

**ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ ԿՐԹՈՒԹՅԱՆ, ԳԻՏՈՒԹՅԱՆ,
ՄՇԱԿՈՒՅԹԻ ԵՎ ՍՊՈՐՏԻ ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ**

ՀԱՅԱՍՏԱՆԻ ԱԶԳԱՅԻՆ ՊՈԼԻՏԵԽՆԻԿԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

(Սինոփսիս)

Ինստիտուտ՝ Տեղեկատվական և հաղորդակցական տեխնոլոգիաների ու
էլեկտրոնիկայի

Ամբիոն՝ Միկրոէլեկտրոնային սխեմաներ և համակարգեր

ԿՈՒՐՍԱՅԻՆ ԱՇԽԱՏԱՆՔ

Կուրս՝	3-րդ
Առարկա՝	Ծրագրավորման C++ լեզու
Թեմա՝	«Տետրիս» խաղի կոնսոլային տարբերակ
Կատարող՝	Վերոյան Գրիգորի
Ղեկավար՝	Շահոյան Ռաֆայել

Երևան 2021

Կուրսային աշխատանքի թեմա

Իրագործել «Տետրիս» խաղի կոնսոլային տարբերակը

Ծրագիրն իրենից ներկայացնում է ՏԵՏՐԻՍ(Tetris) խաղի կոնսոլային տարբերակը: Խաղը բաղկացած է մենյուից, որն ունի 5 բաժնից.

- PLAY
- RECORDS
- INSTRUCTIONS
- ABOUT GAME
- EXIT

PLAY բաժին մտնելով սկսվում է խաղը: Խաղը դասական տետրիս է: Պատահական խաղաքարեր ընկնում են, և դասավորվում խաղադաշտում: Երբ խաղադաշտում լրացվում է խաղաքարերի մասնիկներից լրիվ տող, այդ տողն անհետանում է, և խաղացողն ստանում է միավորներ:

RECORDS բաժնում գրանվում է խաղից ստացված միավորները: Այս բաժնի իրագործումը դեռևս իրականացված չէ ժամանակի սղության պատճառով:

INSTRUCTIONS բաժին մուտք գործելիս արտածվում է մեյույում և ընդհանրապես խաղում տեղաշարժվելու համար օգտագործվող ստեղների մասին տեղեկատվություն:

Key_UP կամ w, W:

Մենյուում բարձրանում է վերև:

Խաղում շրջում է խաղաքարը 90 աստիճանով:

Key_DOWN կամ s, S:

Մենյուում իջնում է ներքև:

Խաղում արագացնում է իջնող խաղաքարի արագությունը:

Key_LEFT կամ a, A:

Խաղում տեղափոխում է խաղաքարն ձախ:

Key_RIGHT կամ d, D:

Խաղում տեղափոխում է խաղաքարը աջ:

ENTER:

Մենյույում ընտրում է բաժինը:

ABOUT GAME բաժին մուտք գործելիս արտածվում է Տետրիս խաղի մասին փոքրածավալ ինֆորմացիա:

EXIT բաժին մտնելով ավարտվում է կոնսոլում ծրագրի(խաղ և մենյու) արողջ աշխատանքը՝ ետ վերադառնալու խոստում ստանալով:

ԾՐԱԳՐԻ ԿԱՌՈՒՑՎԱԾՔԸ

Ամբողջ ծրագրի ընթացքը բաժանված է տրամաբանական փոքր մասերի, որոնք գտնվում են src «թղթապանակում», իսկ .cpp ֆայլերում գտնվող ֆունկցիաների նախատիպերը գտնվում են includes «թղթապանակի» headers.hpp ֆայլում: includes թղթապանակում է գտնվում նաև ծրագրի հատվածներում օգտագործված գույների իմպլեմենտացիան՝ styles.hpp ֆայլում:

Includes՝

headers.hpp

styles.hpp

src՝

about.cpp

exit.cpp

input.cpp

menu.cpp

instructions.cpp

menu_animation.cpp

game.cpp

general_functions.cpp

src:

- about.cpp

Ֆայլում գրված է aboutGame() ֆունկցիան որը կոնսոլում արտածում է խաղի մասին փոքրածավալ ինֆորմացիա:

- exit.cpp

Ֆայլում գրված են printExit() և gameOver() ֆունկցիաները, որոնք համապատասխանաբար կոնսոլում արտածում են COME BACK LATER և GAME OVER արտահայտությունները:

- input.cpp

Ֆայլի cbreak() ֆունկցիան տերմինալին նախապատրաստում է ներմուծման, normal() ֆունկցիան տերմինալը վերադարձնում է նորմալ վիճակի, keypress() ֆունկցիան վերադարձնում է ներմուծված ստեղծ, և - եթե ստեղծ չի ներմուծվել:

- instructions.cpp

Ֆայլում գրված instruct() ֆունկցիան արտածում է ծրագրում տեղակողմնորոշվելու(նավիգացման) մասին տեղեկատվություն:

- `menu_animation.cpp`

Ֆայլում առկա printTetris(int, int) ֆունկցիան արտածում է տետրիս բառը, իսկ

fallingTetris(bool, int&) ֆունկցիան արտածում է փոքրիկ անիմացիա՝ ընկնող տետրամինոյի տեսքով:

- `menu.cpp`

Ֆայլում գտնվող ֆունկցիաները և նրանց գործառույթները.

printSectionName(char*, bool) տպում է բոլոր բաժինները(ներառյալ նաև ընտրված):

printAll(int) տպում է բաժինները, փոփոխվող ընտրված բաժնի պայմանում:

menu() իրականացնում է դիսամիկ մենյուի աշխատանք:

- `general_functions.cpp`

Ֆայլի gotoxy(int, int) ֆունկցիան տերմինալի ընտրված կոորդինատներով հատվածում է իրականացնում արտածումը:

- `game.cpp`

Ֆայլում գտնվող ֆունկցիաները և նրանց գործառույթները.

getPiece(int) ստանում է պատահականորեն գեներացված խաղաքարի կոորդինատները գեներացնում խաղաքարը:

rotatePiece() շրջում է խաղաքարը 90 աստիճանով, և ստանում դրա կոորդինատները:

check() ստուգում է արդյոք խաղաքարը գտնվում է խաղադաշտի «վալիդ» հատվածում:

insertPiece() նոր խաղաքարը ներմուծում է խաղադաշտ:

initEmptyField() ստեղծում է դատարկ խաղադաշտ:

displayField() արտածում է խաղադաշտը:

game() իրականացնում է խաղի հիմնական լոգիկան:

includes:

- `headers.hpp`

```
#ifndef HEADERS_H
```

```
#define HEADERS_H
```

```
/* Brief: Sets up terminal for one-char-at-a-time reads*/
```

```
void cbreak();
```

```
/* Brief: Resets terminal to normal state after cbreak() */
```

```
void normal();
```

```
/*
```

```
    Brief: Checks keyboard buffer (stdin) and returns the pressed key
```

```
    Return: Pressed key or -1
```

```
*/
```

```
int keypress();
```

```
/*Brief: Sets (x, y) coordinates to display things in the terminal*/
```

```
void    gotoxy(int x, int y);
```

```
/*
```

```
    Brief: Little beautiful animation in the menu - falling tetromino in the menu
```

```
    : Gets boolean variable which sets when start animation, and an integer, which is y coordinate of  
    falling tetromino
```

```
*/
```

```
void    fallingTetris(bool magic, int &z);
```

```
/*
```

```
    Brief: Displays word "TETRIS"
```

```
    : Gets (x, y) coordinates to print the word
```

```
*/
```

```
void    printTetris(int x, int y);
```

```
/*Brief: Displays letter after exiting the game*/
```

```
void    printExit();
```

```
/* Brief: Displays brief information about game and it's developer :) */
```

```
void aboutGame();
```

```
/* Brief: Displays information about game control */
```

```
void instruct();
```

```
/* Brief: Displays GAME OVER */
```

```
void gameOver();
```

```
/*
```

```
    Brief: Displays Menu sections. If Boolean value is true, section displayed as selected
```

```
    : Gets boolean value which defines is section selected or not, and a string, which is section name
```

```
*/
```

```
void printSectionName(char *str, bool isSelected);
```

```
/*
```

```
    Brief: Prints all menu sections
```

```
    : Gets Integer which says which section is selected
```

```
*/
```

```
void printAll(int selected);
```

```
/* Brief: Displays interactive menu */
```

```
void menu();
```

```
/*
```

```
    Brief: Getting randomized piece's (x, y) coordinates
```

```
    : Gets random piece's identifier
```

```
*/
```

```

void  getPiece(int n);

/*  Brief: Rotating piece, and getting it's rotated (x, y) coordinates*/
void  rotatePiece();

/*
    Brief: Checking if piece is ininside gaming FIELD or not
    Return: returns truthy value if piece is in field, and falsy otherwise
*/
bool check();

/*  Brief: Inserting new Piece in the gaming FIELD */
void  insertPiece();

/*  Brief: Initializing empty gaming FIELD*/
void  initEmptyField();

/*  Brief: Displaying gaming FIELD */
void  displayField();

/*  Brief: Implementing the main logic of game */
void  game();

#endif

```

- styles.hpp

```
#ifndef STYLES_H
#define STYLES_H

#define BOLDBLUE    "\033[1m\033[34m"    // Bold Blue
#define YELLOW      "\033[33m"           // Yellow
#define RED         "\033[31m"           // Red
#define WHITE       "\033[37m"           // White
#define GREEN       "\033[32m"           // Green
#define CYAN        "\033[36m"           // Cyan
#define RESET       "\033[0m"           // Resetting color
#define BLACK       "\033[30m"           // Black
#define BOLDRED     "\033[1m\033[31m"    // Bold Red
#define MAGENTA     "\033[35m"           // Magenta

#define ITALIC      "\e[3m"              // Italic
#define BLINK       "\e[5m"              // Blink
#define UNDERLINED  "\e[4m"              // Underlined
#define BOLD        "\e[1m"              // Bold

#endif
```


ԵԶՐԱԿԱՑՈՒԹՅՈՒՆ

Կուրսային աշխատանքի կատարման ընթացքում սովորեցի աշխատել vim-ով կոդ գրել, չէ որ ոչ բոլոր պահերին կարող եմ ձեռքի տակ լինել code editor-ներ: Vim-ով աշխատելու ընթացքում հանդիպած բոլոր բարդություններից հետո արդեն շատ պարզ է դառնում թե ինչպես են «տակից» աշխատում բազմաթիվ ֆայլերից բաղկացած ծրագրեր:

Ավելի խորն ուսումնասիրեցի տերմինալի հետ աշխատանքի առանձնահատկությունները, սովորեցի աշխատել header ֆայլերի հետ: Այս աշխատանքը նաև մի հետաքրքիր երևույթ բացահայտեց ինձ համար, որ ոչ ամեն ինչ է, որ հնարավոր է գտնել համացանցում:

Written by Grigori Verdyan