

XNA기반 게임 개발의 이펙트 효과 알고리즘 설계 및 구현

서정만*, 최창수**

The Design and An Implementation of effective algorithms Effect Based on XNA Game Development Environment

Jeong-Man Seo*, Chang-Soo Choi**

요약

본 논문은 Visual C#을 이용한 프로그램 기법인 XNA 게임 개발에 관한 것이다. 게임에 필요한 화면 설계, 게임 구현 및 실험 부분에서 게임의 실행화면과 기존의 게임의 차별성과 알고리즘을 제시하였으며, XNA기반의 슈팅 게임을 구현하였다. XNA기반 게임 개발환경에서 게임개발의 이펙트 효과 알고리즘 설계 및 구현에 대한 내용을 제안하였다. 게임개발에서 기존의 Directx C++를 사용하여 게임개발을 하는 것은 게임개발자 입장에서 소스코드가 많이 필요하며, 특히 이펙트 효과부분의 처리에서 많은 어려움이 있었다. 그러나 제안한 논문에서는 이러한 어려움을 해소하고, XNA기반에서 게임개발의 가능성을 보였다. 제안한 논문의 우수성을 위해 기존의 게임과의 성능 분석을 하였으며, 게임개발을 위한 설계를 하였고, 게임구현 환경 및 게임 실행 화면을 보였다. 향후 XNA기반 게임 개발에 대한 발전에 기여 할 것으로 기대한다.

▶ Keywords : 게임설계, 슈팅게임, XNA, 게임이펙트

Abstract

In this paper, by using Visual C # XNA game development application relates to a technique. This paper proposes the design and an implement effective algorithms Effect Based on XNA Game Development Environment. Existing in the development of the game using Directx C++ game developers, game development, it is necessary to position a lot of the source code. Particularly effective effect was much difficulty in the processing section. However, to eliminate this difficulty was proposed in the paper. The possibility of developing the game in XNA-based group. For the

•제1저자 : 서정만 공동저자 : 최창수

•투고일 : 2014. 5. 14, 심사일 : 2014. 6. 18, 게재확정일 : 2014. 7. 9.

* 한국복지대학교 게임콘텐츠과 교수 (Dept. of Game Contents, Korea National College of Welfare)

** 청주교도소 직업훈련교사 (vocational training Teacher, Cheongju correction institution)

superiority of the proposed paper and the comparative analysis of existing games were designed for the development of the game was. The future study will be the design of various effects and events which give more immersive game implementation.

▶ Keywords : Game Design, Shooting Game, XNA, Game Effect

I. 서론

게임이라는 용어는 “흥겨우게 뛰다”라는 인도 유러피안 계통의 “ghem”에서 파생된 단어로 “흥겹다”는 정신적으로 재미 또는 즐거움을 느낀다는 뜻과 “뛰다”라는 동작을 나타내는 동사적 의미의 단어가 합성된 용어이다[1][2][3][13]. 문화콘텐츠의 중요성이 증대되면서 영상, 음반, 애니메이션, 게임 등 문화콘텐츠산업은 전 세계적으로 급속하게 성장하고 있다.

최근 들어 온라인 게임은 단순한 하나의 게임이 아닌 현대 사회를 이끌어 나가는 하나의 산업으로 발전하고 있고, 온라인 게임시장이 전체 게임산업의 56.4% 정도 차지하고 있다. 이러한 온라인 게임 산업의 발전은 게임을 활용한 교육의 발전으로 이어지고 있다. 또한 게임을 통해 공동체들과의 정보를 획득, 탐구하고, 서로 의사소통하는 상호작용을 통해 문제를 해결 할 수 있는 온라인 환경을 제공받으며, 학습자들은 역할 및 과제 수행 과정을 통해 자기 주도적으로 의사결정을 하고 자신의 선택에 의해 행동을 수행한다[4].

게임사용 인구가 늘어나면서, 국내외에서도 게임 활용에 관한 연구도 활발하게 진행되고 있다. 게임의 여러 장르 중에서 시뮬레이션게임은 현실 상황을 컴퓨터, 비디오 게임으로 표현한 게임을 말한다. 처음에는 군사 훈련용 게임이었으나, 점차 일반적인 상황을 컴퓨터로 표현함에 따라 그 범위가 넓어졌다[5]. 이것은 게임이 대중 들 속에 넓게 확산되어 있으며, 일반적으로 가지고 있는 게임에 대한 부정적 인식에서 탈피하여 새로운 관점에서 게임을 바라보아야 할 것이다. 특히 우리나라는 온라인 게임과 모바일 게임에 있어 전 세계에서 앞서가고 있는 나라 중 하나이며, 게임 산업은 국가경제 성장 동력으로 대접받고 있다[6].

현재 게임개발에 사용되어 지고 있는 언어는 여러 가지가 있지만, 본 논문에서는 C#을 사용하여 개발하여 구현하였다. 특히 XNA 게임개발 플랫폼인 XNA Game Studio 4.0을

사용하였다.

2장에서는 관련연구인 이펙트 효과, 키넥트 SDK, XNA 게임개발, 몰입현상과 게임요소를 기술하고, 3장에서는 게임의 구조 및 설계, 이미지 설계, 화면 설계, 4장에는 구현 및 실험결과에서 게임구현환경, 게임의 실행화면과 알고리즘, 기존의 게임과 성능분석, 5장에서는 결론을 기술하였다.

II. 관련 연구

2.1 이펙트 효과

애니메이션에서 리얼리티 이펙트(reality effect)는 실사 영상과는 달리 매우 중요하고 특별한 표현 요소다. 모든 것을 인위적으로 창조해 내야 하는 애니메이션은 무엇보다 우선 시각적으로 사실감, 현실감을 창출해 내야하고, 이를 통해 관객들에게 앞으로 벌어질 이야기에 대해 공감대를 이끌어 내는 것이 중요한데, 이를 성취하기 위한 표현 기법을 리얼리티 이펙트(reality effect)라고 정의할 수 있을 것 같다. 관점에 따라서 그 유형을 달리 분류할 수도 있겠지만, 애니메이션 작품들에서 일반적으로 드러난 리얼리티 이펙트의 보편적 유형은 첫째 사실감, 둘째 현실감, 셋째 혼성모방, 넷째 상호 텍스트성 그리고 다섯째 향수감으로 나눌 수 있다[7].

2.2 키넥트 SDK 1.5

2011년 2월에 윈도우 키넥트 SDK 베타를 출시하면서, 마이크로 소프트는 PC에서의 키넥트 개발을 공식적으로 지원하기 시작했지만, 기존 오픈 소스 라이브러리와 비교해 큰 장점이 없었고, 또한 상업용 라이선스를 지원하지 않았기에 많이 사용되지 않았다[11].

하지만, USB가 지원되는 윈도우 키넥트와 상업용 라이선스 지원 및 향상된 기능의 키넥트 SDK 1.5가 출시되면서 키넥트를 다양한 분야에 상업적으로 사용할 수 있게 되었다. 현재 키넥트

SDK가 지원하는 프로그래밍 언어는 C++, C#, 비주얼베이직(Visual Basic)과 비주얼 스튜디오(Visual Studio) 2010을 통하여 개발해야 한다. SDK에서는 샘플 브라우저를 통해 WPF와 DirectX, XNA관련 예제를 제공한다.

2.3 XNA게임개발

XNA는 마이크로소프트가 '크로스플랫폼(cross platform) -쉬운 개발(easy to use)'이란 모토 하에 개발한 차세대 게임 개발 플랫폼 및 기술을 의미한다.

XNA 4.0은 Microsoft가 Microsoft Visual Studio 2010 Express for Windows Phone을 기반으로 개발한 게임 개발 제품으로, 이를 통해 게임 개발자는 프로그래밍 언어로서 강력한 파워와 단순성을 가진 C#을 이용할 수 있다.

XNA Game Studio 4.0에는 게임에 3 차원(3D) 모델, 텍스처, 사운드 및 기타 리소스들을 가져오는 쉽고 유연한 방법을 제공한다. 그리고 Windows Phone 7을 위한 개발을 쉽게 하는 게임 중심적 애플리케이션 프로그래밍 인터페이스(API)를 제공하는 XNA Framework 와 XNA Framework Content Pipeline 이 포함되어 있다.

XNA Framework는 보다 빠른 게임 개발을 위해 Microsoft 가 개발한 프레임워크이다. 사용이 어렵지는 않지만 게임을 개발하려면 어느 정도 숙련된 기술이 필요하다(8).

XNA는 기존의 C/C++이 아닌 C#언어의 사용으로 개발자가 좀더 쉽게 접근하여 개발이 가능하고 개발도구 및 프레임워크는 모두 무료로 제공된다(9).

XNA에서는 게임개발을 위한 각종 기능을 제공한다. 그래픽 부분은 Direct3D 기반으로 만들었으며 각종 모델과 텍스처, 이펙트, 셰이더 등을 지원한다. 또한 SpriteBatch를 이용하고 2D 그래픽이나 파티클 효과를 구현 할 수 있다. 사운드 XACT라는 크로스플랫폼 오디오 저작도구를 이용해 어떤 환경에서도 사운드 및 각종 사운드 효과를 재생할 수 있게 지원한다(10).

2.4 몰입현상과 게임요소

Hoffman & Novak에 따르면 인터넷 이용에 있어서 몰입이란 인터넷을 사용하면서 주관적인 최적경험(Optimal experience)을 하게 될 때 얻어 질수 있는 것으로 정의하였다.

또한 Webster, Trevino & Ryan 의 연구에 의하면 몰입 상태는 4가지 차원으로 특징을 지워질 수 있는데, 사용자가 컴퓨터와 상호작용에 대해 통제감을 지각하고, 사용자가 스스로 컴퓨터와의 상호작용에 자신의 주의를 집중되어야 한다는 것을 지각하며, 상호작용을 하는 동안 사용자의 호기심이 고

양되고, 사용자가 컴퓨터와 상호작용 그 자체를 흥미 있어 하는 정보를 의미한다(3).

사람들이 게임을 하는 이유는 게임에는 몰입과 상호작용이 있는데, 교육을 하는 데 있어서 게임이라는 요소를 도입한 이유는 다음과 같다(11).

첫째, 컴퓨터 게임이 가진 재미와 몰입의 요소를 학습에 결합함으로써 놀라운 교육적 효과를 기대할 수 있다. 학습에 게임의 요소를 결합하면서 지루하지 않고 즐겁고 재미있게 학습 할 수 있다.

둘째, 컴퓨터 게임이 가진 가장 큰 특징은 상호작용성이며, 게임의 진행은 단순히 '보는(to see)'것이 아니라 '체험(to do)'을 통해 이루어진다(12).

2.5 C++를 이용한 게임개발의 어려움

게임 개발자가 게임을 개발할 때에 여러 가지 컴퓨터 언어를 사용하고, 도구를 사용하지만, 많은 게임개발자들은 Visual Studio C++언어를 사용하여 게임을 개발한다. 아울러 DirectX의 라이브러리를 사용하여 여러 가지 기능과 그래픽 처리, 입출력 처리, 충돌처리, 이펙트 효과 등을 처리한다. 그러나 경험이 많은 게임개발자는 게임을 개발하는데 익숙하여 문제가 없겠으나, 대학에서 공부하고 있는 학생들 중에 전공이 게임 관련학과 학생들이나 초기의 게임개발자들에게는 DirectX 라이브러리를 사용하여 개발하는 데는 많은 시간과 구현하는데 어려움이 따른다. 많은 함수나 기능을 익혀야 하고, 특히 그래픽 처리나, 사운드 처리, 파일 입출력 처리 등에 많은 어려움을 겪게 된다. 이와 같은 어려움을 극복하고, 보다 쉬운 게임개발을 하기 위하여 XNA를 사용하여 게임개발을 제안하고자 한다.

III. 게임 설계

3.1. 게임 구조 및 설계

게임 개발을 위하여, NET Framework와 XNA Framework를 기반으로 설계하였다. 그래픽, 오디오, 입력과 출력 등과 같은 프로그램 구현은 XNA Framework을 통하여 처리하였고, C#언어를 사용하여 구현하였다. 게임을 실행하면, 게임에 필요한 오프닝화면과 게임 로딩하는 화면을 보여준다. 게임이 종료하게 되면 "Game Over"라고 하는 이미지를 화면에 보여지게 된다.

3.2. 이미지 설계

이펙트 효과로 사용할 이미지인 유도탄 이미지와 게임 실행 도중 게임의 현재 사용하는 파워를 나타내는 HP 이미지를 디자인 한다. 아울러 실행도중 유도탄 연기 이미지가 필요하여, 이에 필요한 이미지를 디자인 한다. [그림 1]과 [그림 2]는 유도탄 이미지, [그림 3]은 파워이미지, [그림 4]는 게임에서 사용되어 질 유도탄 연기 이미지를 설계하였다. [그림 5]는 유도탄에 관련된 리소스(Resource) 처리에 대한 설계를 나타낸 것이다.

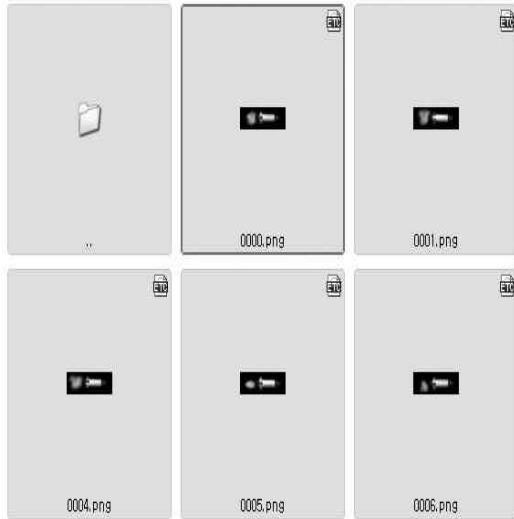


그림 1. 유도탄 이미지 I
Fig 1. Smoke Missile Image I

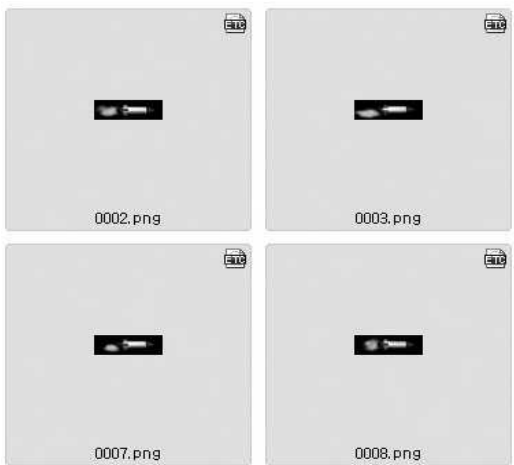


그림 2 유도탄 이미지2
Fig 2. Smoke Missile Image II

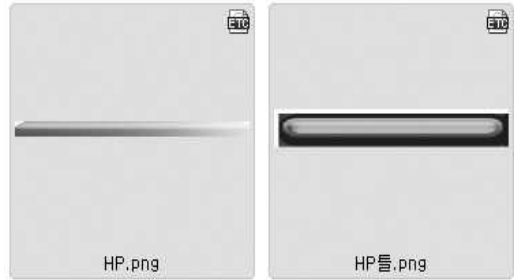


그림 3. HP 이미지
Fig 3. HP Image



그림 4. 유도탄 연기 이미지
Fig 4. Smoke Missile Image

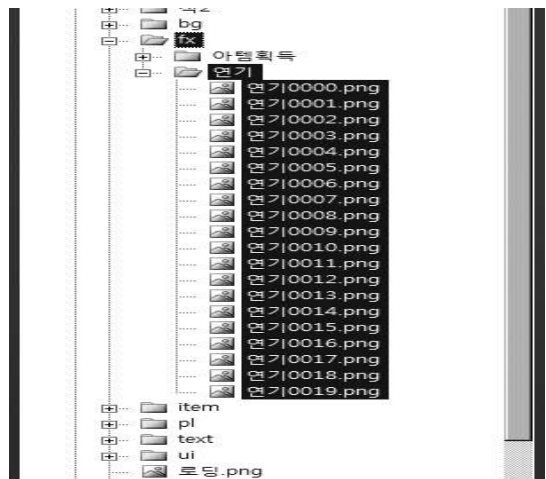


그림 5. 유도탄 연기 리소스
Fig 5. Missile smoke Resource

3.3 화면 설계

슈팅게임에서 화면에 등장하는 주인공과 적기, 아군 총알, 적기 총알 있어야 하고, 적기와 아군기가 충돌하였거나, 적기의 총알이 아군기에 명중하였을 경우 충돌처리, 적기의 총알이 아군기를 명중하였을 경우 충돌처리 등을 고려하여야 한다.

게임을 시작할 때 이미지를 로드하는 시간 동안 'Loading'이라고 글자가 출력되도록 하여 게임 시작화면을 나타내도록 하여 사용자로 하여금 기다리는 지루함을 없앨 수 있도록 설계한다.

게임 시작 또는 종료 시 현재 스테이지를 알려거나 게임 종료를 알려주는 텍스트 배너는 윈도우 폰트를 사용하지 않고 포도삽 등에서 생성한 이미지를 사용하였다.

배너로 사용할 텍스처(이미지)를 로딩하여 특정 시점에 서서히 나타나고, 서서히 사라지도록 하였다.

게임의 종료화면에서 'Game Over'라는 내용이 출력되도록 한다. 아울러 종료될 때까지 획득한 점수를 화면에 출력하도록 디자인하였다.

표 4. 게임화면 객체 설계
Table 4. Game Screen Object Design

| 객체들 | 내용 | 비고 |
|---------|-----------------------|---------|
| Object1 | 도시 배경 | 이미지 |
| Object2 | 한국지도형태 | 이미지 |
| 텍스트1 | Game Over | 배너 |
| 텍스트2 | Mission Clear | 배너 |
| 텍스트3 | PRESS ENTER BUTTON | 영문자(18) |
| Object3 | 유도탄 | 이미지 |
| Object4 | 유도탄 연기 | 이미지 |
| Object5 | 아이템 | 이미지 |

시작화면의 객체들은 [표 4]에서처럼 object 5개, 텍스트 3개로서 이미지와 텍스트 배너와 문자를 넣도록 한다. "PRESS ENTER BUTTON"을 누르면 게임이 시작되도록 설계한다.

게임 배경화면은 [그림 6]에서 기본적으로 도시의 상공에서 내려다 본 내용을 디자인한다. 그러므로 구름이나 도로위의 윤곽 등이 나타나도록 한다.



그림 6. 배경화면 설계
Fig 6. Background Screen Design

게임 실행 화면에서 아군기를 전투기, 또한 적기 1대의 전투기와 적기의 보스로서 디자인 배치하고, 점수와 HP가 출력되도록 하여 [그림 7]에서처럼 설계한다. 또한 유도탄이 발사되면 유도탄의 구름이미지도 나타나도록 설계한다. 게임 실행 시에 아이템이 등장하도록 설계한다.

왼쪽 상단에 점수와 플레이어의 HP를 표시하는 사용자인터페이스를 출력하는 것이다.

즉 여기서 다루는 기능은 숫자 폰트를 이용해 플레이어 점수를 출력하는 것과 플레이어 HP를 게이지(gauge) 형태로 출력한다. 처음에는 적기가 1대가 등장하다가 점차 시간이 흐르면서 적기가 추가되고, 보스가 등장하여 게임의 재미 요소와 속도감을 느낄 수 있도록 하였다.

실행화면에서 적기 2개와 보스1개, 아군기 1개를 배치하여 실행되도록 하며, 적기를 명중할 때 마다 점수가 가산되어 출력되도록 하였다.

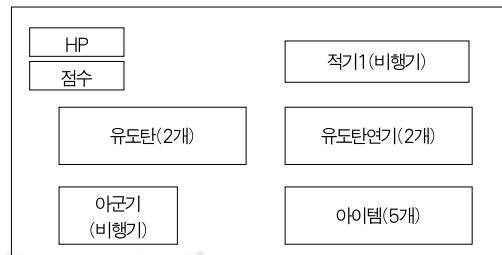


그림 7. 실행화면 배치 설계
Fig 7. Execution Screen layout design

종료화면에 대한 객체 디자인은 [표 5]에서처럼 object 1개, 텍스트 2개로서, "Game Over" 라는 글자가 출력되도록 설계하였다.

표 5. 종료화면 객체 설계

Table 5. End Screen Object Design

| 객체들 | 내용 | 비고 |
|--------|---------------|----|
| Object | 배경화면 | 그림 |
| 텍스트1 | Game Over | 배너 |
| 텍스트2 | Mission Clear | 배너 |

3.4 메뉴설계

메뉴는 사용자의 편리성을 높이기 위하여 화면에서 사용자의 버튼을 만들고, 아울러 윈도우 메뉴도구에 메뉴를 만들어서 사용자의 편리성을 제공하도록 한다. 아울러 사용자의 메뉴 버튼과 단축키를 제공하여 사용자가 선택하여 사용자의 선호하는 메뉴 방식을 선택하도록 설계하였다. 게임의 흥미를 높이기 위해 배경음악과 이벤트 음악을 설정하였다.

IV. 구현 및 실행결과

설계된 게임에 대하여 구현하였고, 그 결과를 보여 주고 있다.

4.1 게임 구현 환경

구현환경으로는 [표 6]에서처럼 펜티엄 4 PC, 메모리 4G 바이트, 게임을 개발하는 언어로는 Microsoft Visual Studio 10.0의 C#을 사용하였고, 캐릭터 디자인은 포토샵을 이용하였고, 운영체제로는 윈도우 7을 사용하였다.

표 6. 구현 환경

Table 6. Implementation Environment

| 사용언어 및 도구 | 버전 및 용량 |
|-----------|---------------------------------|
| 프로그래밍언어 | Microsoft Visual Studio 10.0 C# |
| 사용컴퓨터 | PC 펜티엄 4 |
| 메모리 | 4G Byte |
| 캐릭터디자인 도구 | 포토샵 |
| 운영체제 | 윈도우 7 |

4.2 게임 실행화면

[그림 8]에서처럼 유도탄을 발사하면, 유도탄 연기 궤적을 그림으로 출력하고, 유도탄이 발사되어 출력하게 된다.

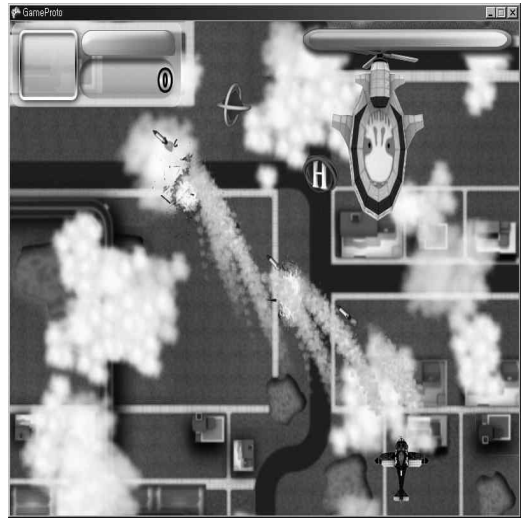


그림 8 유도탄 연기 궤적
Fig 8. Smoke missile trajectory

[그림 9]는 게임실행에서 유도탄이 발사되는 화면을 출력하고 있다.



그림 9. 유도탄 발사 실행화면
Fig 9. Execution screen of Missile launch

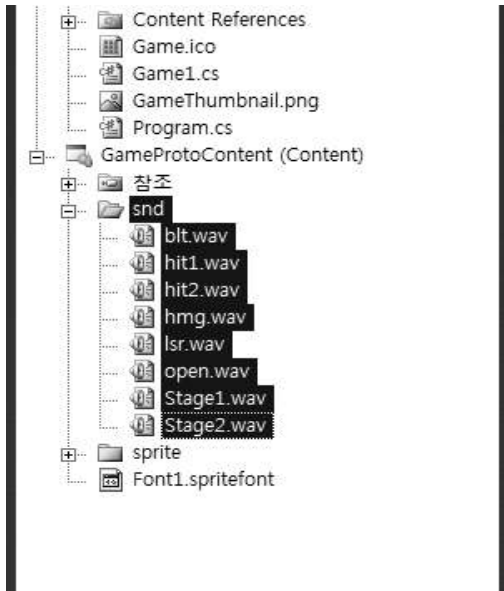


그림 10 음악파일 리소스 삽입
Fig 10. Resource insertion of Sound File

XNA에서 이미지와 마찬가지로 사운드 출력은 콘텐츠 파이프라인을 통해 이루어진다. 솔루션 탐색기에서 [그림 10]과 같이 콘텐츠 폴더에 등록을 시키고, 등록된 사운드를 로딩하는 식으로 사운드 출력이 이루어진다.

4.3 알고리즘

[표 7]은 플레이어가 발사하는 총알에 따라 다른 효과음이 발생하도록 한다. 적의 총알에 맞았을 때의 효과음을 발생시키도록 한다.

표 7. 효과음 출력 소스
Table 7. Effective Sound Output Source

```
public override void DecHP(int damage)
{
    base.DecHP(damage);
    Game1.sndPlay(SND.HIT1);

    public override void OnShow()
    {
        base.OnShow();
        SetSID(SID.OPENING);
        m_iBaseShowCnt = this.m_iShowCnt * = 4;
        ScenePlay.stage = 0;

        Game1.SndPlay(SND.OPEN);
    }
}
```

오프닝 화면이 나타날 때 배경음악을 출력한다. 플레이 씬

이 화면에 나타날 때 플레이 씬의 배경음악을 출력한다. 이때 스테이지에 따라 다른 음악을 출력할 수 있도록 stage 변수값을 더하여 해당 스테이지의 음악이 출력되도록 한다. 정상적으로 enum SND 정의시 STG1, STG2, STG3의 순서대로 정상적으로 음악이 출력된다.

[표 8]은 오프닝 배경 음악을 출력하는 알고리즘을 나타내고 있다. 씬의 배경음악이 출력되고 나서 한번만 음악이 재생되고 나서 종료된다는 문제가 생긴다. 또한 배경음악이 출력되던 도중에 이벤트가 발생하여 이벤트 효과음을 출력해주어야 하는데, 기존에 배경음악이 계속 재생되는 문제가 생긴다.

표 8. 오프닝 배경음악 출력 소스
Table 8. Opening Background Sound Output Source

```
public override void OnShow( )
{
    base.OnShow( );
    bg.Clear(); // 배경 클래스 초기화

    if( ScenePlay.stage == 0 )
        bg.Add

        Game1.SndPlay(SND.OPEN);
}
```

이러한 문제점을 해결하는 알고리즘이 필요하다. [표9]에서 알고리즘을 살펴보면 먼저 충돌음, 발사음 등 일회성 이벤트 사운드 출력에 사용하는 알고리즘인데, 먼저 사운드를 먼저 재생한다. 그리고 씬 배경 음악을 연주할 때 사용하기 위해 함수 씬이 변경될 때 함수를 호출한다.

기존의 배경음악은 출력을 잠시 중지하고, 인자로 넘어온 사운드를 출력한다. 계속하여 다시 기존의 배경 음악을 재생한다. 기존의 배경 음악이 필요한 곳에서 SndPlayBg() 함수를 호출하기만 하면 계속하여 배경음악을 출력할 수 있게 된다.

표 9. 씬 배경음악 출력 알고리즘
Table 9. Algorithm outputs of Background music scene

```
static SoundEffect( ) soundEngine = new SoundEffect(
(int) SND..N_SND);
static SoundEffectInstance( ) soundEngineInstance =
new SoundEffectInstance( (int)SND..N_SND);
static int sndBg = -1;

static public void SndPlay(SND snd)
{
    soundEngineInstance( (int)snd).Play();
}

static public void SndStop(SND snd)
```

```

{
soundEngineInstance( (int)snd).Stop();
}

static public void SndPlayBg(SND snd)
{
if( sndBg >= 0) soundEngineInstance(sndBg).Stop( );
sndBg = (int)snd;
if(soundEngineInstance(sndBg).IsLooped )
soundEngineInstance(sndBg).IsLooped = true;
if( soundEngineInstance(sndBg).State !=
SoundState.Playing )
soundEngineInstance(sndBg).Play( );
}
}
    
```

4.4 실행 화면

비행 슈팅 게임에서 속도감을 느낄 수 있도록 하기 위하여 다중스크롤 기법을 이용하여 구현하였다. 배경 이미지를 이동하여 스크롤 되도록 하였다.



그림 11 게임 시작 화면
Fig. 11 start of the game screen

게임실행화면에서 처음에 시작하면 [그림 11]에서처럼 스크롤 기법을 이용하여 그림의 배경이 스크롤 되는 것으로 구현하였으며, 이펙트 효과 처리에서도 알고리즘을 사용하여 여러 가지의 기능을 구현하였으며, 아군기와 적기의 등장, 구름 효과 등을 볼 수 있다.

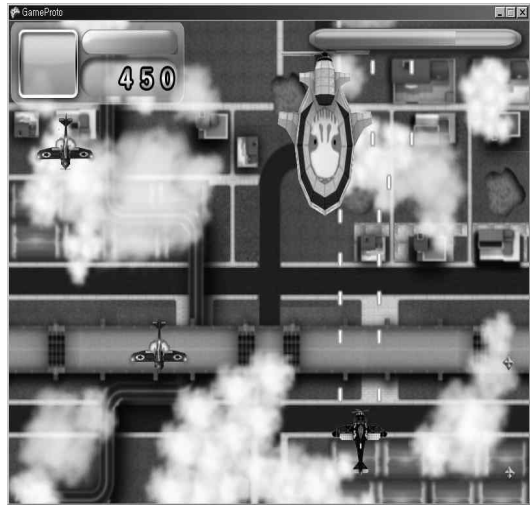


그림 12 게임 실행화면
Fig. 12 Game play screen

[그림 12]는 적기와 아군기 등장하고, 적기를 명중하면 점수가 기록되고, 총알 발사되는 효과와 같은 다양한 이펙트 효과 처리, 사운드 효과 등을 화면에서 알 수 있으며, 적기2개와 보스 1개, 아군기 1개가 실행화면에서 나타나고 있음을 알 수 있다.

4.5 게임의 성능분석

표 10. 기존 게임과 성능 비교분석
Table 10. Comparison and analysis of existing games

| 게임의 기능 | 프로그래밍 용이성 | 하드웨어 접근의 용이성 | 코드의 양 | 애니메이션 기법 | 게임의 속도감 | 이미지 로드 용이성 | 사운드 알고리즘 처리 | 이펙트 효과 용이성 |
|----------------------|-----------|--------------|-------|----------|---------|------------|-------------|------------|
| C++ DirectX 게임 | 중 | 중 | 100% | 상 | 상 | 중 | 상 | 중 |
| XNA 제안 게임 | 상 | 상 | 30% | 상 | 중 | 상 | 상 | 상 |

기존의 게임들은 DirectX를 이용하여 visual c++로 구현하였을 경우, 비슷한 기능의 게임을 제작할 때 프로그램의 코드의 양이 상당히 많이 필요하지만, C#을 이용한 XNA 기반의 게임 구현에서는 코드의 양이 70% 정도 감소하였으며, 하드웨어 접근의 용이성 부분에서 XNA를 사용하면 입력장치나 오디오, 그래픽 장치의 접근이 단 소스 몇 줄로 접근을 쉽

게 할 수 있다. 애니메이션 기법에서도 제한한 게임이 더 좋은 점수를 부여 받았음을 보여 주고 있다. 아울러 프로그램 개발자의 입장에서 본 게임개발의 어려움은 상대적으로 C#을 이용하여 게임개발이 더 쉽게 접근할 수 있음을 보였고, 게임 개발에서 이미지를 로드하는 부분의 코드처리에서도 기존의 DirectX를 사용한 게임소스코드는 많은 부분이 필요하지만 제한한 게임에서는 단지 리소스부분에 필요한 이미지를 로드 하여 줌으로써 게임개발을 쉽게 접근 할 수 있다는 것을 보여 주었다. 본 논문에서 구현한 게임은 기존의 게임 제작에서보다 효율적인 차별성을 [표 10]에서 보여주고 있다. 아울러 알고리즘에서 효과음처리 또는 배경음악을 처리할 때 문제점은 기존의 배경음악이 출력되고 있는 도중 이벤트 효과음을 처리하고자 할 때 기존의 배경음악을 중복적으로 출력되는 문제점이 있었다. 이러한 문제점을 해결하고자 제안한 논문의 알고리즘을 사용하여 문제점을 해결할 수 있었다.

V. 결론

슈팅게임을 구현하는데 여러가지의 구현방법이 있고, 다양한 언어를 사용하게 된다. 그러나 본 논문에서는 Visual C#을 이용하여 구현하는 방식의 프로그램 기법인 XNA 게임 개발 설명과, 이펙트효과, 게임의 몰입현상과 게임요소를 기술하였고, C++를 이용한 게임개발의 어려움, 게임에 필요한 화면 설계, 게임 구현 및 실험 부분에서 게임의 실행화면과 기존의 게임의 차별성과 알고리즘을 제시하였으며, XNA기반의 슈팅 게임의 구현과 구현에서 필요한 이펙트 효과와 알고리즘을 제시하였다. 기존의 게임개발에서 C++를 사용하여 개발한 슈팅게임과 XNA를 사용하여 개발한 슈팅게임과의 비교 분석에서 XNA를 사용하여 개발한 프로그램의 기법에서 프로그램의 효율성이 높다는 것을 보였고, 이펙트 효과의 기법의 실행화면을 제시하였으며, 이미지 설계나 화면 설계에서 보다 자연스러움과 사용자들에게 보다 친숙하면서 지루하지 않는 내용의 이펙트나 이벤트 처리를 강화하여 설계 구현하여 게임실행 화면을 보였다. 제안한 알고리즘에서 효과음처리 또는 배경음악을 처리할 때 문제점은 기존의 배경음악이 출력되고 있는 도중 이벤트 효과음을 처리하고자 할 때 기존의 배경음악을 중복적으로 출력되는 문제점이 있었는데 알고리즘에서 해결 방안을 제시하였다. XNA기반 게임 개발의 가능성을 보였다.

참고문헌

- [1] Jeong-Man Seo, "A Design Consideration Element and Serious Game for Disabled person" Journal of The Korea Society of Computer and Information, Vol 16, No 1, pp.87-87, Jan. 2011.
- [2] Jeong-Man Seo, "Implementation of Shooting game using collision detection algorithm of" Journal of The Korea Society of Computer and Information, Vol 11, No 3, pp188-192, July. 2006.
- [3] Jeong-Man Seo, "Design and Implementation of SADARI Game" Journal of The Korea Society of Computer and Information, Vol 14, No 8, pp19-24, Aug. 2009.
- [4] Bandura, A. Self-efficacy: Toward a unifying theory of behavioral change, Psychological Review, 84(2), 191-215, 1997.
- [5] Seong-Do Song, Eun-Ju Kim, "Campus Upbringing Simulation Game using XNA", Vol 12, No 1, pp.629-629, Jan. 2009.
- [6] Um Myoung-Yong, "Factors Influencing Online Game Play of Adolescence: An Examination of Social and Individual Playfulness", Journal of Korea Corporation Management Association, Vol 13, No 1, pp.165-179, Jun 2006.
- [7] Baek Syung-Kyeun, "A Study on Reality Effect of Animation", Journal of Animation Society of Korea, Vol 8, No 1, pp.48-67, May 2012.
- [8] Jeong-Man Seo, "Create a shooting Game Using XNA step-by-step", Jungil Publishing, Aug. 2011
- [9] Benjamin Nitschke, "ProfessionalXNA game programming : for Xbox 360 and Windows", Wiley Technology Pub., 2007.
- [10] Young Mee Choi, Moon Won Choo, "Object-oriented Concept Learning using Rever-engineering Method Based on XNAe Game Development Environment", Journal of Digital Contents Society, Vol. 10, No 1, pp.45-54, Mar. 2009.

- [11] WHITE PAPER ON KOREAN GAMES 2010,
Korea Creative Contents Agency
- [12] Jeong-Man Seo, "The Design and An
Implemetaion of Shooting Game Based on XNA
Game Development Evnvironment", Journal of
The Korea national universty of Welfare, Vol 1,
No 1, pp.87-87, Dec. 2011.
- [13] Jeong-Man Seo, "Design and Implementation of
Typing Game for the Disabled" Journal of The
Korea Society of Computer and Information, Vol
18, No 1, pp.55-62, Jan. 2013.

저 자 소 개



서 정 만

2003년: 충북대학교
컴퓨터공학과 공학박사
2002년~현재 : 국립한국복지대학교
게임콘텐츠과 교수
관심분야: 데이터베이스,
게임프로그래밍, 실시간처리
Email: seojm@knuw.ac.kr



최 창 수

1999년: 한국기술교육대학교
정보통신공학과 졸업(공학사)
2004년: 한국기술교육대학교
전기전 자공학과 졸업
(공학석사)
2010년: 충북대학교
컴퓨터공학과 박사
1999년~현재: 청주교도소
직업훈련교사
관심분야: 생체인식, 영상처리,
컴퓨터비전
Email: c415119@naver.com