

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Problem Set 6: Curses

TECHNICKÁ UNIVERZITA V KOŠICIACH
Fakulta elektrotechniky a informatiky

Problem Set 6: Curses

Študijný program:	Informatika
Študijný odbor:	Informatika
Školiace pracovisko:	Katedra počítačov a informatiky
Vyučujúci	Bc. Lukáš Mikolaj

Znenie zadania

Pomocou knižnice *ncurses* vytvorte ľubovoľný program (hru, prezentáciu alebo iný umelecký počín), 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

Rozhodol som sa vytvoriť 2D hru *Tetris*. Hra predstavuje pravouhlé pole, v hornej časti ktorého sa objaví náhodná figúra a postupom času sa posúva nadol. Hráč môže figúrkou posúvať doľava, doprava a otočiť ju v smere hodinových ručičiek o 90 stupňov. Hra končí, keď figúra dosiahne vrchol. V mojej verzii hry chcem vytvoriť okno, v ktorom sa zobrazí ďalšia náhodná figúra a bude a hru môžem pozastaviť stlačením tlačidla. Tiež chcem vytvoriť menu s nastaveniami, kde môžem zmeniť farebnú tému okna, a po reštarte sa obnovia nastavenia z predchádzajúcej relácie.

Za vizuálnu časť hry bude zodpovedná knižnica <ncurses.h>.

Herné pole bude uložené v 2D poli s veľkosťou 38x19. Figúra je 2d pole s veľkosťou 3x3. Všetky tvary sú uložené v 3D poli s veľkosťou 7x3x3, z ktorého sa budú náhodne vyberať.

Hru som rozdelil do niekoľkých funkcií:

```
• void PrintField(WINDOW* gamewin, int field[38][19]);
• void RotateShape(int Shape[3][3]);
• int InsertNewShape(int field[38][19], int Shape[3][3], int* Y, int* X);
• int ChangeShape(int field[38][19], int Shape[3][3], const int Y, const int X);
• int ShiftDown(int field[38][19], int Shape[3][3], int* Y, const int X);
• int ShiftSide(int field[38][19], int Shape[3][3], const int Y, int* X, const int side);
• void game(int level);
```

PrintField - Funkcia vykreslí aktuálne herné pole do okna

RotateShape - Funkcia otočí figúru o 90 stupňov

InsertNewShape - Funkcia vkladá novu figúru do herného poľa

ChangeShape - Funkcia mení otočenu figúru v poli hry

ShiftDown - Funkcia posunie figúru o jeden riadok nadol

ShiftSide - Funkcia posúva figúru doľava/doprava, závisí od vstupného parametra

Game – Funkcia s hernou logikou a hernou obrazovkou

Použitie programu

Postup, ako preložiť a spustiť program:

- `gcc -std=c11 -Wall -Werror program.c -lm -lcurses -o program`
- `./program` + (voliteľný argument `'game'` ktorý okamžite spustí hru)

Ukážkové situácie v programe:

Keď spustíme hru v prvý krát, vytvore sa súbor **settings.txt** obsahujúci predvolené nastavenia hry. V predvolenom nastavení je uvítacia obrazovka zapnutá, ale môžeme ju vypnúť v nastaveniach.



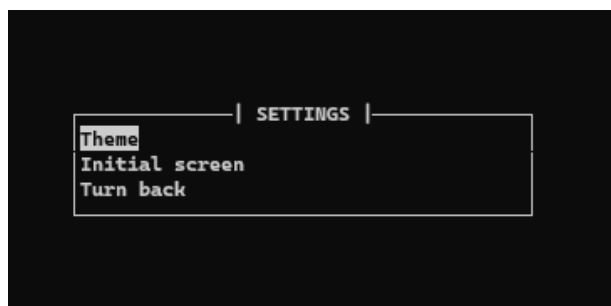
Obrázok 1 Uvítacia obrazovka

Po stlačení ľubovoľného tlačidla sa zobrazí hlavne menu, obsahujúce hru, nastavenia a výstup z hry.

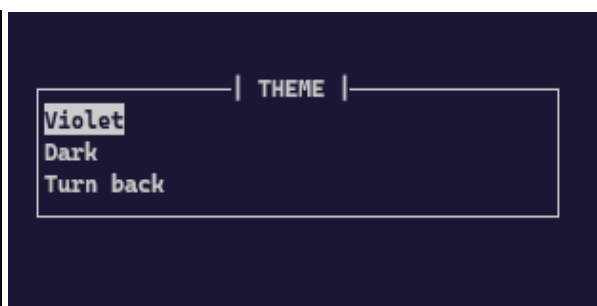


Obrázok 2 Hlavne menu

Obrázok 3 Nastavenia



Obrázok 4 Farebné prevedenie



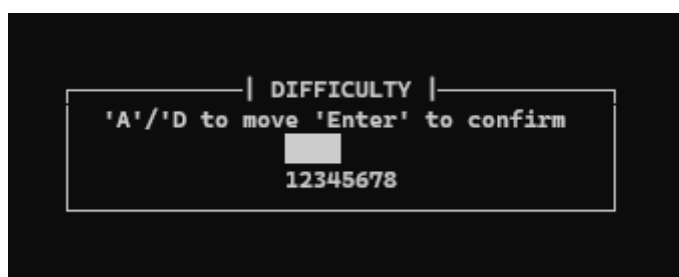
Obrázok 5 Výber uvítacej obrazovky



Obrázok 6 Farebné prevedenie



Obrázok 7 Výber náročnosti hry

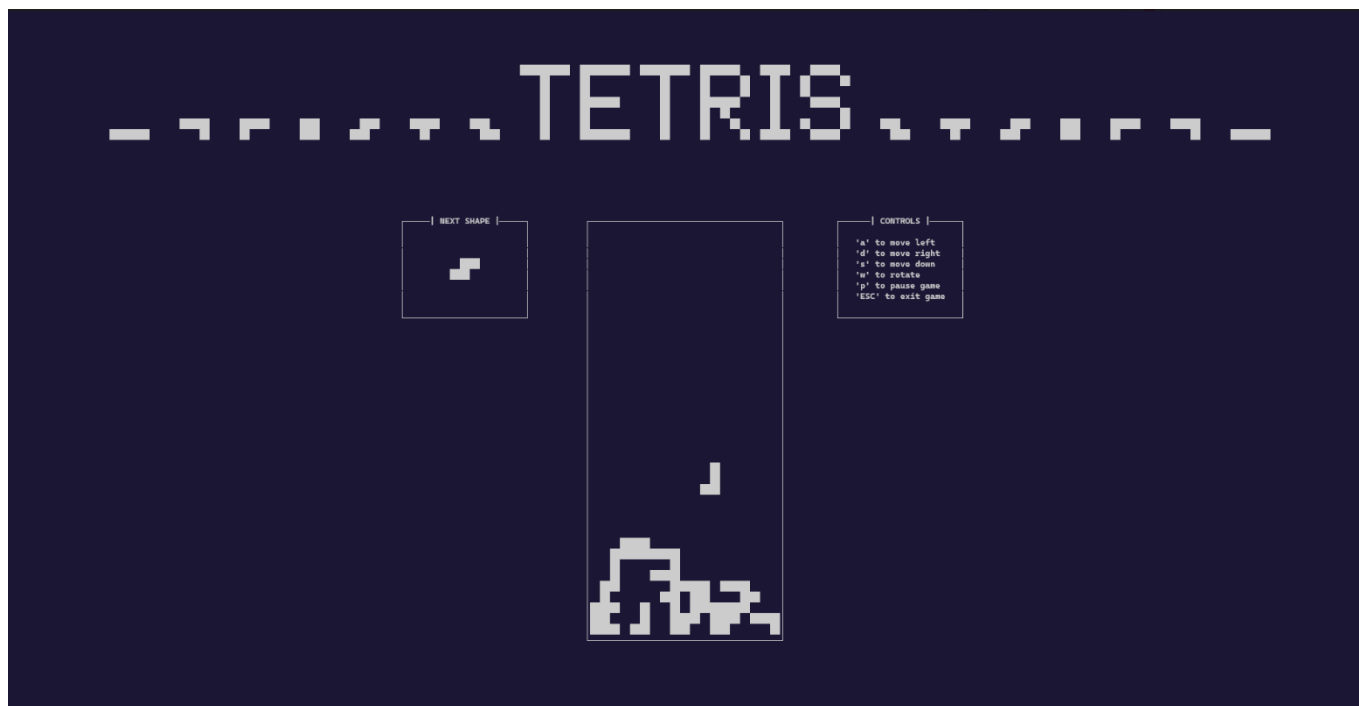


Parametre nastavení sa ukladajú do súboru "settings.txt" a prečíta sa pri spustení programu. Úroveň obtiažnosti ovplyvňuje rýchlosť hry, čím vyššia úroveň, tým vyššia rýchlosť.

Na ovládanie hry sa používajú nasledujúce klávesy:

- 'a' pre posunutie figúry doľava
- 'd' pre posunutie figúry doprava
- 's' pre posunutie figúry nadol
- 'w' pre pre otočenie figúry o 90 stupňov
- 'p' pre pauzu
- 'ESC' pre výstup z hry

Herná obrazovka obsahuje tetrisové pole, ďalšia figúra, sprievodcu ovládania:



Obrázok 8 Herná obrazovka

Hra sa skončí, keď figúra dosiahne vrchol.



Obrázok 9 Obrazovka ukončenia hry

Zaver

Po napísaní programu, zistil som, že je zlé rozlíšiť stĺpce poľa, keď sú figúre rovnakej farby. Ale na zafarbenie každého tvaru musím upraviť každú funkciu. Myslím, že to môžem urobiť 2 spôsobmi. Môžem vytvoriť nové 3D pole s veľkosťou [38][2][19] kde budú uložené nielen figúrky, ale aj farba tejto figúrky, alebo len zmeniť pole z figúrkami, a konkrétne ich obsah. Teraz sú 1 alebo 0, a budú 1,2,3,4,5,6,7 alebo 0, kde rôzne čísla sú zodpovedné za farbu. Takže takto funkcia PrintField bude vedieť, akú farbu má táto figúrka.