Namira Soccer 2D Simulation Team Description Paper 2018

Ehsan Asali[1], Farzin Negahbani[1], Saeed Tafazzol[2]
Mohammad Sadegh Maghareh[1], Shahryar Bahmeie[1]
Sina Barazandeh[1], Shokoofeh Mirian[2], Mahta Moshkelgosha[1]

[1] Department of Electrical and Computer Engineering & IT
Shiraz University, Shiraz, Fars, Iran
asali@cse.shirazu.ac.ir,farzin.negahbani@gmail.com
sadegh.maghareh@gmail.com,shahryarbahmeie@gmail.com
sina.br.shirazu@gmail.com,moshkelgoshamahta@gmail.com
[2] Qazvin Islamic Azad University, Iran
saeed.tafazzol@gmail.com,shokoofeh.mrn021@gmail.com

Abstract. In this paper, algorithms, methods and goals in Namira Soccer 2D Simulation team has been described. This article introduces approaches which we have utilized and implemented in Namira Soccer Simulation 2D team during its activity. Our defensive decision making and some useful software for debugging, training and holding matches have been discussed here. The base code that Namira has used is agent2d-3.1.1 and the library is librcsc-4.1.0 with little changes. Namira, soccer 2D simulation, defensive decision making, intelligent formation detection, debugger, trainer, TPAS.

1 Introduction

Namira Robotics team has been formed by students of Shiraz University and Qazvin Islamic Azad University (QIAU). This team is a combination of some members of Persian Gulf Soccer 2D Simulation Team in World Cup 2017 and some recently added students who study Software & Hardware Engineering and Information Technology at Shiraz University and QIAU. Persian Gulf Soccer 2D Simulation team is the core of Namira Soccer 2D Simulation team and their joint members could achieve 1^{st} place in IranOpen 2016 technical challenge, 5^{th} place in IranOpen 2016 and 2017 leagues and 6^{th} place in RoboCup WorldCup Competitions 2016. The technical leader of the team was a active member of Cyrus Soccer 2D, too; and participated in various competitions since spring 2012. He could achieve 5^{th} place in IranOpen 2017, 1^{st} place in IranOpen 2014, 5^{th} place in WorldCup Brazil 2014, 8^{th} place in WorldCup Netherlands 2013, 9^{th} place in WorldCup China 2015 and 1^{st} place in Kordestan 2013. He is now the TC member of the league for RoboCup WorldCup 2018 Competitions which will be held in Montreal, Canada. Moreover, one of the members of Namira have participated in the soccer2D competitions like Iran Open 2015 and Kazeroon Robotics Competitions 2015. Introducing novel approaches in soccer simulation,

increasing accuracy of agents, reducing search space for agents' decisions and using more artificial intelligent algorithms to make agents more dynamic are the most important goals of Namira team.
In the following, first our defensive decision making algorithm will be discussed; then, Namira's TPAS Software features and parameters will be demonstrated. Eventually, we will introduce our Trainer and DebugWatch software.

2 CDDMS Method

In traditional methods, each player makes his own decision for marking or blocking an opponent agent which would make team's defense line so messy. In order to solve this problem, we utilize a central decision making algorithm that can fix the problem of such chaos.
We proposed a method which is called Central Defensive Decision Making (CDDMS). In CDDMS, we assign the job of decision making about our team's defensive strategy to our goalie as our team leader in defense. Goalie has some specific features that helps him to be the best leader in defense. For instance, he should always keep looking at the ball and also dangerous opponents; consequently, his information about ball and players is the most updated among others. By having the most updated information, goalie assigns each teammate, except our offender and himself, an opponent player. The most effective factors for this decision are our teammates' current positions, opponents' current positions and our teammates' home positions which is their positions according to the current formation.
We divided the task of defense into two diverse sub-tasks. The first one is to find out the opponent player that each agent should cover. The second one is to choose between performing a block or a mark or none of them. In CDDMS, our goalie is responsible for the first sub-task in a way that he calculates the best teammate to cover dangerous opponents; then, by using say command, goalie tells players whom to cover and nothing more. After receiving the command, each player will understand his target opponent and based on ball position and opponent position he would select between marking or blocking actions. Figure 1 is an example of a message sent by our goalie to our teammates; this message contains a header and nine other cells which include the target opponent for our teammates with numbers 2 to 10 respectively. zero number in a cell means that there is no one to be marked or blocked for our teammate.

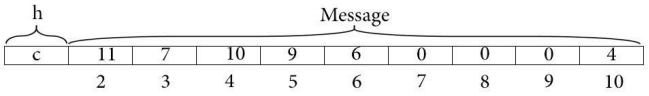

Fig. 1. A CDDMS message example

To mention the pros and cons of our algorithm, the most positive point here is an impressive reduction in defensive decision making chaos between teammates and the most negative point here is that we have some limitations and occasional problems like unpredicted delays or collision of messages in the network that cannot be fixed now. Figure 2 represents how CDDMS works in our team and arrows in the picture represent assigned opponents to our teammates.

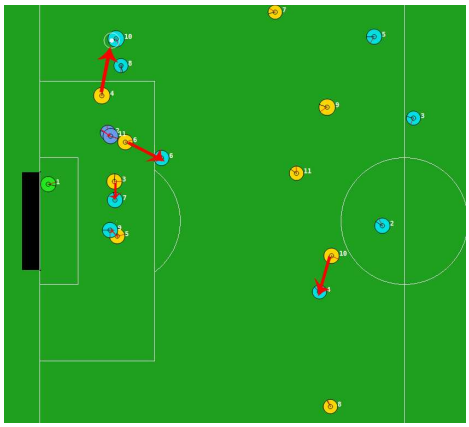

Fig. 2. Central Defensive Decision Making System

3 Software Development Project

In order to improve the quality and accuracy of actions in Namira, we needed some special software to facilitate the process of running, debugging and analyzing. To do so, we decided to implement three different software and we will discuss their features and capabilities in the following subsections.

3.1 Tournament Planning and Analyzing Software

Planning a tournament so that a number of teams play together based on a special order of facing each other is not always an easy task. More than that, current soccer server generates a logfile after each game that can be used to analyze the game. Handling the execution of the matches in a tournament and extracting the details of each team's performance in defense, offense and midfield and many other interesting statistics can be performed by our proposed software. Tournament Planning and Analyzing Software (TPAS) is a powerful tool written by Java language and is able to hold diverse types tournaments such as cup, league, world cup, a simple friendly match or even create arbitrary tournaments with arbitrary rules. Each tournament type has its own order of playing matches and corresponds to different real tournaments. For instance, in the cup mode,

each team will quit the competitions if it looses against its rival; winning a match leads a team to the next round and it can be continued until the final match that a victory there makes the team the champion of the tournament. Figure 3 shows an overview of the software graphical interface and a sample cup with three teams and the table and result of the competition. Because soccer 2D simulation league in WorldCup competitions uses an specific method for holding matches in its different rounds in the recent years, we have put a world cup mode in our software to handle its tournament matches easily.

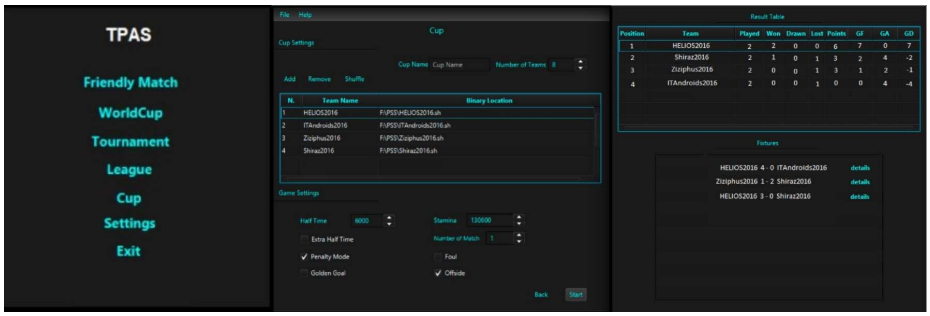

Fig. 3. A sample cup in Tournament Planning and Analyzing Software

TPAS lets us configure any tournament by just doing some clicks and in a well-designed graphical interface and keeps us away from server configurations complexities. Moreover, TPAS analyzes each game separately and generates some statistics about matches to help us learn more about our and opponents' strengths and weaknesses.

3.2 Trainer

We changed some features in the soccer server and soccer monitor and developed a trainer that aids us to test and improve our algorithms in different controlled situations against other teams. It has a json configuration file that gets parameters including ball position, ball velocity, ball owner, movement margin for each team, game area margin, number of players for each team, period of each episode, condition of ending the episodes, number of total episodes, etc. for the training which we want to be simulated. Figure 4 shows a two-by-two training in which the left team is in the right side teams defensive area and wants to take the ball and get score but the right team wants to push the ball away from its defensive area to make the situation safe.

4 DebugWatch

Debugging an agent's code is not easy during an implementation of a high-level algorithm. Most of the actions involve more than one cycle and need to

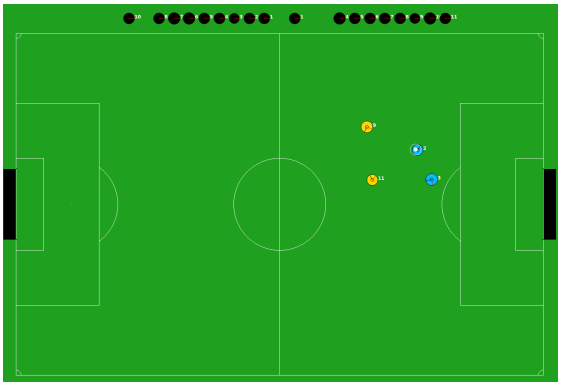

Fig. 4. Trainer view with two teammates and two opponents

be considered in a sequential timeline so that we find the time in which agent makes a wrong decision. Besides that, we need to check more than just one parameter and handling a lot of parameters and checking them in a row for different cycles requires special effort that is not constantly feasible. To make debugging process easier, we have designed a new software that helps you debug your problems during the match. It can be cooperate with soccer logplayer so as to synchronize the debugging process in both graphical and textual aspects. Figure 5 represents DebugWatch window. In this example, we have printed the x and y coordinates of each player. In traditional methods, we had to check each agents outputs separately in different files and it was a very time consuming task. By utilizing Debug Watch we are able to see all agents outputs in each cycle just in a moment.

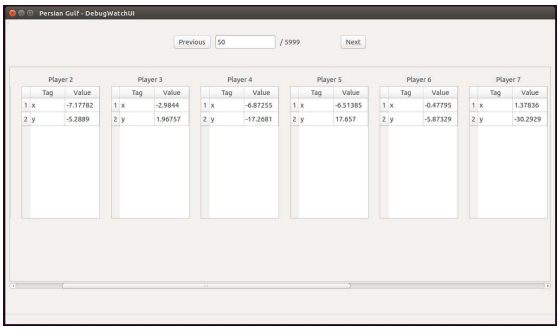

Fig. 5. DebugWatch (in this example) shows x and y coordinates of the player in its monitor for cycle 50

5 Conclusion and Future Work

In this description paper we introduced Namira soccer 2D simulation defensive strategy and three efficient software that pave the way to amend our team performance much easier and faster. Having a central defensive decision making system is crucial for keeping your defense line's discipline and arrangement and we tried to fill the gap of the base in this part at first.
We are working on a perfect analyzing tool that will be added to TPAS to extract the most possible information from every match. We need some information during the match such as our opponents' formation in different situations; we use a formation detection system but it has some defects that we are working on it. Moreover, our pass accuracy and defensive actions need to be revised and some corrections are needed in agents' movements.

References

1. E.Asali, M.Valipour, A.Afshar, O.Asali, M.R.Katebzadeh, S.Tafazol, A.Moravej, S.Salehi, H.Karami, M.Mohammadi, Shiraz Soccer 2D Simulation Team Description Paper 2016, The 20th annual RoboCup International Symposium, Germany, Leipzig, 2016.
2. N.Zare, M.karimi, A.Keshavarzi, E.Asali, H.Alipoor, A.Aminian, E.Beheshtian, H.Mola, H.Jafari, M.J.Khademian, CYRUS Soccer 2D Simulation Team Description Paper 2015, The 19th annual RoboCup International Symposium, China, Hefei, 2015.
3. R.Khayami, N.Zare, M.karimi, P.Mahor, F.Tekrar, E.Asali, A.Keshavarzi, A.Afshar, M.Asadi, M.Najafi, CYRUS 2D Simulation Team Description Paper 2014, The 18th annual RoboCup International Symposium, Brazil, Joao Pessoa, 2014.
4. R.Khayami, N.Zare, H.Zolanvar, M.karimi, P.Mahor, F.Tekara, E.Asali, M.Fatehi, Cyrus Soccer 2D Simulation Team Description Paper 2013, The 17th annual RoboCup International Symposium, The Netherlands, Eindhoven, July 1, 2013.
5. Akiyama, H.: Agent2D Base Code. http://www.rctools.sourceforge.jp (2010).
6. HELIOS2010 Team Description. In: RoboCup 2010: Robot Soccer World Cup XIV. Volume 6556 of Lecture Notes in Computer Science., Springer (2011).
7. A.Afshar, M.Zolghadri Jahromi, A.Hamzeh, A parameter-free method for detecting local communities based on attainable information, Advances in Social Networks Analysis and Mining (ASONAM), 2016.
8. E.Asali, M.Valipour, N.Zare, A.Afshar, M.R.Katebzadeh, GH. Dastghaibyfard, Using Machine Learning approaches to detect opponent formation, Artificial Intelligence and Robotics (IRANOPEN), 2016.
9. M.Karimi, M.Ahmadzadeh, International Journal of Advanced Research in Computer Science, International Journal of Advanced Research in Computer Science, 2014.