

Proiectarea Algoritmilor - Tema 2

Data publicare: 27 martie 2013

Deadline: 14 aprilie 2013 - 23:55

Vicențiu Ciorbaru
Radu Iacob
Alexandru Tudorică

1 Introducere

Ion și Vasile sunt foarte pasionați de fotbal și mai ales de strategiile echipelor. Cum rivalitatea dintre cei doi este atât de mare, fiecare se crede cel mai bun. Vasile vrea să-i vină de hac lui Ion pentru gardul stricat așa că l-a provocat la un meci de fotbal. Cum doar ei, fără echipă nu își pot arăta calitățile de căpitan s-au rezumat la o alta variantă a fotbalului, fotbalul pe hârtie.

Vasile știe ca Ion va încerca să trișeze și ca probabil va folosi un program care să-i spună unde să mute. Ca să-și asigure câștigul, Vasile, vă roagă pe voi să îi scrieți un program ce va juca pentru el acest meci și care îi va garanta succesul împotriva lui Ion.

2 Regulile jocului

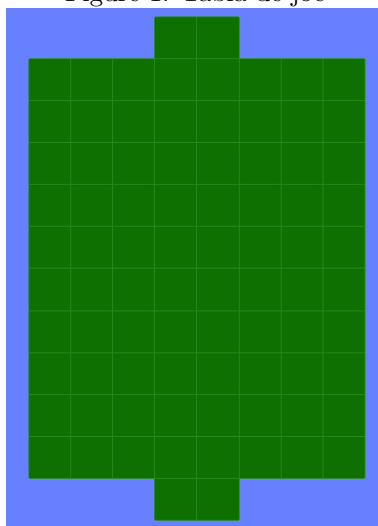
Fotbalul pe hârtie se joacă în doi jucători, fiecare mutând pe rând.

Scopul jocului este de a ajunge în poarta adversă. Jocul se termină când s-a ajuns într-o poartă sau nu mai sunt mutări valide. Cel care era la mutare și nu mai poate muta în nici o direcție pierde.

Tabla de joc

Jocul se joacă în mod normal pe o hârtie de matematică. Spațiul de joc este format dintr-un dreptunghi de 10 pătrățele în lungime și 8 în lățime ce reprezintă terenul de fotbal. La cele două capete ale terenului se mai adaugă încă două pătrățele, pe mijloc, ce vor reprezenta porțile.

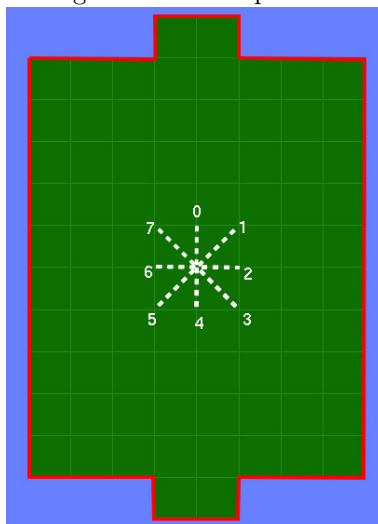
Figure 1: Tabla de joc



Mutările posibile

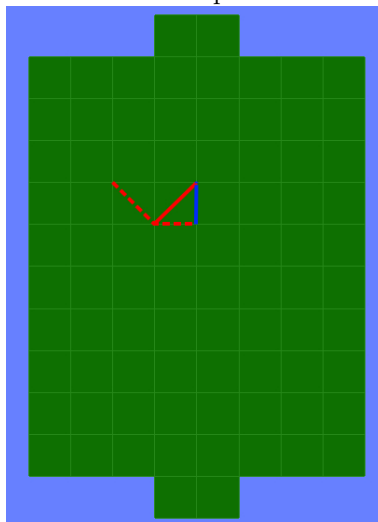
O mutare constă în trasarea unei linii pe o latură a unui pătrat sau pe o diagonală a acestuia. Acestea sunt codificate în sensul acelor de ceasornic conform în imagine.

Figure 2: Mutari posibile



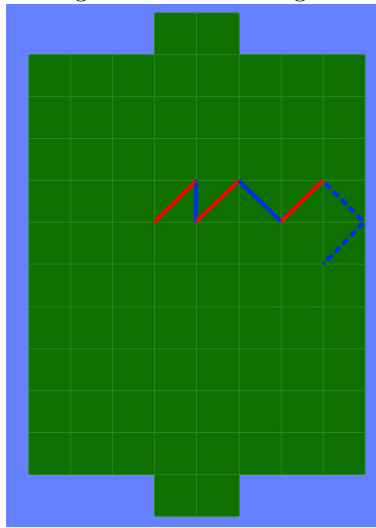
Dacă vârful în care s-a ajuns după mutare a mai fost atins în prealabil (de oricare dintre cei doi jucători) cel ce era la mutare este obligat să mai facă o mutare. O latură sau diagonală folosită **NU** mai poate fi refolosită.

Figure 3: Mutari multiple în aceeași rundă



Deasemenea, un jucător nu poate face o mutare care să completeze o latură ce reprezintă marginea terenului (laturile roșii din figura 2), dar dacă a ajuns pe margine este obligat să mai facă o mutare care îl readuce în interiorul terenului. În figura 3, jucătorul roșu a făcut prima mutare, apoi cel albastru. La mutare este jucătorul roșu și printre celelalte mutări disponibile poate face cele două mutări trasate cu linii punctate, amândouă în aceeași rundă.

Figure 4: Lovire margine



Jocul începe din punctul central al terenului. Jucătorii trasează linii alternativ, dar sunt obligați să continue de unde a rămas celălalt jucător.

3 Mod de utilizare

3.1 Server

Jocul împotriva a doi boți este gestionat de un server. Acesta pornește primul și apoi va lansa în execuție cei doi boți dați ca parametru. Primul parametru îl reprezintă comanda pentru a porni jucătorul 1, iar cel de-al doilea este comanda pentru a porni jucătorul 2.

Serverul redirectează intrarea și ieșirea boților astfel:

Boții vor citi de la intrare mesajele de la server, iar serverul va citi de la ieșirea fiecărui bot mesajele acestuia. Astfel, pentru a comunica cu serverul, un bot va **citi** de la **standard input** și va **scrie** la **standard output**

Pentru a rula serverul:

```
python2 server.py '<comanda-bot1>' '<comanda-bot2>' [-t <timeout>]
```

3.2 AI

Botul utilizat pentru testarea temei poate fi rulat în felul următor:

```
./<nume-exec> <ID-jucător A sau B sau silent> <nivel-dificultate  
0,1,2> <seed>
```

Botul de dificultate 0 este cel mai ușor de învins, acesta făcând în mare parte doar mutări valide, fără o strategie clară. **Botul cu ID-jucător "silent" nu va afișa mesaje pe ecran. Util pentru rulare în scripturi.**

3.3 Viewer

Pentru a vizualiza meciurile jucate vi s-a pus la dispoziție un program de vizualizare log-uri de jocuri. Aceste loguri sunt generate de server sub numele `server_log`

Viewerul poate fi rulat cu următoarea comandă:

```
./viewer [nume-log]
```

Viewerul poate fi rulat și fără fișier de intrare. Formatul datelor de intrare este același cu cel al log-ului și este explicat la rularea viewerului.

Viewer-ul are următoarele comenzi definite:

- Tasta D - o mutare înainte
- Tasta A - o mutare înapoi
- Tasta ESC - Iesire viewer

NOTĂ: Viewerul la ieșire se închide cu segmentation fault. Bug-ul vine din biblioteca freglut. Comportamentul este "normal".

4 Protocol de comunicare

Botul primește de la inputul standard câte un mesaj/linie. Mesajele vor avea primul caracter S, M sau F.

- S, acest mesaj va fi trimis botului care va face prima mutare, celălalt program va primi un mesaj de tip M prima dată
- M, acest mesaj reprezintă ce a făcut adversarul când a fost tura lui. După M este un întreg ce reprezintă numărul de mutări făcute în tură (N). După acest întreg urmează o serie de N cifre, fiecare cifră reprezintă o mutare. Fiecare mutare este descrisă de o cifră de la 0 la 7 după cum urmează:
 - 0 - N
 - 1 - NE
 - 2 - E
 - 3 - SE
 - 4 - S
 - 5 - SV
 - 6 - V

– 7 - NV

Exemplu de mesaj M:

M 5 0 1 2 3 4

Acest mesaj semnifică 5 mutări într-o tură: Prima spre N, a doua spre NE, a treia spre E, a patra spre SE și ultima spre S

- F, acest mesaj va indica terminarea jocului.

Catre iesire botul va genera mesaje identice cu mesajul M ce va indica mutările pe care le face. botul va considera tot timpul că poarta lui se află spre Sud, iar poarta adversă spre Nord.

Atenție! Fiecare mesaj trimis de bot trebuie să se finalizeze cu caracterul '\n'

5 Punctare

Tema va fi rulată cu un seed prestabilit (vezi precizări) în ambele configurații, cazul în care botul vostru are prima mutare precum și cazul în care este al doilea la mutare. Se vor rula teste pe toate cele trei nivele de dificultate 0, 1 și 2 (vezi AI).

Vor fi jucate în total 6 meciuri: câte 2 meciuri pe fiecare dificultate

Câștigarea unui meci pe dificultatea 0 acordă un punctaj de 10 PUNCTE.

Câștigarea unui meci pe dificultatea 1 acordă un punctaj de 15 PUNCTE.

Câștigarea unui meci pe dificultatea 2 acordă un punctaj de 20 PUNCTE.

Tema trebuie să respecte regulile generale de trimitere a temelor.

În fișierul README va trebui să descrieți soluția pe care ați ales-o pentru a implementa logica AI-ului.

Corectorii își rezervă dreptul de a scădea puncte dacă vor considera acest lucru necesar.

6 Format arhiva și testare

Temele pot fi testate automat pe vmchecker - acesta suportă temele rezolvate în C/C++ și Java. Dacă doriți să realizați tema în alt limbaj trebuie să trimiteți un e-mail lui Traian Rebedea (traian.rebedea@cs.pub.ro) în care să îi cereți explicit acest lucru.

Arhiva cu rezolvarea temei trebuie să fie .zip și să conțină:

- Fișierul/fișierele sursă
- Fișierul Makefile
- Fișierul README

Fișierul pentru make trebuie denumit **obligatoriu** Makefile și trebuie să conțină următoarele reguli:

- build, care va compila sursele și va obține executabilul.
- clean, care va șterge fișierele obiect și executabilul generat.

Atentie! Numele regulilor trebuie să fie exact cele de mai sus. **Atenție** Pentru cei ce folosesc C/C++ NU este permisă compilarea cu opțiuni de optimizare a codului (O1, O2, etc.).

7 Precizări

- Numele executabilului trebuie să fie: **bot** sau **Bot.class**
- Modul de rulare a botului vostru trebuie să fie următorul:
`./bot <bot-ID A|B> <seed>`
sau
`java Bot <bot-ID A|B> <seed>`
- Seed-ul ce va fi folosit la testare este **42** pentru amândoi boții
- Pentru a putea rula viewer-ul pe linux trebuie instalat pachetul freeglut a distribuției folosite. Exemplu pentru Ubuntu:
`sudo apt-get install freeglut3-dev`
- **Repository disponibil cu resursele temei la zi: [aici](#)**