

Programe OO în C++



Obiective

Specificarea, proiectarea și implementarea problemelor complexe în C++.



Aspecte teoretice

Aplicații în C++.
STL, relația de moștenire, template, polimorfism



Termen de predare

Lab 12 (partea 1), respectiv Lab 13 (partea 2)



Cerințe

Predare în lab 12 (partea 1):

- jocul funcțional cu interfață consolă

Predare în lab 13:

- același joc dar cu posibilitatea alegerii interfeței (consolă sau GUI)

	Element	Punctaj
Partea 1	Implementare funcțională	
	- start joc	3
	- mutări alternative ale jucătorilor reflectate în tablele de joc	4
	- încheiere joc & anunțare câștigătorului	1
	- începerea unui nou joc	1
	- diagrama UML	1
Partea 2	Implementare funcțională & GUI	
	- joc cu GUI	2
	- alegerea între consolă și GUI	2
	- polimorfism la nivelul pieselor	2
	- polimorfism la nivelul citirii datelor (consolă sau fișier) pentru configurațiile inițiale ale jocului	2
	- validări & excepții	2

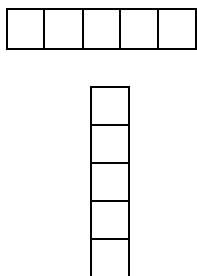
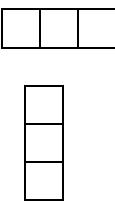
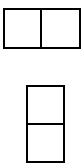
Dezvoltați o aplicație care să permită jucarea unui joc conform regulilor:

- se joacă de către 2 jucători (un utilizator și un calculator)
- se joacă pe o tablă matriceală matriceală de dimensiune $n \times m$, cu liniile numerotate cu litere, iar coloanele cu numere (n, m - numere naturale mai mici decât 32)
- fiecare jucător amplasează piesele pe tabla lui de joc; piesele se amplasează astfel încât să se încadreze pe tabla de joc și să nu se suprapună; piesele jucătorului uman se amplasează conform dorințelor acestuia (citite de la tastatură sau preluate dintr-un fișier); piesele jucătorului computer se amplasează la întâmplare la pornirea jocului.
- jucătorii efectuează, pe rând și alternativ, câte o mutare
- jocul se termină când toate piesele unui jucător au fost distruse sau identificate
- jocul este câștigat de jucătorul care a efectuat ultima mutare (care a distrus sau identificat complet toate piesele adversarului)

Observație: nu este nevoie de implementarea unei strategii de joc inteligente pentru jucătorul computer (este suficientă generarea aleatoare de mutări corecte).

1. Vaporășe

Fiecare jucător are propria lui tablă pe care amplasează k vase de război, p yacht-uri și q submarine. Fiecare piesă este formată din mai multe pătrățele, amplasate pe aceeași linie sau pe aceeași coloană. Un vas de război este format din 5 pătrățele, un yacht din 3 pătrățele, iar un submarin din 2 pătrățele.

		
Vase de război	Yacht-uri	Submarine

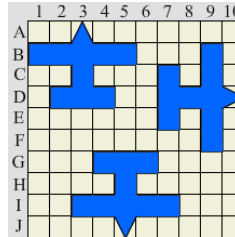
La o mutare, un jucător trebuie să ghicească poziția pieselor celuilalt jucător, comunicând poziția acestei căsuțe; drept răspuns celălalt jucător poate răspunde:

- *aer* - dacă acea poziție nu conținea nici o piesă
- *atins* - dacă pe acea poziție se află o parte dintr-o piesă

Scopul unui jucător este de a identifica poziția tuturor vaselor de pe tabla adversarului, deci jocul se termină când toate piesele unui jucător au fost identificate complet. Jocul este câștigat de jucătorul care a efectuat ultima mutare (care a identificat complet toate piesele adversarului).

2. Avioane

Fiecare jucător are propria lui tablă pe care amplasează k avioane. Fiecare piesă este formată din mai multe pătrățele sub forma unui avion (mai multe pătrățele identice și un pătrățel special numit panou de comandă al piesei curente).



Exemplu de tablă cu $k = 3$ avioane.

La o mutare, un jucător trebuie să ghicească poziția pieselor celuilalt jucător, comunicând poziția acestei căsuțe; drept răspuns celălalt jucător poate răspunde:

- *aer* - dacă acea poziție nu conținea nici o piesă
- *lovit* - dacă pe acea poziție se află o parte dintr-o piesă
- *distrus* - dacă pe acea poziție se află panoul de comandă al unei piese

Scopul unui jucător este de a identifica poziția tuturor vaselor de pe tabla adversarului, deci jocul se termină când toate piesele unui jucător au fost identificate complet. Jocul este câștigat de jucătorul care a efectuat ultima mutare (care a identificat complet toate piesele adversarului).