

유니티 3D 엔진의 효율적인 이용 방법

김수균*, 송기섭[○], 이희범*, 강지훈*, 임광혁**, 김석훈***

[○]배재대학교 게임공학과

**배재대학교 전자상거래학과

***수원여자대학교 모바일미디어과

e-mail: {nicesk}@gmail.com

An Efficient Use Method for Unity 3D Engine

SooKyun Kim*, Giseob Song[○], Hee-Bum Lee*, Gi-Hun Kang*

KwangHyuk Im**, SeokHun Kim***

[○]Dept. of Game Engineering, Paichai University

**Dept. of Electronic Commerce, Paichai University

***Dept. of Mobile Media, Suwon Women's College

● 요약 ●

최근 게임 시장은 모바일 분야의 급격한 성장과 더불어 많은 지각 변화가 일어나고 있다. 특히 많은 게임 엔진 제작사에서는 멀티 플랫폼을 지원하는 것이 큰 이슈가 되고 있다. 또한 많은 메이저 게임 개발사들도 하나의 플랫폼을 지원하는 게임이 아니라, 다양한 플랫폼을 지원하는 게임을 제작하기에 이르렀다. 이런 멀티 플랫폼이 주목받기 시작하면서 유니티 엔진 또한 주목받게 되었다. 본 논문에서는 멀티 플랫폼을 지원하는 유니티 게임 엔진을 이용하여 게임을 개발할 때 필요한 객체지향 설계방향에 대해 설명한다.

키워드: 유니티 엔진(unity engine), 멀티 플랫폼(multi-plaform), 게임 엔진(game engine)

I. 서론

다양한 성능을 가진 게임엔진들 통해 메이저 게임 개발사들은 고성능의 게임을 개발하고 있다. 그러나 현재는 스마트 폰 분야의 급격한 성장으로 인해 멀티 플랫폼을 지원하는 게임을 개발하는 것이 하나의 큰 이슈가 되고 있다. 특히 게임 엔진 제작사들도 현재는 지원하려고 노력 중이다. 이러한 가운데 유니티 엔진[1, 2, 3]의 가장 큰 특징 중 하나는 PC, iOS, Android, Web, Flash 등의 멀티플랫폼을 지원한다는 점이다.

개발이 끝난 게임은 그 게임이 서비스 될 플랫폼에 맞도록 필요한 옵션을 설정하고 빌드하면 동일한 게임이 서로 다른 코드로 컴파일 되어 각각의 플랫폼에 맞는 최종 배포 판이 생성된다. 이러한 강력한 기능으로 인해 유니티 게임엔진은 게임 개발자들에게 많은 호응을 얻고 있다.

본 논문에서는 유니티를 이용하여 게임을 개발할 때 필요한 객체지향 설계방향에 대해 간단히 설명한다.

II. 본론

유니티는 객체 지향 설계(그림 1)와 데이터 주도적 설계(그림 2)를 근본적으로 지니고 있는 도구이다. 객체 지향 설계란 그림 1과 같이 모든 세상에 존재하는 물체를 객체(Object)로 표현할 수 있으며, 모든 설계를 해당 대상을 중심으로 설계하는 것을 말한다. 그리고, 각 객체에는 메소드(Method)와 속성(properties)을 가지고 있다.

데이터 주도적 설계란 모든 처리를 주어지는 작은 요소들로 쪼개고, 각 요소에 대입하는 데이터만으로 다양한 처리가 가능한 최적의 기능만으로 설계하는 것이다. 예를 들면, 그림 2와 같이 코드의 유지보수가 효율적인 장점을 가지고 있다.

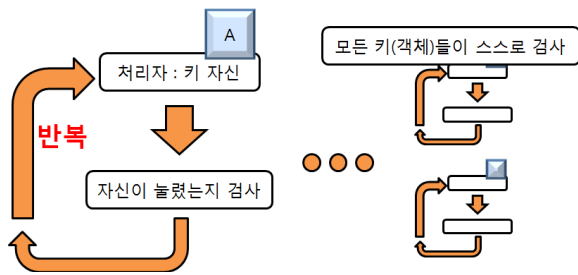


그림 1. 객체 지향 설계의 예
Figure 1. Example of Object Oriented Design

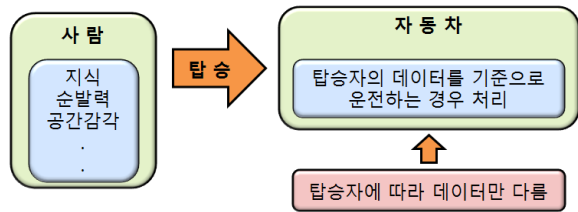


그림 2. 데이터 주도적 설계의 예
Figure 2. Example of Data Driven Design

이와 같이 객체의 개념만 잘 이해하고 있으면 초보 프로그래머라도 간단한 문법을 이용하여 쉽게 설계할 수 있다. 또한, 객체의 독립성, 코드의 최소화와 함께 데이터에 의해 다양하게 구동되어질 수 있는 코드를 제작하여 유지 보수에도 효과적이다. 이와 같이 객체의 개념만 잘 이해하고 있으면 초보 프로그래머라도 간단한 문법을 이용하여 쉽게 설계할 수 있다. 또한, 객체의 독립성, 코드의 최소화와 함께 데이터에 의해 다양하게 구동되어질 수 있는 코드를 제작하여 유지 보수에도 효과적이다.

본 논문에서 다루는 유니티 개발 언어로는 자바스크립트와 C#, Boo스크립트 중에 선택하여 프로그램에서는 스크립트 형태로 작성되고 기본적인 문법만 사용하므로 프로그래밍 언어에 대한 깊은 지식이 없어도 개발이 가능하다. 유니티의 자바스크립트의 경우에는 유니티에 특화되어있어 기존의 자바스크립트보다 훨씬 빠르게 처리된다[3, 4]. 본 연구에서는 자바스크립트와 C#을 연동하여 사

용하게 되었다. 서로 다른 언어이기 때문에 충돌이 발생할 우려가 있는 단점을 가지고 있지만, 본 연구의 해당 부분에서는 충돌 우려가 상당히 적고, 유니티에서는 패키지 단위로 처리되기 때문에 이와 같은 상황이 발생하지 않기 위해서는 데이터 처리를 패키지 단위가 아닌 오브젝트와 코드를 하나하나 직접 새로 작성해야만 한다.

일반적으로 게임 개발은 구상-설계-개발-테스트의 순서로 장기간에 걸쳐 진행된다. 하지만 유니티 엔진의 특성상 위의 과정을 주 단위의 짧은 주기로 반복해 나갈 수 있어 비교적 빠르게 개발할 수 있다.

IV. 결론

현재 게임 시장은 모바일 분야의 급격한 성장과 더불어 많은 지각 변화가 일어나고 있다. 이런 지각 변동은 게임 개발자들이 멀티 플랫폼의 지원 여부를 매우 중요하게 여기게 되었고, 유니티 엔진은 다양한 플랫폼을 지원하며, 개발자들의 주목받게 되었다.

본 논문에서는 멀티 플랫폼을 지원하는 유니티 게임 엔진을 이용하여 게임을 개발할 때 필요한 객체지향 설계방향에 대해 설명한다.

참고문헌

- [1] Will Goldstone, "Unity 3.x Game Development Essentials," IPackt Publishing, 2011.
- [2] Michelle Menard, "Game Development with Unity," Course Technology PTR; 1 edition. 2011.
- [3] Sue Blackman, "Beginning 3D Game Development with Unity: All-in-one, multi-platform game development," Apress; 1 edition, May 25, 2011.
- [4] Ryan Henson Creighton, "Unity 3D Game Development by Example Beginner's Guide" Packt Publishing, September 24, 2010.