

# Genetic Algorithm을 기반을 둔 사용자에 따른 NPC의 FuSM 동적 행동패턴 생성기법

## FuSM the dynamic behavior pattern generation technique of the NPC according to user based on Genetic Algorithm

김동주<sup>1</sup>, 김정윤<sup>2\*</sup>

DongJu Kim<sup>1</sup>, JungYoon Kim<sup>2\*</sup>

### 요약

게임에서는 그래픽, 스토리, 난이도 등 다양한 재미 요소를 가지고 있으며, 특히 난이도는 게임을 처음 접하는 초보자, 게임을 능수능란하게 하는 숙련자 모두에게 적합한 난이도가 존재한다. 본 논문에서는 FSM에 Fuzzy Algorithm를 접목시킨 FuSM과 다윈의 진화론을 토대로 한 Genetic Algorithm을 기반으로 Player(사용자)에 따른 난이도를 조절하는 알고리즘을 제안한다. Genetic Algorithm을 활용해 Player의 성적을 기준으로 실시간 동적 난이도 조절을 하는 알고리즘이다. 이를 통해 Player의 몰입도를 증진시킨다.

핵심어 : 행동 패턴, 동적 난이도 조절, 유전자 알고리즘, 인공지능

### Abstract

A game is comprised of various fun elements such as graphic, story and the degree of difficulty. Specifically, the element of the game is of different level between the game beginners and the skilled players. In this study, we propose an algorithm for adjusting the degree of difficulty based on the player (user) using the Genetic Algorithm based on a theory of Darwin's survival and the FuSM which is a combination of Fuzzy Algorithm and FSM. The utilization of the Genetic Algorithm based on the results of the player is an algorithm for adjusting the dynamic degree of difficulty in real time. As a result, the proposed algorithm enhances the immersion of a player.

Keyword : FSM, Fuzzy, FuSM, Genetic Algorithm, AI

1 Graduate School of Advanced Imaging Science, Multimedia & Film, Chung-Ang University, Seoul, 156-756, Korea  
e-mail : pknstdj@gmail.com

2 Graduate School of Game, Gachon University, 1342 Seongnam Daero, Sujeong-Gu, Seongnam-Si, Gyeonggi-Do, Korea.  
e-mail : kjoyoon79@gmail.com (Corresponding author)

Received(February 09, 2016), Review(February 23, 2016), Accepted(June 3, 2016), Published(June 30, 2016)

## 1. 서론

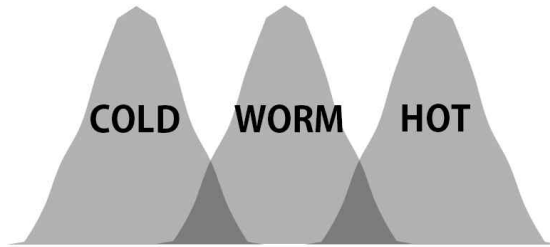
컴퓨터 기술이 발전함에 따라 온라인 게임이 생겨나고 더욱 더 화려하고 멋진 게임들이 하루가 멀다 하고 출시되었고 게임 AI도 점점 발전해 나갔다. 사람들은 더욱 재밌는 게임 즉, 몰입도가 높은 게임을 찾기 시작했고 몰입도를 높이는 큰 요건은 게임의 난이도 조절이었다. 이러한 난이도 조절의 가장 대표적인 것이 NPC(Non Player Character)의 발전이다. 인공지능(AI)가 발전함에 따라 동일한 혹은, 규칙 없는 난수를 이용해 조절되던 게임의 난이도에 체계성이 잡히면서 NPC의 행동 패턴이 다양해지고, 그에 따라 사용자의 몰입도 또한 더욱 증가할 수 있었다[1][2]. 게임AI 특히 Enemy의 AI는 게임의 난이도에 크게 영향을 미치게 되는데 너무 과하거나 너무 떨어져도 Player는 게임에 대한 몰입도가 떨어지며 재미와 흥미를 잃게 된다[3]. 이로 인해 실시간으로 게임 난이도를 조절하는 방법은 계속 연구진행 중이다[4]. 본 논문에서는 다양한 게임에 쓰였던 FSM에서 각 상태 전환의 중간 값의 경우 하지만 기존의 FSM의 경우 상태가 변화되는 과정에서 주어진 인자 값이 True, False보다 애매한 값이면 상황에 알맞은 상태로 변화되는 것에 어려움을 가지게 된다. 본 연구에서는 FSM에 Fuzzy Algorithm을 접목한 FuSM을 이용하여 NPC가 다양한 행동패턴의 변화를 하는데 있어서 더 유연한 결정을 내릴 수 있도록 하여 NPC와 Player간의 거리감을 좁혀 Player의 게임 몰입도를 더욱 높였다. 또한 NPC의 행동을 구조화 한 후 각각의 패턴들로 나누어 패턴을 유전 인자로 삼아 Genetic Algorithm을 이용해 게임을 거듭할수록 점점 더 게임의 난이도를 조절할 수 있다. 이러한 Genetic Algorithm을 비롯해 진화 알고리즘들은 여러 응용분야에 유용하게 사용할 수 있으며, 인공지능 기법 중 Player를 공격하는 적대적 NPC(이하 Enemy)의 행동 패턴을 통해 난이도를 조절하며 사용자에게 적합한 난이도를 제공한다면 작게는 Player의 즐거움, 크게는 인공지능의 발전과 게임 산업의 발전으로 나아가는 계기가 될 수 있을 것이다.

## 2. 관련연구

### 2.1. FuSM(Fuzzy State Machine)

FuSM은 기존의 FSM(Finite-State Machine)기법에 Fuzzy Algorithm을 접목시킨 인공지능 기법이다. FSM은 게임 내 NPC가 행동하는 여러 가지의 상태들(이동, 탐지, 공격 등)을 제어하는 알고리즘이다[2][5-7]. 매우 쉽게 구현이 가능하며 예전부터 지금까지 많은 게임에서 사용되고 있다. Fuzzy

Algorithm은 True와 False의 중간 ‘애매함’을 표현하기 위한 알고리즘이다. [그림 1]을 보면 Fuzzy Algorithm의 예시가 나와 있는데 Fuzzy Algorithm은 그래프의 양 끝의 Cold(차가움)와 Hot(뜨거움) 사이의 Worm(따뜻함)을 나타내는 알고리즘 이다. 양쪽에 속하지 않지만 둘의 성질을 어느 정도 내포하기 때문에 하는 애매함 때문에 회색이론 이라고도 불리며 사람들이 평소 사용하는 자연언어의 형용사에 표현되는 정도의 값을 구하는데 주로 사용되기에 단순히 게임만이 아니라 시스템의 제어, 조절, 관리에서도 자주 쓰인다. 이와 같은 FuSM은 FSM기법에 의해 NPC의 행동패턴 및 상태가 변하는 과정에 Fuzzy Algorithm이 작용하여 True, False만으로 구분 짓기에 어려운 애매한 상황에서 적합한 방향으로 변하게 되므로 NPC가 행동 변화를 결정할 때 사용자와 NPC간의 괴리감을 줄일 수 있으며 이러한 점으로 인해 사용자의 게임 몰입도를 증진 시킬 수 있다[6].

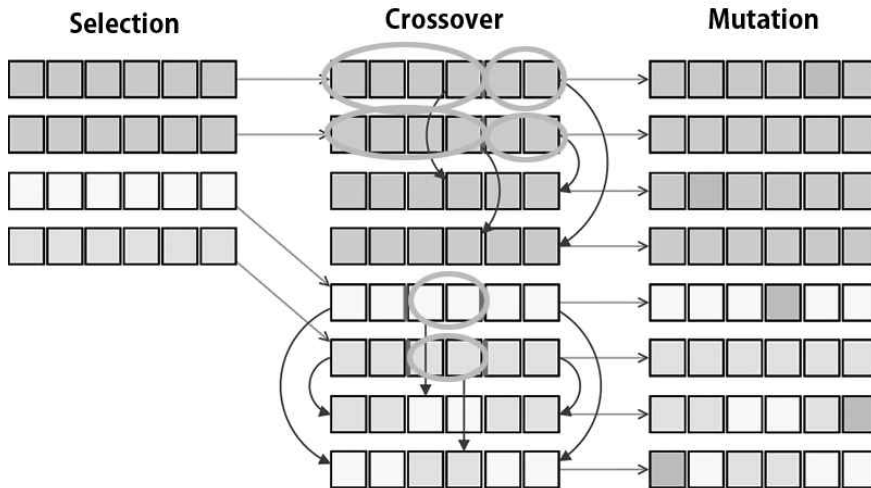


[그림 1] 차가움, 따뜻함, 뜨거움 으로 나타내는 Fuzzy Algorithm의 예시  
[Fig. 1] The example of Fuzzy Algorithm COLD, WORM, HOT

## 2.2. Genetic Algorithm

Genetic Algorithm(GA)는 다윈의 적자생존이론을 토대로 고안된 알고리즘 이다. 자연에서 자손들이 부모의 유전자를 가지고 태어나지만 그 유전인자 중 일부가 부모에 의해 교차 혹은 돌연변이를 일으켜 부모와는 약간 다른 유전인자를 가진 자손이 태어나게 된다. 이 자손들 중 우수한, 목표 값에 더 근접한 자손을 뽑아 새로운 부모가 되고 이들을 토대로 다시 자손을 만드는 과정을 반복하여 최적 값을 탐색하는 알고리즘으로 유전 알고리즘의 연산자는 선택(Selection), 교배(Crossover), 돌연변이(Mutation)가 있다[7]. [그림 2]는 GA의 선택, 교배, 돌연변이 연산자를 나타낸 그림이다. [그림 2]의 Selection부분을 보면 집단 중 가장 목표 값에 근접한 개체를 선택한 다음 Crossover 단계로 넘어간다. 선택 된 개체들의 유전인자의 일부분을 서로 교환하는 것이다. 우수한 개체들의 인자를 서로 교환 함으로써 다음 세대가 목표에 더 근접하게 하기 위함이다. 마지막 Mutation은 서로 교환 된 인자를 가진 자손들의 일부 유전인자가 무작위로 변환한 모습이다. 이렇게 생성된 돌연변

이 인자들로 인해 자손들은 의외성을 띄게 되는데 이는 목표 값에서 멀어질지, 가까워질지 예측할 수 없기 때문이다. 하지만 이미 교배 과정을 거쳤기 때문에 리스크가 적은 의외성을 띄고 있기에 실제로 목표 값에 더욱 빠르게 도달 할 가능성을 열어준다.



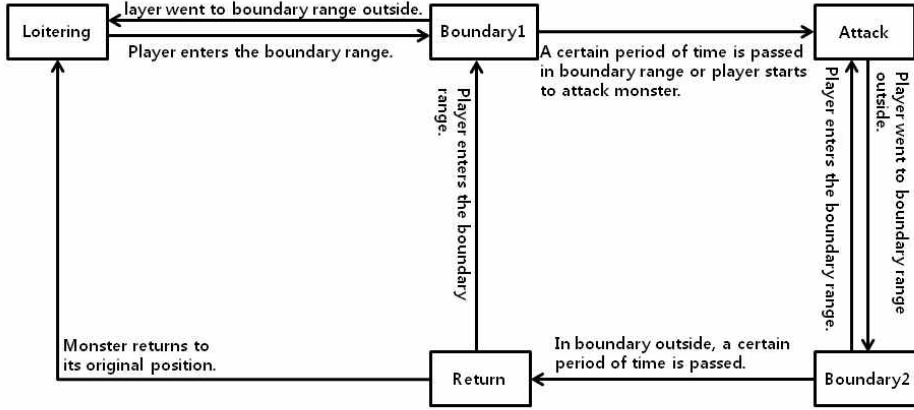
[그림 2] Genetic Algorithm의 연산자 Selection, Crossover, Mutation의 개념

[Fig. 2] Concept of Genetic Algorithm Selection, Crossover, Mutation

### 3. 실험 설계

[그림 3]은 FuSM기법을 사용한 Enemy의 행동패턴의 설계도 이다. 행동 ‘경계’가 둘로 나뉜 이유는 공격 전 경계 상태(이하 경계1)와 Player를 공격 후 경계 상태(이하 경계2)가 차이가 있기 때문이다. Enemy는 최소 행동범위 내에서 배회 하던 중 경계 범위 내의 Player와 마주칠 경우 경계1 상태로 바뀌게 된다. 일정시간 이상 Player가 경계범위를 벗어나지 않으면 Enemy는 공격 상태가 되어 Player에게 달려간다. 이때 Enemy의 경계 범위는 최대치로 늘어난 후 최소범위까지 점점 줄어들게 된다. 만약 공격상태에서 Player에게 공격받을 경우 경계범위는 다시 최대치로 늘어나게 된다. Enemy가 공격을 하던 중 Enemy의 최대 행동범위 밖으로 벗어날 경우 배회 상태로 강제 전환되어 최소 행동범위 내로 이동한다. 만약 Player가 경계범위 밖으로 벗어날 경우 Enemy는 경계 2 상태를 유지한다. 그 후 일정시간이 지나게 되면 Enemy는 배회 상태로 바뀌며 원래의 최소 행동범위 내로 이동하게 되며 일정시간이 지나기 전에 다시 경계범위 내로 Player가 진입 할 경우 즉시

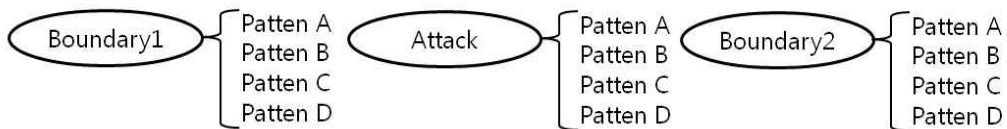
공격상태로 전환된다.



[그림 3] Enemy의 행동패턴 설계도  
 [Fig. 3] Behavior patterns of Enemy designs

Enemy의 행동의 값을 조정하여 경계1, 공격, 경계2 상태의 값을 조정하여 각각 4개씩 나누었다. 각각의 이름을 행동 뒤에 A-D로 표기하여 [그림 4]와 같이 무작위로 행동패턴을 생성하였다. (ex 경계1A - 공격D - 경계2B)

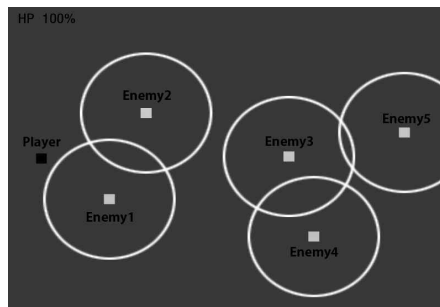
실험은 ①알고리즘 적용 전(무작위로 생성된 행동패턴을 가진 Enemy) ②알고리즘 적용 후 (1Round가 지날 때 마다 행동 패턴이 GA를 통해 진화하는 Enemy) 2가지로 나누어 각 7Round씩 Enemy의 총 수를 변경하여 3Set 진행 하였고, 2가지 실험방법 모두 난이도가 너무 쉽거나 어려워서 게임오버가 되지 않을 정도로 하기위해 Player가 목적지에 도달 했을 때 5%-25%의 체력이 남는 것을 목표로 하였다. 이 외의 조건을 같게 하기 위해 Player의 경로 및 속도, Enemy의 생성 위치는 ①②가 동일하며 7Round 후 1Set가 종료되면 바뀐다.



[그림 4] 행동패턴 적용 방식  
 [Fig. 4] Application of the behavior patterns

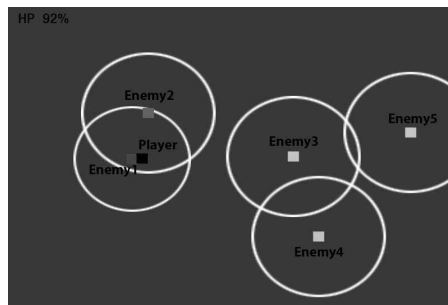
## 4. 실험 결과

[그림 5, 6, 7]의 알고리즘을 적용하여 Unity게임엔진을 이용해 제작한 프로그램의 시연화면 이다. [그림 5]은 플레이어와 몬스터의 초기 배치상태 이다. 프로그램 실행 시 Player가 오른쪽으로 이동하여 Enemy1에게 공격을 가하게 된다. [그림 6]은 Player와 Enemy1이 전투를 하고 있으며 동시에 Enemy2의 경계범위 내에 들어온 Player를 경계하고 있는 상황이다. [그림 7]은 Player가 정해진 경로로 계속 이동하고 있고 주변의 Enemy 들은 Player를 공격하기 위해 다가오고 있다. 알고리즘 적용 전과 적용 후의 실험을 각각 7Round씩 3Set 실행하였고 [표 1]은 Round가 지날 때마다 GA에 의해 진화하는 FuSM의 행동패턴을 표로 나타낸 것이다. 실험이 후반으로 갈수록 패턴의 변화 폭이 줄어들고 있는 것이 보인다. [표 2]와 [표 3]은 각각 1Round가 끝났을 때 Player의 남은 체력%와 Player가 Enemy에게 공격을 받은 시간의 총 합이다. 행동 패턴이 진화 하면서 목표 값과 가까워지기 위해 Enemy에게 더 많은 공격을 받게 되는 것을 알 수 있다.



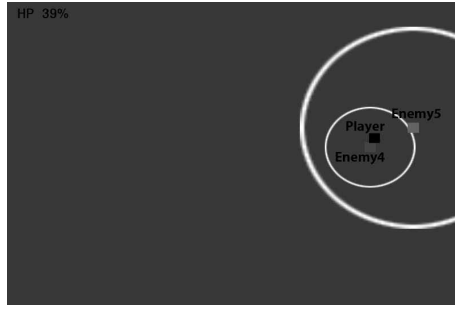
[그림 5] 프로그램 시작 초기상태

[Fig. 5] Initial state of the start of the program



[그림 6] Enemy1과 전투, Enemy2의 경계상태, 체력 92%

[Fig. 6] Enemy1 of battle, Enemy2 boundaries of state, strength 92%



[그림 7] Enemy1-3사망, Enemy4와 전투, Enemy5 경계상태, 체력 39%  
 [Fig. 7] Enemy1-3 death, Enemy4 battle, Enemy5 boundaries conditions, strength 39%

[표 1] 매 Round마다 Genetic Algorithm에 의해 진화하는 ②의 행동패턴  
 [Table 1] Behavior pattern of ② to evolve by each Round by Genetic Algorithm

	1Set	2Set	3Set
1Round	DAA	CAB	CAB
2Round	DBA	BBD	DBA
3Round	BAC	DAB	CBA
4Round	BAD	DDA	CCA
5Round	ACD	DAD	AAD
6Round	ABD	AAD	AAD
7Round	AAD	AAD	AAD

[표 2] 각 Round가 끝날 때 Player의 남은 체력%  
 [Table 2] Player's strength each Round end

	1Set		2Set		3Set	
	적용 전①	적용 후②	적용 전①	적용 후②	적용 전①	적용 후②
1Round	42%	42%	61%	60%	47%	47%
2Round	42%	30%	60%	45%	47%	45%
3Round	43%	27%	60%	42%	47%	43%
4Round	42%	25%	60%	37%	47%	29%
5Round	42%	22%	60%	33%	47%	20%
6Round	43%	18%	61%	24%	47%	19%
7Round	42%	16%	60%	23%	47%	20%

[표 3] 각 Round별 몬스터가 공격한 총 시간

[Table 3] The total time for each Round with different monsters attacked

	1Set		2Set		3Set	
	적용 전①	적용 후②	적용 전①	적용 후②	적용 전①	적용 후②
1Round	5.4s	5.4s	4.1s	4.1s	5.2s	5.2s
2Round	5.4s	6.0s	4.1s	5.3s	5.2s	5.3s
3Round	5.4s	6.2s	4.1s	5.4s	5.2s	5.4s
4Round	5.4s	6.3s	4.1s	5.7s	5.2s	6.2s
5Round	5.4s	6.4s	4.1s	5.9s	5.2s	6.5s
6Round	5.4s	6.6s	4.1s	6.4s	5.2s	6.5s
7Round	5.1s	6.8s	4.1s	6.4s	5.2s	6.5s

#### 4. 결론

게임에 재미요소와 몰입을 한 단계 높이기 위해서 인공지능 이론인 퍼지, 추론, 뉴런, 전문가시스템 등을 활용하고 있다. 이를 구현하기 위한 하드웨어, 소프트웨어, 통신기술의 비약적인 발전이 이루어지고 있다. 이러한 시대적 배경에 따라서 본 논문은 게임에서의 재미요소와 몰입을 높이기 위하여 Enemy의 행동패턴을 FSM에 Fuzzy알고리즘을 적용한 FuSM에 으로 제어하며 GA기법을 적용하여 Enemy의 행동 패턴을 실시간으로 Player에게 맞추어 난이도를 조절하는 방식으로 초보자와 숙련자가 게임에 흥미를 느껴 몰입감을 증가시키는 기법에 대하여 제안하였다. 제안하는 알고리즘을 활용한 실험결과에서는 Player의 실력에 따라 AI가 학습을 하며, Player의 실력에 맞춰서 난이도를 동적으로 조절 하는 것을 확인할 수 있었다. 향후에는 Player의 과거 정보를 데이터화 하여 해당 Player에게 특화된 맞춤형 난이도조절 방법을 제안 하고자한다.

## References

- [1] E.-H. Lee, C.-S. Park, S.-H. Cho, "A Study on the Intelligent NPC in MMORPG", Korea Contents Association, Spring Conference, Proceeding, vol.4, no.1, (2006).
- [2] J. Yang, K. Cho, K. Um, "A Dynamic Utilization method of FSM for Adaptive NPC Generation", Journal of Korea Multimedia Society, vol.11, no.9, (2008), pp. 1258-1266.
- [3] H. Park, k.-J. Kim, "Recent Research Trends in Game Artificial Intelligence", Communications of the Korean Institute of Information Scientists and Engineers, vol.31, no.7, (2013), pp. 8-15.
- [4] S. Um, T. Kim, J. Choi, "Dynamic Algorithm for Game difficulty control", Korea Computer Congress 2003, vol.1, no.2, (2003), pp. 188-195.
- [5] K.-D. Kwon, "Behavior Control of NPC(Non-Player Character) using Hierarchical Finite State Machine", The Journal of Korean Institute of Information Technology, vol.7, no.3, (2009), pp. 23-30.
- [6] S.-W. Mun, H.J. Cho, "Intelligent AI Technique Adaptive for Online Game Using Fuzzy Extension Principle", Journal of the Korean society for computer game, vol.8, no.3, (2008), pp. 77-85.
- [7] S.-W. Um, J.-S. Kim, J.-I. Shim, T.-Y. Kim, J.S. Choi, "Game Difficulty Controlling using Evolutionary Algorithm", Journal of The Korean Society for Computer Game, no.11, (2007), pp. 20-27

