

M3L : 모바일 장치를 위한 멀티미디어 라이브러리†

도창욱⁰* 김정숙^{**} 오세만*
*동국대학교 컴퓨터공학과, **삼육의명대학 인터넷과
moonchld@dongguk.edu kimjs@syu.ac.kr smoh@dongguk.edu

M3L : Multimedia Library for Mobile Device

Chang-Wook Doh⁰* Jung-Sook Kim^{**} Se-Man Oh*
*Dept. of Computer Engineering, Dongguk University,
** Dept. of Internet, Sahmyook College

요 약

모바일 장치의 기능 향상과 무선 데이터 통신 기술의 발달로 유선 인터넷 서비스를 모바일 장치에서 사용하기 위한 다양한 지원 모델이 연구되고 있으며, 이에 관련된 응용 프로그램이 개발되고 있다. 또한, 장치의 한계성으로 인해 콘텐츠의 표현을 정적인 형태에 의존하였던 초기와는 달리 하드웨어의 발전으로 멀티미디어 형태의 표현 방식이 요구되고 있다. 그러나, 대다수의 모바일 장치에서 멀티미디어 기능 구현은 플랫폼 종속적이므로, 다른 모바일 플랫폼으로 이식하는 데는 많은 부분을 다시 작성하여야 하는 단점이 있다.

본 논문에서는 다양한 모바일 장치에서 동일한 프로그래밍 인터페이스를 사용하여 멀티미디어 응용 프로그램을 구현할 수 있는 멀티미디어 라이브러리를 개발하였다. 라이브러리의 개발을 위해 이전 라이브러리 기능 분류를 통해 모바일 장치에서 사용 가능한 멀티미디어 기능을 정의하였고, 다른 환경으로 쉽게 이식할 수 있도록 계층적인 구조를 가지는 플랫폼 독립적인 멀티미디어 라이브러리를 개발하였다.

1. 서 론

멀티미디어 기술은 유, 무선 네트워크 기술과 더불어 빠른 속도로 발전하고 있으며, 향후 IMT-2000의 도입을 통한 고속의 통신 기술의 발달로 인하여 유선 인터넷에서 제공되는 다양한 형태의 멀티미디어 서비스가 모바일 환경에서도 가까운 미래에 가능해질 것으로 예상된다. 인터넷의 접속 유형은 유선 데이터 통신 영역과 모바일 영역으로 구분할 수 있으며, 최근 모바일 영역의 규모가 빠른 속도로 성장하고 있다. 모바일 기술의 발달로 인해 무선 인터넷 사용자는 모바일 장치를 통해 언제 어디서나 원하는 정보를 손쉽게 취득할 수 있다. 무선 인터넷은 기존 유선 인터넷의 특성과 매우 다르며 무선 인터넷 환경을 위한 모바일 장치의 멀티미디어 기능 개발의 복잡도가 증가하고 있어 모바일 장치를 위한 멀티미디어 라이브러리의 필요성이 대두되고 있다[1].

본 논문에서는 동일한 프로그래밍 인터페이스를 이용하여 다양한 장치에서 멀티미디어 서비스를 가능하게 하는 모바일 장치를 위한 멀티미디어 라이브러리인 M3L을 개발하였다.

2. 관련연구

2.1 OpenGL

OpenGL은 3차원 그래픽 응용 프로그램 개발을 위한 업계 표준 API이다. OpenGL은 SGI사의 워크스테이션을 공학용, † 본 연구는 중소기업청의 “산학 컨소시엄 지원사업 (과제번호:02-05)”의 연구 지원에 의한 것임.

과학용, 또는 특수효과 개발용 플랫폼으로 사용되었던 Silicon Graphics사의 방대한 그래픽 라이브러리인 IRIS GL 라이브러리를 계승하여 좀더 쉽게 이용할 수 있고 직관적이며 미래를 대비하여 확장 가능한 구조를 지니는 API 집합이 될 수 있도록 개발되었다. 그러나 OpenGL은 이전의 IRIS GL라이브러리와는 달리 플랫폼이나 운영체제에 독립적으로 설계되었다. 운영체제에 독립적인 성격으로 인하여 또한 응용 프로그래밍의 측면에서는 API의 내부적 변화에 대해 무관하도록 작성되어 OpenGL을 이용한 응용 프로그램은 어떠한 플랫폼으로도 쉽게 이식되어질 수 있다.[2]

OpenGL은 플랫폼 독립성을 얻기 위하여 윈도우 작업에 관련된 명령이나 사용자의 입력을 받는 부분을 분리하고 복잡한 모델을 서술하는 명령어를 제공하지 않고 기본 API를 바탕으로 복잡한 모델을 사용할 수 있도록 서술된 라이브러리가 별도로 제작되었다. [3] 따라서 플랫폼 개발자는 이에 해당하는 기본적인 인터페이스를 제공하여 주며 응용 프로그램 개발자는 그 상위에서 제공되는 다양한 유틸리티 API를 사용하여 응용 프로그램의 개발 효율성을 높일 수 있다.

2.2 DirectX와 GAPI

DirectX는 Microsoft사의 플랫폼인 윈도우 환경에서 멀티미디어 하드웨어를 효율적으로 사용하기 위해 발표된 API이다. OpenGL이 플랫폼 독립적이며 행렬을 이용한 그래픽 연산을 목적으로 삼는 반면 DirectX는 장치 독립성과 그래픽

픽 연산을 포함하고 사운드와 동영상과 같은 보다 넓은 의미의 멀티미디어 기능을 제공하는데 그 목적을 두고 있다.

DirectX는 초기 윈도우 기반의 멀티미디어 API에서 발생하는 오버헤드를 피하기 위해 장치 의존적인 특성을 숨기는 HAL(Hardware Abstraction Layer)을 기반으로 구성된다. HAL을 통해 구현되는 기능은 하드웨어로 구현되어 있어 효율성을 기대할 수 있다.

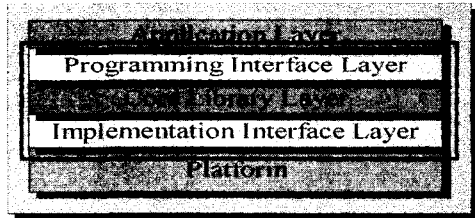
또한, DirectX는 추후 확장을 위해 현재 발표된 장치에서 널리 유효하지 않은 특징에 대해서도 정의되어 있으며, 이와 같이 현재 장치에서 유효하지 않은 특성에 대해서는 해당 특징을 HEL(Hardware Emulation Layer)을 통해 에뮬레이션하여 지원하도록 설계되어 있다. 이처럼 HAL과 HEL을 통해서 DirectX에서 광범위하게 정의되어 있는 기능들은 장치 독립적으로 구현 가능하여 윈도우 응용 프로그램 개발자들이 다양한 장치간 호환성이 없는 저수준 코드를 사용하지 않고 장치 독립적인 코드를 통하여 효율적인 상태로 실행되는 응용 프로그램을 작성할 수 있다.[4]

GAPI(Game API)는 모바일 장치를 위해 설계된 WinCE(Windows Compact Edition)의 PocketPC 플랫폼에서 실행 성능에 많이 의존하는 게임과 같은 멀티미디어 응용 프로그램을 위해서 화면 장치 메모리 영역에 직접 접근 가능하도록 설계된 API 집합이다. GAPI는 장치 의존적이며 화면 장치에 대해 정의되어 WinCE PocketPC를 기반 플랫폼으로 하는 장비에서 빠른 화면의 조작을 목적으로 삼고 있다.[5]

3. 플랫폼 독립적인 모바일 멀티미디어 라이브러리 모델

모바일 장치를 위한 플랫폼 독립적인 멀티미디어 라이브러리의 설계 목표는 라이브러리 자체의 기능 확장과 다양한 모바일 장치에 대한 라이브러리 이식을 위해 분리된 구조이다.

계층적으로 분리된 구조의 멀티미디어 라이브러리는 플랫폼에 의존성을 가지는 구현 인터페이스 계층과 목적 기계에 상관없이 멀티미디어 기능을 구현함에 있어 필요한 과정을 기술하는 핵심 라이브러리 계층 그리고 이식이 완료된 라이브러리의 기능을 이용하여 유용하게 사용될 수 있는 다양한 유틸리티 API가 제공되는 프로그래밍 인터페이스의 3가지의 구조이다.



[그림 1] M3L의 계층 구조

[그림 1]과 같이 구현 인터페이스는 멀티미디어 데이터를 모바일 장치에서 실현하기 위해 플랫폼에 의존적인 부분으로 구현되어 장치를 조작하고 관리한다.

핵심 라이브러리는 플랫폼 독립적인 부분으로 구현 인터페이스를 이용하여 멀티미디어 데이터를 처리하는 알

고리즘이 기술된 API의 집합으로 구성된다.

프로그래밍 인터페이스는 구현 인터페이스와 핵심 라이브러리에서 제공되는 기능을 응용프로그램 상에서 확장하고 사용하기 위한 인터페이스를 제공한다. 이러한 계층 구조는 모바일 장치에서 멀티미디어 기능을 개발하기 위하여 개발 목적에 따라 서로 다른 최종 사용자를 대상으로 분리되어 있다.

4. M3L의 개발

멀티미디어 기능의 수행 성능을 보장하고 일반적인 제 목적 이식성을 살펴 보았을 때 가장 적합한 언어가 C로 판단되어 M3L은 구현 언어로 C를 선택하였다.

4.1 분류

M3L에서는 기존 연구에 따라 멀티미디어 기능 구현을 위해 장치별 특성과 목적에 따라 나뉘어진 API 그룹으로 정의되는 라이브러리 카테고리를 분리하였다.

M3L은 모바일 게임과 같은 모바일 멀티미디어 응용프로그램에서 가장 많이 사용하는 API들에 대하여 카테고리화 작업을 행하였다. 여기서 정의된 카테고리는 다수 다수의 API로 세분화되었으며, 3장에서 설명한 구조에 의해 구현되었다.

[표-1] M3L의 API 카테고리

| 카테고리 | 설명 |
|--------------|-------------------|
| M3Surface | 화면 메모리 맵 라이브러리 |
| M3Graphics2D | 2D 그래픽스 라이브러리 |
| M3Image | 2D 이미지 라이브러리 |
| M3Sprite | 2D 스프라이트 라이브러리 |
| M3Animation | 애니메이션 라이브러리 |
| M3Font | 폰트 라이브러리 |
| M3MsgAudio | 메시지 기반 사운드 라이브러리 |
| M3Wave | 비메시지 기반 사운드 라이브러리 |
| M3Astream | 비동기 스트림 라이브러리 |
| M3Device | 범용 장비 제어 라이브러리 |

M3Surface는 화면 표시에 관련하여 필요한 메모리 맵을 관리하며, M3Graphics2D는 점, 선, 면과 같은 2차원 그래픽의 기본 요소를 표현한다. M3Image는 2차원 비트맵 이미지를 처리하기 위한 라이브러이며, M3Sprite는 스프라이트에 대한 API들이 기술되어 있다. 스프라이트의 특성상 연속적인 투명색이 많아 이미지의 크기를 줄이고, 스프라이트 표현시 실행 성능을 높이기 위해 압축 알고리즘을 사용하였다. 그러나, LZW나 허프만 인코딩과 같은 압축 방식은 용량면에서 효과가 높지만 디코딩 과정에 시간을 많이 소요하므로 RLE0를 사용하였다.

M3Font는 스프라이트 라이브러리를 기반으로 구현되었다. M3Font는 비트맵 폰트를 제공하며 한글의 경우 KS-5601C를 지원한다.

M3Device는 장치에서 제공할 수 있는 추가 기능을 유연성 있게 구현하기 위해 장치 열거와 범용 인터페이스를 제

공한다. 열거에 의해 얻어지는 기능은 M3L API에 의해 사용할 수 있으며, API로 제공되지 않는 기능은 범용 인터페이스를 통해 사용할 수 있다.

4.2 API의 명명 규칙

라이브러리는 사용자의 명칭 기억에 있어 편의성을 제공하기 위해 모든 API에 대하여 일관되고 동일하게 사용되어야 한다. 현재 가장 널리 쓰이고 있는 명칭 규칙은 헝가리언 표기법과 JAVA Beans에서 제시된 명칭 방법, 그리고 Microsoft사의 명칭형태로 나눌 수 있다.

헝가리언 표기법의 경우 명칭에 있어 형과 관련된 부분에 있어 엄격히 표현할 것을 제시하고 있다. OpenGL과 같은 API는 기본적인 형태를 헝가리언 표기법을 따르고 있다.

JAVA Beans로 널리 알려진 JAVA의 표기법은 메소드의 정의에 있어서 동사가 접두사로 사용되고 목적 대상이 접미사로 사용된다. 또한 JAVA는 객체지향 개념에 따라 모든 메소드가 클래스나 인터페이스에 종속되도록 정의되어 있어 코드의 가독성이 높은 장점이 있다.

M3L의 API는 관련 기능에 대해 카테고리화되어 있으나, C를 기반으로 작성하여 Java Beans와 같은 형태의 명명 규칙의 경우 카테고리를 표현하기 위해 클래스 개념을 사용할 수가 없다.

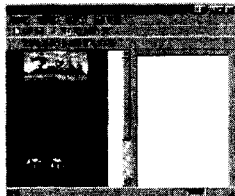
따라서 M3L API를 사용자 함수, 플랫폼의 API의 명칭과 분리하기 위해 모든 M3L의 API는 M3로 시작한다. 또한 카테고리를 표시하기 위해 카테고리를 접두사로 사용하고 동사를 접미사로 사용하여 작성하였으며, 식별성을 위해 M3SurfaceCreate와 같이 모든 분리부가 대문자로 시작한다.

4.3 관련 도구

M3L에서 스프라이트나 폰트와 같은 데이터는 적은 리소스를 차지하도록 크기를 감소시키거나 API에서 쉽게 사용할 수 있도록 변환 과정을 거쳐야 한다. 이와 같은 과정은 특별히 제작된 관련도구로써 변환에 사용되는 노력을 절약할 수 있다.

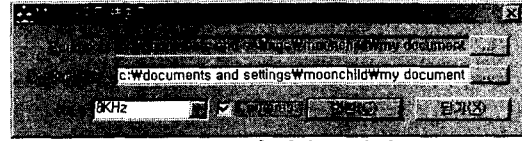
M3L에서는 이미지 관련 도구, 애니메이션 편집 도구, 사운드 관련 도구 3가지의 관련도구를 제공한다.

이미지 도구는 이미지와 스프라이트, 폰트를 편집 및 압축하고 변환한다. 이 3가지 요소는 서로 밀접한 관계가 있으므로 하나의 도구에서 작업이 가능하도록 통합하였다. 애니메이션 편집도구는 이미지 도구에서 작성된 스프라이트를 이용하여 API에서 사용 가능한 형태로 작성한다.



[그림 2] M3L의 이미지 관련 도구

사운드 변환 도구는 일반 웨이브 데이터에서 필요없는 정보를 삭제하고, 사운드 연주에 필요한 정보를 생성한다.



[그림 3] M3L의 사운드 관련 도구

5. 적용 사례

본 논문에서 개발한 M3L의 적용 및 실험을 위해 Microsoft사의 Window CE-PocketPC 2000 플랫폼[5]과 게임파크의 GP32[7]에 M3L을 적용하고 다양한 멀티미디어 기능을 실험해 볼 수 있는 응용 프로그램인 게임을 제작하였다. 위의 두가지 환경에서 M3L을 사용하는 응용 프로그램은 플랫폼에 관련된 코드의 수정만으로 다른 플랫폼에 쉽게 이식되었다.



[그림 4] M3L이 적용된 게임 프로그램

6. 결론 및 향후 연구

본 논문에서는 모바일 플랫폼 상에서 멀티미디어 기능을 구현하기 위한 라이브러리를 설계, 개발하였다. M3L은 다양한 플랫폼에서 멀티미디어 기능을 구현하기 위해 3계층 구조로 설계되어 사용자는 각 업무에 관련된 계층에만 관련하여 작업함으로써 멀티미디어 기능을 쉽게 구현, 응용할 수 있다. 이러한 라이브러리의 제공은 사용자의 요구가 다변화되고 있는 모바일 멀티미디어 환경하에서 다른 계층에 영향을 주지않으며 빠른 수정이 가능하여 라이브러리의 적용성을 높일 수 있다. 앞으로 모바일 멀티미디어 응용 프로그램의 개발 효율성을 높일 수 있는 추가적인 개발 도구와 통합 환경의 연구가 필요하다.

참고 문헌

- [1] Intromobile, Mobile Multimedia Technology Trend, <http://www.intromobile.co.kr/solution/>
- [2] Edward Angle, Interactive Computer Graphics: A top-down approach with OpenGL, 2nd Ed., Addison-Wesley, 2000.
- [3] Wright sweet, OpenGL Superbible, 2000, Waite.
- [4] Peter j. Kovach, Inside Direct3D, 2000, Microsoft.
- [5] Microsoft, PocketPC Official site, <http://www.microsoft.com/mobile/pocketpc/>
- [6] GamePark, GP32 Developer's site, GamePark, <http://www.gameparkdev.co.kr/>
- [7] D. Hearn and M. P. Baker, Computer Graphics: C version, 2nd Ed., Prentice-Hall International, 1999.