Programozói dokumentáció

A program felépítése

A program indításakor, megjelenik a menü, mely az egyes menüpontok kiválasztását követően meghívja a megfelelő függvényeket a megfelelő paramétereivel.

A játék folytatása, az új játék kezdete, illetve a készített pályán való játék kezdete mind ugyan azt a függvényt hívják meg, más-más paraméterekkel.
A toplista és tudnivalók menüpont mind egy másik ablakot nyitnak meg, különböző függvényekkel.
A pálya készítő, illetve a beállítások menüpont helyileg nyitnak meg egy másik kisebb menüt, ahonnan azokat vezérelni lehet. Az előbbi két menüpont közül ad választást, melyek: a készített pálya játszása, illetve a pálya készítése, ezek új ablakot nyitnak. Ezeken kívül van még egy kilépés menüpont, melynek függvénye több helyen is előfordul a programban, ezért tanácsos újrafelhasználhatónak írni.

Az játékot betöltő függvény létrehozza a játékhoz szükséges változókat pl.: lépések száma, hányadik pálya, pálya mérete, maga a karaktertömb, ami a pályát tartalmazza stb., majd feltölti ezeket egy függvénnyel, mely beolvas megfelelő fájlokból és át konvertálja a tömböt, a paramétereknek megfelelően.
Meghívódik a játékot irányító központi függvény, mely a megadott paraméterekkel dolgozik, beolvassa a felhasználó utasításait, az alapján cselekszik: mozgatja a játékost, betölti az előző pálya állását(visszalépés), előröl, kezdi a pályát, vagy behozza az játék belső menüjét. Valamint kirajzolja a pályát, illetve kezeli azt is, ha sikerült teljesíteni egy pályát. Majd a felhasználó döntése alapján, le kell mentei az eredményt a toplistába, illetve az eddigi játék állását, hogy később folytatható legyen.

A pálya készítő menüpont kiválasztása után, választania kell a felhasználónak, hogy betölteni akarja a korábban beolvasott pályát, avagy készíteni akar egy másikat. Ez alapján, ha játszani akar, meghívódik az előbb is használt játék betöltő függvény, ha pályát akar szerkeszteni, nyílik egy új ablak melyben rajzolni lehet objektumokat, azok kiválasztásával, majd enterrel való „leütésével”. A függvény szintén kezeli, ha a játékos ki akar lépni, ezt a játék belső menü függvénnyel érjük el, mely visszatérési értékei alapján dolgozik a külső függvény.

A beállítások és tudnivalók, egy-egy függvényből állnak, az utóbbi egy hosszú szöveget ír ki a konzolra, míg a másik egy irányítható menüt nyit, mellyel lehet élőben változtatni a témákat.

Adatszerkezetek választása

A pálya méretei tárolásához egy két egészet tároló struktúrát választottam.

```
typedef struct coordinates
{
    int x;
    int y;

} coordinates;
```

A toplista kiírása binárisan történik, mely három egészet és egy karakter tömböt tárol, melyben a felhasználó leírhatja véleményét a pályáról, de lényegében bármit írhat 50 karakter hosszúságig.

```
typedef struct scoreinfo
{
    char name[50];
    int level;
    int steps;
    int stepbacks;
} scoreinfo;
```

Enumerátorok használandók: a hibakezelésre,

```
enum EXIT_CODE
{
    OPENING_MAPS            = 90,
    OPENING_LOADED_MAPS     = 91,
    ONE_DIM_ARRAY_MALLOC    = 92,
    TWO_DIM_ARRAY_MALLOC1   = 93,
    TWO_DIM_ARRAY_MALLOC2   = 94,
    SAVING_TO_LOADED_MAPS   = 95,
    SAVING_MAP_LIST_MALLOC  = 96,
    SAVING_NEW_LIST_MALLOC  = 97,
    OPENING_HIGHSCORE_WRITE = 98,
    OPENING_HIGHSCORE_READ  = 99,
    TWO_DIM_ARRAY_MALLOC3   =100,
    TWO_DIM_ARRAY_MALLOC4   =101,
    OPENING_CREATEDMAP_WRITE=102,
    OPENING_CREATED_MAPS    =103,
};
```

melyeket a hibakezelő függvény használ. A menüpontok helyzetének tárolására, illetve a menü vezérlésére.

```
enum MENU
{
    LOAD_MENU   = 5,
    LOAD_SAVED  = 7,
    LOAD_GAME   = 9,
    NEW_GAME    = 11,
    MAP_CREATOR = 13,
    HIGH_SCORE  = 15,
    ABOUT       = 17,
    OPTIONS     = 19,
    EXIT        = 21,
    EXIT_SAVED  = 22,
    DONT_EXIT   = 23,
    LOAD_CREATED= 24,
    CONTINUE    = 0,
    STEPBACK    = -1,
};
```

Az objektumok kezelésére,

```
enum WALLS
{
    NOTHING             = '0',
    WALL                = '1',
    BOX                 = '2',
    FINISHZONE          = '3',
    BOX_IN_FINISHZONE   = '4',
    PLAYER              = '5',
    PLAYER_IN_FINISHZONE= '6',
};
```

melyeket később is használ a program, például a pálya készítőnél, a rajzolt elem kiválasztásánál. A billentyűparancsok beolvasásánál,

```
enum KEYS
{
    UP      = 'w',
    DOWN    = 's',
    LEFT    = 'a',
    RIGHT   = 'd',
    CTRL_Z  =  26,
    ENTER   =  13,
    ESC     =  27,
    BCKSPC  =   8,
    RESTART = 'r',
};
```

ezek a játék vezérléséért felelnek, például a menüt és a játékot is a ‘w,a,s,d’ karakterekkel lehet irányítani, ezzel lehet még kezelni a visszalépést, a jóváhagyást, és még sok minden mást. Egyszerűbbé teszi a játék konfigurálását hiszen a program mindenhol ezt használja, tehát elég átírni az enum egy változójának értékét és a program máris alkalmazkodik ahoz. Globál változót használtam a szín tárolására, mivel ezt néhány kivételével minden függvény megkapná paraméterként, ezért így jelentősen könyebb kezelni. Kétféle szín tárolandó, egy erősebb és egy halványabb szín, ez segíti elő a játék vezérlését, például ettől látszik melyik menüpont van kiválasztva.

```
int COLOR, LIGHTCOLOR;
```

A pálya egy két dimenziós tömb, melyben karakterek, igazából számok, tárolódnak amelyeket a kirajzoló függvény alakít át, az ember által kedveltebb, szebb és színesebb karakterekké. Azonban a pálya lementéskor és beolvasásakor ez még egy hosszú egy dimenziós karaktertömb. Beolvasáskor először a pálya mérete olvasódik be, ennek segítségével lesz a tömb átalakítható két dimenzióssá, mellyel a többi függvény dolgozik.

A programot vezérlő főbb függvények

Menü:

A játék nevének kirajzoláért felelős függvény, mely ugyan azzal a pálya rajzoló fügvénnyel dolgozik, amivel a játékot vezérlő is. Létrehoz egy “pályát”(karaktertömböt), majd szintén a játék során használandó függvénnyel átkonvertálja két dimenziós tömbbé, amit kirajzol, így lesz a “sokoban” felirat.

A menüpontok kirajzolásáért felelős függvény, mely paraméterként kapja az aktív menüpontot, ezt kell kivilágitani.

A menüpontok választásáért velelős függvény, mely vissza térési értéke a kiválasztott menüpont.

A játék indításáért felelős függvény, paraméterként kapja a játékmódot és az alapján kezeli, a benne lévő függvényeket.

```
void drawmenu( void );
void draw_outtermenu(int y);
int stepmenu(int y);
int newgame(int loader);
```

Játék:

A pálya betöltő és konvertáló függvény, mely paraméterei között van, hogy melyik fájlból olvassa be a pályát.

A pálya lementéséért felelős függvény, mely vissza konvertálja a pályát egy dimenzióssá, és kiírja a pálya méreteit, és az eddigi eredményeket.

A játék motorját kezelő függvény, ebben van a mozgatásért felelős függvény, az eredmény tábla kiírásáért felelős fügvény, a pálya kiírásáért felelős függvény és például a billentyűparancsokat kezelő függvény.

```
int loadnewmap      (coordinates *Map, int loader, int *whichmap, char ***mapfinal, char **mapstart, int *steps, int *stepbackcounter);
int savemap         (char **maptosave, coordinates Map, int whichmap, int steps, int stepbackcounter);
int gamecore        (char **mapfinal, coordinates *Map, int whichmap, int *steps, char **savedmap, int stepbackcounter, int *stepbackenable);
```

Egyéb:

A beállításokért felelős függvény.

A tudnivalók kiiírásáért felelős függvény.

A toplista kiirásáért felelős függvény.

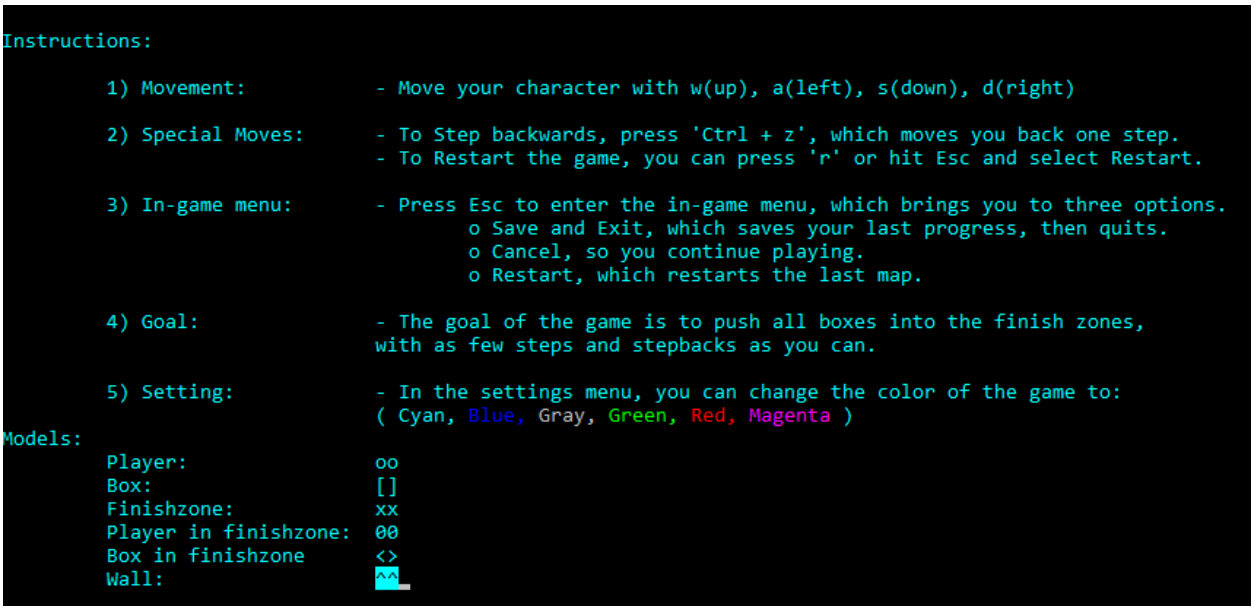
A pálya készítéséért felelős függvény.

```
void options();
void about();
void highscore();
void map_creator_menu();
```

Felhasználási és Tesztelési dokumentáció

Felhasználás

A játék működtetéséhez a tudnivalók megtalálhatóak a játék About szekciójában.



Tesztelés

A hibakezelésre nincs sok lehetőség, hiszen a program csak a nem hibás lépésekre fog reagálni. Hibák csak akkor keletkezhetnek, ha nincs jelen a beolvasandó fájl, vagy a memória foglalás hibásan működött, például:


```
enum EXIT_CODE
{
    OPENING_MAPS            = 90,
    OPENING_LOADED_MAPS     = 91,
    ONE_DIM_ARRAY_MALLOC    = 92,
    TWO_DIM_ARRAY_MALLOC1   = 93,
    TWO_DIM_ARRAY_MALLOC2   = 94,
    SAVING_TO_LOADED_MAPS   = 95,
    SAVING_MAP_LIST_MALLOC  = 96,
    SAVING_NEW_LIST_MALLOC  = 97,
    OPENING_HIGHSCORE_WRITE = 98,
    OPENING_HIGHSCORE_READ  = 99,
    TWO_DIM_ARRAY_MALLOC3   =100,
    TWO_DIM_ARRAY_MALLOC4   =101,
    OPENING_CREATEDMAP_WRITE=102,
    OPENING_CREATED_MAPS    =103,
};
```