AUJ.T. 7(4): 175-179 (Apr. 2004)

Issues in Soccer Simulation Software Development

Peerapol Moemeng

Faculty of Science and Technology, Assumption University
Bangkok, Thailand
prl@s-t.au.ac.th

Abstract

This paper deals with the related issues in the development of soccer simulation software. A guideline to the development of soccer simulation software is given. This paper is suitable for computer science students who plan to develop another version of soccer simulation. The paper also discusses soccer software simulation in terms of computer science education. Thus, students who plan to develop soccer simulation software should review this paper and prepare the issues discussed in this paper for their project proposals.

Keyword: *RoboCup, artificial intelligence, computer programmers, game software.*

Introduction

Soccer is extremely popular among Thai teenagers and is increasing in its popularity. A lot of young computer programmers dream of having their own versions of soccer simulation games. This is a great motivator for many computer programmers, especially at the college level, to develop soccer simulation software. The motivation is good, but the work is hard. From the computer science point of view, soccer is often used in several experiments. One reason that soccer is chosen among other sports is its popularity; hence an increased willingness of the students to participate in this work. Three aspects that soccer serves in computer science will be discussed in this paper.

Game Software

Because of its popularity, millions of people choose soccer to play as their hobby. This also applies to soccer simulation games, such as Wining Eleven (Konami 2004), FIFA Soccer (2004), and Championship Manager (2004). These games are developed for console-based systems such as: Sony Playstation, Super Nintendo Entertainment System (SNES), and personal computers. As shown in Fig. 1, the screen shots from Wining Eleven show virtual players competing on a soccer field. The movements of the virtual players are captured and simulated from human players, and then represented in the game to gain realistic feelings. Fig. 2 shows screen shots from FIFA Soccer 2004, while Fig. 3 shows screen shots from Championship Manager. Different from other games, Championship Manager allows the player to act as the manager of a soccer team hence, enhancement of realistic motions is not necessary. We will discuss in the next section, Game Development Issues.

AI Testbed

Soccer is a distributed control problem with a collaborative and competitive environment. This is because players in a soccer match have their own decision-making process and actions. These abilities are not centralized into any specific unit, but instead distributed to every individual player. The goal of the game is that a team wins when it scores higher than another (this is claimed as a completive environment); while every player in the same team has to cooperate with other team-mates to accomplish the team's goal. (This is claimed as a cooperative environment.) Soccer involves many aspects in computer science. An organization known as RoboCup (2004) was formed to serve such a purpose. RoboCup turned out to be very exciting for researchers

and students to explore and experiment several issues in artificial intelligence and machine learning techniques, distributed control system, multi-agent system, task scheduling and planning. This will be discussed in the next section, Artificial Intelligence.

Strategic Planning (Simulation)

Software simulation is an effective method to observe the ongoing results and final outcomes of an expected hypothesis. The hypothesis is set and given with an input outcome. The input outcome to a simulation could reflect an actual situation. Therefore, a variety of input outcomes can be fed into the simulation. In the case of soccer, examples of input outcomes can be a set of player statistics, game strategies, events, and surrounding environments. The simulation presents a real world situation, and the outcome is affected by several factors. This is a highly complex system. This will be discussed in the next section, Software Simulation Development.

Fig. 1. Screenshots from Wining Eleven

Fig. 2. Screenshot from FIFA Soccer 2004

Fig. 3. Screenshot from Championship Manager

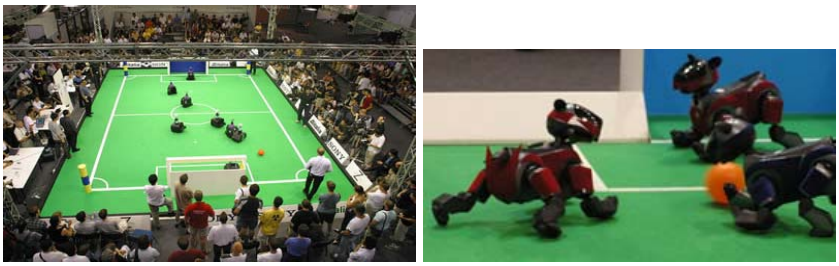

Fig. 4. Pictures from RoboCup league

RoboCup Software Simulation League

Besides the soccer league ran by robots, RoboCup also provides soccer leagues in software simulation. The RoboCup software simulation defines a set of standard protocols for all participants to attach their virtual teams to the system. The system is run by a central computer that will simulate events of the soccer game and distribute them to all the attached soccer player modules. The standard implementation can be found in Chen (2004). Fig. 5 shows the class diagram of the RoboCup software simulation. This class diagram defines the standard structure of the simulation, therefore external modules can be developed based on this class design and plugged into the simulation. The design of this simulation has been improved continuously, and will continue to be improved, as long as it is a world famous framework. The design reasons, of course, are rational, since they are approved by several software designers and developers. One very important question is, "Does the listed events and statistics represented in the RoboCup software simulation league reflect a real soccer game?" Of course, it does not completely represent the real world, but through years of research, conferences, and a lot of comments from participants, the standard protocols, events, and statistics have been improved time-to-time. RoboCup is a large and renowned framework. Therefore, any student that intends to create another soccer simulation should consider this framework and discuss rationally why a new simulation type should be created. Discussions afterwards will refer to RoboCup frequently.

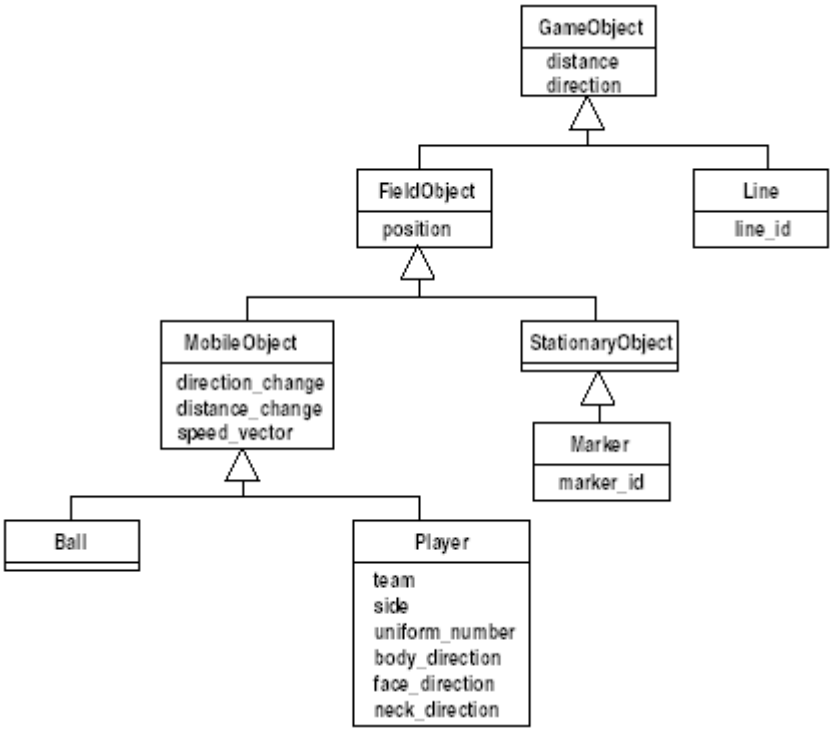

Fig. 5. A sample class diagram of the objects in the simulation

Game Development Issues

Game software is considered as one of the most difficult types of software. Besides meeting the requirements of the design itself, software must present impressions and attractiveness during the game play. Software testing can only explore faults and bugs in the program, but not a sense of excitement that could only be experienced by software users (game players). The history of game development is very long, starting from arcade games using a very low specification of computing devices, to the very high computing power of today. The samples of soccer games shown earlier, allows the game players to be a part of a virtual soccer game, as the player actually running in the field (action game) or even the manager who manages the team in a soccer league (strategy game). In action games, game players can only have control over one player at a time. At the same time, other soccer players have to make decisions and act accordingly. Here is a very challenging issue. How can the teammates cooperate with the soccer players controlled by humans? How can opponents manage to defend the team according to the pre-defined formation? Therefore, soccer games cannot avoid dealing with artificial intelligence. Other units in the game must more or less be able to think as well.

Artificial Intelligence Issues

Artificial intelligence (AI) is recently a hot issue in computer science. We may find that several commercial products can claim their capabilities with AI technology in their products. People who are involved in this field of research will look beyond the selling points. AI products sold today contain a limited set of features, and well-refined low risk functions, in order to prevent unexpected uncontrollable cases from the products' behaviour. Claiming that AI in soccer is a complex problem. Thus, completing the entire facts and requirements for soccer simulation involves many fields of research. For a clearer picture, let us look at soccer's nature. A soccer match contains two teams, 11 players for each team. The goal of each team is to score as many goals as possible, such that each player must act to serve the ultimate goal. An individual player in a soccer game can be modelled as a software agent (AGENT). Each agent makes decisions and acts. This is a form of distributed control problem. Distributed control problem is another huge research field being explored today. It is not limited to any specific machine learning techniques. Several techniques have been deployed for use within the soccer game, competing with others. A very complex and wide range software simulation is very challenging. Each research concerning this framework has to state its objective clearly on what it is going to prove. For a game development project, it may include the technologies found in existing research work or develop its own technologies. However, developing new technology may mislead the objective of the project. A game can be a combination of the existing technology such as, 'packaged software to go'. This is very different from a research project. Students planning to develop this kind of game will certainly need to state the objectives clearly. Making an agent to be smart can be done in a number of ways. One kind of agent is an agent that knows everything in soccer, and knows the best moves to take in each situation. Another kind of agent is the one that knows almost everything in soccer and is enhanced with learning capability. Once the agent is said to be a learner, it takes quite a lot of effort to define what learning means in soccer. For example, what kind of knowledge the agent should learn, what the data representation looks like to represent knowledge, what knowledge can be combined with the existing knowledge? And the questions involved will keep going. Machine learning is an attempt to enable machines to imitate human behaviour, and be able to improve performance in several ways. This is also a very challenging research topic.

Software Simulation Development Issues

A group of software programmers are not accredited as fine soccer players. Most of the time, programmers assume several formulae,

such as computing the chances that a player can control the ball, representing the strength of a player measured in numbers, taking several strengths and statistics of the players into calculation with no prove of representing actual facts. Software simulation has to deal with modelling problems. Representing the real world involves a series of works in several researches. It is very necessary to prove that the proposed formula really represents a real game. This issue could be overcome by conducting research on a real soccer player, or having an expert of soccer working on the project. As saying goes, 'easier said than done', we cannot expect every professional soccer player to have similar skills, and probably would not work with the programmers for the whole duration of the project. Making assumptions would not necessary convince other people, since everyone has different experiences with soccer. A game player could win the soccer simulation game, such as Winning Eleven, within a given time. If the player practises enough until they mastered the game, they could win the game with a lower-statistic team. Thus, the simulation is related to our discussion that develops a need to prove that the formula is close to the real situation.

Conclusion

In RoboCup, the formulae are provided, however, with a description of how and why. Reviewing the existing formulae may indeed save time, rather than re-development. So, any doubts that come up during the development are no good, could then prove that the development of new and improved versions could be better, since at least they could refine history. Not to mention the need for software engineering and the plan that every software construction is necessary. Software designs will help the developer to save a lot of time by not falling into programming traps and struggling with fixing bugs. In developing the term project or senior project, the software must certainly be complete. Students are required to be able to discuss issues related to the soccer game and the software development. The main point of this discussion is not only emphasizing the above issues, but also the software engineering. So students are reminded that only motivation is not enough, they also need hard work and dedication to succeed.

References

Chen, Mao. 2004. Users Manual RoboCup Soccer Server (for Soccer Server Version 7.07 and later).

Konami. 2004. Wining Eleven. Official Website of Kanami Computer Entertainment Tokyo, http://www.konamityo.com/we/

FIFA Soccer. 2004. EA Sports FIFA Soccer 2004. http://www.easports.com/games/fifa 2004/ home.jsp

Championship Manager. 2004. The Official Championship Manager Magazine Website, http://www.championshipmanager.co.uk

RoboCup. 2004. RoboCup 2004 Lisbon Official Website, http://www.robocup.org