FSM을 기반으로 한 사용자 행동 모델링 및 추천 시스템

User behavior modeling and recommendations based on FSM

김기웅, 강상현, 손우성, 최진영

영어 이름 / 메일

**요약**

방대한 데이터를 얻을 수 있는 현대 사회에서는 이러한 데이터를 처리하고 사용자에게 제공하는 방식에 대한 방안이 중요한 문제로 떠오르고 있다. 이러한 문제의 해결 방안으로 다양한 추천 알고리즘이 등장하고 있지만 모든 환경에서 제공되고 있지 않다. 본 논문은 게임 환경에서 많이 사용되는 FSM 알고리즘을 기반으로 사용자 행동 모델링 및 추천 알고리즘을 구현하는 것을 목표한다. 알고리즘을 실행시키면 사용자로부터 제공된 데이터를 입력 받아 데이터베이스에 저장한다. 이러한 데이터베이스를 FSM 알고리즘을 바탕으로 구현한 추천 알고리즘을 통해 분석하고 사용자에게 FSM 중 가장 효율적인 상태를 추천한다. 실험 과정은 베이즈 정리를 통해 NPC가 특정 상태에 돌입했을 시에 어떠한 상태가 플레이어에게 가장 효율적인지 계산한 결과를 참고하여 텍스트 게임을 제작하였다. 위 실험을 통한 결과로, 게임이 진행될수록 NPC의 상태에 따라 플레이어에게 가장 효과적인 대처 방안을 제시하는 것을 확인할 수 있었다.

**ABSTRACT**

In the modern society where vast amounts of data can be obtained, a method of processing such data and providing it to users is emerging as an important problem. As a solution to this problem, various recommendation algorithms are emerging, but they are not provided in all environments. This paper aims to implement user behavior modeling and recommendation algorithms based on the FSM algorithms that are widely used in game environments. When the algorithm is executed, data provided by the user is received and stored in a database. These databases are analyzed through recommendation algorithms implemented based on FSM algorithms and recommended to users the most efficient state of FSM. In the experimental process, a text game was produced by referring to the result of calculating which state is most efficient for the player when the NPC enters a specific state through Bayes' theorem. As a result of the above experiment, it was confirmed that as the game progressed, the most effective countermeasure was presented to the player according to the state of the NPC.

**I. 서론**

플레이어들은 게임을 즐길 때 게임 내에 등장하는 NPC(Non Player Character)와 다양한 상호작용을 하게 되고 고전 게임을 즐기다 보면 NPC가 비교적 최근에 나온 게임들의 NPC에 비해 행동 패턴이 단순하고 플레이어와의 상호작용이 원활하지 않은 것을 확인할 수 있다. 이러한 행동이 단순한 NPC는 플레이어에게 진부한 경험을 제공할 위험이 있는 반면, 현대의 게임에는 다양한 알고리즘을 적용해

해당 게임플레이어에게 적응시켜 NPC의 고정적인 패턴을 개선하고 있다. 이러한 방식의 NPC를 적응형 NPC한다. 이러한 적응형 NPC를 구현하는 방안중에는 실시간으로 NPC 고유의 행동 데이터베이스를 통해 NPC의 행동을 변화시키는 FSM(유한상태기계) 알고리즘이 있다. 본 논문은 연구를 통해 이러한 데이터베이스를 FSM에 국한하지 않고, 플레이어에게 NPC의 상태에 따라 특정 행동을 추천하는 데에 사용하는 것을 추가하도록 연구하였다. 이러한 연구를 통하여 실시간으로 행동이 변화하는 게임 환경에서도 플레이어에게 보다 나은 환경을 제시하여 준다. 추가로 게임 외의 대량의 정보가 이동하는 인터넷이나 스트리밍 서버에 적용시켜 활용할 수 있다.

**II. 관련 연구**

II-1 기존 연구

게임 환경에서 NPC(Non-Player Character)의 행동을 제어하고 변화시키는 연구는 오랫동안 게임 인공지능 분야의 주요 과제로 다루어져 왔다. 기존 연구에서는 게임 내 NPC가 상황에 따라 다양한 행동 패턴을 보일 수 있도록 설계하는 접근 방안을 제안하였다. 이러한 연구들은 게임의 몰입도를 높이기 위해 NPC가 고정된 행동 패턴에서 벗어나, 플레이어의 행동과 환경에 적응하는 방식을 도입하였다. 이를 통해 NPC의 행동이 보다 유연하게 변화할 수 있도록 하였으나, 사용자 맞춤형 대응 전략을 제공하기보다는 NPC의 자체적인 적응 능력에 초점을 맞추고 있다.

또 다른 연구들은 NPC의 상태를 전이시킬 수 있는 구조를 활용하여, 게임 내 상황 변화에 따라 NPC가 자동으로 적응하도록 하는 방안을 제시하였다. 이러한 연구들은 NPC의 상태를 여러 가지로 구분하고, 게임 내 발생하는 사건에 따라 NPC가 새로운 상태로 전이하도록 설계되었다. 그러나 이와 같은 접근 방법들은 NPC의 행동을 제어하는 데 중점을 두고 있으며, 플레이어의 행동 데이터를 바탕으로 최적의 대응을 추천하기보다는 NPC의 상태 변화 자체에 집중하는 경향이 있다.

II-2 기존 연구와의 차별점

본 연구는 기존 연구들이 NPC의 행동 변화를 유연하게 제어하는 데 초점을 맞춘 것과 달리, 사용자와의 상호작용 데이터를 실시간으로 반영하여 맞춤형 추천을 제공하는 아록리즘을 제안한 점에서 차별성을 가진다. 본 연구에서는 NPC의 상태뿐만 아니라, 플레이어의 행동 데이터를 활용하여 상황에 따른 최적의 대응 전략을 추천하도록 설계되었다. 이를 통해 기존의 단순한 행동 제어 방식을 넘어, 실시간 데이터 분석을 통해 사용자 맞춤형 대응을 제공하는 지능형 시스템을 구현하였다. 이는 플레이어의 몰입도를 높이고, 사용자 경험을 향상시키는 데 기여할 수 있다는 점에서 기존 연구들과의 차별성을 지닌다.

**III. FSM을 이용한 추천 알고리즘 적용**

게임 내에 NPC의 행동에 따른 추천 알고리즘을 구현하기 위하여 NPC의 행동을 데이터베이스로 저장하고 해당 데이터베이스를 기반으로 알고리즘을 적용해야 한다. 게임 플레이 시 NPC의 행동에 대응하는 플레이어의 대응을 바탕으로 베이즈 정리에 따른 추천값을 조정하여 사용자에게 가장 높은 추천값을 지닌 결과를 제공한다. 해당 챕터에서는 FSM을 적용시킨 환경에서 플레이어의 행동을 저장하고 추천방법에 대하여 설명할 예정이다.

III-1 FSM의 개념

게임에서 유동적인 AI를 지닌 NPC를 구현하기 위해 많이 사용하는 알고리즘으로 FSM은 비교적 구현이 간편하다는 장점이 있다.

[정의 1] FSM(유한상태기계)는 인공지능 기법 중 하나로 유한한 상태들의 집합을 통해 하나의 상태를 취하는 추상기계를 의미한다.

[정의 2] FSM에서의 전이는 취하고 있던 상태에서 다른 상태로 전환하는 것을 의미한다.

[정의 3] 효율값이란, FSM의 변화가 일어난 뒤 상태의 변화가 이뤄지고 실행된 행동의 성공여부에 따른 결과값을 말한다. 이를 산출하는 식은 다음과 같다.

[정의 3]에서의 효율값은 NPC의 상태의 전체 승리 데이터와 패배 데이터를 기반으로 시행횟수로 나누어 해당 행동의 결과가 효율적인지 판단하는 요소이다. NPC의 상태 전이시에 플레이어의 행동 선택에 따라 효율값이 변화하게 되고 값의 변화에 따라 플레이어에게 제시하는 상태의 종류가 달라질 수 있다. (보완필요)

III-2 데이터베이스 형식

FSM을 위한 요소는 NPC의 행동 데이터베이스에 저장하여 게임에서 발생하는 상태를 S, 각 상태를 선택한 횟수를 List, 상태의 성공/실패 횟수를 각각 W와 L, 해당 상태를 선택했을 때의 효율값 E를 포함하며 저장 형태는 다음과 같다.

표 1. 데이터베이스 저장 형태

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S | List | W | L | E |
| 게임 내 NPC 및 플레이어가 선택하는 상태 | S내  전이가 일어난 횟수 | 상호작용 이후 유효한 경우의 횟수 | 상호작용 이후 불리한 경우의 횟수 | S에  따른 상태별  효율값 |

III-3 추천 알고리즘

본 연구에서의 추천 알고리즘을 구현하기 위해서 베이즈 정리를 활용하였다. 베이즈 정리란 추천알고리즘에서 활용되는 기본적인 공식으로 사전 확률을 통하여 사후 확률을 구하는 정리이다.

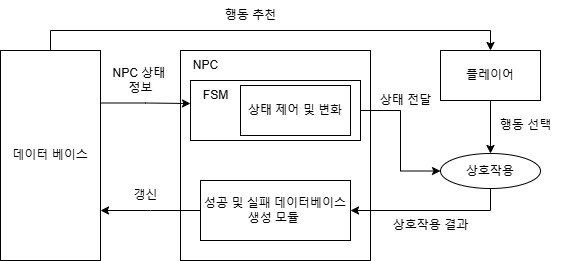
데이터 베이스에 저장된 요소들을 해당 정리에 대입시켜서 효율적이라 판단되는 사후 확률을 계산하게 되는데, 이를 해당 공식으로 변환하여 다음과 같이 계산하여 값을 도출한 뒤 데이터베이스에 저장한다.

위 공식은 전체 정보가 저장된 List값과 W, L에 저장된 값을 통해 해당 S로 진입할 시에 성공할 확률을 도출한다.

III-4 FSM기반 추천 구조

NPC의 구조는 FSM과 데이터베이스 생성 모듈 등으로 구성되어 있다. FSM은 NPC의 상태를 제어하고 상태의 변화를 구현한다. 이러한 FSM을 통해 NPC에서 상태를 얻은 뒤 플레이어와 상호작용을 하고, 결과를 NPC의 데이터베이스 생성 모듈을 거쳐 데이터베이스에 갱신한다. 또한 갱신된 정보를 바탕으로 플레이어에게 추천할 상태를 추천 알고리즘을 거쳐 기존 추천 정보를 삭제 및 수정한다. 이와 동시에 데이터베이스에서 추천 정보를 플레이어에게 전달한다. 해당 추천을 기반으로 플레이어의 선택이 종료되면 다시 NPC와의 상호작용이 이뤄지고 이를 반복한다.

그림 1. FSM을 활용한 추천 구조



**IV. 실험 및 분석**

위에서 제안한 기법들을 바탕으로 NPC의 상태 데이터를 기반으로 플레이어에게 효과적인 선택지를 제시하는지 실험한다. 효과적인 선택지를 제시하는지 실험하기 위해서는 FSM 내에서 상태 변화를 저장하고 성공과 실패가 일어날 때마다 실시간으로 저장된 수치와 확률을 갱신할 필요가 있다. 실험 단계에서는 상태 변화가 일어날 때마다 상태가 갱신되는지 확인하고 플레이어와 상호작용이 일어난 후 저장된 데이터와 공식의 결과를 비교하여 가장 높은 효율 값을 지닌 상태를 정상적으로 도출하는지 실험한다.

IV-1 실험 방법

본 실험을 실행하기 위해서 NPC의 행동 변화를 도출하는 것, 플레이어와 상호작용한 결과를 계산하는 과정, 결과를 데이터베이스에 갱신하고 해당 정도를 바탕으로 플레이어에게 행동을 추천하는 과정을 추가하여 적절한 결과를 산출하는지에 대한 여부를 조사해야 한다. 실험은 2가지 형태로 진행한다. 먼저 FSM알고리즘이 NPC에게 정상적으로 적용되는지 확인하기 위해, 효율값을 참고하여 상태를 선택하고 데이터가 정상적으로 저장되고 변화를 표현하는지 실험을 진행한다. 다음 실험은 FSM이 정상적으로 작동하는 경우 플레이어와의 상호작용이 일어난 뒤에 성공/실패를 판정하고 결과를 데이터베이스에 전송한 뒤, 플레이어에게 성공률이 가장 높은 행동을 추천하는 것을 확인하는 실험을 진행한다.

IV-2 게임 제작

실험용 게임은 플레이어와 NPC가 각자 자신의 속도에 따라 공격과 방어를 주고받는 턴제 전략 게임으로 Unity 환경에서 제작하였다. 그림 2는 실험용 게임의 실행할 경우 보여주는 게임의 화면이다. 표 2는 게임 실행 중 플레이어가 선택할 수 있는 행동을 공격 및 수비 턴에 맞춰 각각 정리한 표이다.

그림 2. 게임 실행 화면텍스트, 스크린샷, 만화 영화, 폰트이(가) 표시된 사진

자동 생성된 설명

표 2. 게임 내 사용하는 기술

|  |  |  |  |
| --- | --- | --- | --- |
| 종류 | 기술명 | 데미지 | 효과 |
| 공격 | 일반 공격 | 15 | 특수 효과 없음 |
| 강 공격 | 30 | 일정 확률로 공격 |
| 수비 | 방어 | - | 입는 데미지를 경감. |
| 회피 | - | 일정 확률로 데미지를  무효 |
| 회복 | - | 체력 회복  (최대 HP를  초과하지 않음) |

성공과 실패를 판정하는 결과는 현재 체력의 변화를 턴이 끝날 때마다 확인하여 효율값을 구한다. 기존 체력과 이전 체력의 차이를 통해 체력 증감률을 구한 뒤 체력이 이전보다 소모한 비율이 체력의 10%이상의 피해를 입었다면 W의 값이 증가하고 반대라면 L의 값이 증가하도록 설정했다.

IV-3 실험 결과

실험 중 수집한 각 상태별 추천 확률의 증감과 상태의 변화 로그가 파일로 생성된다. 실험 결과에서는 로그를 통해 상태의 변화가 정상적으로 작동하는 지와 상호작용시에 추천하는 행동을 집중적으로 분석한다.

그래프 1은 실험 1 이후 생성된 로그 파일의 개수와 추천 상태가 다른 상태로 갱신될 때마다 기록한 결과이다. 실험 1에서는 추천하는 행동의 변화는 상태가 변화할수록 점차 감소하는 결과가 나타났다. 그래프 2는 실험 2에서 행동 변화가 증가할수록 추천 확률의 변화량이 점점 줄어들고 확률이 거의 고정되는 결과가 나타났다.

그래프 1. 실험 1의 행동 및 상태 변화

그래프 2. 실험 2의 행동 및 추천 확률 변화량

IV-4 실험 분석

실험 1의 결과에서 행동 변화가 증가할수록 변화의 기울기가 0에 수렴하고 있다. 이를 통해 시행 횟수가 증가할수록 상태의 변화가 적어지는데 이를 분석하면 처음 플레이어의 행동 변화가 적을 경우 베이즈 정리에 따른 확률에 기반한 상태의 변화가 자주 일어나고 있다. 그리고 이후 플레이어의 행동변화가 증가하면 NPC의 상태 변화가 점점 줄어드는데 이를 상태 제어에 포함된 상태를 선택하는 확률의 변화와 비교하면 그래프 3에서 상태 선택 확률의 변화량이 시행횟수가 추가될수록 점점 적어진다. 즉, 상태를 변화시킬 확률이 점점 변화가 적어지며 플레이어의 상태에 대응하는 가장 효율적인 상태를 선택하도록 NPC의 상태 선택 확률이 변화한다는 것을 확인할 수 있다.

그래프 3. 실험 1의 상태 선택 확률 변화

실험 1의 결과를 기반으로 실험 2를 진행한 경우 플레이어에게 적용시켰을 경우에도 상태 선택 확률의 변화량이 점차 감소하는지 확인하고 목표로 하는 효율값에 도달할 때 까지의 시행횟수를 각각 기록하고 이를 추천 확률 변화량과 비교하여 분석하였다.

그래프 4. 효율값 목표에 따른 확률 변화

추천 알고리즘의 목표 효율값을 60으로 설정한 경우에는 49번의 상태 추천 이후 목표를 달성하였고 목표 효율값을 80으로 설정한 경우에는 94번의 상태 추천 이후 목표 효율값을 달성하였다. 시행 횟수가 증가할수록 플레이어에게 추천하는 행동은 NPC에게 적용했을 때와 비슷하게 점차 변화가 줄어들었다. 이 결과를 통해 시행횟수가 증가할수록 효율값과 추천 알고리즘을 통해 도출되는 확률의 증감률이 유사한 결과를 내놓으며, 효율값을 기반으로 플레이어에게 가장 효율적인 행동을 추천하고 이 행동의 변화는 시행이 증가할수록 줄어든다는 결과를 얻을 수 있었다.

**V. 결론**

본 연구에서는 FSM(유한상태기계)과 베이즈 정리를 결합하여 게임 내 NPC와의 상호작용에서 사용자에게 최적의 대응을 추천하는 지능형 추천 시스템을 제안하였다. 본 연구의 기여점은 FSM을 단순히 NPC의 행동을 제어하는 도구로 사용하는 것을 넘어, 플레이어의 행동 데이터를 실시간으로 반영하여 보다 능동적이고 개인화된 추천을 제공하는 시스템을 설계한 것이다.

이는 기존 연구에서 제안된 적응형 NPC 구현 방법론을 대부분 NPC의 고정적 행동패턴을 동적으로 변화시키는데 집중했던 것과 차별화된다. 특히, 실험 과정을 통해 FSM을 통해 게임 내 모든 NPC의 상태를 관리하고, 상태 전이마다 저장된 데이터를 활용하여 효율값을 산출하였다. 이러한 FSM의 전이 과정은 플레이어의 상호작용 데이터를 기반으로 행동의 효율성을 정량화하며, 이를 바탕으로 최적의 상태를 추천하게 된다. 이를 통해 FSM의 전이 상태를 실시간으로 반영하고, NPC의 행동 패턴을 더욱 정교하게 제어할 수 있었다.

또한, 베이즈 정리를 추천 알고리즘의 핵심 원리로 적용하여 기존의 사전 확률과 새로운 데이터에 기반한 사후 확률을 계산함으로써, NPC와의 상호작용에서 플레이어가 성공 확률을 실시간으로 조정하였다. 이는 게임 속 변화하는 다양한 환경에도 정확한 추천이 가능하도록 기여하였다. 이러한 접근 방식은 기존 연구들과 달리, 사용자와 NPC 사이의 상호작용 데이터를 동적으로 분석하여 확률을 지속적으로 업데이트함으로써, 보다 정교한 추천 시스템을 구현했다는 점에서 차별화된다.

그 결과, FSM과 베이즈 정리를 활용한 알고리즘이 플레이어와 NPC 간의 상호작용에서 적응적으로 최적화된 행동을 제시함으로써 게임의 몰입감과 사용자 경험을 향상시키는 데 기여하였음을 확인할 수 있었다.

**VI. 참고 문헌**

[1]Cho, Byeong-heon, Jung, Sung Hoon, Seong, Yeong Rak, and Oh, Ha-Ryoung, “An Implementation of Intelligent Game Characters using Neural Networks,” The KIPS Transactions:PartB, vol. 11B, no. 7, pp. 831–840, Dec. 2004.

[2] Se-Woong Oh, Hee-Min Kang, Young-Jin Cho, Man-Sik Lim, Sang-Muk Kim, Jong-Beom Lee, Ko-Eun Sin, Ji-Hoon Lee, Myung-Ju Kang, Chan-Il Park, Jong-Won Lee, Hyoun-Ju Oh, and Sang-Jung Kim, "A Design of Effective NPC AI Patterns Using the Theory of ‘Flow’ and FSM in the Adventure Game", Proceedings of the Korean Society of Computer Information Conference , pp. 297-301, July 2014.

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

[4]M. Lee, “An Artificial Intelligence Evaluation on FSM-Based Game NPC,” Journal of Korea Game Society, vol. 14, no. 5. Korea Academic Society of Games, pp. 127–135, October 2014.

[5]Ho Suk Lee, "On-line Bayesian Learning based on Wireless Sensor Network", 한국정보과학회 학술발표논문집, pp. 105-108, June 2007

[6]Jae-Won Jo and Jung-Won Bang, "A study on The Implementation of Monster AI using Finite-State Machine", Proceedings of the Korean Society of Computer Information Conference , 경북, pp. 349-350, January 2019.

[7] Yeong-Ju Yun, Jong-Hyen Kim, Hyeon-Jeong Jo, Gi-Jae Yun, Ka-Hwa Jeong, Seung-Pyo Hong, Byeong-Man Kim, and Hae-Yeoun Lee, "Game Recommendation System Using Naive Bayesian Classification", Proceedings of KIIT Conference, 광주, pp. 185-187, June 2018.

[8]de Finetti, Bruno. “Bayesianism: Its Unifying Role for Both the Foundations and Applications of Statistics.” International Statistical Review / Revue Internationale de Statistique, vol. 42, no. 2, 1974, pp. 117–30. October 2024.