Abstract. Since it is hard to develop AI system and collect the development log from the large number of changing personnel in HfutEngine coded under traditional UVA base code, HeliosBase, a new base code, has been introduced in HfutEngine2014. After re-adjusting formation and evaluation weights preliminarily, HfutEngine2014 has been more competitive and get a more promising prospect.

1 Introduction

HfutEngine was a traditional 2d simulation soccer team, which is founded in 2002. From 2005, UVA BASE 2003 was used as our base code and our own AI methods was added to it. As a 2d Simulation team developed in 12 years, many AI methods has been dabbled in and some of them was discovered into team code by developers from generation to generation, such as Q-learning, neural network learning and C4.5 decision tree. However, the comprehensive strength of HfutEngine was still not escalated as the personnel fluctuation and restrict of development in UVA BASE 2003, which is an old base code without complete AI method designation system[1][2][3].

2 Background

The framework of traditional Hfutengine made use of evaluator to design to do something like shoot, dribble, etc and then to perform the relevant action whose evaluator mainly depended on the product of income and the probability of success which are estimated by environment beforehand to decide the action. Fig.1 shows the framework of traditional Hfutengine[2][4].

The evaluation system of HfutEngine is incomplete which only included pass and shoot. Fig.2 shows an example of passing evaluation system[12][13]
The traditional model calculates out the largest space to pass at first. The value of passing points, which are divided into several angles with lines for some passing to be evaluated, is trained uniquely by the relative details of passing success standard using method of inductive leaning[4].

Although such methods have been effective from 2005 to 2009, they can’t be designed more competitively in UVA BASE 2003 any longer facing with the rapid development of other base codes which is more intelligent with systematic AI method. Such a circumstance forces us to use HeliosBase, a more suitable and aggressive base code for our further research, as our base code and some fundamental modification of HeliosBase have been done in HfutEngine2014[1][3][9][10].

3 Formation Implementation of HeliosBase in HfutEngine2014

3.1 Delaunay triangulation method

The new Hfutengine2014 based on HeliosBase have implemented its formation at first with method of Delaunay triangulation.

Definition Let S be a set of points in the plane. A triangulation T is a Delaunay triangulation of S if for each edge e of T there exists a circle c with the following properties:
(1) The endpoints of edge e are on the boundary of c.
(2) No other vertex of S is in the interior of c.
If no four points of S are cocircular, then the Delaunay triangulation is unique. The 11 teammates are assumed to be 11 points belong to set S, and their positions are settled automatically according to the Delaunay triangulation so that each player can have the maximum number of choice to pass ball as well as avoid the situation of force snatch[6][7].
3.2 Initialization

step 1: Given a set V of N points within a rectangle, remove any points which fall on the vertices of the rectangle.
step 2: Partition the rectangle into approximately \( N^{\frac{1}{2}} \) bins (smaller rectangular regions).
step 3: Reorder the points by bins, starting at some bin and proceeding to neighboring bins.
step 4: Place the first point into the rectangle. Connect the point to the four corners of the rectangle to produce an initial triangulation[6][7][8].

![Fig. 3: introducing points by iteration](image)

3.3 Iteration

step 1: Input the next point to the existing triangulation. Connect this point to the vertices of its enclosing triangle.
step 2: Each of these quadrilaterals has an alternate diagonal. Swap a diagonal with its alternate, if doing so is required to satisfy the definition of Delaunay triangulation within the quadrilateral.
step 3: Each swap performed in step 2 may result in two new quadrilaterals that need to be tested. If one of these quadrilaterals doesn’t satisfy the definition of Delaunay triangulation, swap its diagonal with its alternate.
step 4: This swapping procedure may propagate further outward.
step 5: If all points in V have been used then stop, otherwise go to step 1 [6][7][8].

Fig. 3 shows the iteration procedure for newly introduced points.
Subroutine This subroutine locates the triangle $r$ which encloses the point $(x_0, y_0)$. 
(Comment: $X(r, i)$ denotes the $x$ value of the $i_{th}$ vertex of triangle $r$.)

$r \leftarrow$ last triangle created.

LOOP: DO FOR $I \leftarrow 1$ to $3$

$I + 1 \leftarrow I \mod 3 + 1$

IF $[(y_0 - Y(r, I)) \times (X(r, I + 1) - x_0) \gt (x_0 - X(r, I)) \times (Y(r, I + 1) - y_0)]$

(Comment: If $(x_0, y_0)$ is not in $r$, jump to the neighbor of $r$ which is in the
direction of the point.)

$r \leftarrow N(r, I + 1)$

GO TO LOOP

END IF

END DO FOR

RETURN ($r$)

END

4 Experiment Results

Table 1 shows an ideal result of competing with some teams in RoboCup2013, which proves that agent2d base code with the preliminary implements of formation in HfutEngine2014 is successful.

5 Conclusions

Surely, it is the optimized formation model with Delaunay triangulation in H-
futEngine2014 that ensures both stability and flexibility in either defense or
attacking strategy[9][10][11].

However, a single optimizing method can’t escalate the comprehensive ability
of whole team immediately and rapidly since all other aspects in agent2d base
code has not been combined and applied with high level strategy from tradition-
al HfutEngine developed in UVA base code and other outstanding and advanced
AI method proposed in multi-agent system nowadays.[1][13] In the coming time,
more implements and AI method will be considered to optimize HfutEngine2014
for an impressing achievement in the future competition.

<table>
<thead>
<tr>
<th>opponent</th>
<th>Ave Goals Scored</th>
<th>Ave Goals Conceded</th>
<th>win</th>
<th>draw</th>
<th>lose</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeliosBase</td>
<td>2.25</td>
<td>0.75</td>
<td>39</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>FC-Perspolis</td>
<td>0.64</td>
<td>2.1</td>
<td>22</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>HfutEngine2013</td>
<td>5.9</td>
<td>0.13</td>
<td>47</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1: result of competing with other teams
References

3. Aijun Bai, Haochong Zhang, Guanghui Lu, Miao Jiang and Xiaoping Chen: WrightEagle 2D Soccer Simulation Team Description2013 http://uurl.cc/cecm
5. Long Li.: Learning Based on Axiology and its Application in RoboCup2D Simulation. HeiFei University of Technology, (2009)