Universidade de Brasília

Departamento de Ciência da Computação

Professor Eduardo Alchieri

Amanda Oliveira Alves 150116276

Frederico Pinheiro Dib 150125925

Turma A

Mancala

21 de Junho de 2016, Brasília

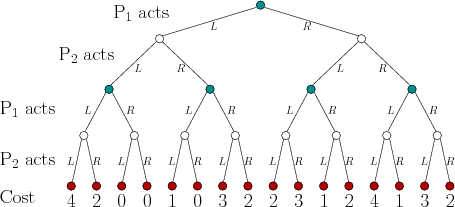
**Introdução**

O principal objetivo do trabalho é utilizar uma árvore genérica para implementar inteligência artificial no jogo de tabuleiro chamado Mancala. Essas árvores são chamadas de game trees, elas basicamente representam as possibilidades de jogadas a partir de um estado de jogo, a raiz é o estado atual do jogo e suas filhas são todas as jogadas possíveis. A inteligência artificial deve prever as próprias jogadas e a do adversário e escolher as opções que o ajudem a ganhar mais rapidamente.

O jogo contém modo multiplayer e modo singleplayer com três dificuldades (fácil, médio e difícil). Antes de tudo aparecerá um menu com as opções de jogo disponíveis, ao escolher a opção desejada o jogo será iniciado, depois é só escolher o número correspondente a uma das casas válidas e a jogada será feita. São sete casas para cada jogador, seis para executar as jogadas e duas Kahalas, onde guardam-se os pontos feitos durante o jogo. Ao final quem obter mais pontos em sua respectiva Kahala vence.

**Implementação**

A estrutura de dados usada para implementar o jogo foi uma árvore com seis filhos em cada no. A raiz representa o estado de jogo atual e os seus filhos as jogadas possíveis a partir daquele ponto. A seguir exemplo da estrutura de uma árvore:



O programa começa com a função menu que mostra ao usuário todas as opções de jogo disponíveis. Se o modo selecionado for multiplayer o programa executará as jogadas de acordo com a casa escolhida respeitando a vez de cada jogador, continuando até haver um vencedor.

No caso do jogo singleplayer, a grande diferença será quando for a vez do jogador dois, que corresponde ao computador. Com o uso da função criarArvore o computador criará uma árvore com todas as jogadas possíveis a partir do estado de jogo atual, depois que a árvore estiver pronta uma função irá avaliar todas as jogadas e escolher a melhor opção. Existem duas funções que serão usadas para essa finalidade, elas possuem algumas diferenças no funcionamento que serviram para configurar os níveis de dificuldade, elas são:

minimax – Essa função é usada no nível médio e difícil, ela prioriza jogadas que farão com que o computador jogue novamente (quando a última peça do jogador é semeada em sua própria kahala ele pode efetuar outra jogada), então se está previsto que uma jogada resultará em repetir sua vez ela será a escolhida. Se não houver jogadas que se encaixem no primeiro critério serão analisadas as seis opções e o computador escolherá a melhor jogada, ou seja, a que aumentar mais sua pontuação.

Paralelamente a função retornará a folha que contém a maior diferença entre os pontos dos jogadores, considerando quando o computador estiver vencendo, em conjunto com os dois critérios anteriores.

O computador nessa função está sendo “inteligente”, pois ele considera que o seu adversário também fará as melhores jogadas possíveis, planejando assim suas jogadas de acordo também com as jogadas do seu oponente.

A diferença de execução dessa função no modo médio e no difícil é somente a variável lim, essa variável é usada para especificar quantas jogadas futuras o computador deve prever a partir do estado atual de jogo, em outras palavras, limita a altura da árvore. No nível médio o limite da altura é menor, então o computador faz menos previsões, consequentemente a dificuldade diminui.

MinimaxFacil – Essa função é usada somente no nível fácil, diferente da função anterior essa não considera como prioridade executar jogadas que farão o computador jogar novamente, ela apenas retorna o ramo que possui a folha com a maior diferença de pontos. O computador nessa função é “burro” porque ele não leva em consideração as jogadas do adversário para planejar suas próprias jogadas.

A função Fim mantém o jogo funcionando, ela checa o campo dos dois jogadores, se algum deles estiver vazio, retorna-se 1 para a main e o jogo acaba. Após o fim do jogo a função PontoFinal move as peças remanescentes para sua respectiva Kahala e informa o jogador com a maior soma de pontos, ou seja, o vencedor.

**Conclusão**

Esse trabalho ajudou a compreender melhor como funciona a implementação da estrutura de dados árvore. Foi preciso entender bem como funciona a estrutura, aprender a manipular os ponteiros e entender como criar uma inteligência artificial. A ideia de criar um computador “inteligente” fez do trabalho um processo muito interessante.

As maiores dificuldades encontradas foram fazer a função minimax, sobretudo definir seus parâmetros para avaliar as jogadas do oponente, e montar a estrutura da árvore com as previsões de jogadas válidas. O problema foi trabalhoso pelo fato de precisar converter os índices do vetor que forma o tabuleiro, saber aplicar as regras do jogo a cada situação e definir bem a vez de cada jogador, contudo o restante não representou grandes dificuldades.

**Bibliografia**

- Slides do professor

- Mancala Wikipédia

- Minimax Wikipédia

- Tutoriais no Youtube monstrando como jogar mancala