Dokumentáció (snake)

***Játék főbb részei:***

* Fő menü
* Játék

*Fő menü:*

Amikor elindul a program, ez fogadja a játékost, ESCAPE gomb megnyomásával lehet kilépni a játékból, ENTER megnyomásával lehet játszani a snake-kel. A grafikát SDL-lel oldottam meg (Simple Directmedia Layer).

*Játék:*

Miután a játékos megnyomta az ENTER gombot, a főmenü ablak eltűnik és legenerálódik a pálya egy „map.txt” fájl alapján. Leképződik a kígyó feje, az alma és a falak.

A játék állapota addig nem változik, amíg a játékos nem lép a fel-le-jobbra-balra nyíllal. Miután lépett, időközönként lefut egy algoritmus, ami lépteti a kígyót abba az irányba, ami utoljára meg volt adva, viszont amint a játékos új irányt ad meg, a kígyó rögtön abba az irányba lép. Úgy ítéltem meg, így könnyebb játszani. Ha a kígyó megeszi önmagát, vagy a falba lép a játék visszalép a menübe és kiírja a vesztés okát. A játék akkor is kilép a menübe, ha ESCAPE gomb le van nyomva, viszont, ha bezárjuk a játék ablakát, akkor az egész program bezárul. A menübe visszalépve a játékot ezután újra lehet játszani.

***Részletes megoldás:***

A játék a main.c-ből indul ki, ott a megadom a random seedet és meghívom a Main Menu-t.

**sdl\_init**: inicializálja az ablakot, aminek csinál renderer pointert és window pointert. Hiba esetén logol. Itt kell megadni az ablak szélességét, nagyságát, a kívánt renderer és window pointer mutatását és az ablak címét is.

**writeTextToWindow**: kiír egy szöveget az ablakra, meg kell adni a kiírni kívánt szöveget, a szöveg színét, renderer pointert box-ot, amibe írja a szöveget, és még 2 szöveggel kapcsolatos pointert

**gameMenu**: itt inicializálom az ablakot sdl\_init-tel, a fontot itt olvasom be és létrehozom a megfelelő pointereket is, meg szövegdobozokat. Iniciálom a gameOver kilépő kondíciót is, ami a hibakódnak felel meg. Ehhez 5 darab kód tartozik:

* Amikor falnak megy a kígyó
* Felfalja magát
* Nem sikerül beolvasni a map.txt-t
* Kilép a játékos a játékból a Main Menu-be
* Kilép a játékos a játékból teljesen

Miután kiírtuk az ablakra, milyen inputra várunk a játékostól, várjuk az inputot, ebből 2 fajta lehet:

* Kilép a menüből (piros x a jobb felső sarokban, vagy ESC)
* ENTER-rel elindítja a játékot

Ha a felhasználó kilépet a menüből, termináljuk a programot és a hozzá lefoglalt memóriaterületet.

Ha a felhasználó elindította a játékot, akkor elindít egy egész értéket visszaadó függvényt, ami a hibakódot adja vissza, így sajnos nem lehet megnyerni a játékot :l.

**spawnSnake**: a pálya nagysága tudatában lespawnolja a kígyót egy véletlenszerű blokkra. A margin egy határt szab meg, milyen közel lehet a falhoz a kígyó

**spawnApple**: lespawnolja az almát egy véletlenszerű blokkra, ami nem a kígyó

**giveHeight**: lekéri a map.txt fájlból a pálya magasságát

**giveLenght**: lekéri a map.txt fájlból a pálya hosszát

**giveSizeOfBlock**: lekéri a map.txt fájlból a blokkok pixeles nagyságát

**createMap**: létrehozza az egész számokból álló mátrixot, ami különböző értékeket vehet fel, viszont egyelőre csak a falakat határozza meg:

* 5-ös a fal
* 3-as az alma
* 0-s a semmi
* 1-es a kígyó

Egy pointerre mutató pointert ad vissza, így könnyen lehet hivatkozni a sor elemeire.

**snakeHeadPointer**: egy új pointert ad vissza, ami a kígyó új fejének felel meg

**snakeNeckPointer**: a kígyó eddigi fejének, mostmár nyakának a pointerét adja vissza

**snakeBeforeNeckPointer**: a kígyó nyakát, vagyis a volt nyakát adja vissza, ami a nyak előtti rész

**snakeNextStateDraw**: kirajzolja az új kígyó állását, ha nem hibás a lépés, gameMap elemeit módosítja, nincs benne ellenőrzés, csúsztatja a láncolt lista elemeit, miközben felszabadítja azokat.

**mapNextState**: visszaadja a gameMap módosított változatát, ami már a frissített pályaállás. Leellenőrzi, hogy a felhasználó adott-e be lépést (input = 0), vagy ki akar e lépni a menübe (input = 6). Meghívja a snakeHeadPointer, snakeNeckPointer és snakeBeforePointer függvényeket, hogy frissítse a kígyó új állását.  
Az input által megadott irányba lépteti a fejet, ha nem a nyak előtti koordinátákkal fog megegyezni a kígyó, ez esetben az ellenkező irányba lép a kígyó, ez azért van, hogyha ellentétes irányba lépnénk a kígyóval ne legyen lehetőségünk, hiszen akkor elvesztenénk a játékot, ha több eleme van, mint 1.  
Ha a kígyó almára lépett, az alma helye lesz a kígyó fejének új helye és lespawnolunk egy új almát.  
Ha a kígyó falnak ütközött, visszaadjuk egy lefixált helyen a hibakódot, és leáll aztán a program.  
Ha a kígyó megette önmagát, szintén visszaadjuk egy lefixált helyen a hibakódot, aztán leáll a program.  
Ha semmi baj a lépéssel, akkor meghívjuk a snakeNextStateDraw függvényt, ami úgy módosítja a kígyó listjájának elemeit, hogy az utólsó eleme le legyen „törölve”.

**freeGameMap**: felszabadítja a pálya által lefoglalt memóriaterületet

**freeLinkedList**: felszabadítja a láncolt lista által foglalt memóriaterületeket

**quitGame**: meghívja a freeLinkedList-et és freeGameMap-et, felszabadítja az alma memóriaterületét

**drawMapState**: kirajzolja a pályát a megadott renderer által

**inputHandling**: kezeli az inputokat. 2 féle input létezik:

* Amikor kilép a játék (ESC főmenübe, x a jobb felső sarokba teljesen kilép)
* Amikor irányt adunk a kígyónak

Ha változik az input, akkor rögtön lépteti az kígyót abba az irányba.

Miután megnyitottuk a játékot, a program megpro álja nyitni a map.txt fájlt, ami tartalmazza a pálya adatait, ami a magasság, szélesség, és a kirajzoláshoz szükséges blokkok nagysága. Ha ez nem sikerül, akkor visszalép a menübe és kiírja a megfelelő hibaüzenetet.

Ha sikerül, beolvassuk az adatokat, majd iniciáljuk a különböző értékeket:

* Milyen gyors legyen a játék (delay)
* Milyen messze legyen az alma a faltól (margin)
* Kígyó helye
* Alma helye
* Input

Aztán elindul a ciklus, ami vár a felhasználóra, hogy megadjon egy irányt, amilyen irányba megy majd a kígyó. A kígyót a nyilakkal lehet irányítani. Ha a mapNextState egyik fixált helyére (gameMap[0][0]) hibakód kerül vagy az input az ablak bezárása, vagy az ESC gomb megnyomása, akkor a ciklus leáll és elkezdődik a termináció.

Először az SDL-lel kapcsolatos dolgokat zárom be, majd definiálok egy kilépő értéket, hogy minden memória helyét fel tudjam szabadítani. Megnézem, melyik értéket kell visszaküldeni, majd bezárom a fájlt és felszabadítom a memóriaterületeket is (quitGame, fclose).

Miután visszatértem a főmenübe, kiírja a hibakódom, vagy azt, hogy üdvözöl a főmenüben.

Jó játékot!