

Programozói dokumentáció

Adatszerkezet:

- Definiáltam egy struktúrát, amiben az alábbiakat tárolja a program:

typedef struct mezo

```
{  
    int sz;  
    bool zaszlo, akna, latott;  
} mezo;
```

- Az "sz" integerben tárolja a program minden mezőn azokat a számokat, amelyek jelölik, hogy hány akna van az adott mező körül.
- A "zaszlo" boolean tárolja azt, hogy az adott mezőre tett-e zászlót a felhasználó. Ha nem tett, akkor false, ha tett, akkor true. Ha már zászlózott mezőt ismét megzászlóznak, akkor azzal eltüntetik a zászlót róla, azaz ismét false lesz.
- Az "akna" boolean tárolja, hogy az adott mezőn van-e akna. Ha van, akkor true, ha nincs, akkor false.
- A "latott" boolean mutatja, hogy az adott mező tartalma (szám, vagy akna) látható-e már a felhasználó számára a console-on. Ha true, akkor látható, ha false, akkor nem, azaz még felfedhető a mező tartalma.

A főprogram működése:

- A mezőket egy kétdimenziós dinamikus tömbben tárolja a program, amelynek méreteit a felhasználó adja meg azzal együtt, hogy hány aknát szeretne a táblára. Az aknák számát az "a" integerben tároljuk.
- Miután feltöltjük a kétdimenziós tömböt "mezo" típusú változókkal és beállítottuk azok értékeit, a program random átállítja annyi mező "akna" boolean értékét true-ra, amennyi aknát a felhasználó kért. Ezután az aknával feltöltött mezők körül lévő mezőkön kiszámolja, hogy milyen számnak kell lennie (erről a függvényről később lesz szó).
- Miután a program felrajzolta a táblát, a felhasználó megadja annak a mezőnek koordinátáit, amelyiket fel szeretne fedni, vagy meg szeretne zászlózni. Ekkor rögzít egy időt is, aminek a segítségével a játék végén kiszámolja, hogy meddig tartott a játék.
- A program addig kéri be a választani kívánt mezők koordinátáit, amíg véget nem ér a játék. Minden választás után meghívja a "felderit" függvényt, megvizsgálja, hogy vége van-e a játéknak, és ha nincs, akkor letörli, majd újra kirajzolja az aktuális táblát.
- A játék végén kiírja, hogy nyert vagy veszített a felhasználó, és mennyi ideig tartott ezt elérnie. Ezután felszabadítja a kétdimenziós dinamikus tömbnek foglalt memóriaterületet.

A program működését vezérlő fő függvények:

- void kiir(int maxi, int maxj, mezo** tomb)
Első és második paramétere a felhasználó által megadott táblamagasság és táblaszélesség, harmadik paramétere a kétdimenziós tömb. A függvény kiírja a mezőket, és a tábla mellé és alá egy számozást, hogy ne kelljen a felhasználónak számolgatni a mezőket egy nagyobb tábla esetén. Vizsgálja, hogy a mezők közt van-e zászlózott, melyik mező van felfedve, a felfedett mezők közt van-e akna, és ezek alapján dönti el, hogy mit ír ki az egyes mezőkre.
- void szamol(int i, int j, mezo** tomb, int x, int y)
Paraméterként kapja egy adott mező koordinátáit, a kétdimenziós tömböt, és a tábla magasságát, szélességét. Megnöveli az adott mező "sz" integerének értékét eggyel, ha teljesülnek a feltételek, azaz a mező létezik.
- void felderit(int choosex, int choosey, mezo** tomb, int zszl, int x, int y)
Paraméterként kapja a választott mező két koordinátáját, a kétdimenziós tömböt, a "zszl" integert, aminek értéke a zászlózást jelöli, és a tábla magasságát, szélességét. A függvény egy vizsgálattal indít, hogy az adott mező nem látható-e még. Ha nem látható, akkor megvizsgálja, hogy zászlózni akarja-e a felhasználó a mezőt, vagy felfedni. Ha zászlózni akarja, akkor negálja a "zaszlo" booleant. Ha nem zászlózni akarja, akkor felfedi a mezőt, és megvizsgálja, hogy az adott mező "sz" integerének értéke nulla-e. Ha nulla, akkor felfedi a körülötte lévő mezők értékét is, és ha azok közt is van 0, akkor rekurzívan folytatja az akörül lévő mezők felfedésével, és így tovább.
- int kilep(mezo** tomb, int x, int y, int a)
Paraméterként kapja a kétdimenziós tömböt, a tábla magasságát és szélességét és az akna számát. Visszatérési értéke határozza meg, hogy milyen eredményt (Nyertel!/Vesztettel!) ír ki a főprogram a játék végén.
- void ora_indul()
Rögzíti, hogy hány másodperc telt el 1970. január 1. éjfél óta.
- void ora_leall()
Rögzíti, hogy hány másodperc telt el 1970. január 1. éjfél óta.
- time_t stopper()
Kivonja egymásból az ora_leall() és az ora_indul() függvények által rögzített másodpercek számát és a különbség a játék időtartamát adja.
- void clrscr()
Törli a console tartalmát.