Ferenczi Csongor – LGRPYH   
PacMan Programozói kézikönyv

A játékom 3db c fileból és 2 headerből áll.

## Main.c:

int main(int argc, char \*argv[])

{

Inicializáció(Ablak, hang, időzítő, zene, ikon, fejléc) Surface értékek megadása és ezek megnyitásának ellenőrzése. GameState enum helye, itt dől el hogy a menübe, vagy a játékba lépünk be, illetve hogy a menü/játék mivel tért vissza, mi legyen a következő lépésünk. illetve Zene és kép felszabadítás, és egy return 0;

}

void nyertel(void)

{

Amennyiben a gamestate azzal tér vissza, hogy nyertünk, youwin képet nagyítja ki origo centrikusan 50 lépésben 20ms-es delay-el a képernyőre, illetve mögé tesz egy sötétebb hátteret.

}

void vesztettel(void)

{

Ugyan az mint a nyertel(), csak ez veszett értékre fut le, és YouLose képet nagyítja ki.

}

int highscore(void)

{

Miután lefutott a nyertél/vesztettél, megadhatjuk nevünket. A gép a-z,A-Z,0-9 értékeket fogad el, ékezetes betűket nem, és ezeket egy dinamikusan foglalt memóriában tárolja. Minden lenyomáskor az előző betűk+a jelenlegi betű kerül kirajzolásra. Backspace hatására az addigi névrészlet törlődik.

}

void highscorefajlba(char\*nev)

{

Betölti az aktuális highscore fájlt egy láncolt listába, és az aktuális értékünket belefűzi aszerint hogy melyiknél nagyobb/kisebb. Majd fáljba kerül kiiratásra a láncolt lista.

}

## Menu.c:

int menu\_kezdo(void)

{

Inicializálja a menüt, kirajzoltatja a háttérképet, az ikonokat, szövegeket, majd eventre vár, és a beérkező eventet szétválogatja. Kattintásra az egér koordinátái szerint dönt, melyik függvényt kell meghívja. Új játék és játék betöltés esetén beolvassa a fájlból a térképet.

}

void menu\_hatter\_kirajzol(void)

{

Kirajzol egy fekete képet, majd a hátteret, a címet, és a hang ikont.

}

int menu\_leiras(void)

{

Kiírja a játék leírását tartalmazo tömb tartalmát, illetve a pontoknál a képeket is. Mint minden menüpontban ebben is van eventre várás, melynek céla, hogy a hang változtatható legyen.

}

int menu\_eredmenyek(void)

{

Beolvassa fájlból a highscore-t és kiíratja azt a képernyőre. Szintén van event várás.

}

## Jatek.c:

int jatek\_indit(void)

{

a játék indító függvénye, megkeresi Blinky és Pacman helyzetét, ebből kiszámolja koordinátájukat. Amennyiben előzőleg menü volt, vissza teszi Blinky alatti eredeti értéket. Kezeli az időzítőket, és eventet vár. A játékban az event várásban benne van, hogy a hangot is át tudja állítani. Ha PeekEvent van, Blinky következő irányát számoltatja ki, illetve extra pontokat generáltat(lásd specifikáció) Userevent esetén törli pacman heylzetét, kiszámolja aktuális pálya[][] pozícióját (blinkynek is) majd megvizsgáltatja, hogy nem esik-e pálya szélébe vagy falba a dolog. Irányok szerint új koordot is számíttat. Megvizsgálja nyertünk/vesztettünk-e. bemenet szerint kezeli a cheatet, irányítást illetve a konami-t. Eventet vár.

}

int jatek\_menu(void)

{

ESC lenyomására jelenik meg, 4 menüponttal. Szokásos némító/halkító/kilépő evenetre vár. Kattintás esetén egér koord. alapján dönt milyen esemény fusson le. Kilépésnél meghívja a kilepes fv.-t mely ellenőrzi, mentettünk-e.

}

void kirajzol(void)

{

Végig pörgeti a pályát, és minden karakternek megfelelő képet rajzol az adott helyre( #fal, 1PacDot, 2PowerDot, 3Cherry,4Orange, 5Melon, 6Bell, 7Key)

}

int kilepes(void)

{

Megkérdezi, hogy biztos mentés nélkül akarunk-e kilépni, és ha nem akkor vissza dob minket a játékmenübe.

}

int almenu\_start(void

{

Módosított GameState enum, mellyel az almenü felstackelődésést lehet kiküszöbölni.

}

void palyaszele(void)

{

Megvizsgálja, hogy PacMan ill. a szellem pozíciója elérte-e a pálya elvi széleit(azért elvi, mert ha fal van akkor már nem tud kimenni addig) és ha igen, és az irány ugyan arra mutat, akkor áthelyezi az illetőt a pálya túloldalára.

}

void falvizsgalo(void)

{

Amennyiben a következő irány nem esik falra, átállítja a jelenlegi irányenumot illetve a sebességvektorokat. Amennyiben a jelenlegi irány következő lépése falra mutat( és a koordináta is egységre esik) megállítja pacmant.

}

void pontrendszer(int y,int x)

{

PacMan éppen elfoglalandó pozíciójának nézi meg a tartalmát. Amennyiben az pontot ér, növeli azt, és ki rendereli a pontszámot.

}

void kovkoordinataszamolo(void)

{

Kiszámolja a következő koordinátákat.( jelenlegi+sebességvektor)

}

void kiment(void)

{

Kimenti az egész pályát és a pontot egy save.sav fájlba. A fájlban a tartalom szövegként tárolódik( akár meg is tekinthető tcmd-vel)

}

void extrapontok(void)

{

A specifikációnak megfelelő esélyekkel generál egy extra pontot érő számot, és ezt beilleszti a pályába. a beillesztés csak üres vagy sima pont helyére történhet, illetve ha 10 próbálkozás után sem talál ilyen helyet, akkor a játék folyamatosságának megőrzése érdekében elvetésre kerül az aktuális extra pont.

}

void blinkykoord(void

{

Megegyezik a falvizsgáló fv.-nyel, csak ez Blinky irányát és sebességét változtatja.

}

Megjegyzés: Amennyiben a program írásának közel egy hónappal ezelőtti pillanatában lett volna fogalmam a generikus fv. hívás létezéséről, pár sorból be tudtam volna tenni mind a négy szellemet. Azért van csak egy szellem, mert nincs arcom mind a négyet csak nevében és pár betűben más fv.-ek favágós ctrl-c+ctrl-v módszerével betenni, a generikus átírásra pedig nem volt elég időm.