제22회 한국정보처리학회 추계학술발표대회 논문집 제11권 제2호 (2004. 11)

DirectX를 이용한 네트워크 게임 설계에서 생성 패턴 적용에 대한 연구

김종수°, 김태석
동의대학교 소프트웨어공학과
e-mail:seatree@deu.ac.kr, tskim@deu.ac.kr

Study on the Creational Patterns Application for the Network Game Design using DirectX

Jong-Soo Kim°, Tai-Suk Kim
Dept. of Software Engineering, Dong-Eui Univ.

요 약

현재 우리나라에는 사용자들의 사실감을 더해주는 3D 기반 온라인 게임이 큰 주류를 이루고 있다. 국내에 많은 게임 개발 업체들이 있는데, 이들 업체에서는 게임 엔진 개발 의욕은 높은 반면 관련 전문 인력이 부족하고, 업체 규모가 영세한 경우가 많다. 또한 게임 설계와 관련된 기술은 보안이 철저하기 때문에 기술 공유가 사실상 힘들다. 인력과 시간이 많이 드는 네트워크 게임제작에서 기존에 작성된 코드를 재사용이 가능하도록 소프트웨어를 설계하는 것은 중요한 일이다. 지금까지 검증된 여러 가지 디자인 패턴들이 재사용을 위한 소프트웨어 설계에 도움을 준다. 본 논문에서는 DirectX를 기반으로 하는 네트워크 게임의 클라이언트 측 설계에 GoF(Gang of Four)의 디자인 패턴 영역 중 생성패턴을 이용하여 재사용이 가능한 네트워크 게임 설계에 대한 효율적인 기법을 제안한다.

1. 서론

전 세계적으로 게임 산업은 MS사의 X-Box, Sony의 PlayStation II 등이 시장 선점을 위해 각축하고 있으며, 새로운 차원의 실시간 3D 게임 제작을 위해 노력하고 있다. 현재 우리나라에서도 발달된 네트워크 인프라와 디지털 기술을 바탕으로 SW산업이 기간산업으로 성장할 수 있는 기회를 맞고 있다. 또한 국내의 몇몇 회사들도 기존에는 거의 불가능할 것으로 예상되었던 네트워크 기반 실시간 전략 시뮬레이션 게임을 제작하고 있다[1].

게임 소프트웨어뿐만 아니라 모든 소프트웨어들의 라이프 사이클이 짧아지고 있는 추세이므로, 재미있는 게임을 얼마나 빨리 개발할 수 있느냐 하는 것은 중요한 일이다. 그러나 게임 소프트웨어를 설계하고 개발하기 위한 기법은 회사의 중요 자산이므로 아주 철저하게 보안되고 있고, 정보의 공유가 어렵다. 이러한 이유로 게임 개발 기술에 적용되는 설계기법의 효율성에 대한 평가는 거의 없다[2].

본 논문에서는 DirectX를 기반으로 하는 3D 네트워크 게임 제작에서 소프트웨어의 재사용을 높일 수 있는 GoF의 디자인 패턴 영역 중 생성 패턴의 적용에 대해서 연구한다. 소프트웨어 개발에 있어서 객체 지향 패러다임을 적용하는 이유는 여러 가지가 있지만, 분산된 개발을 가능하게 하고, 기존에 개발된 소프트에어의 재사용을 쉽게 한다는 장점이 있기 때문이다[3]. 본 논문에서는 MS사가 제공하는 DirectX를 기반으로 하는 3D 게임 제작에 있어서 GoF(Gang of Four)가 제안한 몇 가지 효율적인 디자인 패턴에 대해서 논의한다. 본 논문의 2장에서는 DirexctX를 기반으로 하는 네트워크 게임 기술과 소프트웨어 설계와 관련된 디자인 패턴에 대해서 기술하고 3장에서는 사용사례를 이용한 게임 분석과 설계에 대해서 기술하고 4장에서는 UML을 이용한 DirectX 게임 설계에 있어서 디자인 패턴의 큰 분류인 생성패턴의 적용과 그 유효성을 평가한다. 5장에서는 설계를 바탕으로 한 게임 시스템의 구현에 대해서 기술하고, 마지막 6장에서는 결론과 향후 연구과제를 제시한다.

2. 관련 연구

네트워크 게임 엔진 설계와 제작은 소프트웨어의 재사용이라는 측면이 고려되어야 하므로, 엔진간의 인터페이스가 서로 종속적이지 않아야 효율적이다. 그림 1은 일반적인 네트워크 게임에서 엔진의 구성을 보여준다.

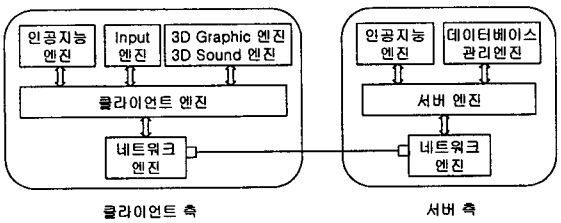

그림 1 클라이언트/서버 게임의 엔진 구성도

3D 게임 엔진은 크게 클라이언트 부분과 서버 부분으로 나눌 수 있다[4]. 인공지능 엔진은 클라이언트와 서버가 모두 필요하고, Input 엔진과 3D Graphic, 3D Sound 엔진은 클라이언트 측, 데이터베이스 관리 엔진은 서버 측에 위치한다.

실시간 네트워크 게임 제작에 있어서, 많은 시간을 투자해야할 부분이 3D 그래픽, 에니메이션 그리고 사운드 엔진 제작이라고 할 수 있다. 사용자를 게임에 몰입시킬 수 있도록 다양한 종류의 애니메이션이나 사운드를 만들어 내야하기 때문이다. 3D 그래픽 엔진은 렌더링과 에니메이션 기능을 가지고 있어야 한다[5]. 렌더링 API 인터페이스로 OpenGL이나 DirectX를 사용할 수 있는데, 본 연구에서는 MS사의 DirectX를 기초로 연구되었다. 현재 DirectX는 버전 9.0이 나와 있고, 이와 관련된 다양한 예도 같이 제공해 준다.

DirectX는 렌더링 엔진으로써 정점(Vertex), 폴리곤(Polygon), 메시(Mesh), 행렬(Matrix), 광원(Lighting), 텍스처(Texture) 등과 같은 것을 쉽게 다룰 수 있는 기능이 있다[6].

게임과 같이 복잡한 소프트웨어를 작성하는데 있어서 패턴의 사용은 매우 효율적이다. 소프트웨어 설계 패턴은 개발자가 다른 사람의 전문성을 이용할 수 있게 한다. 그리고 설계 패턴을 이해함으로써 설계 아이디어를 재사용할 수 있게 된다.

GoF가 제안한 디자인 패턴은 크게 3개의 패턴 영역으로 분류되는데, 각각은 생성, 구조, 행위패턴이다. 본 논문에서는 DirectX를 이용하여 네트워크 게임 제작 시 적용될 수 있는 생성패턴의 영역 중에서 추상 팩토리(Abstract Factory), 팩토리 메소드(Factory Method), 싱글톤(Singleton)패턴이 적용될 수 있는 기법에 대해서 연구하였다.

3. DirectX 게임 분석 및 설계

게임 설계에 있어서 GoF가 제안한 디자인 패턴을 적용하였는데, 게임 소프트웨어의 분석과 설계에도 디자인 패턴이 유용하게 적용될 수 있음을 보이고, 소프트웨어 개발 단계에 따른 분석과 설계 시에 디자인 패턴을 이용함으로써, 객체지향 패러다임이 가지고 있는 소프트웨어 분산 개발과 소프트웨어 재사용이라는 측면이 최적의 게임 소프트웨어를 제작하는데 효용이 있음을 보인다.

3.1 사용사례(use case) 작성

요구 사항은 프로젝트가 갖추어야할 능력과 조건이다. 사용사례는 요구사항을 알아내고 기록하는데 널리 사용되는 기법이다[7]. 게임 소프트웨어 분석 설계에서는 사용자의 요구 사항이 정해지지 않은 경우가 많다. 그래서 사용자에게 요구 사항을 직접듣기보다는 기존의 소프트웨어를 분석하고, 게임 소프트웨어 개발자들 간의 아이디어 회의를 통해서 요구사항을 분석해 내는 경우가 많다.

표 1 "Legend of Avantas" 게임 스토리

스토리 보드
제목 : L.o.A(Legend Of Avantas) 장르 : MMORPG(Massive Multi-player Online Role Playing Game) 게임 스토리 및 배경 고대 상계에 신계와 마계가 있었다. 두 계는 공존을 이루지 못하고 1500년간의 마계전쟁을 치르고 (...생략) 마을별 특징 및 특성 ① 라피돌 마을 아반타스의 신으로부터 내려온 라세스 종족이 머무는 곳 ② 멘카토 마을 북쪽에 위치하고 있는 춥고, 어두운 땅이지만 휴먼종족에 과학의 힘으로 멋진 도시로 탈바꿈 중이다. .....(이하 생략)

DirectX를 이용한 게임은 다양한 장르가 있지만, 본 논문에서 구현된 게임은 리니지(lineage)와 같은 MMORPG(Massive Multi-player Online Role Playing Game) 형태의 네트워크게임을 위한 클라이언트 측 구현이다. 표 1은 구현하고자 하는 게임의 스토리를 보여준다. 표1의 게임스토리를 바탕으로 한 요구사항 분석은 그림 2와 같이 나타났다.

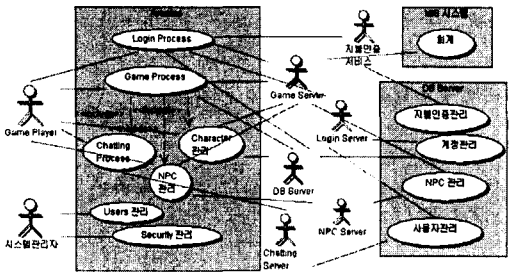

그림 2 사용 사례(Use Case)를 통한 요구사항 분석

요구 사항을 분석해 보면 먼저 클라이언트 측의 가칭 "Avantas" 시스템 경계에서 나타난 Login과 관련된 처리, 게임 진행과 관련된 처리, 게임 캐릭터를 관리하기 위한 처리, 서버 간의 채팅을 처리하기 위한 프로세서, 사용자와 그에 따르는 보안 관리 등이 필요하다.

3.2 DirectX를 이용한 애플리케이션 디자인

표1의 스토리보드와 요구사항 분석을 토대로 분석된 주요 클래스의 기능과 적용되어야 할 애플리케이션은 표 2와 같다.

표 2 애플리케이션 구현에 필요한 클래스 분석

주요 클래스	주요 기능
MainProcess	필요 이미지와 메모리 초기화
GameManager	게임 시작과 끝의 상태 관리
InputManager	마우스와 키보드를 제어
UIManager	사용자 인터페이스를 제어
Effect	움직임에 대한 효과 관리
AIManager	인공지능 관리
Terrain	게임 영역 관리
QuadTree	지형 검색 클래스
AseLoad	Ase 파일 처리
ToolManager	캐릭터에 필요 변수 초기화
DirectInput	마우스와 키보드 조작
Direct3D	렌더링 처리
DecodeTKM	3D Max 파일 처리
Model	모델의 정보 관리
ModelManager	모델의 상태와 렌더링을 처리
Camera	카메라를 관리

표 2의 자료를 분석해 보면, 네트워크 게임 제작에 있어서 소프트웨어를 구현하기 전에 선행되어야 하는 분석과 설계 작업이 효율적인 클래스를 만들 수 있고, 소프트웨어 재사용을 위한 설계에 다양한 생성 패턴이 적용될 수 있음을 볼 수 있다.

4. UML을 이용한 DirectX 게임 설계

네트워크 게임의 클라이언트 측 구현에서 기존에 정의된 상속관계와 클래스의 합성과 집합(Composite and Aggregation)관계를 토대로 하여, 효율적인 소프트웨어의 제작을 위한 생성 패턴의 적용으로 GoF가 제안한 팩토리(Factory), 싱글톤(Singleton) 패턴의 적용을 검토한다.

4.1 Factory 패턴 적용

구조 패턴영역의 어댑터 클래스는 소프트웨어 설계에서 자주 사용되는 패턴이다. 그런데 어댑터를 사용하기 위해서는 생성시점이 문제된다. 현재 나타난 설계에서 어댑터 클래스의 생성과 같은 경우, 특정 클래스에 어댑터클래스를 생성해야하는 책임을 주는 것은 도메인 객체의 응집도를 떨어뜨린다. 어댑터 클래스의 생성에 있어서 설계의 기본 원리인 관심사항의 분리(separation of concerns) 원칙을 적용하는 것이 더 좋은 방법이다.

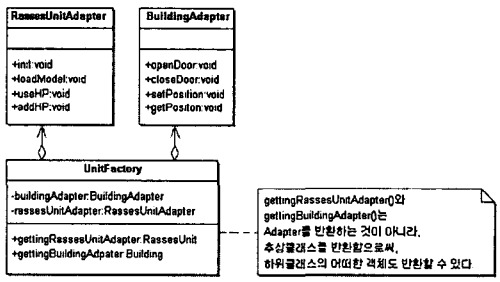

그림 3 Adapter클래스를 생성하기 위한 UnitFactory클래스

그림 3은 RassesUnitAdapter와 BuildingAdapter를 생성하기 위한 팩토리(Factory)를 정의하고 있다. UnitFactory 클래스가 2개의 어댑터클래스 생성 책임을 맡고 있다. UnitFactory 클래스 내에서 gettingRassesUnitAdapter()는 RassesUnit을 반환하고 gettingBuildingAdapter()는 Building을 반환한다. 각각의 메소드는 Adapter클래스를 반환하는 것이 아니라 어댑터의 추상클래스를 반환함으로써, 다형성에 의해 하위클래스의 어떠한 객체도 반환할 수 있게 된다. 어댑터 클래스의 생성에 팩토리 패턴을 사용함으로써 복잡한 객체 생성 책임을 응집도가 높은 협력 객체로 분리할 수 있고, 잠재적으로 복잡한 생성논리를 숨길 수 있다.

4.2 Singleton 패턴 적용

그림 3의 UnitFactory 클래스의 사용에 고려해보

아야 하는 것은 클래스 자체의 생성과 접근을 책임지는 객체를 선택하는 문제다. UnitFactory의 경우 어댑터를 생성하기 위한 팩토리이므로 인스턴스가 여러 개일 필요가 없고 한개만 생성되면 된다. 이러한 문제는 그림 4와 같은 싱글톤(Singleton) 패턴이 사용될 수 있다.

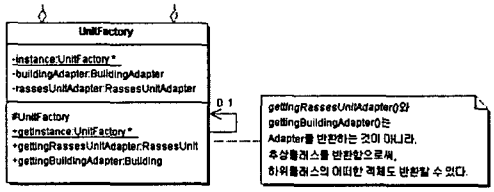

그림 4 UnitFactory에 적용된 Singleton 패턴

그림에서 UnitFactory는 private으로 선언된 인스턴스를 속성으로 가지고 있고, protected로 선언된 생성자가 있으며, private 인스턴스에 접근할 수 있는 정적 메소드인 getInstance()를 가지고 있다.

5. 생성 패턴을 이용한 시스템 구현

그림 6에서 보는 게임은 DirectX 9 API를 이용하여 네트워크 게임의 클라이언트 측 기본 기능 구현에 중점을 둔 게임 애플리케이션이다.

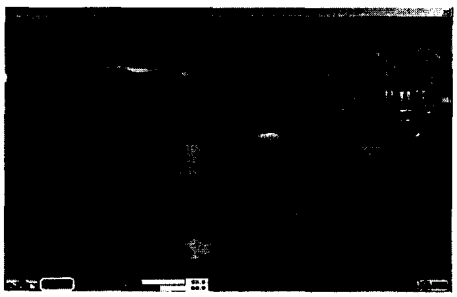

그림 5 DirectX 3D를 이용한 네트워크 게임의 예

DirectX가 지원하는 다양한 스크린 모드가 있지만 구현에 사용된 스크린 모드(mode)는 800x600 모드를 사용하였다. 애니메이션 처리는 메인 메모리의 크기, 선택된 스크린 모드와 컬러, 그래픽 카드의 성능을 모두 고려해야 하는데, 구현된 애플리케이션에서 DirectX 3D 렌더링 API는 대략 50~90 프레임을 처리하였다.

게임에서 간편한 사용자 인터페이스를 위해 캐릭터의 조작에 간단한 키보드 조작과 마우스의 입력만으로 이동과 공격, 메뉴 선택이 가능하도록 하였다.

6. 결론

본 논문에서 DirectX API를 기반으로 하는 네트워크 게임 클라이언트 측 설계와 구현을 통해서 효율적인 설계 기법을 제안하였다. 어댑터 클래스의 객체 생성과 관련해서 클래스의 응집도를 높이기 위해 생성패턴에 속하는 팩토리 패턴과 싱글톤 패턴이 적용될 수 있음을 볼 수 있었다.

본 논문에서는 DirectX API를 이용한 게임의 설계에서 GoF의 생성 패턴이 어떻게 적용될 수 있는지에 대해 연구 하였는데, 향후에는 Direct Play를 이용하여 배틀 넷 서버와 연동할 수 있는 게임 서버의 구성에 대해서 연구할 예정이다.

참고문헌

[1] 최성 "게임 산업과 기술 전망", 정보처리학회지 특집 게임 기술, pp.11-23, 2002.

[2] 김종수(2003) "웹을 기반으로 한 JAVA 네트워크 게임 시스템의 설계와 구현", 부산외국어대학교 대학원, 석사학위논문.

[3] Erich Gamma, Richard Helm Ralph Johnson, Hohm Vissides 공저 "GoF의 디자인 패턴", Pearson education Korea(440 pages)

[4] 남재욱 "온라인 게임 서버 프로그래밍", 한빛미디어, 2004(475 pages)

[5] 안승중 "Direct 3D", 도서출판대림, 2000(1220 pages)

[6] 김용준 "3D 게임 프로그래밍", 한빛미디어, 2003(725 pages)

[7] GRAIG LARMAN 원저 "UML과 패턴의 적용" 홍릉과학출판사(705 pages)

[8] 김종수, 권오준, 김태석 "네트워크 기반 다자간 아바타 채팅 시스템의 구현", 한국멀티미디어학회 춘계학술발표논문집, pp.171-174, 2003.

[9] 김종수, 이종민, 김태석 "생성 패턴을 사용한 네트워크 기반 게임 API 설계", 한국멀티미디어학회 추계학술발표대회논문집, pp.669-674, 2003.

[10] http://www.gpgstudy.com