# Programozói dokumentáció

# Kigyosch – Biró Ferenc - HR4VCG

## Program célja:

A program a közismert kígyós játékot valósítja meg. A program indításkor egyből a játékba lép. Rendelkezik egy- és többjátékos móddal, illetve ranglistával. A kígyókat a játékosok a WASD gombokkal vagy a nyílbillentyűkkel irányítják. Ha egy kígyó a pálya szélébe (falba) vagy önmagába ütközik, vége a játéknak. Ha megeszi a pályán levő almát, akkor az véletlenszerűen újraterem, és nő a kígyó hossza, illetve a pontszáma is.

## Szükséges környezet:

A program az SDL multimédiás könyvtárat használja a grafikus megjelenítéshez és a billentyűzet, illetve az egér kezeléséhez.

Szükséges alkönyvtárak:

- SDL2_gfxPrimitives, SDL_image és SDL_ttf

A játék használ még továbbá betűtípusokat, képeket és egy szövegfájlt. Ezek a szükséges fájlok a játék mappájában megtalálhatóak.

## Játékmenet:

Ez a modul tartalmazza a játékmenethez szükséges függvényeket, ezek kezelik a kígyókat és az almákat.

Alkalmazott adatstruktúrák:

- Pont: (x, y) koordinátájú pontok, ezek a játéktér cellái
- A kígyó: szegmensei láncolt listában vannak, aminek a kígyó feje a kezdete, elemei Elem típusúak.
    - A koord mezője egy Pont típus, az adott szegmens koordinátáját tárolja.
    - Irany mezője az adott szegmens menetirányát tárolja, ennek a megjelenítésnél van szerepe.
    - A kov mezője a következő szegmensre mutat.
- Irany: Felsorolt típus, a kígyó haladási irányát jelenti. Lehetséges irányok:
    - Jobb (0)
    - Le (1)
    - Bal (2)
    - Fel (3)

- Utkozes: A kígyó lehetséges ütközéseit felsoroló típus, ezek lehetnek:
  - Semmivel (0)
  - Kigyoval (1)
  - Almaval (2)
  - Fallal (3)
- Alma: A pályán lévő alma koordinátáját tárolja
- Gyoztes: Felsorolt típus, a jelenlegi játék győztesét tartalmazza a következő játék kezdetéig, értékei lehetnek:
  - NemDoltEl (0)
  - Nyert1 (1)
  - Nyert2 (2)
  - Dontetlen (3)
- Jatek: olyan struktúra, amely tartalmazza az egy- és kétjátékos módhoz szükséges összes adatot, ezek:
  - Elem* fej[2] és Elem* fej[2], két tömb, melyek tartalmazzák a két kígyó fejére és végére mutató pointereket
  - Pontszam psz[2], tartalmazza a két játékos pontszámát
  - Alma alma, tárolja az alma helyzetét
  - int jatekosszam, a játékosok számát jelzi (egy vagy kettő lehet csak)
  - int jatekido, a “lépések” számát tárolja el, egy lépés az az időtartam, amire egyet “üt” az időzítő és lépnek a kígyók
  - Irany d[2] és Irany del[2], a kígyók jelenlegi és előző irányát tárolják el, az előzőt azért, hogy a játékos ne tudjon szembemenni önmagával és azonnal véget vetni a játéknak.
  - SDL_TimerID id, a használt időzítő
  - Gyoztes gyoztes, a győztest tárolja.

Függvények:

Ahol egy függvény bemenete a kígyó fej pointere, ott mindig cím szerint van átadva, hogy egységesek legyenek.

- Pont pontMozgat:
  - Bemenete egy Pont típus és Irány típus, visszatérési értéke megadott pont eltolva a megadott irányba.
- bool kigyoInic:
  - Inicializálja a kígyót, bemenetei a fej pointer cím szerint, a kezdő hossz int-ben és a kezdő koordináta Pont típusban. Lefoglalja a kígyó kezdő hosszának megfelelő memóriaterületet és beállítja a kezdőkoordinátáit. Ekkor a kígyó összes szegmense egy pontban van. Visszatérése igaz, hogyha sikeres volt.
- Elem *kigyoVegetKeres:
  - Megkeresi az adott lista végére mutató pointert, és azt adja vissza. Bemenete a cím szerint átadott fej pointer.

- bool kigyoNyujt:
  - Megnyújtja az adott kígyót, bemenete a kígyó végére mutató pointer (lásd előző függvény) és az utolsó szegmens koordinátája (ide “nyúlik be” a kígyó vége).
  - Visszatérési értéke igaz, ha sikeres volt.
- void kigyoFelszabadit:
  - Felszabadítja az adott kígyó foglalt memóriaterületét, bemenete a kígyó fej pointere cím szerint átadva.
- void kigyoLeptet:
  - Lépteti a kígyó fejét adott irányba, a többi szegmense láncban követi az előtte lévőt. Bemenetei a kígyó fej pointere cím szerint átadva, és a léptetés iránya.
- Utkozes kigyoUtkozes:
  - Megvizsgálja, hogy az első kígyó ütközött-e önmagával, a másik kígyóval, almával (megette azt) vagy semmivel sem, és visszatér az ütközés fajtájával. Bemenetei az első és a második kígyó fej pointere, és az alma pointere, mind cím szerint. Egyjátékos módban a második bemenet NULL, ekkor nincs vizsgálva. Kétjátékos módban mindkét kígyóra meg kell hívni, felcserélve a fej bemeneteket.
- void utkozesKezeles:
  - a fenti kigyoUtkozes fgv. segítségével feldolgozza az ütközést, és ennek megfelelően változtatja a játékállapotot. Bemenetei a jelenlegi játékállapot cím szerint, a játékstruktúra cím szerint, a grafikus struktúra cím szerint, és az SDL_Renderer.
  - (Allapot* a, Jatek* j, Graf* g, SDL_Renderer* renderer)
  - Egyjátékos módban, ha fallal vagy önmagával ütközik a kígyó, vége a játéknak,
- void almaUj:
  - Akkor kell meghívni, ha egy kígyó megette a bogyót. Addig sorsol új koordinátát neki, ameddig az lesz olyan helyen, hogy ne essen egyik kígyóba sem. Bemenetei a bogyó, és a két kígyó, mind cím szerint.
- Uint32 idozit:
  - Az SDL időzítőfüggvénye, meghívásakor növeli a jatekido változót eggyel.
- bool jatekInic:
  - Inicializálja a Jatek struktúrát a játékosszámnak megfelelő kezdőértékekkel, bemenetei a Jatek struktúra cím szerint, illetve a játékosszám egészben megadva.
  - Lenullázza a pontszámokat, seedeli a véletlenszámgenerátort, beállítja a kezdőirányokat, a kezdőpontokat, a kígyók fejeit lefoglalja és megkeresi a végüket, beállítja a játékosszámot a struktúrában, lerakja az almát a játékmódnak megfelelően, elindítja az időzítőt.
  - Visszatérési értéke igaz, ha sikeres volt.

- void billentyuzetKezel:
    - Kezeli a billentyűzetlenyomásokat, megváltoztatja a kígyó(k) irányát úgy, hogy azok nem mehetnek szembe az előző irányukkal. Bemenetei a Jatek struktúra cím szerint, illetve az játékloopban használt SDL_Event esemény cím szerint.
    - Egyjátékosmódban a WASD és a nyílbillentyűk lenyomásával, kétjátékosmódban az első játékos a WASD, a második játékos a nyílbillentyűkkel tudja irányítani a kígyóját.

# Grafika:

Ez a modul felelős a kígyók, a bogyók és a kezelőfelület grafikus megjelenítéséért.

Alkalmazott adatstruktúrák:

- Graf: a grafikához szükséges elemeket tartalmazza, mezői:
    - TTF_Font *font, *fontSzöveg: a ponttábla és a ranglista által használt betűtípusok.
    - SDL_Texture *sprite, *palya: a kígyók és a pálya textúrái
    - SDL_Color szovegszin: a ponttábla szövegszíne
    - Ponttabla ponttabla: lásd lentebb
    - Rekordtabla rekordtabla: lásd lentebb
- Ponttabla: a ponttábla megjelenítéséhez szükséges téglalapok, surfacek és textúrákat tartalmazza
    - SDL_Rect SzovegR1, PontR1, SzovegR2, PontR2: a két játékos pontjának téglalapjai, egyik a szöveg (“Játékos 1 pontszam: “), a másik a pontszám maga
    - SDL_Surface *sSzoveg1, *sPont1, *sSzoveg2, *sPont2: ugyanaz, mint az SDL_Recteknél.
    - SDL_Texture *tSzoveg1, *tPont1, *tSzoveg2, *tPont2: ugyanaz, mint az SDL_Recteknél.
- Rekordtabla: a rekordok megjelenítéséhez szükséges textúrákat tartalmazza tömbökben, a rekordszámnak megfelelő mérettel.
    - SDL_Texture* tHely[REKORDSZAM], *tPont[REKORDSZAM], *tNev[REKORDSZAM]:
    - Egy rekord helyezés, pontszám és a név mezőit tartalmazza textúraként.

Függvények, eljárások:

- void grafikaInic:
    - Inicializálja a grafikus megjelenítést, bemenete a Graf cím szerint, és az SDL_Renderer. Itt állítódik be a betűtípus, a betűszín, a szövegek helyei, nyílnak meg a betűtípusok, töltődnek be a pálya és a kígyók textúrái.
- void kigyoKirajzol:
    - Kirajzolja a kígyót, szegmensenként összerakva textúrából. Bemenetei a fej pointere cím szerint, a renderer, és a haladás iránya.

    - A függvény a nyelvre, a fejre, a testre és a farokra külön-külön rajzolja ki a megfelelő textúrát. Mivel a testrészeknek van menetiránya is, ezért a kirajzolásnál ezt figyelembe kell venni. Ha egy szegmens iránya megegyezik az őt megelőzővel akkor egyenest, ha nem akkor kanyarodót kell rajzolni, a megfelelő iránnyal.
    - Hogyha a testnek több szegmense egyben van, akkor oda nem rajzolja le a testet, csak a kígyó farkát. (ez az állapot kezdetben van csak)
- void almaKirajzol:
    - Kirajzolja az almát. Bemenete a alma cím szerint, a renderer és az almát tartalmazó textúra(sprite.png, ez tartalmazza a kígyót is).
- void pontszamUjraRender:
    - Újrarendereli textúrába a megfelelő játékos pontszámának értékét, hogy azt ki tudjuk rajzolni később. Bemenetei Graf cím szerint, a renderer, a Jatek struktúra cím szerint és a játékos sorszáma int-ben.
- void pontszamKirajzol:
    - Kirajzolja a korábban renderelt textúrákból a ponttáblát. Bemenetei Graf cím szerint, a renderer, és a játékosszám.
    - (egyjátékosmód esetén nem rajzolja ki a második játékos pontját)
- void rekordRenderel:
    - Kirendereli textúrába a rekordtábla elemeit.
    - Bemenetei a Rekord tömb, Graf cím szerint, és a renderer.
- void rekordKirajzol:
    - Kirajzolja a korábban renderelt textúrákból a rekordtáblát.
    - Bemenetei a Rekord tömb, Graf cím szerint, és a renderer.

# Játékvezérlés:

Ez a modul végzi a program vezérlését, ezalatt értve a menüket és különböző játékmódokat.

Alkalmazott adatstruktúrák:

- Allapot: a játék állapotát tároló felsorolt típus, értékei:
    - Fomenu (0)
    - Egyjatekos (1)
    - Ketjatekos (2)
    - Jatekvege (3)
    - Kilepes (4)
- Gomb: a menük gombjainak adatait tárolja, mezői:
    - SDL_Rect sGomb, dGomb: a gomb téglalapjai, első a forrástextúra téglalapja, a második a képernyőn a gomb helyzetének téglalapja
    - SDL_Texture *texture, *texturekiv: a gomb alap és kiválasztott textúrája
    - Allapot kov: az az állapot, ahova a gomb megynomásával jut a program

Függvények, eljárások:

- void gombFeldolgoz:
  - Kezeli az adott gombot, hogyha rávisszük az egeret akkor “kiválasztódik”, és más színnel jelenik meg a szövege, illetve, ha rákattintunk akkor a Gomb kov állapotába lép a játék.
  - Bemenetei a Gomb struktúra cím szerint, a jelenlegi Allapot cím szerint, az eseményhurok SDL_Event-je cím szerint, és a renderer.
- void fomenu:
  - Betölti a szükséges menütextúrákat, inicializálja a gombokat, majd eseményhurokban kirajzolja a menüt és kezeli a gombokat. Kilépéskor felszabadítja a használt textúrákat, átállítja az előző állapotot “Fomenu”-re. Bemenetei a jelenlegi és előző Allapot típusú állapotok (a és a_el) cím szerint, a Jatek struktúra cím szerint, és a renderer.
- void visszaszamol:
  - Minden játék elején meghívódik, megjeleníti a képernyőn a visszaszámlálót, és addig nem kezdődik a játék. Bemenetei a Jatek struktúra cím szerint, a renderer, a Graf struktúra cím szerint, és a céltéglalap, ahová megjelenítünk (az egész ablak).
- void egyjatekos:
  - Az egyjátékos módot bonyolítja le. Meghívja az inicializáló függvényeket, visszaszámol, majd elindul a játék. Játék közben az időzítő által generált eseményre kirajzolja a pályát, kezeli az ütközéseket, majd kirajzolja a kígyót és az almát, a játékos billentyűzetnyomására kezeli azt. A játék végével felszabadítja a kígyót, és az előző állapotot Egyjatekos-ra állítja.
  - Bemenetei a jelenlegi és az előző Allapot cím szerint, a Jatek és a Graf struktúra cím szerint, és a renderer.
- void ketjatekos:
  - A kétjátékos módot bonyolítja le. Meghívja az inicializáló függvényeket, visszaszámol, majd elindul a játék. Játék közben az időzítő által generált eseményre kirajzolja a pályát, kezeli az ütközéseket, majd kirajzolja a kígyókat és az almát, a játékosok billentyűzetnyomásaira kezeli azokat. A játék végével meghatározza a győztest, felszabadítja a kígyókat, és az előző állapotot Ketjatekos-ra állítja.
  - Bemenetei a jelenlegi és az előző Allapot cím szerint, a Jatek és a Graf struktúra cím szerint, és a renderer.
- void jatekvege:
  - A játék végével (vagy a menüből a ranglistát választva) ide jut a felhasználó, az előző játékállapottól függően mást jelenít meg:
  - Egyjátékos módból belépve: Eltávolítja az időzítőt, betölti és beállítja a gombokat, beolvassa fájlból a rekordokat, kirajzolja a képernyő elemeit. Ha a játékos pontszáma meghaladja a ranglistán szereplő pontszámok valamelyikét, akkor megjelenik egy szövegdoboz, ahová be kell írnia a felhasználónak a nevét, hogy felkerülhessen a listára és elmentse azt. Kilépéskor felszabadítja a használt textúrákat, és az előző állapotot Jatekvege-re állítja.

    - Kétjátékos módból belépve: Eltávolítja az időzítőt, betölti és beállítja a gombokat, kiírja a győztest, és kezeli a gombokat. Kilépéskor felszabadítja a használt textúrákat, és az előző állapotot Jatekvege-re állítja.
    - Főmenüből belépve: Betölti és beállítja a gombokat, beolvassa fájlból a rekordokat, kirajzolja a képernyő elemeit. Kilépéskor felszabadítja a használt textúrákat, és az előző állapotot Jatekvege-re állítja.
    - Bemenetei a jelenlegi és az előző Allapot cím szerint, a Jatek struktúra cím szerint, a renderer és a Graf struktúra szím szerint.

# Pontkezelés:

Ez a modul végzi a ranglista és a pontszámok, illetve a fájlok kezelését.

Alkalmazott adatstruktúrák:

- Pontszam:
    - egy játékos pontszámát tartalmazza intként, illetve ugyanezt egy char tömbben, ennek a megjelenítésnél van szerepe.
- Rekord:
    - Tartalmazza a fenti Pontszam struktúrát, és az azt elérő játékos nevét. Ezekből álló tömbben tároljuk a ranglistán szereplő pontokat és neveket.

Függvények, eljárások:

- void rekordBeolvas:
    - Beolvassa a toplista.txt fájl tartalmait a paraméterként átadott Rekord tömbbe. Ha a fájlban levő név túl hosszú (ami csak kézzel való átíráskor történhet) akkor leáll, és nem olvas tovább, végül bezárja a fájlt.
- void rekordKezel:
    - Megvizsgálja a már beolvasott rekordokat és az éppen elért pontszámot, ez alapján:
    - Ha az elért pontszám meghaladja valamelyik már toplistán lévőt, akkor a program szövegdobozban bekéri a játékos nevét. Ha már szerepel nagyobb ponttal a ranglistán, akkor nem történik semmi, hisz nem ért el új személyes rekordot. Ellenkező esetben frissül a lista, fentebb kerül a játékos (amennyiben fentebb kell kerülnie).
    - Bemenetei a Jatek struktúra cím szerint, a beolvasott Rekord tömb, a Graf struktúra cím szerint, és a renderer.
- void rekordMent:
    - Visszamenti a kezelt rekordokat a toplista.txt fájlba.
    - Bemenete a már kezelt rekordok tömbje.

- bool input_text:
    - Ez az InfoC-n megtalálható függvény, annyi módosítással, hogy a szövegdobozban a begépelt szöveg középre legyen zárva, a kurzor ehhez igazodva.
    - Bemenetei: Egy karaktertömb, ahova a beolvasott szöveg kerül, a szöveg maximális hossza, a téglalap, ahova kirajzolódik a szövegdoboz, a háttér és a szöveg színe, a használt betűtípus, és a renderer.
    - Visszatérési értéke igaz, hogyha enter miatt állt le a bevitel.

# A fő modul:

Ez a snakemain modul felelős a teljes program kezeléséért, ebben értve a menüket, és a játékmódok indítását.

Függvények:

- sdlInic:
    - Inicializálja az SDL-t, bemenetei az ablak szélessége és a magassága int-ben, és a leendő ablak és renderer cím szerint. Ha hiba lép fel, kilép a program az annak megfelelő hibaüzenettel.
- sdlKilep:
    - Felszabadít minden grafikus elemet és bezárja az SDL moduljait. Paraméterei a Graf struktúra cím szerint, a renderer és az ablak.

A main függvény beállítja a kezdőállapotokat, meghívja az SDL és a grafika inicializáló függvényt, belép a fő eseményhurokba, és meghívja a jelenlegi állapotnak megfelelő függvényt, alapértelmezetten a főmenüt. Kilépéskor meghívja az sdlKilep függvényt, és bezárul a program.