Maron Tibor

TETRIS - Programozói dokumentáció

A TETRIS játék az SDL grafikus felületével került megvalósításra.
SDL ablak felbontása: 250x525 pixel.

Adatszerkezetek

ranglista: az eredmenyek.h header fájlban került definiálásra az **Eredmeny** nevű struktúra, amelynek elemei egy 9+1 karakterből álló név, és egy egész típusú pontszám. Ez a struktúra alkalmas arra, hogy eltárolja egy adott játékos nevét és pontszámát. A main függvény elején a játék létrehoz ebből a struktúrából egy 10 elemű tömböt, amelyet ranglistának nevez el, és később ezt alkalmazza a TOP 10 eredmény tárolására.

Elemtipus: a jatek.h header fájlban került definiálásra az **Elemtipus** nevű struktúra, amely két egész típusból áll: az egyik az aktuális játékelem típusát, a másik pedig ugyanannak a játékelemnek a jelenlegi állapotát tárolja az elforgatott helyzetétől függően. A játékelem típusa egytől ötig egy egész szám: az egyes az I elem (sárga színű), a kettes az L elem (ciklámen), a hármas a kocka (világoskék), a négyes a T elem (piros) és az ötös pedig a Z elem (zöld).

A **main** függvény

A main függvény indításakor létrehozza az SDL futásához szükséges változókat, a pontszám és a menüpont egész típusú változókat, és a ranglista tömböt, amelyet rögtön nullára inicializál, majd meghívja az eredmenyek_beolvas függvényt, és az eredmenyek.txt fájlból feltölti a ranglistát. Ezt követően pedig létrehozza az SDL ablakot a megfelelő felbontással.

Ezután meghívásra kerül a menu függvény, majd a nyílbillentyűkkel lehet lépkedni a menüpontok között. Minden egyes lépés után újra meghívásra kerül a menu függvény. Az Enter megnyomására a kiválasztott menüponthoz tartozó függvények kerülnek meghívásra.

1: Játék: ezt választva meghívásra kerül a jatek függvény, amelynek visszatérési értéke belekerül a pontszám változóba. Ezután lefut az eredmenyek_modosit függvény, amely módosítja a ranglistát, ha a paraméterként kapott pontszám benne van a TOP 10-ben.

2: Eredmények: kirajzolja a ranglistát, ahonnan az Enter megnyomására térhetünk vissza a menübe.

3: Kilépés: a játék bezárásra kerül.

void menu (int menupont, SDL_Surface* screen);
A menu függvény feladata, hogy kirajzolja a menürendszert. Paraméterként csak 1-től 3-ig terjedő egész számot kaphat. Az éppen aktuális választás piros színű, a többi pedig fehér.

A ranglistához kapcsolódó függvények

Ezek a függvények az eredmenyek.h header fájlban vannak definiálva, és az eredmenyek.c fájlban deklarálva.

void eredmenyek_beolvas (FILE* er, Eredmeny* ranglista);
A main függvény elején fut le, feladata hogy a ranglista tömbbe beolvassa az eredmenyek.txt fájlban tárolt ranglistát. Paraméterei ezért a fájlra és a ranglistára mutató pointer.

void eredmenyek_megjelenit (Eredmeny* ranglista, SDL_Surface* screen);
Feladata, hogy az SDL ablakban megjelenítse a ranglistát: baloldalt a nevek, jobbra mellettük pedig a hozzájuk tartozó pontszámok szerepelnek. Az ENTER billentyű megnyomására térhetünk vissza belőle.

void eredmenyek_kiir (Eredmeny* ranglista);
Egyetlen paramétere a ranglista, amelyet az eredmenyek.txt fájlba exportál ugyanabban a formátumban, ahogyan az eredmenyek_beolvas függvény beolvasta onnan.

void eredmenyek_modosit (int pontszam, Eredmeny* ranglista, SDL_Surface* screen);
Ez a függvény közvetlenül a jatek függvény után hívódik meg. Feladata, hogy megvizsgálja, hogy a jatek függvény által visszaadott pontszam belekerül-e a ranglistába, és ha igen, akkor hová. Ha belekerül, akkor a ranglista többi elemét az adott pozíciótól kezdődően lejjebb tolja eggyel. Utána bekér egy nevet, melyet az Enter lenyomására véglegesít. A névben legfeljebb 9 karakterből állhat (a 10. a lezáró nulla), az elfogadott karakterek pedig az angol ábécé betűi. Ezt követően meghívja az eredmenyek_kiir, majd az eredmenyek_megjelenit függvényt és utána visszatér.

A játékmenethez kapcsolódó függvények

Ezek a függvények a jatek.h header fájlban vannak definiálva, és a jatek.c fájlban deklarálva.

int jatek (SDL_Surface* screen);
Ez a játék fő függvénye. Egyetlen paramétere a screen, amelyen a megjelenítés történik. Visszatérési értéke pedig egy egész, amely a játékos végleges pontszámát tartalmazza.

Meghívásakor létrehoz egy 2 dimenziós dinamikus tömböt, amely 20x10-es: ez lesz a játéktér. Továbbá inicializál egy időzítőt, amely a játékelemek zuhanásáért felelős.

Meghívja az ujelem majd a kirajzol függvényt, és elkezdődik a játék, amelynek alapja egy SDL eseményhurok, ami a nyílbillentyűkkel vezérelhető.

Nincs felhasználói interakció: a játékelem egy blokknyit zuhan.
Bal nyílbillentyű: az adott játékelem balra tolódik eggyel.
Jobb nyílbillentyű: az adott játékelem jobbra tolódik eggyel.
Le nyílbillentyű: az adott játékelem lerakásra kerül a legalsó szabad játékterületre.
Fel nyílbillentyű: az adott játékelem megforgatásra kerül.

Az ezekhez a funkciókhoz tartozó függvények a következő oldalakon kerülnek részletezésre.

Uint32 idozit (Uint32 ms, void* param);
Inicializálja a játékelemek zuhanásához szükséges időzítőt.

void kirajzol (SDL_Surface* screen, int jatekter[][10], int pontszam);
Ennek a függvénynek az a feladata, hogy kirajzolja a játékteret. Paraméterei a screen, a játéktérre mutató pointer és a játékos aktuális pontszáma. Minden egyes blokk 25x25 pixelt jelent a játéktéren, és 0-tól 10-ig terjedő értéket vehet fel a 2D-s tömbben. Nulla az üres blokkot jelenti, 1-től 5-ig a mozgatható blokkok vannak, míg 6-tól 10-ig a nem mozgatható párjuk. Ez a függvény minden esetben meghívásra kerül, amikor valamilyen változás történik a játéktéren.

bool mozdithato (int jatekter[][10]);
Logikai igaz vagy hamis értékkel visszatérő függvény, attól függően, hogy a játéktéren található-e mozdítható játékelem.

bool betelt (int jatekter[][10]);
Szintén Boolean értékkel visszatérő függvény, attól függően, hogy a játéktér kezdőrészén (egy 2x4-es téglalap) található-e nem mozdítható játékelem. Ha igen, akkor a játéktér beteltnek tekinthető: ekkor a játéktér felszabadításra kerül, meghívódik a jatekvege függvény, és a jatek függvény pedig visszatér a játékos pontszámával.

void ujelem (int jatekter[][10], Elemtipus* aktualis);
Paraméterként kapja a játékteret és az aktuális elemre mutató pointert, amelynek értéket ad. A véletlenszámgenerátor segítségével sorsol egy számot 1 és 5 között, amely szám megfelel az öt játékelem egyikének. Ezt követően esetszétválasztás segítségével hozzáadja az új játékelemet a játéktérhez, majd miután rögzítette a játékelem adatait az erre a célra szolgáló aktuális változóban, visszatér. Ez a függvény csak akkor kerül meghívásra, ha a játéktéren nincsen mozgatható játékelem.

void rogzit (int jatekter[][10]);
Feladata, hogy a játéktéren található mozgatható játékelemeket rögzítse úgy, hogy az értékükhöz hozzáad 5-öt. Ez a függvény akkor kerül meghívásra, hogyha mozdítható elem kerül a legalsó sorba, vagy pedig egy mozdítható elem alatt nem mozdítható található.

void zuhanas (int jatekter[][10]);
A játéktéren lévő mozdítható elem egy sort zuhan. Ez a függvény másodpercenként egyszer meghívódik, amennyiben nincsen felhasználói interakció.

void balratol (int jatekter[][10]);
void jobbratol (int jatekter[][10]);
Ez a két függvény nagyban hasonlít működésben, feladatuk, hogy a mozdítható játékelemet az adott irányba tolják egy blokknyival, amennyiben ez lehetséges. A jatek függvény eseményhurok részében kerülnek meghívásra a JOBB vagy BAL nyílbillentyű megnyomása után.

void lerak (int jatekter[][10]);
A játéktéren lévő mozdítható játékelemet rakja le a legalsó üres helyre. A jatek függvény eseményhurok részében kerül meghívásra a LE nyílbillentyű megnyomása után.

void soreltunes (int jatekter[][10], int* pontszam);
Megvizsgálja, hogy a játéktéren van-e betelt sor, és ha igen, megnöveli a pontszámot tízzel és meghívja a soreltunes_seged függvényt. Csak akkor kerül meghívásra, amennyiben a játéktéren nincsen mozdítható játékelem.

void soreltunes_seged (int jatekter[][10], int sor);
Paraméterként megkapja, hogy melyik sort kell eltüntetni, és az összes többi felette lévőt lejjebb rakja eggyel.

void forgatas (Elemtipus* aktualis, int jatekter[][10]);
Ez a függvény paraméterként megkapja az aktuális elemet, és az elem típusától függően hívja meg a hozzátartozó függvényt. A jatek függvény eseményhurok részében kerül meghívásra a FEL nyílbillentyű megnyomása után.

void forgat_I (int jatekter[][10], Elemtipus* aktualis);
void forgat_L (int jatekter[][10], Elemtipus* aktualis);
void forgat_T (int jatekter[][10], Elemtipus* aktualis);
void forgat_Z (int jatekter[][10], Elemtipus* aktualis);
Ezek a függvények nagyon hasonlóak, paraméterként az aktuális elemre mutató pointert kapják meg a forgatas függvénytől, amely esetszétválasztással dönti el, hogy a négy közül melyiket hívja meg (a kocka játékelem nem forgatható).

Működésük is nagyon hasonló, első lépésként megkeresik a játékelem bizonyos pontját, amelyből kiszámolható, hogy az adott elem elforgatható-e, vagy sem. Amelyben igen, akkor ugyanezzel a módszerrel valósítják meg a forgatást is. Végül pedig az aktuális játékelem állapotát átírják a következő állapotra.

void jatekvege (int pontszam, SDL_Surface* screen);
Ez a függvény akkor hívódik meg, amikor a játéktér betelt, azaz a játék véget ért. Kiírja a „A játéknak vége!” feliratot, és a játékos végleges pontszámát. Majd addig vár, amíg a játékos az ENTER billentyű megnyomásával vissza nem tér belőle.