

XNA 게임개발 플랫폼

(사)한국게임개발자협회 부회장/동국대학교 게임멀티미디어공학과 겸임교수 | 이승훈(shlee@kgda.or.kr)

1. XNA 소개

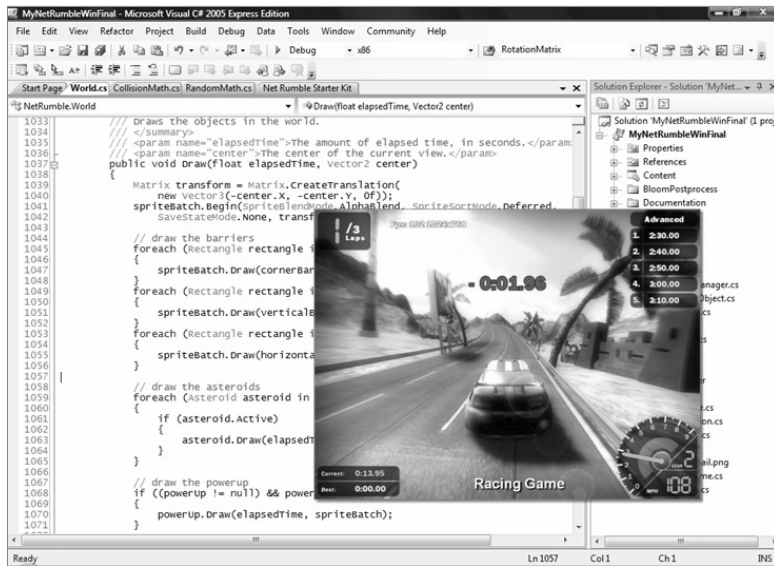
비약적인 발전과 함께 성장하고 있는 컴퓨터 게임은 언제부터인가 하나의 문화산업으로 자리를 잡아가고 있다. 초창기의 컴퓨터 게임은 프로그래머가 어셈블리 언어를 통해 개발을 하기 시작했고, 요즘에는 C/C++ 언어를 통해서 개발을 하고 있다. 그러나 C/C++ 언어를 통해 게임을 개발하는 것은 결코 쉬운 일은 아니다. 목표하는 플랫폼에 대한 그래픽, 물리 계산, 사운드, 컨트롤러 등의 각종 엔진을 개발해야 하고, 메모리 관리 및 로직에 대한 부분도 신경을 써야 하기 때문에 개발 생산성이 낮기 때문이다.

이러한 문제를 해결하고자 마이크로소프트는 2004년 GDC(Game Developers Conference)에서 XNA라는 새로운 게임 개발 환경을 제안하였다. XNA는 사용하기 쉬운 개발환경이다. C/C++ 언어 대신 C# 언어를 사용하기 때문에 C# 언어가 가지는 빠른 생산 능력을 활용할 수 있다. 또한 플랫폼이 아닌 게임 개발 자체에 초점을 맞추고 있기 때문에 누구라도 간단한 게임을 쉽게 만들 수 있다는 장점이 있다. 다음으로 크로스 플랫폼 환경을 지향한다. 하나의 게임 소스 코드를 여러 플랫폼에서 사용할 수 있다는 의미이다. 2.0 버전 기준으로 Windows와 XBOX 360을 지원하고 있으면, 3.0 버전에서는 아이팟과 같은 휴대용 기기인 ZUNE까지 지원을 한다.

2. XNA 게임 개발 예시

위에서 언급했듯이 XNA는 닷넷 프레임워크를 기반으로 하고, C# 언어를 사용하기 때문에 게임개발자들은 더 이상 메모리 누수 및 생산성에 대한 부담을 덜 수 있다.

(그림1)은 자동차 레이싱 게임으로 게임의 완성도 및 게임자체의 재미 요소 들을 볼 때 상용게임과 별 차이를 느끼지 못할 정도인데, 파트타이머를 포함한 4명의 개발자가 6주간 개발한 게임이다.



< 그림 1 > XNA Racing Game

(그림2)는 XNA Field라는 3D 비행 게임으로 기존의 그래픽 데이터를 활용하기는 했지만, 1명의 개발자가 4주간 개발한 게임이다.



< 그림 2 > XNA Field

두 개의 XNA 게임 예시를 보면 짧은 개발 기간에 비해 완성도 높은 게임들이 만들어 졌음을 알 수 있다. 이는 XNA가 게임 개발의 생산성 측면에서 상당한 이점을 제공한다는 것을 의미한다고 할 수 있다.

XNA가 이렇게 게임 개발의 생산성 및 완성도를 높여줄 수 있는 이유는 바로 Framework 구조를 하고 있기 때문이다. 기존 DirectX의 요소들을 모두 포함하고 있지만, Framework 구조를 하고 있기 때문에, 개발자 입장에서는 보다 편리하게 기능들을 사용할 수 있는 것이다.

3. XNA 개발 환경

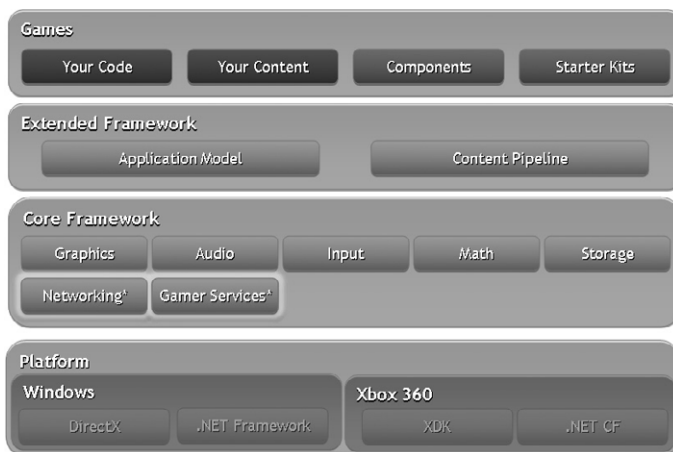
(그림3)은 XNA 2.0 버전 기준으로 XNA의 기본적인 구조를 보여주고 있다. XNA는 기본적으로 윈도우와 XBOX상에서 동작하며, 닷넷 프레임워크를 필요로 한다. 이러한 구조를 가지고 있기 때문에, 단일 개발 환경을 통해 거의 동일한 코딩을 통해 크로스 플랫폼 게임을 개발할 수 있는 것이다.



〈 그림 3 〉 XNA Overview

(그림4)는 XNA Framework Layer 구조를 보여준다. XNA Framework은 XNA 개발환경 구성하는 것으로 마이크로소프트의 XNA Game Studio 통합 환경을 통해서 개발하게 되며, 게임 개발을 위한 기본 템플릿과 많은 라이브러리, 도움말 등을 지원하고 있다.

XNA Game Studio 개발환경은 Visual Studio 2005 C# Express를 기반으로 하고 있다. 그 아래 XNA Framework가 동작하며, 이 XNA Framework는 .Net Framework와 .NET Compact Framework를 기반으로 동작하고 이 두 Framework는 윈도우와 XBOX 360에서 작동하도록 만들어진 Framework이다.



〈 그림 4 〉 XNA Framework Layer

XNA Framework은 게임 개발을 위한 각종 기능들을 Core Framework을 통해 제공한다. 그래픽은 Direct3D9 기반으로 만들어졌으며 텍스처, 이펙트, 셰이더 등을 지원하고 2D 그래픽이나 파티클 효과를 구현할 수도 있다. 또한 XACT 크로스 플랫폼 오디오 Tool을 통해서 각종 사운드 효과를 제공할 수 있는 기능도 제공한다. 이 밖에도 입력장치는 키보드와 마우스 및 XBOX 306 컨트롤러를 지원하며, 수치연산과 저장장치 제어 기능도 제공한다.

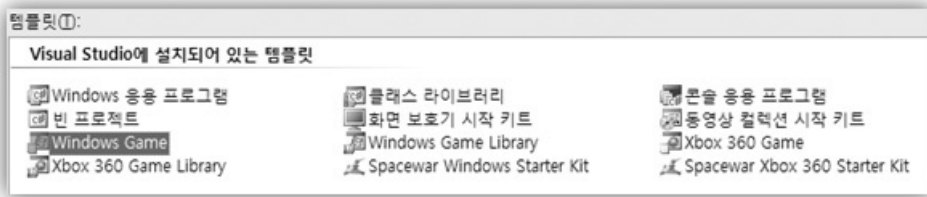
표1은 XNA Framework를 구성하는 각 항목들에 대한 설명이다. 기존 DirectX로 게임을 개발할 때 사용한 많은 부분들을 모듈화 하여 프레임워크별로 나누고 분류하였다.

항 목	내 용
Your Code	실제 개발자가 만들어야 하는 코드 부분이다.
Your Content	실제 개발자가 만들어야 하는 콘텐츠(그림, 음악 등등) 부분이다.
Component	XNA 및 XNA Game Creato에서 지원하는 컴포넌트들이다.
Starter Kits	MS에서 XNA 시작 킷으로 지원하는 툴, 예제, 리소스 들의 집합이다.
Application Model	XNA 프레임워크의 Game 클래스가 제공하는 큰 툴이다. 크게 Update, Draw, Initialize가 있다. 각각의 게임에 맞게 오버라이드하여 사용하면 된다.
Content Pipeline	게임에서 사용하는 자원들을 각 파일 포맷에 따라 관리하고 유지하는데 사용되는 모듈이다.
Graphics	XNA 프레임워크 Graphics 라이브러리는 Low-Level 자원을 로딩하고 렌더링 하는데 사용된다.
Audio	Microsoft.Xna.Framework.Audio 네임스페이스는 오디오 파일들을 재생 하는 함수와 클래스를 제공한다.
Input	Microsoft.Xna.Framework.Input 네임스페이스는 키보드, 마우스, Xbox 360 컨트롤러 장치를 사용하는 함수와 클래스를 제공한다.
Math	벡터나 매트릭스를 다루는 함수와 클래스이다.
Storage	Microsoft.Xna.Framework.Storage 네임스페이스는 파일을 읽고 쓰는 것을 담당하는 클래스를 제공한다.
Networking	게임에서 네트워크와 관련된 부분을 담당하는 모듈이다. 이전에는 하위에 Gamer Services가 포함되었으나, 2.0부터는 Gamer Service가 독립되었다.
Gamer Services	Xbox LIVE와 관련된 기능을 담당하는 모듈이다. 이곳에는 LIVE 관련 GUI 등이 포함되어 있다.

< 표 1 > XNA Frame 구성

4. XNA Game Studio 개발 템플릿

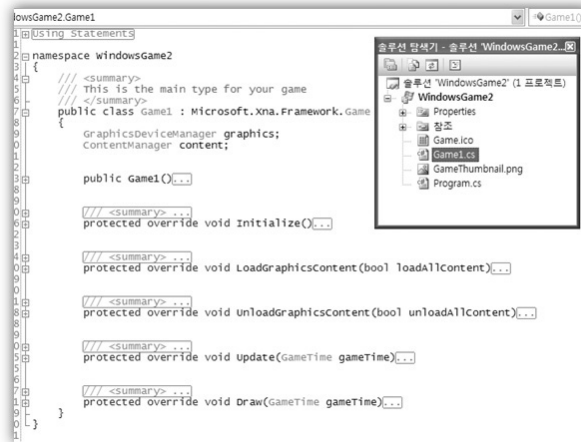
XNA Game Studio의 기본 IDE는 마이크로소프트 Visual Studio와 같은 형태의 템플릿 형태의 개발 환경을 제공한다. XNA 2.0 기준으로 그림5과 같은 형태의 개발 템플릿이 제공되며, 개발자는 몇 번의 마우스 선택만으로 기본적인 게임 구조를 생성할 수 있는 기능을 제공한다.



〈 그림 5 〉 XNA의 개발 템플릿

(그림5)에서 보듯이 XNA 개발 템플릿은 Windows Game, Window Game Library, Xbox 360 Game, XBox 360 Game Library등을 제공한다. 이 중 Windows Game은 윈도우즈 기반의 게임을 만들 때, Xbox 360 Game은 Xbox 360의 게임을 개발할 때 사용한다. Library는 게임에서 사용할 라이브러리를 만드는 개발 템플릿으로, C/C++에서의 LIB, DLL과 유사한 기능을 제공한다.

XNA Game Studio의 개발 템플릿을 통해 게임 구조를 생성하면, Program.cs 파일과 Game1.cs 파일이 생성된다. Program.cs 파일은 Game1.cs 파일을 호출하는데 사용되는 엔트리 포인트가 정의된 곳이고 Program.cs 파일에는 (그림6)와 같이 Initialize, LoadContent, update, Draw 등의 XNA Framework에서 상속받은 코드들이 생성된다. 게임개발자는 이렇게 생성된 코드에 개발하고자 하는 게임의 코드를 작성하면서 게임을 개발하면 된다.



〈 그림 6 〉 Game1.cs 코드 구성

Initialize는 게임이 실행될 때, 단 한번 수행되는 함수로 초기화 함수의 역할을 지원한다. Initialize 함수가 불리고 난 후, Initialize 함수가 종료하기 직전에 LoadContent 함수가 호출된다. LoadContent 함수는 게임에 사용될 리소스(그림, 음악 등)를 부르는 기능을 수행한다.

UnloadContent는 게임에 사용된 리소스(그림, 음악 등)를 메모리에서 해제시키거나 게임의 종료시 사용된다. Update는 게임 타이머(정기적인 이벤트)에 의해 호출되거나, 사용자의 이벤트에 의해 불리는 함수로 게임 로직에 대한 부분이 기술된다. 예를 들면, 입력에 따른 움직임이나 충돌체크 등이 해당된다. 마지막으로 Draw는 표현하고자 하는 이미지나 상태를 화면에 그릴 때마다 호출되는 부분이다.

5. XNA를 활용한 게임 개발의 장단점

XNA는 기존의 DirectX 개발 환경을 대체할 수 있을 만큼 기능적으로는 부족함은 없다. 그러나 기본적으로 C/C++ 언어 대신에 C#을 사용하기 때문에 C/C++ 언어에 익숙한 개발자들에게 있어서는 상당히 불편하게 느껴질 수도 있고, 속도 부분에 있어서도 아직은 느리다. 또한 C/C++ 개발자의 경우 포인터 사용의 문제 및 소멸자를 명시적으로 부르지 못하는 문제점도 있다.

반면 XNA Framework를 통해 제공되는 코드 절감 부분은 상당히 매력적인 부분이다. 또한 XNA 2.0까지는 자료구조의 변경이 없기 때문에, 혼돈이나 케스팅에 대한 위험도 없다고 볼 수 있다. (예, XNA : Rectangle => DX : RECT, D3DXRECT, D3DRect)

다음으로는 멀티 플랫폼 개발 환경 지원인데, 기존 DirectX를 기반으로 게임을 개발할 경우 재사용을 포함한 여러 문제까지 문제점들이 있었지만, XNA의 경우 프로젝트 환경 변화를 통해 간편하게 변환이 가능하기 때문에 강점을 가지고 있다고 할 수 있다.

또한, 생각보다 XNA Framework 개발 환경이 안정적이다. 과거 DirectX의 경우에는 버전에 따라 많은 변경들이 발생했고, 이 부분들이 개발자들에게 부담으로 작용을 했지만, XNA는 기본적으로 Framework 구조이기 때문에, 향후 버전이 바뀌더라도 개발 환경에 크게 영향을 주지는 않을 것이다.

6. XNA 활용 분야 및 전망

XNA는 기존의 DirectX 기능 대부분을 제공하고 있기 때문에, 상당히 매력적인 게임 개발 환경임에는 틀림없다. 그러나 아직까지는 속도를 포함한 여러 가지 개선의 부분이 남아 있는 것도 사실이다.

따라서 현재의 XNA는 아마추어 개발자 및 보드, 라이트 캐주얼 게임 등의 개발에 적합하다고 할 수 있다. 또한 XNA Creator Club을 통해 자신의 게임을 XNA 게임개발자들과 공유할 수 있다.

이 승 훈 (shlee@kgda.or.kr)

- (현) (사)한국게임(shlee@kgda.or.kr)
- (현) 동국대학교 게임멀티미디어공학과 겸임 교수

〈 관심 분야 〉

- 게임 제작 프로세스, 게임 디자인