

Tema 3 PC – Minesweeper

Termen predare: 12 ianuarie 2014, ora 23:55

Ultima modificare: 17 decembrie 2013, 22:55

Introducere

Tema proiectului din anul universitar 2013 - 2014 presupune realizarea binecunoscutului joc **Minesweeper**¹ pe Linux, folosind biblioteca grafică numită **curses**.

Vă rugăm să consultați **cu atenție**, înainte de a începe rezolvarea temei, regulamentul și instrucțiunile de pe [pagina principală dedicată proiectului](#).

Puteți porni în rezolvarea temei de la [Arhiva de Pornire](#). Arhiva conține:

- un fișier **Makefile** cu ajutorul căruia veți putea să compilați tema, să compilați exemplele sau să împachetați arhiva înainte de a trimite tema. Trebuie doar să editați fișierul și să treceți datele voastre personale;
- fișierul **README**, pe care trebuie să îl completați înainte de a trimite tema;
- un director **example**, în care se află două exemple de folosire ale bibliotecii **curses**. Aceste exemple conțin, printre altele:
 - comentarii explicative pentru fiecare pas din exemplu;
 - instrucțiuni despre cum veți implementa citirea tastelor apăsate;
 - un exemplu concret de bune practici de Coding Style (urmăriți, vă rog, indentarea, spațierea, alinierea comentariilor etc.);
- un director **src**, în care se află un schelet de cod **minesweeper.c**, respectiv un fișier header **minesweeper.h**. Atâta timp cât tema voastră compilează cu comanda **make** și rulează cu comanda **make run**, nu sunteți obligați să porniți de la aceste fișiere. Puteți să le modificați în orice mod doriți sau să adăugați la director orice alte fișiere în care să scrieți module din rezolvare, dacă acest lucru vă ajută în rezolvarea mai clară a temei. Alegerea vă aparține în totalitate;
- un director **bin**, inițial gol, în care utilitarul **make** va pune executabilele create.

Cerințe

Dacă nu sunteți deja familiari cu regulile jocului, vă puteți forma o părere mai clară despre acestea consultând un exemplu online de Minesweeper [aici](#).

Cerința 0 [0 puncte]: Completarea datelor personale și familiarizarea cu conținutul arhivei

Descărcați arhiva de pornire și inspectați conținutul acesteia. Deschideți fișierul **Makefile** și modificați variabilele de mediu cu datele voastre personale, apoi salvați și rulați pe rând comenzile (toate comenzile trebuie rulate din același director cu fișierul **Makefile**):

1. **make**
2. **make run**

¹ [http://en.wikipedia.org/wiki/Minesweeper_\(video_game\)](http://en.wikipedia.org/wiki/Minesweeper_(video_game))

Prima comandă compilează scheletul de cod din directorul **src** într-un executabil numit **Minesweeper** pe care îl pune în directorul **bin**, iar a doua comandă lansează în execuție jocul.

Inspectați conținutul fișierelor din directorul **example** și apoi rulați comanda:

1. **make examples**

Această comandă va compila exemplele din directorul **example** în două executabile separate pe care le va pune în directorul **bin**. Le puteți lansa pe rând în execuție cu comenzile:

1. **./bin/exemplu1**

2. **./bin/exemplu2**

Detalii despre cum se folosește biblioteca **curses** pentru a rezolva restul temei se vor găsi:

- [În documentația oficială a bibliotecii](#)
- [Într-o scurtă introducere despre bibliotecă](#)
- [Într-un tutorial cu exemple de bază](#)
- [Într-o prezentare amănunțită a bibliotecii](#)

Cerința 1 [15 puncte]: Layout-ul ecranului de joc

La fel ca în majoritatea implementărilor de Minesweeper, jocul va începe prin afișarea unui „teren minat” și a anumitor informații legate de runda curentă. Mai exact, elementele punctate la această cerință vor fi:

- tabla de Minesweeper propriu-zisă, cu dimensiunea implicită de **9 × 9** celule;
- un panou de informații alături de aceasta, care să includă:
 - data și ora curentă;
 - de cât timp rulează **runda curentă**, în ore, minute și secunde (prima rundă începe odată cu aplicația);
 - numărul de secunde scurse **de la prima lovitură**;
 - numărul presupus de mine rămase: inițial, acesta este egal cu numărul total de mine (implicit **10**);

Mai multe detalii despre ultimele trei elemente vor fi oferite mai departe.

Amplasarea panoului de informații față de tablă, precum și ordinea elementelor din cadrul panoului, sunt la alegerea voastră.

Cerința 2 [20 de puncte]: Meniul principal

Pentru acordarea punctajului pentru această cerință, proiectul trebuie să conțină un meniu principal funcțional, care să poată fi accesat din modul de joc prin apăsarea tastei **Escape**. Acesta va conține cel puțin următoarele opțiuni:

- New Game
- Settings
- Quit

Navigarea în meniu se va face cu ajutorul tastelor direcționale (săgeți), iar opțiunea selectată în mod curent va fi marcată (cu un caracter în dreptul ei, cu un highlight pe scris sau în orice alt mod). Selectarea uneia dintre opțiuni se va face cu **Enter** sau **Space**.

Rezultatul opțiunii **New Game** va fi începerea directă a unei noi runde, ce presupune afișarea ecranului de joc cu o tablă nouă (niciun pătrățel nu a fost lovit) și resetarea informațiilor

din panou. Acesta va fi momentul începerii măsurării timpului corespunzător runde curente.

La selectarea opțiunii **Settings**, utilizatorului îi vor fi prezentate diferite opțiuni de configurare a jocului. Acestea trebuie să includă **particularizarea tablei** prin selecția a 3 parametri:

- numărul de linii ale tablei **r** ($1 \leq r \leq 30$);
- numărul de coloane ale tablei **c** ($1 \leq c \leq 30$);
- numărul de mine **b** ($1 \leq b < r * c$).

Aceste valori vor fi aplicate începând cu următoarea selecție a opțiunii **New Game**, însă nu este obligatoriu să fie memorate și după închiderea aplicației pentru reîncărcarea lor la următoarea pornire.

Opțiunea **Quit** va determina închiderea aplicației.

Cerința 3 [15 puncte]: Deplasarea în cadrul tablei și marcarea minelor

În orice moment din timpul unui joc, este necesar ca una dintre celulele tablei să fie selectată. Acest lucru va fi reprezentat prin una din următoarele metode:

- modificarea culorii;
- afișarea unui caracter în interiorul ei, în cazul în care o celulă se afișează pe cel puțin două coloane (încât să rămână în continuare spațiu și pentru afișarea unei eventuale cifre);
- afișarea liniei și a coloanei în panoul de informații.

Inițial, căsuța selectată va fi cea din **centrul** tablei (pentru dimensiuni pare, se poate alege oricare dintre căsuțele adiacente axei / axelor centrale).

Selecția trebuie să poată fi deplasată folosind săgețile sau introducând de la tastatură noile coordonate după apăsarea unei anumite taste (nu neapărat ambele variante).

În continuare, pentru detectarea minelor, jucătorul poate plasa „stegulețe” în celulele în care acesta **crede** că se află mine. În acest caz, vom spune că acele celule au fost **marcate**. O celulă selectată și nemarcată se poate marca prin apăsarea tastei **M**. În același mod, o celulă selectată care este deja marcată se poate **demarca**.

Celulele care sunt marcate la un momentdat vor fi reprezentate grafic în mod diferit (de exemplu, puteți afișa un anumit caracter în acestea sau le puteți schimba culoarea).

La fiecare marcaj / demarcaj, se va actualiza numărul presupus de mine rămase din panoul de informații. Valoarea acestuia este egală cu diferența dintre numărul de mine și numărul de pătrățele marcate. **Atenție: acesta poate fi și negativ !**

Cerința 4 [10 puncte]: Generarea tablei

La prima apăsare a tastei **X** în timpul unui joc, se vor întâmpla trei lucruri importante:

- startul cronometrului (punctat la cerința 1);
- generarea minelor în cadrul tablei;
- lovirea căsuței selectate (punctată la cerința următoare);

Numărul de mine **b** este implicit **10** (dacă nu a fost modificat în prealabil din opțiunea **Settings**). Pentru acordarea punctajului pentru această cerință, acestea trebuie să fie

poziționate pseudoaleatoriu **uniform** pe suprafața tablei, **exceptând pătrățelul selectat** (toate celulele trebuie să aibă aceeași probabilitate de a conține o mină, **cu excepția celei selectate, pentru care probabilitatea va fi 0**). Indicație: Puteți considera că funcția *rand()* generează numere uniform distribuite, chiar și după aplicarea operației modulo pentru restrângerea intervalului de valori.

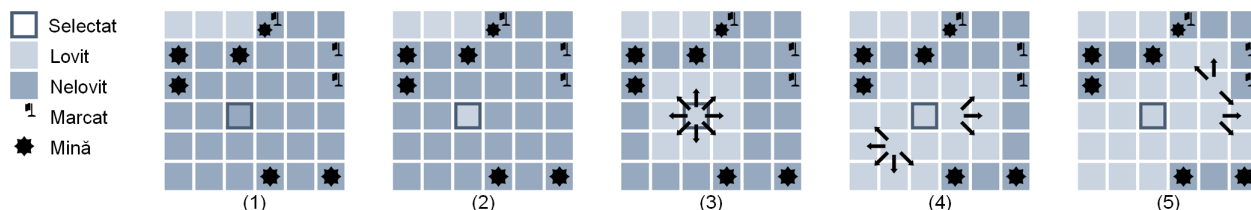
Într-o căsuță poate exista cel mult o mină.

Cerința 5 [20 puncte]: Lovirea căsuțelor

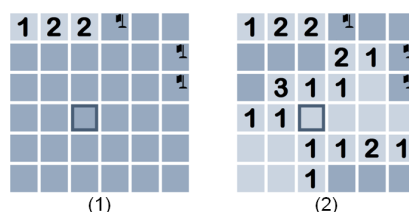
Dacă, la apăsarea tastei **x** în timpul unui joc, celula selectată nu este marcată, se consideră că aceasta a fost **lovită**. La prima lovire a unei celule, distingem trei situații:

1. Dacă în aceasta fusese repartizată o mină, jocul se termină. Terminarea jocului este descrisă amănunțit la cerința următoare.
2. Dacă în cel puțin una din căsuțele vecine se află o mină, în cea lovită se va afișa numărul de căsuțe vecine care conțin mine. Două căsuțe se numesc vecine dacă sunt distincte și diferențele ambelor coordonate sunt în mulțimea $\{-1, 0, 1\}$.
3. Dacă în niciuna dintre căsuțele vecine nu există mine, se vor bifa ca lovite și celulele nemarcate dintre acestea (într-adevăr, procedeul se va realiza în mod recurent, rezultând o „umplere” prin loviri - nu înseamnă că va trebui neapărat să implementați printr-un algoritm recursiv!).

Mai jos, v-am ilustrat un exemplu de lovire a unei căsuțe selectate:



Iată ce ar vedea utilizatorul:



Puteți implementa acest lucru prin iterații asupra întregii matrice până când nicio celulă nu a mai fost lovită sau în mod mai eficient, folosind algoritmul **Flood fill**². Ambele variante vor primi punctajul maxim corespunzător acestei cerințe.

Celulele lovite vor fi reprezentate cu altă culoare față de cele nelovite.

O celulă care a fost lovită nu va mai putea fi marcată.

Cerința 6 [10 puncte]: Terminarea jocului

Jocul se va termina cu insucces, în cazul în care a fost lovită o mină, sau cu succes, dacă toate pătrățelele nelovite ale tablei conțin mine.

² http://en.wikipedia.org/wiki/Flood_fill

Terminarea constă în:

- afișarea unui mesaj corespunzător;
- în cazul terminării cu insucces, afișarea poziționării minelor pe tablă;
- încetarea actualizării numărului de secunde de la prima lovire.

Cât timp nu se apasă tasta **Escape** pentru revenirea în meniul principal, atât ora / data, cât și timpul trecut de la începerea runde continuă să funcționeze.

Cerința 7 [10 puncte]: README, Coding Style

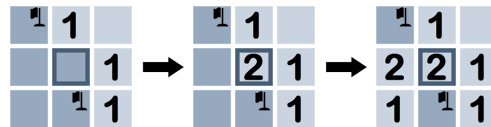
Cele **10** puncte (din **100**) se acordă astfel:

- **0p README:** Lipsa acestui fișier sau necompletarea lui se penalizează cu **10p** din punctajul obținut pe temă;
- **10p Coding Style:** Punctajul va fi acordat conform aprecierii celui care corectează tema.

Bonus 1 [15 puncte]: Lovire multiplă

Presupunem următorul scenariu:

După lovirea unei celule, jucătorul află că numărul de celule vecine care conțin mine este **N**, unde $N \in \{1, 2, \dots, 8\}$ și constată că acesta este egal cu numărul de celule vecine marcate. În cazul în care există celule vecine nelovite care nu sunt marcate, el poate, pornind de la premisa că marcajul efectuat până în acel moment este corect, să tragă concluzia că în niciuna dintre acestea nu se află mine.

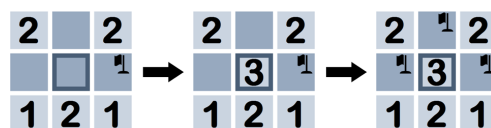


Cum aceste situații sunt foarte frecvente la Minesweeper, vi se cere să atribuiți un folos apăsării tastei **x** și pentru astfel de celule, rezultatul fiind lovirea celulelor vecine nemarcate. **Atenție:** tasta **x** se va apăsa în total de două ori pentru astfel de celule: o dată pentru lovirea simplă (în urma căreia se afișează **N**) și o dată pentru lovirea multiplă.

Evident, dacă jucătorul nu marcase corect celulele, atunci nici inferența nu va mai fi valabilă, deci este posibil ca o mină să fie lovită, iar jocul să se încheie.

Bonus 2 [15 puncte]: Marcare multiplă

Să considerăm acum că, după lovirea celulei și afișarea lui **N**, jucătorul ar fi observat că numărul de celule vecine nelovite este egal cu **N**. În această situație, se va putea trage concluzia (de data aceasta sigur) că toate acestea conțin mine, deci cele care nu erau deja marcate se pot marca acum, pentru a ușura jocul în continuare.



Pentru a primi punctajul pentru această cerință, trebuie să implementați acest

comportament la apăsarea tastei **M** pentru celulele care satisfac proprietatea menționată.

Alte bonusuri:

Pentru această temă, vor fi punctate orice alte feature-uri, proporțional cu originalitatea și calitatea implementării, dar punctajul total nu va depăși **150p**. Câteva exemple pot fi: Pause Game, animații, sunete.

Pregătirea arhivei de trimis

Mergeți în directorul care conține fișierul **Makefile**, după ce ați completat detaliile voastre personale în acest fișier și ați completat formularul din fișierul **README** și scrieți comanda:

make pack

Verificați că arhiva ***.zip** generată are numele corespunzător. Verificați că ea nu conține decât fișierele necesare rulării și notării temei (sursele **C**, **Makefile**, **README**, fișiere suplimentare de saved games, high scores etc.).

Înainte de a uploada arhiva pe site-ul de cursuri, despachetați-o în alt director și verificați din nou faptul că poate fi compilată fără probleme.

După ce v-ați convins că arhiva respectă cerințele impuse, uploadați-o pe site-ul de cursuri la pagina aferentă temei 3.

Ștefan-Gabriel Mirea