

스도쿠 게임 확장

프로그래밍 기초 CSE1017 (23786) 기말 프로젝트



교수님 저희 잘했조

한양대학교 인공지능학과

2023063772 최준석 (발표자)

응용물리학과

2022006971 이민성

2021022242 전경배

2023.06.14

목차

1. 팀원 소개
2. 스도쿠 게임 소개 및 분석
3. 개선점
 1. 개선점
 2. 개선사항 1
 3. 개선사항 2
 4. 개선사항 3
4. 요구사항
5. 알고리즘
6. 실행 안내
7. `time.time()`
8. 보완할 점

1. 팀원 소개

〈교수님 저희 잘했조〉

구성원

- 팀장, 코딩, 발표자료 제작 이민성(응용물리학과 2022006971)
- 조원1, 발표자료 제작 - 전경배 (응용물리학과 2021022242)
- 조원2, 발표 - 최준석 (인공지능학과 2023063772)

2. 스도쿠 게임 소개 및 분석

- 스도쿠 게임

- 4 x 4 미니 스도쿠
- 좌표를 이용한 조작
- 숫자를 중복 되게 못두게 하기
- 난이도 선택
- 무작위로 생성한 스도쿠 게임 문제를 제출
- 빈칸을 맞추면 빈칸이 채워진 후 다시 출력

- 분석 사항

- 스도쿠가 너무 작음
- 결과 메시지를 출력할 때 맞춘 문제만 고려하고 소요시간은 고려하지 않음
- 스도쿠를 그저 해결하는 기능 밖에 없어서 너무 심심함

3. 개선점

- 모드 추가
 - 타임어택 모드를 추가하여 재미를 더함

```
Choose Game Mode!  
Normal=1, TimeAttack=2:
```

- 게임 확장
 - 4x4 뿐만 아니라 6x6과 9x9모드도 추가해서 더 어려운 버전을 플레이 할 수 있게 함

```
Enter your level.  
4x4=1, 6x6=2, 9x9=3:
```

- 시간 측정
 - 각 게임마다 문제를 푸는데 걸린 시간을 측정해서 안내

```
It took you 25.37 seconds to complete the game.  
Well done! Come again.
```

개선사항 1

- 모드 추가
 - 타임어택 모드를 추가하여 재미를 더함

```
Choose Game Mode!  
Normal=1, TimeAttack=2:
```

일반 모드

```
Choose Game Mode!  
Normal=1, TimeAttack=2: 1  
Enter your level.  
4x4=1, 6x6=2, 9x9=3: █
```

타임어택 모드

```
Choose Game Mode!  
Normal=1, TimeAttack=2: 2  
Enter your level.  
4x4=1, 6x6=2, 9x9=3: █
```

개선사항 2

- 게임 확장

- 4x4 뿐만 아니라 6x6과 9x9모드도 추가해서 더 어려운 버전을 플레이 할 수 있게 함

```
Enter your level.  
4x4=1, 6x6=2, 9x9=3:
```

4	2	.	.
1	3	.	.
2	.	.	3
3	1	4	2

1	3	.	5	4	.
.	4	1	6	3	5
6	2	.	.	1	.
5	.	.	.	2	4
.	.	4	1	6	.
.	6	3	.	.	1

.	7	.	.	5	.	1	4	3
2	5	.	3	4	1	9	7	8
3	4	6	5	.
.	.	.	1	3	4	7	.	.
1	3	.	9	8	7	5	2	6
9	8	.	6	2	.	4	.	1
.	9	.	.	6	.	.	.	4
.	1	3	7	.	8	2	6	5
5	6	.	.	1	3	.	9	7

개선사항 3

- 시간 측정
 - 각 게임마다 문제를 푸는데 걸린 시간을 측정해서 안내

일반 모드

```
It took you 21.91 seconds to complete the game.  
Well done! Come again.
```

타IMER택 모드

- 시간 안에 성공 X

```
Time is up! Game Over.
```

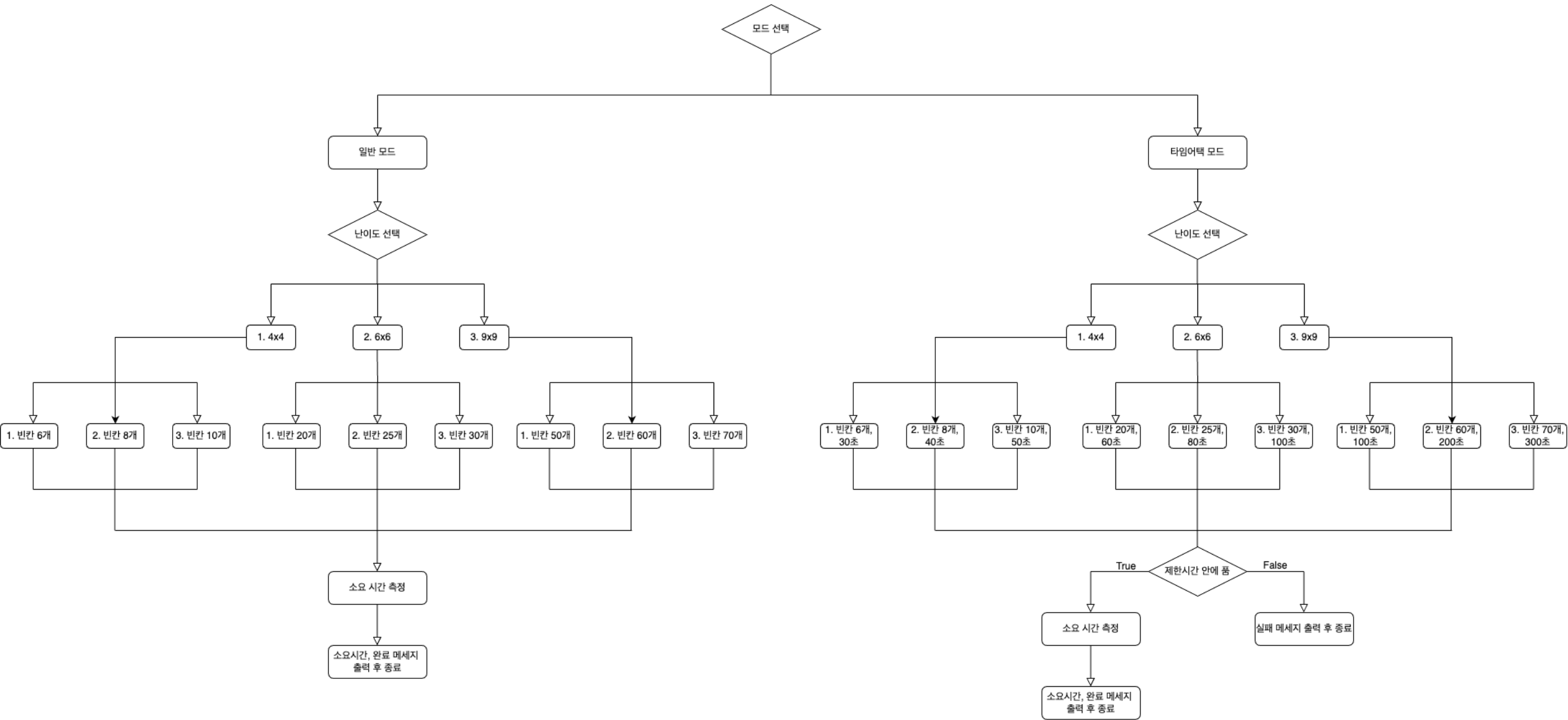
- 시간 안에 성공

```
It took you 26.2 seconds to complete the game.  
Well done! Come again.
```


4. 요구사항

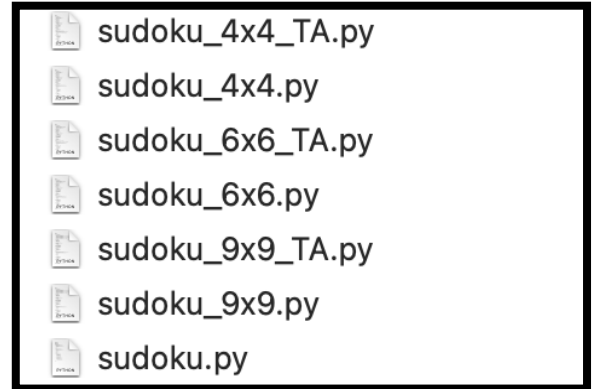
- 게임을 시작할 때 일반 모드와 타임어택 모드를 선택할 수 있게 한다.
- 가로 4칸, 세로 4칸의 정사각형 보드에 1부터 4까지 숫자를 넣는다.
- 전체 보드의 가로/세로줄 모두 1에서 4까지의 숫자를 하나씩만 넣는다.
- 2x2으로 4등분한 조각 보드에도 1에서 4까지의 숫자를 하나씩만 넣는다.
- 빈칸이 많은 수록 퍼즐 풀이가 어려워 지므로, 난이도를 선택할 수 있게 한다.
 - 초급: 빈칸 6개 / 중급: 빈칸 8개 / 고급: 빈칸 10개
- 타임어택 모드에서는 난이도를 선택할 때 빈칸 수 뿐만 아니라 제한 시간도 선택할 수 있게 한다.
 - 초급: 30초 / 중급: 40초 / 고급: 50초
- 가로 6칸, 세로 6칸의 정사각형 보드에 1부터 6까지 숫자를 넣는다.
- 전체 보드의 가로/세로줄 모두 1에서 6까지의 숫자를 하나씩만 넣는다.
- 3x3으로 4등분한 조각 보드에도 1에서 6까지의 숫자를 하나씩만 넣는다.
- 빈칸이 많은 수록 퍼즐 풀이가 어려워 지므로, 난이도를 선택할 수 있게 한다.
 - 초급: 빈칸 10개 / 중급: 빈칸 15개 / 고급: 빈칸 20개
- 타임어택 모드에서는 난이도를 선택할 때 빈칸 수 뿐만 아니라 제한 시간도 선택할 수 있게 한다.
 - 초급: 60초 / 중급: 80초 / 고급: 100초
- 가로 9칸, 세로 9칸의 정사각형 보드에 1부터 9까지 숫자를 넣는다.
- 전체 보드의 가로/세로줄 모두 1에서 9까지의 숫자를 하나씩만 넣는다.
- 3x3으로 9등분한 조각 보드에도 1에서 9까지의 숫자를 하나씩만 넣는다.
- 빈칸이 많은 수록 퍼즐 풀이가 어려워 지므로, 난이도를 선택할 수 있게 한다.
 - 초급: 빈칸 20개 / 중급: 빈칸 30개 / 고급: 빈칸 40개
- 타임어택 모드에서는 난이도를 선택할 때 빈칸 수 뿐만 아니라 제한 시간도 선택할 수 있게 한다.
 - 초급: 100초 / 중급: 200초 / 고급: 300초
- 일반 모드에서 문제를 다 풀면 게임을 플레이하는데 걸린 시간을 계산 한 후 메세지로 출력할 수 있게 한다.
- 타임어택 모드에서 제한시간 안에 문제를 풀면 걸린 시간을 계산 한 후 메세지로 출력하고, 못 풀면 종료 메세지를 출력한 후 프로그램을 종료하게 한다.

5. 알고리즘



6. 실행 안내

- 실행 파일은 sudoku.py입니다.
- sudoku_4x4.py, sudoku_4x4_TA.py, sudoku_6x6.py, sudoku_6x6_TA.py, sudoku_9x9.py, sudoku_9x9_TA.py는 sudoku.py 실행을 위한 모듈입니다.
 - sudoku_nxn.py (n = 4, 6, 9)
 - normal 모드에서 sudoku를 실행하기 위한 모듈
 - sudoku_nxn_TA.py (n = 4, 6, 9 / TA: TimeAttack의 줄임말)
 - TimeAttack 모드에서 sudoku를 실행하기 위한 모듈



7. time.time()

```
import time
```

```
time.time()
```

```
# time.process_time() 대신 time.time()을 사용한 이유:  
# time.time()은 실제 시간을 나타내는 함수이고, time.process_time()은 프로세스 시간을 나타내는 함수이다.  
# 프로세스 시간은 CPU 사용 시간만을 고려하기 때문에 실제 경과 시간과는 차이가 있을 수 있다.
```

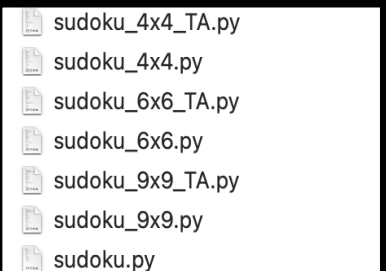
8. 보완할 점

```
start = time.time()
while no_of_holes > 0:
    i = get_integer("Row#(1,2,3,4): ",1,4) - 1
    j = get_integer("Column#(1,2,3,4): ",1,4) - 1
    if puzzle_board[i][j] != 0:
        print("Not empty!")
        continue
    n = get_integer("Number(1,2,3,4): ",1,4)
    if n == solution_board[i][j]:
        puzzle_board[i][j] = solution_board[i][j]
        show_board(puzzle_board)
        no_of_holes -= 1
    else:
        print(n, ": Wrong number! Try again.")
    if time.time() - start > time_limits:
        print("Time is up! Game Over.")
        return
end = time.time()
print("It took you", round(end - start, 2), "seconds to complete the game.")
print("Well done! Come again.")
```

← sudoku_4x4_TA.py의 일부 코드

제한시간 코드가 while문의 맨 아래에 있어,

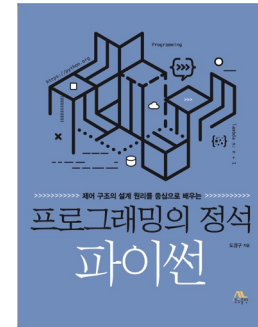
제한시간이 넘어가도 스도쿠를 출력을 한 뒤 종료가 됨 → 제한시간이
되면 바로 종료가 되게끔 보완해야 함



- sudoku_4x4_TA.py
- sudoku_4x4.py
- sudoku_6x6_TA.py
- sudoku_6x6.py
- sudoku_9x9_TA.py
- sudoku_9x9.py
- sudoku.py

모듈에 중복되는 함수가 많은데 모듈을 하나 더 만들어서 중복되는 함수를 넣어 코드의 가독성을 높이도록 보완해야 함

Q & A



감사합니다

