

스크립트를 이용한 스켈레톤 게임 개발

Skeleton Game Development Using Script

최한용, 김귀정*
한북대학교, 건양대학교*

Choi Han-Yong, Kim Gui-Jung*
HanBuk Univ., KonYang Univ.*

요약

엔진을 도입하여 게임을 개발할 때 단지 하부 구조를 구현 하지 않았을 뿐, 엔진 내용을 이해할 정도의 수준이 되어야 한다는 것이다. 그러므로 게임 업계에서는 양질의 게임을 개발하기 위해서 여전히 고급 프로그래머를 필요로 하고 있으며, 이러한 부분은 여전히 개발비 상승에 영향을 미치고 있다. 본 논문에서는 이러한 문제를 해결하기 위해 스크립트를 이용하여 각 조각 부분으로 스켈레톤 게임을 구성할 수 있도록 하였으며, 직접적인 엔진 제어를 줄이는 추상화 작업을 목표로 하기 때문에 추상화된 스크립트 언어를 설계하여 엔진제어 모듈로 변환하도록 하였다.

Abstract

The existing game development way is adopting a direct engine design way. It was hard to learn game development, but it can learn the process that uses a script an abstraction. And it improves productivity and convenience anger because modeling does not do an engine directly and reduce a development cost and can solve a problem of a development human power security. The representative characteristic of this paper was able to change a development environment into an abstract script language with modeling having done an engine in the existing game development environment directly.

I. 서론

게임 시장이 성장하면서 빠른 생명주기에 대응하기 위해 반복적으로 사용되는 모듈의 재사용과 생산성을 높이기 위한 툴 도입을 위한 노력이 게임 개발 업계에서도 끊임없이 이루어지고 있다[1]. 엔진 개발에 소요 비용 대비 효과가 떨어지기 때문에 전문 엔진 개발 업체의 상용 엔진을 사용하는 것이 일반적인 추세가 되고 있다[2]. 따라서 개발기간을 단축시키고 생산성을 높이기 위해서는 기존의 엔진을 도입하여 개발하는 방법이 사용되고 있다. 그러나 엔진을 도입하는 경우 엔진을 직접적으로 제어할 수 있는 정도의 기술 수준이 되지 않으면 개발에 많은 어려움을 겪고

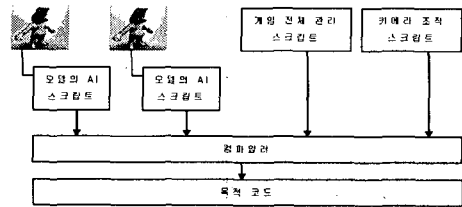
있다. 엔진을 도입하여 게임을 개발하는 경우에도 단지 하부 구조를 구현 하지 않았을 뿐, 엔진 내용을 이해할 정도의 수준이 되어야 하는 문제점을 안고 있다. 그리고 게임 기획자는 엔진의 기술적 정보를 이해하지 못하고 있을 경우 기획의도가 개발환경에서 변형되는 경우가 많고, 기획단계에서부터 기획자와 개발자간 의사소통의 어려움을 겪고 있다. 그러므로 프로그램 이해하는 기획자 수준에서도 스켈레톤 게임을 개발할 수 있는 통합환경이 필요하며, 동일 장르의 게임을 빠른 시간 내에 파생적으로 만들어 낼 수 있는 환경을 제공하여야 한다. 그리고 개발 프로덕트의 최종 품질 및 개발비용 등에 관련하여 사용할 엔

진을 교체한다고 하더라도 개발 환경은 동일하게 가져갈 수 있도록 한다[3][4]. 본 논문에서는 개발에 착수하기 전에 기획자와 개발자간의 의사소통이 가능하며, 엔진단계보다 한 단계 추상화된 수준의 개발환경을 제시하고 있다. 통합개발환경은 엔진의 이해도가 낮아도 간단한 스크립트의 문법정도를 익힘으로써 기획자가 쉽게 게임을 개발할 수 있는 환경을 제공 하도록 하였다[5]. 그리고 직접적으로 하부구조의 엔진 구현 및 엔진의 내용을 잘 이해할 정도의 수준인 고급 개발자가 아니라도 스크립트언어를 이용하여 간단하게 스켈레톤 게임을 개발할 수 있는 통합 환경을 개발하였다.

II. 스크립트언어의 구성

2.1 구성요소

스크립트는 게임 환경에서 프로그래머뿐만 아니라 기획자에 의해서 게임 환경을 컨트롤할 수 있게 도와 줄 수 있다. 게임 로직을 게임 내부에 넣었을 경우에 게임 로직의 작은 변화에도 개발자가 게임로직을 수정해야하는 반면 복잡한 게임 로직을 스크립트의 수정으로 빠르게 게임을 변경할 수 있다. 따라서 본 논문의 스크립트 언어는 그림 1과 같이 게임 전체 관리 부분, 모델의 AI 부분, 카메라 조작 부분의 3가지 스크립트 구성요소를 중심으로 설계하였다. 첫 번째, 게임 전체 관리 부분은 전체적인 게임의 흐름(flow)을 결정하고 게임 제작 전반에 관련한 변수들을 관리하는 부분이다. 게임의 내용을 적재(load)하고 저장(save)하는 기능을 갖는다. 개별 엔진을 유기적으로 통합하고, 각종 데이터를 효율적으로 관리하며, 기획을 충실히 반영하여 콘텐츠를 개발하는 데에 필요한 모듈을 담당한다. 렌더링, 애니메이션 등 엔진의 통합과 게임 객체의 관리를 담당한다. 두 번째, 모델의 AI 부분은 게임에 등장하는 모든 캐릭터의 AI 부분을 조작하는 스크립트 부분이다.

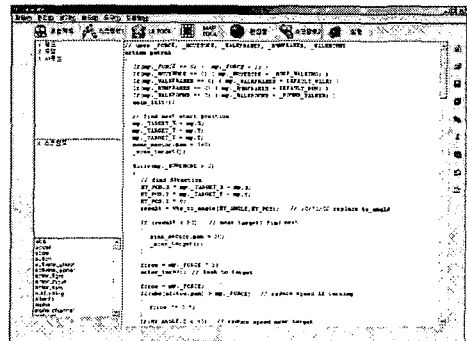


▶▶ 그림 1. 스크립트 언어의 3가지 요소

캐릭터의 AI 부분은 각 캐릭터마다 따로 프로그래밍을 할 수 있도록 UI가 제공되는데, 게임 전체 관리 부분과 연동되어 게임 전체 상황 따라 캐릭터 역시 여러 가지 상황을 만들어 낼 수 있다. 마지막으로, 카메라 조작 부분은 게임 전체와 모델들의 상황을 독자적으로 판단하여 적절한 카메라 뷰(view)를 만들어 내는 스크립트 부분이다

2.2 스크립트 언어의 작성

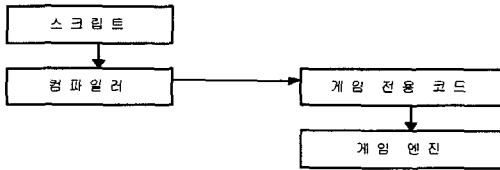
본 연구에서는 스크립트의 문법 학습 시간을 단축하기 위해 C언어의 문법구조를 이용하여 스크립트의 구문을 정의하였다. 스크립트는 그림 2와 같이 두 가지 방법으로 작성할 수 있다. 스크립트는 각 개체(object)의 동작 스크립트를 편집기(editor)로 편집하여 작성되기도 하며, 간단한 마법사(wizard) 형식으로 자동 생성할 수 있다. 그리고 스크립트 편집기는 좌측에 각 컴포넌트 리스트를 관리하고 우측에서 각 컴포넌트의 스크립팅 과정에서 Syntax highlighting 기능을 제공하여, 문법적 오류를 수정할 수 있도록 하였다.



▶▶ 그림 2. 스크립트 편집방법

2.3 스크립트의 추상화

통합 개발 환경을 이용하여 작성되는 스크립트는 컴파일 과정을 거쳐 목적 코드인 게임 전용 코드로 만들어진다. 목적코드에 해당하는 게임전용코드는 직접적으로 엔진을 조작할 수 있도록 하였다. 따라서 그림 3과 같이 스크립트 언어는 사용자의 개발편의성을 고려하여 이 목적코드를 추상화한 것이다. 그러므로 게임엔진을 잘 이해하고 있는 고급 개발자는 게임 엔진 단계에서 직접적으로 제어할 수 있으며, 초급 개발자를 비롯한 기획자는 엔진을 추상화한 개발환경에서 스크립트 언어를 이용하여 게임을 개발할 수 있다. 게임 전용 코드는 게임 엔진을 구동하여 게임에 관련된 실제 작업을 엔진레벨에서 수행하게 된다.



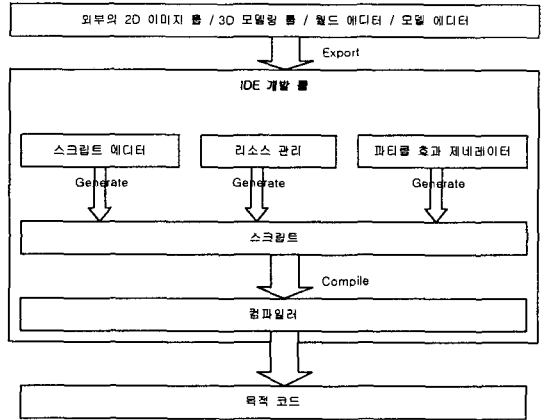
▶▶ 그림 3. 2차 구조의 스크립트 추상화

III. 통합개발환경

3.1 통합개발 환경

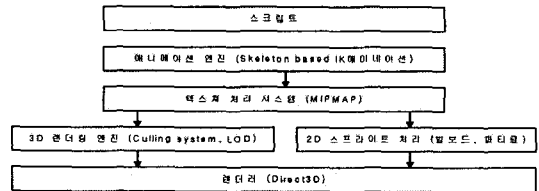
통합 개발 환경은 그림 4와 같이 게임 엔진을 이용하여 게임을 개발하는 기본적인 프로세스를 따른다. 기획에서 나온 스토리보드 내용을 게임 로직에 적용할 수 있도록 통합적인 개발 시스템을 제공하며 복잡한 게임 개발 프로젝트에서 각 컴포넌트의 제작 관리를 쉽게 하기 위해 개발 도구 내에서 스크립트 에디터와 리소스 관리 및 파티클을 비롯한 게임 정보를 구성할 수 있도록 하였다. 그리고 스크립트를 이용하여 스켈레톤 게임을 구성할 수 있으므로, 게임개발 작업에 착수하기 전에 기획단계에서 구성된 시나리오에 의해 게임이 어떻게 보여질 수 있을지 데모게임을 작성하여 시나리오를 검토할 수 있으며 기획자와

개발자간의 의사교환 방법으로 이용할 수 있다.



▶▶ 그림 4. 개발 프로세스

스크립트를 구동하기 위한 비주얼 머신은 Microsoft DirectX 기반의 비주얼C++ 제작된 3D 게임 엔진이며, 그림 5와 같이 3D 그래픽 라이브러리를 중심으로 설계하였다.

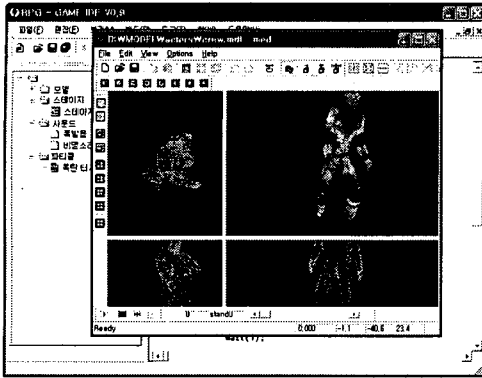


▶▶ 그림 5. 게임 엔진 그래픽 부분

애니메이션 엔진은 게임 내에서 캐릭터의 애니메이션 부분을 구현하는 것으로 모션 캡처링된 게임 애니메이션 데이터를 재생할 수 있는 스켈레톤 기반의 애니메이션을 지원한다. 3D MAX와 같은 외부 3D 모델링 프로그램으로부터 메쉬(mesh), 텍스처(texture), 애니메이션(animation) 등을 자체 포맷으로 외포(export)하여 엔진에서 사용할 수 있다.

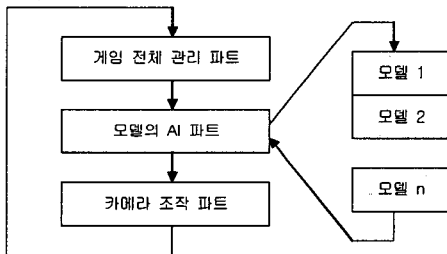
3.2 리소스 관리 및 게임 구성

IDE 툴의 좌측 창은 리소스 관리 창으로서 게임용 리소스 편집을 위한 외부 에디터와 연결되어 있다.



▶▶ 그림 6. 모델 리소스 관리

따라서 리소스와 IDE의 개발과정은 동적링크에 의해 리소스를 관리할 수 있다. 그림 6은 리소스에서 제공하고 있는 정보를 스크립트로 조합하여 모델을 생성한 예를 보이고 있다.



▶▶ 그림 7. 게임 구성

IDE 툴은 각 모델에 대해 고유의 스크립트를 독립적으로 제공한다. 각 모델에 대해 게임 루프를 가질 수 있으며, 모델은 독자적으로 편집이 가능하다. 스크립트에 의해 관리되는 모델은 컴파일 되는 목적 코드 상에서 게임 루프를 그림 7과 같이 구성하고 있으며, 이 때 모델의 AI 파트는 모델의 숫자가 늘어나는 만큼 리스트로 추가되어 관리된다.

IV. 결론

일반적으로 고가의 게임 엔진(수천만원~수억원대)을 도입한다면, 생산성과 개발 편의성은 크게 향상시

킬 수 있으나, 엔진 자체의 라이선스 비용이 수억원에 달하는 등 개발비용에 큰 부담을 준다는 단점이 있다. 그리고 저가의 게임 엔진의 경우(수십만원대~수백만원대)에는 통합적으로 개발을 할 수 있는 IDE 환경이 없으며, 자체 스크립트 편집도구조차 없는 경우가 대부분이라 일반 개발 환경에 비해 편의성이 열악해진다는 단점이 있다. 따라서 기존의 개발방법 중 엔진을 도입한 개발방법에서 나타나는 개발자의 숙련도 문제 그리고 기획자와 개발자간의 의사교환 문제가 있었다. 본 논문에서는 게임개발에서 소요되는 노력 중 엔진개발이후 엔진을 활용한 게임개발 교육에 소요되는 노력을 줄이고, 엔진도입 시점에서 유사게임을 빠르게 개발하려는데 목적이 있으며 또한 개발자와 기획자간의 의사교환을 위해 개발 이전에 엔진을 이용하여 스켈레톤 게임을 기획해 볼 수 있다. 스크립트를 이용하여 엔진을 제어할 경우 개발자 요구수준은 기획자 수준에서 스켈레톤 게임 개발이 가능하고, 엔진용 코드개발과정에서 스크립트방식은 상위 스크립트만 이해하면 엔진용 코드는 컴파일러를 통해서 생성할 수 있다. 장래에 따른 기본적인 프레임 구조를 가지고 있으므로 개발기간이 단축된다. 향후 연구과제로는 그래픽 기반의 기획환경 구축이 필요하며, 엔진의 모듈을 확장할 때 스크립트의 자동 수정이 요구된다.

■ 참고문헌 ■

- [1] 가상현실연구부, 온라인 3D 게임엔진 표준화 연구, 최종연구보고서, 2001.11, 한국전자통신연구원.
- [2] Michael Lewis, Jeffrey Jacobson, "Game Engines in Scientific Research," Communications of the ACM, Jan. 2002, Vol 45. No. 2002.
- [3] Steve Rabin, AI Game Programming Wisdom, Charles Rivers Media, 2002.
- [4] John E.Laird, "Using a Computer Game to Develop Advanced AI," IEEE Computer, July 2001, pp.70-75, 2001.
- [5] Steven Woodcock, "Game AI: the state of the Industry," Game Developer, Aug. 2000, pp24-28, 2000.