

**TECHNICKÁ UNIVERZITA V KOŠICIACH**  
**FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

**Problem Set 6**  
**Arkanoid**

**2022**

**Andrian-Maksym Balaichuk**

## 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.

## Návrh riešenia

Najprv som napísal funkcie, ktoré vypisujú ponuku do konzoly a umožňujú vám vybrať požadovanú položku ponuky:

- void ChooseLevelDifficult(WINDOW\* tmpWin); – funkcia, ktorá zobrazuje ponuku na výber karty;
- void ChooseLevelsName(WINDOW\* tmpWin); – funkcia, ktorá zobrazuje menu na výber úrovne obtiažnosti;
- void PrintAboutGame(WINDOW\* tmpWin); – funkcia, ktorá zobrazuje informácie o hre;
- void CallFunctionMenu(int MenuChoice); – funkcia, ktorá volá príslušné funkcie menu;
- void Menu (void); – funkcia menu.

Ďalej som napísal tri štruktúry: BALL, PADDLE a BRICK a pre každú zo štruktúr sú implementované funkcie kreslenia, mazania, presúvania a resetovania:

- void DrawBall(struct BALL\* Ball); – funkcia, ktorá zobrazuje loptu na obrazovke;
- void DeleteBall(struct BALL\* Ball); – funkcia, ktorá odstráni loptu z obrazovky;
- void ResetBall(struct BALL\* Ball, int PosX, int PosY, int Symbol, COLOR Color); – funkcia, ktorá nastavuje počiatočné hodnoty pre štruktúru BALL.
- void MoveBall(struct BALL\* Ball); – funkcia, ktorá posúva loptičku na hracej ploche;
- void ResetPaddle(struct PADDLE\* Paddle, int PaddleLength, int PaddleHeight, int Symbol, COLOR Color); – funkcia, ktorá nastavuje počiatočné hodnoty pre štruktúru PADDLE;
- void DrawPaddle(struct PADDLE\* Paddle); – funkcia, ktorá zobrazuje pádlo na obrazovke;
- void DeletePaddle(struct PADDLE\* Paddle); – funkcia, ktorá odstráni pádlo z obrazovky;

- void MovePaddle(struct PADDLE\* Paddle); – funkcia, ktorá pohybuje pádlom po hracom poli;
- void DrawBrick(struct BRICK\* Brick); – funkcia, ktorá zobrazuje tehly na obrazovke;
- void DeleteBrick(struct BRICK\* Brick); - funkcia, ktorá odstraňuje tehly z obrazovky.

Pridané aj funkcie na zmenu smeru lopty a kontrolu, či sa odrazila od tehly alebo lopatky:

- void DirectionBallX(struct BALL\* const Ball, DirectionX Direction); – funkcia, ktorá mení smer lopty pozdĺž osi x;
- void DirectionBallY(struct BALL\* const Ball, DirectionY Smer); – funkcia, ktorá mení smer lopty pozdĺž osi y;
- bool CheckPaddleHitBall(struct BALL\* Lopta, struct PADDLE\* Paddle); – funkcia, ktorá kontroluje, či sa loptička odrazila od pádla;
- int BallHitBrick(struct BALL\* Ball, struct BRICK\*\* Tehly); - funkcia, ktorá kontroluje, či loptička zasiahla tehlu.

Pre pole tehly bola vytvorená funkcia, ktorá ich generuje a odstraňuje:

- struct BRICK\*\* CreateBricksMatrix(char\*\* Matrix, int BrickLeght, int BrickHeight); – funkcia, ktorá generuje dvojrozmerné pole typu BRICK z dvojrozmerného poľa Matrix a vracia dvojité ukazovateľ na vygenerovanú maticu;
- void DeleteBricksMatrix(struct BRICK\*\* Matrix); - funkcia, ktorá vymaže pamäť pre vybrané dvojrozmerné pole typu BRICK.

Potom boli pridané funkcie čítania z mapového súboru a niektoré funkcie zobrazovania obrázkov v menu:

- char\*\*\* CreateLevelFromFile(const char\* FileName, int\* LevelCount); - funkcia, ktorá načíta hernú mapu zo súboru a vráti trojrozmerné dynamické pole údajov o umiestnení tehál.
- void DrawAsciiArt(void); – funkcia, ktorá zobrazuje farebný obrázok na úvodnej stránke;

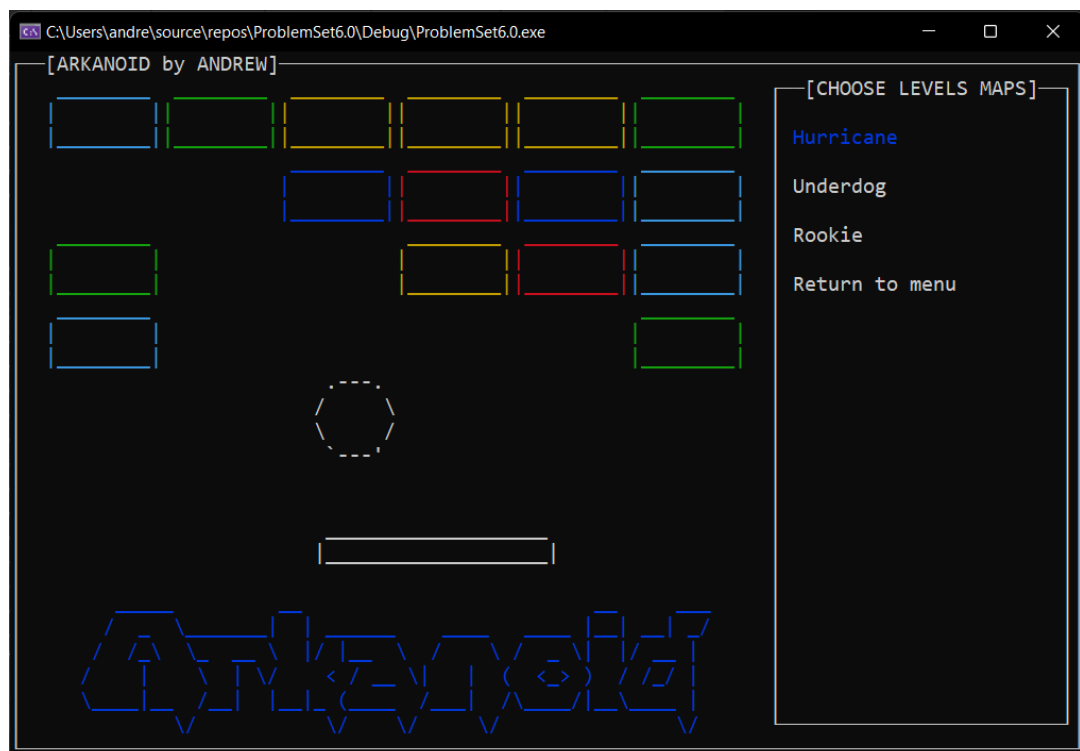
## Použitie programu

Hrateľnosť, hráč ovláda „Vaus“, vesmírne plavidlo, ktoré funguje ako herné „pádlo“, ktoré zabraňuje pádu lopty z hracieho poľa a pokúša sa odraziť loptu proti množstvu kociek. Lopta, ktorá udrie do tehly, spôsobí, že tehla zmizne. Keď sú všetky tehly preč, hráč postúpi do ďalšej úrovne.

Ovládanie v hre pomocou pádla nasmerujte „Vaus“ doľava alebo doprava cez spodnú časť ihriska. Stlačte klávesy so šípkami, vpravo pre pohyb „Vaus“ doprava a ľavé tlačidlo pre pohyb Vaus doľava. Stlačením klávesu Escape môžete ukončiť aktuálnu hru a vrátiť sa do hlavnej ponuky.



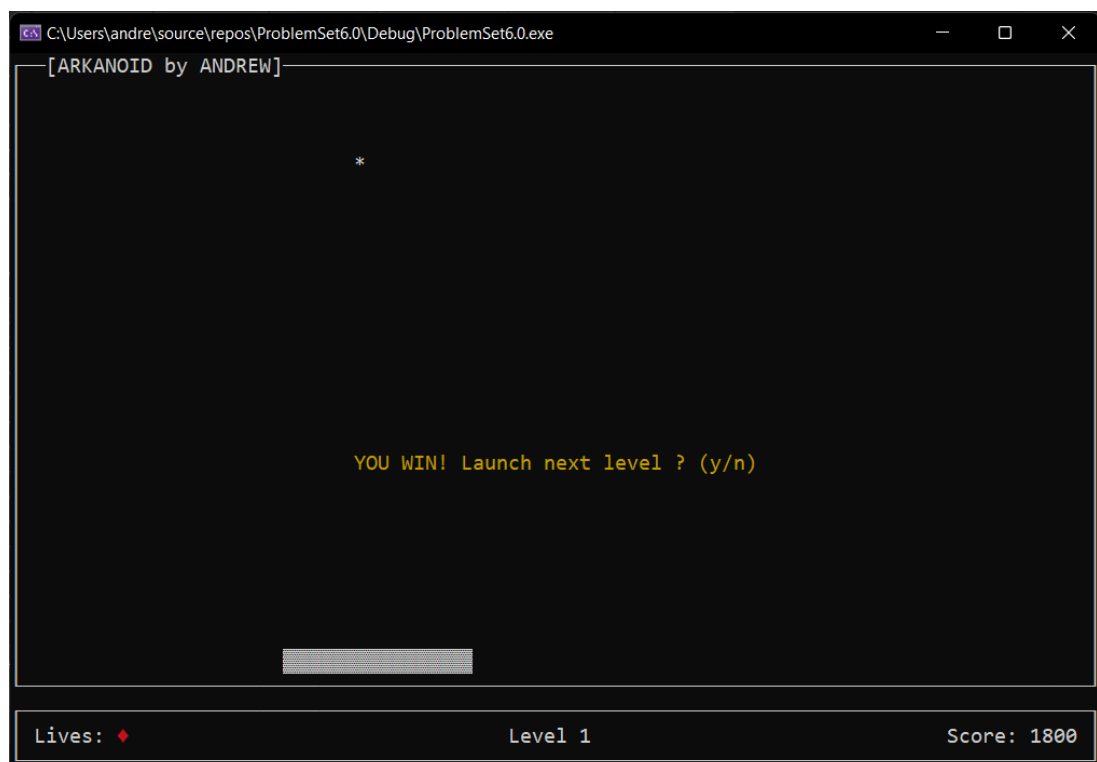
Obr.1 – hlavné menu programu



Obr.2 – menu výberu mapy úrovně



Obr.3 – pohľad na priebeh hry



Obr.4 – obrázok po dokončení úrovne

Pre spuštění programu potřebujete použít tento Makefile:

```
# variables
CC=gcc
CFLAGS=-std=c11 -Wall -Werror -Wno-vla-parameter
LDLIBS=-lm -lcurses
OUTPUT=arkanoid

# targets
all: $(OUTPUT)

$(OUTPUT): paddle.o brick.o ball.o game.o pch.o menu.o main.o
    cppcheck --enable=performance --error-exitcode=1 *.c
    $(CC) $(CFLAGS) paddle.o brick.o ball.o game.o pch.o menu.o main.o $(LDLIBS) -o $(OUTPUT)

main.o: main.c
    $(CC) $(CFLAGS) -c main.c $(LDLIBS) -o main.o

pch.o: pch.c pch.h
    $(CC) $(CFLAGS) -c pch.c $(LDLIBS) -o pch.o

menu.o: menu.c menu.h
    $(CC) $(CFLAGS) -c menu.c $(LDLIBS) -o menu.o

game.o: game.c game.h
    $(CC) $(CFLAGS) -c game.c $(LDLIBS) -o game.o

ball.o: ball.c ball.h
    $(CC) $(CFLAGS) -c ball.c $(LDLIBS) -o ball.o

brick.o: brick.c brick.h
    $(CC) $(CFLAGS) -c brick.c $(LDLIBS) -o brick.o

paddle.o: paddle.c paddle.h
    $(CC) $(CFLAGS) -c paddle.c $(LDLIBS) -o paddle.o

# remove compiled files
clean:
    rm -rf $(OUTPUT) *.o
```

## Záver

V budúcnosti je potrebné do hry pridať bloky, ktoré nezmiznú po tom, ako ich raz zasiahnete. Zlé čítanie zo súboru je možné, ak sa nedodrží prezentovaná forma zápisu mapy. Tiež je podľa mňa potrebné urobiť to tak, aby sa lopta mohla pohybovať pod uhlom 30 alebo 60 stupňov, pretože teraz je možný odraz pod uhlom 45 stupňov, pri zmene tohto sa animácia lopty začína skresľovať, musíme na to nájsť riešenie.