

## 자율 캐릭터를 위한 상태기반 상황인지 기법

김형일

나사렛대학교 멀티미디어학과  
hkim@kornu.ac.kr

### State based Context Awareness Method for Non-Player Character

Hyung-Il Kim

Dept. of Multimedia, Korea Nazarene University

#### 요약

자율 캐릭터는 게임에서 사용자 캐릭터와 협업하여 게임을 진행하거나, 사용자 캐릭터와 대전을 수행한다. 자율 캐릭터의 자연스러운 행동 제어를 위해, 본 논문에서는 상황인지 기법을 제안한다. 상황인지 기법은 자율 캐릭터의 현재 상태에서 활용할 수 있는 상황 정보를 분석한 후, 현재 상황에 맞는 행동을 자동 생성한다. 상황인지 기법은 자율 캐릭터의 현재 상황에서 발생하는 상황 요소의 정보 가치를 분석하고, 분석된 상황 요소 정보를 활용하여 캐릭터의 자연스러운 행동을 생성한다. 본 실험에서 상황인지 기법이 규칙기반 기법보다 평균 39%의 성능 향상을 나타냈고, 정보획득 기법보다는 평균 8%의 성능 향상을 나타냈다.

#### ABSTRACT

The non-player character(NPC) either cooperates with the player character to proceed the game or fight against the game player. This paper proposes the context awareness method for the natural action control of the same NPC. The context awareness method analyzes the context information of the NPC that can be utilized in the current status and creates the actions which suit the current context automatically. The context awareness method analyzes the information value of the context elements of the NPC which occur in the current context and creates natural actions of the character by utilizing the analyzed context element information. In this experiment, average performance improved by 39% and by 8% in the context awareness method as compared to the rule-based method and information gain method, respectively.

**Keywords** : Non-Player Character(자율 캐릭터), Context Awareness(상황인지), Information Gain(정보획득), Machine Learning(기계학습)

Received: Dec. 24, 2013 Accepted: Jan. 15, 2014  
Corresponding Author: Hyung-Il Kim(Korea Nazarene University)  
E-mail: hkim@kornu.ac.kr

© The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

ISSN: 1598-4540 / eISSN: 2287-8211

## 1. 서론

게임산업은 컴퓨터가 활성화됨에 따라 급속한 성장을 지속하고 있다[1]. 게임은 장르마다 특징이 있으며[2], 장르에 따른 환경 특성은 다양한 게임 요소들을 발생시킨다[3]. 다양한 장르에서 동일하게 유지되는 요소는 게임 사용자와 상호작용을 발생시키는 게임 캐릭터이다[4]. 게임 캐릭터는 사용자에게 흥미를 유발시키는 중요한 요소이다[5].

자율 캐릭터의 역할은 게임 내에서 게임 사용자의 캐릭터와 협업하여 게임을 진행하거나, 게임 사용자의 캐릭터와 대전을 수행한다. 게임 사용자가 게임을 진행할 때, 자율 캐릭터와의 상호작용은 필수적인 요소이다. 특정 환경에서 상호작용이 미약할 경우, 사용자는 해당 환경에 몰입하기 어렵고 지루함을 느낀다[6]. 그러므로 게임 사용자의 흥미를 유발시키기 위해, 게임 상황에 적합한 행동을 전개할 수 있는 자율 캐릭터가 필요하다[7]. 그러므로 게임 사용자에게 게임에 대한 흥미를 지속적으로 유지시키기 위해서는 지능적 자율 캐릭터의 행동 제어가 필요하다[8]. 지능적 자율 캐릭터는 확립화된 단순 규칙에 의한 반복적 행동을 배제하고, 게임 상황에 적합한 다양한 행동을 생성해야 한다[9]. 또한 지능적 자율 캐릭터는 사용자의 성향이나 수준에 맞게 적절한 대응을 수행할 수 있어야 한다.

자율 캐릭터의 행동 제어에 활용하는 규칙기반 방식의 대표적인 기술은 유한상태기계 기법이다. 유한상태기계 기법은 자율 캐릭터의 행동 제어를 조건절과 귀결절을 이용하여 규칙화한다. 유한상태기계 기법에서 활용하는 가장 중요한 두 가지 요소는 상태와 전이이다. 상태는 자율 캐릭터가 직면한 상황이고, 전이는 자율 캐릭터의 상황 변화를 의미한다. 상태와 전이를 이용하여 자율 캐릭터의 행동을 제어하면 행동 특성이 명료해지는 장점이 있다. 복잡한 행동 제어가 요구되는 자율 캐릭터에서 유한상태기계 기법을 적용할 경우, 캐릭터 디자이너는 자율 캐릭터에 적합한 모든 규칙을 생성하

여야 한다. 그러므로 캐릭터 디자이너는 모든 게임 상황에서 발생 가능한 자율 캐릭터의 행동을 예측하여 설계해야 한다. 게임 진행에 적합한 자율 캐릭터의 모든 행동 제어를 캐릭터 디자이너가 완성한다는 것은 매우 어려운 작업이다.

자율 캐릭터의 행동을 게임 진행에 적합하게 발생시키기 위해서는 자율 캐릭터의 현재 상태에서 발생할 수 있는 환경 요소를 활용하면 효과적이다. 예를 들면 적을 경계하는 자율 캐릭터가 적을 만났을 경우 무조건 공격을 진행하는 것이 아니라, 자율 캐릭터의 화력, 체력, 시간 등을 고려하여 적절하게 적에게 대응하는 것이 현실적이다.

본 논문에서는 자율 캐릭터의 행동을 게임 상황에 맞게 발생시키는 상황인지 기법을 제안한다. 상황인지 기법은 자율 캐릭터의 현재 상태에서 활용될 수 있는 상황 정보를 분석하여 적절한 행동을 발생시키는 장점이 있다.

## 2. 관련 연구

게임의 수요 증가로 게임 사용자들은 현실적인 다양한 형태의 게임 진행을 요구하고 있으며, 게임 사용자의 기대에 부응하기 위해서는 자연스런 행동을 발생시키는 지능적 자율 캐릭터의 설계가 필요하다[10]. 그러나 자율 캐릭터의 지능화 설계는 게임의 어려운 분야에 속한다[11]. 자율 캐릭터의 행동 제어에 활용되는 가장 대표적인 방식은 유한상태기계 기법이며, 유한상태기계 기법은 규칙기반 방식을 기본 구조로 활용한다[12]. 게임 규모가 커짐에 따라 자율 캐릭터는 많은 규칙을 필요로 하지만, 규칙기반 방식은 자율 캐릭터의 행동 설계에 한계가 있다[13].

Namee 등[14]은 자율 캐릭터가 사용자의 게임 시점에 존재하지 않아도 행동 제어를 진행하여, 사용자의 게임 시점에 자율 캐릭터가 다시 존재하여도 자연스러움을 유지시키는 행동 제어 기법을 제안하였다. Robert 등[15]은 자율 캐릭터의 자연스

러운 활동을 발생시키기 위해 동기, 분류, 행동이라는 세 가지 모듈을 계층적으로 구성하는 방법을 제안하였다. 제안된 제어계층 시스템은 자율 캐릭터의 감지 정보에 따라 행동을 선택하고, 선택된 행동을 기반으로 행동의 세기를 결정한다. Mateas 등[16]은 자율 캐릭터의 반응행동 언어인 ABL을 제안하였다. ABL은 외부 신호를 입력받는 감지기관에 대한 일반화된 구조를 제공하여 활성행동계층화 구조를 제공한다. DeLap 등[17]은 실시간 검사 시스템을 제안하였다. 제안된 시스템은 게임 조건에 대한 임계치를 설정하여 게임 환경이 특정 임계치를 넘을 경우, 게임 규칙에 문제가 있는 것으로 예측한다.

복잡한 게임 구조에서 단순한 구조의 규칙기반 방식만을 이용하여 자율 캐릭터의 다양한 행동을 설계하기 위해서는 많은 종류의 규칙 생성이 필요하다. 이러한 자율 캐릭터의 규칙 생성은 전적으로 게임 디자이너가 담당해야 함으로, 자율 캐릭터의 자연스러운 행동을 발생시키는 제어 규칙을 모두 생성한다는 것은 현실적으로 불가능하다. 이와 같은 연구들은 캐릭터 디자이너가 캐릭터의 행동 제어를 모두 결정해야 한다는 전제가 필요하여, 자연스러운 행동 생성에 한계가 있다.

이러한 문제점을 해결하기 위해서는 상황인지 방식을 적용하는 것이 효과적이다. 상황인지는 인공지능에서 출발한 분야로 객체간의 상호작용을 통해 새로운 정보를 생성하는 것을 목적으로 한다. Feng[18] 등은 상황 모델과 에이전트 모델을 이용한 CaDS 시스템을 제안하여 상황인지 결정기에 활용하였다. Lee[19] 등은 MMORPG에서 자율 캐릭터의 자연스러운 행동을 위한 지능화 프레임워크에 대해 분석하였다. 기존의 상황인지 연구들은 인공지능에서 사용하는 학습 모델을 자율 캐릭터에 적용하여 캐릭터를 지능화하고, 캐릭터의 지능화 모델은 단일한 형태로 표현된다.

본 논문에서는 자율 캐릭터의 다양한 행동 생성과 자연스러운 행동 유지를 위해, 캐릭터 제어를 다중적으로 확장한 상황인지 기법을 제안한다. 본

논문에서 제안한 기법은 게임 상황에서 발생하는 요소를 분석하여 자율 캐릭터의 다중 행동을 생성한 후, 게임 상황에 적합한 행동을 발생시킬 수 있는 장점이 있다.

### 3. 상황인지

#### 3.1 게임 환경에서의 자율 캐릭터

자율 캐릭터가 게임 상황에 적합한 행동을 자연스럽게 발생시켜야, 사용자는 게임에 몰입할 수 있다. 전쟁 게임에서 자율 캐릭터가 경계 상태에서 다른 상태로 변경할 경우, 적이 나타났다고 게임 환경을 고려하지 않고 무조건 공격 상태로 전환하는 것은 부자연스러울 수 있다. 전쟁 게임에서 자율 캐릭터의 행동 결정에 사용할 수 있는 환경 요소는 무기, 시간, 음식, 체력, 기후 등일 것이다. 공격에 적합하지 않은 상태라면 자율 캐릭터는 공격 상태보다는 복귀 상태로 전이하는 것이 현실적일 것이다. 그러므로 자율 캐릭터의 현재 상황에서 상태를 유지할 것인지 상태를 전이할 것인지에 대한 결정에 상황 정보를 이용하면 자율 캐릭터의 행동에 자연스러움을 유지할 수 있다.

다양한 게임 상황에서 자율 캐릭터의 자연스러운 행동을 유지하기 위해서, 자율 캐릭터는 게임 상황에 적합한 다양한 대처 방법을 소유해야 한다. 본 논문에서는 자율 캐릭터의 행동을 게임 상황에 적합하게 발생시킬 수 있는 상황인지 기법을 제안한다. 상황인지 기법은 게임 환경에서 발생하는 게임 요소들의 정보 가치를 이용하여, 게임 상황에 적합한 자율 캐릭터의 행동을 생성할 수 있는 장점이 있다.

#### 3.2 정보획득의 개념

게임 상황에서 발생하는 환경 요소에 대한 정보 가치 예측에 정보획득 기법을 이용하면 효과적이다. 정보획득 기법은 조건에 따라 데이터를 분류하

여 결과트리를 생성한다. 정보획득 기법에서 생성한 결과트리는 규칙 전환이 쉽다는 특징이 있다. 결과트리는 속성 순서에 의미가 있으며, 속성을 이용한 최적화된 결과를 도출한다.

정보획득 기법에서는 결과트리 생성을 위해 *Entropy*를 이용하여 속성 결정에 사용하며, *Entropy*의 공식은 (eq. 1)과 같다. (eq. 1)에 나타난 *D*는 전체 데이터 집합이고 *c*는 분류 클래스를 의미한다.  $P_i$ 는 전체 데이터 집합 *D*에 대한 *i*번째 클래스 *c*의 확률이다.

$$Entropy(D) = \sum_{i=1}^c (-p_i \log_2 p_i) \quad (\text{eq. 1})$$

*Entropy*에 의해 생성된 결과는 전체 데이터의 혼합도를 나타낸다. 그러므로 서로 다른 다양한 부류들이 비슷하게 나타난다면 *Entropy*는 높은 값으로 표현된다. 그러나 특정 부류의 점유도가 증가하게 되면 *Entropy*는 낮아지는 특성이 있다. *Entropy*는 부류들을 다중 분할함으로 특정 데이터들의 분류를 명확히 수행할 수 있다. 그러므로 다중 분할을 이용한 *Entropy*는 데이터 분류를 극명하게 나타내는 특징을 가진다.

*Gain*은 현재 상황에서 특정 속성이 선택되었을 경우, *Entropy*의 현재 상황에 대한 상대적 순정도를 나타낸다. 선택된 속성에 의해 *Entropy*의 혼합도가 낮아지면 *Gain*의 순정도는 높아지기 때문에, 선택된 속성의 정보 가치는 높다고 추정할 수 있다. 그러므로 *Gain*을 이용하여 속성 순정도를 측정하고, 순정도가 높은 속성을 이용하여 데이터를 분류하는 것이 효과적이다. *Gain*의 수식을 (eq. 2)에 나타내었으며, (eq. 2)에 나타난 *Gain*은 *Entropy*를 기반으로 속성 순정도 측정한다. (eq. 2)에 나타난 *a*는 속성을 의미하고, *D*는 전체 데이터 집합을 나타낸다. 그러므로 *Gain(D,a)*는 속성 *a*가 적용된 상태에서의 전체 데이터 집합 *D*의 순정도를 나타낸다. *Entropy(D)*는 *D*의 혼합도를 나타내고, *Entropy(D<sub>v</sub>)*는 *D<sub>v</sub>*의 혼합도를 나타낸다.

*v*는 속성 *a*에 속하는 각각의 속성값을 나타내고, *D<sub>v</sub>*는 속성 *a*의 특정 속성값 *v*를 지닌 데이터 집합이다.

$$Gain(D,a) = Entropy(D) -$$

$$\sum_{v \in Values(a)} \frac{|D_v|}{|D|} Entropy(D_v) \quad (\text{eq. 2})$$

### 3.3 정보획득의 활용

정보획득의 활용에 대해, 간단한 예제를 들어 설명하면 다음과 같다. 특정 데이터 집합 *D*에 *A*, *B*, *C*라는 데이터 속성이 존재하고 분류 속성으로 *Class*가 존재한다고 가정하자. 이러한 가정하에 데이터 속성값은 *T*와 *F*로 표현되고, 분류 속성값은 *Good*와 *Bad*로 표현된다. 이와 같은 전제하의 예제를 [Table 1]에 나타냈다. 전체 데이터 집합에 나타난 8개의 객체는 분류 속성을 기준으로, *Good*인 경우는 5개이고 *Bad*인 경우는 3개이므로 전체 데이터의 정보 혼합도를 측정하는 *Entropy*은 *Entropy(5,3)*를 활용한다.

$$Entropy(5,3) = -\frac{5}{8}(\log_2 \frac{5}{8}) - \frac{3}{8}(\log_2 \frac{3}{8}) = 0.954 \text{ 이다.}$$

[Table 1] Training examples

A	B	C	Class
T	T	T	Good
T	T	F	Good
T	F	T	Good
T	F	F	Bad
F	T	T	Good
F	T	F	Bad
F	F	T	Good
F	F	F	Bad

속성 *C*를 이용한 *Entropy*는 *C*의 값이 *T*인 경우와 *F*인 경우로 나뉜다. 속성 *C*가 *T*인 상태에서 *Good*에 해당하는 것은 4개이고, *Bad*에 해당하는 것은 0개이므로 *C*의 정보 혼합도에 대한 *Entropy*는 *Entropy(4,0)*을 이용하여 계산한다. 그러므로

$$Entropy(4,0) = -\frac{4}{4}(\log_2 \frac{4}{4}) - \frac{0}{4}(\log_2 \frac{0}{4}) = 0 \text{ 이 다 .}$$

속성 C가 F인 상태에서 Good에 해당하는 것은 1개이고, Bad에 해당하는 것은 3개이므로

$$Entropy(1,3) = -\frac{1}{4}(\log_2 \frac{1}{4}) - \frac{3}{4}(\log_2 \frac{3}{4}) = 0.811 \text{ 이}$$

다. 그러므로 전체 집합 D에 대한 속성 C의 Gain은  $Gain(D,C)$ 으로 계산될 수 있다. 그러므로

$$Gain(D,C) = 0.954 - (\frac{4}{8} \cdot 0) - (\frac{4}{8} \cdot 0.811) = 0.548$$

이다. 이와 같은 방법으로 A와 B의 Gain을 구하면  $Gain(D,A) = 0.048$ 이고  $Gain(D,B) = 0.048$ 이다. 그러므로 A, B, C 속성 중에 가장 Gain이 높은 C를 먼저 활용하는 것이 분류에 효과적이다. 이러한 방법을 C가 T인 상태와 F인 상태에서 A와 B를 적용하여 다시 Gain을 구해 두 번째 분류 속성을 선택한다. 이러한 결과로, 최종적 결과트리는 C, B, A나 C, A, B 순서로 표현된 트리이다. 그러므로 최종 결과트리는 해당 데이터 집합에서 생성될 수 있는 최적화된 트리이다.

### 3.4 상태선택 모델

자율 캐릭터의 다양한 행동 생성을 위해 상황인지 기법은 상태선택 모델을 사용한다. 상태선택 모델은 다양한 전이에 현실성을 유지할 수 있도록 현실 가능한 전이 상황에 대한 정보 가치를 고려하여, 불완전한 상태 전이를 배제할 수 있는 장점이 있다.

상태선택 모델을 (eq. 3)에 나타내었다. 상태선택 모델인  $Selector(S)$ 는 전이 상태  $S$ 를 선택하기 위한 모델이며, (eq. 3)에 나타난  $D_i$ 는 각 상태에 해당하는 데이터 집합이다. 그러므로 상태 선택을 위해 활용한  $Entropy(D_i)$ 는  $D_i$ 의 정보 혼합도를 의미하고,  $Entropy(D_{i_v})$ 는  $D_{i_v}$ 의 정보 혼합도를 의미한다.  $F_i$ 는  $i$  데이터 집합의 상황 요소에 대한 발생 빈도를 의미하고,  $\alpha_i$ 는  $i$  데이터 집합에서 발생한 속성의 생성 빈도를 의미한다.  $F_{i_v}$ 는  $i_v$  데이터 집합의 상황 요소에 대한 발생 빈도를 의미하

고,  $\beta_{i_v}$ 는  $i_v$  데이터 집합에서 발생한 속성의 생성 빈도를 의미한다.

상태선택 모델은 자율 캐릭터가 발생시킬 수 있는 여러 상태에서  $Entropy$ 를 기반으로  $D_i$ 의 정보 순정도를 측정하고, 확률에 기반하여 상태를 선택한다. 상태선택 모델에서 확률함수  $P$ 는 자율 캐릭터의 전이 상태를 결정한다.  $P$ 는 0에서 100까지의 값을 이용하여, 0은 정보 순정도가 가장 낮은 상태고, 100은 정보 순정도가 가장 높은 상태를 의미한다. 예를 들어 확률함수  $P$ 를 70으로 결정할 경우, 정보 순정도가 100에서 70까지 해당하는 모든 상태들은 전이대상 집합에 포함된다. 전이대상 집합이 생성된 후, 상태선택 모델은 전이대상 집합에서 랜덤하게 전이 상태를 결정하여 최종적인 전이 상태를 자율 캐릭터에 적용하는 절차를 따른다. 상태선택 모델을 자율 캐릭터의 행동 제어에 적용함으로써 자율 캐릭터는 자연스러운 다양한 행동들을 생성할 수 있다. 자율 캐릭터의 행동 결정에 확률을 사용함으로써 행동 생성에 타당성을 지닌다.

$$Selector(S) = P\left(\left(\frac{Entropy(D_1)}{F_1 + \alpha_1} - \frac{\sum_v \frac{D_{1_v}}{D_1} Entropy(D_{1_v})}{\sum_v F_{1_v} + \beta_{1_v}}\right), \left(\frac{Entropy(D_2)}{F_2 + \alpha_2} - \frac{\sum_v \frac{D_{2_v}}{D_2} Entropy(D_{2_v})}{\sum_v F_{2_v} + \beta_{2_v}}\right), \dots, \left(\frac{Entropy(D_i)}{F_i + \alpha_i} - \frac{\sum_v \frac{D_{i_v}}{D_i} Entropy(D_{i_v})}{\sum_v F_{i_v} + \beta_{i_v}}\right), \dots, \left(\frac{Entropy(D_n)}{F_n + \alpha_n} - \frac{\sum_v \frac{D_{n_v}}{D_n} Entropy(D_{n_v})}{\sum_v F_{n_v} + \beta_{n_v}}\right)\right) \quad (eq. 3)$$

## 4. 실험

### 4.1 실험 환경

슈팅게임과 같은 소규모 게임과 MMORPG와

같은 대규모 게임에서 발생하는 환경 데이터의 크기는 매우 다르며, 게임 환경의 크기는 자율 캐릭터의 행동 제어에 많은 영향을 끼친다. 이러한 게임 환경의 크기에 따라 자율 캐릭터의 행동 변화를 측정하기 위해 다양한 크기의 실험 데이터를 생성하였다. 실험 데이터 집합은 크기에 따라 D1, D2, D3, D4, D5로 분류된다. 생성된 실험 데이터에 속하는 상태 종류의 크기는 50으로 결정하였다. 게임 환경에서 발생하는 상황 요소의 크기는 10, 20, 30, 40, 50으로 결정하였으며, 상황 요소를 다양하게 정의한 이유는 상황 요소의 활용 빈도에 따른 자율 캐릭터의 행동 발생을 측정하기 위해서이다. 자율 캐릭터의 행동 데이터는 환경 상태에 따라 다양하게 생성하였으며, 실험 데이터에 포함된 자율 캐릭터의 행동 정보는 크기에 따라 D1은 100,000개, D2는 200,000개, D3는 300,000개, D4는 400,000개, D5는 500,000개로 결정하였다.

본 논문에서 제안한 기법은 다중 분할을 기반으로 하기 때문에 비교적 빠른 연산을 수행할 수 있다. 이러한 다중 분할 기법의 하나인 decision tree를 모바일 게임 환경에 사용한 연구도 존재한다[20]. 그리고 특정 행동에 대한 학습을 수행할 때는 백그라운드 작업을 사용하는 것이 일반적이기 때문에 본 실험에서 시간 요소는 배제하였다.

행동 데이터의 기본 집합은  $\{a_1, a_2, \dots, a_i, \dots, a_n\}$ 과 같으며, 행동 데이터  $a_i$ 의 속성은 불리언 형식을 취한다. 속성이 T일 경우는 해당 요소가 발생한 것이고, 속성이 F일 경우는 해당 요소가 발생하지 않은 것을 의미한다. 예를 들어 전투용 자율 캐릭터가 전투를 진행할 행동 데이터 요소에 화력, 식량, 육탄진, 지원부대가 사용된다고 가정하자. 이러한 가정에서 행동 데이터 집합이 {T, F, T, F}일 경우라면 화력과 육탄진은 사용할 수 있는 상태이고, 식량과 지원부대는 활용이 불가능한 상태이다.

자율 캐릭터의 상태 전이 벡터의 기본적 형태는  $\langle S_p | S_1, S_2, \dots, S_i, \dots, S_n \rangle$ 과 같다.  $S_p$ 는 현재 상태를 의미하고  $S_i$ 는  $i$ 번째 전이 상태를 의미한다. 상태 속성은 불리언 형식을 취하며, 속성이 T일 경우는

해당 상태로 전이가 가능하다는 것을 의미하고, 속성이 F일 경우에는 해당 상태로 전이가 불가능하다는 것을 의미한다. 예를 들어 전투용 자율 캐릭터가 사용하는 상태가 전투, 수비, 후퇴, 경계, 대기, 휴식이라 가정하자. 이러한 가정에서 전투 상태 벡터가  $\langle T | T, T, T, F, F, F \rangle$ 일 경우라면 전투 상태에서 전투 유지나 수비와 후퇴 상태로 전이가 가능하다. 이 예제에 따라 전투 상태에서 경계, 대기, 휴식 상태로의 전이는 불가능하다.

실험 데이터 생성에는 랜덤함수를 이용하였으며, 게임 환경의 복잡성을 발생시키기 위해 상태 전이에는 0%에서 30%까지의 중복도를 허용하였다. 그러므로 자율 캐릭터는 게임 환경에서 다양한 상태로 전이가 발생할 수 있다. 자율 캐릭터의 행동 선택에 다양성을 부여하고, 다양한 상태에서 자연스러운 행동 결정을 측정하기 위해 상태 전이에 대한 중복도를 허용하였다.

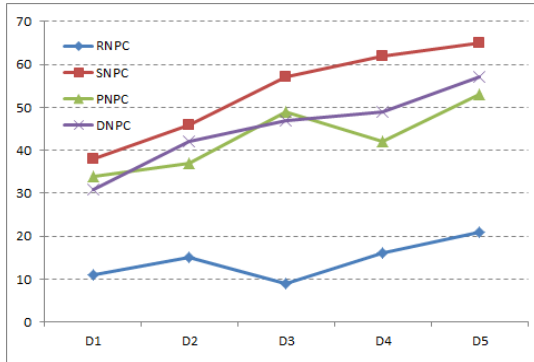
실험에 사용되는 자율 캐릭터는 크게 네 가지로 나눌 수 있으며, 첫 번째 자율 캐릭터(RNPC)에는 게임에서 일반적으로 사용하는 규칙기반 제어시스템을 적용하였다. 규칙기반은 자율 캐릭터의 행동 규칙을 정의한 후, 정의된 행동은 수정될 수 없다. 두 번째 자율 캐릭터(DNPC)는 정보획득 기법을 적용한 제어시스템으로 동작한다. 정보획득기반 자율 캐릭터는 다중 분할 기법을 이용하여 행동 결정을 생성한다. 자율 캐릭터의 다양한 행동 발생을 위해, 세 번째 자율 캐릭터(PNPC)는 상태 전이에 대한 확률함수기반 제어시스템으로 동작한다. 확률함수기반 자율 캐릭터는 각 상태에 발생확률을 적용하여 확률에 따라 상태 발생의 비율에 변화를 줄 수 있다. 네 번째 자율 캐릭터(SNPC)는 본 논문에서 제안한 상황인지 기법을 적용한 제어시스템으로 동작한다.

실험에서는 five-fold cross validation 방법을 사용하였다[21]. 정확도 계산에는 적중률을 활용하였다. 적중률 수식을 (eq. 4)에 나타냈다.

$$Hit\ ratio = \frac{total\ number\ of\ hits}{total\ number\ of\ tests} \quad (eq. 4)$$

## 4.2 실험 분석

[Fig. 1]에 본 실험의 결과를 나타내었다. [Fig. 1]의 X축에 각 실험 데이터 집합을 나타냈고, Y축은 적중률을 나타낸다.



[Fig. 1] Experimental results on five test data sets

D1은 다른 실험용 데이터에 비해 가장 게임 정보가 적은 데이터이다. D1에서 RNPC는 11%의 정확도를 나타냈고, PNPC는 34%의 정확도를 나타내어 PNPC는 RNPC보다 23% 높은 정확도를 나타냈다. PNPC는 상태 전이에 확률을 적용한 제어시스템을 활용하기 때문에, 다양한 게임 상황에서 RNPC보다 우수한 행동 제어를 나타냈다. RNPC는 규칙기반이 적용된 제어시스템을 활용하기 때문에, 예외 상황이 발생하였을 경우에는 비정상적인 행동을 발생시켰다. RNPC는 고정된 규칙을 변경할 수 없기 때문에 다양한 상황에서 적절한 행동을 추출할 수 없는 단점이 있다. 그러나 RNPC는 확률을 이용하기 때문에 다양한 상태에서 행동을 결정할 수 있는 장점이다.

D1에서 DNPC는 31%의 정확도를 나타내어 규칙기반 기법을 활용한 RNPC에 비해 20% 높은 정확도를 나타내었다. 정보획득 기법을 적용한 DNPC는 규칙기반 기법을 활용한 RNPC보다 다양한 게임 환경에 적절히 대응할 수 있다는 것을 본 실험을 통해 확인하였다. SNPC는 38%의 정확도를 나타내어 RNPC에 비해 27% 높은 정확도를 나타냈다. SNPC는 상황인지 기법이 적용된 제어

시스템을 활용하기 때문에, 새로운 게임 환경에 쉽게 적응할 수 있는 장점이 있다. SNPC는 PNPC보다 4% 높은 정확도를 나타냈고, DNPC보다는 7% 높은 정확도를 나타냈다. SNPC는 다른 제어 기법에 비해 우수하게 게임 상황에 적응한다는 것을 본 실험을 통해 확인하였다.

D2에서 RNPC는 15%의 정확도를 나타내어 D1에서의 성능보다 4% 향상되었다. RNPC는 규칙기반 제어시스템을 활용하기 때문에, 자율 캐릭터에 행동 규칙이 많이 포함될수록 우수한 성능을 나타낸다는 특성이 있다. D2에서 PNPC는 37%의 정확도를 나타내어 RNPC보다 22% 우수한 성능을 나타냈다. PNPC도 사용할 수 있는 게임 데이터가 많을 경우, 행동 제어 성능이 우수하게 나타난다는 특성이 있다. 그러나 규칙기반 방식의 RNPC와 다르게 PNPC는 정형화된 규칙을 활용하는 것이 아니라, 게임 상황의 요소를 분석하여 자동적으로 자율 캐릭터의 행동을 생성한다. 이와 같은 이유로 PNPC는 RNPC보다 다양한 행동을 생성할 수 있는 장점이 있다. DNPC는 42%의 정확도를 나타내어 RNPC에 비해 27% 높은 성능을 나타냈으며, PNPC보다는 5% 높은 성능을 나타냈다. SNPC는 D2에서 46%의 정확도를 나타내어 RNPC에 비해 31% 높은 성능을 나타냈으며, PNPC와 DNPC보다 각각 9%와 4% 높은 성능을 나타냈다. SNPC는 게임 상황을 스스로 인지하여 적절한 행동을 발생시키기 때문에, 게임 데이터가 증가할수록 더욱 우수한 성능을 나타낸다.

D3에서 RNPC는 9%의 정확도를 나타냈고, PNPC는 49%의 정확도를 나타내어 PNPC는 RNPC보다 40% 높은 성능을 나타냈다. DNPC는 47%의 정확도를 나타내어 RNPC에 비해 38% 높은 성능을 나타냈고, PNPC보다는 2% 낮은 성능을 나타냈다. SNPC는 57%의 정확도를 나타내어 RNPC에 비해 48% 높은 성능을 나타냈고, PNPC와 DNPC보다 각각 8%와 10% 높은 성능을 나타냈다. 데이터 집합의 규모가 커져도 SNPC는 다른 자율 캐릭터보다 높은 성능을 나타냈다.

D4에서 RNPC는 16%의 정확도를 나타내어 D3에서의 성능보다는 7% 향상되었다. RNPC도 데이터의 증가에 따라 성능은 향상되고 있으나, 높은 성능을 나타내지 못하는 이유는 게임 상황이 커짐에 따라 예외적인 규칙 발생도 증가하기 때문이다. D4에서 PNPC는 42%의 정확도를 나타내어 RNPC보다 26% 우수한 성능을 나타냈으며, PNPC는 게임 규모가 커지더라도 우수한 성능을 유지하였다. DNPC는 49%의 정확도를 나타내어 RNPC에 비해 33% 높은 성능을 나타냈고, PNPC보다 7% 높은 성능을 나타냈다. SNPC는 62%의 정확도를 나타내어 RNPC보다 46% 높은 성능을 나타냈고, PNPC와 DNPC보다 각각 20%와 13% 높은 성능을 나타냈다.

D5는 실험 데이터 집합에서 가장 큰 규모이다. D5에서 RNPC는 21%의 정확도를 나타냈고, PNPC는 53%의 정확도를 나타내어 RNPC보다 32% 높은 성능을 나타냈다. 게임 규모의 증가에 따라 활용할 수 있는 상황 정보도 증가하기 때문에, PNPC는 게임 규모의 증가에 따라 RNPC보다 우수한 성능을 나타냈다. DNPC는 57%의 정확도를 나타내어 RNPC보다 36% 높은 성능을 나타냈고, SNPC는 65%의 정확도를 나타내어 RNPC보다 44% 높은 성능을 나타냈다.

## 5. 결 론

자율 캐릭터의 행동 제어에 활용되는 대표적인 방식인 규칙기반 방식은 규칙의 생성과 갱신이 편리하지만, 자연스러운 행동 생성에 제한적이고 예외 규칙이 발생하였을 경우에는 비정상적인 행동을 유발한다. 이러한 문제점들을 해결하기 위해 본 논문에서 상황인지 기법을 제안한다. 상황인지 기법은 게임 환경에서 발생하는 환경 요소를 활용하여 자율 캐릭터의 행동을 제어한다. 상황인지 기법은 게임 환경에 따른 자율 캐릭터의 상태 결정을 위해 상태선택 모델을 사용하며, 상태선택 모델은 다

양한 상태 전이에 대해 현실성을 부여하는 장점이 있다. 상태선택 모델에서는 자율 캐릭터의 현실감 유지를 위해, 상태 전이에 대한 정보 가치를 분석하여 불완전한 상태 전이를 배제하는 특성이 있다.

다양한 실험에서 본 논문에서 제안한 기법은 우수한 성능을 나타냈고, 확률기반 기법과 정보획득 기법도 규칙기반 기법보다 우수한 성능을 나타냈다. 자율 캐릭터의 행동 제어에 게임 상황에서 발생하는 환경 요소를 활용하는 것이 자율 캐릭터의 행동 생성에 효과적이라는 것을 여러 실험을 통해 확인하였다.

## ACKNOWLEDGMENTS

This work was supported by the research fund of Korea Nazarene University.

## REFERENCES

- [1] P. Khanal, "Design and development of a serious game for central line placement", Proc. of the 26th IEEE International Symposium on CBMS, pp.530-531, 2013.
- [2] S. Lee and M. Kwon, "Classifying Digital Game Genres", Jour. of Korea Game Society, Vol.8, No.3, pp.3-14, 2008.
- [3] S. Kurkovsky, "Work in progress: Evaluating the use of mobile game development in introductory CS courses", Proc. of the IEEE-FIE International Conference, pp.1-2, 2012.
- [4] R. Koh, "Co-creativity fusions in interdisciplinary augmented reality game developments", Proc. of the IEEE International Symposium on ISMAR-AMH, pp.47-56, 2012.
- [5] K. Noh, T. Lee, and S. Cho, "A Formal Study on Game Character Preference through Game User Classification", Jour. of Korea Game Society, Vol.7, No.4, pp.23-32, 2007.



- [6] A. Tychsen, M. Hitchens, and T. Brolund, "Theoretical and Practical Computer Applications in Entertainment", Proc. of the ACM International Conference on IEC, pp.28-34, 2008.
- [7] W. Tang and T. R. Wanl, "Intelligent Self-learning Characters for Computer Games", Proc. of the EGUK International Conference, pp.51, 2002.
- [8] Y. Mishima, K. Fukuda, and H. Esaki, "An Analysis of Players and Bots Behaviors in MMORPG", Proc. of the 27th IEEE International Conference on AINA, Vol.3, No.1, pp.18-21, 2006.
- [9] J. Oh, "Automatic Detection of Compromised Accounts in MMORPGs", Proc. of the International Conference on Social Informatics, 2012.
- [10] J. Kawale, A. Pal, and J. Srivastava, "Churn Prediction in MMORPGs: A Social Influence Based Approach", Proc. of the International Conference on Computational Science and Engineering, 2009.
- [11] T. Bosse and E. Zwanenburg, "Do Prospect-Based Emotions Enhance Believability of Game Characters? A Case Study in the Context of a Dice Game", IEEE Transactions on Affective Computing, Vol.6, No.1, pp.21-29, 2013.
- [12] A. Petrenko, "Testing from Partial Deterministic FSM Specifications", IEEE Transactions on Computers, Vol.54, No.9, pp.1154-1165, 2005.
- [13] M. Borowczak and R. Vemuri, "S\*FSM: A Paradigm Shift for Attack Resistant FSM Designs and Encodings", Proc. of the IEEE International Conference on BioMedCom, 2012.
- [14] B. M. Namee and P. Cunningham, "A Proposal for an Agent Architecture for Proactive Persistent Non Player Characters", Proc. of the 12th IRISH International Conference on AI and Cognitive Science, 2001.
- [15] G. Robert, P. Portier, and A. Guillot, "Classifier Systems As 'ANIMAT' Architectures For Action Selection In MMORPG", Proc. of the 3th International Conference on Intelligent Games and Simulation, 2002.
- [16] M. Mateas and A. Stern, "A Behavior language for story-based believable agents", IEEE Transactions on Intelligent Systems, Vol. 17, No. 4, 2002.
- [17] M. Delap, "Is runtime verification applicable to cheat detection?", Proc. of the ACM SIGCOMM International Conference on NGN, 2004.
- [18] Y. Feng, T. Teng, and A. Tan, "Modelling situation awareness for Context-aware Decision Support", Expert Systems with Applications Vol.36, pp.455-463, 2009.
- [19] B. Lee, C. Park, J. Kim, S. Youk, and K. Ryu, "An Intelligent NPC Framework for Context Awareness in MMORPG", Proc. of the International Conference on Convergence and Hybrid Information Technology, pp.190-195, 2008.
- [20] H. Kim, D. Shin, D. Shin, S. Kim, and M. Lee, "Design of Physics Engine based on Context-Awareness for Mobile 3D Racing Game", Proc. of Korean Society for Internet Information, pp.121-125, 2010.
- [21] R. Tibshirani and G. Walther, "Cluster Validation by Prediction Strength", Jour. of Computational and Graphical Statistics, Vol.14, No.3, pp.511-528, 2005.



김형일(Kim, Hyung Il)

1996-1998 (주)경기은행  
2004 동국대학교 컴퓨터공학과(공학박사)  
2005-2006 동국대학교 컴퓨터공학과 IT교수(정보통신부)  
2007-현재 나사렛대학교 멀티미디어학과 교수

관심분야 : 인공지능, 기계학습, 게임지능, 정보검색,  
데이터마이닝, 임베디드시스템, 가상교육

— 자율 캐릭터를 위한 상태기반 상황인지 기법 —