

A program felépítése

- main.py
 - Itt történik a fő ablak legenerálása tkinter könyvtár segítségével, és a GameConfigurator és a Minesweeper osztályok meghívása.
- GameConfigurator.py
 - Bekéri a játékostól a sorok, oszlopok és aknák számát és példányváltozóiban tárolja. Ha bármelyikre nem kap értéket, akkor egy alapértelmezett értékkel tér vissza.
- GameAppearance
 - A játék megjelenítéséhez szükséges információkat tárolja.
- Cell.py
 - Itt tárolódnak a mezőkhöz tartozó paraméterek.
- Minesweeper.py
 - Ez az osztály felelős a játék fő logikájának kezeléséért. Bemenő paraméterekként a main-ben példányosított tkinter osztályt és a GameConfigurator példányváltozóit kapja meg.
 - Fő komponensek:
 - create_buttons() -> None
 - Legenerálja a mezőket, amelyhez hozzárendeli az egér kattintásokat és a hozzájuk tartozó actiont, ezt pedig a Cell példány kapja meg.
 - A Cell példányokat egy mátrixhoz adja hozzá
 - reveal(cell: Cell) -> None
 - Megjeleníti, hogy hány szomszédos mezőn található akna
 - show_mines(cell: Cell) -> None
 - Megjeleníti az összes mezőt, ahol akna található
 - reveal_zero_field(row: int, col: int) -> None
 - Ha olyan mezőre kattint a játékos, amellyel egy akna sem szomszédos, akkor Depth First Search algoritmus használva felfedi azt a területet, ahol nem található akna

További fejlesztési lehetőségek:

- A játékidő limitálása
- Ranglista kialakítása