid()
for row in range(0, 9):
    res = ''
    for col in range(0, 9):
        res += str(grid.originGrid[row][col]) + ' '
    print(res)
print('\n')

# 어려움 모드일 때 blank 생성 확인
print('difficult')
grid.createBlank(1)
for row in range(0, 9):
    res = ''
    for col in range(0, 9):
        res += str(grid.blankGrid[row][col]) + ' '
    print(res)
print('\n')

# 쉬움 모드일 때 blank 생성 확인
print('easy')
grid.createBlank(2)
for row in range(0, 9):
    res = ''
    for col in range(0, 9):
        res += str(grid.blankGrid[row][col]) + ' '
    print(res)
print('\n')

print(sudoku.finalCheck(grid.originGrid))
print(sudoku.finalCheck(grid.blankGrid))
```

```
D:\Anaconda\python.exe C:/Users/SUBIN/PycharmProjects/pythonP
1 5 7 4 9 2 6 3 8
3 8 6 5 1 7 9 4 2
4 9 2 8 3 6 5 7 1
5 2 3 9 4 1 8 6 7
9 7 4 6 8 3 1 2 5
6 1 8 7 2 5 4 9 3
2 3 5 1 6 4 7 8 9
7 6 9 2 5 8 3 1 4
8 4 1 3 7 9 2 5 6


difficult
1 5 7 0 0 2 6 0 0
0 8 0 0 0 7 9 0 2
4 0 2 0 3 6 0 0 1
0 2 3 9 0 1 8 0 7
0 7 4 0 8 0 0 2 5
6 1 0 7 0 5 4 9 3
2 3 5 0 0 0 0 8 9
0 0 9 0 0 0 3 0 0
0 4 0 3 0 9 2 5 0


easy
1 0 0 4 9 2 0 0 0
0 0 6 5 1 7 9 4 2
0 9 0 8 0 6 5 7 0
0 2 3 0 0 1 0 0 7
0 7 4 6 8 0 0 2 5
0 1 8 0 2 5 4 0 3
0 3 5 0 0 4 7 0 9
7 6 9 0 5 8 0 1 0
0 0 1 0 0 9 0 0 6


True
False
```

2) testSudoku.py

SudokuCheck 클래스의 liveCheck와 finalCheck 가 정상적으로 동작하는 지 확인하는 코드

```
    def testLiveCheck(self):
        self.assertFalse(self.sudoku.liveCheck(self.gridBlank, 0, 0, 1))
        self.assertTrue(self.sudoku.liveCheck(self.gridBlank, 0, 0, 2))
        self.assertFalse(self.sudoku.liveCheck(self.gridBlank, 0, 0, 3))
        self.assertFalse(self.sudoku.liveCheck(self.gridBlank, 0, 0, 4))
        self.assertFalse(self.sudoku.liveCheck(self.gridBlank, 0, 0, 5))
        self.assertFalse(self.sudoku.liveCheck(self.gridBlank, 0, 0, 6))
        self.assertFalse(self.sudoku.liveCheck(self.gridBlank, 0, 0, 7))
        self.assertFalse(self.sudoku.liveCheck(self.gridBlank, 0, 0, 8))
        self.assertFalse(self.sudoku.liveCheck(self.gridBlank, 0, 0, 9))

    def testFinalCheck(self):
        self.assertFalse(self.sudoku.finalCheck(self.gridBlank))
        self.gridBlank[0][0] = 1
        self.assertFalse(self.sudoku.finalCheck(self.gridBlank))
        self.gridBlank[0][0] = 2
        self.assertTrue(self.sudoku.finalCheck(self.gridBlank))
        self.gridBlank[0][0] = 3
        self.assertFalse(self.sudoku.finalCheck(self.gridBlank))
        self.gridBlank[0][0] = 4
        self.assertFalse(self.sudoku.finalCheck(self.gridBlank))
        self.gridBlank[0][0] = 5
        self.assertFalse(self.sudoku.finalCheck(self.gridBlank))
        self.gridBlank[0][0] = 6
        self.assertFalse(self.sudoku.finalCheck(self.gridBlank))
        self.gridBlank[0][0] = 7
        self.assertFalse(self.sudoku.finalCheck(self.gridBlank))
        self.gridBlank[0][0] = 8
        self.assertFalse(self.sudoku.finalCheck(self.gridBlank))
        self.gridBlank[0][0] = 9
        self.assertFalse(self.sudoku.finalCheck(self.gridBlank))


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

```
D:\Anaconda\python.exe "D:\Pycharm\PyCharm Community Edition 2021.2.1\plugins\pytho
Testing started at 오후 4:16 ...
Launching pytest with arguments testSudoku.py::TestSudoku::testFinalCheck --no-head

============================= test session starts ==============================
collecting ... collected 1 item

testSudoku.py::TestSudoku::testFinalCheck PASSED                         [100%]

======================== 1 passed, 1 warning in 0.01s =========================

Process finished with exit code 0
```

2. 통합 테스트

- **Test Case 1-1**

게임 시작 시 스도쿠 빈칸을 선택하면 오류가 발생하여 프로그램이 종료됨.

=> 게임 시작 시에 모든 빈칸을 ReadOnly로 설정.

- **Test Case 1-2**

게임 시작 시에 스도쿠 빈칸을 선택하여도 오류 발생하지 않음.

=> Pass

- **Test Case 2-1**

사용자가 grid 빈칸에 숫자를 입력했을 때 9행 9열에 빨간색으로 표시되고 선택된 빈칸의 색상은 변경되지 않는 오류 발견

=> 인덱싱이 잘못되어 있는 것을 발견하여 수정

- **Test Case 2-2**

잘못된 숫자가 입력되었을 때 배경이 빨간색으로 변경되는 것 확인. 빈칸의 숫자를 정답으로 수정하였을 때도 배경 색상이 빨간색으로 유지되는 문제 발견

=> 빈칸의 문자가 변경되면 빈칸의 색상을 흰색으로 처리하는 코드 추가

- **Test Case 2-3**

잘못된 숫자가 입력된 이후 정답을 입력했을 때 배경이 흰색으로 변하는 것 확인

=> Pass

- **Test Case 3-1**
grid 빈칸에 문자와 기호가 입력되는 오류 처리가 되지 않음.
=> 입력 문자에 대한 오류 처리 코드를 추가
- **Test Case 3-2**
grid 빈칸에 문자와 기호 입력하였을 때 입력되지 않게 설정됨.
=> Pass

- **Test Case 4-1**
Start 버튼을 누르기 전 Hint 버튼을 사용하면 프로그램이 오류로 종료되는 문제 발견.
=> 초기 HintNum을 –1로 설정하여 예외 처리
- **Test Case 4-2**
Start 버튼을 누르기 전 Hint 버튼 사용 시 프로그램이 종료되지 않고 정상적으로 작동함.
=> Pass

- **Test Case 5-1**
Submit 버튼을 눌렀을 때 퍼즐을 완성해도 게임 성공 판정이 되지 않음.
=> finalCheck 메서드 중 liveCheck를 호출할 때 현재 위치의 숫자를 두 번 체크하는 문제 때문인 것으로 확인. 현재 위치의 숫자를 0으로 설정하여 liveCheck 호출
- **Test Case 5-2**
Submit 버튼을 누른 이후 버튼을 다시 누르면 퍼즐을 완성해도 게임 성공 판정이 되지 않음
=> Submit 버튼을 누르면 그리드가 모두 0으로 초기화됨. liveCheck호출 이후 현재 위치 숫자를 원래 숫자로 바꾸어주는 알고리즘 추가.
- **Test Case 5-3**
Submit 버튼을 누르고 게임 성공은 되지만, Submit 버튼을 눌렀을 때 계속 게임 진행 시간이 나타나는 문제 발생.
=> 게임 성공 여부를 확인하는 변수를 추가하여 해결.
- **Test Case 5-4**
Test 5에서 나타났던 문제 모두 해결.
=> Pass