

소프트웨어 상세 설계서

소프트웨어프로젝트II 6조

1. mytime 모듈

- startTime 메서드

time 모듈을 이용하여 게임이 시작되었을 때의 시작 시점을 starttime 변수에 저장한다.

- endTime 메서드

time 모듈을 이용하여 게임 종료 시의 시점을 endtime 변수에 저장한다.

int(종료 시점 - 시작 시점)을 사용하여 초 단위의 게임 진행 시간을 리턴한다.

출력 인자 : int(endtime - starttime)

2. sudoku 모듈

- liveCheck 메서드

입력 인자 : grid, row, col, num

grid의 입력된 row를 순회하며 num과 동일한 숫자가 있으면 False 리턴

grid의 입력된 row를 순회하며 num과 동일한 숫자가 있으면 False 리턴

grid의 (row, col)이 포함된 small grid에 num과 동일한 동일한 숫자가 있으면 False 리턴

모든 조건 만족하면 True 리턴

출력 인자 : True / False

- finalCheck 메서드

입력 인자 : grid

cnt = [0, 0, 0, 0, 0, 0, 0, 0, 0]를 선언하여 1~9의 숫자가 grid 내에 몇 번 사용되었는지 확인

grid를 순회하여 조건 확인

입력되지 않은 빈칸이 있는 경우 False 리턴

cnt에 숫자 사용 횟수를 저장

행, 열, small grid에 겹치는 숫자가 있는 경우 False 리턴. liveCheck 메서드를 활용

반복이 종료된 이후 cnt의 값이 모두 9가 되었는지 확인하여 9가 아닌 요소가 있다면 False 리턴

모든 규칙 만족하면 True 리턴

출력 인자 : True / False

3. grid 모듈

- promising 메서드

입력 인자 : x, y(행, 열)

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]로 선언

입력된 행 내에 이미 사용된 숫자가 있으면 numbers에서 제거함.

입력된 열 내에 이미 사용된 숫자가 있으면 numbers에서 제거함.

x, y가 존재하는 small grid에 이미 사용된 숫자가 있으면 numbers에서 제거함.

행, 열, small grid에서 사용되지 않은 숫자들만 리턴

출력인자 : numbers

- dfs 메서드

입력 인자 : k(그리드 인덱스)

재귀함수를 멈추도록 설정한 신호가 들어오면 멈추도록 설정함. bool 형태의 terminateFlag 변수를 사용함.

그리드 인덱스가 9X9 그리드 범위를 넘으면 재귀 멈추는 신호를 설정함.

그리드 인덱스를 행, 열로 변환

promising 메서드를 이용하여 그리드 인덱스 위치에 들어갈 수 있는 숫자 후보를 받음.

숫자 후보 리스트가 비어있을 경우 False 리턴하며 상위 재귀로 이동

숫자 후보 리스트를 무작위로 섞어 일정한 패턴이 나오지 않도록 설정. 리스트의 배열 순서는 백트래킹 탐색 순서가 됨.

숫자 후보 중 한 숫자를 현재 그리드 인덱스 위치에 저장한 후 dfs(k+1)의 값을 확인하여 True가 나오면 True를 리턴하여 재귀를 벗어나고 False가 나오면 다음 숫자 후보를 현재 그리드 인덱스 위치에 저장하여 탐색을 진행한다.

더 이상 가능한 숫자 후보가 없다면 False를 리턴한다.

출력 인자 : True / False

- createGrid 메서드

재귀 함수 신호를 False로 초기화하여 dfs 메서드를 사용할 수 있도록 설정

originGrid를 모든 요소가 0의 값을 갖는 9x9 2차원 배열로 초기화

initNum = [1, 2, 3, 4, 5, 6, 7, 8, 9]로 설정한 후 랜덤 셔플시켜 originGrid의 1열에 저장한다.

dfs(9)를 호출하여 2~9열 스도쿠를 생성한다.

- createBlank 메서드

입력 인자 : difficult

blankGrid를 모든 요소가 0의 값을 갖는 9x9 2차원 배열로 초기화

blankGrid를 순회하며 difficult가 1일 경우 1/2의 확률, difficult가 2일 경우 1/3의 확률로 빈칸을 생성한다.

random.randint(0, difficult)의 값이 0의 아닐 경우 originGrid의 값을 blankGrid에 저장하는 방식으로 빈칸 생성

4. game 모듈

- **__init__** 메서드

qt designer에서 grid 인덱싱이 불가능하므로 __init__ 메서드에서 스도쿠 칸의 그리드를 gridText[81]에 저장하여 인덱스를 설정함.

콜백 함수 구현

초기화면에서 gridText를 모두 setReadOnly True 설정하여 빈칸에 입력이 불가능하게 설정
hintNum을 -1로 초기화하여 힌트 버튼을 눌러도 아무 반응 없도록 설정

- **startGame** 메서드

grid 레이아웃을 빈칸으로 초기화함. text를 삭제하고 setReadOnly를 False로 설정

난이도를 확인하여 dif에 1 또는 2를 저장하고 createBlank에 전달함.

createGrid와 createBlank(dif)호출

고정 숫자칸 setReadOnly True 설정과, 텍스트 출력, 배경 색상 설정

startTime 메서드를 호출하여 게임 시작 시간 저장

힌트 개수 초기화

게임 성공 여부 False로 초기화

- **hintClicked** 메서드

게임 시작 전 힌트 사용 불가능하도록 hintNum이 -1이면 에러처리

힌트를 모두 사용하면 힌트 사용 불가능

row, col 입력을 받아 잘못된 입력(숫자가 아닌 경우, 그리드 범위를 넘는 경우)이 있으면
에러처리

힌트를 사용할 위치에 숫자 출력, 배경 색상 변경, setReadOnly True 설정

blankGrid에 숫자를 입력

힌트 개수 차감

상태 표시줄에 잔여 힌트 개수 출력

- **submitClicked** 메서드

finalCheck메서드를 호출하여 게임 규칙 만족하는지 확인.

게임 성공하면 endTime 호출하여 게임 시간 출력.

gridText 수정 불가능하게 모든 빈칸 setReadOnly True 설정

success를 True로 설정하여 게임 성공 처리

이외의 경우 상태표시줄에 Try Again 출력.

오류 있는 경우 상태표시줄에 Error 출력

- **submitEntered** 메서드

빈칸에 입력되면 배경 색상 흰색으로 변경

현재 인덱스의 blankGrid 0으로 설정

입력 오류 처리(숫자가 아닌 경우, 1~9의 범위를 넘는 경우) 빈칸 비우기

liveCheck 메서드를 호출하여 규칙 만족하는지 확인 규칙 만족하면 배경 흰색, 규칙 위반하
면 배경 붉은 색.