

Research on the Surakarta chess game program based on Unity 3D

Wang LinLin, Wu YuNu, Wang YaJie, Cao Guoqiang

(Engineering Training Center, Shenyang Aerospace University, Shenyang 110136)

E-mail: wlin_23@163.com

Abstract: Surakarta chess is researched with the computer game technology, which is a kind of computer chess game in the complete information game. In order to increasing the human-computer interaction of the computer games system, the representing method of chessboard with 3D form is studied by use of unity 3D game engine technology. The Surakarta game system is designed using VC++ language. In this system, the Monte Carlo Tree Search(MCTS) algorithm and Minimax Algorithm are used together. This system has improve the visual effect, and combining two searching algorithm is feasibility.

Keywords: computer games, Surakarta, Unity 3D, searching algorithm.

1. Introduction

Artificial Intelligence(AI) **play a more and more important role in this century**. Computer Game, which is an important branch of AI, is paid much more attention gradually. Computer game systems include chess board representation, evaluation function and searching algorithm. Currently, the chess board system is generally simple flat style. Unity 3D is a synthesizing type ultimate game development platform and based on the Unity Technologies. Using Unity can build high-quality 3D and 2D games. In this paper, the Unity 3D technology is employed to build a 3D chess board. The effect of human-computer interaction is increased. Otherwise, the searching algorithm is also researched combining the Monte Carlo Tree Search(MCTS) algorithm with the Minimax Algorithm.

2. Surakarta chess

Surakarta chess is a two-player game, and is names with the place where it derived from, Surakarta of island Java, Indonesia.

The chess board is combined by 6x6 square on the corner of the network and 8 circular arcs at the 8 corners of the grid, as shown in Figure 1.

To distinguish from each other, two different color chess pieces are used. Each side has twelve chess pieces.

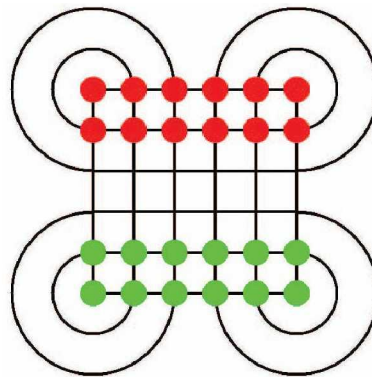


Figure 1. Surakarta chess

Ordinarily pieces move one square vertically or diagonally to a position that is not occupied by another piece beside captures. However, a whole arc line must be passed through if you want to capture the opposite piece.

3. 3D implementation

There are a lot of 3D game engine such as Unreal Engine 4, **Frost Engine 3**, Unity 3D, and so on. In this paper, Unity 3D is chosen as the main development engine due to its good performance and powerful cross-platform ability.

The steps to building the 3D chess board are as follows:

(1) Founding the menu scene. The menu scene are designed using UGUI, the new UGUI are all based on a Canvas. We should set the property of canvas. After all the relating components are added, the event listener for the button should be appended. The event listener mechanism of Unity 3D is very convenient, just adding a Button component for the **object**, and appending a specific method for the 'on click' property.

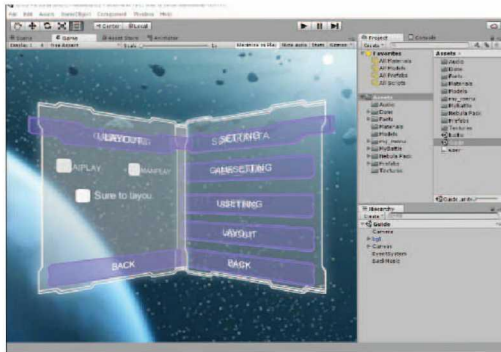


Figure 2 the menu scene

(2) Creating chess board scene. You can add model for each chess piece from Toolbar, or use existing model, such as the model designed by 3Dmax or download from internet. With latter method, the model should be imported into the unity 3D.

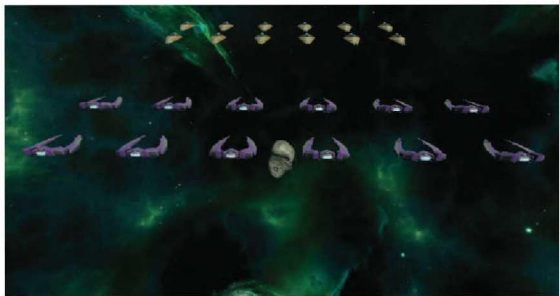


Figure 3 the chess board scene

(3) The store of Game data. The game data is got through CSV file, which is exported from Excel. The CSV scripts(C# script) are created and imported into the 'System.IO' and the 'System.Collection.Generic', then the collection class list should be built to save the data. The

CSV file will be read line by line using the 'StreamReader', and the data is saved in the list.

(4) Establishing the communication mechanisms. In order to improving the searching efficiency, the search engine is mainly implemented with the style of calling a dynamic-link library through the C++ and the C#. Currently, unauthorized server correspondence method or IO stream communication are usually adopted.

A simple IO communication example:

read:

```
string MyRead(string path)
{
    StreamReader sr = new
    StreamReader(path, Encoding.Default);
    string line = sr.ReadLine();
    sr.Close();
    return line;
}
```

write :

```
void MyWrite(string path, string
StepCommand)
{
    FileStream fs = new FileStream(path,
    FileMode.Create);
    //Get byte array.
    byte[] data =
    System.Text.Encoding.Default.GetBytes(Step
    Command);
    //Start write.
    fs.Write(data, 0, data.Length);
    //Clear the buffer,close the stream.
    fs.Flush();
    fs.Close();
}
```

4. Searching algorithm

4.1 Opening library

For the beginning of the production of the chess library, because of the pawn information is more determined at the beginning, the basic law can be fixed as a starting point for the opening Library.

4.2. Endgame library

Endgame library includes our victory library and our losing library, that is including some specific situation, breaking all its way. According to our algorithm, we search the

possible situation which has a great opportunity to win, it is victory library. On the contrary is the losing library. when the search engine search it in depth, once found that the current situation match with our victory database, the branches of the value will be amplified and lead the opponent in this situation as much as possible. However, if the current situation match the losing library, we should mark it immediately and search the node that can avoid this situation.

4.3 History Heuristic

History Heuristic refers to recording and saving the previous best node or trigger pruning node in the searching progress, so that when these nodes are encountered in the move ways generating function, they can be placed in the prior position and searched preferentially.

4.4 Algorithm:

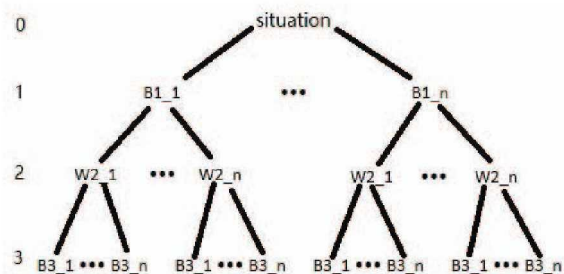


Figure 4 searching progress

B1_1 ... B1_n are Black all tactics of the current situation, W2_1 ... W2_n are while all tactics after black go.

According to the evaluation function according to our interests dislodge a part of tactics and establish a tactics set B1, the tactics be dislodge temporarily not to search. And then ergodic all tactics int the B1, and get all tactics of while after black go in every way. reach the second floor. In the second layer according to white benefit get tactics set W2. The interests of party should be our interest minus the interests of the other, and so on to the root node. If matched to the endgame database features, if in our favor, mark the node and magnifying the

score, or mark the node as "not go", when pushing to the leaves of the tree, will receive specific situation, make it be a starting point to count interest, and then Back to the superior with marks. if let me always in the absolute fail position, search engine Immediately to searching to the superior, re adjust the pruning scheme, search the node which be pruned, get a new tactics set, re-search downward. If our still unfavorable, re-adjust the pruning scheme antorse until the search to let us have the opportunity to gain the optimal solution or the time runs out.

Search function is as follows:

```
int search(int deep, int alpha, int beta, int who){
    If(deep==0){ Get to the leaf node, returns the score }
    If(Matched to Endgame library){
        /* Marking for this node, if absolutely no profit, minus
        a lot of points or mark it 'not go' */
    }
    If(The score of backtrack-back does not make a profit){
        /* Get all tactics before be pruning, relaxed pruning
        scheme, get another set of tactics which doesn't be
        searched and then thought it. */
    }
    Get all tactics from generator of tactics;
    For(all tactics){
        Execute tactics;
        score=evaluation function(parameter)
        Get these after pruning and put into a set of tactics.
        revoke;
    }
}
```

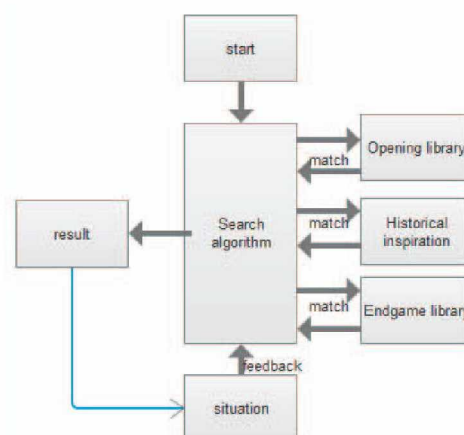


Figure 5 the program follow

A special situation :

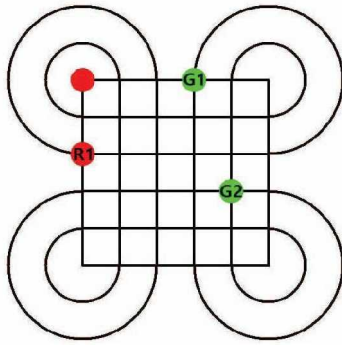


Figure 6 a special situation

In this situation, red move first. If R1 go to break G2, then G1 will break the R1. But the final result is not to our advantage. search engine gets the result of backtrack-back, it will find a new way like go to break G1, does not mean victory, but it will not fail. Also alpha-beta can do it easy. because the score on G2 higher than on G1, but search engine not use the score absolutely, it uses the final result know the way of R1 to G2 is a wrong.

5. Conclusion

The paper introduces the chess board representing method based on Unity 3D. The situation evaluation function is studied from five aspects, except for the alpha-beta algorithm, the opening library and the ending library are employed in the algorithm using as the foundation whether it is beneficial or unfavourable. The value is used in the pruning program to adjust the strategy and search the tactics deemed as unbetter again. The chess board style and the searching method used in this paper present good visual effect and strong performance. In the future, the much more comprehensive opening library and ending library should be studied.

Reference:

- [1] Xu Xin he, Xu Changming. An overview of computer game theory and methodology [M]. Shenyang: machine game research laboratory, Northeastern University, 2009
- [2]Zuo G, Wu C. A heuristic Monte Carlo tree search method for surakarta chess[C]// Chinese Control and Decision Conference. IEEE, 2016:5515-5518.
- [3] Zhang Y. A new storage structure design of Surakarta chess based on location[C]// Chinese Control and Decision Conference. 2014:3961-3964.
- [4] Zhang L, Ding L, Zhang L, et al. A Storage Structure and Capture Judging Algorithm of Realizing the Computer Game Program of Surakarta Chess[C]// Chinese Control and Decision Conference(ccdc. 2011:3442-3445.
- [5] Liu C, Wang J, Zhang Y. Search Strategy Research and Analysis for Surakarta Chess Game[C]// Chinese Control and Decision Conference. 2014:3362-3366.
- [6] Li S, Yuan X, Wen B, et al. The design and implementation of a Surakarta game system[M]. 2013.
- [7] Zhang L, Ding L, Li Z. The Design of Surakarta Chess Battle Platform in Computer Game[C]// Control and Decision Conference. 2013:2332-2335.
- [8]Wang Xiaochun.PC game programming: man-machine game [M]. Chongqing: Chongqing University press, 2002
- [9] Li Shuqin Li Jingbo surakarta game system assessment of [M]. function in Beijing: Computer Science and engineering Beijing Information Science and Technology University, 2012
- [10] Zhang Mingliang. A new game tree search algorithm and its application [M]. Suzhou: Soochow University, 20071001
- [11] Zhang Yanji. Bias equilibrium and the search algorithm of the game model based on surakarta[M]. Shenyang: Institute of systems science Northeastern University, 2010.