Journal of Next-generation Convergence Information Services Technology Vol.4, No.1, June (2015), pp. 9-18
차세대컨버전스정보서비스기술논문지

Generation Technique of Dynamic Monster's Behavior Pattern Based on User's Behavior Pattern Using FuSM

Dong-Joo Kim[1], Jung Yoon Kim[2]*

Abstract

One of the most important items in game development is implementation of difficulty by AI (Artificial Intelligence) of NPC (Non Player Character). In particular, as a game, a form being used by multiple users, and has difficulty in its implementation.

In this study, its objective is to increase a sense of immersion to game users by generating and providing NPC dynamic behavior pattern more suitable for a game in a way of grafting FuSM algorithm through utilization of finite state machine (FSM) and fuzzy algorithm that are artificial intelligence technique in hostile NPC behavior pattern algorithm among game difficulties.

Keyword : Game monster behavior patterns, Finite State Machine (FSM), Fuzzy algorithms, FuSM

1. Introduction

If people were asked to single out the most influential element since invention and development of computer, a lot of people would indicate it as 'Game'. In various platforms including console game, online game and computer game, game has been developing rapidly. The most contributory on the development process of a game is the NPC (Non Player Character). As AI NPC moved voluntarily without directly giving command from users in the game is developed. The behaviour pattern of NPC has been diversified and consequently, a sense of immersion of the users could be further increased while difficulty of the game being controlled by using always identical or random number that are systemized[1][2]. In order to induce users consistently, unlike one-time game, players were made not to lose interest during games by controlling difficulties such as AI strength, behavior pattern and scale to be matched with user's capability. Actually, most of AI in present games control game difficulty by matching with user's capability in real time during game play. In particular, in case of offline games, as one(1) character player is set as main principal, difficulty is controlled by matching with relevant

1 Graduate School of Advanced Imaging Science, Chung-Ang Univ, HeukSeok-dong, DongJak-gu, Seoul, Korea
e-mail : pkmstdj@gmail.com
2 School of Game, Chungkang College of Cultural Industries, 389-94 Chungkang gachang-ro, Majang-Myeon, Icheon-si, Gyeonggi-Do, Korea.
e-mail : kjyoon79@gmail.com (Corresponding author)

Received(March 07.2015), Review (March 20.2015), Accepted(June 30.2015)

ISSN: 2384-101X JNCIST

player's level, and so, a sense of immersion has been increased by a couple of times compared with the past. However, in case of our country, the so-called 'online game' in which a game is played with other players by using computer network rather than single play becomes a mainstream. Contrary to other games, in the case of online game, as either the obstacle or a friendly forces or both are actually of different player. Variable is more frequently taken place than offline in various aspects. A person who had once played RPG (Role-playing game), MMORPG (Massively Multi-player Online role-Playing Game) would understand this trait[3].

As game increases one's own level of the user through hunting and quest, hunting grounds fit the players level also. In addition, hunting grounds on each level does not overlap and its geographical area is relatively wider than other levels on performing various quests and as such, in case of players who are not accustomed to the game, they would often accidentally visit hunting ground far exceeding one's own level and because of this, players often faces game over due to preemptive attacks style by more aggressive monsters that attacks first even though it’s not being attacked. If moving range of monster should be limited due to this reason, players playing with leaving distance with monsters would experience difficulties. Due to this, 'No way' may be talked about or it may be regarded as one minor variable being overlooked by saying 'It can't be that bad'. However, even though not a magnificent element, each one of these detailed elements affects pleasure of game considerably. In this study, for ensuring preciseness of this minor but important part, behavior pattern algorithm of monster grafting it (visual field, movement distance) with Fuzzy algorithm and finite state machine (FSM) would be suggested in RPG game.

2. Related Work

2.1 Paragraph Title

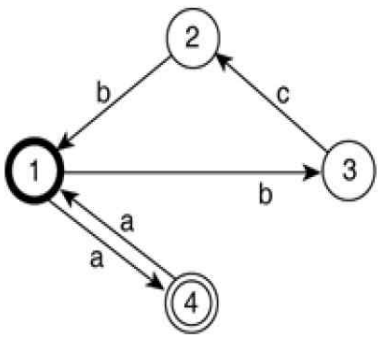
NPC game is mainly divided into three categories: Friendly forces NPC directly supporting players, neutral NPC that plays a role of store, story development and finally, hostile NPC competing through battle with player. Among these hostile NPC (hereinafter called monster) has the closest relation with difficulty. As monsters play a role as obstacle on the way of moving to game goal by player, if it is

too weak or strong by showing serious gap with players, pleasure (fun) of game would be spoiled. That's why a lot of games being available today use a method of changing game difficulty in real time during game by analyzing capability of each player[2][4]. One of the methods of controlling difficulty is behavior pattern algorithm. Among these, as an algorithm being mainly used for behavior pattern of game AI, FSM is available.

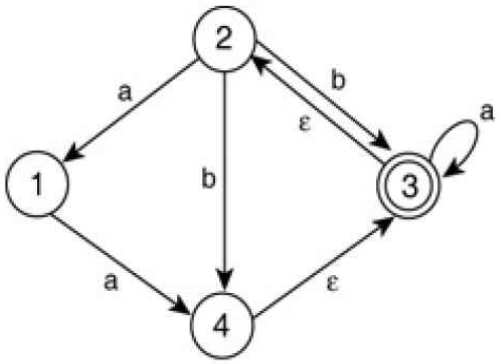
2.2 FSM : Finite-State Machine)

FSM could be called as an algorithm of being switched to a state of next stage being connected with current state when a specific condition is satisfied while having one of various states[2][5-7]. Deterministic FSM has 1 output value as result value being received under current state. When arriving at the last state after being implemented sequentially, its operation is stopped. [Fig. 1] briefly shows an image of deterministic FSM. Non-deterministic FSM of [Fig. 2] enables circulation without its state being changed as designated. After repeat through circulation, more optimized value could be obtained. If Ai should perform simple work like fan, toaster, above method would be enough. However, in case Ai in game, it has various complicated states and this directly leads to game immersion of player, that is, fun (pleasure). As these complicated actions are hard to be expressed as a few states, showing complicated expressions by representing various states as hierarchical tree in nest is hierarchical FSM. [Fig. 3] shows its example.

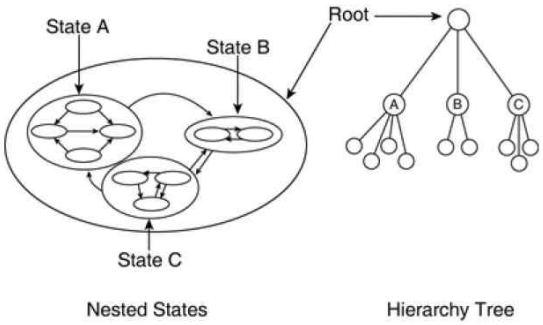
[Fig. 3] shows that in a state, multiple FSMs are nested hierarchically. Like this, expressing one big state by combining various result values is FSM[5-7].



[Fig. 1] Finite State Machine changed on Input Character



[Fig. 2] Non Deterministic Finite State Machine



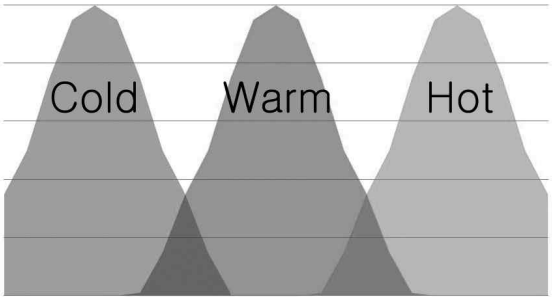
[Fig. 3]Hierarchical Finite State Machine

2.3 Fuzzy Algorithm

Fuzzy algorithm is an algorithm designed to express a degree of 'ambiguity' of natural language being used by us. It is also called as grey algorithm obtaining relevance degree, not true or false, of a certain topic. As shown on [Fig. 4], as it is an algorithm obtaining ambiguity of central area, it is mainly used for system control, management of a system rather than finding out a certain optimal value and adjectives such as very, little, slight, fair could be applied. This is an algorithm started from a concept that when a person is faced with a certain situation, such situation is not always able to be determined by dichotomous thinking. This algorithm is so much close to humans and it demonstrates its value in development of AI similar to humans in particular[6]. As AI itself was made by resembling human being, its completeness is increased as it gets closer to human being to maximum. As AI determines

ambiguity on its own accord, we could see AI more closer to human image[6][7].

In this study, behavior patterns of AI are intended to be designed more intelligently by using Fuzzy State Machine (hereinafter called FuSM) that is an algorithm combining above 2 algorithms. Basic concept of FuSM is similar to that of FSM. Ambiguous approximate values could be processed more conveniently, not dividing true, false just by exact value being calculated after applying Fuzzy algorithm in input value of FSM that is an algorithm controlling various states through input value (event).

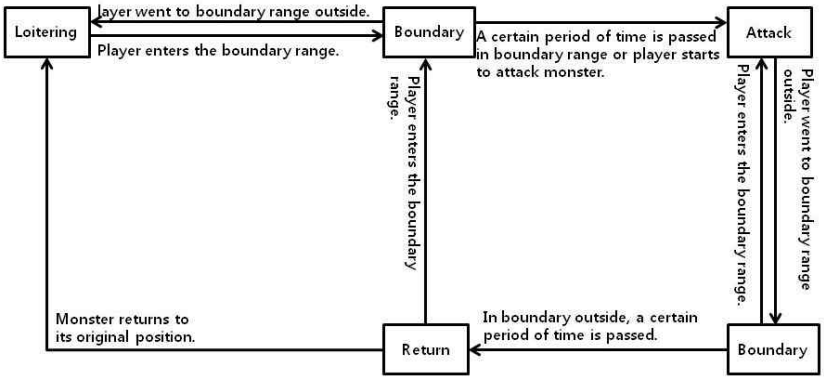


[Fig. 4] Fuzzy algorithm

3. Design and Result

3.1 Experimental Design

Algorithm having behavior pattern of monsters was designed as shown on [Fig. 5].



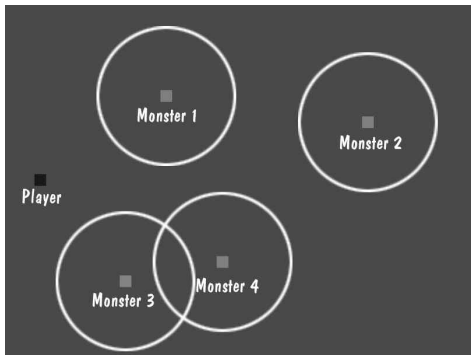
[Fig. 5] monster behavior pattern

In [Fig. 5], it could be seen that behavior 'boundary' has two boundaries. Boundary on top of the middle is called as boundary 1 and boundary under the middle as boundary 2. The reason why boundary of same behavior is divided into two is that there is a difference between the boundary until monster returns after attack and behavior of boundary before attack. Monster that loitered at first movable area moves around in its movable area by continuously maintaining loitering condition until player approaches boundary range inside. In the meanwhile, in case of being encountered with player approaching boundary range inside, it is switched to boundary 1 state and after stopping movement, it stares at player. In case that player remains at boundary range until a certain period of time is passed, monster is ready to attack and boundary range that becomes an attack state is increased to maximum range. Monster that becomes an attack position starts to attack player and boundary range is gradually decreased to minimum range at the same time. Or if attacked by player under boundary 1, 2 state, it is immediately switched to attack state and likewise, boundary range is gradually decreased to minimum range. If attacked by player under attack state, boundary range is increased to maximum range again. If player is deviated from boundary range, monster is switched from attack state to boundary 2 state and it standbys at relevant position for a certain period of time. In case that enemy player does not enter boundary range or start to attack even after passing certain period of time, monster returns to its original position after being switched to return state. Afterwards, monster is switched to loitering state again and loiters at movable area. Switch time to attack state is delayed or boundary range is adjusted by controlling boundary range and contact range, time of player so that true, false will not be reversed by an exact number after applying Fuzzy algorithm at the section where each state is changed[6].

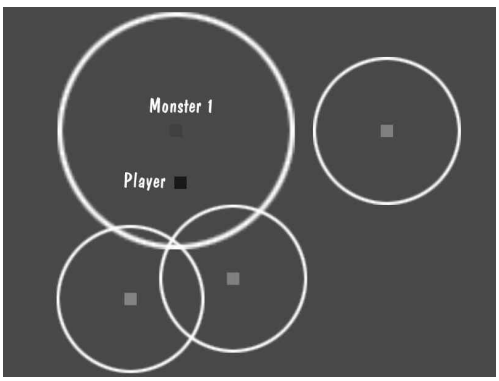
3.2 Result of experiment

[Fig. 6-9] shows demonstration image of a program applied by algorithm after manufacturing it by using unity game engine. [Fig. 6] shows early disposition state of player and monster 1-4. After implementation, player moves to the left and starts to attack monster 1. Attacked monster 1 chases at player after it being switched to attack state immediately. [Fig. 7] shows that after being switched to attack state, boundary range is increased to maximum and player is chased. [Fig. 8] shows that monster 1 is switched to boundary 2 state after player escapes from boundary range of monster 1 and monster 2 is switched to boundary 1 after player contacts boundary range of monster 2. [Fig. 9] shows that after

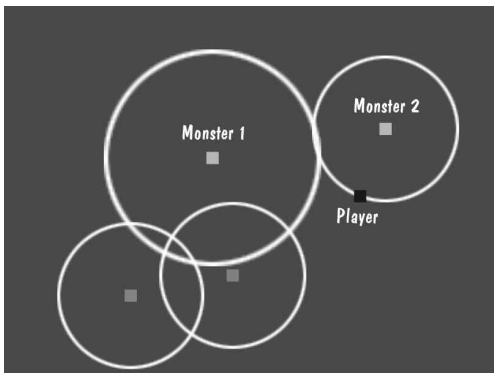
passing certain period of time, monster 1 is switched to loitering state again after returning to original position and monster 2 starts to attack player in boundary range for a certain period of time. In this experiment, time required for moving from start point to arrival point by player is total 3 seconds. Identical experiment before/after disposition of monster was respectively performed for 7 times after dividing before, after algorithm application by setting player level and monster level differently and increasing number of monster from 4 by 1 per day. As result value of output, switch frequency of monster to attack state, total time of maintaining attack state and frequency of direct approach and attack by monster were shown on [Table 1-3].



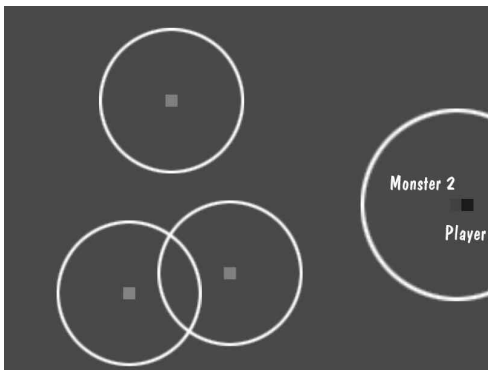
[Fig. 6] Player game start state



[Fig 7] Monster 1 state to attack the Player that came into this boundary range



[Fig 8] After the Player is out of the boundary range of Monster1, state that entered the boundary range of Monster2



[Fig 9] State Monster1 is returned to the original position , Monster2 to attack the Player

[Table 1] Number of times the monster has switched to attack state

	Before application of algorithm	**After application of algorithm**
1 time(4 heads)	2 times	2 times
2 times(5heads)	3 times	3 times
3times(6 heads)	3 times	3times
4 times(7heads)	4 times	3times
5 times(8heads)	5 times	4 times
6 times(9heads)	7 times	4 times
7times(10heads)	8 times	4 times

[Table 2] The amount of time that monster attacked

	Before application of algorithm	**After application of algorithm**
1 time(4heads)	1.1s	1.0s
2 times5heads)	1.7s	1.6s
3 times(6heads)	1.8s	1.6s
4 times(7heads)	2.1s	1.7s
5 times(8heads)	2.2s	2.0s
6 times(9heads)	2.6s	2.2s
7times(10heads)	2.7s	2.2s

[Table 3] Number that received directly damage the monster

	Before application of algorithm	**After application of algorithm**
1 time(4heads)	1 time	1 time
2 times(5heads)	1 time	1 time
3 times(6heads)	2 times	1 time
4 times(7heads)	3 times	2 times
5 times(8heads)	3 times	2 times
6 times(9heads)	3 times	3 times
7times(10heads)	4 times	3 times

4. Conclusion

In this study, more realistic and diversified behavior patterns were applied by providing behavior pattern of dynamic monster to game users by utilizing FSM and Fuzzy algorithm. Therefore, as shown on the result of above [Table 1-3], it could be confirmed that at the time of experiment of game play after applying AI to monster, attack was decreased.

Also, Exceptional situation under game battle state for the purpose of research is omitted. In the future study, the objective is to implement behavioral pattern having high sense of immersion to the game users by researching on behavior pattern of intelligent type monster that could be applied to this study and even to battle state.

References

[1] Lee Eun-Hee, Park Choong-Shik, Cho Sung-Hyun, A Study on the Intelligent NPC in MMORPG, Korea Contents Association, Spring Conference, Proceeding, **(2006)**, Vol.4 No.1.

[2] Jeongmo Yang, Kyungeun Cho, Kyhyun Um, A Dynamic Utilization method of FSM for Adaptive NPC Generation, Journal of Korea Multimedia Society, **(2008)**, Vol.11, No.9, pp.1258-1266.

[3] Park, Hyunsoo, Kim, Kyung-Joong, Recent Research Trends in Game Artificial Intelligence, Communications of the Korean Institute of Information Scientists and Engineers, **(2013)**, Vol.31, No.7, pp.8-15.

[4] Sangwon Um, Taeyong Kim, Jongsoo Choi, Dynamic Algorithm for Game difficulty control, Korea Computer Congress 2003, **(2003)**, Vol.1, No.2, pp.188-195.

[5] Ki-Duk Kwon, Behavior Control of NPC(Non-Player Character) using Hierarchical Finite State Machine, The Journal of Korean Institute of Information Technology, **(2009)**, Vol.7, No.3, pp.23-30.

[6] Sung-Won Mun○, HyungJe Cho, Intelligent AI Technique Adaptive for Online Game Using Fuzzy Extension Principle, Journal of the Korean society for computer game, **(2008)**, Vol.8, No.3, pp.77-85.

[7] Inwhee Joe, Moonwon Choi, A Neural Network-based Artificial Intelligence Algorithm with Movement for the Game NPC, The Journal of Korean Institute of Communications and Information Sciences, **(2010)**, Vol.35, No.12, pp.1181-1187.