제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

컴포넌트 기반 게임엔진 개발 프로세스의 적용모델

김정종, 송의철, 박운재, 송호영
경남대학교 컴퓨터공학과
e-mail:songec@saekyung-c.ac.kr

An Application Model for Game Engine Development Process based on Components

Jung-Jong Kim, Eui-Cheol Song, Woon-Jai Park, Ho-Young Song
Dept of Computer Engineering, Kyungnam University

요약

컴퓨터 게임은 소프트웨어 분야의 새로운 고부가가치 영역으로 발전되어가고 있으나 개발공정과 방법론에 있어서는 경험중심의 개발 방법을 고수하고 있다. 따라서 본 논문에서는 기존의 개발방법을 정형화된 형태로 개선하기 위하여 컴포넌트 기반 개발방법의 적용을 위한 게임개발 프로세스의 정형화 모델을 제안하고, 재사용성의 향상에 따른 문제점과 해결방안을 제시하며, 정형화된 모델의 재사용 효율성을 위하여 컴포넌트 적용 방법을 이용한 프로세스를 정형화하는 방안을 제시하고자 한다.

1. 서론

최근 컴퓨터 게임은 소프트웨어 분야의 새로운 고부가가치 영역으로 발전되어가고 있으나 개발방법, 재사용성 등이 매우 저조한 실정이다. 뿐만 아니라 게임 소프트웨어 이용자의 사용환경은 매우 빠른 속도로 발전하고 있으나 이러한 환경에 빠르게 적응할 수 있는 게임엔진의 개발은 코드중심의 개발로 인하여 유지보수에 많은 비용이 소요되는 개발공정을 가지고있다.

따라서 새로운 시스템 개발시 이전에 개발된 엔진 부분들을 분석단계부터 반복하고 있는 실정이며, 이러한 과정을 반복하는데는 개발자의 독창성을 극대화하기 위한 이유도 있지만 시스템의 규모와 환경 등이 대형화되어 가는 개발환경에서는 부적합 할 뿐 아니라 게임성이 뛰어난 새로운 시스템 개발에 많은 문제점을 가지고 있다.

1) 개발공정의 미비로 개발 기간과 개발비용 산출의 정확성이 떨어지므로 과다한 개발비용 부담.
2) 일반적인 응용소프트웨어에 비해 재사용성이 매우 저조하며, 재사용의 범위도 코드 중심의 참조 수준에 머물고 있다.
3) 컴포넌트 재사용을 위한 적용방법과 게임개발 방법에서 잘못된 적용방법으로 재사용성이 매우 저조함.
4) 게임엔진 개발 방법에서의 컴포넌트 종류와 검정방법, 관계성 등은 일반적인 응용소프트웨어 개발방법으로는 적용이 불가능함.

따라서 이러한 문제점을 해결하기 위하여 컴포넌트 기반 개발방법의 적용을 위한 게임개발 공정의 정형화 모델을 제안하고, 재사용성의 향상에 따른 문제점과 해결 방안을 제시하였으며, 정형화된 모델의 재사용 효율성을 위하여 컴포넌트 적용 방법을 정립하였다.

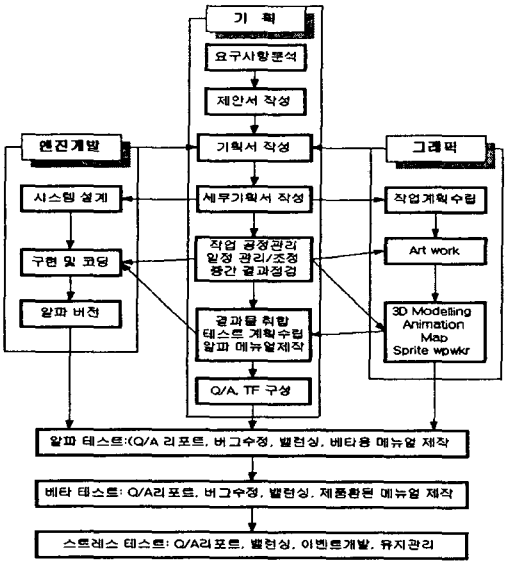
2. 기반연구

2.1 일반적인 게임개발 프로세스

일반적인 게임 개발비용에는 몇몇 중요한 핵심요소가 큰 영향을 미치고 있으며, 게임 소프트웨어 개발의 전체 개발비용 중 40~70% 정도가 게임의 엔진을 개발하는데 투입되고 있다.

게임 개발자가 게임엔진을 게임에 적용하는 방법은 2가지로 구분할 수 있으며, 그 형태는 자체엔진을 개발하는 경우와 상업적인 엔진을 이용하는 방법으로 구분할 수 있다. 이러한 상업적인 엔진의 경우에는 다시 범용적인 것을 지원하는 엔진과 특정 목적에 맞게 개발된 엔진으로 구분할 수 있다.

게임개발 전체공정의 형태는 [그림 1]과 같으며, 전체 공정 중 본 논문에서 고려하는 부분은 게임엔진과 관련된 부분으로 제한한다.



[그림 1] 일반적인 게임개발 프로세스

2.2 CBD 적용의 필요성

본 논문에서 제시하는 방법을 적용하기 위해서는 개발에 필요한 개발 툴과 컴포넌트가 재사용성과 확장성을 고려하여 게임엔진의 분석과 설계단계에서 구축되어야 하며, 게임엔진 개발의 전체적인 개발주기에 기존 개발된 시스템 부분과 컴포넌트 기반 엔진개발 프로세스가 포함되어야 한다. 물론 본 논문에서 제시하는 방법은 단일 시스템을 개발하는 측면에서 고려할 경우에는 낭비적인 요소로 작용할 수 있으나, 새로운 시스템을 연속적으로 개발시 적용할 경우 많은 장점을 포함하고 있다.

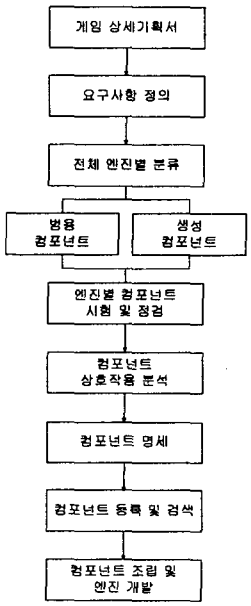
제시한 문제점을 해결하기 위한 적용 부분은 개발 초기단계에서 수정되는 오류와 불필요한 공정을 감소시키는 방법을 제안하고 게임엔진의 개발이 전체 시스템 개발 공정 중 컴포넌트의 적용시점과 분류를 정의한다.

따라서 다음과 같은 문제점을 해결함으로서 시간의 낭비요소는 감소할 수 있으나, 종전의 비정형화된 게임개발 공정에서는 시스템 완료시점이 공정의 복잡도가 증가할 수 있다. 컴포넌트 기반 개발의 적용으로 공정은 단순화되고 게임개발의 주요 목표인 게임성을 증대시키는 실제 개발공정을 처리하는 과정에 투자비용을 할당할 수 있다.

이러한 프로세스의 개선을 위하여 컴포넌트 기반 게임엔진 개발의 프로세스 모델을 제안하고자 한다.

3. 게임엔진 개발의 프로세스

다음 [그림 2]와 같이 제안 프로세스의 워크플로우는 프로세스의 세부기획서, 게임 전체 설계서에 의해 요구사항 정의를 적용하는 부분부터 구현단계에서 컴포넌트를 이용한 게임엔진 개발에 적용하는 모델을 보였다.



[그림 2] 제안 프로세스의 워크플로우

3.1 게임엔진의 분류

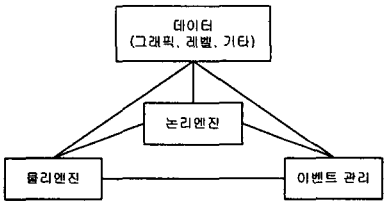
초기의 2D중심의 개발의 경우에는 저해상도의 화면과 적은 메모리용량으로 인하여 운영체제의 제어없이 직접 하드웨어 제어를 할 수 있는 어셈블리 언어를 사용하였다.

최근에는 4GL중심의 프로그래밍언어를 사용하고 대용량의 메모리, 고속의 CPU, 고해상도 디스플레이, 초고속 오디오 프로세서 환경을 이용하여 개발자들은 많은 멀티미디어 컨텐츠를 적용하였으며, 적절한 속도를 유지하기 위해서 다양한 방법을 적용하고 있다.

게임 플레이 선언에 의해 정의된 게임전체 설계와 시스템구조 설계서에 의해 엔진별 모듈을 구분한다. 게임은 크게 엔진부분과 컨텐츠 두 부분으로 구분할 수 있다. 엔진은 레벨 포맷, 그래픽 포맷, 오디오 처

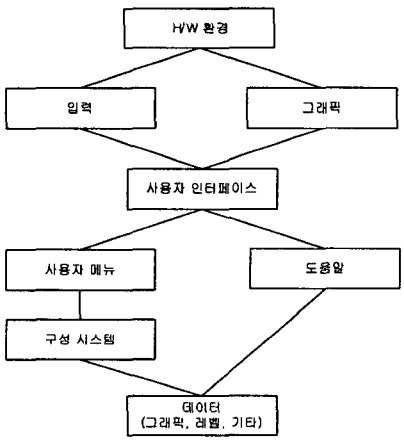
리, 이벤트 관리, 입력 처리, 그리고 중요한 논리적인 영역을 모두 포함한다. 이러한 게임엔진의 영역은 역할에 따라 그래픽엔진, 사운드엔진, 논리엔진, 물리엔진, 사용자 인터페이스 등으로 구분하고 있다.

이러한 게임엔진의 영역별 관계성을 살펴보면 다음 [그림 3]과 같다. [그림 3]은 논리엔진을 중심으로 데이터와 물리엔진, 각종 이벤트 관리관계를 정의하고 있다.

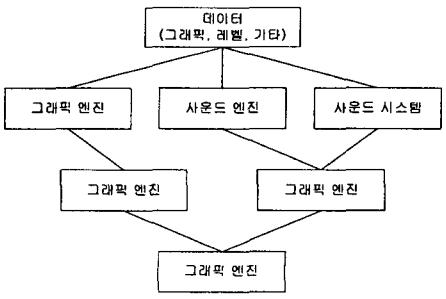


[그림 3] 논리엔진의 관계성

[그림 4]는 사용자 인터페이스와 하드웨어 환경 사용자에게 제공되는 여러 가지 정보와의 관계성을 나타내고 있다.



[그림 4] 사용자 인터페이스의 관계성



[그림 5] 하드웨어 환경과 데이터와의 관계

[그림 5]는 사용자 인터페이스에 필요한 그래픽, 사운드 등에 가공된 게임자료를 지원하기 위한 관계성을 설명하고 있으며, 이러한 데이터는 엔진을 통하여 사용자 인터페이스 부분에 전달된다.

3.2 엔진별 컴포넌트 시험 및 검정

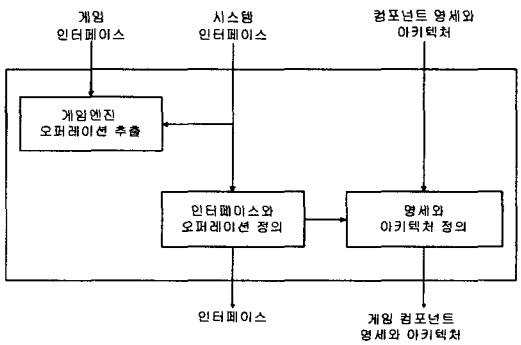
컴포넌트 시험과 검정에 관한 사항은 엔진의 종류에 따라 하드웨어 환경과 데이터의 구성, 시스템에 따라 구분하여 시험 및 검정을 실시하여야 한다. [표 1]은 컴포넌트 검정을 위한 엔진별 상호관계를 나타낸다.

[표 1] 컴포넌트 검정을 위한 엔진별 상호관계

엔진의 종류	데이터의 종류	하드웨어 환경	관련엔진 및 기타
논리엔진	그래픽, 레벨, 기타		물리엔진, 이벤트 관리
사용자 인터페이스	그래픽, 레벨, 기타	데이터 구성 시스템, 입력 장치, 그래픽 장치	사용자 메뉴, 도움말 등
그래픽 엔진	그래픽, 레벨, 기타	그래픽 카드, H/W 기본 시스템	사용자 메뉴, 도움말 등
사운드 엔진	그래픽, 레벨, 기타	사운드 카드, H/W 기본 시스템	

3.3 컴포넌트 상호작용 분석

다음 [그림 6]과 같이 컴포넌트의 상호 작용분석을 위한 방법을 제시한다.



[그림 6] 상호작용 분석방법

3.4 컴포넌트 명세화

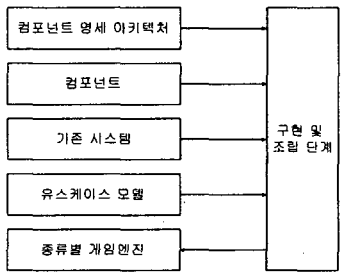
컴포넌트에 의해서 제공되는 인터페이스와 사용되는 인터페이스에 대한 부분도 포함된다. 또한, 예측되는 오브젝트의 개수에 대한 상세한 사항도 포함될

수 있다.

사용자 인터페이스 부분과 각 엔진에 제약을 가할 수 있는 컴포넌트 명세에 상호작용의 간단한 부분까지도 포함한다.

3.5 컴포넌트 조립 및 엔진개발 단계

[그림 7]의 단계는 컴포넌트와 기존의 재사용 가능한 소프트웨어를 이용하여 구현하고자 하는 게임엔진에 결합하고 유즈케이스 모델을 바탕으로 개발목표게임을 위한 인터페이스를 설계하여 목표시스템을 만들어 내는 절차를 말한다.



[그림 7] 구현 및 조립단계

본 논문에서 제시하는 컴포넌트 저장소는 정보의 통합과 교환, 공유를 위한 저장소의 개념이 아닌 게임 소프트웨어 개발의 전 공정에서 효과적으로 이용할 수 있는 통합 지원시스템 형태의 저장소이다.

즉, 아직 개발공정의 정형화가 부족한 게임 소프트웨어 개발공정을 표준화하고 단위 공정별 적용 가능한 항목들을 재사용 할 수 있는 요소들의 구조를 정의하여 다양한 관점의 메타 데이터를 식별하고 분류할 수 있도록 정의한다.

4. 결론

컴포넌트 기반 개발방법의 적용에는 많은 장단점을 가지고 있다. 그러나 개발 기간과 비용적인 측면, 코드의 신뢰성 확보, 유지보수의 편리성을 고려할 경우 기존의 방법과 비교하여 몇 가지 단점을 가지고 있지만 이러한 단점들은 소프트웨어 개발비용과 신뢰성측면을 고려한다면 적용하는 편이 우수하다고 평가할 수 있다.

본 논문에서 제시한 컴포넌트 기반 게임엔진 개발방법의 적용은 평가에서 살펴본 바와 같이 여러 가지 장단점과 다음과 같은 문제점을 해결할 수 있다. 게임개발의 핵심적인 요소라고 볼 수 있는 전체시스템의 자유도와 창조성을 향상할 수 있으며, 소프트웨어 개발시간의 단축과 각 단위별 공정산출의 정확성을 증대시킬 수 있다. 특정한 범용 모듈개발에 집중화와 재사용성의 증대로 단순화된 개발방법의 적용이 가능하므로 정형화된 엔진개발이 가능하다.

따라서 플랫폼으로부터 독립성을 유지할 수 있으므로 코드의 안정적 유지가 가능하며, 재사용 가능성과 관리, 개발자의 지식공유, 시스템 개발주기의 공정에 보다 가시적인 적용이 가능하므로 특화된 시스템 개발 분야인 게임엔진 개발에 효율성을 가져올 수 있다.

물론 위와 같은 여러 가지 장점을 가지고 있지만 컴포넌트의 수집과 최초 적용시 기술의 탄력성이 떨어진다는 문제점도 가지고 있다. 이러한 문제점은 최초의 시스템 개발에 국한된 부분이며, 반복과 지속적인 개발에 있어서는 소프트웨어 생산성 향상에 크게 기여할 수 있다.

앞으로 이러한 시스템을 게임 소프트웨어 개발에 국한된 형태에만 적용하는 형태에서 다양한 멀티미디어 시스템 구축에 필요한 컴포넌트 적용방법의 가능성에 대하여 체계적인 연구가 필요할 것이다.

참고문헌

[1] Andrew Rollings, Dave Morris, "Game architecture and design", LLC, 2000

[2] Luqi, Jiang Guo, "Toward Automated Retrieval for a Softerware Componet Repository", IEEE Conference and Workshop on Engineering of Computer-Based Systems, March, 1999

[3] Ernest Pazera, Isometric Game Programming with DirectX 7.0, Prime Tech, 2001

[4] Andrew Rollings, Dave Morris, Game Architecture and Design, Coriolis Group Books, 2001

[5] Teijo Hakala, Andrew Mulholland, Developer's Guide to Multiplayer Games, WORDWARE, 2001

[6] David H. Eberly, 3D Game Engine Design : A Practical Approach to Real-Time Computer Graphics, 2002

[7] Jim Adams, Programming Role-Playing Games with DirectX, Premier, 2002

[8] Andre Lamothe, Todd Barron, MultiPlayer Game Programming, Prime Tech, 2001.