

Tema 2

Andrei Simescu, Răzvan Mihăescu, Alexandru Axenia

Coordonator: Mihai Nan

Data postării: 15.12.2023

Deadline: 12.01.2024 ora 23:59

1 Obiective

În urma realizării acestei teme, studentul va fi capabil:

- să folosească funcții din biblioteca grafică ***ncurses***;
- să implementeze un joc minimalist folosind limbajul **C**;
- să aloce dinamic memoria necesară stocării unor date;
- să descopere îmbunătățiri care pot fi aduse unui joc pentru a-l face mai interesant;
- să modularizeze codul prin funcții. Implementarea unei aplicații mai complexe, în cazul de față un joc cu o interfață grafică, necesită o atenție sporită la modularizarea codului.

2 Descriere

Tema 2 din anul universitar 2023 – 2024 presupune realizarea binecunoscutului joc **2048** pe Linux, folosind biblioteca grafică numită ***ncurses***.

Probabil toți ați auzit, până în acest moment, de celebrul joc **2048**, un joc single-player realizat de Gabriele Cirulli, fiind bazat pe **1024**, o versiune anterioară produsă de Veewo Studio.

Jocul are la bază o tabla de joc sub forma unei matrice pătratică cu 4 linii și 4 coloane, fiecare pătrat fiind liber sau conținând un număr întreg - o putere a lui 2. La începerea jocului, toate pătratele care compun tabla de joc sunt libere, cu excepția a două care pot conține una din valorile 2 sau 4, alegerea valorii fiind una aleatoare, precum și cea a pozițiilor în care sunt plasate aceste pătrate pe tabla de joc.

Pătratele din tabla de joc care conțin valori (nu sunt poziții libere) se pot deplasa în direcția impusă de jucător (nord, vest, sud sau est). În funcție de direcție, pătratele din tabla de joc care conțin valori (le vom numi, în continuare, pătrate / celule valide) se vor deplasa astfel:

- Se vor deplasa în ordinea în care se află pe tabla de joc spre direcția indicată , ocupând spațiile libere, până se vor întâlni cu alte pătrate valide sau cu marginea tablei de joc;

- Dacă două pătrate valide cu aceeași valoare se întâlnesc în timpul mutării, ele se vor uni într-un singur pătrat care va avea ca valoare suma valorilor pătratelor din care este compus (de două ori suma valorii unuia dintre aceste pătrate);

- Celula obținută prin alipire conform regulii de mai sus nu se poate uni cu o altă celulă de aceeași valoare în aceeași mutare. Au prioritate de alipire pătratele situate cât mai aproape de marginea indicată de direcția de deplasare.

În cazul în care nu se poate muta nicio celulă validă, prin alegerea unei anumite direcții, într-un anumit moment al jocului, mutarea respectivă este considerată invalidă.

După fiecare mutare validă, aplicația trebuie să adauge un nou pătrat, având valoarea 2 sau 4 și fiind poziționat într-o celulă liberă, alegerea poziției pe tabla de joc fiind una aleatoare. În cazul unei mutări invalide, aplicația așteaptă alegerea unei alte direcții impuse de jucător. Dacă nu mai există direcții care să genereze mutări valide, jocul se termină.

Dacă pe tabla de joc se generează în urma unei mutări o celulă cu valoarea 2048 atunci jocul este câștigat.

Pentru o înțelegere mai bună a acestor reguli de validare a unei mutări, se va prezenta o mutare validă. Săgeata de pe marginea imaginii de mai jos indică direcția care a fost selectată. Celula marcată cu * (în exemplul de mai jos **2***) este cea generată aleator după executarea mutării valide.

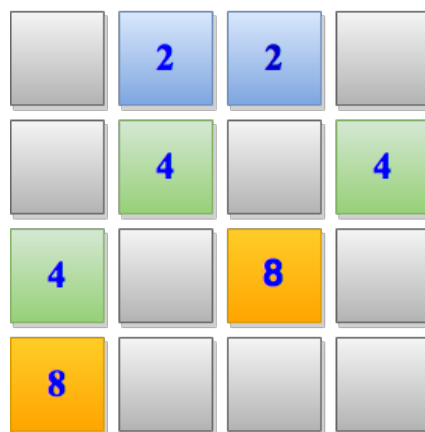


Figure 1: Starea 1

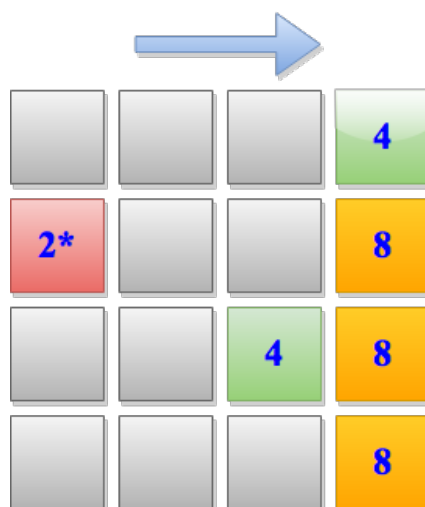


Figure 2: Starea 2 - Est

Dacă nu sunteți familiarizați cu regulile acestui joc, puteți parcurge descrierea de aici, iar pentru a înțelege mai bine acest joc puteți accesa varianta online de aici.

3 Cerințe

Se dorește implementarea unei aplicații care să permită unui utilizator uman să joace 2048, iar pentru acest lucru va trebui să realizați următoarele task-uri.

Observație

Ținând cont că nu vă este impus un format strict al aplicației, din punct de vedere grafic, veți primi puncte **bonus** pentru aspectul artistic al aplicației și pentru diverse îmbunătățiri sau idei originale pe care le aduceți.

Facilitățile adăugate vor fi punctate proporțional cu originalitatea și calitatea implementării, dar punctajul total acordat pentru această tema nu va depăși **150 de puncte**.

Puteți lua în calcul și următoarele exemple pentru puncte bonus: posibilitatea de a relua jocul după închiderea aplicației; buton UNDO pentru revenirea la pasul precedent; afișarea celor mai bune scoruri – menu high scores; folosirea de culori, meniuri; afișarea unor statistici, etc.

3.1 Cerința 1 - 20 puncte

Punctajul aferent acestei cerințe se va acorda pentru realizarea unui meniu principal din care utilizatorul aplicației să poată selecta și alege cel puțin următoarele opțiuni:

- **New Game**
- **Resume**
- **Quit**

Navigarea în meniu se va face cu ajutorul tastelor direcționale (săgețile sus / jos), iar opțiunea selectată în mod curent va fi marcată cu un caracter în dreptul ei sau cu orice alt efect considerat potrivit (de exemplu un highlight pe scris). Selectarea uneia dintre opțiuni se va face cu **Enter**.

Rezultatul opțiunii **New Game** va fi începerea directă a unei noi runde, ce presupune afișarea ecranului de joc cu o nouă tablă plus un panou de control, descrise la cerința 2. După începerea jocului se poate reveni la meniu apăsând tasta **Q**, iar, într-o astfel de situație, opțiunea **Resume** devine accesibilă. Astfel, selectarea acestei opțiuni va determina revenirea la ultimul joc început. Pentru a închide aplicația, se va selecta opțiunea **Quit**.

Pe măsură ce adăugați noi facilități aplicației voastre, puteți adăuga și alte opțiuni în meniu, specificându-le în fișierul **README**.

3.2 Cerința 2 - 25 de puncte

În cadrul acestei cerințe veți defini layout-ul ecranului de joc. Pentru acordarea punctajului aferent acestei cerințe, aplicația voastră trebuie să conțină tabla de joc propriu-zisă, având dimensiunea standard de 4x4 celule, 2 celule cu valori din mulțimea {2, 4} generate în poziții aleatoare, și un panou de control în care să fie afișate următoarele informații: data și ora curente, punctajul acumulat și o legendă a comenzilor valide. Punctajul este inițial 0.

3.3 Cerința 3 - 50 de puncte

Pentru a primi punctajul acordat la această cerință, trebuie să implementați jocul 2048 propriu-zis, descris în secțiunea 2. Jocul cuprinde:

35 de puncte

- Deplasarea / îmbinarea celulelor conform regulilor descrise;
- Generarea aleatoare a unei celule cu valoarea din mulțimea $\{2, 4\}$ în urma unei mutări valide;
- În cazul unei mutări invalide, aplicația va aștepta o comandă validă (în cazul în care jocul nu s-a terminat).

15 puncte

La fiecare operație de îmbinare trebuie actualizat și scorul jocului, afișat în panoul de control. La începutul jocului, scorul este 0, iar după o operație de îmbinare se va aduna la scor valoarea celulei rezultate.

Puncte bonus

Se recomandă să alegeți anumite culori pe care să le folosiți în evidențierea celulelor care au aceeași valoare, oferind, în acest mod, un aspect plăcut tablei de joc și facilitând observarea celulelor care pot fi îmbinate.

3.4 Cerința 4 - 20 de puncte

Implementați mișcarea automată a celulelor: dacă nu se apasă nicio tastă timp de o perioadă de timp, aplicația va decide *singură* direcția în care se deplasează – se va deplasa în direcția (dintre cele 4 direcții posibile) care eliberează cele mai multe din celulele jocului.

3.5 Cerința 5 - 10 puncte

Jocul se termină în momentul în care toate celulele sunt ocupate și nu mai există nicio mutare validă pe tabla de joc sau când s-a ajuns la un bloc cu valoarea **2048**. Astfel, jocul curent a fost pierdut și se va afișa un mesaj corespunzător. Pentru a obține punctajul acestui task, trebuie să implementați această facilitare a aplicației.

4 Precizări

Atenție!

Punctaj: 150 de puncte = 125 de puncte (cerințele 1-5) + 25 de puncte (Bonus)
Pentru obținerea bibliotecii **ncurses** (pe o distribuție de Linux bazată pe Debian):

```
apt-get install libncurses5-dev
```

Detalii despre biblioteca **ncurses** găsiți în următoarele materiale:

- documentația oficială a bibliotecii;
- o scurtă introducere în folosirea bibliotecii ncurses;

Rezolvarea va porni de la arhiva temei – aflată pe site-ul de cursuri.

Tema va fi rezolvată obligatoriu în limbajul C. **NU** folosiți elemente ale limbajului C++.

Veți urca pe site-ul de cursuri o arhivă zip **<grupa>_<nume>_<prenume>.zip** (de exemplu, **313CC_Popescu_Maria**) care va conține fișierele sursă, Makefile și README. Fișierele trebuie să se regăsească direct în rădăcina arhivei.

Precizați în README cerințele rezolvate și modul în care se interacționează cu aplicația voastră. Explicați, pe scurt, cum ați realizat implementarea cerințelor. Specificați, de asemenea, tot ce ați implementat ca bonus.

Temele care nu compilează la comanda `make` și care nu rulează la comanda ***make run*** nu vor fi punctate.

Temele sunt individuale. Copierea va fi sancționată – anularea punctajului temei pentru toți studenții implicați.

După deadline-ul temei, nu se mai pot încărca teme pe site și, ca urmare, temele respective nu vor fi punctate!

Modularizați pe cât posibil programul! Folosiți funcții auxiliare, dacă este cazul!

Tema se va rula manual și prezenta în cadrul ultimul laborator!

Corectorii își rezervă dreptul de a scădea puncte pentru orice problemă găsită în implementare, dacă vor considera acest lucru necesar.

5 Punctaj

Cerința	Punctaj
Cerința 1	20 puncte
Cerința 2	25 puncte
Cerința 3	50 de puncte
Cerința 4	20 de puncte
Cerința 5	10 puncte