



Universitatea „Ștefan cel Mare” - Suceava
Facultatea de Inginerie Electrică și Știința Calculatoarelor

Proiect la POO din cadrul cursurilor de Licență

Inginerie Electrică și Știința Calculatoarelor

Îndrumător: s.l. drd. ing. Prodan Remus

PROIECT

MINESWEEPER

Student:
Bivol Andrei-Sorin

Grupa:
3134A CR

- SUCEAVA 2023 -

Cuprins:

Tema si motivația alegerii	3
Descierea problemei:.....	3
Deschiderea proiectului din consola Linux:	3
Elemente specifice POO	4
<i>Figura 1. Header-ul clasei BoardHandler</i>	<i>4</i>
<i>Figura 2. Header-ul clasei MineSweeper.....</i>	<i>5</i>
<i>Figura 3. Fișierul sursă main.....</i>	<i>6</i>
Cazuri excepționale.....	6
<i>Figura 4. Rezultatul introducerii unei coordonate greșite</i>	<i>6</i>
<i>Figura 5. Tabel cu o căsuță marcată cu flag.....</i>	<i>7</i>
<i>Figura 6. Rezultatul pierderii jocului</i>	<i>7</i>

Tema si motivația alegerii

Tema principală a jocului Minesweeper este rezolvarea unui puzzle logic complex, în care jucătorii trebuie să evite minele și să dezvăluie zonele sigure ale câmpului de joc. Este un joc care pune accent pe gândirea strategică, deducție și evaluarea riscurilor. Deși pare simplu la prima vedere, Minesweeper ascunde o rețea complexă de indicii și strategii, făcându-l un joc intelectual provocator.

Motivul ales pentru crearea jocului Minesweeper se bazează pe simplitatea, distracția și abilitatea sa de a oferi elemente de intrigă, logică și strategie.

Descierea problemei:

Funcționarea jocului poate fi împărțită în două părți: Inițializarea jocului și desfășurarea jocului.

I. Inițializarea jocului:

- Generează un teren minat cu 81 de celule. Acest teren minat conține 9 bombe plasate în mod aleator. Fiecare celulă reprezintă o bombă sau o cifră. Cifra reprezintă câte bombe se află în vecinătatea celulei respective.
- Creează un tabel cu 81 de celule, fiecare celulă reprezentând o mască care ascunde bombele și cifrele.
- Afișează un mesaj cu instrucțiuni care dispar după apăsarea tastei ENTER.
- Afișează tabelul mască.
- Afișează modul de modificare al tabelului și anume STEP sau FLAG.

II. Desfășurarea jocului:

- Jucătorul introduce o coordonată din tabel sau introduce comanda pentru a schimba dintre modul STEP/FLAG.
- La efectuarea comenzii, ecranul consolei se va actualiza și se va afișa tabelul alterat și un mesaj ce indică modul curent, STEP sau FLAG.

Deschiderea proiectului din consola Linux:

Pentru generarea fișierului executabil am specificat terminalului locația proiectului. Pentru a genera fișierul executabil am folosit următoarea instrucțiune:

```
g++ main.cpp ./src/Minesweeper.cpp ./src/BoardHandler.cpp -  
I./include -o run
```

Fișierul generat este cel denumit *run*. Pentru lansarea jocului va trebui să specificăm locația acestui fișier și numele fișierului. Dacă locația fișierului este deja specificată putem lansa jocul specificând numele fișierului astfel: `./run`.

Elemente specifice POO

Pentru crearea jocului am folosit programarea orientată pe obiecte în limbajul C++.

În continuare vor fi descrise elementele specifice POO al acestui proiect:

- a) Clasa “BoardHandler”: Această clasă este folosită pentru generarea terenului, crearea tabelului, gestionarea și afișarea tabelului sau a terenului.

```
1  #ifndef BOARDHANDLER_H
2  #define BOARDHANDLER_H
3
4  #define BLACK "\033[30m"
5  #define RED "\033[31m"
6  #define GREEN "\033[32m"
7  #define YELLOW "\033[33m"
8  #define BLUE "\033[34m"
9  #define MAGENTA "\033[35m"
10 #define CYAN "\033[36m"
11 #define WHITE "\033[37m"
12 #define RESET "\033[0m"
13
14 class BoardHandler
15 {
16
17 public:
18     char a[10][10];
19     char b[10][10];
20
21     BoardHandler();
22
23     void creare_teren(); //genereaza terenul de joc
24     void afisare_teren(); //afiseaza terenul generat
25     void resetare_tabel(); //reseteaza matricea tabelului
26     void afisare_tabel(); //afiseaza tabelul alterat
27     void vecinii(int i, int j); //metoda ce descopera casutele vecine ce au valoarea 0
28
29 };
30
31 #endif // BOARDHANDLER_H
32
```

Figura 1. Header-ul clasei BoardHandler

În această clasă avem 2 variabile, un constructor și alte 5 metode:

- **a**: reprezintă terenul generat
- **b**: reprezintă tabelul mască ce se afișează în consolă
- **BoardHandler()**: reprezintă constructorul clasei
- **creare_teren()**: generează terenul jocului
- **afisare_teren()**: afișează terenul generat la finalul jocului
- **resetare_tabel()**: resetează tabelul mască alterat la începutul jocului
- **afisare_tabel()**: este folosit pentru afișarea tabelului alterat
- **vecinii(int i, int j)**: această metodă descoperă căsuțele vecine ce au valoarea 0

b) Clasa “MineSweeper”: Această clasă este folosită pentru gestionarea întregului joc.

```
16 class Minesweeper
17 {
18 private:
19     int go; // 'go' de la gameover si poate avea valorile
20           // NEFINALIZAT/ PIERDUT/ CASTIGAT
21
22     int s; // corespunde starii STEP/FLAG
23
24 public:
25
26     BoardHandler board;
27
28     Minesweeper();
29
30     void afiseaza_stare(); // afiseaza starea curenta (STEP/FLAG)
31     void afiseaza_instructiuni(); // afiseaza instructiunile jocului
32
33     int convert_input(string input); // returneaza data introdusa
34     void gameover(); // verifica daca jocul a fost castigat
35
36     void procesare_input(int d, int s); // Modifica tabelul in functie de data introdusa
37     void schimba_stare(); // interchimba intre modurile STEP/FLAG
38     void reset_gameover_flag(); // Reseteaza starea jocului ca fiind NEFINALIZAT
39     //dura finalizarea lui.
40
41     void start(); // ruleaza jocul
42
43 };
```

Figura 2. Header-ul clasei MineSweeper

În această clasă avem 2 variabile, un obiect, un constructor, și alte 7 metode:

- **go**: variabilă ce ține evidența stării în care se află jocul (dacă jocul a fost pierdut, câștigat sau dacă încă continuă)
- **s**: ține evidența modului de alterare al tabelului (STEP sau FLAG)
- **board**: obiect ce reprezintă tabla jocului.
- **afiseaza_stare()**: afișează modul curent de alterare al tabelului (STEP sau FLAG)
- **afiseaza_instructiuni()**: afișează instrucțiunile jocului
- **convert_input()**: primește comanda introdusă sub formă de string și o convertește în coordonate sau în comandă de schimbare a modului STEP/FLAG
- **gameover()**: verifică dacă jocul a fost câștigat
- **procesare_input(int d, int s)**: modifică tabelul în funcție de data introdusă
- **schimba_stare()**: folosit pentru a interschimba între modurile STEP/FLAG
- **reset_gameover_flag()**: resetează starea jocului ca fiind nefinalizat
- **start()**: este utilizat pentru rularea jocului, inițializând și gestionând toate acțiunile necesare.

Programul construiește un obiect Minesweeper denumit joc iar apoi apelează metoda `joc.start()`

```
1  #include <iostream>
2  #include "Minesweeper.h"
3
4  using namespace std;
5
6  int main()
7  {
8      Minesweeper joc;
9
10     joc.start();
11
12     return 0;
13 }
14
```

Figura 3. Fișierul sursă main

Cazuri excepționale

În cazul în care jucătorul introduce o coordonată greșită se va afișa un mesaj de avertizare.



The screenshot shows a terminal window with the following content:

```
root@DESKTOP-TT6HSHO: /mnt/c/Users/Andrei/Desktop/MineSweeperOOP
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
- - - - -
a| + | + | + | + | 1 | 0 | 0 | 0 | 0 |
- - - - -
b| + | + | + | + | 1 | 0 | 0 | 0 | 0 |
- - - - -
c| + | + | + | + | 1 | 0 | 0 | 0 | 0 |
- - - - -
d| + | + | + | + | 1 | 0 | 0 | 0 | 0 |
- - - - -
e| + | + | + | + | 2 | 0 | 0 | 0 | 0 |
- - - - -
f| + | + | + | + | 2 | 0 | 0 | 0 | 0 |
- - - - -
g| + | + | + | + | 2 | 0 | 0 | 0 | 0 |
- - - - -
h| + | + | + | + | 4 | 2 | 1 | 0 | 0 |
- - - - -
i| + | + | + | + | + | + | 1 | 0 | 0 |
- - - - -

Ati introdus comanda gresita!
Introduceti comanda!
```

Figura 4. Rezultatul introducerii unei coordonate greșite

În cazul în care vrem să resetăm o căsuță marcată cu flag, va trebui să ieșim din modul FLAG în modul STEP și să introducem coordonata căsuței marcate cu flag.

```

root@DESKTOP-TT6HSHO: /mnt/c/Users/Andrei/Desktop/MineSweeperOOP
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
- - - - -
a| + | + | + | + | + | + | 1 | 0 | 0 |
- - - - -
b| + | + | + | 3 | 2 | 1 | 1 | 0 | 0 |
- - - - -
c| + | + | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
- - - - -
d| + | + | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
- - - - -
e| + | + | 1 | 0 | 0 | 1 | f | 2 | 1 |
- - - - -
f| + | 2 | 1 | 0 | 0 | 1 | 1 | 2 | + |
- - - - -
g| + | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
- - - - -
h| + | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
- - - - -
i| + | + | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
- - - - -

Modul: STEP

Introduceti comanda!
e7_

```

Figura 5. Tabel cu o căsuță marcată cu flag

În cazul în care pierdem jocul se va afișa terenul generat și mesajul corespunzător.

```

root@DESKTOP-TT6HSHO: /mnt/c/Users/Andrei/Desktop/MineSweeperOOP
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
- - - - -
a| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
- - - - -
b| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
- - - - -
c| 0 | 1 | x | 2 | 1 | 2 | x | 2 | 1 |
- - - - -
d| 0 | 1 | 1 | 2 | x | 2 | 2 | x | 1 |
- - - - -
e| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
- - - - -
f| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
- - - - -
g| x | 1 | 1 | 2 | 2 | 1 | 0 | 1 | x |
- - - - -
h| 1 | 1 | 1 | x | x | 2 | 1 | 2 | 1 |
- - - - -
i| 0 | 0 | 1 | 2 | 2 | 2 | x | 1 | 0 |
- - - - -

Ai pierdut! :(
Doriti sa jucati din nou? <y/n>

```

Figura 6. Rezultatul pierderii jocului

În cazul în care câștigăm jocul se va afișa terenul generat și mesajul corespunzător.

```
root@DESKTOP-TT6HSHO: /mnt/c/Users/Andrei/Desktop/MineSweeperOOP
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
- - - - -
a| 0 | 1 | x | 1 | 0 | 0 | 0 | 0 | 0 |
- - - - -
b| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
- - - - -
c| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | x |
- - - - -
d| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 2 |
- - - - -
e| 0 | 0 | 0 | 0 | 0 | 0 | 1 | x | 1 |
- - - - -
f| 0 | 0 | 0 | 0 | 1 | 1 | 2 | 2 | 2 |
- - - - -
g| 0 | 1 | 1 | 1 | 1 | x | 1 | 2 | x |
- - - - -
h| 0 | 2 | x | 2 | 1 | 2 | 2 | 3 | x |
- - - - -
i| 0 | 2 | x | 2 | 0 | 1 | x | 2 | 1 |
- - - - -

Ai castigat!!!! :O
Doriti sa jucati din nou? <y/n>
```

Figura 7. Rezultatul câștigării jocului

Link-ul GitHub al proiectului: <https://github.com/BivolAndrei/MineSweeper>