

Technical University of Košice



**Fakulta elektrotechniky
a informatiky**

**Problem Set 6
Terminal Pacman Game**

Ivan Honcharuk, 2023, 1 rok

Znenie zadania

Pomocou knižnice `ncurses` vytvorte ľubovoľný program (hru, prezentáciu alebo iný umelecký počin), pričom výsledný projekt musí spĺňať nasledujúce podmienky:

- Projekt musí obsahovať 2D svet.
- Dohromady musí projekt zahŕňať aspoň 3 z nasledujúcich výziev:
 - Práca s farbami
 - Ovládanie cez klávesnicu (bez nutnosti potvrdenia Entrom)
 - Viac úrovní (levelov)
 - Práca s časomierou resp. práca v čase (s časom sa program mení)
 - Práca s argumentami príkazového riadku
 - Práca so súbormi
- Projekt musí byť zložitejší ako ukážkové príklady a jeho úroveň musí byť dostatočná.

Riešenie

Hra Pakman

Hra Pacman sa odohráva v 2D svete. Úlohou hráča je ovládať Pacmana, aby zjedol všetky bodky v bludisku a vyhol sa stretnutiu s duchmi, ktorí hrdinu prenasledujú.

V mojej verzii hry je hracia mapa, umiestnenie na nej duchovia a pacman sú v samostatných súboroch: **"level_1.txt"**, **"level_2.txt"**, **"level_3.txt"** a **"level_4.txt"** (Celkovo sú v hre v súčasnosti 4 úrovne). Hlavný kód hry sa nachádza v súbore **"program.c"**.

Hra sa nachádza vo vnútri cyklu `while(replay)` - takže po prehre alebo výhre môže používateľ okamžite spustiť novú hru z ponuky.

Knižnica `ncurses` \ farby

Na podporu farieb bola použitá knižnica `ncurses`. Na samom začiatku programu sa nachádzajú základné parametre, ako napríklad

```
noecho();  
keypad(stdscr, true);
```

Potom nasleduje kontrola podpory farieb v termináli a ukončenie, ak nie je.

Kreslenie hlavného a bočných menu hry

Na vykreslenie hry sa inicializuje dvojica farieb a zavolá sa funkcia:

print_menu(game_map, &difficulty);

"game_map" a "difficulty" sa nastaví na: prvá mapa a normálna obtiažnosť.

Zobrazenie súboru mapy ("level_1.txt"):

```
1 weight=34, height=25, highest_score=0, 0x=16, 0y=18, Px=18, Py=10, Bx=14, By=12, Cx=16, Cy=12, Ix=18, Iy=12
2 #####
3 #.....##.....#
4 #.####.#####..##.#####.####..#
5 #.####.#####..##.#####.####..#
6 #.....#
7 #.####.##.#####..##.####..#
8 #.....##.....##.....#
9 #####.....#####.....#####
10 ____#...##_____##...#____
11 ____#...##_###-###_##...#____
12 #####...##_#_____#_##...#####
13 #_____#_____#_____#_____#
14 #####...##_#_____#_##...#####
15 ____#...##_#####_##...#____
16 ____#...##.....##...#____
17 #####...##..#####..##...#####
18 #.....##.....#
19 #.####.#####..##.#####.####..#
20 #.....#_____#.....#
21 ###..##..##.#####..##..##..###
22 #...##.....##.....#
23 #.#####...##...#####..#
24 #.#####...##...#####..#
25 #.....#
26 #####
```

V prvom riadku sú rozmery mapy a poloha duchov, ich súradnice X a Y. Nasledujúce riadky obsahujú hernú mapu v znakoch ASCII.

Spracovanie súborov

Po úvodnom nakreslení mapy sa inicializujú základné herné premenné a otvorí sa súbor s mapou, ktorý zodpovedá voľbe hráča alebo štandardnej(1).

```
// ! read info from file
FILE *fp;

int weight, height, Ox, Oy, Px, Py, Bx, By, Cx, Cy, Ix, Iy;
int lives = 3, score = 0, delta_time, time = get_start_time();

if (game_map == 1)
{
    fp = fopen( filename: "/root/GitLab/test-dir/ps6_the_game/level_1.txt", modes: "r");
    if(fp == NULL){
        printf( format: "Error opening file.\n");
        exit( status: EXIT_FAILURE);
    }
}
else if (game_map == 2)
{
    fp = fopen( filename: "/root/GitLab/test-dir/ps6_the_game/level_2.txt", modes: "r");
    if(fp == NULL){
        printf( format: "Error opening file.\n");
        exit( status: EXIT_FAILURE);
    }
}
else if (game_map == 3)
```

(a 2 ďalšie ify pre 2 ďalšie mapy).

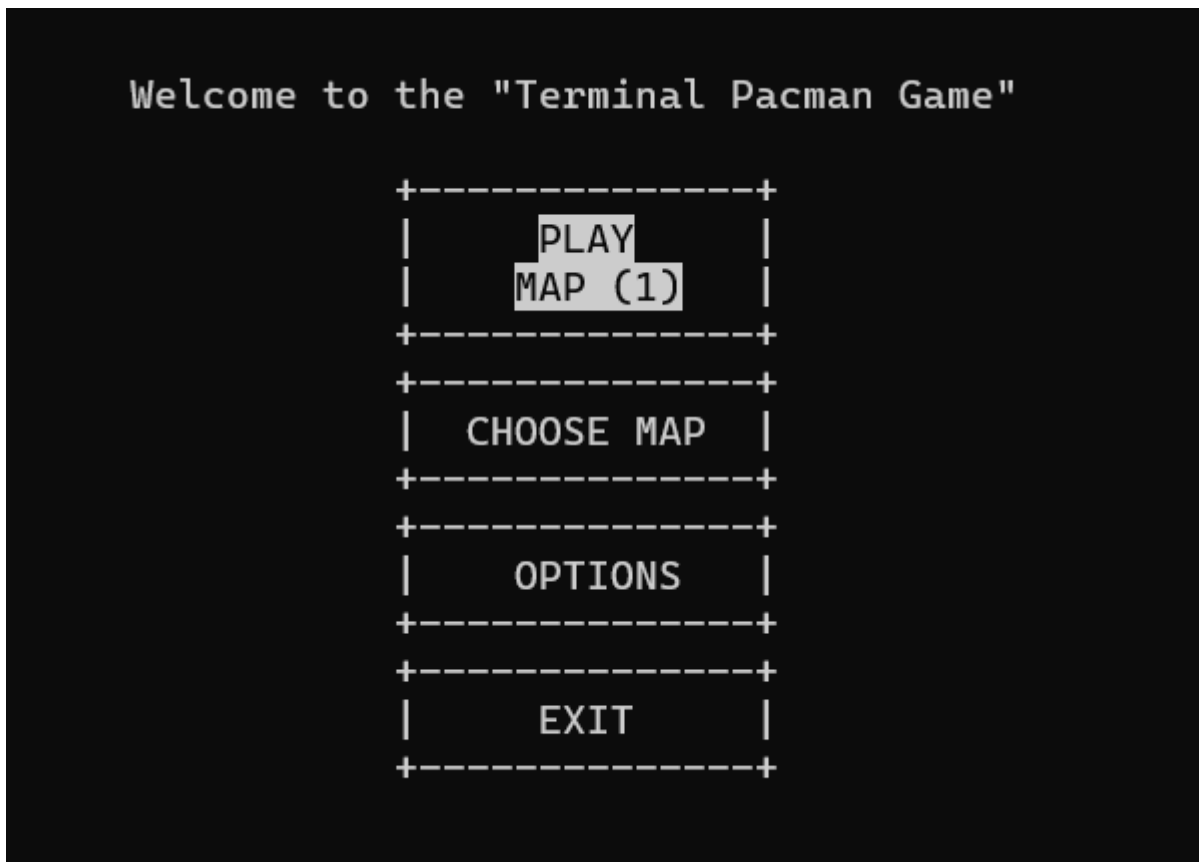
Po otvorení súboru mapy načíta informácie, ako je veľkosť mapy, pozície pacman a ghost, a zapíše ich do hlavných premenných. Potom prejde na vyplnenie matice - čo je reprezentácia mapy a umiestnenie hráča a duchov do tej istej matice na základe prijatých údajov.

Takto nemusíme na začiatku každej hry prehľadávať maticu a hľadať pozície hráčov a duchov, pretože ich už poznáme zo začiatku. A ich umiestnenie na mape po prenesení do matice nám umožní vyhnúť sa zbytočným problémom a chybám. Napríklad: 2 pacmani na mape.

Ďalej nasleduje inicializácia ďalších farebných dvojíc.

Main menu

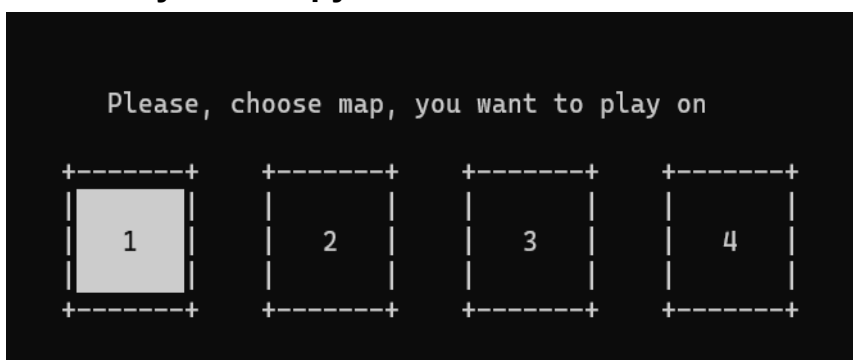
Po spustení spustiteľného súboru sa hráčovi zobrazí ponuka:



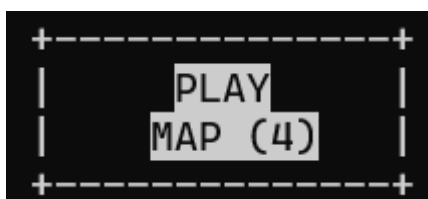
Tlačidlo: **PLAY MAP (1)** - umožní hráčovi začať hru na prvej karte, číslo karty sa zmení, keď sa vyberie ďalšia karta

Tlačidlo: **CHOOSE MAP** - umožní hráčovi vybrať si 1 zo 4 kariet, na ktorých bude hrať. Tento výber zmení aj zobrazenie tlačidla **PLAY MAP**

Ponuka výberu mapy:

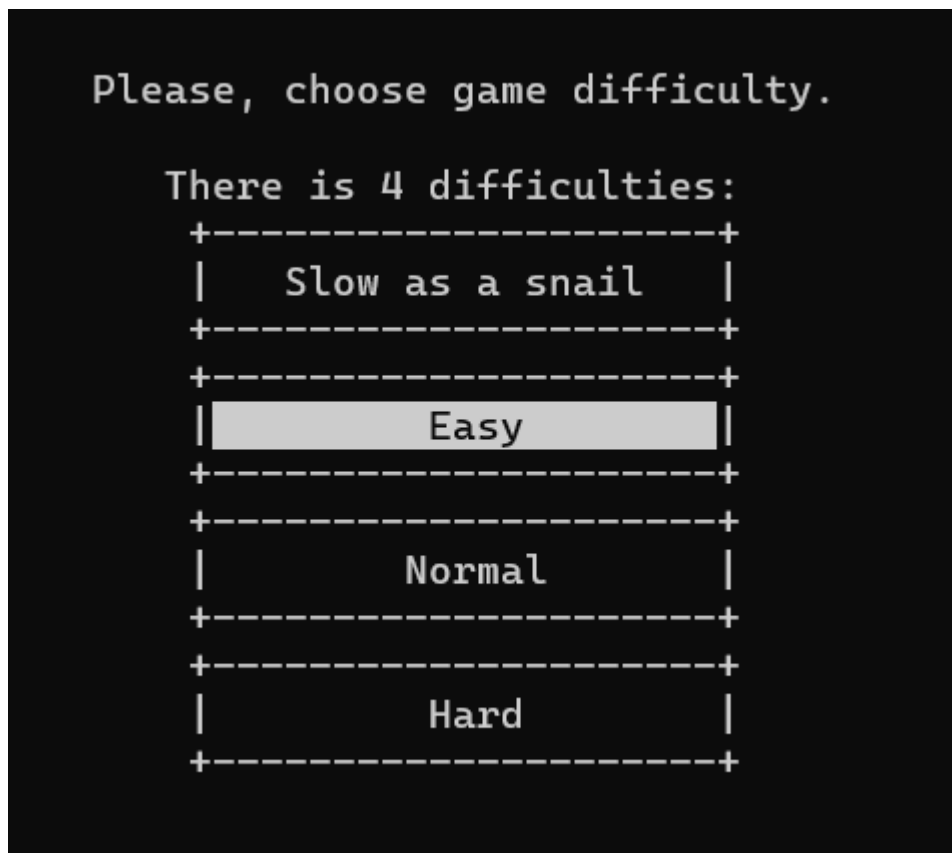


Zobrazenie nového tlačidla PLAY MAP:



Tlačidlo: **OPTIONS** - umožní hráčovi začať hru na prvej karte, číslo karty sa zmení, keď sa vyberie ďalšia karta

Ponuka výberu obtiažnosti:



Tlačidlo: **EXIT** - umožní hráčovi začať hru na prvej karte, číslo karty sa zmení, keď sa vyberie ďalšia karta

Tlačidlá v každej ponuke zmenia svoje "podsvietenie". Ak je tlačidlo stlačené - bude mať biele pozadie, ak nie - pozadie bude čierne a text bude biely.

Pri stlačení tlačidiel: **W** a **S**, v ponukách, kde sú tlačidlá usporiadané vertikálne, sa budeme môcť "pohybovať po tlačidlách".

Tlačidlá: **A** a **D** - nám umožnia pohybovať sa v ponukách, kde sú tlačidlá umiestnené vodorovne

Gameplay

Nakoniec sa dostávame k samotnej hre.

Najprv vytvoríme 2 premenné, ktoré budú uchovávať hodnoty času začiatku hry a aktuálneho času, pričom táto hodnota sa bude počas hry neustále aktualizovať.

- **Výstupné pole hry**

pomocou funkcie:

print_gamefield(height, weight, map, lives, score, delta_time);

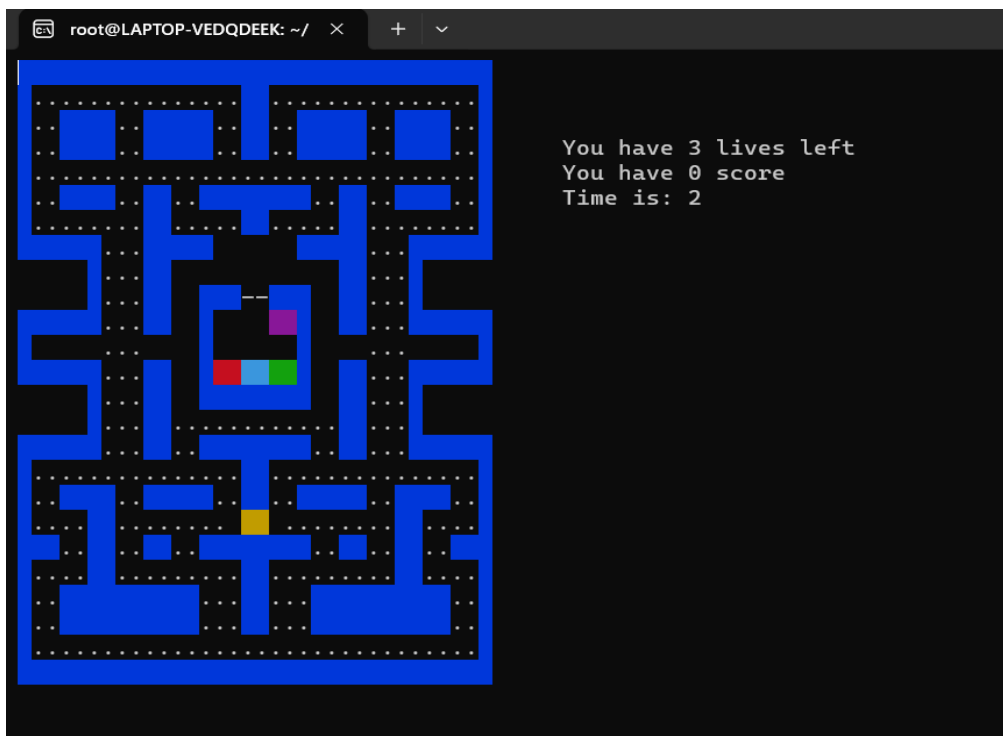
Ďalej potrebujeme niekoľko ďalších premenných:

```
int key_pressed;
char last_move = '0';
int tick = 0;
int clyde_moves = 0, inky_moves = 0, pinky_moves = 0, blinky_moves = 0;
char clyde_smer = 'N', inky_smer = 'N', pinky_smer = 'N', blinky_smer = 'N';
bool is_dead = false;
```

Potrebujeme ich na pohyb duchov a hráča.

Nastavte oneskorenie (difficulty) a v cykle do while spustíte hru:

Príklad hry na mape 1:



- **Pohyb hráča**

Vďaka funkcii `fflush(stdin)` získavame vstup od používateľa v `key_pressed`; a skontrolujeme stlačený kláves. Ak je stlačeným klávesom **AWSD**, posuňte packman správnym smerom, ak tam nie je stena. Zapamätajte si aj posledný

stlačený kláves v last_move. Ak sme počas nášho "ťahu" nazbierali bod, hráčovi sa pripočíta jeden k jeho skóre (score). Ak hráč nestlačil žiadnu klávesu alebo to nie je "**AWSD**" - pacman sa snaží pohybovať v smere poslednej stlačenej klávesy, kým nenarazí na prekážku.

- **Duchy**

Pri 30. posúvaní cyklu (pri 30. "tiku") sa na mape otvorí "brána", ktorá umožní duchom opustiť svoj "domov".

Pri 35, 70, 110 a 140 tikoch sa začnú používať algoritmy pohybu duchov.

To poskytne hráčovi určitý čas na zmenu polohy a získanie nejakých bodov. A spustenie každého algoritmu pohybu ducha zvýši úroveň obtiažnosti postupne v priebehu času, a nie niekoľkokrát naraz, ak by sa všetci duchovia stali aktívnymi naraz.

Príklad hry po niekoľkých minútach čakania:



- **Smrť alebo víťazstvo**

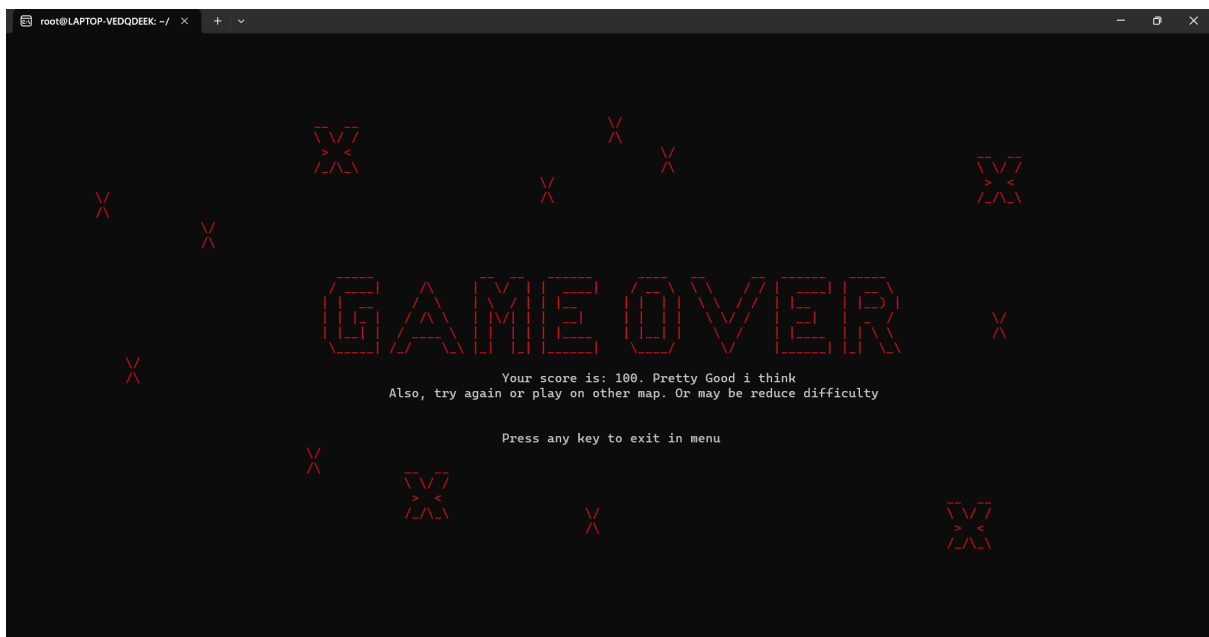
Počas hry na seba môžu naraziť hráči aj duchovia.

- V takom prípade:
 - hráč stratí jeden život
 - získané body sa vynulujú
 - mapa sa vráti do pôvodného stavu
 - duchovia a hráč sa vrátia na svoje pôvodné pozície.
- Ak hráč stratí svoj posledný život:
 - hra sa skončí
 - Na obrazovke sa zobrazí **"GAME OVER"**.
 - Hráč sa vráti do hlavnej ponuky hry, odkiaľ môže hru upraviť alebo začať novú.

Ak hráč vyhrá, zozbiera všetky bodky na mape:

hra sa tiež ukončí, ale s gratulačnou správou o víťazstve a tiež vráti hráča späť do menu.

Príklad víťazstva a porážky:





Záver

Na záver, výroba Pacman z 0 sa ukázalo byť ťažšie, než som očakával (Myslel som, že trochu ťažšie ako had)

Vzhľad hry

Kvôli nedostatočnému množstvu farieb, možnosti importovať vlastné textúry (hra je predsa určená pre konzolu) - nastali ťažkosti s predstavou vizualizácie niektorých prvkov hry.

Pohyb duchov

V pôvodnej hre Pacman má každý duch svoj vlastný algoritmus pohybu, ktorý sa navyše mení v závislosti od herného režimu. Napísal som jeden algoritmus pre všetkých a ten spočíva v náhodnom výbere. Každý duch si náhodne vyberá svoj vlastný smer. Z tohto dôvodu občas dochádza k malým nezrovnalostiam, ktoré podľa mňa vidím len ja ako vývojár.

Možné funkcie hry

Importovanie vlastných máp do hry - je to možné, ale budete musieť zmeniť kód a pridať otvorenie novej mapy

Samostatné vzory pohybu pre každého ducha - príliš veľa práce a príliš málo času, okrem toho nechcem úplne kopírovať pôvodné vzory a nemám dostatok fantázie na napísanie vlastných.