

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

SNAKE

2021

Martin Buzogaň

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

SNAKE

Študijný program: Informatika
Školiace
pracovisko: FEI

2021 Košice

Martin Buzogaň

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.

Čestné vyhlásenie

Vyhlasujem, že som celú prácu vypracoval/a samostatne s použitím uvedenej odbornej literatúry.

Košice, 05. decembra 2021

.....

vlastnoručný podpis

Návrh Riešenia

Cieľom tejto práce bolo zhotoviť hru Snake . Konceptom tejto hry je had ktorý sa postupom času rastie do dĺžky pričom sám je prekážkou pre hráča. Hráč sa pokúša zjesť predmety tak, že do nich narazí hlavou hada. Každá zjedená vec predĺži hada ,ale pridá hadovi isté výhody ako spomalenie alebo krátkodobé zrušenie okrajov obrazovky, takže vyhýbanie sa zrážke s hadom sa stáva postupne zložitejším. Hráč prehrá, keď had narazí na okraj obrazovky alebo sám do seba.

Ovládanie hry je jednoduché pohyb hada je nepretržitý a hráč udáva len jeho smer pomocou klávesových šípok, poprípade pomocou medzerníka alebo klávesov M hráč vie aktivovať power upy .

```
void read_score(int bfile[])
```

- Zabezpečuje načítanie posledného skóre a času zo súboru

```
void score_to_file(int lenght,int stime[])
```

- Uloží skóre a čas do súboru

```
void print_in_midd(int x,int y,char *text)
```

- Funkcia zabezpečuje stred vypísaného textu ktorá sa bude nachádzať na daných súradniciach

```
void erasing(int mxy[],int value)
```

- Funkcia je najmä využitá pri zobrazovaní hlavného menu
- Zabezpečuje že vykresľovanie je dynamické podľa rozmerov okna a v prípade zmeny veľkosti premaže celú plochu a text sa vykreslí znova

```
int welcmenu(int game_run,int mxy[])
```

- V tejto rozsiahlej funkcii sa vykresľuje celé hlavné menu pri tvorbe tohto menu sa využívali funkcie spomenuté vyššie
- Funkcia vráti hodnotu 0 alebo 1 na základe či sa hráč rozhodne hru spustiť alebo vypnúť

Funkcie patriace ku funkcii game:

```
void gen_x_y(int xy[],int mxy[])
```

- Táto funkcia zabezpečuje generovanie náhodnej x a y súradnice ktoré sú neskôr využité pri zobrazovaní „jedla“ pre hada alebo samotné power upy ktoré môže hráč počas hry kedykoľvek využívať

```
score_power_up(int lenght,int mxy[],int colected,int col_power,int power,int stime[])
```

- Funkcia ktorá zabezpečuje vykresľovanie dolnej lišty kde je vykreslená časomiera , dĺžka hada , a či sú okraje obrazovky aktívne alebo počet

spomaľovacích výhod , a ako posledné zobrazuje počet vypínaní okrajov obrazovky ktoré má hráč k dispozícii

```
int checking(int head[],int pos_x_y[],int mxy[],char char_look,int color,float snake_time)
```

- Funkcia checking zabezpečuje kontrolu či had prešiel cez daný znak
- Ak prešiel „zjedol“ daný znak tak funkcia zapíše rýchlosť hada premaže miesto kde bol znak vygenerovaný a potom vygeneruje súradnice nového znaku
- Ak hráč neprejde znakom znak sa na daných súradniciach vypíše v danej farbe ktorá prislúcha jednotlivému znaku
- Funkcia vráti hodnotu 1 ak had prešiel znakom , v inom prípade vráti hodnotu 0

```
int game(int game_run,int mxy[])
```

- Vo funkcií game sa využívajú všetky vyššie uvedené funkcie
- V tejto funkcii prebieha vykresľovanie celého herného poľa , dynamické vykresľovanie hada od smeru pohybu , alebo samotné počítanie času odkedy bola hra spustená
- Aby čas korešpondoval so spomaľovaním alebo so zrýchľovaním času je potrebné počítať tiky programu (1 sekunda / hodnota nanosleep) , nanosleep zabezpečuje malé pozastavenie programu čo opticky vyzerá že sa had pohybuje po hracom poli tato hodnota nanosleep sa postupne každých 30 sekúnd zmenšuje čo ma za následok menšie spomalenie programu a vizuálne to vyzerá že sa had pohybuje rýchlejšie
- Pomocou switch vetvenia je naprogramovaný pohyb alebo dynamické otáčanie hlavy hada

Funkcie využite z prídavných knižníc :

```
box(stdscr,0,0);
```

- Zabezpečuje vykreslenie okrajov na šírku a výšku okna s preddefinovanými znakmi

```
border(' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ');
```

- Zabezpečuje vykreslenie okrajov na šírku a výšku okna pomocou zvolených znakov

```
nodelay(stdscr,bool hodnota);
```

- spôsobí, že funkcia `getch()` bude volaná bez blokovania (hneď)
- ak nie je k dispozícii žiaden vstup, `getch()` vráti **ERR**
- ak je táto funkcionálna blokována (bŕ je **FALSE**), `getch()` čaká, kým nie je stlačená klávesa

```
curs_set(int visibility);
```

- nastavuje viditeľnosť kurzora na základe hodnoty parametra (**0** - neviditeľný, **1** - normálny a **2** - veľmi viditeľný)

```
endwin();
```

- ukončuje prácu v režime knižnice *ncurses* a obnovuje pôvodné nastavenia terminálu

```
keypad(stdscr, bool bf);
```

- ak je `bf` **TRUE**, potom používateľ môže stlačiť funkčné klávesy (napr. šípky) a funkcia `wgetch()` vráti jednu hodnotu reprezentujúcu klávesu, napr. **KEY_LEFT**

```
noecho();
```

- zabezpečí, aby sa znaky po stlačení a načítaní funkciou `getch()` nevypisovali na obrazovku

```
cbreak();
```

- zabráňuje bufrovaniu
- znefunkčňuje mazanie/zrušenie procesu (cez Ctrl-C), takže znaky sú okamžite k dispozícii programu

```
nanosleep(&ts, NULL);
```

- spomalí program v nanosekundách ktoré sú určené v štruktúrach

```
start_color();
```

- inicializácia farieb

```
init_pair(1, COLOR_WHITE, COLOR_BLACK);
```

- definovanie farieb pod určitú premennú

```
attron(COLOR_PAIR(1));
```

- volanie samotnej farby

```
wattron(stdscr, A_REVERSE);
```

- zapnutie inverzie zobrazovania

```
wattroff(stdscr, A_REVERSE);
```

- vypína inverziu zobrazovania

Použitie programu

Preklad programu prebieha prostredníctvom kompilátora gcc s použitím tohto príkazu :
`gcc program.c -o program -std=11 -Werror -Wall -lcurses`

kde :
- program.c je názov súboru s príponou .c
- -o názov súboru ktorý bude skompilovaný
- -std=11 verzia jazyka c v ktorom to ma byť preložené
- -Werror vypisuje error
- -Wall vypíše nevyužité premenné
- -lcurses pri kompilácii je nutné zahrnúť využitú knižnicu

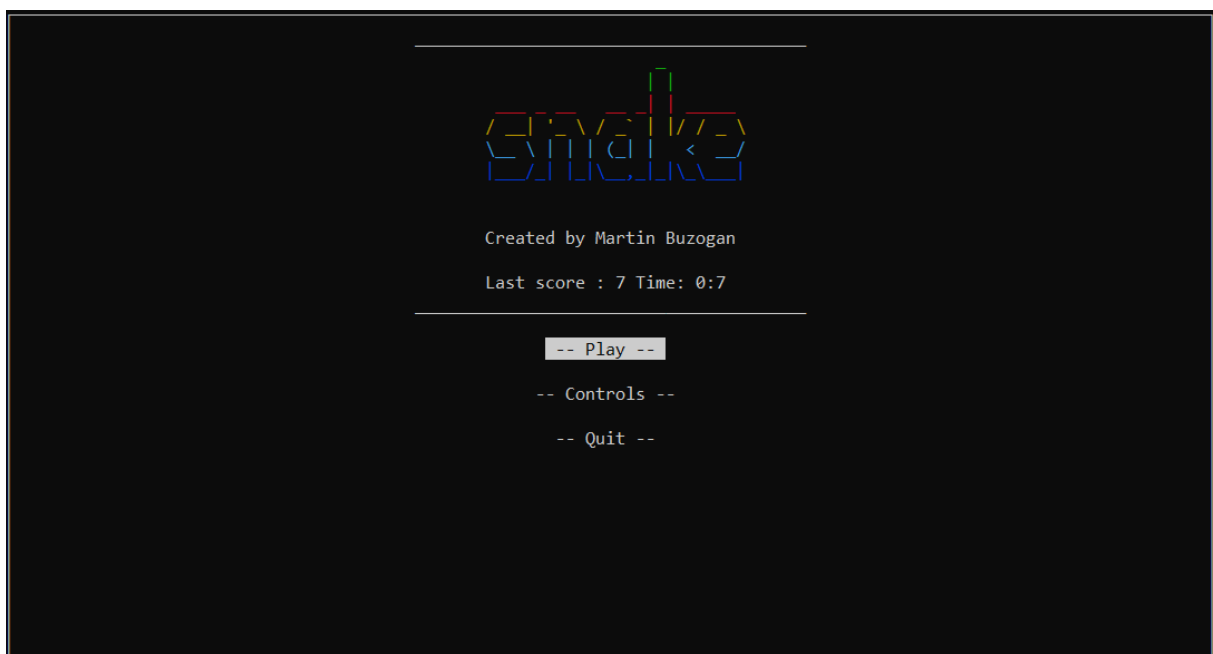
```
gcc program.c -o program -std=c11 -Werror -Wall -lcurses
```

Obr.č.1 Kompilovanie programu

Po spustení kompilovaného programu pomocou príkazu `./program` sa zobrazí základné menu v hry .

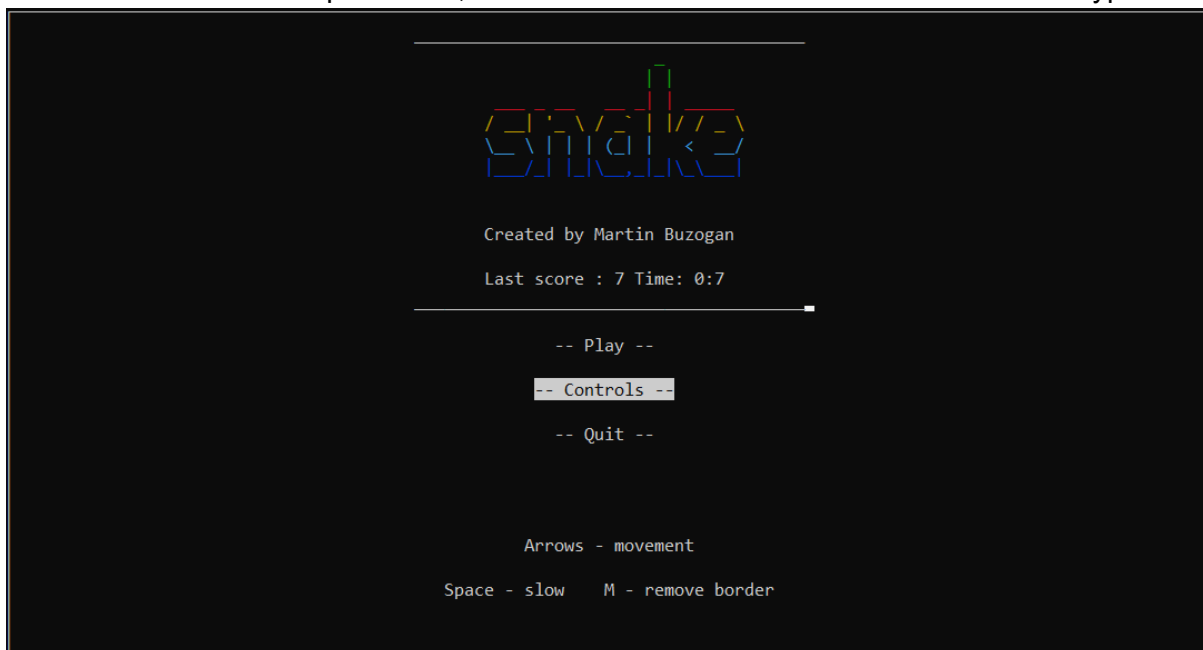
```
./program
```

Obr.č.2 Príkaz na spustenie programu



Obr.č.3 Vstupné menu

-kde hráč ma možnosti spustiť hru , oboznámiť sa s ovládaním alebo samotnú hru vypnúť



Obr.č.4 Ukážka ovládania

Po zvolení možnosti ovládanie sa na obrazovke zobrazí ovládanie hada



Obr.č.5 Spustená samotná hra

Po spustení hry hra automaticky začne had sa bude nepretržite pohybovať v smere ktorý mu určí hráč. Na spodnej časti obrazovky sa nachádza dĺžka hada , čas ako dlho sa hrá hraje , či sú okraje aktívne alebo koľko spomalení ma hráč k dispozícii , alebo koľko vypnutí okrajov ma hráč k dispozícii.



Obr.č.6 Hra sa skončila

Ak hráč narazí do seba alebo do okrajov zobrazí sa nasledujúce okno ,ktoré mu da niekoľko možnosti reštartovať hru alebo sa vrátiť do hlavného menu

Záver

V tejto práci som sa zaoberal vytvorením hry s pomocou vykresľovacej knižnice curses. Počas procesu tvorby hry sa vyskytli mnohé bugy ktoré sa mi časom podarilo vyriešiť. Pri tvorbe tejto hry respektíve hada som využíval polia no iným riešením by bolo používanie štruktúr. Najnáročnejšie na tejto práci bolo vyriešenie vykresľovania pomocou polí a aktívnom počítaní času. Potencionálne nedostatky sú vo vykresľovaní jednotlivých bodov. Možné zlepšenia vidím v aktívne zmenšujúcom sa hracom poly poprípade nejaké prekážky.