Programarea Calculatoarelor - Seria CC

Tema 2

Publicarea enuntului: 09.12.2019

Data ultimei modificari a enuntului: 09.12.2019

Termen de predare: 10.01.2020, ora 23:55

Nu se accepta temele trimise dupa termenul de predare !

Responsabili tema: *Bogdan Nutu, Constantin Raducanu, Daniel Dinca, Ana Secuiu*

Profesor titular: *Carmen Odubasteanu*

Facultatea de Automatica si Calculatoare

Universitatea Politehnica din Bucuresti

Anul universitar 2019 – 2020

OBIECTIVE

In urma realizarii acestei teme, studentul va fi capabil:

- sa foloseasca functii din biblioteca grafica ncurses
- sa implementeze un joc minimalist folosind limbajul C
- sa aloce/dezaloce dinamic memoria necesara stocarii unor date
- sa descopere imbunatatirile care pot fi aduse unui joc pentru a-l face mai interesant
- sa modularizeze codul prin functii. Implementarea unei aplicatii mai complexe, in cazul de fata un joc cu o interfata grafica, necesita o atentie sporita la modularizarea codului.

DESCRIEREA PROBLEMEI

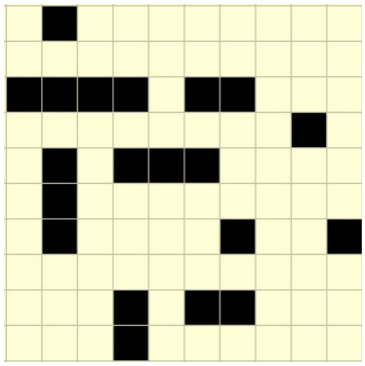
Tema proiectului din anul universitar 2019 – 2020 presupune realizarea binecunoscutului joc **Battleship** pe Linux, folosind biblioteca grafica ncurses.

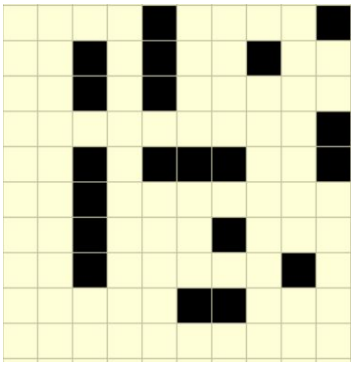
Scopul jocului **Battleship** este de a gasi navele adversarului pe harta si de a i le distruge in mod complet. In mod clasic, jocul presupune existenta a doi jucatori, iar fiecare jucator are un grid de dimensiune 10 x 10 pe care isi plaseaza propriile nave, dupa cum urmeaza:

- 1 nava de dimensiune 1 x 4
- 2 nave de dimensiune 1 x 3
- 3 nave de dimensiune 1 x 2
- 4 nave de dimensiune 1 x 1

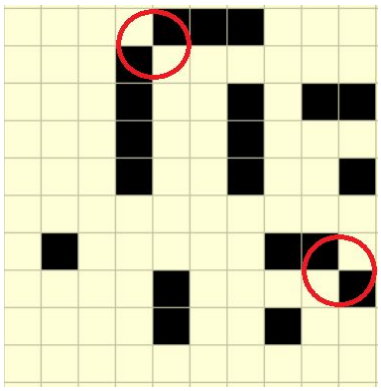
Harta fiecarui jucator este practic o matrice patratica, in interiorul careia se afla cele 10 nave, care pot fi pozitionate atat orizontal, cat si vertical. Singura restrictie care apare in pozitionarea acestora este ca 2 nave nu trebuie sa se invecineze, adica oricare ar fi un patrat din prima nava si altul din a doua nava, ele trebuie sa se afle la distanta de cel putin 1 pe orizontala, verticala si diagonala.

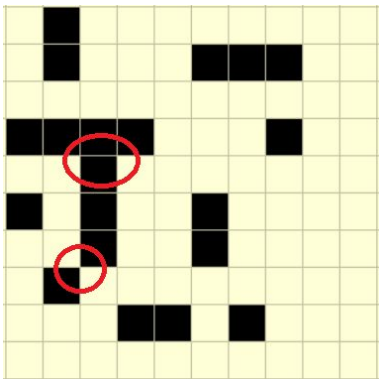
De exemplu, urmatoarele 2 configuratii de nave sunt valide:





In schimb, urmatoarele 2 configuratii **nu** sunt valide:





Pentru a intelege mai bine jocul, va recomandam:

- https://ro.unansea.com/cum-de-a-juca-battleship-reguli-de-joc/
- http://en.battleship-game.org/

CERINTE

Se doreste implementarea unei aplicatii care sa permita unui utilizator uman sa joace Battleship, iar pentru acest lucru va trebui sa realizati urmatoarele task-uri.

CERINTA 1 - 15 puncte

Punctajul aferent acestei cerinte se va acorda pentru realizarea unui meniu principal din care utilizatorul aplicatiei sa poate selecta si alege cel putin urmatoarele optiuni:

- ❖ New Game
- ❖ Resume Game
- ❖ Quit

Navigarea in meniu se va face cu ajutorul tastelor directionale(sagetile sus / jos), iar optiunea selectata in mod curent va fi marcata cu un caracter in dreptul ei sau cu orice alt efect considerat potrivit(de exemplu un highlight pe scris). Selectarea uneia dintre optiuni se va face cu ENTER.

Rezultatul primei optiuni (**New Game)** va fi inceperea directa a unei noi runde, ce presupune afisarea celor 2 grid-uri (a jucatorului si a computerului). Gridul computerului va fi afisat initial gol, pentru ca jucatorul este cel care va ghici unde sunt pozitionate navele, iar gridul jucatorului va contine navele acestuia. Dupa inceperea jocului se poate reveni la meniu apasand tasta **Q**, iar intr-o astfel de situatie optiunea **Resume** devine accesibila. Astfel, selectarea acestei optiuni va determina revenirea la ultimul joc inceput. Pentru a inchide aplicatia, se va selecta optiunea **Quit**.

Pe masura ce adaugati noi facilitati aplicatiei voastre, puteti adauga si alte optiuni in meniu, specificandu-le in fisierul README**.**

CERINTA 2 - 15 puncte

Cerinta 2 presupune generarea celor 2 configuratii valide de grid, pentru computer si pentru jucator.

In ceea ce priveste computerul, harta va fi generata aleator, dar tinand cont de restrictiile descrise mai sus.

In schimb, harta jucatorului curent va fi citita dintr-un fisier, al carui format il veti stabili pe cont propriu. Asadar, fiecare student isi va crea propria sa configuratie a navelor, pe care o va adauga in fisier dupa un format ales de el, iar apoi va face citirea configuratiei din fisier dupa formatul ales si o va afisa.

De exemplu, un model de fisier de intrare ar putea fi urmatorul:

```
1  |  |  |  |  |  |X|  |  |  |  |
2  |X|X|X|X|  |  |  |  |  |  |
3  |  |  |  |  |  |  |  |X|X|X|
4  |  |X|  |  |  |  |  |  |  |  |
5  |  |  |  |  |X|  |  |  |  |X|
6  |  |  |  |  |X|  |  |  |  |  |
7  |X|X|  |  |X|  |  |X|X|  |  |
8  |  |  |  |  |  |  |  |  |  |  |
9  |  |  |X|  |  |  |  |X|  |  |  |
10 |  |  |X|  |  |  |  |  |  |  |  |
```

,

unde X marcheaza existenta unei nave, iar ‘ ‘ marcheaza o celula goala.

CERINTA 3 - 35 puncte

Pentru a primi punctajul acordat la aceasta cerinta, trebuie sa implementati jocul battleship. Jocul cuprinde:

25 puncte

Posibilitatea jucatorului de a alege o celula din harta calculatorului. Puteti face acest lucru fie cu ajutorul tastelor directionale(sus, jos, stanga, dreapta), iar la final apasand ENTER, fie cu mouse-ul. Dupa selectarea optiunii, trebuie sa afisati daca jucatorul a nimerit o celula cu o nava sau o celula goala. Daca celula a fost o nava, atunci urmatorul jucator este tot jucatorul curent. Daca nu, urmeaza calculatorul.

10 puncte

Cand jucatorul care urmeaza la mutare este calculatorul, selectarea celulei din configuratia jucatorului va fi facuta random, din celulele care nu au fost alese inca.

Pentru a oferi o interfata intuitiva si pentru a fi vizibile toate mutarile efectuate de cei 2 jucatori, trebuie ca selectia celulei de catre calculator sa fie facuta la intervale de timp de 3 secunde.

In plus, si cand selecteaza calculatorului se pastreaza aceeasi idee ca mai sus. Daca pozitia aleasa este o nava, atunci tot calculatorul va face si urmatoarea alegere, dupa 3 secunde, cum am specificat mai sus.

CERINTA 4 - 20 puncte

Pentru acesta cerinta este necesar sa implementati 2 posibile optiuni care vor putea fi acesate de jucator in orice stagiu al jocului:

Randomize map: Daca aceasta optiune de joc va fi selectata atunci navele ramase ale jucatorului se vor repozitiona aleator. De exemplu, sa presupunem ca pana in acel moment al jocului, calculatorul a doborat complet 2 nave ale jucatorului, iar restul sunt doborate doar partial sau deloc. Cu ajutorul acestei optiuni, va trebui ca navele doborate partial sau deloc sa fie repozitionate pe harta jucatorului intr-un mod cat mai aleator. De precizat este faptul ca aceasta repozitionare se poate face si pe celulele in care initial nu s-a aflat nimic si care au fost alese de calculator. Asadar, toate celulele goale alese deja de calculator vor fi "deblocate" si navele vor fi repozitionate in aceste celule sau in celulele deja ramase goale pana in acel moment.

Destroy in advance: Daca aceasta optiune va fi selectata, atunci se vor alege 10 casute aleator atat din harta jucatorului, cat si din harta calculatorului si se vor elimina. Casutele alease aleator trebuie sa fie neparat neincercate. Altfel spus, prin aceasta optiune se merge in fata cu "10 pasi", este ca si cum fiecare din jucatori ar face 10 alegeri de celule in fata.

Modul in care se alege una din aceste optiuni va apartine, insa trebuie sa apara clar pe ecranul de joc. De exemplu, ati putea selecta una din optiuni apasand tastele R sau D.

CERINTA 5 - 5 puncte

Jocul se termina cand jucatorul/calculatorul a gasit toate navele adversarului. La final, dupa terminarea jocului, trebuie sa afisati pe ecran cine a castigat si numarul de nave doborate pentru fiecare jucator. Pentru a obtine punctajul aferent acestei cerinte, trebuie sa implementati aceasta facilitate a aplicatiei.

PRECIZARI

Aplicatia va porni prin rularea executabilului numit **battleship**.

- Se verifica daca programul a fost rulat cu argumente. In cazul in care nu s-au dat argumente programului, se va afisa la iesire sirul:
  - [Eroare]: Nu s-au dat argumente de comanda.
  - Dupa afisare se va iesi din program cu codul 1. (Functia main va returna valoarea 1, nu valoarea 0).

- Fiecare argument al programului reprezinta numele unui fisier, care contine o configuratie posibila de joc. Se vor citi toate aceste fisiere linie cu linie, dupa formatul pe care vi l-ati stabilit singuri, salvandu-se in memoria interna a programului fiecare configuratie. Apoi, pentru a alege o configuratie pentru jucatorul vostru veti alege una din hartile citite.

  IMPORTANT: Toate hartile pe care le cititi din fisier, dar si harta generata random pentru calculator trebuie sa respecte restrictia descrisa la inceputul temei.

- Daca vreun fisier nu poate fi deschis, se va afisa la iesire sirul:
  - [Eroare]: Fisierul %s nu poate fi deschis.
  - Dupa afisare se va iesi din program cu codul 1. (Functia main va returna valoarea 1, nu valoarea 0).

OBSERVATII

- Punctaj: 130 puncte = 90 puncte (cerintele 1- 5) + 10 puncte (README+ „Coding Style")+ 30 puncte (Bonus).Pentru obtinerea punctajului legat de scrierea codului, cititi sectiunea dedicata pe site-ul de laboratoare.
- Pentru obtinerea bibliotecii **ncurses** (pe o distributie de Linux bazata pe Debian, instalati cu: **apt-get install libncurses5-dev**)
- Detalii despre biblioteca ncurses gasesc in urmatoarele materiale:
  - documentatia oficiala a bibliotecii
  - o scurta introducere in folosirea bibliotecii ncurses
  - exemple
- Tema va fi rezolvata obligatoriu in limbajul C. Nu folositi elemente ale limbajului C++.
- Veti incarca pe site-ul de cursuri o arhiva zip <grupa>_<nume>_<prenume>.zip (de exemplu, 313CC_Popescu_Maria.zip) care va contine fisierele sursa, Makefile si README. Fisierele trebuie sa se regaseasca direct in radacina arhivei.
- Precizati in README cerintele rezolvate si modul in care se interactioneaza cu aplicatia voastra. Explicati, pe scurt, cum ati realizat implementarea cerintelor. Specificati, de asemenea, tot ce ati implementat ca bonus.
- Temele care nu compileaza la comanda **make** un executabil numit battleship **nu vor fi punctate**.
- Arhivele care nu sunt trimise sub formatul de mai sus nu vor fi punctate.
- Temele sunt **individuale**. Copierea va fi sanctionata –anularea punctajului temei pentru toti studentii implicati. Sursele copiate de pe Internet vor fi, de asemenea, anulate.

RESTRICTII

- Nu se vor folosi variabile globale.
- Memoria se va aloca dinamic.
- Eliberati memoria alocata dinamic. Folositi **valgrind –tool=memcheck –leak-check=full ./battleship** pentru a verifica daca memoria este eliberata corect.

BONUSURI

Se da liber creativitatii. Orice functionalitate adusa in plus, atat din punct de vedere grafic cat si al functionalitatii propriu-zise a jocului vor fi punctate (in functie de compexitate).

Cateva exemple pot fi:

- Mentinerea unui Score list care nu se pierde dupa inchiderea jocului.
- Realizarea unor ‘animatii’ prin culori care sa faca jocul cat mai placut vizual.
- Implementarea unor strategii pentru a imbunatati jocul calculatorului. O strategie interesanta de implementat pentru calculator ar fi urmatoarea: Atunci cand acesta nimereste o celula dintr-o nava sa le incerce pe cele de langa celula nimerita pana cand doboara toata nava.

Orice functionalitate implementata pentru punctajul bonus trebuie specificata si in README !