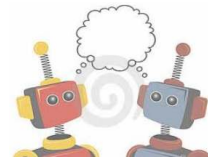


Proiectarea Algoritmilor

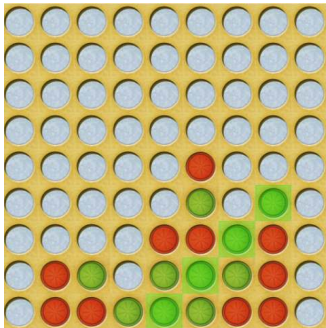
Tema 2 – Connect4



Descrierea problemei

Ionel și Dorel, doi studenți cu mult timp liber, sunt pasionați de jocurile pe calculator. Atunci când au învățat algoritmul Minimax, s-au gândit că îl pot folosi pentru a implementa un program care să joace singur StarCraft II... După câteva ore de de gândire asiduă în care nu au avut prea multe idei, cei doi decid să înceapă cu ceva mai puțin ambițios, pentru a testa dacă algoritmul într-adevar funcționează și dacă ei pot să îl implementeze corect. Astfel, au decis să scrie un program care să joace Connect4. Competitivi din fire, s-au gândit să provoace toți studenții care studiază Proiectarea Algoritmilor să implementeze ceva similar.

Connect4 este un joc cu doi jucători care mută alternativ. Fiecare jucător, atunci când îi vine rândul, adaugă jetonul de culoare proprie la una din coloanele unui grid format din 9 linii și 9



Jucătorul cu jetoanele verzi a câștigat. În figură există 4 astfel de jetoane conectate pe diagonală.

coloane. Jetonul este adăugat de sus și este lăsat să cadă până la prima căsuță neocupată pe acea coloană. Evident, nu se poate adăuga un jeton la o coloană care este plină. Inițial, gridul nu conține niciun jeton. Va câștiga jucătorul care reușește primul să conecteze 4 jetoane proprii pe orizontală, verticală sau diagonală. Jetoanele sunt conectate pe orizontală dacă sunt pe aceeași linie și pe coloane consecutive, pe verticală dacă sunt pe aceeași coloană și pe linii consecutive, și pe diagonală dacă sunt pe linii și pe coloane consecutive.

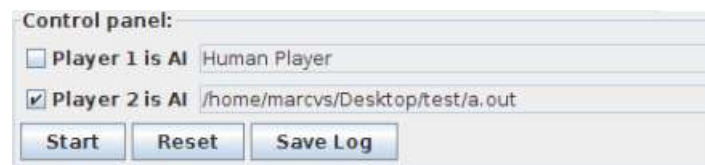
Să se implementeze un program care să joace Connect4 cât mai bine. Ionel și Dorel au terminat deja această sarcină, iar programul lor o să joace cu programul fiecărui student în parte, prin intermediul unei platforme de joc ce poate fi descărcată de pe site-ul de curs (detalii în secțiunea următoare). Singura condiție impusă este să se folosească un algoritm de explorare a arborelui de joc, precum Minimax sau variantele sale (Negamax, Alpha-Beta). Evident, pentru ca programul să joace cât mai bine într-un timp cât mai scurt, va fi necesară o abordare de tip Alpha-Beta.

Interacțiunea cu platforma de joc

Doi jucători vor putea concura prin intermediul unei platforme de joc pe care o puteți găsi în arhiva de pe site-ul de curs (cs.curs.pub.ro), din secțiunea temei 2. De asemenea, în arhivă veți găsi programul care joacă Connect4 scris de Ionel și Dorel. Programul este prezent în două versiuni: *opponent4.exe* pentru sistemele Windows, și respectiv *opponent4* pentru sistemele Linux.

Temele vor fi evaluate doar în Linux. Checkerul a fost scris însă cross-platform și veți putea implementa și testa în ce sistem de operare doriți. Vă rugăm însă în mod insistent să vă asigurați că soluția voastră funcționează pe Linux înainte de a o trimite!

Platforma de joc permite atât interacțiunea dintre jucători umani, cât și interacțiunea dintre programe. În figura de mai jos aveți un exemplu în care primul jucător este uman, iar cel de-al doilea este reprezentat de un program al cărui executabil este situat la calea precizată. Programele care joacă prin intermediul platformei trebuie să respecte un protocol fix, descris în secțiunea următoare. După selectarea jucătorilor trebuie apăsat butonul *Start*.



Interfața grafică rulează pe orice platformă unde aveți instalată o mașină virtuală Java. Recomandăm instalarea unei versiuni recente a mașinii virtuale. Puteți descărca gratuit software Java direct de pe site-ul oficial (<http://www.java.com/en/download/index.jsp>). În Linux puteți instala pachetul **sun-java6** pentru distribuții Ubuntu/Debian, sau un pachet similar pentru alte distribuții.



Două programe care concurează prin intermediul platformei. Jucătorul roșu câștigă.

Protocolul folosit pentru interacțiunea cu platforma

Interacțiunea cu platforma se va realiza prin intermediul intrării și ieșirii standard (*stdin*, respectiv *stdout*). Astfel, un program care joacă Connect4 va afișa mutarea pe care dorește să o efectueze la ecran, și va citi mutarea efectuată de oponent de la tastatură. După orice afișare se va tipări și sfârșitul de linie și **se va face obligatoriu flush la stream-ul de ieșire**. De exemplu, dacă doriți să efectuați o mutare în coloana 4, în C va trebui să folosiți comenzile:

```
printf("4\n"); fflush(stdout);
```

O mutare validă este specificată printr-un număr între 0 și 8, indicând coloana pe care se adaugă un jeton. Dacă un program afișează un șir care nu reprezintă un număr între 0 și 8, atunci automat va pierde. Programul va pierde și dacă încearcă adăugarea unui jeton la o coloană plină.

În plus față de indicii coloanelor, un program poate primi unul din cele 3 coduri speciale, 10, 11 și 12. Semnificația codurilor este următoarea: dacă un program citește de la tastatură numărul 10, a câștigat, dacă citește numărul 11 a pierdut, iar dacă citește 12 jocul s-a terminat remiză. Aceste coduri pot fi doar citite. În cazul în care programul vostru afișează unul din aceste coduri va pierde.

Deoarece platforma de joc gestionează interacțiunea dintre programe, puteți considera că mutările pe care le veți citi vor fi întotdeauna valide.

Un program care va juca prin intermediul platformei va primi ca argument în linia de comandă unul din șirurile *first* sau *second*. Dacă programul primește *first*, atunci va trebui să efectueze prima mutare, iar în caz contrar va muta al doilea.

Inițial, înainte să afișați orice altceva, trebuie să citiți un singur număr, care poate fi ignorat. Acest *magic number* are o semnificație specială doar pentru Ionel și Dorel. Numărul va fi citit o singură dată (la început), după care programul va trebui să respecte protocolul anterior.

Pentru mai multe detalii despre platforma de joc și despre protocolul de comunicare puteți consulta programele orientative din arhiva de pe site-ul de curs sau tutorialele video din ultima secțiune.

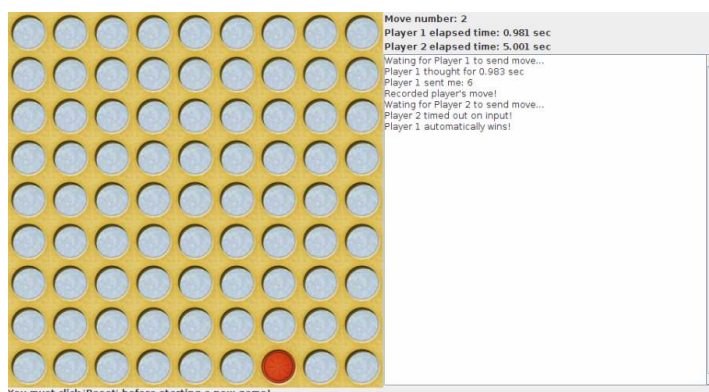
Reguli de notare

80% din punctajul temei se va acorda pentru rezultatul obținut de programul vostru în urma unui singur joc cu programul lui Ionel și Dorel. Acest program poate fi setat să joace la nivele diferite de dificultate. Puteți obține următoarele punctaje, astfel:

- 20 de puncte dacă programul vostru învinge adversarul la *Level 0*;
- 50 de puncte dacă programul vostru învinge adversarul la *Level 1*;
- 80 de puncte dacă programul vostru învinge sau face remiză, la *Level 2*.

Puteți selecta nivelul de dificultate pentru oponent din interfața grafică. **Va trebui să precizați în fișierul README împotriva cărei variante doriți să fie testată soluția voastră.**

La corectare, programul vostru va începe întotdeauna al doilea. Pentru testare pe sistemele proprii, puteți opta ca programul vostru să înceapă și primul.



Jucătorul 2 pierde automat deoarece timpul de gândire pentru o mutare depășește 5 secunde

Nu veți obține niciun punctaj la această parte în următoarele situații:

- programul vostru se termină neașteptat (cu eroare);

- programul vostru încearcă efectuarea unei mutări invalide;
- timpul de gândire pentru o mutare depășește 5 secunde;
- programul vostru pierde împotriva variantei specificate în README;
- nu se specifică nicio astfel de variantă în fișierul README.

Cu 10% din punctaj va fi notată claritatea codului, iar restul de 10% se vor acorda pentru comentariile din fișierele sursă și pentru explicațiile din fișierul README.

Soluțiile care folosesc algoritmi euristici fără să exploreze parțial arborele de joc nu vor fi luate în considerare și vor obține 0 puncte.

Temele copiate se punctează cu -10 puncte (se vor penaliza toți participanții la procesul de fraudare cu minus punctajul maxim pentru temă).

Tutoriale video

Pentru mai multe detalii, vă recomandăm să urmăriți următoarele tutoriale video:

- http://www.youtube.com/watch?v=o_pWN9Se470
- <http://www.youtube.com/watch?v=nkEN1drRG9E>