



## Responsabili Proiect

-  Ștefan Rușeți
-  Dragoș Corlătescu
-  Ionuț Bîrsu
-  Andrei Dumitrescu
-  Ianis Bacula
-  Marius Vintilă

## Regulament proiect 2020

Tema proiectului de anul acesta din cadrul materiei Proiectarea Algoritmilor va fi realizarea unei inteligențe artificiale capabile să joace șah[1]. Proiectul se va realiza în echipe de câte 2-3 studenți (minim 2, maxim 3 studenți pe echipă) și se va desfășura pe parcursul a 4 etape. Se va folosi platforma  XBoard 4.8 pentru a interfața, vizualiza, înregistra și reda partidele jucate între aplicațiile realizate în cadrul proiectului. Dacă vreți să testați pe Windows, puteți folosi varianta WinBoard, pe care o puteți descărca de  aici.



**In cazuri deosebite** acceptam ca proiectul sa fie realizat individual. Pentru asta, va trebui sa trimiteti un mail echipei de proiect in care sa motivati decizia. **Nu recomandam acest lucru**, avand in vedere ca volumul de lucru este gandit pentru 2-3 studenti!

 Search

### Diverse

- [Hall of PA](#)
- [Regulamente PA 2020](#)
- [Proiect](#)
- [Catalog](#)
- [Test practic](#)
- [\[TEME\] Configuratie vmchecker](#)
- [\[skel\\_graph\] Precizari laboratoare 07-12](#)

### Laboratoare

- [00: Introducere și Relaxare](#)
  -  skel-lab00.zip
- [01: Divide et Impera](#)
  -  skel-lab01.zip
- [02: Greedy](#)
  -  skel-lab02.zip
- [03: Programare Dinamică 1](#)
  -  skel-lab03.zip

**Reguli de upload:** La fiecare etapă (în afară de etapa 0) veți salva în repository-ul vostru (cel specificat la înscriere) o arhivă zip, numită "EtapaX.zip" (unde X reprezintă numărul etapei: e.g. "Etapa1.zip")

Arhiva va conține:

- Codul sursă (**toate sursele folosite**)
- Makefile (sau CMakeLists.txt)
- README

Pentru notare, **toate arhivele vor fi descărcate automat** - asigurați-vă că arhiva există în repository, este numită corect și este completă.

#### Makefile:










Boții vor fi evaluați pe un sistem Linux/GNU pe 64 biți, deci **este foarte important să verificați că aveți un makefile care funcționează corect** pe Linux, chiar dacă lucrați pe Windows sau MacOS. Scriptul de testare va apela "cmake ." (dacă ați trimis un fișier CMakeLists.txt), "make build" pentru compilare, apoi "make run" pentru rulare.

Asigurați-vă că fișierul Makefile nu conține path-uri absolute (să poată fi rulat și pe alt sistem). Este util să avți o regula "clean", pentru a șterge executabilul și fișierele obiect/log-urile generate (dacă este cazul).

Fișierul **Readme** va respecta următoarea structură:

- Instrucțiuni de compilare.
- Detalii despre structura proiectului.
- Detalii despre abordarea algoritmică a etapei: ce algoritmi ați folosit, cum i-ați combinat, de ce, complexități, etc.
- Surse de inspirație
- Responsabilitatea fiecărui membru al echipei.
- Documentația valorează 10% din punctajul proiectului.

Atenție! **Deadline-urile la proiect sunt hard**, deci nu se permit întârzieri. Dacă s-a ratat deadline-ul la o etapă, se poate lucra în continuare pentru celelalte etape, dar punctajul se va pierde pentru respectiva etapă.

- **04: Programare Dinamică 2**
  -  skel-lab04.zip
- **05: Backtracking**
  -  skel-lab05.zip
- **06: Minimax**
  -  skel-lab06.zip
- **07: Parcurgerea Grafurilor. Sortare Topologică**
  -  skel-lab07.zip
- **08: Aplicații DFS**
  -  skel-lab08.zip
- **09: Drumuri minime**
  -  skel-lab09.zip
- **10: Arbori minimi de acoperire**
  -  skel-lab10.zip
- **11: Flux Maxim**
  -  skel-lab11.zip
- **12: A\***
  -  skel-lab12.zip

#### Materiale Suplimentare Test Practic

- Articole
- Probleme

#### Crash-course Opțional

- Debugging și Structuri de Date

## Etapa 0

Etapa 0 presupune formarea unor echipe de câte 2-3 persoane. Echipele se pot forma între oricare 2-3 studenți din aceeași serie. Studenții restanțieri pot participa cu colegi de la orice serie.

Fiecare echipă va desemna un membru al acesteia cu rolul de căpitan. Căpitanul va avea, printre altele, sarcina de a trimite în numele echipei sale soluțiile pentru fiecare din etapele rămase. Ceilalți membri ai echipei nu vor trimite soluții, decât în cazul în care căpitanul nu poate face acest lucru din motive obiective.

Pentru a ajuta colaborarea în echipă, veți folosi un sistem de versionare a surselor numit **git**. Mai multe indicații [aici](#).

Pentru înscriere, accesați acest [link](#).

Această etapă va avea **1 săptămână**. Punctajul alocat acestei etape este de 0 puncte.

Deadline-ul acestei etape este **duminică, 1 martie 2020, ora 23:55**.

## Etapa 1


Se va realiza reprezentarea internă a tablei de joc și a pieselor de joc, precum și o interfațare cu programul XBoard.

Interfațarea va presupune posibilitatea de a interpreta și interacționa cu următoarele comenzi ale XBoard: xboard, new, force, go, white, black, quit, resign, move.

De asemenea, programul va trebui să primească mișcări de la XBoard și să trimită mișcări legale. Nu este nevoie pentru această etapă să trimită orice mișcare legală posibilă.

Este suficientă mișcarea legală a unei singure piese (ex: pion, cal etc). Dacă programul nu mai găsește mișcări valide, va putea ieși din joc (prin resign sau orice altă metodă; de preferat una elegantă)

Această etapă va avea **2 săptămâni**. Punctajul alocat acestei etape este de **0.4 puncte**.

-  Schelet: Debugging și Structuri de Date

### Table of Contents

- Responsabili Proiect
- Regulament proiect 2020
  - Etapa 0
  - Etapa 1
  - Etapa 2
    - Testare
    - Punctare
  - Etapa 3
    - Testare
    - Punctare
  - Etapa 4
  - FAQs
    - De ce șah?
    - De ce Xboard? Nu pierdem timp stând să implementăm nu știu ce protocol în loc să implementăm algoritmi?
    - Nu exista deja o droaie de algoritmi pe net

Deadline-ul acestei etape este **17 martie 2020, 23:55**.

## Etapa 2

Programul va trebui sa implementeze macar un algoritm de bază minimax pentru a putea juca împotriva unui adversar. Programul va trebui să poată interpreta orice mișcare legală primită de la adversar, ceea ce presupune implementarea tuturor regulilor jocului.

Această etapă va avea **4 săptămâni**. Punctajul alocat acestei etape este de **0.5 puncte**.

Deadline-ul acestei etape este **21 aprilie 2020, ora 23:55**.

## Testare

Pentru testare ne vom folosi de engine-ul FairyMax care vine preinstalat cu xboard/winboard. Acesta va fi limitat să vadă doar 2 mutări în față (depth=2), deci nu va juca foarte bine.

Comanda care va fi rulată pentru testare va arăta (foarte) asemănător cu comanda de mai jos:



```
xboard -fcp "make run" -scp "fairymax" -secondInitString  
"new\nrandom\nsd 2\n" -tc 5 -inc 2 -autoCallFlag true -mg 10 -sgf  
partide.txt -reuseFirst false
```

Explicarea parametrilor:

- **xboard** - aplicația pe care o folosim 😊 poate fi atât xboard cât și winboard, dar checkerul va rula cel mai probabil pe linux deci vom folosi xboard.
- **-fcp "make run"** - setează primul engine ca fiind engine-ul vostru. Vă reamintim că "make run" ar trebui doar să vă pornească engine-ul, nu și xboard-ul.

despre  
sah? Dar  
proiectele  
din anii  
precedenți?

- Cum imi  
aleg  
coechipierii?
- Nu toți  
membrii  
echipei  
lucrează la  
fel de  
mult.  
What's up  
with that?
- In ce  
limbaj de  
programare  
avem voie  
sa  
realizam  
proiectul?
- Avem voie  
sa folosim  
fire de  
executie  
(thread-  
uri) in  
program?
- Avem voie  
să folosim  
baze de  
date  
pentru

- **-scp "fairymax"** - setează al doilea engine ca fiind engine-ul FairyMax.
- **-secondInitString "new\nrandom\nsd 2\n"** - string de inițializare pentru FairyMax care îi limitează adâncimea de căutare (depth) la 2.
- **-tc 5 -inc 2** - setează timpul de joc la 5 minute (timpul de start pentru fiecare jucător), iar la fiecare mutare jucătorul care a efectuat-o va primi încă 2 secunde. *Observație:* Nu este neapărat nevoie să implementați gestionarea timpului (deși probabil v-ar ajuta). Acest timp, împreună cu următorul parametru (-autoCallFlag true) previne situația în care voi ați da un depth prea mare engine-ului vostru și acesta ar sta foarte mult să se gândească și situația în care engine-ul vostru intră într-o buclă infinită și nu mai mută.
- **-autoCallFlag true** - setează xboard-ul ca atunci când i se termină timpul unuia din jucători să atribuie victoria celui alt.
- **-mg 10** - numărul de partide jucate va fi 10 pentru teste. Automat se vor juca 5 cu albul și 5 cu negru. Voi puteți modifica parametrul sau chiar să îl scoateți pentru teste mai mici.
- **-sgf partide.txt** - în acest fișier se vor salva partidele jucate.
- **-reuseFirst false** - acest parametru îi specifică lui xboard să reîncarce engine-ul vostru după fiecare partidă. Dacă nu este prezent se va da "new game". Puteți să-l vedeți ca pe o plasă de siguranță: dacă cumva engine-ul vostru se "strică" în timpul unei partide, el se va reseta la următoarea deci teoretic eroarea nu se va propaga.



Pentru fi siguri că primiți de la xboard/winboard mutările în formatul de coordonate (g1f3, care mută piesa de la g1 la f3) în momentul în care primiți "protover" de la xboard răspundeți cu "feature san=0". Dacă ați implementat mutările în formatul "piesă la câmp" (ex: Nf3, care mută calul la f3) atunci nu mai este nevoie să trimiteți acest "feature".

"feature"-ul puteți să-l trimiteți pe tot odată: foarte mulți dintre voi ați implementat în etapa 1 să răspundeți cu "feature sigterm=0". Acum puteți trimite "feature sigterm=0 san=0"

deschidere  
sau alte  
scopuri?

- Putem comunica cu un calculator la distanță prin TCP/IP sau alte mijloace pentru a avea acces la o bază de date suplimentar sau alte resurse?
- Putem trimite programe executabile?
- Unde putem cere lamuriri suplimentar în legatura cu desfasurare proiectului?

- Referințe

Mai multe detalii pentru configurarea xboard-ului din linie de comandă puteți găsi aici [documentație xboard](#).

## Punctare

Punctajul de **0.5 puncte** se va acorda în felul următor:

- **0.2 puncte** implementarea corectă a tuturor detaliilor ce țin de jocul de șah
- **0.3 puncte** implementarea algoritmului minimax/negamax. Acest punctaj va fi și el divizat după cum urmează:
  - *0.2 puncte* implementarea unui algoritm minimax/negamax mediocru
  - *0.1 puncte* implementarea unui algoritm minimax/negamax bun

Punctele se vor acorda după următoarea formulă:

```
punctaj_etapă = 0.2 * (număr_de_partide_terminate_corect / 10) +  
                0.2 * (număr_de_partide_terminate_după_mutarea_20 / 10)  
                min(0.1, 0.1 * (număr_de_puncte_din_meci / 5))
```

Clarificări:

- număr\_de\_partide\_terminate\_corect - număr de partide în care engine-ul vostru nu face illegal moves și termină partida. Evident că vom scădea dacă nu sunt implementate toate mișcările tuturor pieselor și așa mai departe. Adică nu veți primi punctaj dacă veți muta doar un pion și avoi veți ceda ca în etapa 1.
- număr\_de\_partide\_terminate\_după\_mutarea\_20 - numărul de partide în care partida se termină după mutarea 20 (mutarea 20 înseamnă că atât albul, cât și negrul au făcut 20 de mutări) cu orice rezultat **sau** dacă partida se termină înainte de mutarea 20, dar cu victoria voastră. *Observație:* În mod normal, un minimax cât de slab cu un depth > 2 ar trebui să îi reziste lui FairyMax mai mult de 20 de mutari.
- min(0.1, 0.1 \* (număr\_de\_puncte\_din\_meci / 5)) - pe scurt, pentru a obține toate punctele la această secțiune va trebui să obțineți minim 5 puncte din

cele 10 posibile în orice combinație (adică 5 victorii, sau 4 victorii și 2 remize, etc.)

## Etapă 3

Această etapă este dedicată implementării unor algoritmi avansați pentru a crește calitatea jocului prestat de program. Punctajul pe această etapă poate fi obținut în urma unor meciuri cu boți puși la dispoziție de către noi. Recomandăm implementarea algoritmului alpha-beta și a cel puțin două tehnici avansate/euristici pentru a putea calcula cât mai multe mutări în față. Administrarea timpului rămas este o componentă foarte importantă acestei etape.

Detalii despre boții puși la dispoziție, regulile de punctare și formatul meciurilor vor fi anunțate ulterior. Această etapă va avea **4 săptămâni**. Punctajul alocat acestei etape este de **0.5 puncte**.

Deadline-ul acestei etape este **19 mai 2020, ora 23:55**.

## Testare

Pentru testare ne vom folosi de engine-ul FairyMax care vine preinstalat cu xboard/winboard. În funcție de caz, acesta va fi limitat ori la depth=3 ori la depth=4. Acest lucru îl va face să joace mai bine decât la etapa precedentă.

Comenzile care vor fi rulate pentru testare vor arăta (foarte) asemănător cu comenzile de mai jos. Diferența constantă în string-ul de inițializare pentru FairyMax (depth 3 vs depth 4):



```
xboard -fcp "make run" -scp "fairymax" -secondInitString  
"new\nrandom\nsd 3\n" -tc 5 -inc 2 -autoCallFlag true -mg 4 -sgf  
partide.txt -reuseFirst false
```



```
xboard -fcp "make run" -scp "fairymax" -secondInitString  
"new\nrandom\nsd 4\n" -tc 5 -inc 2 -autoCallFlag true -mg 4 -sgf  
partide.txt -reuseFirst false
```

Pentru detalii legate de parametrii comenzilor analizați instrucțiunile de la etapa precedentă.

## Punctare

Punctajul de **0.5 puncte** se va acorda în felul următor:

- **0.35 puncte** obținerea a 2 puncte (orice combinație de remize + victorii) din 4 partide vs FairyMax cu depth-ul 3
- **0.15 puncte** obținerea a 2 puncte (orice combinație de remize + victorii) din 4 partide vs FairyMax cu depth-ul 4

Punctele se vor acorda după următoarea formulă:

$$\text{punctaj\_etapă} = \min(0.35, 0.35 * (\text{număr\_puncte\_din\_meci\_vs\_fm3} / 2)) + \min(0.15, 0.15 * (\text{număr\_puncte\_din\_meci\_vs\_fm4} / 2))$$

## Etape 4

În cadrul acestei etape se va realiza un concurs între proiectele realizate. În funcție de numărul de proiecte finalizate se va stabili formatul concursului: campionat, tournament tree sau grupe+tournament tree.

Punctajul aferent acestei etape este de **0.6 puncte** și se va acorda după următorul algoritm:

- Locul 1 va primi 0.6 puncte
- Locul 2 va primi 0.5 puncte
- Locul 3 va primi 0.4 puncte



- Restul proiectelor vor primi în funcție de locul în clasament între 0 și 0.35 puncte, marja între două proiecte consecutive fiind de  $0.35/(\text{numar\_proiecte\_finaliste}-3)$

## FAQs

---

### De ce șah?

Didactic, este o idee de proiect bună din mai multe puncte de vedere (provocări de implementare, înțelegerea unor concepte de algoritmi avansați, provocări de time management, abilitatea de autodocumentare, abilitatea de a lucra în echipă, etc). Șahul a fost ales datorită rezonanței sale ca joc. Din experiență vă spunem că un motor de șah decent făcut dă bine la un interviu.

### De ce Xboard? Nu pierdem timp stând să implementăm nu știu ce protocol în loc să implementăm algoritmi?

Winboard/Xboard a devenit un program standard de interfațare între diferitele motoare existente în momentul de față. Fiecare motor minimalist existent știe să lucreze cu acest program. Ușurarea lucrului de mediere a unui joc justifică implementarea sa, precum și faptul că proiectul realizat va putea juca cu orice alt motor cât de cât performant de pe net. Cât despre timpul pierdut, interfațarea minimalistă cerută în etapa 1 se poate realiza în aproximativ o oră. La ce chestii vor fi de implementat la protocoale de comunicații, asta vi se va părea floare la ureche 😊.

### Nu exista deja o droaie de algoritmi pe net despre șah? Dar proiectele din anii precedenți?

Da, exista multe surse de documentație pe net despre cum să implementezi un joc de șah. Dar de la vorbe la fapte e cale lungă. Doar pentru că există mulți algoritmi nu înseamnă că oricine va ști să-i implementeze și să o facă și bine. Implementarea corectă și eficientă presupune mai mult decât copy paste de pe net. Cât despre proiectele de anii trecuți, ele există în posesia echipei de PA, și cele de anul acesta vor fi verificate contra celor precedente pentru a nu avea

cazuri de fraudă. De asemenea, codul fiecarui proiect va fi făcut public înainte de etapa 4 pentru a putea fi inspectat de fiecare echipă ce dorește să se asigure că nu a fost comis vreun act de fraudă din partea concurenței. Dacă se constată o tentativă de fraudă din partea oricărei echipe, toți membrii echipei vor avea restanță la Proiectarea Algoritmilor și vor pierde punctajul aferent proiectului.

## Cum imi aleg coechipierii?

Cum vrei, pe baza oricăror criterii crezi tu că sunt importante.

## Nu toți membrii echipei lucrează la fel de mult. What's up with that?

Din experiența anilor trecuți, niciodată nu se întâmplă ca toți membrii echipei să lucreze la fel de mult. Acest fapt se întâmplă din varii motive: fie nu toți pot, fie unul sau doi acaparează toată munca și nu le dau celorlalți o șansă (nu râdeți, s-a întâmplat :P), fie <insert another reason here>. Fiecare echipă va avea un căpitan care va avea, printre altele, datoria de a comunica echipei de PA (asistentului de laborator, titularului de curs etc) orice neregulă din cadrul propriei echipe. În afară de asta și de a vorbi puțin individual cu membrii fiecărei echipe nu există altă soluție. Depinde de prioritățile fiecărui membru de echipă și a echipei per ansamblu.

## În ce limbaj de programare avem voie să realizăm proiectul?

În oricare limbaj de programare cu care vă înțelegeți mai bine, atât timp cât va putea rula pe linux. De remarcat este faptul că pentru acest proiect timpul de execuție este critic.

## Avem voie să folosim fire de execuție (thread-uri) în program?

Nu. S-a pus problema și în alți ani. Motivul este că nu toți știu să lucreze cu thread-uri, iar un astfel de avantaj ar fi incorect față de ceilalți. NU postați în

legătură cu asta pe forum...it's final and that's that.

**Avem voie să folosim baze de date pentru deschidere sau alte scopuri?**

Da, atât timp cât toata arhiva este până în 5Mb.

**Putem comunica cu un calculator la distanță prin TCP/IP sau alte mijloace pentru a avea acces la o bază de date suplimentară sau alte resurse?**

Categoric NU. Proiectul trebuie să fie stand-alone. Nu se admit conexiuni de genul celor menționate mai sus.

**Putem trimite programe executabile?**

Nu. Doar cod sursă (și eventuale adiționale precum opening data bases). Codul se va compila și link-edita după regulile prezente în makefile-ul, ant file-ul, script-ul etc-ul vostru de build.

**Unde putem cere lamuriri suplimentare în legatura cu desfasurarea proiectului?**

Pe forum-ul dedicat proiectului de pe pagina web a cursului de PA.

## Referințe

[1] -  [http://en.wikipedia.org/wiki/Rules\\_of\\_chess](http://en.wikipedia.org/wiki/Rules_of_chess)

pa/regulament-proiect-2020.txt · Last modified: 2020/05/01 14:42 by dragos.corlatescu

 Old revisions

 Media Manager  Back to top

