

PacMan (NHF 4. Programozói dokumentáció)

Funkciók

A program sikeresen beolvassa szöveges fájlból a pályát. Ezután ki is rajzolja SDL segítségével a képernyőre, saját készítésű (Gimp-ben) fájlból beolvasott rajzokból, amik hasonlítanak az eredeti játék rajzaira.

PacMan mozgását lehet irányítani a “WASD” gombokkal, de PacMan itt nem tud megfordulni vagy falak irányába kanyarodni (úgy túl könnyű lenne a játék). A képernyőn megjelenő sárga pontokat rajtuk áthaladással lehet összeszedni, pontonként nő a bal felső sarokban kijelzett pontszám 100 ponttal, az összes pont összeszedéséért is jár extra 10000 pont és minél gyorsabban sikerült ezt megtenni annál több bónusz pont jár.

A szellemek az eredeti játékhöz hasonló módon, szellemenkénti különböző módszerrel próbálják elkapni PacMan-t. Ha PacMan egy helyen van egy szellemmel vagy összeszedte az összes pontot, akkor a játéknak vége és kiíródik a képernyőre a játékos végleges pontszáma. A megadott mezőbe be lehet írni egy tetszőleges (50 karakternél rövidebb) nevet, amit ezután kiír a játék az eddigi összes eredmény listájába. Ezután a képernyőn megjelenik a legjobb 10 játékos neve és pontszáma (ha a jelenlegi játék is ide tartozik akkor az is látszik). Másodpercenként 3-szor (minden 20-adik frameben) lejátszódik PacMan hangja.

Beolvasás

Pálya Beolvasása

A pályát a “Palya.txt” fájlból olvasom be. Ennek a szöveges fájlnak az első sora tartalmaz egy számot(dbSor változó), ami a sorok és oszlopok számát jelenti, mivel csak négyzetes pályákkal működik a játék (főleg az SDL-es megjelenítés miatt lett így). A következő “dbSor” sor, “dbSor” karaktert tartalmaz soronként, ebből áll össze a négyzet alakú pálya. A beadott fájlok között található 3 db példa pálya, emellett lehet készíteni sajátokat is szövegszerkesztő programokkal, arra figyelni kell, hogy ne legyenek a pályán zsákutcák, és a széle fal legyen.

Egy karakter lehet:

- ‘space’ azaz üres hely ahol lehet mozogni
- ‘#’ azaz fal ahol nem lehet mozogni
- ‘0’ ahol összegyűjtendő pont van és lehet mozogni
- ‘1’, ‘2’, ‘3’ vagy ‘4’ számok amik a 4 fajta szellem kezdő helyét jelzik
- P betű ahonnan Pac Man indul

A pályát a program egy négyzetes két dimenziós dinamikusán foglalt mátrixban tárolja el, aminek a neve “palya” és típusa egy “Mezo” nevű enum típus. A Beolvasas függvény először soronként beolvassa a fájlt, aztán karakterenként eldönti hogy hol mi van. Összeszámolja azt is hogy hány felvehető pont van összesen a pályán.

Felhasználónév Beolvasása

A felhasználónevet SDL-el olvassa be a program. Ehhez az InfoC oldalról az SDL segédletből a NagyHF-hez felhasználható “input_text” függvény enyhén módosított változatát használom. (forrás: <https://infoc.eet.bme.hu/sdl/#7>)

Karakterek mozgása

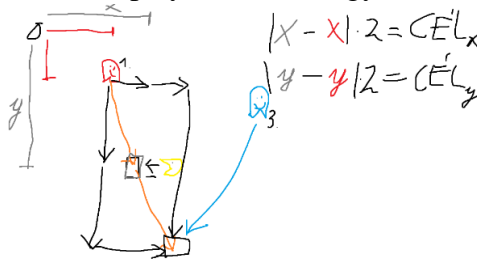
PacMan mozgása

PacMan irányát a WASD gombok lenyomásának érzékelésével állapítja meg a program, de ha ez pont szemben van a mostani iránnyal vagy a megadott irányba menve egyből fal következne, akkor csak tartja az eddigi irányt. PacMan minden 10-edik frameben mozog ténylegesen, a legutóbb megadott irányba, amíg az nem fal. (A játék időzítője 16.6666 ezredmásodpercenként vesz új frame-et ez nagyjából 60 frame per másodperc (60 fps))

Szellemek mozgása

A 4 szellem irányát a „SzellemDontesek” modul számolja ki. A 4 szellem különböző módon mozog, de az összes csak a kereszteződésekben tud megfordulni, nem fordulhat abba az irányba ahonnan jött és úgy fordul, hogy a céljához legkisebb távolságra levő kijárat négyzetére lépjen.

- Az első szellem célja Pac Man jelenlegi helye. Ezt a „FordulasNelkul” függvény számolja ki és ez fel van használva más szellemeknél is. A négy lehetséges irány által elért mezőkből megnézi milyen távol van PacMan, ha a mező fal vagy abból az irányból jött a szellem, akkor azt a pályában lehetséges értéknél nagyobb értékre állítja, majd minimum kiválasztással megkeresi a legjobb irányt.
- A második célja a Pac Man előtti negyedik négyzet (ha az nem pályán kívüli vagy fal, ebben az esetben közelebbi négyzetre amíg nem egy elérhető terület, legrosszabb esetben ő is a játékos helyét keresi). Ezt a „MasodikSzellem” függvény számolja ki, ami igazából csak az első szellem függvénye, azzal a kivétellel, hogy előtte megkeresi a PacMan előtti legtávolabbi olyan mezőt ami 4-nél közelebb van és nem fal.
- A harmadik szellem célja az első szellemből Pac Man előtti második négyzetbe húzott vektor kétszeresének helye (vagy ahhoz legközelebbi pályán belüli négyzet). Ezt a „HarmadikSzellem” függvény számolja ki, ahogy az ábrán is látható, az első szellem és PacMan előtti második mező koordinátáinak a különbségeinek abszolútértékeinek kétszeresei lesznek a keresett koordináták (vagy a pályán belüli ehhez legközelebbiek), amit az első szellem keresése számít ki.
- A negyedik szellem pedig ha 8 négyzethossznál közelebb van Pac Man-hez akkor a pálya bal alsó négyzetét, ha távolabb akkor Pac Man jelenlegi helyét célozza. Ezt a „NegyedikSzellem” függvény számolja ki, azután, hogy megnézi a szellem és PacMan távolságát eldönti, hogy hova kell keresnie az első szellem keresésével.



Pontozás

Ha PacMan átmegy egy felvehető ponton, akkor a 0-ról induló int típusú „Score” változóhoz hozzáad 100 pontot és a még pályán levő pontok számát jelző int típusú „dbPont” változóból levon 1-et. Ha a „dbPont” eléri a 0-t, akkor a „Score”-hoz hozzáad a program 10000 pontot és a 100000 ponrról induló int típusú „TimePoint” változó jelenlegi értékét is, feltéve ha az nagyobb, mint 0. A „TimePoint” minden harmadik frameben (azaz másodpercenként ~3szor) csökken 100 ponttal.

Megjelentés

A megjelenítést SDL-el oldja meg a program. Az ablak 1024x1024 pixelből áll, plusz egy felső sáv, ahol a jelenlegi pontszámot jelzi ki. Ezért az ablak méretéhez képest arányosítva rajzolja ki a pályát. A karaktereket az általam megrajzolt “PacmanTexturak.png” fájlból olvassa be a program. Mozgás után a volt pozíciót elfedi és az új pozícióra rajzolja a megadott karaktereket. PacMan esetében kirajzolásonként változik, hogy a nyitott vagy a zárt állapotban.

Eredménylista

Tárolás

A játékosok eredményeit a “LegjobbEredmenyek.txt” fájlba írja ki a program és onnan is olvassa be. Ez soronként pont szerint csökkenő sorrendben tartalmazza az eddigi játékosok nevét és pontszámát szóközzel elválasztva (az utolsó sort a -1 pontszámú sor jelzi). A program az eredményeket dinamikus memóriában foglalt duplán láncolt listában tárolja, amiben dinamikus foglalja a játékosok nevének helyét is. Felszabadtásra és új eredmény beszúrására (a csökkenő sort megtartva) külön függvények vannak.

Kiírás

Az eredményeket SDL-el írja ki a program a listán végigmenve a tizedik elemig a képernyőre. És az összes elemet kiírja a “LegjobbEredmenyek.txt” fájlba, megtartva annak a formai követelményeit.

Hangok lejátszása

Hangokat is SDL-el játszik le a program, ezeket hangfájlból nyitja meg. PacMan hangja saját készítésű és minden 20-adik frameben játssza be a program, az “SDL_mixer.h” könyvtár felhasználásával.

A modulok leírása

A program a “main”-en és a “debugmalloc”-on kívül 4 modult tartalmaz: “InputOutput.c”, “JatekLoop.c”, “SDLSzovegBeolvasas.c” és “Szellemdontesek.c”.

- Ezek közül az “SDLSzovegBeolvasas.c” az InfoC oldal segédletéből (forrás: <https://infoc.eet.bme.hu/sdl/#7>) származik, ez olvassa be SDL-ben a játékos nevét.
- Az “InputOutput.c” oldja meg a pálya beolvasását, az eredmények listájának beolvasását, kiírását és kezelését, emellett még az SDL-be szöveg kiírását is.
- A “JatekLoop.c” használja a “Szellemdontesek.c”-t is, és ez a modul felel a fő játékmenet lebonyolításáért. Ebben találhatóak a karakterek kirajzolását és eltüntetését végző függvények, az SDL időzítő és a fő SDL ciklus, ami az időzítő és a billentyűzet hatására futtatja a játékot.
- A “Szellemdontesek.c” a szellemek logikáját irányító függvényeket tartalmazza.

Szükséges Könyvtárak

A program működéséhez szükséges az SDL, ennek az “SDL_image.h”, “SDL2_gfxPrimitives.h”, ”SDL_mixer.h” és az “SDL_ttf.h” könyvtára. Emellett kell a “consola.ttf” betűtípus, a “Palya.txt”, a “PacmanTexturak.png”, a “LegjobbEredmenyek.txt”, a “zlib1.dll” és a “libpng16-16.dll” fájlok.

Források

A programhoz nagy segítséget jelentettek az InfoC óra vázlatai és SDL segédlete. A egyik modul majdnem teljesen innen származik (ahogy fentebb említettem). A játék algoritmusai és a pálya rajza az eredeti PacMan játék leírása alapján készültek, így jelentősen hasonlítanak.