Development of an artificial intelligence for a 2 player game

Introduction

Project's theme : Artificial Intelligence

Still in developing

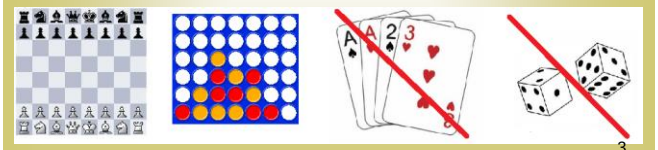
In purpose to create an intelligence similar to an human intelligence

Introduction

Applied on a two-players game

Using the Min-Max algorithm

For zero-sum game, with complete information



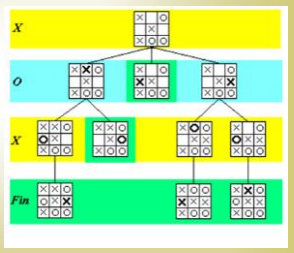
Game representation

The game can be represented by a tree :

- The root is the initial situation

- Each node is a potential position

- An arc symbolizes the move between two positions



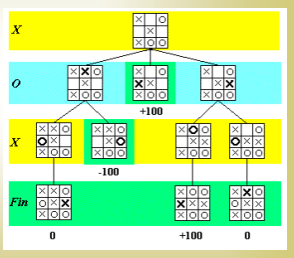
Final Situation

For each leaf we mark the game position with:

- A high mark if we win

- 0 in the case of a draw

- A low mark (negative) if we lose



Intermediary situation

Impossible to finish the algorithm in some complex games

Using of a cost function to estimate the player's advantage

Cost Function

Depend on the game and determine the IA's quality

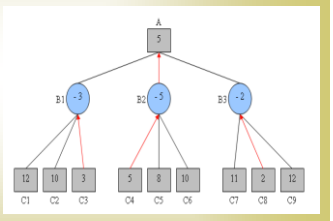
Comparison between players pieces

7

Negamax (or Max-Max)

Always represents the best move of the present player by a positive number

Take the opposite of the value when it is the opponent's turn and use the maximum in place of the minimum



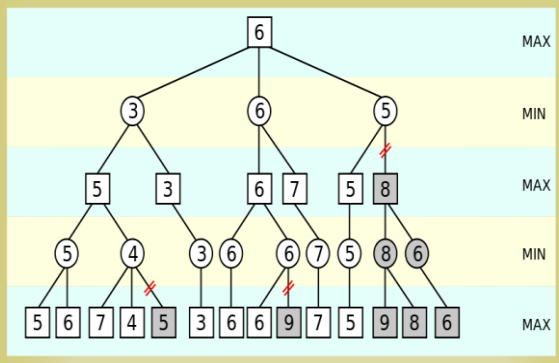
8

The Alpha-beta pruning

Objective:
Decrease number of nodes evaluated by the algorithm

How ?
Stop to develop the tree when we know that the result won't be used

9



10

Implementation of the algorithm in a game

Possibility to work on many games:

- Connect Four

- Chess

- Checker

11

Programming choice

Language chosen : C++ for Speed execution (with open mp)

Graphic library chosen : QT

C++

Java

12

Open MP

```
#include<stdio.h>
#include<stdlib.h>

…

for(n=0;n<8;n++){
      printf("Element %d processed \n",n);
}
…
```

13

Open MP

```
#include<stdio.h>
#include<stdlib.h>
#include<omp.h>
…
#pragma omp parallel for
for(n=0;n<8;n++){
      printf("Element %d processed by
      the thread %d \n",n
      ,omp_get_thread_num());}
…
```

14

Open MP

```
Element 0 processed by the thread  0
Element 4 processed by the thread  1
Element 1 processed by the thread  0
Element 5 processed by the thread  1
Element 2 processed by the thread  0
Element 6 processed by the thread  1
Element 3 processed by the thread  0
Element 7 processed by the thread  1
```

15

Conclusion

Knowledge about the Min-Max algorithm

Game structure created in C++

Ready to implement

16