

3차원 게임에서 객체들의 상호 작용을 디자인하기 위한 제어 기법

(A Control Method for designing Object Interactions
in 3D Game)

김 기 현 [†] 김 상 욱 ^{**}
(Ki Hyun Kim) (Sang Wook Kim)

요 약 3차원 게임은 게임 시나리오의 다양한 요소에 의한 복잡도가 증가함에 따라 게임 객체들의 상호 관계를 제어하기 위한 문제점을 가진다. 그러므로, 게임 시스템은 각 게임 객체들의 응답을 조정하는 방법의 필요성을 가진다. 또한, 게임 시나리오의 결과에 따라 게임 객체들의 행동 애니메이션을 제어하기 위한 개념들도 필요하다. 사실적 게임 시뮬레이션을 생성하기 위해 시스템은 게임 객체들의 상호작용 설계를 디자인 할 수 있는 구조를 포함해야 한다. 본 논문에서 게임 시나리오상에 게임 객체들의 상호작용 설계를 위해 동적 제어를 디자인하는 기법을 소개한다. 이 방법을 위해 특정 규칙을 이용한 의사결정이 가능한 지능적 에이전트 기반 구조로써 게임 에이전트 시스템을 제안한다. 게임 에이전트 시스템은 환경 데이터 처리, 게임 객체 시뮬레이션, 게임 객체들간의 상호작용 제어, 게임 객체들의 다양한 상호 관계를 정의할수 있는 시각 저작 인터페이스를 제공하기 위해 이용되어진다. 이들 기술들은 게임 객체의 자율성과 연관된 충돌 회피 기법 등을 처리한다. 또한, 장면의 변경으로부터 게임 객체들의 일관된 의사 결정력을 가능하게 한다. 본 논문에서는 규칙기반 행동 제어가 게임 객체의 시뮬레이션을 안내하기 위해 디자인되어졌다. 시각적 요소들로 구성된 에이전트 상태 결정 네트워크는 정보전달과 게임 객체들 사이의 현상태를 추론할 수 있다. 이들 기법들은 실시간으로 게임 객체들간의 동작 상태 변화를 체크하고 모니터링 할 수 있다. 마지막으로 간단한 사례 연구 예와 함께 제어 기법의 타당성을 제시한다.

키워드 : 게임 객체, 에이전트 시스템, 메시지, 3차원 게임, 제어기법, 상호작용, 대화 규칙, 지식 베이스, 게임 시나리오

Abstract As the complexity of a 3D game is increased by various factors of the game scenario, it has a problem for controlling the interrelation of the game objects. Therefore, a game system has a necessity of the coordination of the responses of the game objects. Also, it is necessary to control the behaviors of animations of the game objects in terms of the game scenario. To produce realistic game simulations, a system has to include a structure for designing the interactions among the game objects. This paper presents a method that designs the dynamic control mechanism for the interaction of the game objects in the game scenario. For the method, we suggest a game agent system as a framework that is based on intelligent agents who can make decisions using specific rules. Game agent systems are used in order to manage environment data, to simulate the game objects, to control interactions among game objects, and to support visual authoring interface that can define a various interrelations of the game objects. These techniques can process the autonomy level of the game objects and the associated collision avoidance method, etc. Also, it is possible to make the coherent decision-making ability of the game objects about a change of the scene. In this paper, the rule-based behavior control was designed to guide the simulation of the game objects. The rules are pre-defined by the user using visual interface for designing their interaction. The Agent State Decision Network,

[†] 학생회원 : 경북대학교 컴퓨터학과
gimkh@woorisol.knu.ac.kr
^{**} 정 회 원 : 경북대학교 컴퓨터학과 교수
swkim@cs.knu.ac.kr

논문접수 : 2002년 10월 2일
심사완료 : 2003년 2월 4일

which is composed of the visual elements, is able to pass the information and infers the current state of the game objects. All of such methods can monitor and check a variation of motion state between game objects in real time. Finally, we present a validation of the control method together with a simple case-study example.

In this paper, we design and implement the supervised classification systems for high resolution satellite images. The systems support various interfaces and statistical data of training samples so that we can select the most effective training data. In addition, the efficient extension of new classification algorithms and satellite image formats are applied easily through the modularized systems. The classifiers are considered the characteristics of spectral bands from the selected training data. They provide various supervised classification algorithms which include Parallelepiped, Minimum distance, Mahalanobis distance, Maximum likelihood and Fuzzy theory. We used IKONOS images for the input and verified the systems for the classification of high resolution satellite images.

Key words : Game Object, Agent System, Message, 3D Game, Control Method, Interaction, Conversation Rule, Knowledge Base, Game Scenario

1. 서론

3차원 게임 환경 내에는 여러 게임 객체들 즉, 동적인 객체(사람, 동물 캐릭터 등)와 정적인 객체(자연 현상, 건물 등)가 존재한다. 게임 객체들간에는 게임 시나리오의 전개상 일정한 논리적인 규칙과 데이터로 이루어진다. 논리적인 규칙에는 게임 객체들간의 동적인 의사 소통 규칙들과 알고리즘들(충돌 탐지, 회피처리, 경로탐색 등)과 같은 다양한 요소들이 포함되고, 데이터에는 각 게임 객체들 서로간의 반응에 따른 지능적 처리와 행동 방식(캐릭터의 동작 애니메이션, 자연 현상 애니메이션 등)을 제어하기 위한 세부적인 사항들이 포함된다. 게임 진행상 규칙이 변경되거나 주변 게임 객체의 상태 변경으로 인한 게임 시나리오 상 게임 디자이너가 원하지 않는 상황이 발생할 수 있다. 게임에서는 스토리 전달의 문제점이 발생할 수 있으므로 게임의 외부와 내부 모듈에서 상호작용인 면을 유동적으로 처리할 수 있어야 한다. 그러므로, 논리적인 규칙과 데이터가 이러한 변화를 반영 처리 할 수 있는 조정 기법이 필요하다. 특히, 3차원 게임 환경과 같은 복잡한 상황에서 다양한 변화에 적응하고 대처할 수 있는 수많은 문제들을 해결하기 위한 방안이 필요하다. 가상 캐릭터와 정적 객체간의 상호 작용 기술은 처음 OSR 모듈을 이용한 자연어 명령 기반 시뮬레이션에 언급되었다[1]. 그러나, 주로 작업을 이해하는 측면만 고려하고 복잡한 여러 환경에서의 객체들간의 조정, 상호 작용 측면은 고려하지 않는다. 지금까지의 주된 관심은 사용자와 주변 환경 객체간의 직접적 상호작용이었다[2-4]. 이것은 적절한 행동을 가지도록 객체를 고려하지는 않는다. 어떤 시스템은 객체의 기능성만을 강조할 뿐 사용자 객체와 정적 객체의 상호작용은 고려하지 않는다[5]. 상호 관계 디자인 과정을 도

시 환경에 대한 시뮬레이션에 적용한 경우 주변 가상 환경 객체에 제약조건을 부여하였다. 이 시스템에서 가상 인체 모델 캐릭터와 객체와의 상호 작용 제어 방법을 제시하였다[6]. 가상 객체의 동작 정보를 정의하고 가상 인체 모델 객체에 상호 작용 능력을 부여한 기법[7]의 경우 스크립트 기반 동작 오토마타와 시각적 상태 머신을 제안하였다. 그러나, 순차적인 행동의 결과에 따른 판단처리만이 설정되어 있고 3차원 게임 환경과 같은 유동적인 환경에서 사용자 객체나 다른 객체의 추론적, 인지적 행동 처리는 부족한 면을 가지고 있다. 캐릭터 상호간의 제어보다는 동작 생성의 제약 조건에 관심을 두는 경우 서로간의 대화 규칙을 통한 제어 관계가 결여되어 있다[8]. 특히, 3차원 게임의 경우 전체 게임 시나리오에 따른 게임 전개시 게임 객체간 많은 상호작용 등이 요구된다. 이러한 상호작용 관계가 올바르게 없을 때 잘못된 애니메이션을 생성할 수 있다. 그러므로, 3차원 게임 환경과 같이 애니메이션의 장면이 유동적으로 변하는 상황에서는 게임 캐릭터들뿐만 아니라, 주변의 정적 객체들 사이에 자신의 개별 동작과 상호작용을 제어하기 위한 조정이 필요하다. 또한, 게임 캐릭터 객체 자신이 정한 목표를 달성하기 위한 내부적인 인지적인 능력과 신뢰성 있고 일관성 있게 다른 가상 객체와 반응할 수 있는 요소들이 필요하다. 지능적, 자율성의 속성을 제공하는 에이전트 기법[9-12]은 각 게임 캐릭터의 동작 상태 변동에 따른 자원을 관리하기 위한 구조적인 방법을 제시할 수 있고, 동적인 환경 변화에 대응할 수 있는 방안을 제공할 수 있다. 가상 인체 모델 에이전트 시뮬레이션을 위한 플랫폼을 제안한 ACE는 특징적 모델링 접근법을 이용하여 에이전트와 객체간의 상호작용을 디자인하고 스크립트된 모듈을 통해 각 에이전트의

추론 및 행동을 제어하였다[13]. 그러나, 인체 모델 캐릭터의 세밀한 동작과 상호 작용 관계만을 디자인하여 전체적인 상호 작용 관계 디자인 모듈이 부족하다.

본 논문은 3차원 게임 환경에서 사용자를 대신하는 게임 캐릭터뿐만 아니라, 다른 게임 캐릭터나 정적인 사물 객체와 의사소통을 위해 에이전트 개념을 적용하였다. 이를 기반으로 시스템을 설계하여 전체 게임 시나리오 목적에 맞는 애니메이션을 생성하도록 제어하고 디자인 할 수 있는 환경을 제공한다. 또한, 시각적 그래픽 인터페이스를 이용하여 각 게임 객체들의 현 장면 시나리오 목적에 맞는 상태를 설계하고 실시간으로 모니터링 할 수 있는 방법을 제안한다. 이러한 기법은 게임 시나리오의 목적에 맞지 않는 상황과 오류 상태를 체크 할 수 있다.

본 논문의 제 2절에서는 게임 객체간 조정을 위한 에이전트 특징과 전체 설계 디자인 개념에 대해서 정의한다. 제 3절에서는 게임 객체간 의사전달을 위한 디자인 특징과 행동 설계를 위한 시각적 인터페이스를 제공하는 도구와 분석과정에 대해서 설명한다. 제 4절에서는 게임 객체간 상호 작용 설계 기법과 에이전트 개념을 바탕으로 한 간단한 게임 시나리오 구현 예를 통해 게임 객체간 상호 작용 관계를 모니터링하는 과정을 보인다. 마지막으로 결과 및 향후 방향을 제 5절에서 제시한다.

2. 게임 객체간 조정과 설계 디자인 개념

본 절에서는 게임 객체간 조정을 위한 에이전트 기법에 대해서 설명하고, 전체 시스템 구조 디자인에 대해서 언급한다. 여기서 에이전트는 스스로 게임 객체간 상호 작용 결과에 따른 인지, 계획, 추론, 학습 능력을 가지는 모듈을 구성한다. 주변 게임 객체와의 반응과 지능적 행동들은 에이전트 속성에 따라 계층적 형태로 분해된다.

2.1 게임 객체의 행동 제어 모듈

게임 객체는 에이전트 형태로 인지적 능력을 기반으로 한 행동 단위 속성을 가지고 있다[14]. 에이전트 기술은 상호작용을 위한 지능적인 가상 객체를 형성하는 컴퓨터 그래픽, 가상현실, 3차원 게임과 같은 곳에서 많이 사용된다. 에이전트의 지식은 다른 객체와의 모든 가능한 상호작용을 풀기위해 사용된다. 그러므로, 이러한 개념을 게임 객체간의 행동과 상호 작용 관계를 디자인하는 설계 개념에 적용하여 3차원 게임 환경과 같은 다양한 장면 시나리오에 따른 동작 정의와 게임 객체간 조정을 가능하게 할 수 있다. 각 게임 에이전트 객체는 자신의 게임 목표를 달성하기 위한 계획된 정의 모듈을 가지고 있다. 각 에이전트의 행동에 대한 동작 정보는

동작 모듈에서 호출 및 지정한다.

2.1.1 에이전트 객체의 속성과 종류

에이전트는 쓰레드로 디자인 되어 실시간으로 각 에이전트의 정보를 해석 처리한다.

(1) 에이전트 종류

- 동적 에이전트 객체: 사람, 동물 등 움직임과 인지적 능력을 가진 객체
- 정적 에이전트 객체: 나무, 건물, 책상 등의 움직임이 부여되지 않은 객체

본 논문에서는 캐릭터 에이전트 객체라는 용어를 동적 에이전트 객체와 동일한 의미로 사용하고 사물 에이전트 객체는 정적 에이전트 객체와 동일한 의미로 사용한다. 정적 에이전트 객체는 상황에 따라 인지적인 능력이 부여된다. 이것은 다른 동적인 에이전트 객체에 대한 인식과 사용자의 입력에 따른 판단 추론을 정적인 에이전트 객체에게 부여하여 각 에이전트 객체가 가진 정보를 중앙 집중화 하지 않고 분산 처리하여 효율적인 추론 형태를 가능하게 한다.

(2) 에이전트 속성

- 기본 객체 정보들 : 크기, 움직임, 위치 등
 - 상호작용 정보 : 다른 객체와의 상호작용을 위한 상태 정보
 - 반응 정보 : 상호작용 결과에 따른 에이전트 객체의 반응행동을 나타냄. 상황에 따라 동작 정보가 달라짐
- 속성 정보는 지식 베이스 형태로 정의되어 있다. 이러한 정보를 바탕으로 상호 작용 관계를 해석하기 위한 판단 모듈을 에이전트가 포함하고 있다.

2.1.2 에이전트 구조 형태

에이전트는 계층 구조 형태의 속성을 지닌다. 그림 1은 에이전트가 가지는 내부 모듈의 구조와 특징을 나타

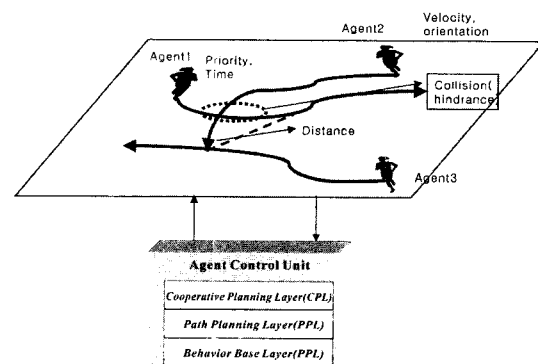


그림 1 에이전트 설계 구조

낸다.

에이전트 제어 유닛은 각 에이전트가 3차원 게임 환경에서 가져야 할 요소를 포함하고 있다. 경로 탐색, 다른 에이전트와 상호작용을 위한 처리, 충돌 탐지, 동작 형태 등을 해석하여 게임 시나리오의 요구 상황에 맞는 정보를 분석하여 실시간으로 게임 객체를 제어하고 실행한다.

2.1.3 의사 전달을 위한 매개체

각 게임 에이전트 객체는 메시지를 이용해 다른 게임 캐릭터 에이전트 객체나 정적 에이전트 객체에게 의사 전달을 할 수 있다. 즉, 자신의 상태 정보를 전달 하기 위해 서로 통신한다. 메시지는 시나리오 상황에 따라 달라질 수 있다. 메시지에 대한 정보는 테이블 형태로 저장된다.

- 메시지의 기본 형태

sender Si, receiver Ri, message_content(N1,... Nn), Timer 등

에이전트간 의사소통을 위한 메시지 속에는 동작 상태 타입과 객체이름, 위치, 속도 등과 같은 정보들이 포함될 수 있다. 메시지는 복잡한 정보를 캡슐화하여 전달하고자 하는 에이전트에게 효율적인 정보를 제공한다. 메시지들은 사용자의 입력과 게임 시나리오 장면 전환 시에 발생한다. 그림 2는 메시지의 전달 처리 구조를 나타낸다. 어떠한 메시지가 발송되면 그 메시지는 항상 메시지 라우터로 전달된다. 라우터는 미리 정해진 형태로 준비된 메시지를 받아서 그 메시지가 의도하는 곳으로 전달하는 역할을 한다. 또한, 라우터는 요청에 따라 상태들을 변경하는 일도 처리한다. 메시지를 에이전트 상태 결정 네트워크(ASDN)상에 구성된 구조를 바탕으로 해당 게임 객체에게 전달한다.

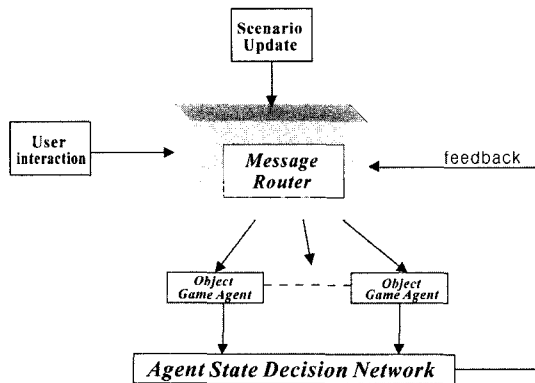


그림 2 메시지 전달 구조

2.2 시스템 구조와 디자인

그림 3은 에이전트를 기반으로 3차원 게임 환경에서 게임 객체간 상호작용 관계성을 디자인하기 위한 전체 시스템 구조를 나타낸다.

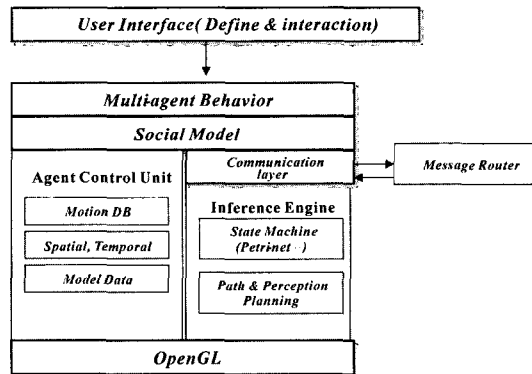


그림 3 3차원 게임 에이전트 시스템(GAS)

3차원 게임 에이전트 시스템(GAS)은 객체 추론 모듈을 가지고 있어 각 에이전트 객체의 행동 계획에 대한 상태정보를 트리 구조 형태로 저장, 탐색하여 정보를 추출한다. 또한, 각 에이전트가 가지는 계획된 일을 더 작은 개체 단위로 분할한다. 이것은 각 상태별 행동 동작들을 더 세분화된 동작으로 구성하고 해석한다. 3차원 게임 에이전트 시스템은 멀티 에이전트 시스템 구조로 디자인되어 정보가 하나의 객체에만 중앙 집중처리 되는 제약성을 완화한다. 게임 에이전트 객체간 순차적인 제어, 동시성 처리 등의 작업 환경을 제공한다

3. 시각적 상태 디자인 도구

본 절에서는 게임 에이전트 객체간의 상호 관계를 디자인 처리하기 위한 시각적 그래픽 인터페이스 환경에 대해서 설명한다. 3차원 게임 환경에는 각 에이전트들의 상호 작용 결과에 따른 동적 변수가 많으므로 게임 사용자의 의견을 반영할 수 있는 효율적 디자인 도구 개발이 매우 중요한 문제가 될 수 있다. 3차원 에이전트 객체의 동작 변이나 상호작용을 기술하는 방안으로 상태 머신을 주로 이용한다. 그래픽 인터페이스를 이용한 상태 머신 디자인은 공통된 접근법이고 많은 시각적 프로그래밍 기술에 사용되어 왔다[15-17]. ACE에선 상호 작용 관계 설정을 위해 텍스트 명령어와 상태 연결을 위한 그래프를 사용하였다[13]. 본 논문에선 개별 게임 에이전트 객체에 대한 행동정의와 상호관계 디자인을

확장한 멀티 에이전트 기법을 사용한 다중 게임 에이전트 객체의 상호 작용 관계와 동기화, 병행적 처리 등을 위한 시각적 기법을 사용하였다. 이를 바탕으로 디자인된 상태 정보는 에이전트와 서로 협력하는 네트워크 형태로써 조직되어진다. 간단한 게임 시나리오를 바탕으로 에이전트 객체간 상호 작용 조정 기법과 해석 방법의 예를 제시한다.

3.1 에이전트 객체간 의사전달 특징

본 논문에서는 에이전트 개념을 적용한 게임 환경에 맞는 높은 수준의 상태 적용 원리들을 설계하여 유연성과 사용의 용이함을 추가하였다. 상호 작용 계획은 각 에이전트 객체와 시나리오의 특성에 따라 적절히 디자인되고 추론되어진다. 에이전트 객체 각각은 다양한 목표를 지니고 있다. 게임 장면 시나리오 전개에 따른 상호 작용 관계성을 디자인하는 에이전트 판단 상태 디자인 도구는 동시성, 분산, 병렬 처리, 비동기적인 특징을 가질 수 있는 상황을 디자인 하기 위한 그래픽적 요소를 가지고 있는 모델링 도구이다. 이러한 기법은 복잡한 시스템 모델링을 가능하게 한다. 3차원 게임 환경과 같은 복잡한 상황에서 장면전환에 따른 가상 객체의 상황 판단과 다른 에이전트 객체간의 의사 소통, 병렬적 처리 문제를 체크하고 조정 해결할 수 있는 방법을 제공한다. 에이전트 객체 상황에 대한 네트워크의 일반적 형태는 다음과 같다. 본 논문에서 이러한 상태 디자인 구조를 에이전트 상태 결정 네트워크(ASDN)라 명명한다.

에이전트 상태 결정 네트워크는 다음과 같이 4개 튜플 ASDN = (P, T, F, W)로 구성된다.

$P = \{p_1, p_2, \dots, p_m\}$: 플레이스들(Places)의 유한 집합
 $T = \{t_1, t_2, \dots, t_n\}$: 트랜지션들(transitions)의 유한 집합

$F \subseteq (P \times T) \cup (T \times P)$: 아크들(arcs)의 집합 (네트워크 관계)

$W: F \rightarrow \{1, 2, 3, \dots\}$: 가중치 함수

P는 에이전트 객체인 게임 객체의 현 장면 시나리오에서의 상태를 의미한다. T는 메시지 정보에 따른 현 시나리오 상황을 판단하여 다음 상태로 전개하기 위한 조건 규칙들을 의미한다. W는 에이전트 객체간의 우선 순위를 설정하기 위한 가중치 함수를 나타낸다. 게임 실행시 각 게임 에이전트 객체의 현 상태를 체크하기 위해 플레이스들이 변경 될때마다 상황을 판단할 수 있도록 플레이스의 색상이 변경된다. 이를 통해 실시간 게임 상황에서 발생할 수 있는 제어의 오류를 체크하고 모니터링 할 수 있다. 그림 5는 게임 장면 시나리오를 기반으로 하여 각 게임 에이전트 객체간 상태 결정 네

트워크 구조 작성을 위한 도구 화면과 초기 작성 결과를 나타내는 예로 3.2에서 설계 방법과 간단한 예를 통해 설명한다. 에이전트 상태 결정 네트워크 구조의 간단한 예를 나타내는 그림 5의 2번에서 원은 그림 5의 3번 버튼 도구에 있는 두 번째 아이콘을 선택하면 나타나는 것으로 플레이스를 의미한다. 그림 5의 3번 버튼 도구에 세 번째 버튼 아이콘은 트랜지션을 의미한다. 트랜지션 아이콘을 선택하면 상태 전환을 위한 조건 및 명령어들을 설계하기 위한 다이얼로그 상자가 나타난다. 동작 제어를 위한 기법은 주로 규칙 기반 방법을 사용해 왔다. 규칙은 자연어 형태로 정의되어 시뮬레이션 동안 동작 애니메이션을 제어하기 위해 해석되어 진다[6]. 본 논문에서는 스크립트 언어를 기반으로한 에이전트 객체간 동작 제어를 위해 메시지 형태에 따른 조건을 부여하여 사용자가 의도하는 동작이 전환되도록 설정한다. 사용자는 원하는 조건 규칙과 메시지 정보를 설계하여 에이전트 객체간 분기 명령어를 선택 지정할 수 있다. 그림 4는 트랜지션 아이콘을 선택후 나타나는 스크립트 디자인 설계화면의 예로 사용자는 시스템에서 제공하는 스크립트 언어 형태와 메시지 정보를 참조하여 각 에이전트 객체에 대한 제어 형태를 디자인한다. 디자인된 스크립트는 파싱되어 게임 실행 환경에서 실시간 해석 처리 되어진다.

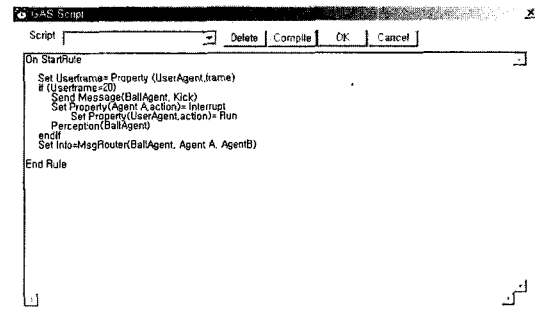


그림 4 게임 스크립트 디자인 설계

아크는 화살표를 의미하는 것으로 각 상태간의 연결과 네트워크 관계를 나타낸다.

3.2 게임 객체간 상호 작용 설계 디자인 기법과 해석 방법

그림 5는 게임 에이전트 객체의 동작 애니메이션 제어를 위해 상호 작용 관계를 설계하는 인터페이스 환경을 나타내는 것으로 장면 상황 판단 모델의 시각적 구조 형태와 디자인 도구를 나타낸다.

에이전트는 서로간에 대화를 통해 자신의 상황을 판단한다. 대화란 어떤 목표를 달성하기 위한 에이전트의 계획이다. 이를 통해 다른 에이전트의 상호작용에 기반한 모든 가능한 대화를 식별한다. 각 에이전트가 가지는 목표에 따라 조정된다. 또한, 각 에이전트들의 현재 메시지 정보 형태에 따라 상태 전이가 변경된다. 본 논문에선 이러한 대화의 속성을 통한 게임 객체간 상호 작용 관계를 유한상태모델처럼 표현하고 장면 전개에 따른 판단 규칙에 따라 상태의 전이를 일으킬 수 있도록 디자인하는 저작 환경을 제공한다. 에이전트 모듈에서 이러한 각 규칙에 대한 여러 실행 조건과 행동을 식별한다.

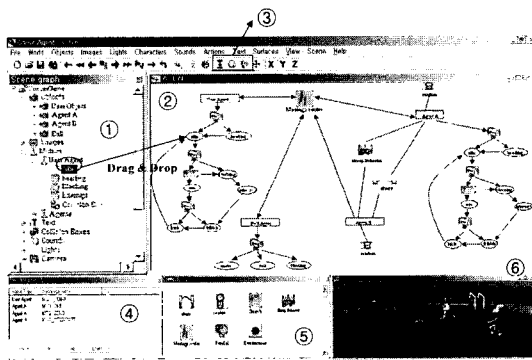


그림 5 시각적 저작 도구 인터페이스

다음은 인터페이스 상에 나타난 각 도구들의 기능과 특징에 대해서 설명한다.

1번은 장면 그래프를 나타내는 것으로 게임 장면의 데이터 구조와 정보를 계층적인 형태로 나타낸다. 게임 객체의 모델 데이터 정보와 속성을 나타낸다. 각 에이전트 객체에 대한 동작 정보도 나열되어 있다. 사용자는 각 게임 객체의 움직이는 동작 데이터에 관한 정보를 마우스를 이용하여 선택 후 저작 영역에 지정할 수 있다. 화면 상단의 메뉴는 게임 실행 처리 관련 기능과 에이전트 객체 모델에 대한 속성 지정, 에이전트 상태 결정 네트워크의 해석에 관한 설정 등이 포함되어 있다. 6번은 여러 에이전트 객체가 파일 형태로 로드되어 3차원 게임 환경 상에 구성된 모습을 나타내고 있다. 이와 같은 3차원 게임 환경에서 사용자는 게임 시나리오에 따른 게임을 실시간으로 실행할 수 있다. 2번은 에이전트 상태 결정 네트워크를 디자인하는 저작 환경을 나타낸다. 마우스를 이용하여 드래그 앤 드롭 형태로 게임 시나리오에 맞는 네트워크 구조를 디자인한다. 여기서

디자인된 정보가 파싱 처리되어 각 에이전트가 장면 각각의 시나리오에 맞는 정보 처리를 가능하도록 구조적인 형태로 해석된다. 3번의 버튼에는 상태 디자인을 위한 도구들을 나타낸다. 플레이스들에 해당하는 동작 정보를 1번에서 마우스를 이용하여 선택후 저작 환경에 디자인한다. 플레이스들은 원 모양의 형태로 나타난다. 3번의 버튼에서 첫 번째 아이콘은 각 플레이스들의 네트워크 관계를 설정하기 위한 화살표 기호를 나타낸다. 5번은 2번에서 디자인 할 내용 중 각 에이전트간 동기화, 병렬처리, 랜덤 속성, 공유 설정, 우선 순위 제어 등을 나타내는 아이콘들의 종류를 나열하여 에이전트간 부가적 기능 제어를 선택처리 할 수 있도록 구성되어 있다. 4번은 각 에이전트간 전달되는 메시지 종류와 기능을 나타낸다.

본 논문에선 에이전트 객체간의 상호작용관계와 장면 상황 모니터링의 예로써 3명의 가상 캐릭터가 미니 축구 게임을 한다는 게임 시나리오를 설정하였다.

3.2.1 미니 축구에서의 게임 에이전트 객체 종류와 속성들

(1) 사용자 캐릭터 에이전트

- 사용자의 상호 작용 처리
- 다른 캐릭터 에이전트의 행동 판단 처리
- 다른 캐릭터 에이전트와 주변 환경에서의 충돌여부 판단
- 정적 에이전트 객체에게 의사 전달

(2) 다른 캐릭터 에이전트

- 랜덤하게 상태 변경 및 이동처리
- 다른 캐릭터 에이전트간의 충돌 여부 판단
- 사용자 캐릭터 에이전트에 대한 방어
- 같은 팀에 대한 정보 처리

(3) 공 에이전트 객체

- 공 이동 방향에 대한 각도와 거리 체크
- 랜덤하게 이동, 다른 캐릭터의 목표에 따른 이동 처리

(4) 각 에이전트 객체의 공통적 속성

거리, 시간, 속도, 충돌 여부 판단, 동작 타입 정보 등 각 에이전트 객체는 공통적인 정보 처리 능력을 가진다. 각 에이전트 객체에게 자신들의 정보를 실시간으로 전달한다.

3.2.2 상태 디자인과 대화 모델 분석

3.2.1의 미니 축구 게임 시나리오에 따라 그림 5의 1번 객체 데이터 정보와 동작 정보에 관한 것을 나타내는 구조에서 에이전트 객체를 마우스를 이용하여 선택 후 이동하여 저작 환경에 설계 한다.

그림 6은 에이전트 상태 결정 네트워크(ASDN)를 디자인한 예를 나타내는 것으로 게임 환경에서 게임 객체간의 의사소통 처리에 따른 상황 판단의 그래픽적 상황을 나타내는 예를 보이고 있다. 그림 5와 같은 디자인 환경에서 우선 게임 시나리오에 적합한 형태를 구성한다.

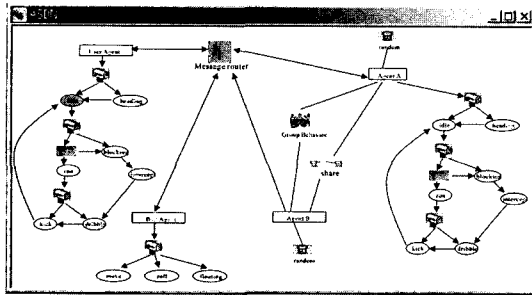


그림 6 에이전트 상태 결정 네트워크 설계 과정

본 논문에선 미니 축구 게임 시나리오에 상호 작용 관계와 기법을 적용한 예를 설명한다. 그림 5의 1번 장면 그래프 계층 구조에서 사용자 캐릭터를 선택하여 마우스로 드래깅하여 저작 환경에 디자인 하고자 하는 사용자 에이전트 객체를 놓아 둔다. 그림 6의 1과 같은 사각형 상자가 나타나며 사용자 에이전트 객체의 이름이 등록된 정보가 표시된다. 여기서부터 사용자 에이전트 객체에 대한 동작 상태나 제어 정보가 설계된다. 사용자 에이전트 객체가 가져야 할 축구 동작은 “달리다”, “공을 차다”, “공격을 방어하다”, “공을 가로채다” 등과 같은 요소를 가진다. 다시 동작 정보가 등록되어 있는 그림 5의 1번으로 가서 에이전트 객체에게 부여하고 싶은 동작 상태 요소를 선택한다. 여기서는 사용자 에이전트 객체의 초기 상태를 idle 상태로 지정하며 idle 플레이스의 색상이 변경된다. 지정 이유는 목적에 따라 다음 상태를 탐색하기전에 자신의 자체 행동을 계속 진행처리하기 위한 의도이다. 동일하게 마우스를 드래깅하여 그림 6의 2번과 같이 저작 환경 공간 상에 적절히 배치한다. 그림 6의 4번 편지함 모양의 아이콘은 트랜지션을 의미하며 이 아이콘은 그림 5의 3번의 버튼 도구에서 선택하고 에이전트의 다음 상태에 대한 조건식을 정의한다. 그림 5의 3번 도구 버튼에서 연결을 의미하는 화살표 버튼을 클릭 후 트랜지션과 동작 요소를 연결하기 원하는 에이전트 객체를 클릭하여 그림 6의 3번과 같이 서로 연결시킨다. 이로써 첫번째 동작 상태 요소가 추가 되고 이와 같은 형태로 다른 동작 상태 요소도 추가 할 수 있다. 그림 6에서 5번의 탐색을 의미하는 아이콘은

사용자 에이전트 객체가 목표로 하는 공 에이전트 객체에 대한 정보와 다른 에이전트 객체의 정보를 탐색하는 제어 과정을 의미한다. 이와 같은 방법으로 원하는 에이전트 객체에게 동작 판단 상태 정의와 제어 정보를 추가 하여 게임 환경에 적절한 판단 네트워크 구조를 가지게 한다. 다른 에이전트 객체인 에이전트 A와 B는 거의 동일한 정보 구조를 가지고 있으므로 그림 5의 에이전트간 부가적 기능 제어를 선택 처리 할 수 있는 5번의 제어 아이콘에서 공유 아이콘을 선택한다. 그림 6의 6번과 같은 공유 아이콘은 동일한 정보를 서로 공유할 수 있도록 하는 역할을 한다. 공유 아이콘으로 연결된 에이전트 객체들의 각 상태 정보는 그림 5의 4번에서 각 에이전트 객체의 메시지와 기능을 체크하여 현 장면상의 공유된 에이전트 객체의 각 상태를 알아낼 수 있다. 서로 같은 그룹에 속한 에이전트 객체는 그림 6의 8번과 같이 그룹 형태로 지정하여 그룹의 행동 형태가 구성되도록 디자인 한다. 그림 5의 5번에서 그룹 행동 아이콘을 선택하여 지정한다. 또한, 사용자 에이전트 객체와는 달리 다른 캐릭터 에이전트 객체는 여러 명이 있을 수 있으므로 행동의 상태가 저마다 랜덤 형식으로 속성을 정하여 보다 자연스러운 장면이 생성되도록 한다. 그림 6에서 7번인 시계 모양의 아이콘이 이러한 속성을 지정할 수 있도록 그림 5의 5번에서 선택하여 저작 영역에 배치한다. 공 에이전트 객체는 간단히 몇 개의 동작으로 구성한다. 공이 위로 튕기다. 공이 굴러가다 등과 같은 동작 요소를 가진다. 각 에이전트 객체의 정보 전달은 메시지 라우터를 통해 전달되며 메시지 라우터는 각 에이전트 객체의 상황에 맞는 메시지 정보를 처리하게 된다. 이렇게 디자인 한 후 컴파일 버튼을 누르면 전체 구조와 잘못된 상태를 체크하여 이상이 없으면 파싱된 정보는 게임 실행시 실시간으로 체크되어 상황을 전개해 나간다. 이와 같은 게임 실행 과정은 사용자 캐릭터 에이전트 객체의 이동 처리 및 최종 목적을 달성하기까지 계속 반복되어 진다. 실시간 게임 실행 과정에서 각 에이전트 객체의 현 상태 정보인 플레이스가 색상을 변경함으로써 오류나 상황 판단을 사용자가 인터페이스 상에서 체크하여 볼 수 있게 한다. 그림 7은 위의 그림 6과 같이 디자인된 내용이 그림 7과 같이 파싱되어 처리되는 내부 과정을 나타낸다. 게임 객체간의 더 세부적인 묘사와 디자인은 게임 시나리오의 목적에 따라 추가, 변경 가능하다. 이 정보는 객체 지향적인 형태로 지식 베이스가 저장된 파일 매니저에서 저장하여 실시간으로 메시지를 체크하여 의도하는 정보를 처리한다. 파싱된 정보는 일종의 추론 모듈을 구성하여 상호작용의 결과를 예측한다.

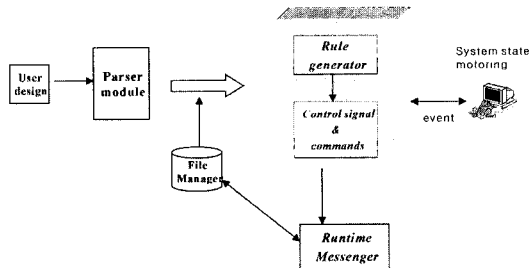


그림 7 ASDN의 해석 처리 과정

규칙 결정을 위한 상태정보(P)와 실행 조건 규칙(T) 들은 게임 시나리오에 따라 지식 베이스 형태로 저장된다. 그림 6과 7의 이러한 대화 모델 분석을 통해 전체적인 문제를 단일화하고 분석하여 게임 객체간 데드락을 회피한다. 이것은 현재의 게임 시나리오 상황에 따른 게임 에이전트 객체간 동작 애니메이션을 안정성 있고, 충돌회피를 보장하는 역할을 하게 된다.

4. 상호작용 모델 설계를 위한 시스템의 구현

본 논문에서 제안한 3차원 게임 환경 내에서 게임 객체간의 상호 작용 모델 디자인과 조정을 위한 기법의 시스템은 Visual C++ 6.0과 OpenGL을 사용하여 구현하였다. 그림 8, 9, 10은 이런 기법을 사용하여 미니 축구 게임 시나리오에 적용한 결과를 나타낸다.

그림 8과 같이 상태 변화를 나타내는 플레이스들이 색상이 변경되므로 실시간으로 체크하여 게임 환경의 내용에 따른 실제 변화 모습이 변경 처리되는 상황을 보여준다. 다른 캐릭터 에이전트 객체의 정보 처리 또한 동시에 모니터링 할 수 있으므로 오류와 상황 전개를 판단하는 중요한 요소를 제공한다.

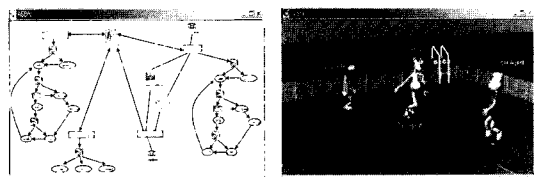


그림 8 각 에이전트 객체의 상태(a)

그림 8은 사용자 캐릭터 에이전트가 공을 차려는 내부적 의도를 가지고 공 에이전트 객체를 향해 달려가는 상태를 나타낸다. 주변의 다른 에이전트 객체인 에이전트 A, B는 사용자 캐릭터와 공 에이전트 객체를 인지하여 사용자 캐릭터 에이전트가 공을 소유하려는 상태를 막기 위해 공 에이전트 객체를 향해 달려가는 상황을

묘사하고 있다.

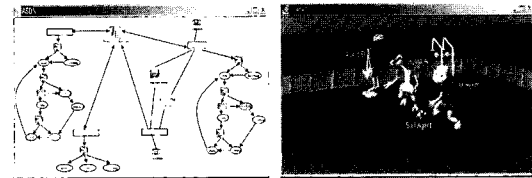


그림 9 각 에이전트 객체의 상태(b)

그림 9는 사용자 캐릭터 에이전트 객체가 공을 소유한다는 가정을 나타내고 있다. 계속해서 사용자 캐릭터 에이전트 객체는 공을 차는 행위를 하려고 한다. 이때 에이전트 A가 이를 방해하려는 동작 상태를 구성한다. 에이전트 B는 점프하여 공 에이전트의 이동 방향을 바꾸려는 동작 형태를 형성한다.

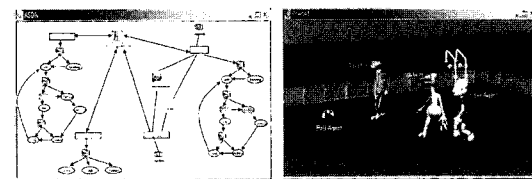


그림 10 각 에이전트 객체의 상태(c)

그림 10은 사용자 캐릭터 에이전트가 공을 찬 후의 상태를 나타낸다. 공 에이전트 객체는 사용자 캐릭터 에이전트가 의도하는 방향으로 이동되도록 처리된다. 에이전트 A는 사용자 캐릭터 에이전트의 진로나 공을 차는 행위를 막으려는 행동을 계속 실행한다. 에이전트 B는 공이 이동된 방향으로 공을 소유하려는 의도를 가지고 공 에이전트를 향해 달려가는 상황을 나타낸다. 그림 8, 9, 10에서 각 에이전트의 내부 상태가 상황에 따라 변경되는 모습을 모니터링하여 체크할 수 있다. 이와 같은 과정이 사용자 에이전트 객체가 자신의 최종 목표를 달성할 때까지 입력 처리된다.

본 논문에서는 에이전트 기법을 기반으로 하여 3차원 게임 환경에서 여러 게임 객체간의 동작 및 상호 작용 결과를 설계하고 해석하는 방법이 게임 시나리오 즉, 장면 개발을 위한 효과적 조정 프레임 워크를 제공한다는 것을 실례로 보여준다. 이러한 구조가 일관된 게임 장면 시나리오의 조정 해결책을 가져다 주기 위해 사용되어 질 수 있다. 이 기법은 여러 가지 게임 시나리오 작성에 따른 조정 등에 응용될 수 있다. 또한, 복잡한 상호작용 묘사가 가능하다. 각 에이전트 객체 관련 동작 정보는

일련의 파일 형태로 저장되어 있다. 인체 모델 캐릭터와 유사한 캐릭터 에이전트들의 동작 정보는 상황에 따라 역 운동학 기법을 사용하여 적용하였다

5. 결론

3차원 게임 환경에서 각 객체들은 단순 기하학 정보만을 갖지는 않는다. 많은 의미적 정보와 상호작용 관계의 의미를 포함해야 한다. 본 논문에서 개발한 기법은 3차원 게임 환경에서 사용자의 제어와 게임 객체와의 상호 작용을 디자인 할 수 있는 환경을 제공한다. 시각적 인터페이스 도구를 통하여 에이전트 객체간 상호 작용 디자인과 사용자의 의도를 반영한 시나리오 설계를 가능하게 하는 환경을 개발하였다. 사용자 인터페이스의 동적인 속성을 설명하기 위한 시간적, 논리적, 상호 작용 관계들의 효율적 디자인을 환경을 제공한다. 3차원 게임 환경에서의 게임 에이전트 객체의 행동 설계를 위한 에이전트 상태 결정 네트워크는 동시에 기본적 동작과 복잡하고 다양한 동작 형태에 추가하여 상호작용 행동, 전체 동작 스케줄링에 효율적으로 사용될 수 있다. 이를 통해 3차원 게임 환경 내에서의 게임 캐릭터들간의 동작을 조정하는 역할을 할 수 있다. 스크립트 형태의 언어적 방법을 사용하여 보다 유연성 있는 생성 규칙을 정의할 수 있도록 하였다. 즉, 여러 에이전트 객체와 사용자 객체가 상호작용하는 관계뿐만 아니라 에이전트 객체간 서로 의사 소통할 수 있는 방안을 제시하여 일관된 상호 작용 형태가 가능하도록 설계하였다. 이 기법은 특정 문제 즉, 게임 시나리오 형태에 따라 다양하게 확장하고 적용 가능하다. 또한, 동적으로 동작이 변경되는 애니메이션 형태에 대한 각각의 게임 캐릭터가 가지는 에이전트 객체 사이의 의사 교환과 조정, 협동 지식의 획득과 수정, 모델링을 하기 위한 유틸리티와 에이전트 서비스를 제공한다. 사용자는 세부적인 사항에 관심을 두지 않고 게임 전체의 시나리오에만 집중하여 개발할 수 있는 환경을 제공한다. 인간과 컴퓨터 상호작용 측면에서 게임의 시나리오에 따른 변화를 반영할 수 있다. 그러나, 게임 시나리오는 게임 장르에 따라 유동적이므로 관련 객체 데이터 정보를 테이블 형태로 구조화되어 처리되어 있어야 한다. 따라서 좀더 다양한 인터페이스 도구 개발과 인지적, 추론 능력을 가지는 모듈이 계속, 수정 보완되어야 한다.

참고 문헌

- [1] L. Levison, "Connecting Planning and Acting via Object-specific Reasoning," PhD thesis, Department of computer & Information Science, University of Pennsylvania, 1996.
- [2] C. Hand, "A survey of 3D Interaction Techniques," Computer Graphics Forum 16, 269-281.
- [3] D. Bowman and L. Hodeges, "Formalizing the Design, Evaluation, and Application of Interaction Techniques," Journal of Visual Languages and Computing 10, 19-35.
- [4] I. Poupyrev and T. Ichikawa, "Manipulating Objects in Virtual Worlds: categorization and Empirical Evaluation of Interaction Techniques," Journal of Visual Languages and Computing 10, 19-35.
- [5] Y. Okada, K. Shinpo, Y. Tanaka and D. Thalmann, "Virtual Input Devices Based on Motion Capture and Collision Detection," Proceedings of Computer Animation 99, Geneva, May 1999.
- [6] N. Farenc, S. R. Musse, E. Schweiss, M. Kallmann, O. Aune, R. Boulic & D. Thalmann, "A Paradigm for Controlling Virtual Humans in Urban Environment Simulations," Applied Artificial Intelligence Journal 14, 69-91 ISSN 0883-9514, 2000.
- [7] M. Kallmann and D. Thalmann, "Modeling Behaviors of Interactive Objects for Real-Time Virtual Environments," Journal of Visual Languages and Computing, pp. 177-195, 2002.
- [8] Wagner da Silva, F., Garcia, L. M. Farias, R. C., Oliveira, A. A. F., "A Control Theory Approach for Real-time Animation of Artificial Agents," Proceedings XIII Brazilian Symposium on Computer Graphics and Image Processing, pp.211-218, 2000.
- [9] P. Bechieraz, D. Thalmann, "A Behavioral Animation System for Autonomous Actors Personified by Emotions," Proc. Of first Workshop on Embodied Conversational characters(WECC98), Lake Tahoe, California, 1998.
- [10] H. Noser, D. Thalmann, "The Animation of Autonomous Actors Based on Production Rules," Proceedings of Computer Animation, Geneva, Switzerland, 1996.
- [11] C. Reynolds, "Steering Behaviors For autonomous Characters," Game Developers Conference, San Jose CA, March 15-19, 1999.
- [12] X. Tu, D. Terzopoulos, "Artificial Fishes: Physics, Locomotion, Perception, Behavior , Proceedings of SIGGRAPH94," Computer Graphics, July, 1994
- [13] M. Kallmann, J. Monzani, A. Caicedo and D. Thalmann, "ACE: A Platform for the Real Time Simulation of Virtual Human Agents," EGCAS 2000-11th Eurographics Workshop on Animation and Simulation, Interlaken, Switzerland.
- [14] S. Franklin and A. Graesser, "Is It an Agent, or just a Program?: A Taxonomy for Autonomous

- Agents," Proceedings of the 3rd International Workshop on Agent Theories, Architectures, and Languages, Springer Verlag, Berlin/Heidelberg, Germany.
- [15] M. Kallmann and D. Thalmann, "A Behavioral Interface to Simulate Agent-Object Interactions in Real-Time," Proceedings of Computer Animation 99, IEEE Computer Society Press, Geneva, 138-146, 1999.
- [16] A. Scholer, R. Angros, J. Rickel, and W. L. Johnson, "Teaching Animated Agents in Virtual Worlds," Proceedings of Smart Graphics, March 20-22, Stanford, USA, 2000.
- [17] C. Barnes, "Virtual Programming Agents for Virtual Environments," Proceedings of Smart Graphics, March 20-22, Stanford, USA, 2000.



김 기 현

1997년 대구대학교 컴퓨터공학과 졸업(학사). 1999년 경북대학교 컴퓨터과학과 졸업(석사). 2001년 경북대학교 컴퓨터과학과 박사 수료. 관심분야는 캐릭터 애니메이션, 에니전트 시스템, 인간과 컴퓨터 상호작용, 멀티미디어 시스템 등

김 상 욱

정보과학회논문지 : 컴퓨팅의 실제
제 9 권 제 1 호 참조