

적응형 NPC를 생성하는 행동 정보 관리 기법

김나라[○], 엄기현*, 조경은**

동국대학교 영상대학원 멀티미디어학과[○], 동국대학교 게임·멀티미디어공학과* **
{aranmik[○], khum*, cke**}@dongguk.edu

An Action Information Management Method for Creating Adaptive NPC

Nara Kim[○], Kyhyun Um*, Kyungeun Cho**

Dept. of Multimedia, Graduate School of Contents & Digital Media, Dongguk University[○]

Dept. of Game & Multimedia Engineering, Dongguk University **

요 약

최근 게임의 종류가 다양해지고 접할 기회가 많아지면서 게임을 즐기는 사용자의 성향과 수준도 다양해지고 있다. 기존의 NPC는 행동이 단순하고 획일적이기 때문에 다양한 사용자를 상대하는데 한계가 있다. 따라서 각기 다른 사용자와 비슷한 수준으로 대응할 수 있는 적응형 NPC를 생성하는 기술이 필요하다. 본 논문은 적응형 NPC를 생성하기 위한 행동 정보 관리 기법을 제안한다. 행동 정보 저장 방법은 적응형 NPC가 사용자의 행동을 관찰하고 [상태-행동]의 관계로 정보를 수집한다. 수집한 행동 정보의 효율값을 구하고, 유사한 상태정보들은 군집화하여 행동 데이터베이스에 저장한다. 게임시스템은 행동 데이터베이스를 갱신하며 다양한 행동을 저장하고 효율값이 좋은 행동을 선택하여 사용자에게 적응해가는 NPC를 생성한다. 본 연구에서 제안한 적응기법을 액션 게임에 적용하여 실험하였다. 임의의 실험자는 적응형 NPC와 실시간으로 1:1 전투를 한다. 게임 결과 로그 파일을 통해 실험자와 NPC의 행동 성향을 분석하여 유사성을 판단한다. 다양한 실험 결과의 통계에서 오차를 6% 이내의 사용자와 비슷한 수준으로 적응해가는 NPC를 생성할 수 있었다.

ABSTRACT

Although people have had more opportunities to enjoy various types of game, most of players have felt less satisfaction with the games. This is the reason that since most Non-player Characters (NPCs) are simple and uniform, they have some limitations for competing with a variety of players. Thus, technologies for creating intelligent NPCs that can compete with each player at a similar level are required. In this paper, we present an action information management method for creating adaptive NPCs based on the algorithm for calculating their action efficiency. This algorithm is useful to the adaptation method for saving and controlling player-appropriate action. In our method, adaptive NPCs observe the actions of players and collect the relationship data between status and action. The efficiency value of the action data is calculated and data of similar status are grouped, and finally stored at the action database. The game system of NPC updates the action database and stores diverse actions. Then, NPC selects action with high efficiency value. We have tested our algorithm on an action game. A random test subject performed a one-on-one game against an adaptive NPC in real-time. As a result, the action dispositions of both the subject and NPC are analyzed in a log file to determine whether or not the disposition of the subject is similar to that of the NPC. The statistics of the diverse test results shows that NPCs become adaptive to players with error rate within less than 6%.

keyword : Adaptive NPC, Action Information Management Method

※ 교신저자(Corresponding Author): 조경은, 주소: 서울시 중구 필동 3가 26 동국대학교(100-715), 전화: 02/2260-3834, E-mail: cke@dongguk.edu

1. 서론

최근 거의 모든 컴퓨터 게임에 인공지능 기술이 적용되고 있다. 이러한 추세는 앞으로도 지속될 전망이다[1]. 기존 게임 개발자들은 게임의 그래픽 요소만을 중시하던 성향이 강했다. 최근 게임 개발자들은 인공지능 기술이 도입된 NPC(Non-Player Characters) 즉, 가상 인공지능 캐릭터를 생성하기 위해 노력하고 있다. NPC는 사용자가 게임을 진행하는데 필요한 수단으로 사용되고 있다. 예를 들어, “용병”이라고 불리는 NPC는 사용자가 전투할 때 도움을 주는 캐릭터다. 용병은 사용자와 전투중인 몬스터를 공격하고, 사용자 캐릭터에게 이로운 방법을 사용하여 좀 더 쉽게 전투할 수 있도록 돕는다. 물건을 사고 팔 수 있도록 “상인”과 같은 중개인의 역할을 하는 NPC도 있다. 게임에서 퀘스트는 임무를 수행하고 보상받는 시스템이다. 퀘스트를 도와주는 NPC는 원활한 게임 진행을 위한 안내자 역할을 한다. “몬스터”라고 불리는 전투와 관련된 NPC는 사용자 캐릭터를 성장시키기 위한 도구로 사용되고 있다.

게임 개발자가 사용자와 같이 복잡한 행동을 하는 NPC를 생성하기 위해서는 개발 기간이 길어지는 등의 어려움이 따른다. 따라서 기존 NPC는 개발자가 정한 패턴대로 행동하는 단순하고 획일적인 캐릭터가 대부분이었고, 여러 가지 문제가 있다. NPC의 행동이 매우 단순하기 때문에 사용자들은 금방 지루해지고 게임 수명이 줄어들게 된다[2]. 그리고 NPC의 약점이 드러났음에도 불구하고 수정할 수 있는 능력이 없다. 몇몇 사용자는 NPC의 약점을 이용하여 많은 수의 NPC를 혼자 독점하여 빠르게 성장할 수 있지만 상대적으로 다른 사용자에게 피해를 주게 된다. 앞에서 설명한 버그성 플레이를 악용하는 경우 대응할 수 없는 치명적인 오류를 가지고 있다. 사용자의 수준과 성향이 다양해지면서 모든 사용자에게 적절하게 대응할 수 있는 적응형 NPC 생성 기술이 필요하다. 적응형 NPC는 변화하는 환경에서 자신의 행동을 변화시켜 환경에 적합하게 행동한다. 할 수 없던 행동들

을 할 수 있고 다양한 행동들이 가능해지도록 적용하는 캐릭터이다.

본 논문에서는 적응형 NPC를 생성하기 위한 행동 정보 관리 기법을 제안한다. NPC가 행동 정보를 가지고 있지 않다면 다양한 행동 조합을 할 수 없고, 상황에 맞게 적용하는 것이 불가능하다. 따라서 적응형 NPC를 생성하기 위해서는 새로운 행동을 저장, 관리할 수 있어야 한다. 새로운 행동 정보들이 저장되면 이전에 할 수 없던 행동들을 조합할 수 있고, 다양한 행동을 구사할 수 있는 적응형 캐릭터가 생성된다. 적응형 NPC를 통해 기존 NPC가 가지고 있는 문제점을 해결하고 범용적인 게임 시스템에 적용 가능하도록 설계한다.

본 논문의 구성은 다음과 같다. 2장에서는 적응형 캐릭터와 관련된 기존 연구들을 설명한다. 그리고 기존 연구와 본 연구를 비교 논의한다. 3장에서는 적응형 NPC를 생성하기 위한 NPC 행동 효율 산출 알고리즘을 설명한다. 4장에서는 제안 기법을 구현 실험한 내용을 보이며 실험 결과를 분석한다. 끝으로 5장에서는 본 연구를 통해 얻은 결론을 도출한다.

2. 관련 연구

본 장에서는 적응형 NPC 생성 방법에 대한 기존 연구들을 살펴본다. 또한 기존 연구들의 문제점을 살펴보고 본 연구와 비교한다. 적응형 NPC와 관련된 기존 연구들을 살펴보면 학습 기법을 이용하여 NPC 캐릭터를 학습시켜 적응해 가는 연구 사례가 있다[3]. 행동규칙기반의 연구에서는 규칙기반 시스템과 추론시스템을 이용하여 캐릭터의 행동 변화를 통해 적응해가는 캐릭터를 생성하였고, 대표적으로 동적 스크립팅(Dynamic Scripting) 기술을 사용한 연구가 있다[4,5,6]. 그리고 다단계 적응 기법(Fast Multi-Level Adaptation) [7]과, 신경망 기술을 이용해 적응형 캐릭터를 생성한 연구 사례 [8]가 있다.

다단계 적응기법을 사용한 연구에서는 사용자의 행동을 모방하여 행동, 일, 목표를 다단계적으로 학습한다. 이 논문에서는 적응형 캐릭터를 만들기

위해서 High, Mid, Low의 세 단계로 학습 시스템을 구성한다. Low 레벨의 시스템은 관찰을 기본으로 하는 추론 시스템으로 행동을 선택한다. Mid 레벨에서는 경험과 계획 기반의 작업 선택 시스템으로 일을 선택한다. High 레벨은 감정적인 상태를 감지하는 시스템으로 목적을 선택하기 위해 사용된다. 단계적인 학습을 통해 빠른 시간에 적응이 가능한 캐릭터를 생성한다[7]. 다단계 적응기법을 사용한 연구사례에서는 게임 환경 정보나 게임의 요소를 최대한 단순하게 구성해야 한다. 이 시스템을 그렇지 않은 일반 게임에 적용할 경우 과부하가 생기기 때문에 적용할 수 있는 모델의 수가 매우 적다.

동적 스크립팅 기술은 행동 규칙 기반의 적응 기법이다. 각 행동을 하기 위한 규칙들이 정해져 있고, 이 규칙들은 NPC 규칙 베이스에 저장되어 있다. 이를 기반으로 NPC의 행동을 제어하는 스크립트를 생성한다. 이 스크립트에 의해 제어되는 NPC는 사용자와 상호작용하고, 그 결과를 이용하여 NPC 규칙 베이스를 갱신한다. 갱신된 규칙 베이스에서 가장 적합한 행동 규칙들을 조합하고 새로운 스크립트를 생성하여 NPC를 제어한다. 이 과정을 반복하여, 사용자에게 적응해가는 NPC를 생성한다 [4,5,6]. 동적 스크립팅 기법은 규칙 베이스를 반복적으로 갱신하므로 사용자에게 적응하는 NPC가 생성된다. 하지만 사전에 게임 캐릭터의 스크립트를 제작하는데 많은 작업시간이 걸리며, 스크립트의 갱신을 통해 사용자에게 적응하므로 스크립트를 지원하지 않는 게임에서는 적용하기 어렵다. 또, 상호작용 결과를 통해 규칙을 갱신하기 때문에 게임 도중에 실시간으로 행동 변화를 할 수 없다.

신경망과 학습을 사용하여 적응형 캐릭터를 생성한 연구에서는 NPC가 상대방 캐릭터의 행동 및 거리가 어느 정도인지를 관찰하고 관찰한 정보를 신경망에 입력한다. 신경망을 통해 출력된 결과를 이용해서 초기 NPC는 행동을 무작위로 선택하여 규칙을 학습한다. 이후 신경망은 상대방과의 행동의 결과로 얻은 게임점수를 강화값으로 이용한다. 강화학습을 진행하면서 NPC의 신경망은 특정한 상황에

따라 어떤 행동이 적절한 행동인지를 학습하고 적응하는 캐릭터가 생성된다[8]. 신경망을 이용한 적응기법은 원하는 결과 노드의 값이 나오도록 적절한 가중치를 설정하는 과정이 어렵다. 또, 무작위 행동을 통해 기초 학습을 하는 과정이 필요하므로 학습하는데 시간이 필요하다.

사용자 기반 상호작용 알고리즘을 사용한 연구에서는 게임 난이도 조절 요소와 적응도 조절 요소를 추출한 결과를 유전자 알고리즘을 이용하여 적응하는 기법을 제안했다. 유전자 초기화 단계에서 초기 게임의 집합 세대를 구성한다. 이후 게임 난이도와 적응도 요소를 이용해 만든 적합도를 평가하고 세대를 개선한다. 게임이 진행되면 지속적으로 유전자 알고리즘을 연산하고 적합도 평가 요소를 통해 사용자의 수준을 판단한다. 유전자의 선택과 교배, 돌연변이를 수행하여 새로운 세대를 구성하고 사용자의 수준에 맞는 유전자를 출력해서 알맞은 난이도로 적응해간다[3].

본 연구에서는 행동 정보 관리 기법으로 적응형 NPC를 생성하는 방법을 제안한다. 관리 기법은 행동 효율 산출 알고리즘을 통해 플레이어의 반응으로부터 얻은 새로운 행동 정보들을 적응형 NPC 행동 데이터베이스에 저장한다. NPC는 사용자와 상호작용할 때, 저장된 행동 정보의 효율값을 이용해 행동을 선택하고 대응한다. 알고리즘은 상호작용 도중 행동 정보를 저장하고 관리할 수 있으므로 실시간 행동 변화가 가능하다. 또, NPC 내부에 조작모듈과 함께 구성하도록 설계하였다. NPC는 조작모듈을 통해 행동하게 되고 그 결과를 NPC 행동 효율 산출 알고리즘에 넘겨준다. 알고리즘을 통해 얻은 정보는 NPC 지식베이스에 갱신되면서 다시 NPC 조작 모듈에서 사용할 수 있도록 순환한다. 따라서 NPC 내부에 조작모듈이 있는 장르의 게임이라면 범용적으로 적용할 수 있다. 다양한 수준의 사용자에게 적절한 수준으로 적응하여 단순하고 확실적인 NPC의 문제점인 게임의 지루함을 해결할 수 있고 재미요소로 활용이 가능하다. 또, 사용자가 버그성 플레이를 하는 경우 사용자를 통해 저장된 행동 조합을 통해 대응할 수 있다.

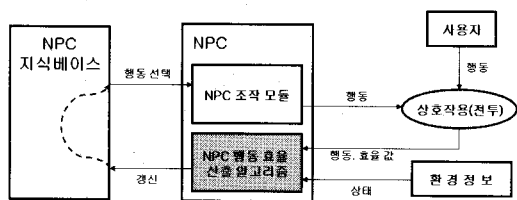
3. NPC 행동 효율 산출 알고리즘

본 장에서는 적응형 NPC를 생성하는 행동 정보 관리 기법에 대해 설명한다. 기존 NPC는 지식베이스에 저장된 행동 정보가 제한되어 있기 때문에 단순하게 반복하는 움직임에서 벗어날 수 없다. NPC가 사용자에게 적응하기 위해서는 변화하는 환경에서 상황에 맞는 적절한 행동을 취할 수 있어야 한다. 따라서 NPC 스스로 다양한 행동 정보들을 저장하고 관리할 수 있는 능력이 필요하다.

본 장에서는 다양한 행동 정보를 저장하고 관리하기 위한 NPC 행동 효율 산출 알고리즘을 제안한다. NPC 행동 효율 산출 알고리즘은 [상태-행동]의 관계 정의, [상태-행동]의 효율값 계산, 유사한 상태의 클러스터링, 적응형 NPC 행동 데이터베이스의 절차를 따른다. 각 단계별 절차는 3.2부터 3.5절에서 자세히 설명한다.

3.1 적응형 NPC 생성 시스템

기존의 NPC는 정해진 행동이 조작 모듈을 통해 일정 패턴으로 반복하는 것이 대부분이었다. 본 연구에서는 일반적인 NPC를 설계하는 틀에서 NPC 행동 효율 산출 알고리즘 모듈을 추가하여 적응형 NPC의 생성 시스템을 설계한다. 다음 [그림 1]는 적응형 NPC 생성 시스템 구조도이다.

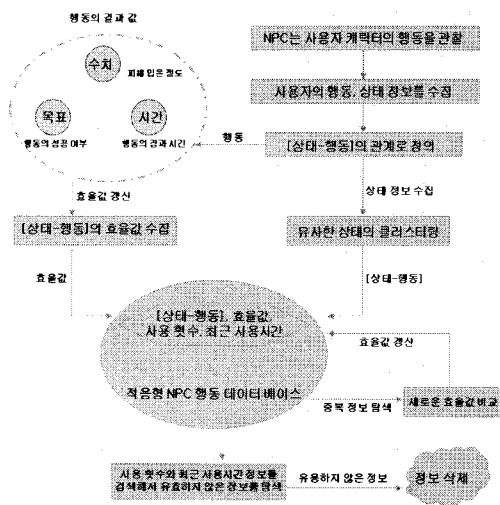


[그림 1] 적응형 NPC 생성 시스템

NPC 지식베이스에는 NPC가 취할 수 있는 행동들에 대한 정보들이 저장되어 있다. 이 지식베이스에서 적절한 행동을 선택하여 NPC 조작 모듈을 통해 게임 상에서 행동하게 된다. 사용자 캐릭터의 행동을 통해서로 만났을 때 전투라는 상호작용을 하게 된다. 적응

형 NPC 생성 시스템에서는 NPC와 사용자의 상호작용한 결과값과 게임 환경의 상태 정보값을 NPC 행동 효율 산출 알고리즘에 넘겨준다. 이 결과값을 이용하여 행동 정보를 갱신 하도록 설계하였다.

NPC 행동 효율 산출 알고리즘은 지식베이스에 행동 정보가 저장되어 있지 않은 NPC에게 행동 정보를 저장하는 역할을 수행한다. 지식 베이스에 다양한 행동 정보가 없다면 모든 상황에 적합한 행동을 취할 수 없기 때문에 적응형 캐릭터 생성이 불가능하다. 행동 정보를 저장하기 위해서 사용자의 행동을 관찰하여 유효한 정보를 수집한다. 수집된 정보에는 행동 정보와 상태 정보가 있다. 행동 정보의 결과값을 이용하여 행동에 대한 효율값을 구한다. 수집된 상태정보의 유사성을 판단해서 비슷한 상태 정보들을 군집화한다. 군집화된 상태 정보들은 대표 상태값을 가진 유효한 정보가 된다. 최종적으로 상태에 따른 행동 정보와 효율값의 정보들을 모아 행동 데이터베이스를 구축한다. NPC 행동 효율 산출 알고리즘의 절차는 다음[그림 2]와 같다.



[그림 2] NPC 행동 효율 산출 알고리즘 구조도

3.2 [상태-행동]의 관계 정의

지식 베이스에 행동 정보가 저장되어 있지 않은 초기 NPC가 행동 정보들을 저장하기 위해 게임에

서 사용자 캐릭터의 행동을 통해 정보를 수집한다. 사용자는 컨트롤러를 통해 게임의 사용자 캐릭터를 조작한다. 사용자 캐릭터는 컨트롤러를 통해 입력 받은 값을 행동으로 처리한다. NPC는 이 정보를 수집해서 지식베이스에 없는 새로운 행동 정보를 저장한다.

새로운 행동 정보가 저장된 지식베이스에서 행동을 선택하는 조건으로 상태 정보가 필요하다. 상태는 게임의 환경 정보로 게임의 모든 상황을 표현하는 정보다. 예를 들어, 사용자와 NPC 사이의 거리와 캐릭터의 시선, 공격력과 체력수치 등의 정보들을 말한다. 게임이 시작되면 적응형 NPC와 사용자가 상호작용하는 순간부터 실시간으로 행동정보와 상태정보를 수집한다. 이 정보들을 어떤 상태에서 어떤 행동을 하였다는 [상태-행동]의 관계로 정의하고 [상태-행동] 수집 테이블에 저장한다. 이 정보는 [표 1]에서 확인할 수 있다.

[표 1] [상태-행동] 수집 테이블

상태 정보 수집	거리	사용자와 NPC의 거리
	방향	사용자와 NPC의 시선
	수치	사용자와 NPC의 공격력, 체력 등 파라미터 정보
	시간	행동의 경과 시간
행동 정보 수집	행동	단일 행동, 연속된 행동

3.3 [상태-행동]의 효율값 계산

기본 정보 수집 단계에서 얻은 [상태-행동]의 정보는 지식베이스의 갱신을 통해 저장, 관리된다. 지식베이스에 저장된 행동 정보는 NPC 조작 모듈을 통해 활용된다. NPC 조작 모듈에서 사용하기 위해 선정 기준에 맞춰 행동 정보를 선택한다. 선정 기준으로 효율값의 수치를 이용한다. 효율값이란 NPC가 행동한 결과가 얼마나 효율적 인가를 판단하는 수치다. NPC가 얼마나 효율적으로 사용자에게 대응했는가를 알 수 있는 기준이 된다.

적응형 NPC는 사용자와 상호작용을 통해 얻은 결과값에서 수치, 목표, 시간의 정보를 이용해 [상태-행동]의 효율값을 구한다. 수치(V)는 행동을 취

했을 때 상대방에게 피해를 입힌 양과 자신이 피해 입은 양에 대한 수치적 정보들을 담고 있다. 상호작용의 행동결과는 상대방에게 피해를 입힌 수치가 클수록, 자신이 피해 입은 양이 적을수록 효율이 좋은 행동이라고 판단할 수 있다. 목표(G)는 행동에 대한 성공 여부를 판단한다. 효율이 좋은 행동만을 선택하더라도 상대방의 위치 등 상태 값에 따라 행동의 효율이 다르기 때문에 사용된다. 시간(T)은 행동의 경과 시간을 나타낸다. 연속적인 행동을 취했을 때 연속행동의 경과된 시간이 다르기 때문에 단위 시간당 행동의 효율성을 판단하기 위해 사용된다. 앞에서 설명한 효율값을 구하는 식은 [수식 1]과 같다.

$$\text{효율값} = \frac{\sum(V \times \alpha)}{T \times \beta} \times (G \times \gamma)$$

(α : 수치가중치, β : 시간가중치, γ : 목표가중치)

[수식 1] [상태-행동]의 효율값 계산

[수식 1]의 수치가중치, 시간가중치, 목표가중치는 게임의 장르나 특징에 따라 변화하는 값이다. 게임의 특성에 따라 효율적인 행동이라는 것은 차이가 있을 수 있기 때문에 필요하다. 효율값을 계산한 이후 행동을 선택할 때 비교적 효율이 좋은 행동을 선택하여 사용자에게 대응한다. 효율값은 비교적 좋은 행동임을 판단할 때 사용되는 기준이 된다.

3.4 유사한 상태의 클러스터링

앞에서 설명한 절차를 통해 저장된 행동 정보를 살펴보면, 상태 정보들은 거리, 방향, 수치, 시간의 4가지 정보로 구성된다. 따라서 다양한 형태로 구성된 상태 정보를 뚜렷하게 정의할 수 없다. 이로 인해 중복되거나 거의 비슷한 상태들의 무의미한 정보들도 함께 저장되어 프로세스의 과부하가 발생한다.

이를 해결하기 위해 클러스터링 기법을 도입한다. 클러스터링을 통해 상태 정보를 군집화하여 대표 상태값을 가진 정보로 정의하고, 의미 없이 비슷한 상태 정보들 때문에 생기는 과부하의 문제점

을 해결할 수 있다[9].

상태 정보들 중 거리와 체력수치에 대한 정보는 뚜렷한 값으로 정의할 수 없기 때문에 군집화를 진행하는데 필요한 정보로 활용된다. 예를들어 10M터의 거리와 10.1M의 거리는 비슷한 상태이지만 전혀 다른 상태 정보로 저장되기 때문에 정의하기 힘들고 불필요한 정보의 양이 많아진다. 체력 요소 역시 거리의 요소와 마찬가지로 체력이 100%일 경우와 99%일 경우 비슷한 상태로 볼 수 있지만 전혀 다른 상태 정보가 된다. 따라서 거리와 체력 수치의 정보를 이용하여 유사성을 판단하고 클러스터링을 한다. 클러스터링은 [상태-행동] 수집 테이블에 두 개 이상의 정보가 저장되었을 때부터 진행된다. x_i 의 값은 거리에 대한 수치, x_j 의 값은 체력에 대한 수치이고 (거리, 체력)의 관계로 묶는다. 각각의 상태 정보들 간의 거리 크기에 따라 유사성을 판단하여 클러스터링을 진행한다. 거리 크기에 따라 유사성을 판단하고 구하는 식은 [수식 2]와 같다.

$$d_p(x_i, x_j) = \sqrt[p]{\sum_{k=1}^d (|x_{i,k} - x_{j,k}|^p)}$$

[수식 2] 거리크기에 따른 유사성 계산

유사성을 판단할 때 거리와 체력의 두 가지 정보로 2차원에서 계산하므로 p 값은 2 이다. 계산된 결과를 통해 유사한 상태들을 군집화 하고 군집화된 상태의 중심값이 대표값이 되어 갱신한다.

3.5 적응형 NPC 행동 데이터베이스

앞에서 설명한 내용의 결과들을 모아서 최종적으로 적응형 NPC 행동 데이터베이스를 구성한다. NPC 행동 데이터베이스의 구성은 [표 2]에서 볼 수 있다.

[표 2] 적응형 NPC 행동 데이터베이스

상태 (State)	군집화로 얻은 상태정보
행동 (Action)	사용자의 반응으로 얻은 행동 정보
효율값 (Efficiency)	상호작용 결과로 얻은 수치
행동 선택의 빈도 수 (Frequency)	행동 선택되면 값이 증가한다.
마지막 사용시간 (Last Time)	마지막으로 사용했을 때의 시간정보

행동 선택의 빈도수는 행동 데이터베이스에서 행동 정보를 얼마나 사용하는 가를 나타내는 값으로 행동이 선택될 때마다 값이 증가한다. 마지막 사용 시간은 행동 선택이 마지막으로 이루어진 시간을 의미하는 정보이다. 이 두 가지는 효율값이 좋지 못한 행동을 탐색하고 판단하는 요소로 활용된다. 게임 시스템은 NPC 행동 효율 산출 알고리즘을 통해 행동 정보들이 NPC 행동 데이터베이스에 실시간으로 저장, 관리된다. 좋은 효율값의 행동을 선택해서 사용자에게 적절하게 대응하는 적응형 NPC가 생성된다.

4. 실험 및 결과 분석

4.1 실험 방법 및 내용

본 실험의 목표는 본 논문에서 제안한 행동 정보 관리 기법을 통해 적응형 NPC가 생성되고 제대로 동작하는지를 살펴보는 것이다. 실험하기 위해 적응기법을 적용하는 간단한 2D 액션 게임을 개발하였다. 게임에서 사용자는 자신의 캐릭터를 자유롭게 사용할 수 있고, 기술들의 조합을 통해 NPC를 공격한다. 체력 수치를 0으로 만들면 캐릭터는 소멸하고, 다시 등장해서 전투를 한다. 이 과정을 반복하면서 NPC가 사용자의 행동 성향과 수준에 따라 적응해 가는 과정을 관찰한다. NPC와 사용자 캐릭터는 14 가지 기술과 이들을 최대 5개 까지 조합하여 사용할 수 있으므로 다양한 행동이 가능하다. 사용자 캐릭터가 사용하는 구체적인 행동 정보는 [표 3]과 같다.

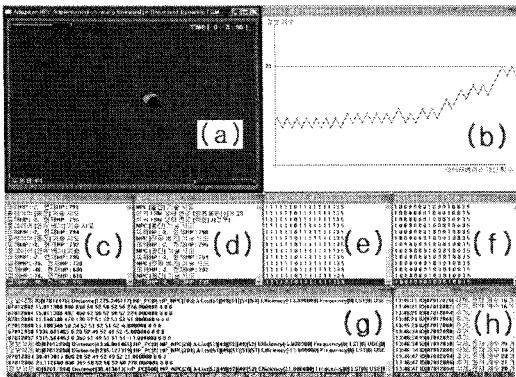
[표 3] 캐릭터 행동 목록

기술명	조작키	데미지	효 과
일반공격	F	15	특수 효과 없음
혼란	R	15	적의 방향을 반대로 돌림
밀쳐내기	A	20	적을 한 칸 뒤로 밀어냄
반격	S	-	상대 피해를 2배로 되돌려줌
뒤 잡기	D	10	적의 뒤로 빠르게 돌아감
회피	T	-	3초간 모든 공격 회피
도주	G	-	3초간 달리기 속도 증가
돌진	1	15	일정 거리를 빠르게 달려가 공격
강배기	2	15	1.5배 데미지
찌르기	3	10	일정 떨어진 거리에서 공격 가능
연속배기	4	30	2배 데미지
진공배기	Q	20	일정 거리에서 검기 공격
기습	W	15	적의 뒤에서 공격한다면 데미지 30
기절	E	25	3초간 적을 기절(행동불능)시킴

본 실험에서 적응형 NPC의 행동 데이터베이스에 행동 정보가 잘 저장되는지, 사용자와 상호작용할 경우 사용자에게 어느 정도 적응하는지 관찰한다. 로그 파일을 통해 게임의 결과를 분석하는 것이 가능하다.

4.2 구현된 실험 결과

NPC 효율 산출 알고리즘을 추가하여 구현한 게임의 결과는 다음과 같다.



[그림 3] 실험 시스템 화면 구성

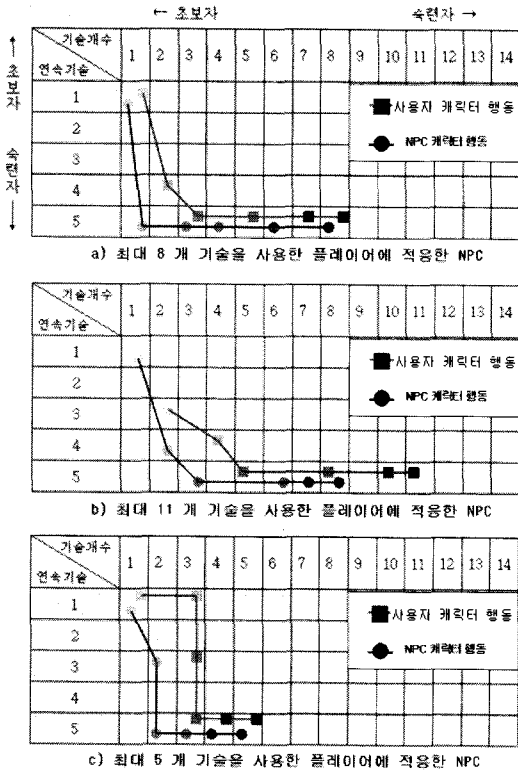
사용자 캐릭터와 적응형 NPC의 실시간 전투를 통해 로그파일의 정보와 그래프가 실시간으로 변환

다. (a)는 구현된 게임으로 사용자와 적응형 NPC의 전투를 실시간으로 볼 수 있다. (b)는 행동 정보의 개수 변화에 대한 그래프이다. 사용자를 통해 행동 정보가 적응형 NPC 행동 데이터베이스에 저장, 관리 되는 것을 실시간으로 확인할 수 있다. (c), (d)는 각각 사용자 캐릭터와 NPC의 행동이 실시간으로 기록되는 로그파일이다. (e), (f)는 캐릭터의 사용기술과 연속기술의 개수가 실시간으로 표기된다. (c), (d), (e), (f)의 로그파일을 통해 사용자와 NPC의 행동 성향을 분석할 수 있다. (g)의 로그파일에는 NPC 행동 데이터베이스가 갱신될 때 어떤 정보들을 저장하는지를 알 수 있다. (h)는 알고리즘에서 클러스터링을 통해 상태 정보들이 실시간으로 저장, 삭제되는 것을 확인할 수 있다. 사용자와 NPC의 로그파일의 결과를 분석하여 적응형 NPC 생성을 입증한다.

4.3 캐릭터 행동 분석

실험 결과의 로그파일을 통해 사용자 캐릭터와 NPC의 실시간 행동 변화를 분석한다. 최대 사용 기술의 개수와 연속기술은 사용자의 수준을 구분 짓는 기준이 된다. 연속기술은 행동이 끝나기 전 정확한 시간에 맞춰 다음 기술을 사용해야 진행된다. 그리고 연속으로 사용할 수 있는 행동과 없는 행동이 나뉘어져 있어서 연속기술을 잘 활용할수록 조작이 뛰어난 사용자임을 알 수 있다. 또, 많은 기술을 다양하게 사용할수록 숙련자이고 사용기술이 적고 행동이 단순할수록 초보자이다. 다양한 실험자 중에서 대표로 3가지 유형의 사용자를 선정한다. NPC가 사용자에게 어느 정도 유사하게 적응하는지를 캐릭터 행동 분석표를 통해 살펴본다. 실험자와 적응형 NPC의 행동 정보가 기록된 로그파일을 참고해서 시간에 따라 최대 사용 기술의 개수와 연속기술 개수의 변화를 행동 분석표에 표시한다. a) 실험자는 최대 사용기술이 8 개로 보통 수준의 사용자다. b)의 실험자는 최대 11 개의 기술을 조합하여 사용한 숙련자이고, c)의 실험자는 5개의 기술밖에 사용하지 못한 초보자이다. 행동 분석표를 이용해 캐릭터의 행동 성향을 분석하여 적응형 NPC가 각각의 실험자에게 얼마나 유사하

게 적응했는지를 판단한다. 실험자 a), b), c)의 행동 분석표 결과는 [그림 4]와 같다.



[그림 4] 캐릭터 행동 분석표

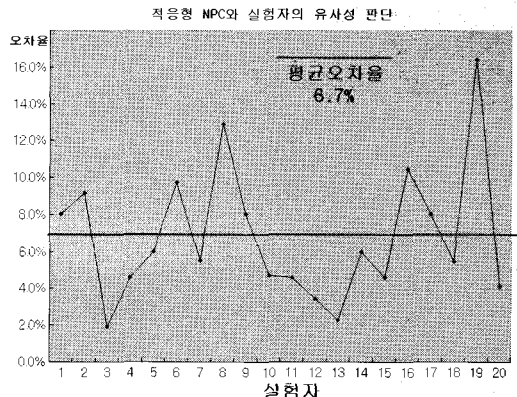
로그파일을 통해 일정 시간마다 누적된 기술개수와 연속기술의 개수를 행동 분석표에 표시한다. 실험의 총 경과 시간에서 6개 구간으로 나누어 도형을 표기하고 그래프를 만든다. 도형의 색깔이 진할수록 많은 시간이 경과했다는 것을 의미한다. 네모로 표기된 도형의 그래프 곡선은 사용자 캐릭터의 행동을 통해 사용한 기술개수와 연속기술이 얼마나 누적되었는지 알 수 있다. 사용자는 시간이 지날수록 자신의 역량에 맞는 행동 패턴을 보인다. 따라서 대체로 시간이 지날수록 사용기술과 연속기술의 수가 증가하는 그래프가 만들어진다. 적응형 NPC는 사용기술의 개수와 연속기술의 개수가 사용자의 수준과 비슷하게 맞춰가며, 효율값을 통해 비교적 좋은 행동들을 선택해서 사용자에게 적응해 간다. 사용자가 초급자면 초급자의 수준에 맞게, 숙련자면 숙련자 수

준에 맞게 행동을 조합하여 사용한다.

행동 분석표를 통해 유사성을 판단하는 방법으로 행동 분석표에 표시되어 있는 각 구간의 거리값을 이용한다. 같은 구간에 있을 경우 거리값이 0이므로 오차율은 0%이다. 분석표에서 가장 멀리 떨어져 있는 두 구간은 (기술개수1, 연속기술1)과 (기술개수14, 연속기술5) 사이의 거리로 약 14.52이고 오차율은 100%이다. 행동 분석표에 표시되어 있는 각 구간마다의 평균 거리값을 구하여 %단위로 오차율을 계산할 수 있다.

실험 a)에서 적응형 NPC는 보통 수준의 사용자와 약 4.11%의 오차율을 보였고 사용자와 거의 유사한 행동 조합으로 대응했다. 실험 c)에서 초보자 수준의 사용자에게 적합한 NPC는 5.47%의 오차율을 보였다. 숙련자가 실험한 실험 b)에서는 약 16.4%의 오차율이 발생했다. 본 연구의 알고리즘을 적용한 적응형 NPC는 저장된 다양한 행동 정보들 중 비교적 효율값이 좋은 행동들만을 선택해서 사용한다. 숙련자는 행동 조합을 통해 다양한 행동 패턴을 구사한다. 따라서 적응형 NPC는 숙련자의 행동 조합 중에 효율값이 좋지 않은 행동 정보들이 저장되지 않기 때문에 생기는 문제로 확인되었다. 또, 숙련자와 비슷한 수준까지 적응하기 위해서는 시간이 많이 소요되었다.

임의의 실험자 20명을 대상으로 실험을 진행하였다. 각 실험자에게 적합한 NPC의 오차율을 구하고 평균 오차율을 계산한 결과는 다음 [그림 5]와 같다.

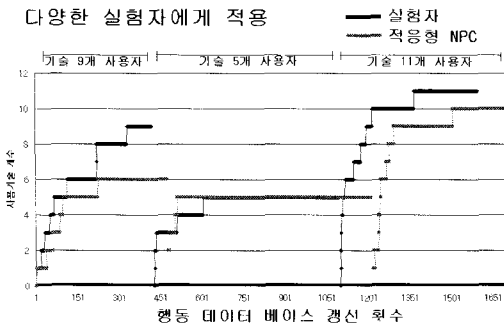


[그림 5] 적응형 NPC 유사성 판단

[그림 5]는 수준이 다른 임의의 다양한 실험자로 실험한 결과이다. 적응형 NPC와 실험자의 유사성을 판단했을 때 평균 오차율 약 6.7%의 결과를 얻었다.

4.4 다양한 실험자에게 실험한 결과

수준이 서로 다른 실험자가 실험을 한다면 적응형 NPC는 실험자의 수준에 맞게 대응해야 한다. 실험자가 바뀌면 실험자의 수준에 따라 NPC도 변해야 한다. 중급자, 초급자, 숙련자의 순서로 실험을 진행했을 때 동일한 적응형 NPC가 어떤 결과를 얻는지를 실험한 결과를 [그림 6]에서 확인할 수 있다.



[그림 6] 다양한 실험자에게 적용한 결과

[그림 6]의 결과를 살펴보면 실험자가 사용한 기술의 개수에 따라 적응형 NPC도 사용기술의 개수를 바꾸어가며 적응해 가는 것을 확인할 수 있다. 그래프에서 적응형 NPC는 기술 9 개를 사용하는 실험자에게 6 개의 기술을 사용할 때까지 적응하였다. 최대 5개의 기술을 사용하는 실험자로 바뀌었을 때 적응형 NPC는 행동 데이터베이스에 저장된 행동 정보들이 있으므로 빠른 시간에 기술 5개를 사용하여 실험자와 동일한 수준으로 적응하는 것을 확인할 수 있다. 마지막으로 11 개의 기술을 사용하는 실험자에게는 10 개의 기술을 사용하도록 적응하였다. 실험이 진행되면서 행동 데이터베이스에 다양한 행동들이 저장되어 있고 많은 시간이 경과 되었으므로 11 개의 기술을 사용하는 숙련자에게

도 거의 비슷하게 적응해가는 것을 확인할 수 있었다.

5. 결론 및 향후 과제

본 논문에서는 적응형 NPC를 생성하는 하나의 방법으로 행동 정보 관리 기법을 제안했다. 게임이 진행되면서 사용자 캐릭터의 행동을 관찰한 적응형 NPC는 다양한 정보들을 수집한다. 행동 결과 정보를 이용하여 효율값을 산출하고, 상태정보들의 클러스터링을 통해 새로운 군집으로 정의된 상태 정보들을 얻을 수 있다. 최종적으로 [상태-행동], 효율값, 행동 선택 빈도수, 행동의 마지막 사용시간의 정보를 이용하여 적응형 NPC 행동 데이터베이스를 구축한다. 이 알고리즘을 지속적으로 갱신하면서 행동 데이터베이스에 효율적인 새로운 행동 정보를 저장하고 관리한다. NPC는 효율값을 이용하여 비교적 효율이 좋은 행동들이 선택하여 사용자에게 대응한다. 알고리즘을 지속적으로 갱신하기 때문에 적응형 NPC는 게임에서 실시간으로 행동 변화가 가능하다. 적응형 NPC는 실시간 관찰을 통해 행동 정보를 저장, 관리하므로 결과 분석을 통해 적응형 NPC는 사용기술 개수와 연속기술 개수를 사용자와 비슷한 수준으로 바꾸어가며 적응하는 것을 확인할 수 있었다. 사용자의 수준에 따라 적응하는데 소요되는 시간 차이가 있다. 숙련자일수록 많은 시간이 걸리고 초보자일수록 짧은 시간에 적응이 가능하다. 대부분의 실험에서 적응형 NPC는 사용자와 유사성을 판단할 때 평균 오차율이 6.7% 이내로 거의 동일하게 적응하는 결과를 얻을 수 있었다. 또, 동일한 적응형 NPC를 다양한 실험자에게 적용한 실험 결과를 통해, 사용자의 수준에 따라 사용자와 비슷한 행동 조합을 통해 적응해 가는 것을 확인하였다.

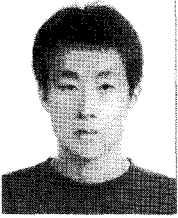
본 연구에서 적응형 NPC를 생성하는 행동 정보 관리 기법을 통해 실시간으로 행동 변화가 가능하고 적응해 가는 NPC를 얻을 수 있다. 또, 적응형 NPC 행동 효율 산출 알고리즘이 NPC 내부의 조작모듈과 같이 구성되어 있으므로 범용적인 게임

장르에 적용이 가능하다. 그리고 실험 결과를 통해 다양한 수준의 사용자에게 적절하게 대응하는 적응형 NPC를 생성할 수 있었다.

이를 통해 획일적인 행동을 하는 NPC에서 탈피하여 NPC로 인한 게임의 지루함과 그에 따른 여러 가지 문제점을 해결할 수 있을 것이다. 다양한 사용자에게 적용 가능하므로 게임의 재미요소로 활용이 가능하다.

참고문헌

- [1] 워자드소프트, "AI 에이전트 NPC 기반의 게임 스타일 이식 엔진 개발" 보고서, 정보통신부, 2004.
- [2] 앤드류 롤링스, 어니스트 아담스, 게임기획개론, 제우미디어, 2004.
- [3] 엄상원, "인공지능 기법을 이용한 사용자 상호작용 게임 난이도 조절 알고리즘", 중앙대학교 첨단대학원 석사 학위 논문, 2003
- [4] P. Spronck, E. Postma, "Adaptive game AI with dynamic scripting", Machine Learning archive Volume 63, Issue 3, p217-248, June 2006
- [5] P. Spronck, "Adaptive Game AI ", SIKS Dissertation Series 2005-06, 2005
- [6] M. Ponsen, P. Spronck, "Improving Adaptive Game AI with Evolutionary Learning", University of Wolverhampton. (Presented at the CGAIDE 2004).
Computer Games: Artificial Intelligence, Design and Education (CGAIDE 2004, p389-396)
- [7] J. Dinerstein, P. K. Egbert, "Fast multi-level adaptation for interactive autonomous characters" ,ACM Transactions on Graphics(TOG), pp.262-288, April 2005.
- [8] 조병현, "지능형 게임 캐릭터를 위한 학습 및 적용방법에 관한 연구", 국민대학교 대학원 박사학위 논문, 2005.
- [9] 클러스터링 알고리즘
<http://home.dei.polimi.it/matteucc/Clustering/tut>



김 나 라 (Nara Kim)

2005.3 한국IT전문학교 게임프로그래밍학과 (공학사)
2008.2 동국대학교 영상대학원 멀티미디어학과 게임제작
전공 (예술 공학 석사)

관심분야 : 게임기획, 게임인공지능, 적응형 게임캐릭터



조 경 은 (Kyungeun Cho)

1993.2 동국대학교 전자계산학과(공학사)
1995.2 동국대학교 컴퓨터공학과 대학원(공학석사)
2001.8 동국대학교 컴퓨터공학과 대학원(공학박사)
2002.3~2003.2 안양대 디지털미디어학부 강의전담교수
2003.3~2003.8 영산대학교 게임공학과 전임강사
2003.9~2005.8 동국대 정보산업대학 컴퓨터멀티미디어
공학과 전임강사
2005.9~2006.8 동국대 정보산업대학 컴퓨터멀티미디어
공학과 조교수
2006.9~현재 동국대 영상미디어대학 게임멀티미디어공
학과 조교수

관심분야 : 컴퓨터 게임 알고리즘, 게임 인공지능, 멀티
미디어 정보처리



엄 기 현 (Kyhyun Um)

1975.2 서울대학교 공과대학 응용수학과 공학사
1997.2 한국과학기술원 전산학과 이학석사
1994.2 서울대학교 대학원 컴퓨터공학과 공학박사
1978.3~2006.8 동국대 컴퓨터멀티미디어공학과 정교수
2006.9~현재 동국대학교 영상미디어대학 게임멀티미디어
공학과 교수

1995.3~1999.2 동국대학교 정보관리처장
2001.3~2003.2 동국대학교 정보산업대학 학장
1998.9~2000.8 한국정보과학회 데이터베이스연구회 위원장
1999.4~2005.3 Int. Conf. on Database Systems for
Advanced Applications Steering Committee
위원
2001.1~2002.12 한국정보과학회 논문지편집부위원장(데이
타베이스)
1998.1~현재 한국멀티미디어학회 부회장, 수석부회장, 회
장 역임, 자문위원(현재)
2004.1~현재 한국 게임학회 부회장 역임, 자문위원(현재)

관심분야 : 게임시스템 설계, 멀티미디어 데이터베이스,
멀티미디어 정보시스템