Jocuri cu adversari rationali

Stategi implemenate:

1) Random
2) MinMax - Alpha-Beta - ID
3) Monte Carlo Tree Search
4) Monte Carlo Tree Search - rollout : MinMax

Pentru reprezentarea hărţi am implementat clasa Map, Ghost, PacMan care sa retina configuratia actuala statri, unde se afla PacMan, fontomele, mancare, cate vieti mai sunt etc si foate important o funcţie care da starea urmatoarei harti dupa aplicarea unor mutari: run_turn . Clasa Game joacă jocul pana se ajunge intr-o stare finală aplicand diverse strategii pentru jucători, ele fiind date ca parametru.

I) Random:
Se genereza miscarile posibile pentru PacMan/Ghosts si se alege random din accea lista o mutara.

II) MinMax Apha-Beta Iterative deeping:
Perntru acesta se considera ghosts ca fiind o singura entitate care muta 2 piese. PacMan este Max si Ghosts este Min, functia heuristica fiind
factor * castigul_pac_man + dist_pac_ghost1 + dist_pac_ghost2, daca este pierdut de catre PacMan este factor * max_win + 2 * (height + width), acesta are urmatoarea interpretare: un factor care sa il determine pe Pac Man sa alega mancare/puntele in detrimintul sa fuga de fantome, Pac Man va vrea sa colecteze cat mai multe punte si sa fie cat mai departat de catre ghosts, iar fantomele se fie cat mai apropria si cat mai puţine puncte, daca Pac Man nu se misca o sa alega pozitia care te duce cel mai aproape de Pac Man.
Se executa in maximul de buget, se contorizeaza numarul de stari create.

III) Monte Carlo Tree Search:
S-a folosit implementarea din laborator. Se aplica algoritmul si pentru parte de reward se foloseşte aceeaşi funcţie de evaluare de la MinMax, cu excepţia ca

pentru Min se calculeaza complementul şi pentru ca se alege cu cea mai mare reward.

IV) Monte Carlo Tree Search + MinMax:
Acesta doar modifica parte de rollout: in loc sa fie random se foloseste Min Max implementat. El durează cel mai mult din cauza combinării celor doi algoritmi, pasul de rollout direaza cel mai mult ducand pe harta cea mai mare ca sa dureze in jur de 1-2s pentru a genera o pereche de mişcări.

Comparatie:
Am testat pe mai 3 harti: 5 x 7, 15 x 15, 7 x 12 si pentru buget am folosit 9, 30 si 100 cu depth = 7.

I) PacMan : Random

1. Ghosts: Random
   La acesta de obicei castiga fantomele, dat fiind ca Pac Man nu stie cum sa isi aleaga poziţia care sa-l duca către o poziţie care sa maximizeze castigul, indiferent de harta, mai exista si cazuri in care Pac Man catiga in jur de 1/10.
2. Ghosts: MinMax:
   Aici fantomele castiga de fiecare data, deoarece stie cum sa urmareasca pacman, din cauza euristici.
   Aici castiga in jur de 91 pentru harta marea si 30 buget, acesta scade la 41 pentru 100.
3. Ghosts: Monte Carlo Tree Search:
   Tot fantoma castiga: harta2: 30 - 59, 100 - 62; harta1: 100 - 11, 30 - 27, 9 - 34, harta3: 100 - 25.
4. Ghosts: Monte Carlo Tree Search + MinMax:
   Tot fantoma castiga
   harta1: 9 -> 103; 30 -> 74; 100 -> 53
   harta2: 9 -> 465; 30 -> 281, 100 -> 169
   harta3: 100 -> 133; 30 -> 175; 9 -> 47
   Se poate observa ca scade numarul de ture dar crete timpul efective de rulare.

II) Pac Man: MinMax:

1. Ghost: Random
   Aici castiga PacMan pe hartile mici, medii, pe harta mare pierde chair si cu maximul numarului de mutari, acest lurcu se intampla din mtive ca este prea complexă harta si nu poate sa ajunge la unde punte.

   In Turn 2108
   Life player : 0
   Map configuration:

   X X X X X X - - - X X X X X X
   G - - - - - - - - - - - - -
   - - X - - - X X X X X X X X -
   - - X - - - - - - - - - - -
   - - X _ _ _ X X X X X X X X_ -
   - - X _ _ _ _ _ _ - - - - - -
   - _ X X X X X X _ - - - - - -
   - - _ _ _ _ _ _ _ - - _ - - -
   - - X X X X X X X X X X X X -
   - - X - - - - - - - - - - -
   - - X - X X X X X X X X - - -
   - - X - - - - - - - - X - - -
   - - X X X X X X X X X X - - -
   G - - - - - - - - - - - - -
   X X X X X X - - - X X X X X X

2. Ghost: MinMax:
   Pentru harta mare se observa un efect foarte ciudat in care fantoma urmareste pe Pac Man pentru hartile mari/medii. Pe cele mici, medi variaza in functie de bugetul maxim. Acest lucru apare din cauza hartlor ciirculare.
3. Ghost: Monte Carlo Tree Search:
   Pe hărţile mari se durează foarte mulţi pentru bugetul maxim, si am oprit programul. Pe harta 1 si 2 castiga pentru un buget mare.
4. Ghost: Monte Carlo Tree Search + MinMax:
   Pe harta1 el castiga pentru orice buget. Pentru harta2, 3, castiga fantomele dar dureaza foarte, foarte mult.

III) Pac Man: Monte Carlo Tree Search:

1. Ghost: Random

   Aici este o surpriză deoarece Pac Man pierde pentru majoritatea simularilor pe majoritatea hartilor mari si medii, si castiga pe harta mici.

2. Ghost: MinMax

   Pac Man pierde pentru toate hartile si maximul de buget, existant si cazuri pentru harti mici in care castiga Pac Man. Exista cazul pentru n

3. Ghost: Monte Carlo Tree Search:

   Pentru acesta fantomele castiga pe majoriatea hartile si pe cea mica pentru un buget mic si mediu castiga Pac Man.

4. Ghost: Monte Carlo Tree Search + MinMax:

   Fantoma castiga pentru acesta de fiecare data indefirent de configuratie, folosind in loc random, min max acesta va oferi un rezulatat mai bun decat cea random si ofera mai multe date.

IV) Pac Man: Monte Carlo Tree Search + Min Max

1. Ghost: Random:

   PacMan castiga tot timpul, dar dureaza foarte mult pe harat2 cu 100/30. Aici pierde cel mai greu o vieţi.

2. Ghost: MinMax:

   Pentru cazul in care este harta1 cu 9 (un horizont min) programul se blocheaza in starea:

   X X X _ X X X
   X - _ G _ _ X
   X - X X X G X
   X - - P _ _ X
   X X X - X X X

   Dupa aceai de obiecei castiga cea cu MinMax, fantoma.

3. Ghost: Monte Carlo Tree Search:

   Dureaza mai mult aici, dar aici castiga Pac Man, deci MCTS cu MinMax in etapa de rollout este mai buna decat cea random.

4. Ghost: Monte Carlo Tree Search + MinMax:

   Si pe hartile mici dureaza foarte mult, este cea mai lenta varianta.
   Pentru un buget mai mic catiga PacMan iar pentru cea in care este un buget mai mare este castigator Ghosts, este valabil pentru orice harta.

harta1-9:

	Random	MinMax	MCTS	MCTS_B
Random	W: 9 L: 91	W: 0 L: 100	W: 4 L: 96	W: 0 L: 100
MinMax	W: 68 L: 32	W: 100 L: 0	W: 68 L: 32	W: 59 L: 41
MCTS	W: 26 L: 74	W: 1 L: 99	W: 17 L: 83	W:1 L: 99
MCTS_B	W: 86 L: 14	W: 27 L: 67	W: 52 L: 48	W: 30 L: 70

harta1-30:

	Random	MinMax	MCTS	MCTS_B
Random	W: 6 L: 94	W: 0 L: 100	W: 2 L: 98	W: 0 L: 100
MinMax	W: 65 L: 35	W: 0 L: 100	W: 53 L: 47	W: 26 L: 74
MCTS	W: 43 L: 57	W: 2 L: 98	W: 29 L: 71	W: 4 L: 96
MCTS_B	W: 90 L: 10	W: 16 L: 84	W: 53 L: 47	W: 12 L: 88

harta1-100:

	Random	MinMax	MCTS	MCTS_B
Random	W: 7 L: 93	W: 0 L: 100	W: 5 L: 95	W: 3 L: 97
MinMax	W: 65 L: 35	W: 0 L: 100	W: 33 L: 67	W: 20 L: 80

MCTS	W: 57 L: 43	W: 2 L: 98	W: 16 L: 84	W: 14 L: 86
MCTS_B	W: 85 L:15	W: 32 L: 68	W: 55 L: 45	W: 53 L: 47

harta3-9:

	Random	MinMax	MCTS	MCTS_B
Random	W: 0 L: 15	W: 0 L: 15	W: 0 L: 15	W: 0 L: 15
MinMax	W: 12 L: 3	W: 0 L: 15	W: 10 L: 5	W: 9 L: 6
MCTS	W: 0 L: 15	W: 0 L: 15	W: 0 L: 15	W: 0 L: 15
MCTS_B	W: 11 L: 4	W: 2 L: 13	W: 8 L: 7	W: 9 L: 6

harta3-30:

	Random	MinMax	MCTS	MCTS_B
Random	W: 0 L: 15	W: 0 L: 15	W: 0 L: 15	W: 0 L: 15
MinMax	W: 14 L: 1	W: 4 L: 11	W: 11 L: 4	W: 2 L: 13
MCTS	W: 0 L: 15	W: 2 L: 98	W: 0 L: 15	W: 0 L: 15
MCTS_B	W: 12 L: 3	W: 3 L: 12	W: 8 L: 7	W: 5 L: 10

harta3-100:

	Random	MinMax	MCTS	MCTS_B
Random	W: 0 L: 15	W: 0 L: 15	W: 0 L: 15	W: 0 L: 15
MinMax	W: 12 L: 3	W :5 L: 10	W: 6 L: 9	W: 4 L: 11
MCTS	W: 12 L: 3	W: 0 L: 15	W: 2 L: 13	W:0 L: 15
MCTS_B	W: 13 L:2	W: 5 L: 10	W: 10 L: 5	W: 8 L: 7