Developing a fuzzy logic based game system

Article · January 2012

1 author:

Utku Köse
T.C. Süleyman Demirel Üniversitesi
147 PUBLICATIONS   **488** CITATIONS

Some of the authors of this publication are also working on these related projects:

Project   Spatial Cognition of Surrounding Objects by the B&VI People Using Image Processing and Ultrasonic Sensing View project

Project   "Zorunlu Ortak Derslerin Uzaktan Eğitim için İnteraktif Ders İçeriklerinin Hazırlanması (Preparing Interactive Lesson Contents for Distance Education of Mandatory Common Lessons)" View project

Developing a Fuzzy Logic Based Game System

Utku Kose

Directorate of Computer Center, Usak University, Usak 64200, Turkey

Abstract: The fuzzy logic, which is a technique of the artificial intelligence, rises as a result of studies based on simulating the human brain. It is a type of logic that recognizes more than simple true and false values. Linguistic variables can be represented with degrees of truthfulness and falsehood by using fuzzy logic. Like other artificial intelligence techniques, the fuzzy logic is used in many different areas. In computer game industry, it can be used to develop artificial intelligence based games. In this paper, the author discusses about usage of the fuzzy logic technique in computer games and developed a basic game based on the fuzzy logic. In this game, a computer controlled character can behave differently according to changing situations.

Key words: Artificial intelligence, fuzzy logic, computer games, game system.

1. Introduction

In computer science, artificial intelligence has an important place with its studies based on simulating human thinking and behaviors in machines. Artificial intelligence is used in many fields such as electronic, biology, physics, medicine, machine industry and military. Computer game industry is another field that artificial intelligence is used. Computer games have several uniquely identifiable systems that putting them together provides the expected entertainment. These systems include graphics rendering, playing audio, user input and game artificial intelligence [1]. With using existing artificial intelligence techniques, the game artificial intelligence system may be created. As a result, different game characters that have ability to behave itself can be created.

In the beginning, simpler methods and algorithms have been used to develop an artificial game character in computer games. But fast increase of processing power of computers and fast technical progress has given rise to development of applications using artificial intelligence in computer games. By using them and some other new developed algorithms, computer games have gone through drastic improvements. It seems as if Moore's law applies to computer games as well as processor speed [2]. With stronger computer systems and more complex game ideas, most advanced and complex methods of artificial intelligence such as neural networks, genetic algorithms and fuzzy logic have begun to use.

The topic of this paper is to discuss about developing a game system using an artificial intelligence technique—the fuzzy logic and developing a fuzzy logic based computer game. Controlling computer game characters can be done by using the fuzzy logic easily. Suitable input variables and output values may help to develop a fuzzy logic based game system. Some environmental variables can be used to develop more advanced game system. After developing, the game must be tested to know if the game and the fuzzy logic system used in are strong enough to reach aims.

The rest of the paper is organized as follows: First, the fuzzy logic technique and its basic rules are introduced. Following that, the author talks about developing process of the game. In this chapter, the fuzzy logic system used in the game system is explained first. After that, strategy in the game and the

Corresponding author: Utku Kose, lecturer, M.Sc., research fields: artificial intelligence, the chaos theory, distance education, computer education. E-mail: utku.kose@usak.edu.tr.

interface are introduced and test results of the game are reported. Next, the results got after testing the game are talked about. Importance and advantages of the game system, some additions and modifications that might be done are explained and discussed about the future of the artificial intelligence in computer gaming world.

2. Fuzzy Logic and Fuzzy Set Theory

Fuzzy set theory is first presented by L.A. Zadeh [3] in a seminal paper at University of California at Berkeley, USA. After that, the theoretical ideas about the fuzzy logic have been explained in many books [4-8]. The Fuzzy logic is developed from fuzzy set theory to reason with uncertain and vague information and to represent knowledge in an operationally powerful form. The main idea about developing fuzzy logic from fuzzy set theory is to build a conceptual framework for linguistically represented knowledge [9].

2.1 Fuzzy Logic and Crisp Logic

The fuzzy logic is an extension of crisp logic. In crisp logic, such as binary logic, variables are true or false. Certain classifications are made with using the crisp logic. A member can be in only one set. For example, for the relationship with the set "A", the characteristic function gives us 1 (true) if an element named "u" is the member of the set "A" or 0 (false) if an element named "u" is not the member of the set "A" in the crisp logic [10]. But in the fuzzy logic, this member may be in one or more sets. Membership degree of an element can be defined by using a characteristic function named membership function in the fuzzy logic. For the set "A" and the element named "u", this relationship can be defined as follows:

$$\mu A(u):U \rightarrow [0, 1] \qquad (1)$$

General set U is the fuzzy set of AU [10]. To understand differences between fuzzy sets in the fuzzy logic and crisp sets in the crisp logic, characteristic function graphics for both fuzzy and crisp sets are shown in Fig. 1.
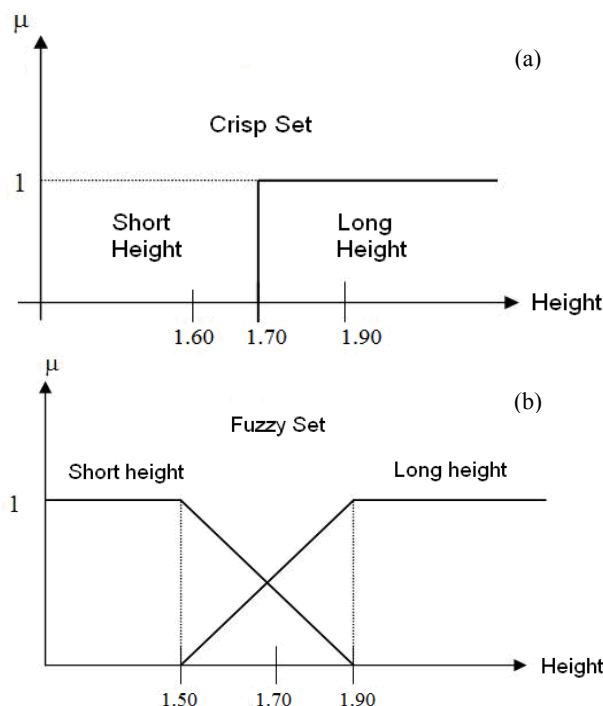

Fig. 1   (a) Crisp set and (b) fuzzy set.

2.2 Defining Membership Functions

In the fuzzy logic, after choosing fuzzy sets that will be used, a membership function for the sets should be assigned. A membership function is a typical curve that converts the numerical value of input within a range from 0 to 1, indicating the belongingness of the input to a fuzzy set [11]. Fuzzy sets in each input and output variable are defined by the assigned membership function. The membership functions for inputs and outputs have to be defined by expert experience [12]. Because fuzzy systems are based on thinking of the person who has an expert experience about the subject the fuzzy logic used in.

There are many different types of membership functions such as triangle, trapezoid and Gaussian. Different types of membership functions have different calculations on fuzzy sets. The most used membership function types: triangle, trapezoid and Gaussian functions are described in Fig. 2.

2.3 Fuzzy Linguistic Rules

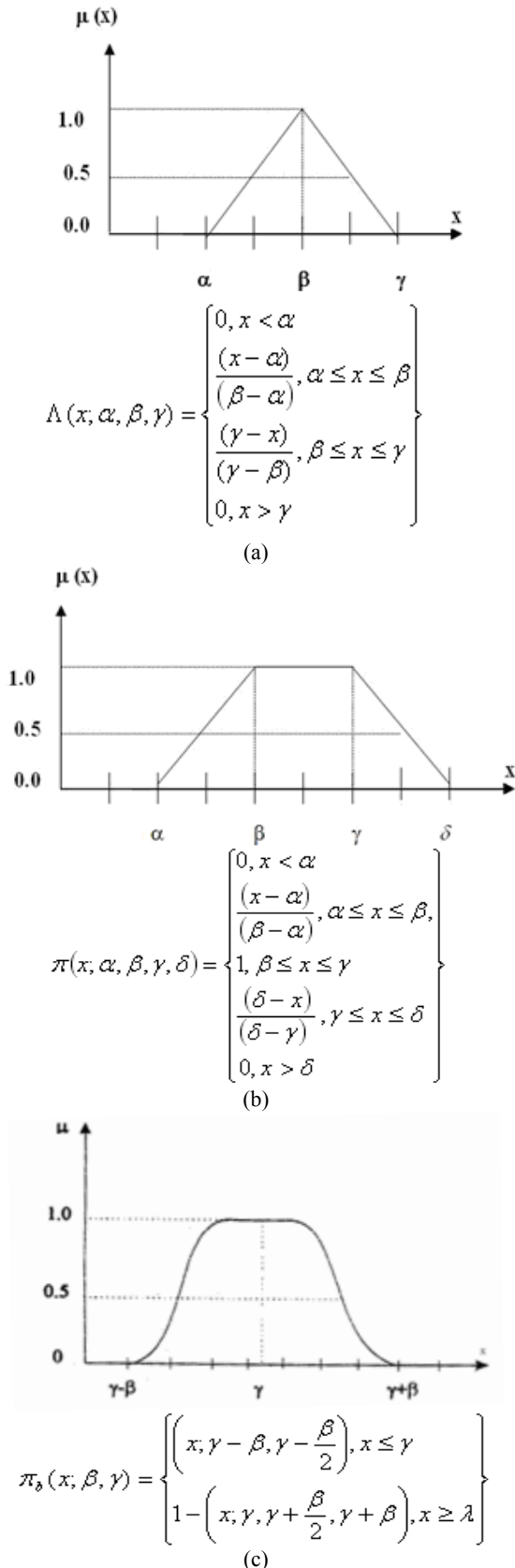Fuzzy linguistic rules transform the given input

$$\Lambda(x,\alpha,\beta,\gamma)=\begin{cases}0, x<\alpha\\\dfrac{(x-\alpha)}{(\beta-\alpha)}, \alpha\le x\le\beta\\\dfrac{(\gamma-x)}{(\gamma-\beta)}, \beta\le x\le\gamma\\0, x>\gamma\end{cases}$$

(a)

$$\pi(x,\alpha,\beta,\gamma,\delta)=\begin{cases}0, x<\alpha\\\dfrac{(x-\alpha)}{(\beta-\alpha)}, \alpha\le x\le\beta,\\1, \beta\le x\le\gamma\\\dfrac{(\delta-x)}{(\delta-\gamma)}, \gamma\le x\le\delta\\0, x>\delta\end{cases}$$

(b)

$$\pi_\delta(x,\beta,\gamma)=\begin{cases}\left(x,\gamma-\beta,\gamma-\dfrac{\beta}{2}\right), x\le\gamma\\1-\left(x,\gamma,\gamma+\dfrac{\beta}{2},\gamma+\beta\right), x\ge\lambda\end{cases}$$

(c)

Fig. 2   (a) Triangle (b) trapezoid and (c) Gaussian membership function types.

variables to the output variables. A fuzzy rule is created by writing a fuzzy if-then rule. For example, in a fuzzy logic system that uses three inputs for one output, a simple rule may be written as follows:

IF (age IS VERY OLD) AND (height IS TALL) AND (weight IS HIGH) THEN (result IS ILL)

For each combination of input variables, if-then rules are written to define outputs. These rules are combined in a table named rule table. By reading the created rule table, each output for different input variables can be observed.

2.4 Defuzzification

The fuzzy logic system gets the output from given input variables. This output is also a fuzzy set. In order to use the output value in real life, its fuzzy value should be translated into real values. This translation is named as defuzzification [13]. There are many methods of deffuzification such as centroid, centre of sums, mean of maxima and left-right maxima.

3. The Fuzzy Logic Based Game

In this paper, a fuzzy logic based game application developed. In the application, two virtual game characters are used to create a fuzzy logic based computer game system. One of these characters is controlled by the computer user that is using keyboard and mouse. This character is used to create a variable environmental condition for computer controlled character. The other character is controlled by the computer and behaves according to some actions states. Actions taken by computer controlled character are set by two variables: ammo and health. These variables are changed after any damage that computer user made on the computer controlled character or bullets that the computer controlled character spent. If the computer controlled game character does not contact with the user controlled game character, ammo and health levels may increase according to action states. Some elements located in game ground may be used by computer controlled character in some different action states. Action states are set by values that fuzzy logic system

calculated. The fuzzy logic system uses ammo and health level information and set the action that will be taken by the computer controlled character. Fig. 3 below shows an overview of the fuzzy logic system in the game developed.

3.1 Fuzzy Logic System in the Game

As mentioned before, ammo and health level information are fed into the fuzzy logic system as inputs and action states are calculated as outputs. There are three different fuzzy techniques, known as the Mamdani [14], Tagaki-Sugeno [15] and Tsukamoto [16] fuzzy models. In this work, fuzzy logic Mamdani is used as the technique for the fuzzy logic system. For inputs and output, the triangular membership function is used.

Ammo and health inputs has five membership sets: very low, low, medium, high and very high. Fig. 4 shows membership sets for inputs.

For output, five action states define membership sets. These action states are: hide, run away, stop, walk around and attack. Higher ammo and health levels mean actions aimed more on attack. Lower ammo and health levels means actions aimed more on run away and hide. Values of each action state calculated by the fuzzy logic system define strength of the actions that will be taken by the computer controlled game character. Fig. 5 shows membership sets for output.

3.2 Generating the Rules

It is extremely important to write rules which cover every possible set of inputs. Because of any missing rule, the fuzzy logic system will not know how to respond. There will be no value for the output in that case.

Rules for the output are written using If-Then statements. Having five membership sets for each input variable (ammo and health), $5 \times 5 = 25$ different rules must be written for output in this work. Some rules that are written for the game system are given below:

IF (ammo IS VERY LOW) AND (health IS VERY HIGH) THEN (action IS STOP)
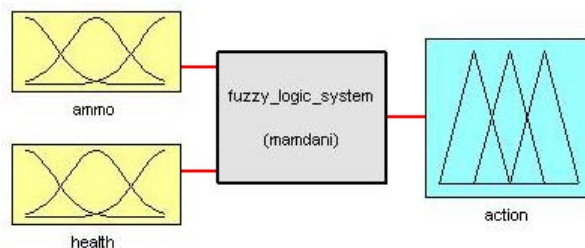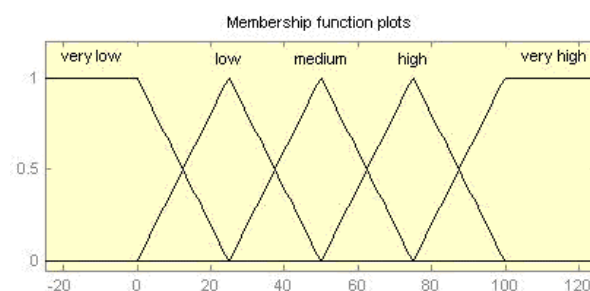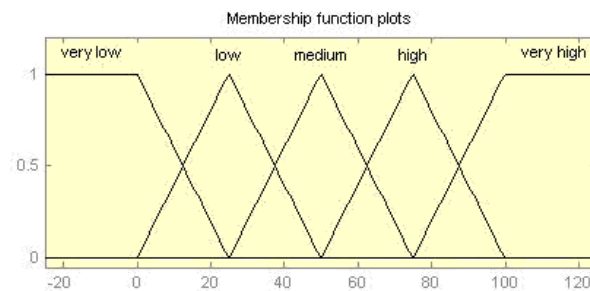

Fig. 3 Overview of the fuzzy logic system.

(a)

(b)

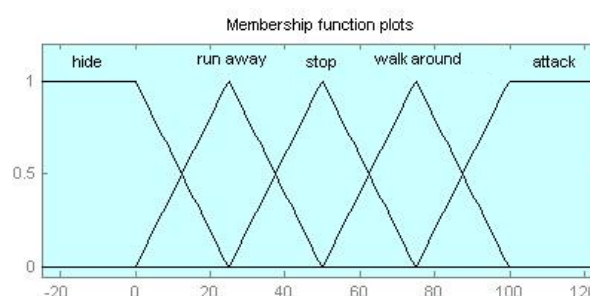**Fig. 4 Membership sets for (a) ammo and (b) health inputs.**

Fig. 5 Membership sets for output.

IF (ammo IS LOW) AND (health IS HIGH) THEN (action IS STOP)

IF (ammo IS MEDIUM) AND (health IS LOW) THEN (action IS RUN AWAY)

IF (ammo IS HIGH) AND (health IS VERY LOW) THEN (action IS RUN AWAY)

…

If all of the rules are combined in a table, a rule table

is gotten as shown in Fig. 6. As seen on the rule table, ammo membership sets are located on top of the table and health membership sets are located on left side. For example, if ammo input is very low and health input is medium, the rule "run away" for action state of the computer controlled character is gotten.

Effects of the inputs on outputs can be shown in two graphics in Fig. 7. These graphics created using Fuzzy Logic Toolbox-FIS Editor-Surface Viewer in Matlab R2008a. The 3D graphic created using plot type "surface" and the 2D graphic created using plot type "Psuedo-Color". On the 3D graphic, the output values in 3D surface according to health and ammo values can be seen. On the 2D graphic, ammo input values are located on *x* coordinate and health input values are on *y* coordinate. According to these values, the output values are shown in different color levels between red and yellow.

By using Rule Viewer in FIS Editor, the rules used in the fuzzy logic system and their reactions according to inputs can be watched easily. For example, two different calculations are shown in Fig. 8. Fig. 8a shows results for ammo at level 83 and health at 22 while Fig. 8b shows results for ammo at level 17 and health at 52.

3.3 Action States in the Game System

After getting output values according to ammo and health level information, the computer controlled character uses them to start to show suitable behavior. If the character spends its ammo too much or gets too damage, its behavior starts to move from attacking oriented actions to hiding oriented actions. If any action state has higher output value than others, the character knows that the action is more important than others. Also the action must be done stronger if it has a higher value. The computer controlled character behaves for different action states as explained below:

• If the action state for computer controlled character is "attack", it stars to fire at user controlled character and be more sensitive on moves of user controlled character. If the attack value is higher, the

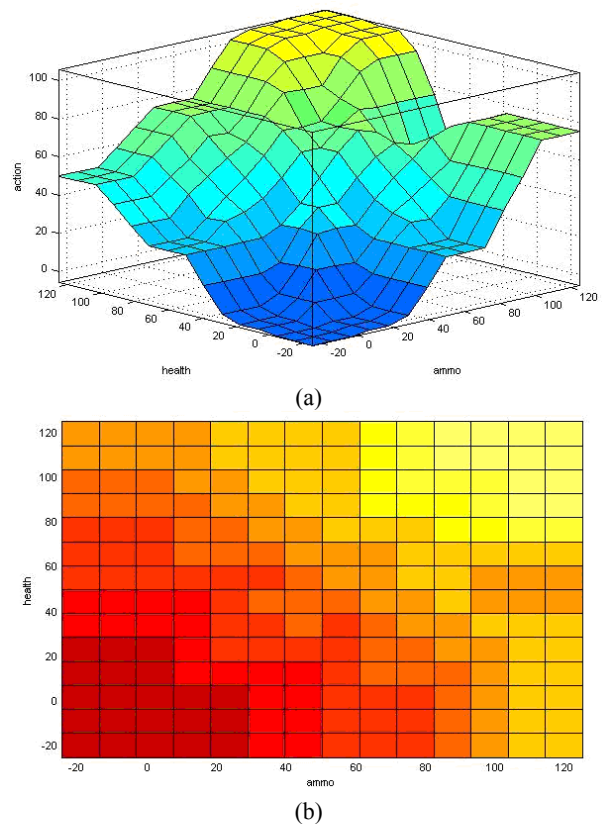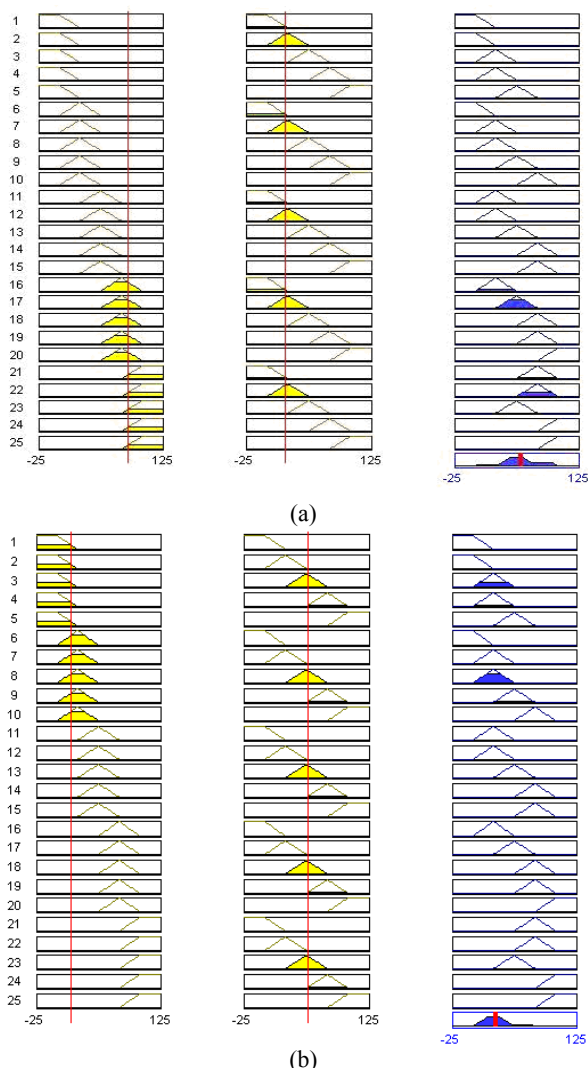|	Ammo				
Health	**Very Low**	**Low**	**Medium**	**High**	**Very High**
Very Low	HIDE	HIDE	RUN AWAY	RUN AWAY	STOP
Low	HIDE	RUN AWAY	RUN AWAY	STOP	WALK AROUND
Medium	RUN AWAY	RUN AWAY	STOP	WALK AROUND	WALK AROUND
High	RUN AWAY	STOP	WALK AROUND	WALK AROUND	ATTACK
Very High	STOP	WALK AROUND	WALK AROUND	ATTACK	ATTACK

Fig. 6   The rule table.

(a)

(b)

Fig. 7   Effects of the inputs on outputs as (a) plot type surface and (b) plot type pseudo-color.

character stars to move faster and spend more bullets. Predictably, as a result of spending too many bullets, its action state will start to move hiding oriented actions;

• If the action state for computer controlled character is "walk around", it starts to walk around as controlling the position of user controlled character. If the value is higher, the character stars to move faster and moves different locations of the game ground;

(a)

(b)

Fig. 8 Two example for reactions of the rules according to (a) ammo at level 83 and health at 22 and (b) ammo at level 17 and health at 52.

• When the action state is "stop", the computer controlled character stops moving sometimes. The higher output value means longer time the character waits. Stopping moving does not mean having an unprotected situation. The character keeps its eyes open for any dangerous actions that will be taken by user controlled character and defend itself in any dangerous situation;

• The action state "run away" means ammo and health levels of the computer controlled user are under normal level. The character starts to be more sensitive on bullets fired by user controlled character and defend itself by moving faster. It tries to take no damage by

user controlled character;

• The worst action state for computer controlled character is "hide". There are four shelters on game ground. The computer controlled character may use one of them to protect against bullets. The computer controlled character thinks about its position on game ground according to shelters and user controlled character and runs to behind of the best located shelter. The higher value means faster the computer controlled character runs to the shelter.

3.4 Gaining Ammo or Health

For computer controlled character, it is possible to gain ammo or health according to action situations. For five different action states, the character gains ammo or health in different percents. Faster actions reduce gaining percentage of ammo and health. The computer character does not gain ammo or health all time. It also does not gain ammo and health at same time. This function may be different for different action states and output values.

3.5 The Game System Interface

As a game system interface, an interface which is suitable for aims of this work was developed. Different programs like Adobe Flash, Adobe Fireworks and 3D Studio Max are used to programming and designing of the system. The program code of the game is written in Action Script of Adobe Flash. By using Action Script, a C programming language based game system is programmed. Two screenshots from the game are shown in Figs. 9-10.

Game characters are designed and rendered in 3D Studio Max 2009. V-Ray Renderer is used to get best result. Four shelters used in the game are also designed and rendered in 3D Studio Max. Game ground is designed as suitable to work aims. Some indicators are used to show working fuzzy logic system and input values. Ammo and health indicators of the characters are shown in Fig. 11 and the action state indicator is shown in Fig. 12.

Fig. 9   A screenshot from the entrance screen.
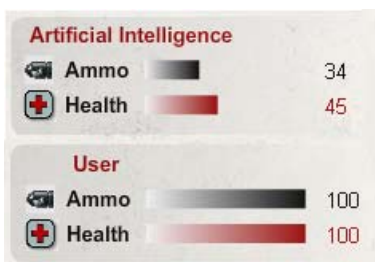

Fig. 10   A screenshot from playing screen.

Fig. 11   Ammo and health indicators for the characters.

Fig. 12   Action state indicator.

3.6 Testing the Game System

After development, the game system is tested many times for different situations. By adding a small code to the programming interface, ammo and health values of the computer controlled character are set to "10". After starting the game the computer controlled character is started to move without firing. Moves of the computer controlled character are watched in this situation. As a result, there is not any error in changing action situations with no changing environmental factors. The developed game is played many times in normal mode without editing any programming code or design. The computer controlled character is successful in aims of this work.

4. Conclusions

In this paper, the author discusses about developing a game system using an artificial intelligence technique—the fuzzy logic and developed a fuzzy logic based computer game. As shown by the results of the work, it is possible to control computer game characters by using fuzzy logic easily. The rules used in this work give a chance to computer character to think about environmental situations and act by itself in different states. The action states used in this work are main action situations that are preferred in many important computer games. The chosen action states show different programming algorithms in developing a wide oriented game system. The fuzzy logic system developed for the game show the importance of the rules used in and its connections with the computer controlled character. If there is a missing rule in the system, the character will not know what to do. Also, having the correct, simpler rules and properly formed membership functions can save significant development time.

In developing a fuzzy logic based computer game, more inputs and outputs may be used for the games which have complex game strategy. It is also important to use a suitable membership function and sets to develop a more sensitive or complex computer game.

In programming the artificial intelligence of the game, different techniques like neural networks and genetic algorithms may be used. Different artificial intelligence technique gives different results in testing. So, different techniques must be tested in developing a computer game.

The artificial intelligence has an important role on developing better technologies in many different areas. The author thinks that the artificial intelligence is also a milestone in computer gaming industry. As can be seen, the artificial intelligence and its techniques will be the next revolution in gaming industry. In the future, many artificial intelligence techniques which can achieve the goal of having more realistic and extraordinary games will be developed.

References

[1]   C. Weddle, Artificial intelligence and computer games, Dissertation, Computer Science, Florida State University.

[2]   J. Wexler, Artificial intelligence in games: A look at the smarts behind Lionhead Studio's "Black and White" and where it can and will go in the future, Dissertation, Computer Science, University of Rochester, May 7, 2002.

[3]   L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338-353.

[4]   H.J. Zimmerman, Fuzzy Set Theory and Its Applications, Allied Publishers Limited, New Delhi, 1996.

[5]   R.C. Berkan, S.L. Trubatch, Fuzzy Systems Design Principles, Standart Publishers Distributors, New Delhi, 2000.

[6]   S.V. Kartalopoulos, Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications, Prentice-Hall of India Pvt. Ltd., New Delhi, 2000.

[7]   G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall of India Pvt. Ltd., New Delhi, 2000.

[8]   C.R. Bector, S. Chandra, Fuzzy Mathematical Programming and Fuzzy Matrix Games, Springer, Berlin, 2004.

[9]   P. Mahonen, T. Frantti, Fuzzy classifier for star-galaxy separation, The Astrophysical Journal 541 (2000) 261-263.

[10] N.K. Kasabov, Foundations of Neural Networks, Fuzzy Systems, and Knowledge, The MIT Press, U.S.A., 1998.

[11] A. Majumdar, A. Ghosh, Yarn strength modelling using fuzzy expert system, Journal of Engineered Fibers and Fabrics 3 (2008) 61-68.

[12] S. Kutuva, N.P. Reddy, Y. Xiao, X. Gao, S.I. Hariharan, S. Kulkarni, A novel and fast virtual surgical system using fuzzy logic, in: Proceedings of the IADIS International Conference Computer Graphics and Visualization 2006 May 15-19, 2006, pp. 277-281.

[13] P. Dadone, Design optimization of fuzzy logic systems, Dissertation, Electrical Engineering, Virginia Polytechnic Institute and State University, May 18, 2001.

[14] E.H. Mamdani, Applications of fuzzy algorithms for control of simple dynamic plant, in: Proceedings of the Institution of Electrical Engineers, 1974, pp. 1585-1588.

[15] M. Sugeno, G.T. Kang, Structure identification of fuzzy models, Fuzzy Sets and Systems 28 (1988) 15-33.

[16] Y. Tsukamoto, An approach to fuzzy reasoning method, in: M.M. Gupta, R.K. Ragade, R.R. Yager (Eds.), Advance in Fuzzy Set Theory and Applications, North-Holland Pub. Co., Amsterdam, 1979.