

Programozói dokumentáció

Név: Pribék Jonatán

Neptun-kód: P1PGS0

Tárgy: Programozás alapjai 1.

Játék neve: Space Invaders

Dátum: 2021. 11. 28.

A program

A program a Space Invaders nevű videójátékot tartalmazza. Ebben a játékos a képernyő alján lévő űrhajót tudja balra és jobbra mozgatni, és közben löni a fentről érkező űrlényekre. Az űrlények egyre növekvő számú sorokban jelennek meg, egyre gyorsabban haladnak lefelé, illetve a játék haladtával egyre több lövedéket is lőnek a játékos űrhajója felé. A képernyőn lévő összes űrlény elpusztítása esetén a játékos teljesítette az adott kört és a következőbe lép. A játékban végtelen hullámban jönnek az ellenségek, így a játéknak akkor van vége, ha az űrhajót háromszor eltalálja egy ellenséges lövedék, vagy ha az űrlények elérték a játéktér alját.

Modulok

A projekt 4 modulból áll, ezek a következők:

- Alapprogram (main.c),
- A játékot tartalmazó rész (jatek.c, jatek.h),
- A menüt tartalmazó rész (menu.c, menu.h),
- Az inicializálás (init.c, init.h).

A modulok közti kapcsolatot a menürendszer köti össze. A program elindításával az inicializálás után a felhasználó a menübe kerül, ahonnan elindíthatja a játékot. A játék befejeztével a felhasználó szintén a menübe kerül vissza.

Szükséges fájlok

A program sikeres és hibamentes elindításához szükségesek a következők:

- A kódot tartalmazó fájlok (main.c, jatek.c, jatek.h, menu.c, menu.h, init.c, init.h)
- A játékban lejátszott hangok (elet.ogg, ellov.ogg, loves.ogg, robbanas.ogg)
- A képernyőre kirajzolt képek (ellenseg.png, hajo.png, logo.png)
- A használt betűtípus fájlja (LiberationSerif-Regular.ttf)
- [A dicsőséglista szöveges fájlja (dicsoseglista.txt)]

Adatszerkezetek

6 különböző adatszerkezetet használ a program, amik a következők: hajó, játékos, csillag, dicsőséglista, menüállapot, kellékek.

A hajó:

```
typedef struct Hajo {  
    int x, y, vx, vy;  
    bool van, lott;  
} Hajo;
```

A hajó struktúrája tartalmazza a koordinátákat (x, y) és az ezekben való jelenlegi elmozdulás sebességét (vx, vy). Az ellenségek létének ellenőrzését a „van” bool adja meg, egy lövés leadásának lehetőségét pedig a „lott” bool határozza meg. A játékos úrhajója nem használja fel a „vy” és „van” értékeket.

A játékos:

```
typedef struct Jatekos {  
    int menupont, nehez, kor, pont, elet, ellensegszam;  
    bool loves, irany_bal, irany_jobb;  
    Hajo *hajo, *lov;  
} Jatekos;
```

A játékos struktúrája a legösszetettebb. Az adatszerkezetben az adott menüpont (menupont), a nehézségi szint (nehez), a jelenlegi kör (kor), az összegyűjtött pontszám (pont), a megmaradt életek száma (elet), és a hátralévő ellenség száma kerül nyílvántartásra (ellensegszam). Továbbá a lövés (loves) és a mozgás (irany_bal, irany_jobb) lehetőségére fenntartott bool-ok is ide tartoznak. A struktúrába beágyazva szerepel két Hajo struktúra, ezek a hajo és a lov, amik a hajóért és a lövésért felelősek.

A csillag:

```
typedef struct Csillag {  
    int x, y, vym, rg, ba;  
} Csillag;
```

A háttérben mozgó csillagokat 5 egész szám érték határozza meg. A jelenlegi pozíciójukat az „x” és az „y”, a sebességét és méretét a „vym”, illetve a színét és az áttetszőségét az „rg” és „ba” értékek adják meg.

A dicsőséglista:

```
typedef struct Dicsoseglista {  
    int pont;  
    char nev[31];  
    struct Dicsoseglista *kov;  
} Dicsoseglista;
```

Ez a struktúra egy egyirányú láncolt lista. A dicsőséglista „pont” egész szám értékébe kerül be a személy pontszáma. A ponthoz tartozik egy 30 karakteres nevet tároló „nev” karaktersorozat is.

A menüállapot:

```
typedef enum MenuAllapot {  
    fomenu,  
    nehezseg,  
    jatek,  
    dicsoseglista,  
    gameover,  
    kilepes  
} MenuAllapot;
```

A menüállapot adatszerkezet határozza meg az alapprogramban (main.c) kiválasztásra kerülő ciklust. Itt a menüpontok, a játék főciklus, és a kilépésre való érték különül el.

A kellékek:

```
typedef struct Kellekek {  
    Mix_Chunk *loveshang, *robbanashang, *ellovhang, *elethang;  
    SDL_Texture *hajokep, *ellensegkep, *logokep, *felirat_t;  
    TTF_Font *font;  
    SDL_Surface *felirat;  
} Kellekek;
```

Ebben a struktúrában jelennek meg a programhoz szükséges kiegészítő kellékek (hangok, képek), illetve az SDL grafikus környezethez szükséges elemek (felirat, font).

A környezet

A program az SDL2 platformfüggetlen multimédiás függvénykönyvtárat használja fel, amely egy egységes felületet biztosít a grafikus megjelenítéshez, hangok megszólaltatásához, billentyűk kezeléséhez. Ehhez a könyvtár 5 header fájlját vettem igénybe: SDL.h, SDL_image.h, SDL_mixer.h, SDL_ttf.h, SDL2_gfxPrimitives.h.

Ezek felelősek a program grafikus részének helyes működéséért, ezek a program elindításakor a fejlesztőkörnyezet (nekem CodeBlocks) mappájában kell szerepelniük.

A fontosabb függvények

- **void jatek_loop**(SDL_Renderer *renderer, MenuAllapot* állapot, Jatekos* jatekos, SDL_Event event, Csillag* cs, Hajo* ellenseg, Hajo* ellov, Kellekek* kellekek);

- A fő játékciklus függvénye, amely megkap minden szükséges struktúrát, képet és hangot. Ez frissít minden grafikus és eltárolt adatot.

- **void adat_reset**(SDL_Renderer *renderer, MenuAllapot* állapot, Csillag *cs, Jatekos* jatekos, Hajo* ellenseg, Hajo* ellov);

- Az adatokat alapértékekre állító függvény. Ez megkapja a játékban használt struktúrákat és a menü állapotát. A struktúrákban tárolt adatokat olyanra állítja, hogy az új játék indításánál megfelelő legyen. Ezek mellett a játékos a főmenübe tér vissza.

- **void hatter_rajzol**(SDL_Renderer *renderer, Csillag *cs, Jatekos* jatekos);

- Ez a függvény rajzol a paraméterként megkapott megjelenítő rendererre egy programablak hosszú fekete négyzetet, majd azután a paraméterként megkapott Csillag struktúra elemeit. Amennyiben egy csillag elérte az ablak alját, akkor az visszakerül az ablak tetejére, véletlenszerű értékekkel. Ha nem, akkor az y-sebességével kerül lejjebb. Továbbá ez a függvény végzi el az adott pont, kör és életek számának képernyőre való kiírását is.

■ **void hajo_mozgas**(SDL_Event event, Jatekos* jatekos);

- Ez a függvény felelős a játékos által irányított hajó mozgásáért. Ha a hajó az ablak szélein kívülre kerül, akkor azt az ablak szélére helyezi vissza. A hajó vízszintes sebessége alaphól 0, ha valamelyik iránygomb le van nyomva, akkor a hajó abba az irányba kap sebességet. Végül a hajó koordinátájához hozzáadódik a sebesség.

■ **void hajo_loves**(SDL_Renderer *renderer, Jatekos* jatekos);

- Ez a függvény bonyolítja le a játékos hajója által kiadott lövés logikáját. Ha a lövés a játéktér keretein belül van, akkor annak koordinátájához hozzáadódik a 30-ig ötösével gyorsuló sebességhez. Ellenkező esetben a lövés koordinátái kinullázódnak, és a játékos újra lövésre készen áll.

■ **void hajo_rajzol**(SDL_Renderer *renderer, SDL_Texture *kep, int x, int y, int meretx, int merety);

- Ez a függvény rajzolja ki a paraméterként megkapott, a játékos által irányított hajó textúráját a szintén megkapott x és y koordinátákra.

■ **void ellenseg_logika**(Hajo* ellov, Hajo* ellenseg, MenuAllapot* állapot, Jatekos* jatekos, Kellekek* kellekek);

- Ez a függvény bonyolítja le az ellenség cselekvéseit; ilyen a mozgás, lövés, találatra eltűnés.

■ **void kor_vege_check**(Hajo *ellenseg, Hajo *ellov, Jatekos* jatekos);

- Ez a függvény nézi meg, hogy a pályán minden úrhajó el lett-e pusztítva. Ha igen, akkor a kört eggyel megnöveli, megnöveli az ellenségek számát, és új értéket ad az ellenséges úrhajóknak.

■ **void dicsoseglista_beolvas**(FILE *fp, Dicsoseglista *dl);

- Ez a függvény végzi el a dicsőséglista beolvasását egy szöveges fájlból. Amennyiben ez a fájl nem létezik, úgy létrehozza azt, előre beállított nevekkkel és pontszámokkal. Ha létezik akkor onnan beolvassa az 5 pontszámot és a hozzá tartozó nevet is.

■ **bool input_text**(char *dest, size_t hossz, SDL_Rect teglalap, SDL_Color hatter, SDL_Color szoveg, SDL_Renderer *renderer, Kellekek* kellekek);

- Beolvas egy szöveget a billentyűzetről. Az első paraméter a tömb, ahova a beolvasott szöveg kerül. A második a maximális hossz, ami beolvasható. A visszatérési értéke logikai igaz, ha sikerült a beolvasás.