TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHIKY A INFORMATIKY

SPACE SHIFTER

2022 **Dmytro Maistruk**

Zadanie práce

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.

Návrh riešenia

Space Shifter – jednoduchá hra, v ktorej musíte na vesmírnej lodi nazbierať 10 mincí a zničiť nepriateľské pirátske lode.

Najprv som sa rozhodol otestovať argument obtiažnosti hry na príkazovom riadku. Ak tam nie je alebo neleží od 0 do 2, štandardne bola priradená hodnota 1. Ak je prítomná a leží v rámci od 0 do 2, zložitosť sa jej bude rovnať.

Ďalej vytvoril metódu **int space_shifter(int difficulty)**, v ktorej sa celá hra odohrávala. Metóda berie iba obtiažnosť hry a po dokončení hry vráti 0. Táto metóda držjí všetky údaje (napríklad o zdraví hráča), ktoré používajú iné metódy. Potom hra zmení obtiažnosť na zadanú a spustí knižnicu "curses.h" a nastaví veľkosť herného okna. Keď je všetko pripravené, algoritmus prejde do hlavného cyklu:

while(!game_is_over(current_difficulty, coins, killedEnemys, lifes, shots))

Táto slučka skontroluje, či sa hra skončila (hra sa skončí, keď hráč nazbiera 10 mincí a dokončí úroveň), vygeneruje herné pole, obrazovku so správou pre hráča a spustí ďalšiu slučku, ktorá skontroluje, či hráč dokončil úroveň:

while (!level_is_over(current_difficulty, coins, killedEnemys, lifes, shots))

V tomto cykle sa hracie pole načrtne, hráč a nepriateľ kráčajú. Nepriateľ sa pohne až po tom, čo hráč vystrelí (ak strieľa laserom, ťah neprejde na nepriateľa). Po tomto cykle sa program vráti do hlavného cyklu. Ak má hráč 0 zdravia, program zobrazí záverečnú obrazovku so skóre a štatistikou.

Nakoniec hlavný cyklus zvyšuje počet laserových výstrelov a úrovní o 1. Ak má hráč 10 mincí, program zobrazí záverečnú obrazovku so skóre a štatistikou. Hra počká, kým stlačíte ľubovoľnú klávesu, keď ju budete chcieť zavrieť.

Vytvorené metódy:

- **int generate_number(int min, int max)** – vygeneruje ľubovoľné číslo od minima po maximum.
- **int calculate_score(int current_difficulty, int coins, int killedEnemys, int lifes)** – jednoduchý výpočet bodov podľa vzorca (killedEnemys*20 + lifes*50 + coins*100)*(current_difficulty+1).
- **int space_shifter(int difficulty)** – hlavná metóda, v ktorej sa celá hra odohráva.
- **bool game_is_over(int current_difficulty, int coins, int killedEnemys, int lifes, int shots)** – ak je zdravie 0, zobrazí sa obrazovka porážky. kontroluje, či hra ukončená.
- **bool level_is_over(int current_difficulty, int coins, int killedEnemys, int lifes, int shots)** – kontroluje, či je úroveň ukončená.
- **int move_y(const int,int[2])** – Pohyb hráča po Y.
- **void loadMessageMenu(char center_message[],char lower_message[], int border_color, int text_color)** – zobrazí správu.

- **void loadBorderBox(int border_color,int text_color)** – zobrazí rámček.
- **void initialise_color_pairs()** – inicializuje farby.
- **bool find_symbol(char symbol)** – vyhľadá ľubovoľný znak na obrazovke.
- **void move_enemys(struct gamedifficulty *gd, int *lifes)** – hľadá nepriateľský symbol a presúva ho pomocou metódy send_enemy().
- **void send_enemy(const int position[], int side, int steps, char znak,int *lifes)** – Pomocou rekurzie presunie nepriateľa a skontroluje, či v hre prešiel na 0. Ak áno, odoberá to hráčovi zdravie.
- **void give_bonus(const char symbol, int *coins, int *killedEnemys, int *lifes)** – metoda "switch", pri ktorej je možné zvýšiť zdravie, mince a počet zabitých nepriateľov.
- **void generate_znak(int color,int size, char znak)** – vygeneruje zadaný znak na akomkoľvek prázdnom mieste.
- **void generate_field(struct gamedifficulty *gd)** – generuje hracie pole pomocou generate_znak().
- **struct gamedifficulty** – udržuje úroveň obtiažnosti hry.
- **void set_game_difficulty(struct gamedifficulty *gd,int coins, int enemys, int enemy_steps,int stoppers)** – zmení hodnotu *gd(úrovne obtiažnosti).
- **void change_difficulty(int current_difficulty,struct gamedifficulty *gd, char current_difficulty_name[])** - načíta hodnotu obtiažnosti a podľa toho zmení úroveň obtiažnosti na predpísanú. Tiež zapíše úroveň obtiažnosti do riadku.
- **void loadHUD(int level, char command[5],int lasers, int lifes, int coins)** – zobrazí rozhranie hry.
- **void get_user_move(int[2],char command[5],int *lasers, int *coins, int *killedEnemys, int *lifes, bool *playerTurn, int *shots)** – sleduje, ktoré klávesy hráč stlačil a zapisuje ich do príkazu. Ak stlačíte ovládacie tlačidlo, vykoná sa špecifikovaný postup.
- **void bullet(const int position[], int side, char znak,int *coins, int *killedEnemys, int *lifes)** – s pomocou rekurzie, pohybu strely v priamom smere. Keď sa stretne s objektom, zmizne a zavolá metódu "give_bonus".
- **void laser(const int position[], int side, char znak, int *coins, int *killedEnemys, int *lifes)** – s pomocou rekurzie, pohybu strely v priamom smere. Keď sa stretne s objektom, zavolá metódu "give_bonus", ale ne zmizne do konca stĺpcov.

Použitie programu

Program môžete skompilovať pomocou súboru makefile(priložený v ps6) alebo pomocou tohto príkazu:

gcc -std=c11 -Wall -Werror program.c -lm -lcurses -o program

program môžete spustiť takto:

./program

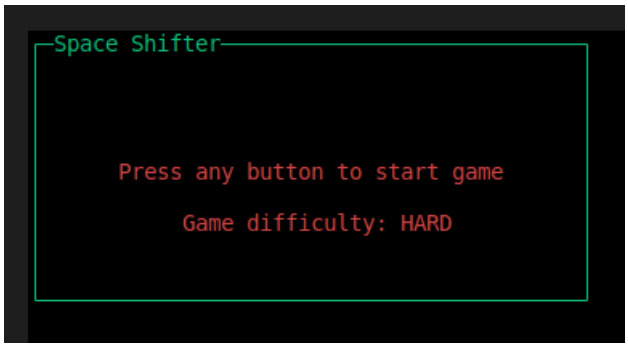
Alebo môžete použiť argument príkazového riadku od 0 do 2:

./program 0

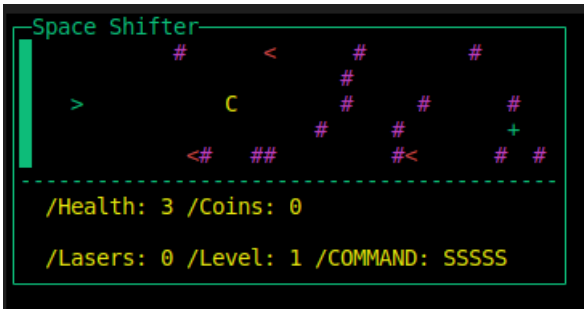
./program 1

./program 2

Tento argument ovplyvňuje úroveň obtiažnosti. 0 – najjednoduchšie a 2 – najťažšie. Úroveň obtiažnosti možno vidieť pri spustení programu:



Hra beží. Ovládajte svoju zelenú vesmírnu loď pomocou kláves „W“ a „S“ a strieľajte pomocou „SPACE“. Upozornenie, počas hry nepoužívajte šípky ani nemeňte veľkosť okna.



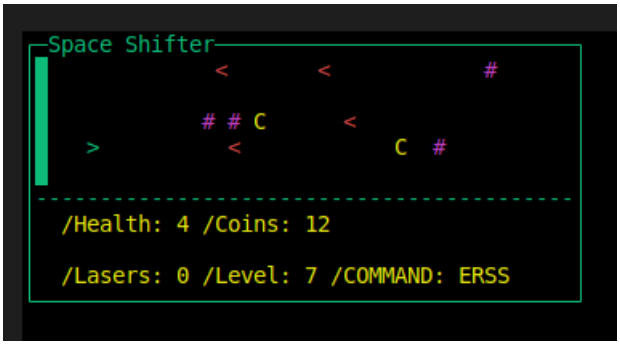
Príkazy môžete zadávať pomocou klávesnice:

EXIT – rýchle opustenie programu

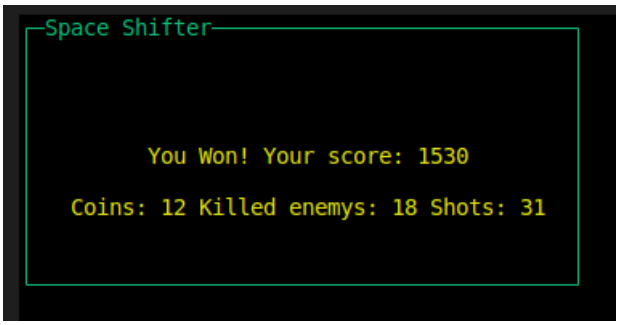
LAZER – vesmírna loď vypáli laser

DEV – označenie autora

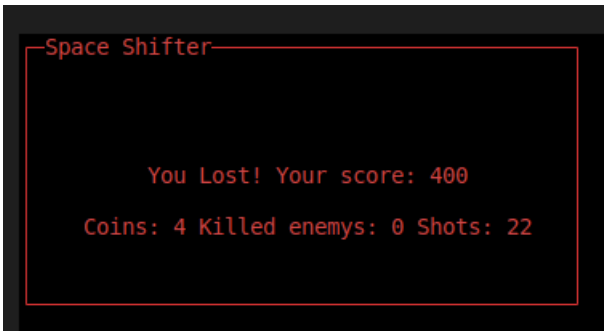
Upozornenie, keďže laser je 0, nebudete ho môcť vystreliť



Ak chcete vyhrať, musíte nazbierať 10 mincí a dokončiť úroveň.



Prehráte, ak váš zdravotný stav klesne na 0



Ak stlačíte ľubovoľnú klávesu na poslednej obrazovke, hra sa vypne.

Záver

Moja implementácia má niekoľko výhod. V prvom rade sa program nespúšťa podľa veľkosti okna terminálu, ale podľa zadanej veľkosti. To znamená, že keď program spustíte na akomkoľvek počítači, bude fungovať perfektne. Po druhé, program možno spustiť bez argumentov príkazového riadku.

Jednou z najväčších nedostatkov je moja implementácia príkazového riadku. Funguje to dobre, ale nie je oddelené na samostatnú metódu a môže sa zlomiť, keď zadáte šípky. Ďalšou nevýhodou je, že moje metódy sú viazané na konkrétnu veľkosť poľa.

Mám pár nápadov na zlepšenie mojej hry. Pridal by som nový typ zbrane, ktorá bude strieľať nelineárne a presunie zbraň do samostatnej konštrukcie. Vytvoril by som kvarkové asteroidy, ktoré by zmenili svoju pozíciu na ihrisku a zabránili hráčovi strieľať do nepriateľov. Možno by som vytvoril novú štruktúru, do ktorej by som umiestnil štatistiky hráča. Na úvodnú obrazovku alebo do príkazového riadku by som pridal pole na zadanie mena hráča a namiesto záverečnej obrazovky je možné urobiť tabuľku rekordov hráčov.