

PacMan

Programozói dokumentáció

Készítette: Fehér Mátyás

Nagy házi feladatnak a klasszikus PacMan játékot választottam.

A feladatot grafikusan oldottam meg. A futtatás előtt a program működéséhez a Pygame multimédiás könyvtár letöltése szükséges.

A mappa a forrásfájlokön kívül tartalmazza a futáshoz szükséges egyéb dokumentumokat: a beolvasandó szöveges dokumentumokat és a grafikus megjelenítés által alkalmazott képeket. A szellemek és a Pacman megjelenítésére használt képeket a Paint3D segítségével készítettem.

A játék lényege, hogy a PacMan-t irányítva összeszedjük a pályán található étkeket, miközben elkerüljük a szellemekkel való ütközést. A PacMan-nek kezdetben három élete van, ha egy szellem megeszi, akkor az életek száma eggyel csökken. Ha minden élet elfogy, a játéknak vége. Ha minden étket összegyűjtöttük az pályán, akkor szintet lépünk, új pálya rajzolódik ki. A játék célja, hogy minél több pontot gyűjtsünk össze – az étkek megevése növeli a pontok számát.

A program szerkezete

A program indulásakor megjelenik a grafikus ablak. A játék szimulált beöltéssel indul, majd megjelenik a menü. Három lehetőség közül lehet választani: a játék elindítása, kilépés, valamint a toplista megtekintése. A játék elindítását követően kirajzolódik a pálya, a PacMan és kezdetben egy szellem. Összesen 4 szellem van, amelyek bizonyos időközönként megjelennek.

A program a játék futása alatt folyamatosan számlálja az elért pontokat. A pontok száma a pálya felett bal oldalt tekinthető meg a játék alatt, az életek aktuális száma pedig jobb oldalt. A játék végeztével a pontok elmentésre kerülnek, a felhasználó ehhez meg tudja adni a nevét. Ezután megjelenik a dicsőséglista az eddigi legjobb 10 játékos nevével és elért pontszámával.

Ezután ismételten megjelenik a betöltő képernyő – a felhasználónak lehetősége van új játék indítására.

A program tervezése során törekedtem az áttekinthetőségre, ennek érdekében a program több modulra van bontva. A modulokhoz általában egy osztály tartozik, valamint az osztály különböző függvényei

A program elkészítése során több helyen is használtam fájlkezelést. A pálya beolvasása fájlból történik, így a dokumentum módosításával akár más pályaszerkezet megadására is van lehetőség. A dicsőséglista is fájlkezelésen alapul. A játék végeztével az addig aktuális lista

beolvasásra kerül, majd az új eredményt figyelembe véve – ha szükséges – megtörténik a módosítás. Ezután a lista visszaírásra kerül az adott fájlba.

Forrásfájlok és főbb függvények

A program 7 modulból áll. A modulokra bontás a legtöbbször osztályok szerint történt, így egy osztályhoz egy modult használtam.

A program megírása során használtam öröklést, mivel korábban márt tanultam erről és így a szellemek megvalósítása egyszerűbb volt, mivel vannak közös tulajdonságaik.

Palya.py

A Pálya osztályt tartalmazó modul. Itt történik a pálya fájlból való beolvasása és a kirajzolása.

csomopont_be(self)

Beolvassa a pálya csomópontjait (az elágazások helyét) egy fájlból. A szellemek mindig egy megfelelő csomóponthoz mennek. Visszatérési értéke a csomópontok listája.

def palya_be(self)

Beolvassa a pályát egy fájlból. Értelmezi a fájlban szereplő karaktereket, amelyeket ennek megfelelően ételnek, üres útnak vagy falnak feleltet meg. Visszatérési értéke a pálya falainak listája.

palya_ki(self, p, ablak)

Kirajzolja a beolvasott pályát a képernyőre. Paraméterként átveszi a kirajzolandó pályát, valamint az ablakot, amire ki kell rajzolni.

Kaja.py

A pályán lévő étkekért felelős osztályt tartalmazó modul.

def kiskaja_be(self)

Beolvassa a sima étkeket abból a fájlból, amiből korábban a falak is be lettek olvasva. Visszatérési értéke az étkek listája.

def nagykaja_be(self)

Beolvassa a szuper étkeket abból a fájlból, amiből korábban a falak is be lettek olvasva. Visszatérési értéke a szuper étkek listája. Ezek azok az étkek, amelyeket a Pacman ha megesz, meg tudja enni a szellemeket.

def kaja_ki(self, kiskaja, nagykaja, ablak)

Kirajzolja a képernyőre a megfelelő pozíciókra a sima és a szuper étkeket is egyaránt. Paraméterként átveszi az ezeknek megfelelő listákat és az ablakot, ahova ki kell őket rajzolni.

def kkaja_eves(self, kiskaja, pacman)

A sima étkek megevését valósítja meg a függvény. Ha a Pacman egyet megesz, akkor a pontok száma kettővel nő. A megevett étkeket el kell tüntetni a pályáról. Paraméterként

átveszi a sima étkek listáját és a Pacmant, aki megeszi őket. Visszatérési értéke a megmaradt étkek listája.

def nkaja_eves(self, nagykaja, pacman, szellemek)

A szuper étkek megevését megvalósító függvény. Ha a Pacman megesz egyet, akkor egy ideig meg tudja enni a szellemeket. A megevett étkeket el kell tüntetni a pályáról. Paraméterként átveszi a szuper étkek listáját, a Pacmant, aki megeszi őket, valamint a pályán tartózkodó szellemeket. Visszatérési értéke a megmaradt étkek listája.

Pacman.py

A Pacman osztály definícióját és függvényeit tartalmazza. A Pacman-t a felhasználó irányítja a nyilakkal, ez az osztály tartalmazza a mozgás megvalósításának fő részét. A Pacman minden irányához más képet rendelt, hogy a játék élethűbb legyen. A pontok és az életek számlálása is itt történik.

def uj(self)

Egy új Pacman létrehozása. Erre akkor van szükség, amikor szintet lép a játékos, hogy a pontszámok változatlanok maradjanak.

def melyik(self)

Itt történik a Pacman aktuális irányához való kép kiválasztása.

def irányit(self)

Beolvassa a billentyűzetről az aktuális irányt, amerre a Pacmant mozgatni kell. Ha az irányításra való billentyűk egyike került beolvasásra, igaz értékkel tér vissza. Ha a felhasználó nem megfelelő billentyűt nyomott le, akkor hamissal.

def mozog(self, palya)

A függvény felelős a Pacman mozgatásáért. Átveszi a pályát, amin a Pacman mozog. Először megvizsgálja, hogy mozoghat-e az adott irányba a Pacman. Ha igen, akkor megvalósítja a mozgást. A mozgáshoz a lepes függvényt hívja meg megfelelően.

def mozoghat(self, irany, falak)

Megvizsgálja, hogy mozoghat-e az adott irányba a Pacman. Ehhez átveszi a megadott irányt és a falakat. Ha a Pacman falba ütközne, akkor nem mozoghat, ekkor a függvény hamis értékkel tér vissza. Különben igazzal.

def lepes(self, irany)

Egyel arrébb mozdhatja a Pacmant a megfelelő irányba.

def teleport(self)

A pálya jobb és bal oldalán egy-egy teleport van. A függvény célja, hogy ha a Pacman ide megy, akkor át tudjon menni rajta.

def pont_kiir(self)

Kiírja a képernyőre a pontok aktuális számát.

def elet_kiir(self)

Kiírja a képernyőre az életek aktuális számát.

Szellemek.py

A különböző szellem osztályok definícióját és függvényeit tartalmazza. Az ide tartozó osztályok: Kozos, Inky, Clyde, Pinky, Blinky. A négy különböző szellem a Kozos osztály leszármazottja az azonos tulajdonságok miatt. A szellemek maguktól mozognak, a Pacmant követik.

class Kozos

def megehet(self)

A Pacman ilyenkor meg tudja enni a szellemet. Ennek jelzésére a szellem színét a függvény kékre változtatja.

def nem_eheto(self)

A függvény meghívása után ismét a szellem tudja megenni a Pacmant, nem pedig fordítva. A szellem színe visszaváltozik pirosra.

def eheto_e(self)

A szellemek egy bizonyos ideig ehetők. Ez a függvény számolja az időt, ha letelik, akkor meghívja a nem_eheto(self) függvényt. Az idő utolsó egyharmadában a szellemek villognak – felváltva világos és sötétkékek.

def mozoghat(self, irany, falak))

Megvizsgálja, hogy mozoghat-e az adott irányba a szellem. Ehhez átveszi a megadott irányt és a falakat. Ha a szellem falba ütközne, akkor nem mozoghat, ekkor a függvény hamis értékkel tér vissza. Különben igazgal.

def lepes(self, irany)

Egyet arrébb mozdítja a szellemet a megfelelő irányba.

def lepesek(self)

A szellem Pacmantól való távolsága és elhelyezkedése alapján feltölt egy 4 elemű tömböt a 4 lehetséges iránnyal olyan sorrendben, amilyen sorrendben ezekre az irányokra szükség lesz a mozgás során.

def ut(self, pacman, sz_x, sz_y, b)

Kiszámolja a legrövidebb utat a Pacmanhez az A* algoritmus segítségével.

def mehet_e(self)

A szellemek a kezdő pozícióból bizonyos késleltetéssel indulnak el. Ez a függvény csökkentti a várakozási időt. Ha az nullára csökken, akkor igaz értékkel tér vissza, különben hamissal.

def teleport(self)

A pálya jobb és bal oldalán egy-egy teleport van. A függvény célja, hogy ha a szellem ide megy, akkor át tudjon menni rajta.

class Inky(Kozos), Clyde(Kozos), Pinky(Kozos), Blinky(Kozos)

def mozog(self, palya, csomopontok, pacman, b)

Mindegyik szellem kicsit máshogy valósítja meg a mozgást.

Menu.py

A menühöz tartozó függvények szerepelnek benne.

def forma(szoveg, szin, meret = 20, tipus = 'Arial Black')

Beállítja a megjelenítendő szövegek formátumát és színét.

def fajl_be()

Beolvassa az eddigi dicsőséglistát fájlból, majd visszaírja a fájlba az új dicsőséglistát, amiben már a mostani játkos neve és elért pontszáma is szerepel – ha elég pontot ért el hozzá.

def nev_be(jAblak, b)

Miután a játék véget ért, be kell olvasni a játékos nevét.

def animacio(jAblak, b)

A kezdő animáció a játék elindításakor.

def fomenu(jAblak, b)

A fő menü, ahol ki lehet választani a következő opciók egyikét: játék indítása, dicsőséglista megtekintése, kilépés.

Main.py

Magát a játék menetét megvalósító main függvényt tartalmazza ez a modul.

def main()

Magát a játék menetét valósítja meg ez a függvény. Megtörténik a pálya beolvasása, a Pacman és a szellemek létrehozása. Ha a Pacmannek minden élete elfogyott, a játék véget ér és megjelenik a dicsőséglista. Ha a Pacman megevett minden étket a pályán, akkor elindít egy új játékszintet.

Beallitasok.py

A játékok során szükséges beállításokat tartalmazó modul.

A_star.py

A feladat elkészítése során felhasznált A* algoritmust tartalmazó modul. Próbálkoztam egyéni implementációval, de sajnos nem sikerültek elég hatékonyra, így alkalmaztam egy interneten talált megoldást.

Forrásmegjelölés:

#A Star search code from <https://www.redblobgames.com/pathfinding/a-star/>

#Copyright 2014 Red Blob Games <redblobgames@gmail.com>

#Feel free to use this code in your own projects, including commercial projects

#License: Apache v2.0 <<http://www.apache.org/licenses/LICENSE-2.0.html>>