

2023년 2학기

프로그래밍과 문제해결

Assignment #4

담당 교수: 윤은영

학번: 20230499

학과: 무은재학부

이름: 김재환

POVIS ID: carotinoid

명에서약(Honor code)

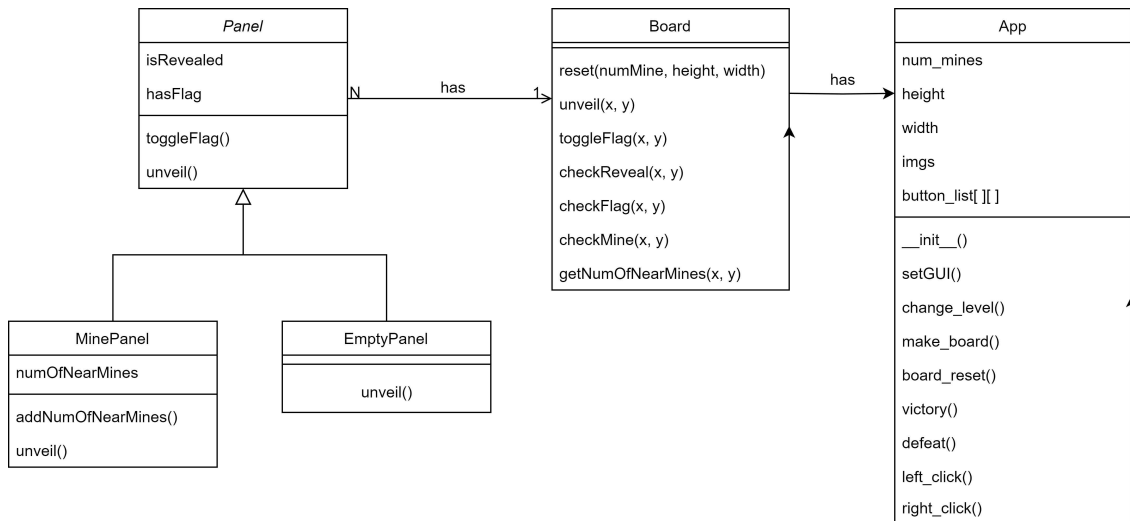
“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”

1. 문제의 개요

본 프로그램을 간략히 설명하면 다음과 같다.

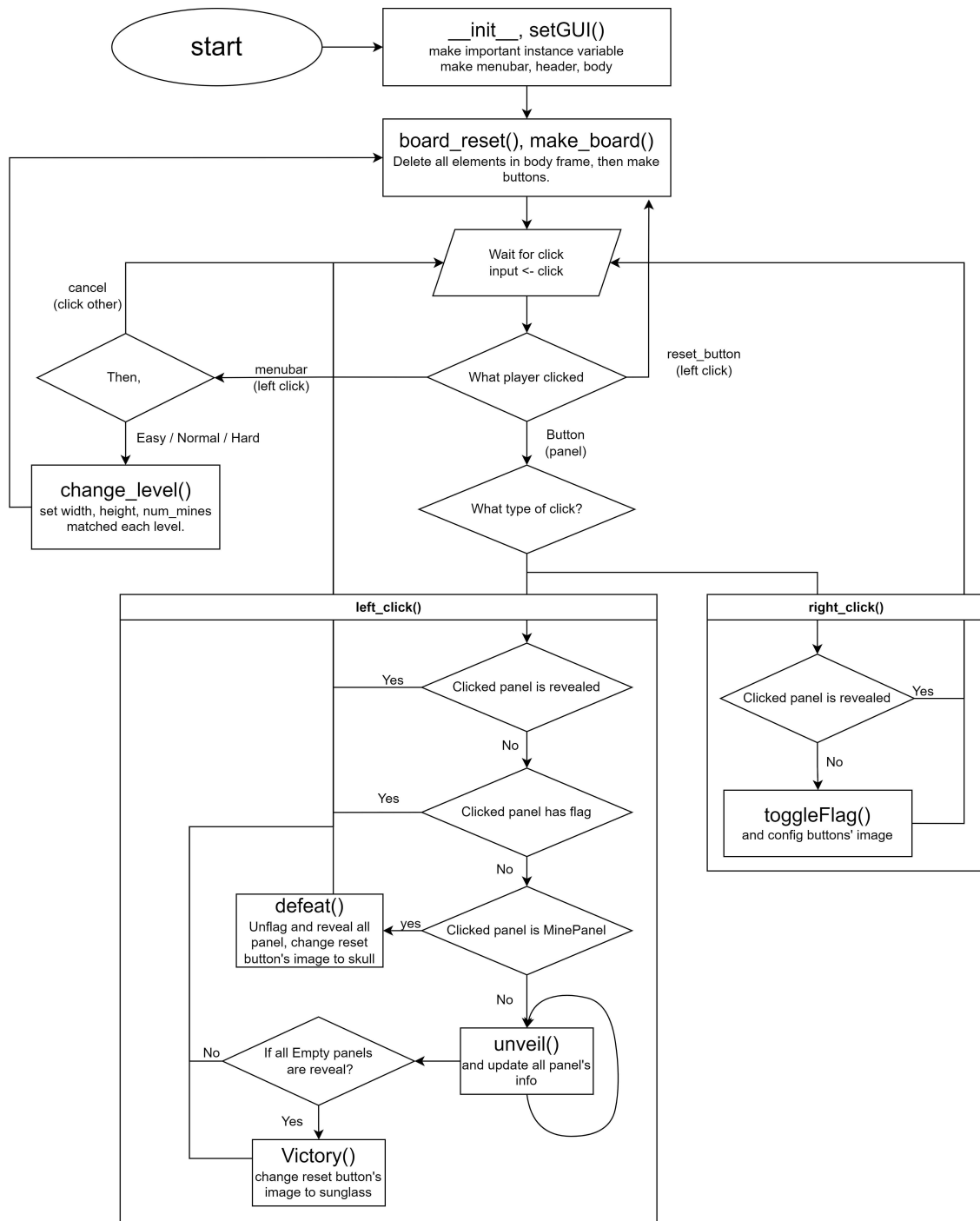
- 본 프로그램의 이름은 지뢰 찾기이며, 기억력과 추리력을 요구하는 간단한 게임이다.
- 지뢰를 피해 모든 빈칸을 찾으면 승리, 지뢰를 클릭하면 패배한다.
- 보드의 각 칸은 지뢰가 있는 칸과 없는 칸으로 나누어지며, 지뢰가 없는 칸을 클릭한 경우 인접한 8칸에 대해 지뢰의 개수를 숫자를 통해 힌트로 제공한다.
- 좌클릭으로 해당 칸을 밝히고, 우클릭으로 해당 칸에 깃발을 세워 지뢰를 표시할 수 있다.
- 좌클릭으로 칸을 밝혔을 때, 근처에 지뢰가 없다면 주위에 재귀적으로 밝힌다.

본 프로그램의 구조도(structure chart)는 다음과 같다. 해당 프로그램은 클래스를 통해 관리되므로 클래스 다이어그램을 나타내었다. 각 메서드의 관계는 workflow chart를 참고하라.



2. 알고리즘

본 프로그램의 흐름도(work flow chart)는 다음과 같다. 해당 프로그램의 종료는 코드로 표현되지 않으므로 흐름도에 표현하지 않았으며, 윈도우(창) 종료 버튼을 누르면 종료된다.

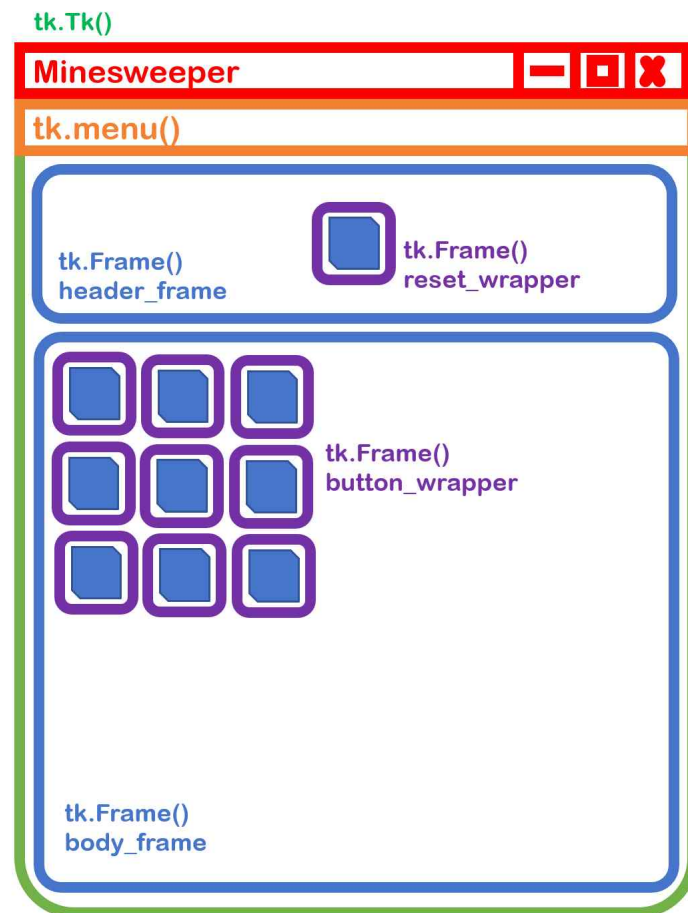


본 프로그램의 핵심은 GUI를 구현하는 법을 배우고 사용하는 것이기 때문에, input을 명령 줄이 아닌 클릭으로 전달받아 그에 할당된 함수(메서드)를 실행하는 것이 중요하다. 이 부분은 플레이어가 클릭한 것과 좌클릭인지 우클릭인지 물어보는 방식으로 흐름도에 구현하였다. 또한 버튼에 config로 접근하여 이미지를 변경하는 것은 출력이 아닌 process의 과정으로 생각하여 흐름도를 작성하였다.

3. 프로그램 구조 및 설명

a) 프로그램의 GUI 구조

tkinter는 파이썬으로 그래픽 UI를 만드는 라이브러리 중 하나로, 매우 간단하다는 특징을 갖는다. 전체 `tk.Tk()` 안에 프레임과 여러 요소를 배치할 수 있으며, 프레임 안에 또한 마찬가지로 프레임과 요소들을 배치할 수 있다. 각 요소를 배치하는 방법에는 `pack`과 `grid`가 있다. `pack`은 위에서 아래로 차례대로 배치하는 방법이며, `grid`는 격자 형태의 위치에 요소를 배치하는 방법이다. 한 프레임 안에서 `pack`과 `grid` 코드가 공존할 수 없다. 본 프로그램의 GUI 구조를 그림으로 나타내면 다음과 같다.



법이다. 한 프레임 안에서 `pack`과 `grid` 코드가 공존할 수 없다. 본 프로그램의 GUI 구조를 그림으로 나타내면 다음과 같다.

b) 좌클릭, 우클릭

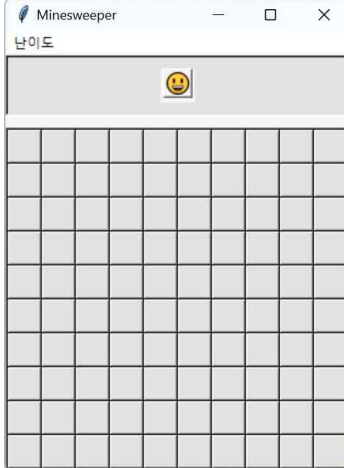
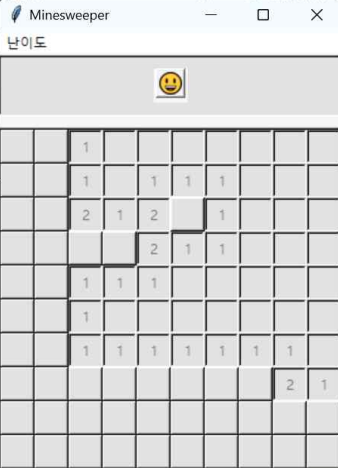
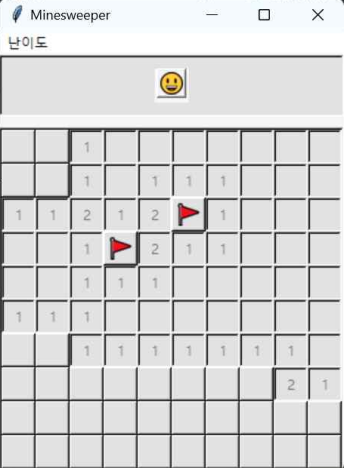
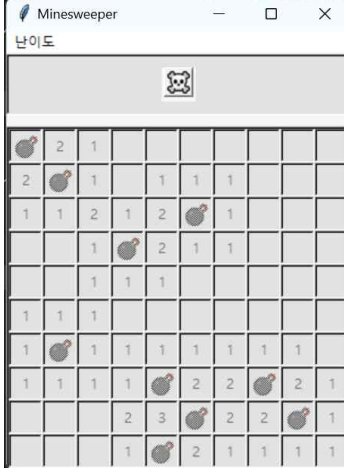
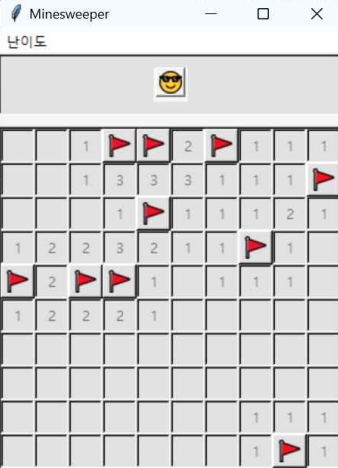
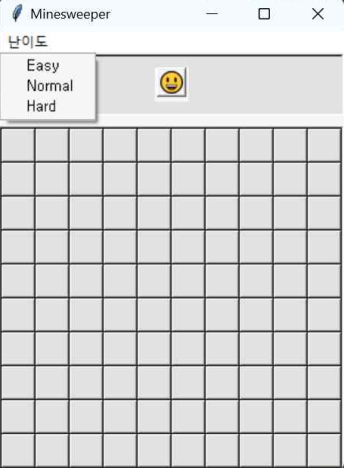
버튼의 상태에 따라서 좌클릭 한 번에 다양한 경우가 생기므로, 흐름을 적절하게 제어하는 것이 중요하다. 좌클릭의 경우 깃발이 표시되어 있거나 이미 밝혀진 경우, 아무것도

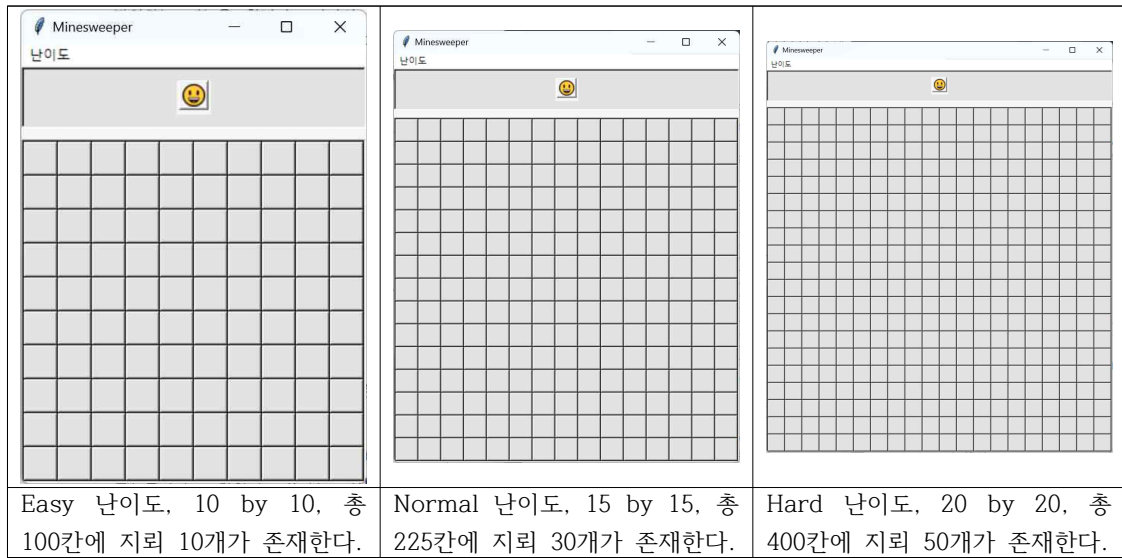
일어나지 않는다. 해당 위치가 지뢰라면 게임은 종료되고 플레이어는 패배하며, 해당 위치가 빈칸이라면 인접한 8칸의 지뢰의 수를 표시한다. 만약, 인접한 8칸의 지뢰의 수가 0인 경우, 해당 과정을 각 8칸에서 다시 반복한다. 이후 전체 패널에서 모든 EmptyPanel이 밝혀졌는지 검사하고, 승리인지 판정한다. 우클릭의 경우 밝혀진 칸일 때는 아무 일도 일어나지 않고, 밝혀지지 않은 칸이면 깃발을 세워 지뢰라고 생각되는 칸을 표시할 수 있다.

c) 클래스

Model.py에서는 지뢰 찾기의 로직을 담당하는 클래스가 작성되어 있으며, 구조는 문제의 개요 항목의 클래스 다이어그램을 참고하라. 이때 Board는 Panel 클래스의 인스턴스로 구성되지만, app.py에서는 Board 클래스만을 import 하므로 isinstance(~, EmptyPanel)과 같은 문법은 사용할 수 없다. 또한 Board 클래스는 지뢰 찾기의 기능을 담당하는 클래스이지만, GUI에서의 버튼은 Board 클래스에서는 없으므로 app.py에서 버튼을 저장하는 이차원 리스트를 만들어 관리하였다. 각 메서드의 역할은 코드의 주석을 참고하라.

4. 프로그래밍 실행 방법 및 예시

		
<p>초기화면. 프로그램이 처음 실행되면 Easy 난이도의 10*10 크기의 보드가 출력된다.</p>	<p>한 번 클릭했을 때, 주위 지뢰가 없는 칸일 때 재귀적으로 인근 칸을 모두 오픈한다.</p>	<p>지뢰라고 의심되는 칸을 우 클릭하면 해당 위치에 플래그를 세울 수 있다. 플래그는 클릭해도 아무 일도 일어나지 않는다.</p>
		
<p>지뢰를 클릭하면, 세워둔 플래그가 제거되면서 모든 칸이 밝혀진다. 또한 리셋버튼이 해골 그림으로 바뀐다. 리셋버튼을 눌러 초기화면으로 돌아갈 수 있다.</p>	<p>모든 빈칸을 밝혀내면 승리하며, 리셋버튼이 선글라스 이미지로 바뀐다.</p>	<p>메뉴바의 난이도 탭에서 난이도를 선택할 수 있다. 난이도는 Easy, Normal, Hard 3가지가 있다.</p>



5. 토론

- 지뢰를 클릭해 패배하거나, 한 번에 연속해서 밝혀질 때 플래그가 꽂혀 있더라도 모든 플래그를 제거하고 밝혀내야 한다. 다시 말해, SUNKEN 상태이면서 image = flag인 버튼은 존재할 수 없다.
- 칸을 밝힐 때, 주위에 지뢰가 없는 칸일 경우 주변 칸을 이어서 만드는 기능은 재귀함수를 통해 구현하였다. 우선 클릭한 패널을 밝힌 후, 밝혀진 패널이 근처에 지뢰가 없는 EmptyPanel인 경우 주위의 3*3칸을 이중 반복문으로 순회한다. 이때 x좌표는 0과 너비 사이, y좌표는 0과 높이 사이, 그리고 밝혀지지 않은 칸이면 그 칸을 다시 밝히는 원리이다.

6. 결론

- 본 과제에서는 객체지향프로그래밍의 클래스와 인스턴스를 사용해 보드의 각 패널을 구성하고, 클래스와 함께 tkinter 라이브러리를 불러와 GUI로 지뢰 찾기를 제작하였다. 이 과정에서 tkinter 라이브러리가 동작하는 원리, GUI가 구성되는 원리를 학습하였으며, 특히 frame 안에 여러 요소를 넣는 방법인 pack과 grid의 차이를 배웠다.
- 기존 명령줄 인터페이스 환경에서는 각 명령을 사용자가 외워서 직접 타이핑 해야 하며, 특히 이번 프로그램 같은 경우 한 번의 명령을 실행하기 위해 좌/우, 버튼의 위치 총 3개의 인자를 작성해야 하는 불편함이 있었다. 이를 GUI로 구현하면서 단순한 클릭을 통해 명령을 실행하게 되었다. 이를 통해 CLI와 GUI 환경의 차이점과 장단점을 학습하였다.

7. 개선 방향

- 실제 지뢰 찾기 게임에 있는 기능을 구현하면 사용자 경험을 더 좋게 만들 수 있을 것이다. 예를 들어, 주변 지뢰에 모두 플래그를 세웠을 때, 밝혀진 칸을 휠 클릭(<Button-2>) 하면 주변 8칸 중 밝혀지지 않은 칸을 모두 밝혀 주는 화음 기능이 있다. 그 이외에, 원래 지뢰 찾기에는 첫 클릭은 항상 지뢰가 아니어야 하며, 또 어느 지뢰 찾기에서는 첫 클릭은 항상 넓은 영역을 밝혀 주기도 한다.
- 다른 지뢰 찾기 게임을 보면 클래식 지뢰 찾기를 넘어서 다양한 효과나 디자인을 가진 게임이 많다. 단순히 구글에 지뢰 찾기를 검색해서 할 수 있는 것도 플래그를 취소하면 뿔혀서 떨어지거나, 지뢰를 클릭했을 경우 근처 지뢰부터 차례대로 터지는 효과가 있다. 이런 것처럼 tkinter의 테마 기능을 이용하여 더 좋은 그래픽 디자인을 작성하면 좋을 것이다.