

Strategii pentru jocul de dame

“Dame Inteligente”

Rezumat

Acest raport detaliaza dezvoltarea unui program pentru a juca jocul de dame englezesti. Scopul acestui proiect a fost de a proiecta si implementa un program care ar putea juca dame la un nivel inteligent. Acest lucru a necesitat o intelegere a jocului de dame si diferite strategii abordate de catre jucatori. Este necesar de asemenea cunoasterea principiilor teoriei jocurilor, precum si abilitati de programare necesara pentru a le pune in aplicare. Astfel, raportul abordeaza aspecte cheie ale programului, inclusiv proiectarea deciziilor luate, optimizarile utilizate pentru a imbunatati performanta, si testarea implementarii care a fost folosita pentru a aprecia cât de bine programul poate juca impotriva omului.

Scopul principal al acestui proiect a fost de a elabora si implementa un program de joc care joaca jocul de dame la un nivel inteligent. Obiectivul initial a fost ca programul ar putea detine propriile sale arme impotriva adversarilor umani de diferite de calificari. Programul a fost scris in Java si se bazeaza pe principiile teoriei jocurilor. Numele ales pentru program este “Dame Inteligente ”.

Mulumiri

As dori sa multumesc supraveghetorului meu, domnului profesor Catalin Stoean, pentru tot sprijinul si sfatul pe care mi l-a dat pe parcursul proiectului. Domnul profesor a fost intotdeauna abordabil si a dorit sa ma ajute cu orice probleme sau intrebari pe care le-am avut.

Scopurile si obiectivele

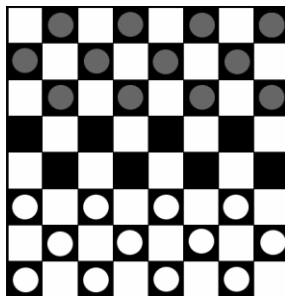
Planul de proiect ofera o lista completa a scopurilor si obiectivelor proiectului, cu toate acestea ele pot fi rezumate dupa cum urmeaza:

- obtineti o versiune de lucru a jocului de dame capabila sa joace impotriva oamenilor
- scrieti un program care incearca sa joace dame(chiar daca acesta joaca slab)
- imbunatatiti programul astfel incat sa joace la un nivel ridicat
- permiteti utilizatorului sa selecteze intre o varietate de stari de dificultate
- testati programul impotriva oamenilor si a altor programe de dame

Regulamentul jocului de dame englezesti

Programul de joc este implementat pe baza regulilor din jocul de Dame englezesti. Jocul de Dame englezesti este jucat de doi jucatori, "negru" si "alb", care incep pe laturile opuse ale unei table. In Damele englezesti, versiune a jocului pe care este proiectat programul in cauza, jocul este jucat pe o tabla de 8x8, si fiecare jucator incepe cu douasprezece piese. Piesele sunt prevazute dupa cum se arata in figura 1. Scopul jocului este de a elimina fie toate piesele adversarului de la bord, fie de a le pune intr-o pozitie in cazul in care nu au mutari legale la rândul lor.

Fig. 1



Piesele se misca in diagonala, ceea ce inseamna ca patrate albe nu sunt utilizate in joc, toate piesele se misca in diagonala intre patrate negre. O piesa poate fi mutata intr-o directie de inainte cu un patrat, fie spre dreapta sau spre stânga. Figura 2 prezinta o bucata neagra ce se deplaseaza inainte.

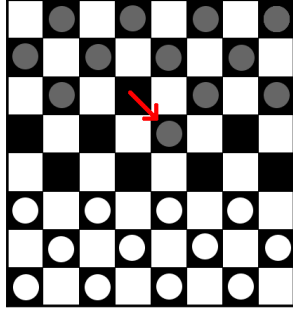


Fig 2

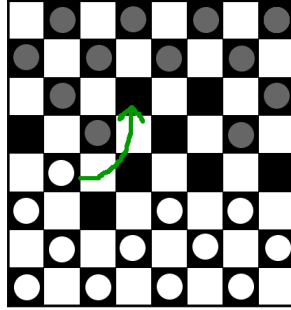


Fig 3

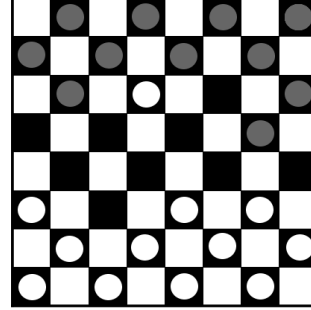


Fig 4

Daca o piesa este diagonal adiacenta pentru o piesa opozanta, din fata ei, si exista un patrat gol in partea de dincolo a piesei opuse, piesa poate sari peste piesa opusa scotind-o de la bord. Acest lucru este cunoscut sub numele de "salt" sau "captarea" piesei adversarului, si este demonstrata in figurile 2.3 si 2.4.

Daca un "om", o piesa ajunge in partea departata a bordului (cunoscuta ca "rândul regelui"), piesa este actualizata la denumirea de "rege". Regii sunt puternici, deoarece acestia se pot deplasa si face salturi inapoi, precum si inainte. In versiunile de calculator a jocului, regii sunt de obicei etichetati cu o coroana sau un "R".

Implementarea

Limbajul ales pentru program este Java. Un motiv pentru aceasta este faptul ca, in Java este posibil sa creezi obiecte pentru a reprezenta diferite aspecte ale domeniului, in acest caz un joc de dame. Obiectele implementate in Dame Inteligente includ un obiect de mutare, care a fost utilizat ori de câte ori detalii despre o miscare au trebuit sa fie transmise intre metode. De asemenea, un obiect "patrat" a fost implementat, care reprezinta un patrat de pe bord si contine informatii despre piesa de pe acel patrat (sau lipsa acestora).

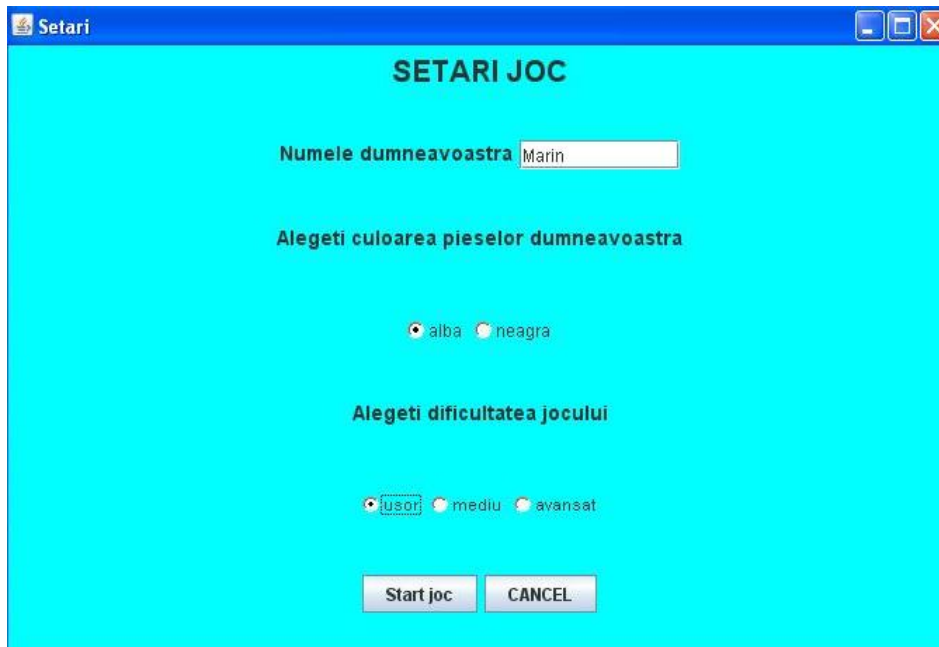
Un alt motiv pentru a alege Java a fost ca Java ofera pachetul Swing, care permite crearea unei interfete grafice pentru utilizator atractiva. Desi acest lucru nu este critic pentru functionarea programului, este mult mai convenabil atât pentru scopuri de testare si pentru utilizatorii finali ai programului. Acesti factori au condus la decizia de a folosi Java pentru punerea in aplicare a programului.

Interfata grafica a jocului cuprinde trei ferestre principale in care se desfasoara jocul.

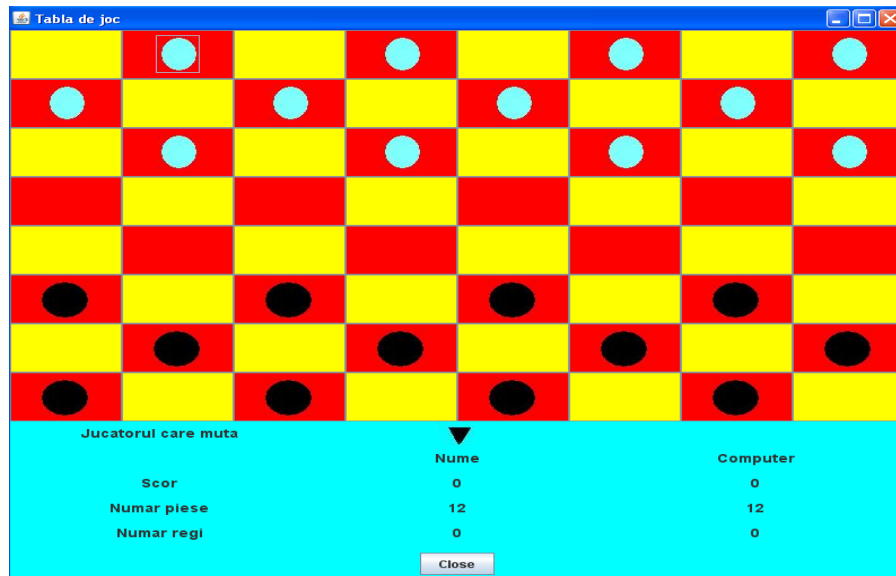
Fereastra “Intro” este fereastra de start a jocului. Oferă informatii despre denumirea jocului, realizatorul jocului, etc.



Fereastra “Setari” permite utilizatorului sa-si aleaga setarile jocului. Utilizatorul poate sa-si aleaga un nume altul decat cel implicit, sa-si aleaga culoarea pieselor si nivelul de dificultate.



Fereastra “Tabla de joc” este fereastra principala a jocului, aici desfasurandu-se practic jocul in sine. Afiseaza informatii despre starea curenta a jocului, scorul realizat de jucatori, jucatorul care este la rand sa mute, numarul de piese al fiecarui jucator la un moment dat, etc.



Arborele de joc

Un mod util de a analiza jocuri este de a le vedea ca o structura arborescenta, in cazul in care fiecare nod din arbore reprezinta o pozitie posibila (sau stare a jocului). Radacina arborelui reprezinta pozitia initiala, si in aceasta stare initiala se afla mutarea jucatorului care este la rand. Fiecare mutare valabila care poate fi facuta de catre jucator duce la un nod copil care reprezinta noua pozitie care rezulta din aceasta miscare. Aceasta craca continua in josul arborelui pe masura ce jucatorii fac mutari. Nodurile frunze ale arborelui reprezinta starile terminale unde jocul s-a incheiat, in mod normal, in cazul in care un jucator a castigat sau jocul s-a incheiat la egalitate. De obicei, un jucator are posibilitatea de a alege intre cateva potentiale mutari de fiecare data, si ca rezultat dimensiunea arborelui jocului poate creste exponential. Acest lucru este valabil in special in jocuri cum ar fi damele si saahul, pentru ca in timp ce jocul progresa jucatorii pot avea mai multe miscari disponibile decât au avut la inceput, cand tabla se deschide.

Algoritmul Minmax

Strategiile poate fi utile in jocuri mici, cum ar fi x si zero, cu toate acestea pentru jocuri mai mari, cum ar fi damele acestea nu pot fi utilizate. Acest lucru se datoreaza faptului ca arborele de joc este prea mare pentru ca un calculator sa poata stoca o strategie completa.

Arborele jocului in sine este, de asemenea, prea mare pentru a fi stocat in memorie, din cauza factorului de bransament si adâncimii mari a arborelui (multe jocuri de dame dureaza mai mult de 100 de mutari). Prin urmare, atunci când scrieti un program de joc, este comun numai sa generezi parti ale arborelui care sunt cerute in acel moment. Algoritmul Minmax este o metoda de cautare inainte prin arbore,de a alege cea mai buna mutare pentru un jucator atunci când se presupune ca adversarul va juca un joc perfect.

Funcția de evaluare

Pentru jocuri mari, ca damele, arborele complet al jocul nu poate fi generat in acelasi timp, dupa cum am mentionat mai devreme. Prin urmare, nu este posibil sa se aplice algoritmul minmax standard pentru un astfel de joc.

Deoarece noi nu putem cauta toate modalitatile de a ajunge la nodurile frunza (unde castigurile/rasplatile sunt gasite), suntem in imposibilitatea de a gasi adevarata valoare a oricarei pozitii intermediare in arborele de joc. Cu toate acestea, putem obtine o aproximare a valorii pozitiei utilizând o functie de evaluare.

Cel mai important si interesant aspect al unui program de joc este functia de evaluare. Functia de evaluare este necesara pentru a incerca sa indice cat de favorabila este pozitia atunci când nu putem cauta inainte pana la nodurile frunza ale arborelui jocului. Deoarece nu putem cauta inainte la pozitiiile finale, functia de evaluare poate fi vazuta ca o aproximare la valoarea reala a unei pozitii, si, ca atare, nu poate fi in intregime corecta. Eficienta functiilor de evaluare, prin urmare, poate fi masurata doar prin testare.

Funcția de evaluare analizeaza o pozitie bazandu-se pe o serie de factori specifici jocului. Fiecare factor da o ponderare bazata pe efectul pe care factorul il are asupra jocului, si prin combinarea acestor factori se calculeaza o valoare totala a pozitiei. Atunci când este echipat cu o functie de evaluare, programul de joc poate cauta in fata un anumit numar de miscari, gasi valorile pozitiiilor la acest nivel, si apoi trimite aceste valori inapoi prin arbore folosind algoritmul minimax. Valoarea obtinuta prin utilizarea functiei de evaluare nu va reflecta

intotdeauna valoarea reala a pozitiei. Pentru a gasi adevarata valoare conceptual, am avea nevoie sa facem o cautare minmax exhaustiva a tuturor modalitatilor de a gasi noduri frunze (pozitii finale), dar acest lucru nu este posibil. Singura modalitate de a imbunatati estimarea functiei de evaluare a valorilor pozitiiilor este de observare in timp a jocului, si de ajustare a coeficientilor in consecinta, pentru a imbunatati performanta.

Realizarea scopurilor si obiectivelor

Principalele obiective ale proiectului au fost indeplinite. Initial am intentionat sa dezvolt o versiune de lucru a jocului de dame, iar acest lucru a fost realizat in intregime. Scopul de baza urmator a fost sa scriu un program care incerca sa joace dame, chiar daca juca slab. Acest lucru a fost realizat prin implementarea unui algoritm minimax de baza si a unei functii de evaluare.

Urmatorul scop este deasemenea la fel de important: de a imbunatati programul, astfel incât acesta sa joace jocul destul de bine pentru a putea invinge majoritatea adversarilor umani.

In ansamblu sunt multumit de programul creat pana acum. Sper ca in curand sa fie gata in intregime si sa poata juca la un nivel foarte bun!

Referinte

a) Referinte Web

1. A Knowledge-Based Approach of Connect Four – Victor Allis

http://www.farfarfar.com/games/connect_four/connect4.pdf

2. Chinook – Jonathan Schaeffer

<http://www.cs.ualberta.ca/~chinook/>

3. Rules of Draughts – The Official English Draughts Association

<http://home.clara.net/davey/newpage4.htm>

4. The Rules of Draughts – Masters Games

<http://www.mastersgames.com/rules/draughts-rules.htm>

5. The International Draughts Site – Nicolas Guibert

http://membres.lycos.fr/nic55/dames/dames2_old.htm

6. Checker Bitboard Tutorial – Jonathan Kreuzer

<http://www.3dkingdoms.com/checkers/bitboards.htm>

7. Solving Connect Four – Victor Allis

<http://web.archive.org/web/20041204210315/www.cs.vu.nl/~victor/connect4.html>

8. Reviving the Game of Checkers – Jonathan Schaeffer

<http://www.cs.ualberta.ca/~jonathan/Papers/Papers/olympiad.ps>

9. Chess Evaluation Functions - François Dominic Laramée

<http://www.gamedev.net/reference/articles/article1208.asp>

10. Sharp Checkers – Paulo Pinto

http://www.progtools.org/games/projects/sharp_checkers/sharp_checkers.html

11. ICheckers – Adnan Khalid

<http://www.geocities.com/icheckers/>

b) Referinte literare

12. Andrea Schalk, COMP30191: The Theory of Games and Game Models,

University of Manchester, School of Computer Science, 2006

13. Catalin Stoean, Ruxandra Stoean : Evolutie si Inteligenta artificiala. Paradigme moderne si aplicatii, Editura Albastra, 2010