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

**ՀԱՅԱՍՏԱՆԻ ԱԶԳԱՅԻՆ ՊՈԼԻՏԵԽՆԻԿԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ**

**Ինստիտուտ՝** Տեղեկատվական և հաղորդակցական տեխնոլոգիաների ու էլեկտրոնիկայի

**Ամբիոն՝**  Միկրոէլեկտրոնային սխեմաներ և համակարգեր

**ԿՈՒՐՍԱՅԻՆ ԱՇԽԱՏԱՆՔ**

**Կուրս՝** 3-րդ

**Առարկա՝** Ծրագրավորման C++ լեզու

**Թեմա՝** Իրագործել «Ականազերծիչ» (“Minesweeper”) խաղի կոնսոլային տարբերակը:

**Կատարող՝** Բաղոյան Յուրա

**Ղեկավար՝** Շահոյան Ռաֆայել

**Երևան 2021**

**Բովանդակություն**

[Minesweeper խաղի մասին 3](file:///C:\Users\user\Downloads\Telegram%20Desktop\Course_Project.docx#_Toc91162828)

[Ֆայլային համակարգի կառուցվածքը 4](file:///C:\Users\user\Downloads\Telegram%20Desktop\Course_Project.docx#_Toc91162829)

[Հաշվիչի աշխատանքի սկզբունքը 5](file:///C:\Users\user\Downloads\Telegram%20Desktop\Course_Project.docx#_Toc91162831)

[Ֆունկցիաները ըստ ֆայլերի 5](file:///C:\Users\user\Downloads\Telegram%20Desktop\Course_Project.docx#_Toc91162832)

[input.cpp 5](file:///C:\Users\user\Downloads\Telegram%20Desktop\Course_Project.docx#_Toc91162833)

main.cpp [5](file:///C:\Users\user\Downloads\Telegram%20Desktop\Course_Project.docx#_Toc91162837)

[menu.cpp 6](file:///C:\Users\user\Downloads\Telegram%20Desktop\Course_Project.docx#_Toc91162838)

preGame.cpp [6](file:///C:\Users\user\Downloads\Telegram%20Desktop\Course_Project.docx#_Toc91162839)

game.cpp [7](file:///C:\Users\user\Downloads\Telegram%20Desktop\Course_Project.docx#_Toc91162840)

[options.cpp 8](file:///C:\Users\user\Downloads\Telegram%20Desktop\Course_Project.docx#_Toc91162841)

[records.cpp 8](file:///C:\Users\user\Downloads\Telegram%20Desktop\Course_Project.docx#_Toc91162842)

[Օգտագործողի ինտերֆեյս 10](file:///C:\Users\user\Downloads\Telegram%20Desktop\Course_Project.docx#_Toc91162845)

[Օգտագործված ռեսուրսներ 16](file:///C:\Users\user\Downloads\Telegram%20Desktop\Course_Project.docx#_Toc91162848)

**Minesweeper խաղի մասին**

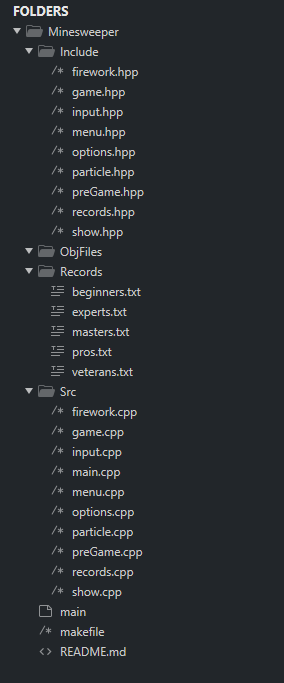
Minesweeper- ը առաջին համակարգչային խաղերից մեկն էր: Դրա պատմությունը սկիզբ է առել 1960-ականներից, և մարդիկ մինչ այժմ խաղում են այն: Կան խաղի մի շարք տարբերակներ, սկայան ամենատարծվածը դրա 2D տարբերակն է:

Հարթ խաղադաշտը բաժանված է բջիջների (քառակուսիների), որոնցից մի քանիսը «ականապատված են»: Հայտնի է «ականապատված» բջիջների թիվը եւ խաղի նպատակն է բացել բոլոր բջիջները, որոնք ականներ չեն պարունակում: Խաղացողը բացում է բջիջները՝ փորձելով չբացել ականով բջիջը։ Ականով բջիջը բացելով՝ պարտվում է։ Եթե ​​բացված բջջի տակ ական չկա, ապա դրա փոխարեն հայտնվում է մի թիվ, որը ցույց է տալիս, թե բացված բջջի հարեւանությամբ քանի բջիջ է «ականապատված»: Այդ թվերի միջոցով, խաղացողը փորձում է հաշվարկել ականների գտնվելու վայրը: Խաղացողը կարող է նշել «ականապատված» բջիջները, որպեսզի դրանք պատահաբար չբացվեն: Բացելով բոլոր «ականապատված» բջիջները՝ խաղացողը հաղթում է։

**Ֆայլային համակագի կառուծվացքը**

Ընդհանուր ֆայլային կառուցվածքը ունի հետեւյալ տեսքը՝ 4 պանակ, որոնք պարունակում են բոլոր անհրաժեշտ ֆայլերը: *Include* պանակը իր մեջ պարունակում է բոլոր header ֆայլերը: իսկ *Src* պանակը պարունակում է բոլոր cpp ֆայլերը: *ObjFiles* պանակը պարունակում է կոմպիլացիայից հետո ստացված օբյեկտ ֆայլերը: Records պանակը իր մեջ պարունակում է ըստ մակարդակների բաժանված txt ֆայլեր որտեղ գրված են record գրանցած օգտատերի անունները եւ ժամանակները: Ինչպես նաեւ makefile եւ *README* ֆայլեր:

***input.cpp / input.հpp***



Ֆունկցիաներ որոնք օգտագործվում են բոլոր ֆայլերի մեջ: Օրինակ ֆունկցիյա որով ստանում ենք օգտատիրոջ սեխմած ստեղնը:

***firework.cpp, particle.cpp / firework.hpp, particle.hpp***

Ֆայլեր որտեղ իրագործված է հրավառություն նոր record գրանցած օգտատերի համար:

***menu.cpp / menu.hpp***

Ֆայլեր որտեղ իրագործված են խաղի գլխավոր մենյուն:

***options.cpp / options.hpp***

Ֆայլեր որտեղ իրագործված են խաղի մակարդակներ եւ ռեժիմներ փոփոխելու մենյուն:

***preGame.cpp / preGame.hpp***

Ֆայլեր որտեղ իրագործված են բուն խաղը սկսելուց առաջ կատարվող գործողությունները: Որոնցից են ընտրված մակարդակից կախված խաղադաշտի ստեղծումը, ականների դաշտում բաշխումը, դաշտը թվերով լցնելը եւ այլն:

***game.cpp / game.hpp***

Ֆայլեր որտեղ իրագործված են բուն խաղը:

***records.cpp / records.hpp***

Ֆայլեր որտեղ իրագործված են 2 գաղափար՝ record գրանցած օգտատերերի ցանկը արտածելու ֆունկցիաները, ինչպես նաեւ նոր record գրանցած օգտաերերի տվյալների գրանցումը:

***show.cpp / show.hpp***

Ֆայլեր որտեղ իրագործված են բոլոր մեծ չափեր ունեցող օբյեկտների արտածուծը էկրանին:

**Ֆունկիցիաները ըստ ֆայլերի**

**Input.cpp**

* void cbreak();

Կարգավորում է տերմինալը մեկ նիշը մեկ անգամ կարդալու համար:

* void normal();

Վերականգնել տերմինալը նորմալ վիճակի cbreak()-ից հետո:

* int keypress();

Ստուգում է ստեղնաշարի բուֆերը (stdin) և վերադարձնում սեղմված ստեղնը: Եթե ոչինչ չի սեղմվել ապա վերադաձնում է -1:

* void gotoxy(int,int);

Տեղափոխում է կուրսորը տված x y կորդինատների վրա որտեղ x-ը սյուներն են, իսկ y-ը տողերը:

* void colorCout(const std::string, const int);

Ստանում է տեքստ եւ գունավորման թիվ ու տպում տեքստը ընտրված գույնավորման թվին համապատասխան գույնով:

* void userWinSize(int&, int&);

Վերադաձնում է օգտատիրոջ կոնսոլի չափերը խաղը կոնսոլի կենտրոնում տպելու նպատակով:

* bool winSizeChanged(int&, int&, const int, const int);

Վերցնում է օգտատիրոջ կոնսոլի չափերը եւ ստուգում դրանք նախկին չափերի հետ, եթե դրանք չեն համապատասխանում ապա փոխում է նախկին չափերը նորի հետ:

* void clearLine(const int)

Ջնջում է ընթացիկ տողը:

**main.cpp**

int main()

Տեղի է ունենում խանի անվան անիմացիոն հայտնվում, վերցվում է օգտատիրոջ կոնսոլի չափերը՝ խաղը կոնսոլի կենտրոնին տպելու նպատակով: Որից հետո կանչում է մենյուն:

**menu.cpp**

* void CursorMoveInMenu();

Տեղափոխում է կորսորը ըստ սեխմված ստեղնի:

* void Menu\_choose();

Գլխավոր մենյուն:

* void printMenu();

Տպում է մենյուն իր բոլոր օբյեկտներով:

* void selectedMenu();

Կանչում է նոր էկրանը ըստ օգտատերի ընտրության:

**preGame.cpp**

* void addOneInCellsAround();

Ավելացնում է 1 Back[i][j] բջջի շուրջ գտնվող բջիջներում:

* void fillRandomMines();

Կախված օգտատիրոջ մակարդակի ընտրությունից՝ պատահական սկզբունքով լրացնում է Back մատրիցը տրված քանակի ականներով:

* void fillBoundsOfMatrix();

Back մատրիցի սահմանները սկզբնարժեքավորում է -2, որպեսզի հետագայում կուրսորը այդ բջջի վրա չքայլի:

* void fillNumbersAroundMines();

Գտնում է ականները եւ կանչում addOneInCellsAround() ֆունկցիան տվյալ ականի շուրջ գտնվող թվերին 1 գումարելով:

* void gamePreparation();

Ստեղծում է Back (integer) եւ Front (char) մատրիցները: Back մատրիցը պարունակում է դաշտի թվերը, իսկ Front մատրիցը այն մատրիցն է, որը տեսնելու է օգտատերը:

* void level();

Կախված օգտատիրոջ մակարդակի ընտրությունից՝ վերադարձնում է Front եւ Back մատրիցների չափերը եւ ականների քանակը:

**game.cpp**

* void Boom();

Օգտատերը բացել է ականով բջիջը: Այս դեպքում մեծ տառերով գրվում է GAME OVER, օգտատերին ցույց է տրվում բոլոր ականնետի տեղերը եւ տվում է խաղը վերսկսելու կամ գլխավոր մենյու վերադառնաու հնարավուրություն:

* void Empty();

Ֆունկցիան կանչվում է երբ օգտատիրոջ բացած բջիջը դատարկ է՝ այսինքն Back[i][j] = 0: Այս դեպքում ռեկուրսորեն բացվում են այդ բջջի շուրջ բոլոր բջիջները, եթե դրա դատարկ բջիջները

* void game();

Գլխավոր խաղը:

* bool isWin();

Ստուգում է արդյոք խաղադաշտի մեջ մնացել են չբացված բջիչներ բացի ականներից թե ոչ: Եթե այո ապա վերադաձնում է false, հակառակ դեպքում նշանակում է որ օգտատերը հաղթել է՝ վերադարձնում է true:

* void Open();

Բացում է ընտրված բջիջը: Կարդում է տվյալ բջիջը Back-ից եւ գրում այն Front-ի մեջ, այնուհետեւ տպում:

* void OpenAround();

Եթե ընթացիկ բջջի շուջ արդեն նշված են ականների տեղերը ապա տվյալ բջջի վրա սեխմելով (որը արդեն բաց էր) բացում է ընտրված բջջի շուրջ եղած բոլոր բջիջները բացի նշվածներից: Կարդում է տվյալ բջիջը Back-ից եւ գրում այն Front-ի մեջ, այնուհետեւ տպում: Եթե օգտատերը սխալվել էր եւ ընթացիկ բջջի շուրջ ական եղավ, ապա GAME OVER:

* void pause();

Խաղի դադար ռեժիմը: Այս ռեժիմ կարող են մտնել կամ այդտեղից վերադառնալ խաղ սեղմելով ‘p’ կամ ‘P’: Դադարի ռեժիմից կարող են նաեւ դուրս գալ գլխավոր մենյու ‘esc’ սեղմելով: Դադարի ռեժիմում խաղում ժամանակը կանգ է առնում:

* void printChar();

Ստանում է Front եւ ընթացիկ i j կորդինատները: Այնուհետեւ տպում է այդ նիշն իր հատուկ գույնով:

* void printBombCount();

Տպում է այդ պահին մնացաց ականների քանակը որոնք օգտատերը դեռ չի նշել:

* void printGreenChar():

Ստանում է Front եւ ընթացիկ i j կորդինատները: Այնուհետեւ տպում է այդ նիշն կուրսորի գույնով (մեր դեպքում կանաչ):

* void win();

Այս դեպքում մեծ տառերով գրվում է WINNER, օգտատերին ցույց է տրվում բոլոր ականնետի տեղերը: Այնուհետեւ ստուգվում արդյոք նոր record գրանցվել է թե ոչ: Եթե այո ապա կանչվում է համապատասխան ֆունկցիան records.cpp-ից: Իսկ եթե ոչ ապա տվում է խաղը վերսկսելու կամ գլխավոր մենյու վերադառնաու հնարավուրություն:

**options.cpp**

* void cursorMoveInCustom();

Տեղափոխում է կորսորը custom ռեժիմի մենյուի մեջ ըստ սեխմված ստեղնի:

* void cursorMoveInOptions();

Տեղափոխում է կորսորը options մենյուի մեջ ըստ սեխմված ստեղնի:

* void customMode();

Custom ռեժիմ որտեղ օգտատերը կարող է ընտրել դաշտի չափսերը եւ ականների քանակը: Դաշտի չափերը չեն կարող գերազանցել 18x18 չափերը, իսկ ականների քանակը չեն կարող գերազանցել 99:

* void options();

Գլխավոր ֆունկցիան որը կանչվում է menu.cpp-ից:

* void pressedGodMode();

Միացնել կամ անջատել God ռեժիմը: Այս ռեժիմում խաղադաշտի աջ մասում երեւում են բոլոր բջիջների պարունակությունները:

* void printOptions();

Տպում է options էկրանի բոլոր օբյեկտները:

**records.cpp**

* void checkingTimeInTop();

Օգտատերի հաղթելուց հետո ստուգում նոր record գրանցելու համար:

* std::string fileWay();

Ըստ օգտատերի կողմից ընտրված մակարդակի որոշում է txt ֆայլի ճանապարհը, որտեղից որ պիտի կարդա արդեն գրանցված օգտատերերի անուններն ու ժամանակները:

* void InputFromFile();

Ըստ ընտրված մակարդակի եւ նրան համապատասխան txt ֆայլի՝ կարդում է այդ ֆայլից նախկինում գրանցված օգտատերերի անուններն ու ժամանակները եւ պահում զանգվածներում:

* std::string inputUserNickName();

Record գրանցելու նպատակով պահանջում է օգտատերի անունը:

* void newRecord();

Գտնում է նոր գրանցված record-ի տեղը տոպում:

* void outputFile();

Օգտատերի անունը ստանալուց հետո գրանցում է նրա անունը եւ ժամանակը մակարդակին համապատասխան txt ֆայլում:

* void printRecords();

Տպում է records էկրանի բոլոր օբյեկտները:

* void printTime();

Ստանում է օգտատերի ժամանակը վարկյաններով եւ տպում է այն ժամ/րոպե/վարկյան ֆորմատով:

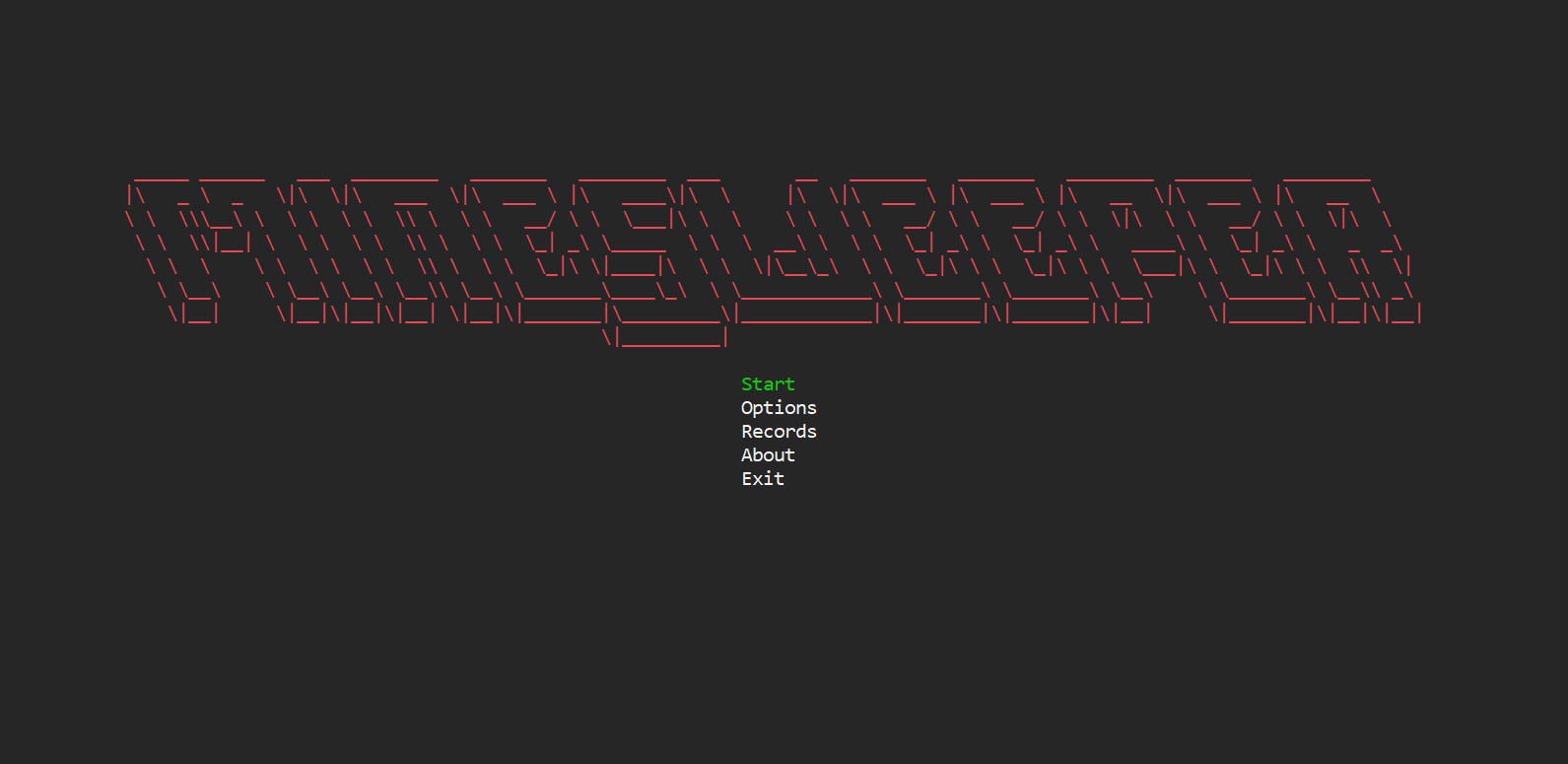
* void printTop();

Կարդում է txt ֆայլից տվյալ մակարդակի տոպ անունները եւ ժամանակը ու տպում էկրանին:

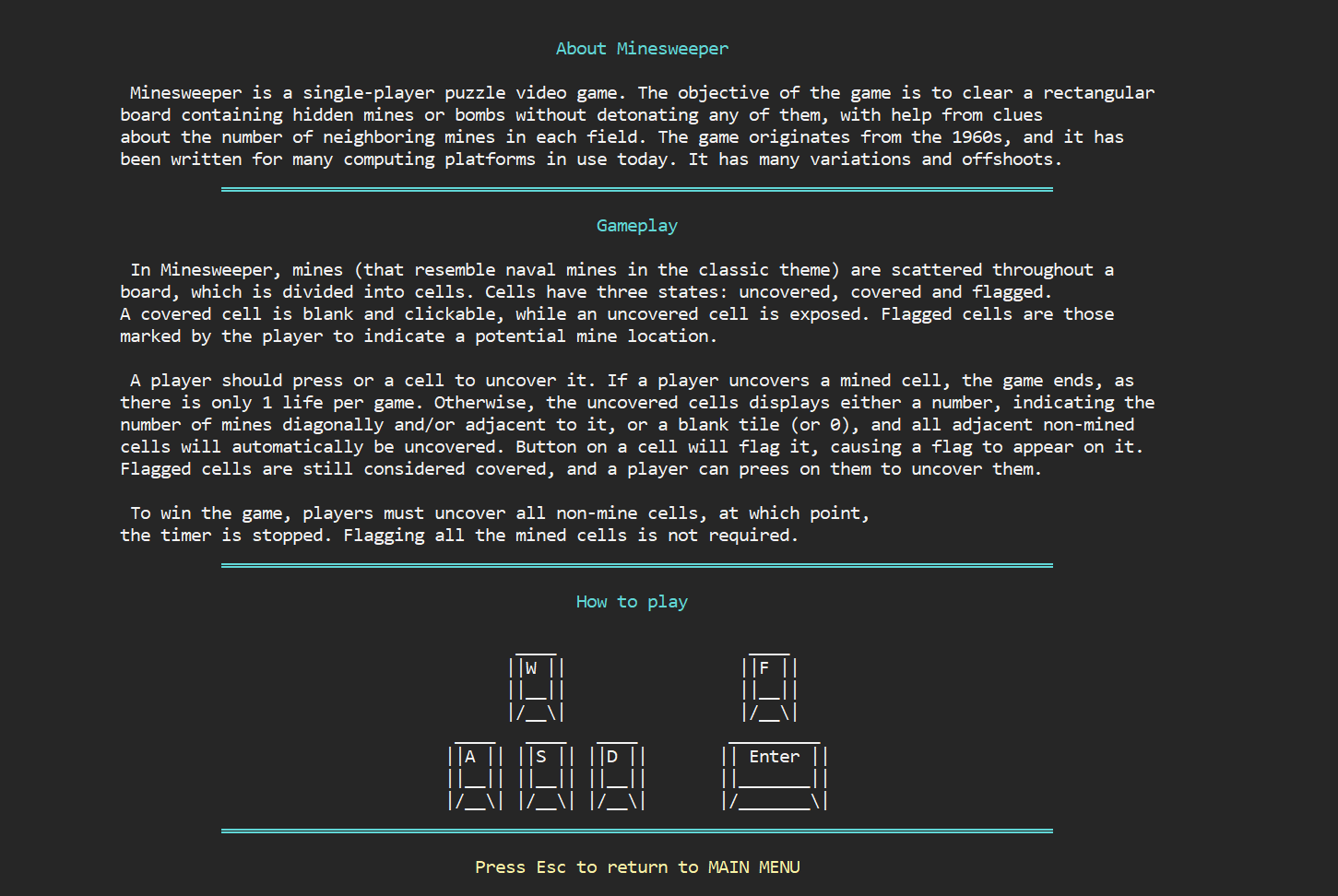
* void records();

Գլխավոր records էկրանը:

**Օգտագործողի ինտերֆեյս**

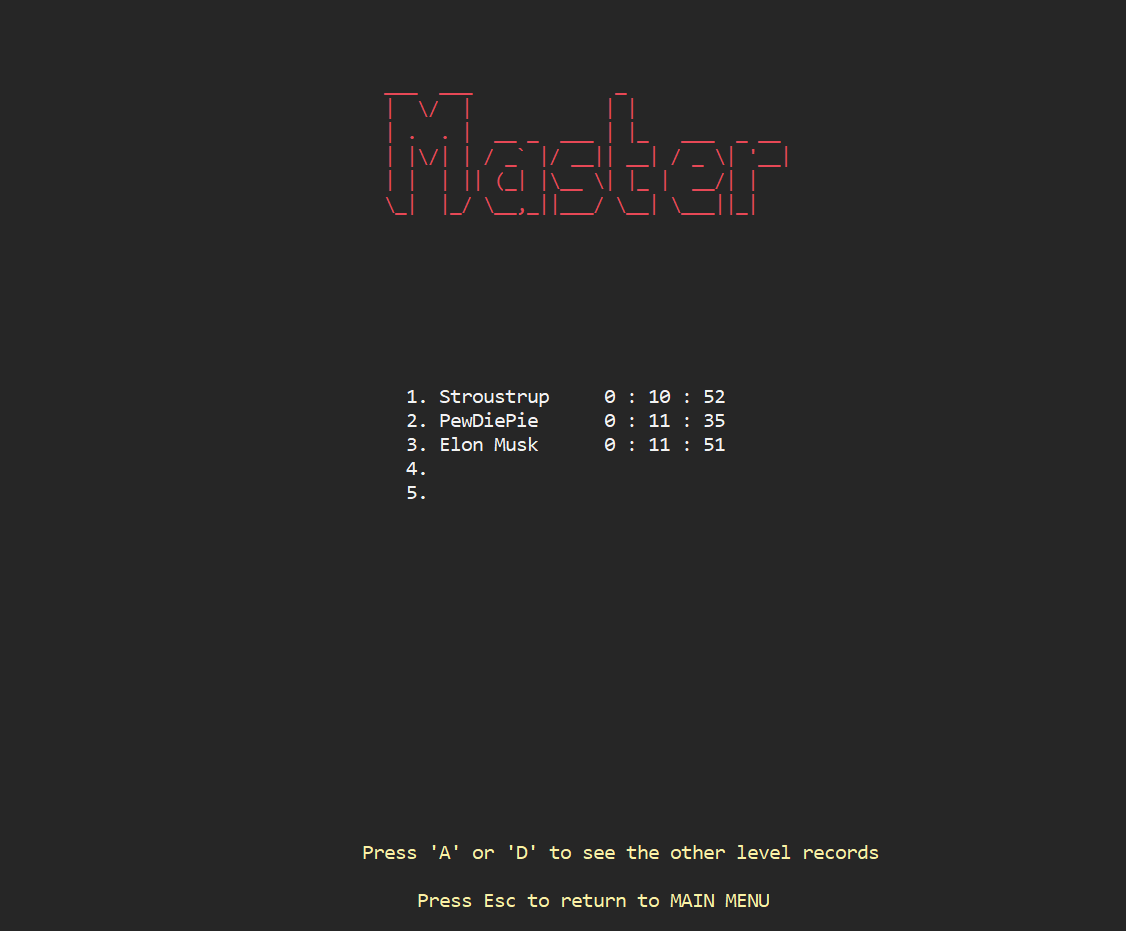
**Menu**

**About**

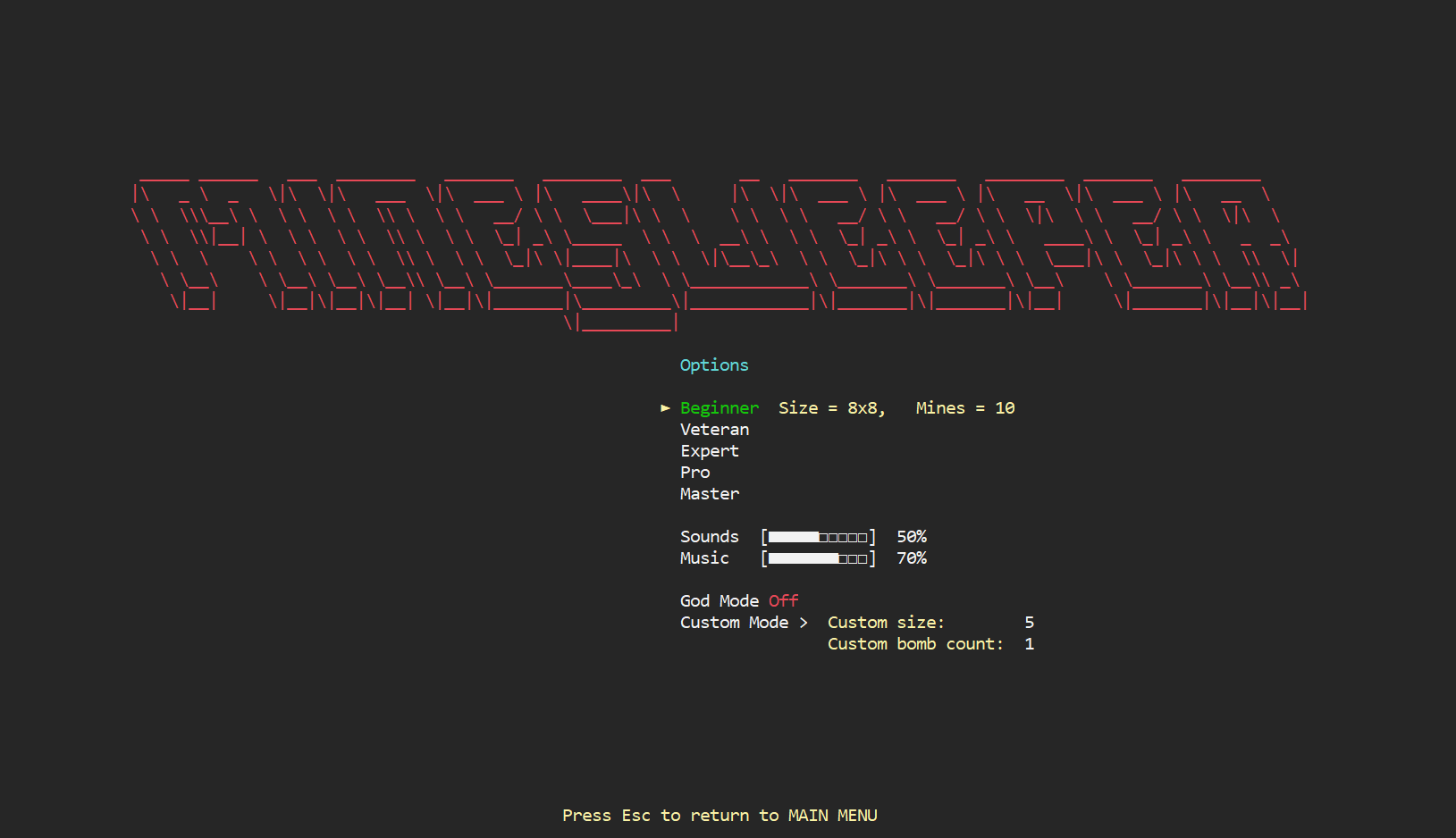


**Records**

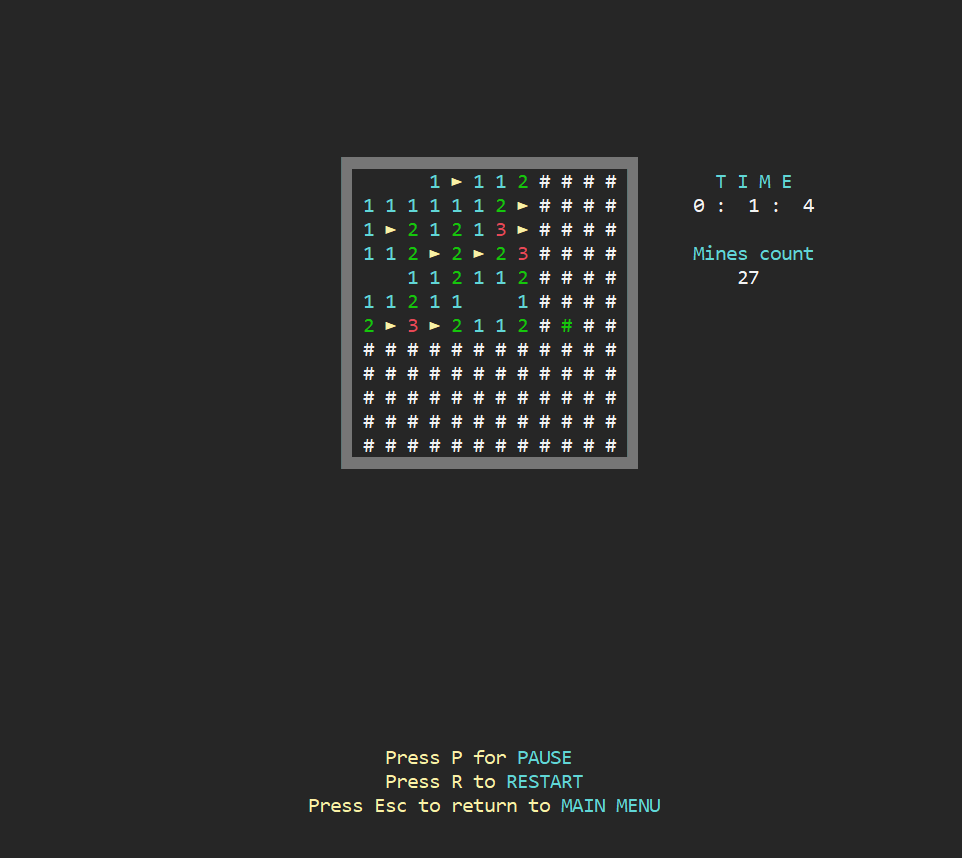
****



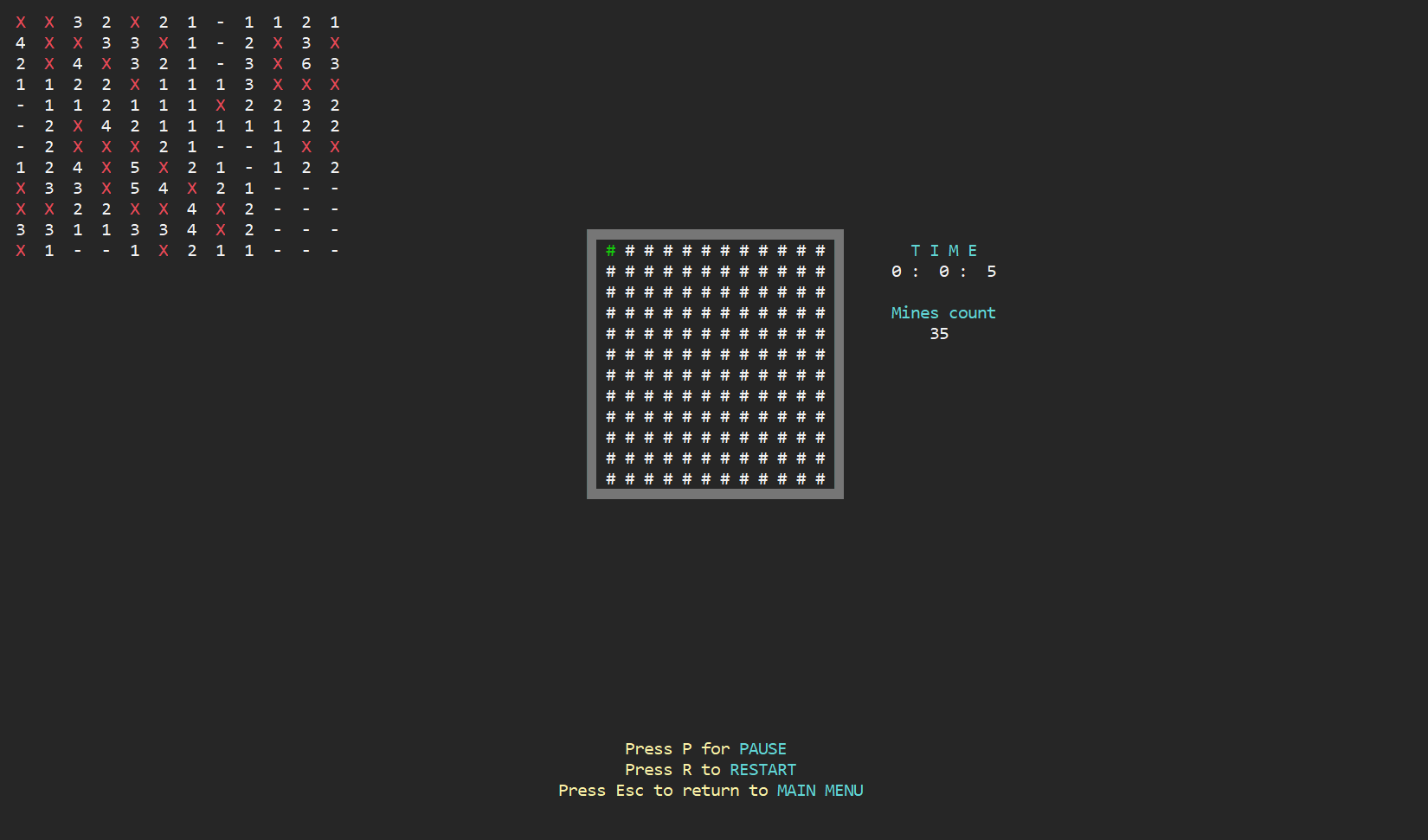
**Options**



**Main game**



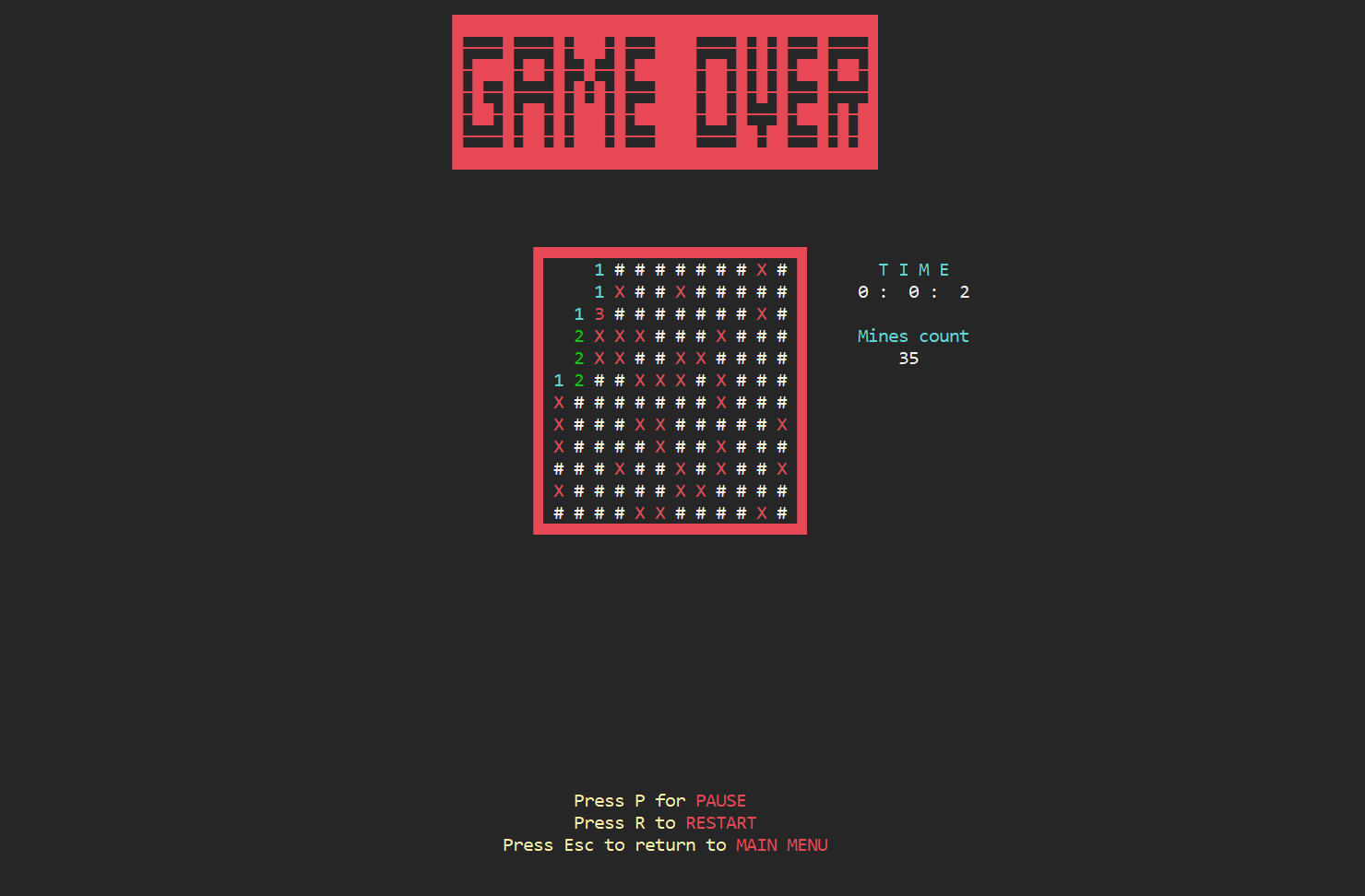
**God Mode**



**Winner**



**Game over**



**Օգտագործված ռեսուրսներ**

**About game in Wikipedia:**  *https://en.wikipedia.org/wiki/Minesweeper\_(video\_game)*

**Color scheme**: *https://en.wikipedia.org/wiki/ANSI\_escape\_code#8-bit*

**Unusual symbols:**  *https://coolsymbol.com*

**For text design:** *https://www.messletters.com/en/big-text/*