

PC게임에서의 플래시 기반 게임 인터페이스 설계 및 구현

문성원[○], 한성호, 조형제**
동국대학교 멀티미디어학과**

vincent1@chollian.net, maxtopia@hotmail.net, chohj@dgu.edu

Designed and Implementation of Flash Game Interface based on
PC Games

Sung-Won Mun[○], SungHo Han, HyungJe Cho**
Dept. of Multimedia, Dongguk Univ

요 약

최근 3D게임 개발에 있어서 게임의 재미나 성능 못지않게 게임 인터페이스의 중요성이 부각되고 있다. 플래시는 손쉬운 콘텐츠 제작과 다양한 멀티미디어 표현이 가능하므로 향후 게임 인터페이스의 표현에 있어서 새로운 가능성 및 발전을 제공할 수 있다고 본다.

본 논문에서는 이러한 점을 감안하여 3D PC 게임의 인터페이스에 플래시를 사용하기 위한 시스템을 설계하고 이를 이용한 게임을 구현하여 그 유용성을 확인하였다.

ABSTRACT

In the recent development of 3D game, the importance of the game interface as well as the amusement and performance of the game is highlighted.

Considering the fact that the flash can easily express the various multimedia contents will provide new possibilities and improve the presentation of game interfaces, the system using the flash in 3D PC game interface was designed and tested in this paper.

Keyword : Flash, Game Interface

접수일자 : 2008년 12월 05일

심사완료 : 2009년 01월 31일

1. 서론

일반적으로 유저 인터페이스라고하면 컴퓨터를 사용하면서, 그림으로 된 화면 위의 물체나 틀, 색상과 같은 그래픽 요소들을 어떠한 기능과 용도를 나타내기 위해 고안된 사용자를 위한 컴퓨터 인터페이스를 말한다[1].

최근 3D 게임에 있어서 인터페이스는 다른 게임의 요소에 못지않게 중요성이 부각되고 있다. 그 이유는 게임 기술의 발전과 함께 게임의 표현 방식이 복잡해지고 유저들의 게임에 대한 요구사항이 점점 다양해짐에 따라 게임의 내용을 유저들에게 좀 더 효과적으로 표현하기 위해서 게임 인터페이스의 중요성이 부각이 되고 있기 때문이다. 요즘 개발되고 있는 게임의 인터페이스를 살펴보면 과거의 정보 전달 방식의 단순화된 인터페이스에서 게임의 유저들과 상호 교류가 가능한 발전된 형태의 인터페이스를 제공하고 있다. 그러나 이러한 발전된 인터페이스를 표현하기 위해선 기존의 게임 인터페이스 개발방식으로는 한계가 있다. 표현이 복잡해질수록 비례적으로 개발 기간 및 비용이 증대하게 되고 애니메이션과 같은 역동적 화면 구성을 구성하는 것은 과거의 인터페이스 방식으로는 한계가 있다[2].

그런데 최근 들어 웹 개발이나 광고에 일부 한정적으로 사용되던 플래시 기술이 휴대폰에 이어 홈오토메이션·PMP·디지털 카메라 등으로 적용 범위를 넓혀가고 있다. 정보기기에서 디자인의 중요성이 가속화되며 기존 비트맵(bit map) 중심의 그래픽 유저 인터페이스(GUI) 보다 역동적 이미지를 구현하는 플래시 GUI가 각광받게 된 것이다. 이러한 추세는 게임에도 이어져 3D온라인 게임에도 플래시 유저 인터페이스를 사용하는 게임이 서비스되기 시작했다[3]. 이러한 시도에도 불구하고 아직 국내의 온라인 게임에서의 플래시 인터페이스 사용은 초기 단계이고 플래시를 게임의 인터페이스로 사용하는 것에 대한 국내의 연구도 전무한 상태이다[4].

플래시가 가지고 있는 역동적 이미지 표현 능력은 동일한 장르의 게임을 중복해서 개발하고 있는 게임의 소재 표현의 한계를 극복하여 좀 더 다양한 장르의 게임의 개발에 도움을 줄 것으로 본다.

본 논문은 이러한 점을 감안하여 실제 게임에 플래시 인터페이스를 적용하기 위한 방법을 연구하였다. 3D 게임 내에서 플래시를 이용한 인터페이스 제작 방식을 설계하였고 게임 내에 플래시를 연동하기 위한 플래시 모듈을 제작하였으며 이를 적용한 실제 게임을 제작하였다.

2. 관련 연구

게임 인터페이스를 표현하기 위한 가장 고전적인 방식은 비트맵 이미지를 이용한 방식이다. 3D 게임에서도 비트맵 이미지를 이용한 방식은 사용되고 있는데 3D 게임의 렌더링 파이프라인을 사용하지 않고 바로 투영을 하게 된다. 이러한 방식은 역동적인 화면인터페이스를 표현하는데 효과적이지 않다.

스프라이트를 이용한 출력방식은 인터페이스의 각 프레임 이미지를 제작하고 각각 프레임의 속도 제어를 통하여 인터페이스의 애니메이션 효과를 추가하여 보여 줄 수 있다. 가장 손쉽게 구현할 수 있으므로 많이 사용은 되지만 좀 더 다양한 화면을 구성하기 위해서는 별도의 틀을 제작하여야 하고 역동적 애니메이션 효과를 표현하는데 있어서 한계가 있다. 또한 인터페이스는 보여 지는 화면 이외에도 유저의 이벤트에 대한 처리를 해야 하므로 이벤트에 대한 추가적 프로그램 작업을 진행해야 한다[5].

플래시는 웹 개발에 많이 사용되어온 틀이지만 이를 게임 인터페이스 개발에 사용하게 되면 추가적 개발이 필요 없는 다양한 기능을 제공하는 인터페이스 틀의 역할을 수행할 수 있다.

인터페이스 개발 관점에서의 플래시의 기능은 아래와 같다. 애니메이션을 신속하게 제작하고, 모션 경로를 손쉽게 수정할 수 있으며, 객체 기반의

애니메이션을 통해 개별 애니메이션 속성을 완벽하게 제어할 수 있다. x, y, z 축을 따라 움직임을 조정할 수 있는 새로운 3D 평행 이동 툴과 회전 툴을 사용하면 3D 공간에서 2D 객체에 애니메이션을 적용할 수 있으며 거의 모든 종류의 이미지 리소스의 사용과 편집이 가능하다[6].

3. 시스템 설계

본 논문에서는 이러한 플래시를 게임속에 연동하기 위하여 Adobe 사에서 공개한 플래시 데이터 포맷(format)을 기반으로 플래시 렌더러(Renderer)를 제작하고 게임 개발에서 가장 보편적으로 사용되고 있는 DirectX 기반의 게임 엔진에 플래시 렌더러를 연동하는 시스템을 설계했다.

3.1 플래시 파일 구조

플래시 파일인 swf은 확대와 축소에 도 외형을 유지하는 벡터 파일 및 텍스트, 사운드 등을 인터넷에서 실시간으로 로딩할 수 있는 확장성을 고려한 파일 포맷이다[7].



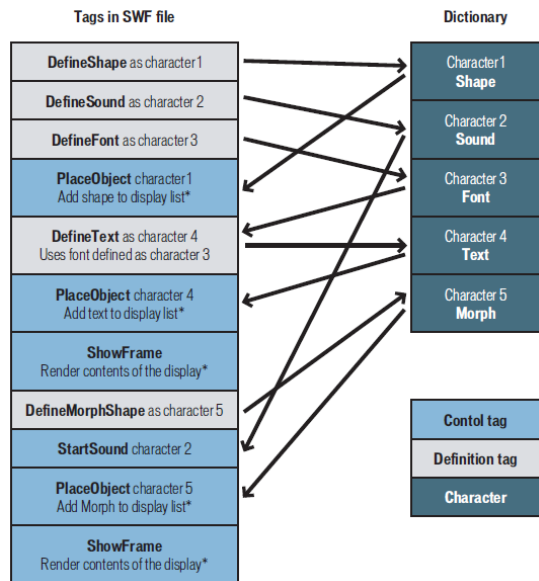
[그림 1] 플래시 swf 파일의 구조

위의 [그림 1]에서처럼 플래시 swf 파일은 헤더 정보와 파일 속성 정보, 일련의 태그로 구성되어 있다. 플래시 파일의 헤더 정보에는 버전정보와 애니메이션 정보가 들어있고 파일의 속성은 Flash 8 버전이상에서 사용되는 것으로 보안에 대한 정보를 설정한다. 플래시 파일의 가장 중요한 정보가 tag 인데 tag안에 오브젝트 정보가 들어가고 각각의 tag는 삭제와 삽입이 가능하므로 플래시 오브젝트를 객체 단위로 관리하는 것이 가능하다[8].

swf의 tag는 3가지 구조로 구성된다. 렌더링을 위하여 개별 오브젝트를 선언하는 define tag와 이를 참조하여 사용하기 위한 control tag, 세 번째

는 define된 오브젝트를 hash형태의 맵 구조로 저장하고 있는 dictionary tag로 구성된다.

플래시 사용자는 원하는 형태의 오브젝트를 선언하게 되면 이것은 define tag가 된다. 이 선언된 define tag는 dictionary tag에 저장이 되며 control tag에 의해서 오브젝트들의 위치 및 크기, 깊이(depth)정보가 control tag에 설정이 되고 실제 데이터는 dictionary tag를 참조하여 관리된다. 하나의 define된 character는 control tag에 의해서 중복하여 사용이 가능해진다.



[그림 2] swf 파일 구성 방식

위의 [그림 2]에서처럼 define tag는 dictionary에 저장이 되고 control tag에 의해서 실제적으로 화면에 배치가 된다. 객체의 정보를 설정한 define tag 5개가 dictionary에 저장이 되고 6개의 control tag에 의해서 화면의 배치를 결정하는데 각 control tag는 dictionary를 참조하여 같은 객체를 여러 개 추가해도 객체의 중복 없이 리소스를 공유하는 구조를 가진다.

3.2 플래시 렌더러 제작

플래시 파일을 렌더링하기 위해서는 아래의 4 단계를 거쳐서 플래시 파일을 로딩하고 출력을 하게 된다.

- 1단계 : tag 파일의 로드 및 dictionary 구성
- 2단계 : control tag에 따라 character object를 display 목록에 추가
- 3단계 : display 목록의 depth 에 따라 정렬
- 4단계 : 애니메이션 프레임에 따라 display 목록을 재 갱신

1, 2단계는 플래시 파일을 File Open 클래스를 이용하여 파일을 로드하고 파일을 tag 단위로 처리하여 이를 플래시 객체인 character로 구분한다. 구분된 character를 dictionary 맵 자료 구조에 넣고 control tag를 참조하여 실제 display 하기위한 character를 출력 목록인 display list에 추가한다.

3, 4단계는 실제 화면을 구성하는 단계로 로드된 무비의 character 객체를 개별 애니메이션 프레임 값에 따라 해당하는 프레임의 character 객체를 디스플레이 목록에 추가하고 character 객체의 depth에 따라 순서를 정렬하고 성능을 위해 가려지는 면에 대하여 클리핑 처리를 한다.

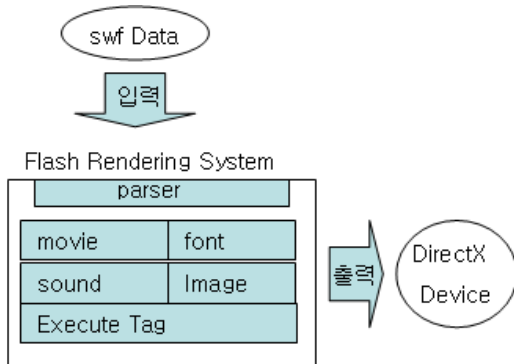
핵심 모듈로 구성이 되어있다. 데이터를 로딩 하는 파서 부분과 로딩 된 데이터를 각각의 오브젝트로 분류하는 과정, 마지막으로 각 애니메이션에 따라 출력 목록을 구성하고 이를 렌더링 하기위해 렌더링 파이프라인으로 전송하는 과정이다.

3.3 게임 엔진 연동

플래시 swf 파일을 3D 게임에 사용하기 위해서는 플래시 파일을 로드하여 얻어진 display 목록을 DirectX 나 OpenGL과 같은 하드웨어 레벨의 렌더링 파이프라인을 이용하여 처리해야 한다. 최근 대부분의 게임 개발이 DirectX 기반으로 제작되고 있으므로 DirectX 의 렌더링파이프 라인을 사용하여 플래시 게임 인터페이스를 출력하였다.

위의 [그림 3]의 구조에서처럼 로드된 플래시 캐릭터 정보를 제어하기 위한 출력 목록의 인스턴스를 얻어온다. 이 인스턴스에서 얻어진 최종 display 목록을 DirectX의 device의 표면(surface) 정보에 렌더링을 하여 플래시 게임 인터페이스를 출력한다[9].

속도 향상을 위하여 각 프레임별 플래시 display 목록을 DirectX의 Texture surface로 메모리 복사를 하여 게임 엔진의 화면 관리 프로세스의 마지막 단계인 인터페이스 단계에서 위의 플래시 유저 인터페이스를 출력 한다.



[그림 3] 렌더러 구조 설계

위의 [그림 3]에서처럼 플래시 렌더러는 3가지

4. 구현 및 성능분석

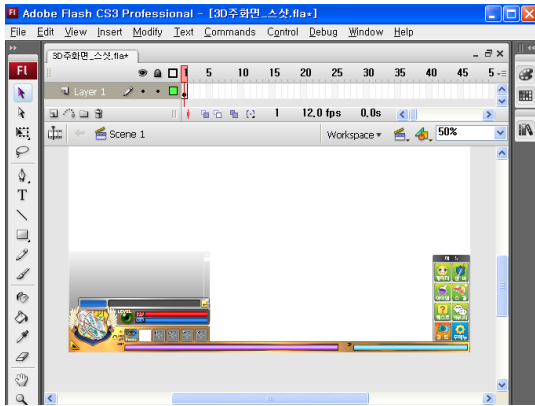
본 논문에서 제작한 플래시 인터페이스 렌더러를 실제 3D 온라인 게임에 적용하였다. 이를 통해 기존 인터페이스 제작 방식과의 차이점을 비교 분석하였고 기존 방식과 플래시를 이용한 인터페이스 방식의 성능 분석을 행하였다.

4.1 시스템 구현

플래시를 이용한 게임 인터페이스의 구현은 3단계로 작업이 진행된다. 첫째로 인터페이스용 데이터 작업 제작 단계와 두 번째로 이 데이터를 이용

한 게임속의 로딩 과정, 세 번째로 로드된 플래시와 게임 로직간의 상호간의 이벤트 호출 작업이다.

1단계의 콘텐츠 제작 단계는 일반적으로 웹 서비스에서 사용되는 플래시 제작 단계와 차이가 없다. 디자이너는 이미지를 등록하고 애니메이션이 있는 경우 프레임에 따라 애니메이션 무비클립을 작업하면 된다.



[그림 4] 플래시 인터페이스 제작 작업

두 번째는 제작된 플래시 리소스를 게임 프로세스에서 메모리로 로드하는 과정이다. 이는 게임에서 텍스처를 출력하기 위해 PC의 메모리에 올리는 과정과 같다. 차이가 있다면 DirectX에서 제공하는 텍스처 로딩 함수를 이용하는 것이 아니라 플래시 렌더링을 위해 제작된 컴포넌트의 메서드를 사용하여 최종 출력 결과물을 이미지로 만든다.

세 번째로 인터페이스의 가장 중요한 유저와의 상호간 인터랙션 처리를 위하여 게임에서 호출되는 이벤트를 플래시 컴포넌트로 보내게 되고 이에 따라 플래시 렌더러는 이벤트를 처리하게 된다. 마찬가지로 플래시 렌더러로부터 발생한 이벤트는 게임 프로세스에게 보내지게 된다. 위의 3가지 단계를 통하여 플래시 게임 인터페이스가 게임 내에 통합되어 동작을 하게 된다.



[그림 5] 게임내의 플래시 인터페이스 연동

위의 [그림 5]는 수학을 소재로 한 교육용 게임인 “판타지수학대전”이다[10]. 게임의 스킬을 사용하는 부분에서 위의 [그림 4]의 작업 1단계에서 제작된 리소스를 플래시 렌더러를 사용하여 게임 내에 연동한 것이다. 기존의 인터페이스 방식에 비해 역동적인 화면 구성이 가능하고 게임내의 스킬 사용과 같은 특별한 연출을 실감나게 표현할 수 있었다.

4.2 성능 분석

기존 게임에서 사용되는 이미지 출력 방식의 게임 인터페이스와 플래시를 이용한 게임 인터페이스의 성능 비교를 위하여 테스트를 위한 플랫폼을 선정하였다. 테스트 플랫폼으로 CPU는 Pentium 4 Dual Core 기반으로 하였고 메모리는 1G, 비디오 카드는 Geforce 5600 기반의 중 사양의 컴퓨터가 활용되었다.

테스트는 두 가지로 진행을 하였다. 첫째는 게임의 메인 화면과 같이 애니메이션 변화가 적은 인터페이스 출력 시의 프레임 차이를 비교 하였고 두 번째는 위의 [그림 5]에서 게임 상의 스킬을 사용하는 경우와 같이 화면의 인터페이스가 애니메이션 연출이 되어 복잡하게 구성되는 경우의 성능을 비교하였다. 그리고 좀 더 현실성 있는 테스트 환경이 되기 위하여 인터페이스의 사용 조건을 실제 게임 환경처럼 설정하였다. 테스트 게임 환경에서

프레임 여유가 많은 경우 인터페이스의 추가가 게임의 성능 하락에 영향을 주지 못하므로 실제 게임 서비스와 같이 게임의 캐릭터, 지형, 효과 등과 같은 게임의 요소를 추가하여 전체 게임의 프레임을 게임 서비스를 위한 기준 프레임인 30 fps(Frame per Second) 로 떨어트린 후 인터페이스 추가시의 프레임의 변화를 테스트하였다.

(fps : 초당 프레임)

구 분	정적 메뉴	애니메이션 메뉴
이미지 출력 방식	30.0	29.4
Flash 출력 방식	29.8	28.7

[그림 6] 성능 비교표

위의 성능 비교에서처럼 기본 인터페이스의 출력 시에는 이미지 출력 방식과 플래시 인터페이스 방식의 프레임이 차이가 크지 않다. 정해진 크기의 이미지를 특별한 애니메이션 없이 고정적으로 출력하는 것이므로 프레임의 차이가 보여 지지 않는다. 애니메이션이 들어가는 스킨 연출 화면의 출력에서는 플래시를 이용한 방식이 약간의 프레임 저하가 보인다. 내부적으로 렌더링 파이프라인에 통합되어 그래픽 카드의 디바이스에 접근하는 이미지 출력 방식이 좀 더 나은 성능을 보이나 그 차이가 근소하므로 플래시를 이용한 게임 인터페이스의 개발이 가지는 개발의 편이성과 표현의 자유도를 고려한다면 플래시를 이용한 게임 인터페이스의 개발은 장점을 가지고 있다고 본다.

5. 결 론

본 논문은 게임 인터페이스 표현의 새로운 가능성을 제시하는 플래시 기반의 유저 인터페이스 개발 방법론을 제시하고 이의 구현을 통하여 게임 인터페이스 개발의 새로운 접근법을 제시하였다.

플래시가 가지고 있는 화려한 애니메이션 효과

와 깔끔한 벡터그래픽의 UI의 표현 방식은 게임의 외형적 호감도의 변화와 UI제작의 편이성을 향상 시키는데 기여를 할 것으로 본다.

참고문헌

- [1] Andrew Rollings, Game Architecture and Design: A New Edition, NEW RIDERS, 2004
- [2] 이주호, 게임 인터페이스 디자인 효율성에 대한 연구, 성균관 대학교, 2004
- [3] <http://www.scaleform.com/>
- [4] 게임 백서, 한국게임 산업 진흥원, 2007
- [5] 게임 인터페이스 현황 분석 및 중장기 추진 계획, 한국 게임 산업 진흥원, 2005
- [6] <http://www.adobe.com/kr/products/flash/>
- [7] <http://www.adobe.com/devnet/swf/>
- [8] <http://www.tulrich.com/geekstuff/gameswf>
- [9] Eberly, David H. 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics. Morgan Kaufmann, 2001
- [10] <http://www.edunplay.co.kr>



문성원(Sung-Won Mun)

1997년 단국대학교 경영학과 졸업(학사)
2002년 동국대학교 연극영화학과 졸업(석사)
2006년 동국대학교 영상대학원 멀티미디어학과 박사수료
1997~1999 KONET(美) Lead Designer
1999~2002 디지아이 프로덕션 감독
2001~2004 광운대학교 정보과학교육원 교수
2005~현재 광운대학교 교육대학원 멀티미디어전공 초빙교수
2006~현재 뷰모션 주식회사 대표이사

관심분야 : 게임그래픽, 가상현실,
실험 애니메이션, 모션캡처 등



조형제(HyungJe Cho)

1973년 부산대학교, 전자 공학과(학사)
1975년 한국과학기술원, 전기 및 전자 공학과 대학원(공학 석사)
1975년~1982년 금성통신(주)연구소 실장
1986년 한국과학기술원, 전기 및 전자 공학과 대학원 (공학 박사)
1986년~현재 동국대학교 영상대학원 멀티미디어학과 교수

관심분야 : 멀티미디어 정보처리, 컴퓨터비전,
컴퓨터 그래픽스, 형태인식,
게임 프로그래밍



한성호(SungHo Han)

1997년 단국대학교 경영학과 졸업(학사)
2002년 동국대학교 연극영화학과 졸업(석사)
2006년 동국대학교 영상대학원 멀티미디어학과 박사수료
1997~1999 KONET(美) Lead Designer
1999~2002 디지아이 프로덕션 감독
2001~2004 광운대학교 정보과학교육원 교수
2005~현재 광운대학교 교육대학원 멀티미디어전공 초빙교수
2006~현재 뷰모션 주식회사 대표이사

관심분야 : 게임그래픽, 가상현실,
실험 애니메이션, 모션캡처 등