

Programozói dokumentáció

Aknakereső

A program felépítése

fájlok: main.c, jatek_logika.h, jatek_logika.c, memoriakezeles.h, memoriakezeles.c, megjelenes.h, megjelenes.c, econio.h, econio.c, debugmalloc.h, dicslista.txt

A econio.h, econio.c, debugmalloc.h függvénykönyvtárak nem a saját szellemi termékem. A tárgy honlapjáról töltöttem le ezeket. (<https://infoc.eet.bme.hu/>)

jatek_logika.c/.h: Ebben a fájlban a játék alapvető logikai funkcióinak működéséhez szükséges függvények, illetve struktúrák találhatók. Például: aknamező legenerálását végző függvény.

memoriakezeles.c/.h: Ebben a programhoz szükséges dinamikus memóriafoglalást és dinamikus memóriafelszabadítást végző függvények találhatók.

megjelenites.c/.h: Ebben az egyes menürendszerek, az aknamező, a dicsőséglista, illetve az időmérő kiíratását végző függvények szerepelnek.

main.c: A program főmodulja. Ebben összpontosulnak az egyes moduloknak a függvényei, itt látszik a program teljes felépítése.

Adatszerkezetek dokumentációja:

typedef struct Cella:

Ez a struktúra a játéktér építőeleme, mely megadja, hogy egy adott cella bomba-e, fel van-e fedezve, meg van-e jelölve, hány darabszomszédos cellája van, illetve hogy hogyan néz ki az aktuális cella megjelenítéskor.

typedef enum Jatekallas:

Ennek a változónak a segítségével van számontartva, hogy egy adott játékos a játéknak épp melyik fázisában van.

Függvény dokumnetációk:

Cella **palyafoglal(int oldal):

A játék pályájához szükséges mátrix dinamikus lefoglalása. A függvény nem kap külön szélességet és magasságot, mivel a játéktér ugyanolyan széles, mint amilyen magas. Lefoglal a megadott méretben sorfolytonos leképezéssel helyet a memóriában, amely a játékhoz szükséges pályát fogja tartalmazni, és visszatér a lefoglalt pályára mutató pointerrel.

void palyafelszabadit(Cella **palya):

A függvény felszabadítja a játék nehézségének kiválasztása során lefoglalt dinamikus memóriaterületet, azzal, hogy megadjuk neki a lefoglalt terület címét.

char **lista_foglal(int mag, int szel):

A függvény a dicsőséglistának foglal helyet a memóriában. Paraméterként megkapja, hogy hány eleme legyen a listának(mag), és hogy az elemek milyen szélesek legyenek. Végül visszatér a listára mutató pointerrel.

void lista_felszabadit(char **lista, int mag):

A függvény felszabadítja a dicsőséglistának lefoglalt memóriaterületet. A paraméterei a felszabadítandó lista pointere és a lista magassága, azaz elemeinek a száma.

void palya_feltolt(int meret, Cella **palya, char c):

A függvény feltölti a megjelenítendő játéktér egyes celláinak változóit. Paraméterként megkapja a pálya méretét, a feltöltendő pályát, valamint kap egy karaktert, amivel az összes cella kinézetét beállítja arra az adott karakterre.

void palya_kirajzol(int meret, Cella **palya):

A függvény kirajzolja a megjelenítendő pályát és a pálya mellett, illetve fölött elhelyezkedő segédindexeket. Paraméterként megkapja a kirajzolandó pálya méretét és magát a kirajzolandó pályát.

bool kornyezo_cellak(int max, int x,int y):

Ez a függvény megnézi, hogy egy adott koordinátájú cella rajta van-e a pályán, vagy túl van indexelve. Paraméterként megkapja a pályaméretet, valamint a vizsgálandó cella x és y koordinátáját. Végül visszatér a megfelelő logikai értékkel.

void palya_generator(int meret, Cella **palya, char c, int bombak_szama):

A függvény először kitölti az argumentumban megadott karakterrel az előre lefoglalt memóriaterületet. Ezután véletlenszerűen generált koordinátákra bombákat helyez el, majd a maradék mezőkbe beleírja, hogy az egyes cellák körül hány bomba található. Paraméterként megkapja a pálya méretét, a pályára mutató pointert, egy karaktert, amivel minden más művelet előtt feltölti a pályát, illetve, hogy hány bombát kell szétszórnia a pályán.

bool nyert_e(Cella **palya, int bombak, int meret):

Megnézi, hogy nyert-e már a felhasználó úgy, hogy megvizsgálja, hogy a felfedezettlen cellák száma megegyezik-e az adott pályán található bombák számával. Paraméterként megkapja a pályára mutató pointert és a méretét, valamint, hogy hány bomba van a pályán. Legvégül visszatér a megfelelő logikai értékkel.

void cella_felfedez(int x, int y, int meret, Cella **valos, Cella **megjel):

A függvény automatikusan felfedezi az egymás mellett lévő üres cellákat. Paraméterként megkapja kiinduló cella x és y koordinátáját, és két egyenként egy-egy pályára mutató pointer párost.

int jatekmenet(Jatekallas j, Cella **palya_v, Cella **palya_m, int meret, int bombak):

A függvény maga a játék futtatását végzi. Vizsgálja, hogy veszített vagy nyert-e már a játékos vagy még folyamatban van a játék. Másrészről kezeli azt is, hogy a játékos a megadott koordinátát felfedni, megjelölni akar vagy éppen jelölést szeretne eltávolítani. Paraméterként kap egy a játékalást tároló felsorolt típusú változót, két egyenként egy pályára mutató pointert és egy bombaszámot. Visszatérési értéke pedig az, hogy egy játékos mennyi idő alatt vitt végig egy adott pályát. Viszont ha veszített a játékos akkor a visszatérési érték 0.

char menu_2():

Ez a függvény egy menüt ír ki, ha a játékos befejezte a játékmenetet (nyert vagy veszített). Visszatérési értéke egy karakter, aminek a függvényében visszatér minket a főmenübe vagy bezárul a program.

char *jnev_beolvas():

A függvény beolvas egy karaktersorozatot addig amíg nem nyom enter a felhasználó. Visszatérési értéke pedig egy string, amiben a felhasználó által begépelte karaktersorozat található.

void menu_1(int *palyameret, int *bombak, char **dlista):

Menürendszer melyben a játékos ki tudja választani, hogy játszani vagy a ranglistát megnézni szeretné-e. Ezen felül, ha a játék indítása menügomb után kiválaszthatja, hogy milyen nehézségi fokon szeretne játszani a játékkal. A függvény cím szerint megadott változókba elmenti a felhasználó által kiválasztott nehézségi szint adatait

int idomero(clock_t kezdes):

Ez a függvény az idő mérésére és kiíratására szolgál. A paraméterként kapott időtől mérve írja ki az eltelt időt, illetve adja vissza az eltelt másodpercek számát.

void dicslista_kiir(char **dlista):

Ez a függvény kiírja a file-ból beolvasott dicsőséglista elemeit. Egyetlen paramétere a dicsőséglistára mutató pointer.

void ujrekordot_felvesz(int mp, char *jnev, char **dlista):

A függvény paraméterként kap egy számot, mely másodpercet tartalmaz, kap egy stringet, mely a játékos nevét tartalmazza, valamint kap egy file-ból beolvasott listára mutató pointert. A függvény létrehoz egy stringet melyet a dicsőséglista megfelelő helyére beszúr.

char **dicslista_elment():

A függvény file-ból beolvas beolvasott szöveget, melyet dinamikusan foglalt memóriába lement. Visszatérési értéként pedig a többdimenziós tömbre mutató pointert ad, melyben a szöveg sorai vannak eltárolva.