

# Programozói dokumentáció:

## A kígyó játék

Az én programom a kígyójáték volt, SDL megjelenítéssel.

A játékot elindítva 3 gombra lehet kattintani 1,2 játékos mód, illetve a dicsőséglistát lehet megnézni.

### A main.c:

Megrajzolja a menüt és kezeli az eseteket amikor a gombokra rányomnak, akkor hívja meg a játék-ot 1 vagy 2 játékos módban vagy a dicsőséglistát írja ki.

### A Jatek.c:

Ez kezeli a játékot 1 illetve 2 személyes módokban is amit a menüből lehet kiválasztani. A kígyókat az AWSD gombokkal, illetve a nyilakkal lehet irányítani. A kígyók pontjai számon vannak tartva a kijelzőn, illetve el is menti ezt egy txt-be a játék, ha a kígyó meghal és elegendő pontszámot elért, hogy a legjobb 10be belekerüljön.

Az **enum “Merre”** tartalmaz 4 értéket, ami a kígyó haladási irányát tárolja magába.

Az **“Iranyom” struktúra** tartalmazza a kígyóknak megadott irányt, illetve egy pointert a következő irányra így láncolt lista lesz.

A **“Kigyo” struktúra** az tartalmazza a kígyó egyes részeinek középpontját, sugarát, illetve a következő elemre egy pointert, így láncolt listát csináltam.

A **“Narancs” struktúra** tartalmazza a narancs középpontjának koordinátáit, illetve sugarát, ezt “eszi” meg a kígyó.

### Függvények:

#### **Uint32 idozit(Uint32 ms, void \*param)**

Ez a függvény hívodik meg az időzítő által. betesz a feldolgozandó események közé (push) egy felhasználói eseményt.

#### **Narancs narancsgeneral(Narancs narancs,Kigyo \*kigyofej,Kigyo \*kigyofej2,int palyateteje)**

Narancsnek generál egy x és y-t ami a pályán van és nem ütközik a kígyóval vagy a kígyókkal.

#### **void felszabadit\_kigyo(Kigyo \*kigyofej)**

A lefoglalt Kigyo típusú memóriaterületeket felszabadítja a program végén.

#### **bool jatek(int ABLAK,SDL\_Renderer \*renderer,bool ketjatekos)**

Ez maga a játékot teszi össze, megkapja, hogy 1 vagy 2 játékos módban indul, illetve az ablak méretet és a renderert, visszatér azzal, hogy vége van-e a játéknak.

## **A kigyomozgasa.c:**

### **Függvények:**

**void mozog(Kigyo \*kigyofej,Iranyom \*iranyeje,int ABLAK,int vx, int vy,int palyateteje)**

Kiszámolja a kígyó elejének a következő helyét és meg is változtatja.

**void leptet(Kigyo \*kigyofej,int pont)**

A kígyó részeit sorban lépteti, követve az első elemet.

**Kigyo \*kigyono(Kigyo \*kigyofej)**

A kígyónak növeli a hosszát, létrehoz egy új bogyót, amit a pályán kívül ad meg helyzetet mert a léptet függvénnyel úgyis sorba rakja

**void kirajzol(SDL\_Renderer \*renderer,Kigyo \*kigyofej,int szinr,int szing,int szinb,int szina)**

Kirajzolja a kígyót, megkapja a kígyó fejét és a színét.

**void torol(SDL\_Renderer \*renderer,Kigyo \*kigyofej)**

A pályán kívülre helyezi a kígyót.

**bool utkozik(Kigyo \*kigyofej1,Kigyo \*kigyofej2)**

Megkapja a 2 kígyófejét es ellenőrzi, hogy az első átadott kígyófeje ütközik e a második kígyóval.

## **A scorekiir.c:**

**A dicsőséglistával foglalkozik fájlba ir, olvas, kiirja azt SDL-ben.**

### **Függvények:**

**void beolvas(int \*tomb,int meret)**

Beolvasom egy 10 elemű tömbbé a fájlban tárolt dicsőséglistát.

**void beleir(int \*tomb,int meret,int pont)**

Kíírom fájlba a módosított tömböt, ami a dicsőség lista (abban az esetben más ez a tömb, ha a jelenlegi játék pontszáma belekerült).

**bool pontszamok(int ABLAK,SDL\_Renderer \*render)**

Kiírja SDL-ben a 10 legjobb pontszámot.

## **Az iránykezeles.c:**

### **Függvények:**

**void bevesz(Iranyom \*eleje,Merre ujirany)**

A FIFO kap egy új irányt.

**Iranyom \*kiad(Iranyom \*eleje)**

A FIFO kiadja az első irányt, amit épp tárol.

**void felszabadit\_irany(Iranyom \*eleje)**

A lefoglalt Iranyom típusú memóriaterületeket felszabadítja a program végén.