Aknakerescho – programozoi dokumentacio

1. A projekt felepitese, forditasi tudnivalok

A program C nyelven keszult, kulso konyvtarat nem igenyel a szabvanyos C header-eken kivul.

A projekt tobb modulra van bontva, hogy a kód atlathato legyen:

main.c
– a program entry pointja, a fomenut kezeli
– meghivja a jatekmenetet es a kulonfele modulokat
– itt tortenik a nehezseg beallitasa is

ui.c
– minden, ami a terminal kimenetet illeti
– fomenuk kiirasa
– nehezsegi menunek a szoveges felulete
– palya kiirasa fix szelesseggel
– kepernyo torlese

parancs.c
– a felhasznalo altal beirt parancsok beolvasasa
– hibakezeles (rossz formatum, tullepes, stb.)
– muvelet es koordinatak szetosztasa
– a jatekmenet modulnak elokeszitett parancs visszaadasa

board.c
– a palya letrehozasa (dinamikus 2-dimenzios tomb)
– bombak veletlenszeru lerakasa
– korulotte levo bombak megszamlalasa
– a tabla memoriajanak felszabaditasa a jatek vegen

game.c
– a jatek fo ciklusa
– mezo megnyitasa (rekurzivan tovabbnyitja az ures mezoket)
– flag toggle
– gyozelem vizsgalata
– kapcsolat a parancs es ui-modullal

history.c
– a history.csv fájl kezelese
– file letrehozasa, ellenorzese, fejlec potlasa
– eredmeny hozzaadasa
– eredmenyek kiirasa a felhasznalonak
– teljes torles

Forditas:

```
gcc main.c ui.c parancs.c board.c game.c history.c -o aknakereso
```

Futatashoz nincs kulon beallitas, a terminalbol indithato az aknakereso binaris fajl.

2. Adatszerkezetek es tervezes indoklása

A jatek alapeleme a Cella nevű struktura:

```
typedef struct {
    int isBomb;
    int isOpen;
    int isFlagged;
    int around;
} Cella;
```

A tabla egy Cella** tipusu ketdimenzios tomb. Azért választottam mert:

– egyszeru lekezelni a soronkénti malloc miatt
– a free_table fuggvenyben a sorok felszabaditasa egyesével tortenik, ami atlathato
– a mezok logikai allapotai egyutt maradnak egy strukturban
– a jatek futasa kozben nem kell ujra szamolni a szomszedos bombakat, mert a palyageneralas eleve minden mezohoz eltarolja az "around" erteket

A Beallitasok strukturat azert hoztam letre, hogy egy helyen kezeljem a player nevet es a jatek parametereit (palya merete, bombak szama). Ez csokkenti a fuggvenyek szukseges parametereinek szamat, es atlathatobb.

A tabla generálása (board modul) is egyszeru: eloszor letrehozom a tabla 2D tombjet, majd rakok le bombakat veletlenszeruen. A veletlenszeru lerakasnal ellenorzom, hogy ugyanarra a mezore ne keruljon ket bomba.

A parancsok kulon modulba szervezese azert tortent, mert a jatek elejen nagyon duplikalodott a kod (m5:3 es m 5:3).

A jateklogika marad a game modulban, hogy a parancsok es megjelenites ne keveredjen.

A history mozgasa kulon modul, mert a fajlkezeles altalaban kulonallo modul. Igy a main-ben csak annyi van, hogy "eredmenyek_kiir()" es tarsai.

3. Fuggveny-dokumentacio

ui modul

kepernyo_torol()
– torli a terminal kepernyojet

menu_kiir()
– fomenu kiirasa

nehezseg_menu_kiir()
– kiirja a nehezsegi beallitas menut

tabla_kiir(tabla, rows, cols, mutasd_bombakat)
– kiirja a palyat szepen igazitva(volt, hogy el volt csuszva)
– parameterek:
  tabla: a Cella** tabla
  rows, cols: sor es oszlop meret
  mutasd_bombakat: ha 1, akkor a bombakat is kiirja (jatekok vegen)

jatek_allapot(b, tabla, kezdes)
– kiirja a jatek allapotat idovel es nevvel
– parameterek: beallitasok pointer, tabla pointer, kezdes ideje

parancs modul

parancs_beolvas(buffer, hossz, muvelet*, sor*, oszlop*)
– a felhasznalo bevitelt szetbontja
– buffer meretet figyeli, hogy ne legyen tullepes
– visszater: 1 ha sikeres, 0 ha hibas input

parancs_vegrehajtasa(b, tabla, muvelet, sor, oszlop, gyozott*)
– vegrehajtja a parancsot (megnyit vagy flag toggle)
– visszater: 1 ha a jatek veget ert (bomba vagy gyozelem), 0 ha mehet tovabb

board modul

tabla_gen(rows, cols)
– letrehozza a 2-dimenzios tablat, minden mezo alaphelyzetben
– visszater: Cella**

bomba_gen(tabla, rows, cols, bombs)
– veletlenszeruen elhelyezi az aknakat

korulotte_szamit(tabla, rows, cols)
– minden mezore elore kiirja, hany bomba van korulotte

free_table(tabla, rows)
– felszabadítja a dinamukusan foglalt tablat

game modul

megnyit(tabla, sor, oszl, rows, cols)
– megnyit egy mezot, uresnel rekurziv nyitast vegez
– visszateresi ertek: -1 bomba, 1 siker, 0 ha mar nyitva van

flag_valt(tabla, sor, oszl, rows, cols)
– flag valtas egy mezon

gyozelem_check(tabla, rows, cols)
– visszater 1-el ha nyert a jatekos

jatek(b)
– maga a jatek fo ciklusa
– itt hivodik minden mas modul

history modul

history_init()
– elokesziti a history.csv fajlt
– hibas fajl esetén ujra letrehozza
– erre szukseg volt, mert folyamat csak appendelte a fejlecet a legelejen

eredmeny_ment(nev, ido, meret)
– a CSV fajl vegere ir egy uj sort

eredmenyek_kiir()
– kiirja a fajl tartalmat

eredmenyek_torol()
– torli a fajlt, majd ujra letrehozza fejleccel

Amelyik fuggvenynel nincs emlitve visszateresi ertek, ott nincs

A szines-print reszet sajnos kitoroltem, mert ahogy atjottem linuxra valamiert nem mukodott es inkabb nem kockaztattam.