Kígyó játék

by Sárközi Péter

Kígyó játék

### Specifikáció

### Megvalósítás:

A pálya kijelzését és adminisztrálását egy 2 dimenziós mátrix fogja szolgálni. A pálya elemei, a kígyók, az ételcsomagok, mind a mátrixban megfelelően elhelyezett karakterekből fognak állni.

Nagyvonalakban a következő modulokból fog állni a program:

* Mozgás-leképzés/Irányítás
* Pontok számlálása/Mentése
* Interface megvalósítása
* Mátrix műveletek

A program célja, hogy 1, vagy 2 db kígyó folyamatos mozgását modellezze egy pályán.

Egyéb tulajdonságok:

A pályán véletlenszerűen elhelyezett „étel-csomagok” lesznek. Ha egy kígyó ezeken átmegy, növekedni fog.

Az étel-csomagok felvételét pontokkal is díjazza. Ezeket egy „High-score” elven működő listában tárolja, amit fájlba is ment.

Ha kígyó kígyóhoz ér(akár máshoz, akár önmagához), akkor a játék véget ér, az addig összegyűjtött pontszámok lesznek a véglegesek.

Extrák:

Állítható sebesség

Állítható az étel-csomagok száma

Lehetőség van a pályára Teleportokat beállítani, ezek száma állítható.

Beállítható, hogy a pálya falán át lehet-e menni

Lehetőség van időlassításra (BULLET-TIME), 5 ételenként. (Kikapcsolható)

Lehetőség van a függőleges mozgás lassítására

Lehetőség van egyjátékos módra kapcsolni

Minden halálos lépés előtt egy frissítésnyi időt kap a játékos, hogy korrigáljon(Második esély) (Kikapcsolható)

Felhasználó dokumentáció

# LINUX

Ez az oldal csak a linuxos(vagy egyéb UNIX-Like környzetet használó) felhasználóknak szól, így ha ez nem vonatkozik rád, lapozz tovább!

A programhoz mellékelve van egy bináris(kigyo), ami néhány rendszeren futhat, feltétele, hogy az ncurses fel legyen telepítve! Ha mégsem futna, akkor fordítani kell(Nem lapot, forrást)!

Ez esetben ***erősen,*** de tényleg, nagyon ajánlott, hogy telepítsd az ncurses-devel csomagot(vagy libncurses-dev, stb, vagy fordítsd le forrásból és úgy telepítsd /usr prefix-szel), különben a program ANSI módban fog fordulni, amiben mindent lehet vele csinálni, csak teljesen élvezhetetlen lesz!

Ha mindent telepítettél(vagy nem és mindenképp ANSI-ben akarod fordítani), a **forrás főmappájából** (ahol a configure van) futtasd le a configure-t(./configure)! Fontos, hogy abban a mappában legyél, különben nem fogja tudni jól megírni az ncurses-t betöltő header fájlt!

Ezután add ki a

*make clean && make*

parancsot, és ha szerencséd van, hiba nélkül lefordul a játék, és már indíthatod is a ./kigyo paranccsal!

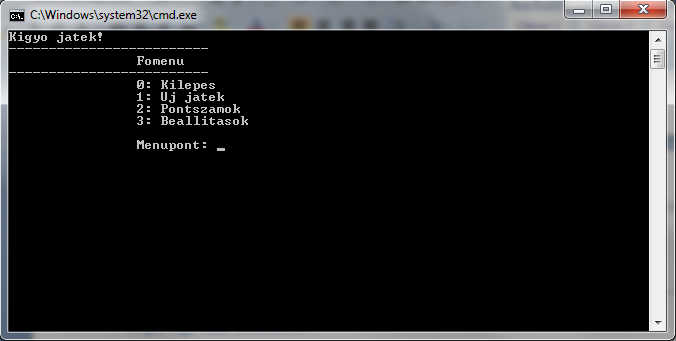
A játék telepítése(make install) nem ajánlott, hacsak a játékot nem olyan felhasználóval futtatod mindig, aki képes írni a /usr/bin-be(Bár ebben az esetben sem ajánlott).

Bármilyen kérdéssel nyugodtan fordulhatsz hozzám: xmisterhu@gmail.com

## A program felülete:

A játék egy konzolos alkalmazás, ezért használhatához egérre nincs szükség.

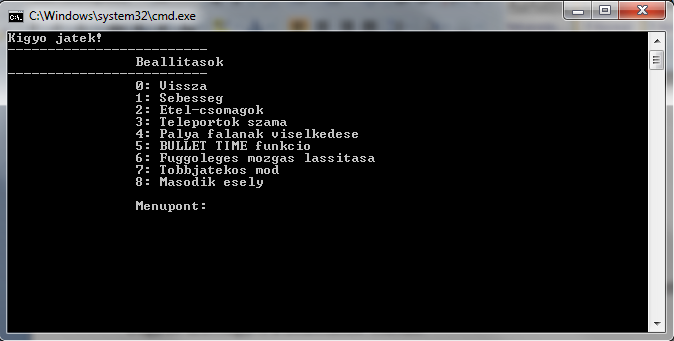
A Főmenü a következőképpen néz ki:



Mint látható 4 menüpont közül választhatunk:

* 0: Kilépés a programból
* 1: Új játék indítása a meglévő beállításokkkal
* 2: A legmagasabb pontszámokat elérő játékosok megtekintése
* 3: Átlépés a Beállítások menübe.

A menüpontok közül a hozzájuk tartozó számbillentyű leütésével választhatunk.

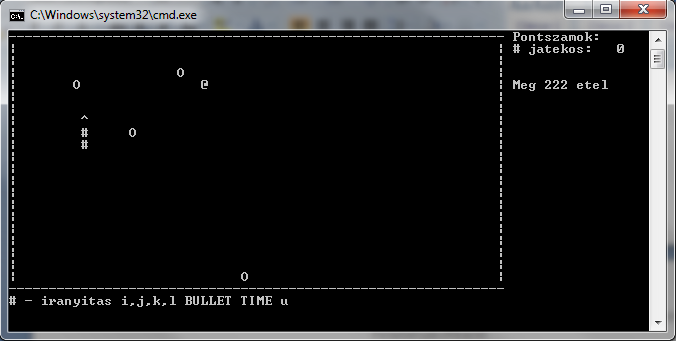
Vegyük szemügyre a Beállítások menüt:

Mint látható 9 menüpont közül választhatunk:

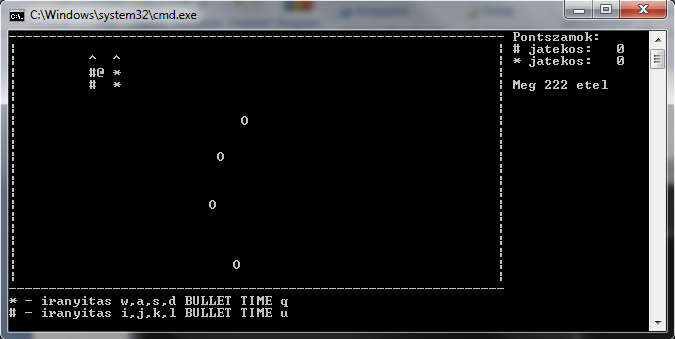
* 0: Visszalépés a Főmenübe
* 1: Sebesség beállítása
* 2: Étel-csomagok számának beállítása
* 3: Teleportok számának beállítása(0 esetén kikapcsolása)
* 4: Pályafal viselkedésének beállítása
* 5: BULLET TIME(Időlassítás) funkció beállítása
* 6: Függőleges mozgás lassításának beállítása
  + Erre akkor lehet szükség, ha a le és fel irányű mozgás túl gyorsnak tűnik az oldalirányúhoz képest.
* 7: Egy – vagy többjátékos mód közti váltás
* 8: Második esély funkció beállítása
  + Ez a funkció minden halálba vezető lépés után hagy még egy kis időt az adott lépés megváltoztatására, ez az idő a beállított sebességtől függ.

# A játék

## Egyjátékos mód



## Többjátékos mód



A fenti képeken látható a játék indulásakor megjelenő kép. Ezen éppen bekapcsolva láthatók a teleportok (O jelek) és az időlassítő funkció (BULLET TIME).

Az étel-csomagokat a ’@’ jelöli.

A kígyók mindig máshol kezdenek, és mindig felfelé néznek.

Minden kígyó akkor indul el, amikor az első irányt megkapja!

Ahogy a pálya alatt látható is, az irányító billentyűk játékosonként:

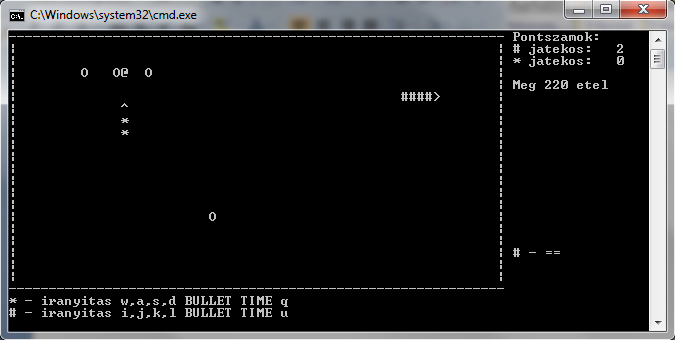
’\*’ jelű játékosnak:

* w: fel
* s: le
* a: balra
* d: jobbra
* q: BULLET TIME bekapcsolása

’#’ jelű játékosnak:

* i: fel
* j: balra
* l: jobbra
* k: le
* u: BULLET TIME bekapcsolása

A pálya jobb oldalán felül található az egyik információsáv, ami a játék aktuális állásáról tájékoztat.

Ahogy a következő képen látszik:Jobb oldalt alul található a BULLET TIME információ sávja, ami jelzi a funkció töltődöttségét, valamint ha feltöltődött egy „BULLET READY” felirattal jelzi azt.

Használatba vételkor bevillan egy „BULLET TIME” felirat, majd pedig azt kísérhetjük itt figyelemmel, hogy mennyi van még hátra a lassú időnkből.

**Megjegyzés:** A játékból mindig lehetőség van kilépni az ESC billentyű megnyomásával, ebben az esetben pontozással dől el a végeredmény.

Fejlesztői dokumentáció

Forrásfájlok

types.h

Tartalmazza az egész programra kiterjedő konstansokat(mátrix dimenziói, étel-csomagok száma), típusokat(pozíció – struktúra, irány – enum, kígyó – láncolt lista, pontszámok tárolásához szükséges adattípusok, menürendszer adattípusai).

kigyo.h,kigyo.c

Tartalmazza a kígyó kezeléséhez használatos függvényeket:

* newK : új kígyó létrehozása véletlen pozícióval.
* incSize : étel-csomagok felvétele után a kígyó méretét megnöveli, ahogy a hozzá tartozó pontszámot is.
* checkpos : minden lépésnél ellenőrzi, hogy a kígyó milyen területre lépett(másik kígyóra, vagy esetleg ételre).
* himself: a lépések végén szolgál az önmagunkra lépés ellenőrzésére.
* lep: az adott irányba lépteti a kigyót.
* lepes: a felhasználótól kapott jel alapján eldönti merre léptesse a kígyót, kezeli a képernyőfrissítéseket, és az extra funkciók egy részét.
* keptorl: Coss-Platform képernyőtörlésért felel.
* globális változókat beállító függvények (get…,set… függvények)

matrix.h,matrix.c

Tartalmazza a mátrix kezeléséhez használatos függvényeket:

* initMatrix : lefoglalja a 2. dimenzió memóriaterületeit, és beírja a mátrix minden elemébe az üres elemet(space).
* writeMatrix : kirajzolja a játék interfészét : a mátrixot, a falakat, és a pontszámkijelzőt.
* refreshMatrix: csak a megváltozott karaktereket frissíti a konzolon, ezt egy „szellem-mátrix” használatával éri el, ami az előző állapotot tartalmazza.
* clear…,write… függvények: a BULLET TIME információsávjának kezelésére szolgálnak.

program.c

* newGame : idő alapján inicializálja a véletlenszerűséget, meghívja a kígyókat létrehozó függvényeket, lerakja az étel-csomagokat, meghívja a mátrix-kirajzoló függvényt.
* main : lefoglalja a mátrix 1. dimenzióját, meghívja az initMatrix, valamint a newGame függvényeket, ezek után pedig a lepes-t, és ennek visszatérési értékét figyeli(Csak akkor lesz visszatérése, ha a játék valamiért véget ért)
* gethighscore: fájlból beolvassa, és egyben sorbarendezi a pontszámokat. meghívástól függően ki is írja azokat, visszaadja a 10. pontszámot, ha nincs annyi, akkor a legutolsót.
* getT: Teleportokat tartalmazó tömb címének átadására szolgál.
* jatek: Az új játék menüpont után meghívott függvény, inicializálja a mátrixot, a kígyókat, meghívja a kigyo.c-ben található lepes függvényt, majd ha a játék véget ért, lekezeli az eredményeket, ha a játékos bekerülne az első 10-be, bekéri a nevét is.
* loadSettings: beállítások beolvasása.
* saveSettings: beállítások mentése.
* setSavePath: beállítások helyének a bináris mappáját állítja be.
* A nem említett függvények, az adott menüpontok meghívásakor lefutó függvények.

Algoritmusok

initMatrix(m: mátrix)

Változók

i, j : egész;

Ciklus i:=0-tól Y-1-ig /\*Y konstans, a mátrix sorainak száma\*/

m[i]=X db karakter lefoglalása a memóriában /\*X konstans, a mátrix oszlopainak száma\*/

Ciklus j:=0-tól X-1-ig

m[i][j]=’ ’;

Ciklus vége

Ciklus vége

Függvény Vége.

writeMatrix(m: mátrix, k: kígyók tömbje)

Változók

i, j : egész;

Ciklus i:=-1-től Y-ig

Ciklus j:=-1-től X-ig

Ha i = -1 vagy i = Y akkor kiír(’-’); /\*Mátrix teteje\*/

különben ha j = -1 vagy j = X akkor kiír(’|’); /\*Mátrix oldala\*/

különben kiír(m[i][j]);

Elágazás vége.

Ha i=-1 és j= X kiír(„Pontszámok:”);

különben ha i=0 és j=X kiír(„1. játékos:”,k[0].pont);

különben ha i=1 és j=X kiír(„2. játékos:”,k[1].pont);

Elágazás vége.

Ciklus vége.

kiír(„\n”);

Ciklus vége

Függvény vége.

jatek()

Változók

mátrix m=Y db karaktermutató lefoglalása;

kigyok: 2 elemű kigyo típusú tömb;

egész ir;

egész res,i;

karakter c;

score sc;

Fájl f;

initMatrix(m);

newGame(m,kigyok);

res:=lepes(m,kigyok);

Kurzorpozíció(0,Y+3);

//A továbbiakban „Az ellenfél” kifejezés, a másik kígyó karakterét jelöli.

Ha res = 0 akkor kiír(„Sajnos rossz helyre léptél, a jéték véget ért. Az ellenfél nyert”);

kiir(„Pontozással … nyert”);

Elágazás vége;

Ha 4 <= res < 6 akkor

Ha többjátékos mód akkor

kiir(„Az öngyilkosság halálos bűn, nem tudtad? Az ellenfél nyert”);

Különben

kiir(„Meghaltál”);

Elágazás vége;

Különben ha res > 6

kiir(„Gyilkos! Megölted az ellenfelet! Az ellenfél nyert”);

Elágazás vége;

Ha bekerülne a pontszáma az első 10-be akkor

kiir(„Győztes neve?”);

fájlbaolvas(f);

Függvény vége