

게임 이식을 위한 미들웨어 플랫폼 설계 및 구현

Design and Implementation of Middleware Platform for Game Porting

김현미¹, 이원형²

Hyun Mi Kim¹, Won-Hyung Lee²

요약

오늘날, 게임 산업의 발달에 따라 플랫폼들이 다양해짐에 따라 아케이드, PC, 콘솔, 모바일, 온라인등에서 게임이 다양하게 되었다. 이러한 발전과 플랫폼으로 온라인 게임을 개발, 시판 하다가 모바일 게임을 출시하기도 하고, PC게임, 콘솔게임, X-BOX로 개발 등 장르와 개발방법이 다양화 및 멀티 플랫폼을 갖는 게임 등이 개발되고 있는 추세이다. 그러나, 이러한 게임 형태를 갖는 게임을 개발하는 것은 생산성과 경제성 측면에서 비효율적이다. 따라서 본 논문에서는 어느 플랫폼에도 구애받지 않는 게임 미들웨어를 설계하여, 개발자에게 보다 플랫폼 이식의 용이성과 편리성을 제공하는데 본 논문의 목적이 있다.

핵심어: 게임플랫폼, PC게임, 미들웨어, 플랫폼 독립게임, 게임산업

Abstract

In game, the most profitable thing for game industry is to develop professional gaming: create a large scale game league and broadcast it to some special game channels in cable TV and satellite channels, or build an internet broadcasting station, and even broadcast it in public broadcasting TV.

There are many kinds of platforms where game can be played, i.e. Arcade, PC, console, mobile, online game. In these platforms, game is developed to support multiplatform such as converting PC game to console game like play station and portable X-BOX, converting online game into mobile game.

In this paper, we propose a middleware for game development process, so the platform-dependent problem (or multi-platform problem) can be solved. Moreover, because independent platform game can be developed, game can be played in a various platforms. That is to say, our proposed method offers many benefits for the game developers such as reducing cost and development time, in developing a platform-independent game.

Keyword: Game platform, PC game, middleware, platform-indepent game, game industry

1 Graduate School of Advanced Imaging Science, Multimedia & Film, Chung Univ, 221 Huksuk-dong, Dongjak-ku, Seoul, 156-756, Korea, e-mail: hyunmi-kim@cau.ac.kr

2 Graduate School of Advanced Imaging Science, Multimedia & Film, Chung Univ, 221 Huksuk-dong, Dongjak-ku, Seoul, 156-756, Korea, e-mail: whlee@cau.ac.kr (Corresponding author)

* 이 논문은 김현미의 2009년도 석사 학위논문에서 발췌 정리하였음

Received(April 20.2014), Review (June 09.2014), Accepted(June 30.2014)

1. 서론

인터넷 인프라(infra-structure)를 바탕으로 정보의 디지털화와 빠른 전송 기술의 발전은 통신망을 이용하여 멀티미디어 서비스의 제공을 가능하게 하였고, 음성과 데이터 간의 구분이 없어지고 있으며, 유무선 통합 서비스 시대로 접어들고 있다. 즉 언제(anytime), 어디서나(anywhere), 어느 기기 (anydevice)든지 미디어에 구애받지 않고(anymedia) 통신 서비스를 이용할수 있는 4ANY의 유비쿼터스(ubiquitous) 시대가 열리고 있다. 지금 유무선 연동, 무선간 네트워크(network) 게임의 본격화가 예상되는 시점에서 유명 PC, 온라인 게임업체들의 모바일 게임 진출 또한 가속화되고 있다[1][2].

게임의 발달과 함께 게임을 즐길 수 있는 플랫폼(Platform)들도 다양해져서 아케이드(arcade), PC, 콘솔(console), 모바일(mobile), 온라인(online) 등 여러 가지의 플랫폼에서 게임이 가능하게 되었다. 이처럼 플랫폼이 다양해지면서 온라인 게임으로 발매하였다가 모바일 게임으로 출시하거나, PC 게임으로 개발하였다가 콘솔 게임인 플레이스테이션(play station), X-BOX로 개발하는 등 멀티플랫폼을 가지는 게임들이 개발되고 있다[3][4].

그러나 게임 개발자 측면에서는 하나의 게임을 PC, 콘솔, 모바일 등의 기기들에서 운영이 되는 다수의 플랫폼용으로 개발해야하는 부담이 있으며 게임 단말기 제조업체 또한 플랫폼에 의존된 시스템을 제각각 만들어야 하는 불편함이 있다. 또한 일반 사용자 측면에서는 특정 사의 게임 포털 정보만을 제공 받는 폐쇄적인 서비스 제공이라는 불만을 사고 있다[4][5].

따라서 본 논문에서 제안한 미들웨어(middleware)를 통해 게임이 개발 되면, 게임을 개발할 때마다 플랫폼에 맞추어서 다시 개발해야 하는 플랫폼 종속적인 개발의 문제를 해결할 수 있게 되어 플랫폼에 독립적인 게임의 개발이 가능하다. 즉, 어느 플랫폼에도 구애받지 않는 플랫폼 독립적인 게임 미들웨어를 설계하여 개발자에게 보다 플랫폼 이식의 용이성과 편리성을 제공함으로써 게임 개발 시간 중 이식에 들어가는 시간을 최소화하는 게임 미들웨어를 제안하고 구현하는 것을 그 목적으로 한다.

2. 관련연구

2.1 게임의 개요

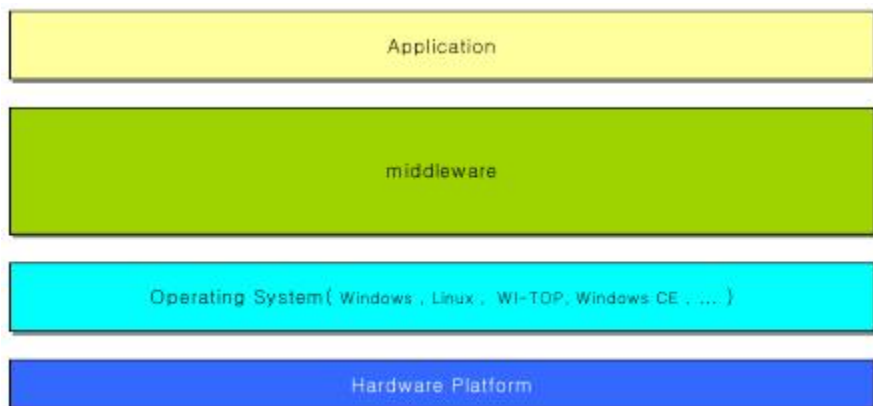
게임 역사를 살펴보면, 1958년 미국 브룩헤이븐(Brookhaven) 연구소에서 개발한 텍스트(text) 형태의 아케이드 게임인 테니스 게임이 최초의 원형으로 알려져 있다. 게임기가 일반 대중에게 소개되어 본격적인 시장을 형성하기 시작한 것은 1970년대 후반 가정용 콘솔 게임이 개발되면서부터이고, 1980년대와 1990년대 업무용 PC가 보급되기 시작하면서 본격적으로 게임 산업이 발전하게 되었다[6][7].

20세기 이후로 세계 각국에서는 문화관련 산업에 많은 관심을 보이기 시작 하였으며, 이러한 문화관련 산업은 빠른 성장세를 바탕으로 미래를 주도할 산업으로 주목을 끌었다. 이 중 게임 산업은 짧은 기간 동안 가장 높은 성장률을 보이며, 문화관련 산업의 주역으로 성장하였다. 게임 산업은 멀티미디어 (multimedia) 기술, 디지털 기술, 통신 기술 등 여러 첨단 기술과 엔터테인먼트(entertainment) 요소 등이 부가된 복합적인 산업분야이다. 따라서 게임 산업은 창의성을 바탕으로 높은 수익을 창출 할 수 있는 첨단 문화 산업이며, 한 나라의 문화적 역량과 첨단 기술의 수준을 동시에 드러내는 지식기반 산업이다. 이처럼 게임 산업은 높은 성장 가능성 때문에 전세계적인 주목을 받고 있으며, 우리 정부에서도 많은 인력과 자금 등 정책 지원을 확대하고 있는 분야이다[4].

2.2 플랫폼 구조

OS 플랫폼은 컴퓨터 하드웨어와 사용자 응용프로그램 사이에 위치하여 응용프로그램이 하드웨어를 쉽게 사용할 수 있도록 해주고, 전체적인 시스템의 효율을 극대화시키기 위해 하드웨어 및 소프트웨어 자원(resource)을 관리하는 API의 집합체로 프로세스 관리, 기억장치 관리, 입출력 서비스들을 한다[7].

운영체제는 하드웨어의 제약을 받으므로 하나의 컴퓨터 시스템을 위해 하나의 OS 플랫폼이 존재하는 것이 보통이고, 응용 프로그램은 시스템에 관계 되는 작업을 직접 처리하지 못하고 운영체제가 제공하는 기능을 호출하여 사용한다. 즉 하드웨어에 종속적인 OS 플랫폼 구조로 응용 프로그램도 OS 플랫폼에 종속적일 수밖에 없다[8][9].



[그림 1] 독립적인 플랫폼 구조
[Fig. 1] Independent platform structure

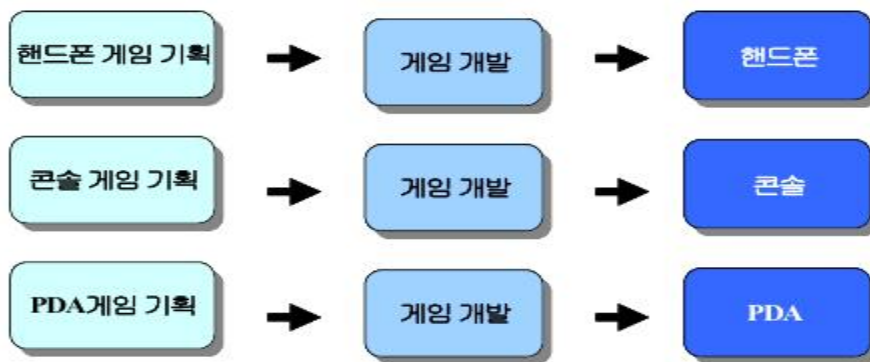
미들웨어는 애플리케이션들을 직접 연결하는 방식에 비해 중요한 이점이 있다. 일반적으로 애플리케이션이 OS에 독립적이기 때문에 미들웨어 부분의 추상화, 표준화 된 곳만 플랫폼과 연결을(porting)해주면 추가구현이 필요 없이 구동된다. 또한 OS에서 제공해 주는 API 보다 고수준(high level)의 함수 들을 제공해 줄 수 있다. 예를 들어, 플랫폼에서는 간단한 그리기 함수만 제공한다고 할 때 콤보박스(combo box)나 리스트박스(list box)와 같은 고수준의 기능적인 함수를 제공해 주기 어렵다. 그러나 이러한 기능을 미들웨어에서 추상화 하였을 때 쉽게 제공 될 수 있다. 그 외에 메모리 관리, 파일 관리, 자료 구조 관리, 데이터베이스(database)관리 등 고수준의 기능들을 제공할 수 있다.

3. 제안한 게임 미들웨어

3.1 게임 미들웨어의 필요성

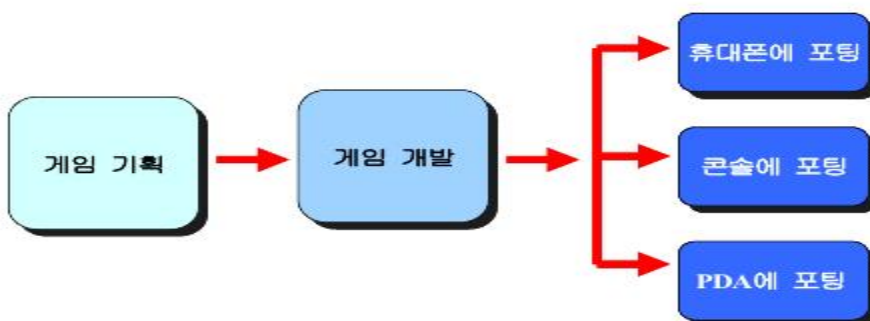
기존 게임 개발 방법은 플랫폼마다 특성이 있기 때문에 플랫폼에 맞추어 개발할 수밖에 없는 플랫폼 종속적인 개발 방식이다. 즉 현재 게임 개발 방법은 그림 2과 같이 각 플랫폼에 맞게 게임을 기획하고 각 플랫폼에 맞게 기획된 게임을 개발하여 각 플랫폼에 옮기는 플랫폼 종속적인 방법이다.

그러나 제안한 방법은 그림 8와 같이 다양한 플랫폼에 포팅 되는 경우라도 표준화된 게임 미들웨어를 통해 게임을 개발함으로써 한 번의 게임 기획으로 다양한 플랫폼에 쉽게 이식할 수 있어 지금 보다 플랫폼이 더 다양해져도 기존 보다 빠르게 게임을 이식할 수 있게 된다[10]. 이런 개발 방법은 다양한 함수 및 구조를 제공하여 개발자들에게 편리하고 유연한 게임 개발환경을 제공하고 게임 개발에 소요되는 시간을 단축할 수 있게 한다. 따라서 개발사 측면에서는 개발비용의 절감이라는 효과를 기대할 수 있다. 사용자 측면에서는 다양한 플랫폼으로 동일한 게임을 즐길 수 있고, 더 나아가 다양한 게임을 제공 받을 수 있는 Win-Win의 효과를 기대할 수 있다.



[그림 2] 기존 게임 개발 방법

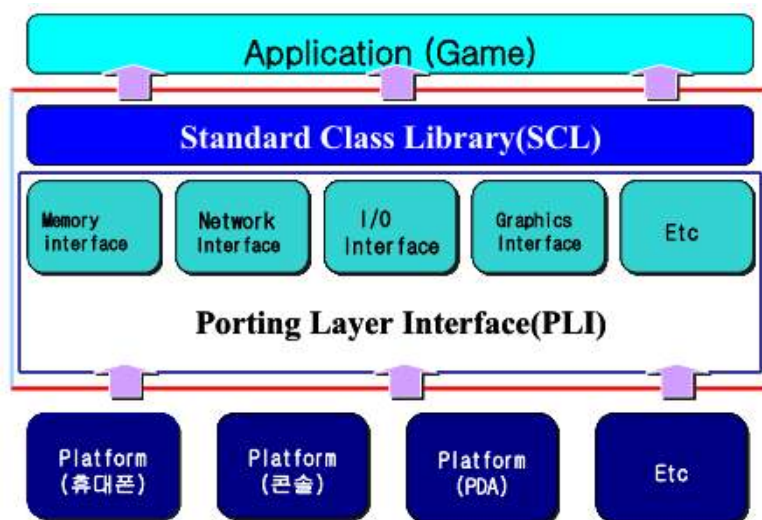
[Fig. 2] Existing game development methods



[그림 3] 제안된 게임 개발 방법

[Fig. 3] Proposed Game Development Method

3.2 제안한 게임 미들웨어의 구조



[그림 4] 게임 미들웨어 구조도

[Fig. 4] Game middleware structure diagram

본 논문의 제안은 그림 4와 같이 각 플랫폼으로부터 게임을 개발하는데 있어서 필요한 인터페이스 부분을 추상화한 PLI(Porting Layer Interface)를 설계한다. 그리고 추상화된 PLI 위에 SCL(Standard Class Library)를 설계한다. 이렇게 구성된 미들웨어를 통해서 게임이 개발되기 때문에 각 플랫폼에 독립적이다. PLI는 메모리 인터페이스 부분, 네트워크 인터페이스 부분, 입출력 인터페이스 부분, 그래픽 인터페이스 부분 등으로 구성된다[11][12].

- 메모리 인터페이스 부분은 메모리 할당(malloc), 해제(Free)에 관한 부분을 추상화하고, 메모리 단편화 방지, 액세스 시간 단축에 관한 부분을 추상화 한다.
- 그래픽 인터페이스 부분은 라인(line), 사각형(rectangle), 비트맵, DC관련 함수 등을 추상화 한다.
- 입출력 인터페이스 부분은 키보드, 마우스, 파일 입출력 등을 추상화한다.
- 네트워크 인터페이스 부분은 Open, Close, Read, Write부분을 추상화 한다.
- 이외에도 추상화가 필요한 부분은 추상화 한다.

이 추상화된 포팅층을 이용해서 SCL을 개발하는데 SCL함수는 자료구조, 즉 링크드리스트(linked list), 스택(stack), 큐(queue), 정렬, 검색 등에 관한 것들로 게임 개발의 편의를 제공하는

함수들이다. 그러나 본 연구에서는 자료구조에 관계된 알고리즘이 사용되지 않은 관계로 자료구조에 관한 구현 부분은 완료되지 않았다.

게임 쪽에서는 SCL과 PLI를 통해 게임 개발을 하게 되고 플랫폼에 이식을 할 때에는 PLI만 플랫폼에 맞도록 구성 한다. 즉 이식할 때에 전체적인 게임 구조를 개선하고 다시 검증해야 하는 기존 개발과는 달리 PLI만 새롭게 구성해 주고 검증해 주면 기존 개발과 다르게 쉽게 이식이 된다[13][14].

3.3 구현된 미들웨어 인터페이스

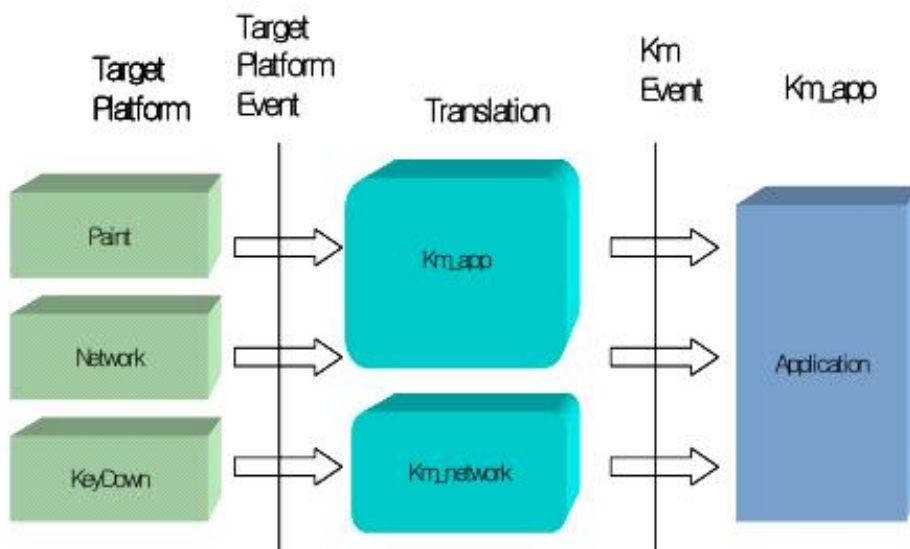
구현에 있어서 변수명은 변수의 타입이 무엇인지 알 수 있도록 헝가리안 표기법을 썼으며, 지뢰 찾기 게임에서 호출되는 메소드는 km_로 시작되고, 플랫폼 쪽에서 호출되는 메소드는 pa_로 시작된다. 구현된 미들웨어를 여기서는 'km_ware'라 칭한다.

본 절에서는 지뢰 찾기 게임 미들웨어 설계로 구현된 인터페이스 부분과 이벤트의 흐름에 대하여 살펴보고, 구현된 Win32 플랫폼 함수와 WI-TOP 플랫폼 함수를 비교해 본다.



[그림 5] 구현된 미들웨어 인터페이스
[Fig. 5] Implemented middleware interface

3.4 이벤트 흐름도(event flow diagram)



[그림 6] 이벤트 흐름도

[Fig. 6] Event flowchart

위 그림 6는 이벤트의 관점에서 이벤트의 흐름을 보였다.

본 연구에서는 PC 대전 게임, 모바일 싱글 게임, PC 대 모바일 대전 게임을 구현하였기 때문에 타겟(target) 플랫폼이 Win32 플랫폼과 WI-TOP 플랫폼이다. 이 타겟 플랫폼으로부터 Paint, network, KeyDown 등의 이벤트가 발생하면 km_ware의 Win32 표준화 부분이나 WI-TOP 표준화 부분을 통한 다. WI-TOP에서 발생한 이벤트이면 WI-TOP을 km_ware에 맞게 변환 과정을 거치고, Win32에서 발생한 이벤트이면 Win32를 km_ware에 맞게 변환 과정을 거쳐서 Application쪽으로 이벤트가 넘어간다.

4. 구현 결과 및 평가

4.1 제안한 미들웨어를 통해 구현된 지뢰 찾기 게임

- PC 플랫폼으로 1:1 대전 게임

km_ware를 통해 구현된 지뢰 찾기 게임을 PC 플랫폼에서 1:1로 대전한 결과이다. 구현된 km_ware를 통해 서버에 연결되고 km_ware를 통해 게임이 플레이 된다.

[표 1] 구현 환경 시스템 사양 1

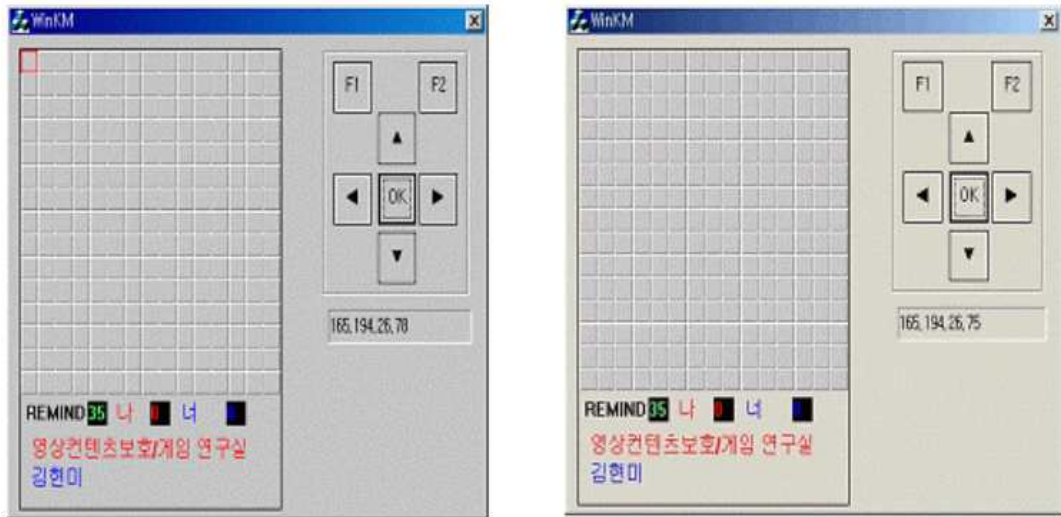
[Table 1] Implementation Environment System Specification 1

- 컴퓨터 사양 : Pentium 4, 512MB memory
- 운영 체제 : Microsoft Window XP
- 구현 프로그램틀 : Microsoft Visual studio 6.0



[그림 7] km_ware서버(왼쪽)측 화면과 클라이언트(오른쪽)측 화면

[Fig. 7] Km_ware server (left) side screen and client (right side) side screen



[그림 8] km_ware를 통하여 구현된 PC대전 게임 초기화면
[Fig. 8] initial screen of PC game played through km_ware



[그림 9] km_ware를 통하여 구현된 PC게임 대전화면
[Fig. 9] PC game match screen implemented through km_ware

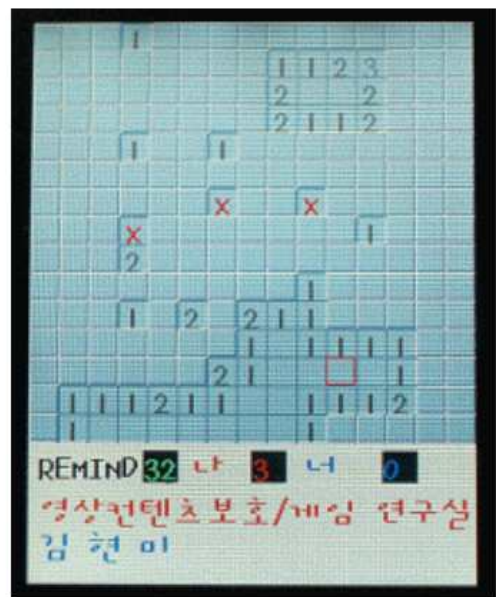
• 모바일 플랫폼 게임

km_ware를 통해 구현된 지뢰 찾기 게임을 모바일 플랫폼에서 플레이한 결과이다. ARM Developer Suite 1.01에서 컴파일 하여 모바일 플랫폼으로 다운로드한다.

[표 2] 구현 환경 시스템 사양 2

[Table 2] Implementation Environment System Specification 2

- 운영체제 : WI-TOP
- 구현 프로그래밍 : ARM Developer Suite 1.01
- 단말기 : SKTT IM-7100 (CPU : MSM-5500)

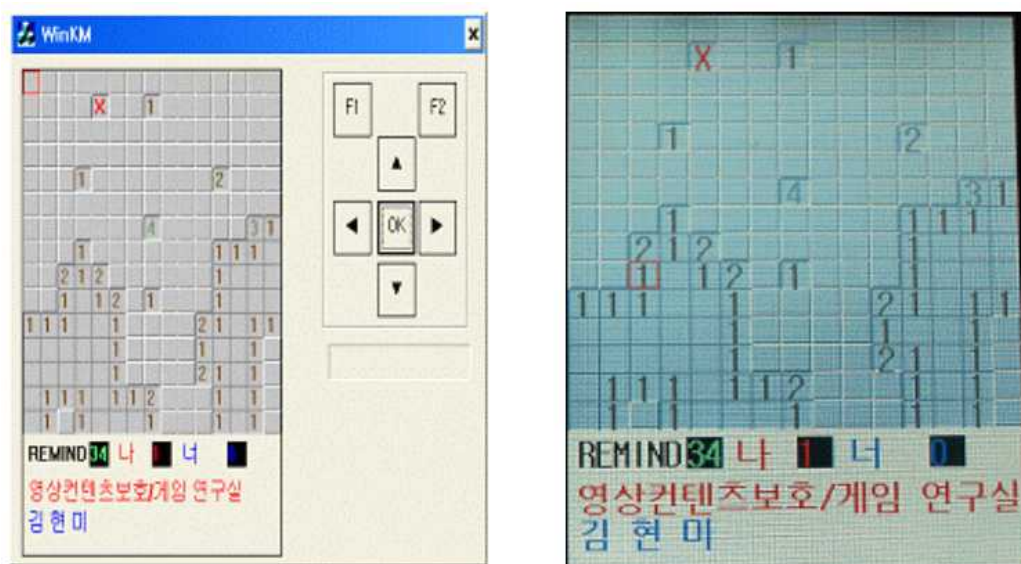


[그림 10] km_ware를 통해 구현된 모바일 게임 초기화면(왼쪽)과 게임 진행 화면(오른쪽)

[Fig. 10] The mobile game initial screen (left) and the game progress screen (right) implemented through km_ware

- PC 플랫폼과 모바일 플랫폼 대전 게임

km_ware를 통해 구현된 지뢰 찾기 게임을 PC 플랫폼과 모바일 플랫폼으로 1:1 대전한 결과이다. 구현된 km_ware를 통해 서버에 연결되고 km_ware 를 통해 게임 플레이가 된다. 표2의 PC 플랫폼 사양과 표3의 모바일 플랫폼 사양으로 구현되었다.



[그림 11] km_ware를 통하여 진행되는 PC와 모바일의 대전 화면

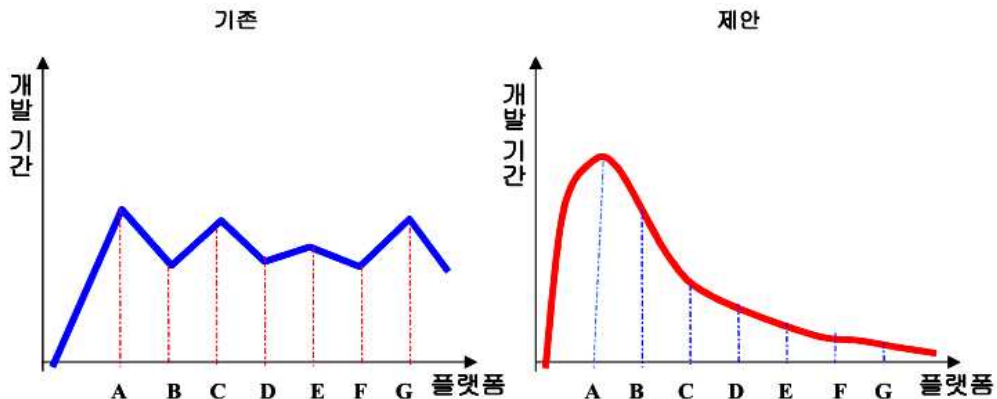
[Fig. 11] Match game between PC and mobile through km_ware

4.2 구현 결과 평가

앞의 4.1절에서 보인 게임들은 플랫폼으로부터 독립성을 보장해주는 미들웨어를 통하여 개발하였기 때문에 기존의 플랫폼에 맞추어 개발하는 플랫폼 종속적인 개발 방법보다 쉽게 게임을 이식할 수 있었다.

그림 19는 한빛소프트 게임 개발팀원 47명과 인프라웨어 개발팀원 50명을 대상으로 설문 조사한 결과를 분석하여 그래프로 나타낸 것이다. 그림 12의 왼쪽 그래프는 기존의 개발 방법을 통하여 동일한 게임을 다양한 플랫폼으로 개발할 때를 나타낸 것이다. 이 방법은 플랫폼이 다양해져도 플랫폼의 특성에 맞추어서 개발해야하기 때문에 개발 시간의 단축을 할 수 없다. 그러나

그림 12의 오른쪽 그래프는 제안한 개발 방법을 통하여 동일한 게임을 다양한 플랫폼으로 개발 할 때로 표준화, 추상화 단계에서 기존 게임 개발 방법 보다 초기 개발 시간이 더 소요 되는 것을 볼 수 있다. 그러나 표준화 작업이 완료되면 기존 게임 개발 기간 보다 짧은 기간 동안 다양한 플랫폼에 쉽게 게임을 이식할 수 있다는 장점이 있다.



[그림 12] 기존 개발 방법과 제안한 개발 방법의 비교
[Fig. 12] Comparison of existing and proposed development methods

5. 결론

본 논문에서는 하나의 게임을 다양한 플랫폼에서 즐길 수 있도록 플랫폼 간 게임 호환성을 보장하는 미들웨어를 제안하였다. 제안한 미들웨어는 플랫폼에 종속적이었던 인터페이스 부분을 포팅 레이어로 구현하여 플랫폼에 독립적으로 게임이 수행되도록 하였다. 이렇게 구현된 포팅 레이어는 다른 플랫폼일지라도 같은 계산식이나 같은 알고리즘의 형태로 구현되는 함수로 코드를 구현하는 경우 코드의 재사용(reusability)으로 게임 개발의 효율성이 커지고, 게임 개발 시간을 단축 할 수 있다.

제안한 미들웨어를 통하여 구현된 지뢰 찾기 게임은 PC 플랫폼으로 1:1 대전, 게임을 다운로드하여 모바일 플랫폼에서 싱글 플레이, PC 플랫폼과 모바일 플랫폼으로 1:1 대전 게임을 할 수 있었다. 이와 같이 제안한 미들웨어를 통하여 게임을 구현한 결과 플랫폼 독립적인 개발 방법이라는 장점으로 인해 게임 개발 시간의 단축을 꽤 할 수 있었다. 그러나 본 연구로 구현된 미들웨어는 단지 게임 포팅에 중점을 두고 구현하였기 때문에 플랫폼의 OS 레벨에 올려졌다. 본 연

구의 미들웨어를 통해 구현된 지뢰 찾기 게임은 다른 게임에 비해 그 규모가 작은 게임이므로 OS 레벨에 미들웨어를 올렸다고 해서 속도에 영향을 받지 않지만 3D게임, 실시간 게임의 경우는 속도가 중요하기 때문에 본 연구와 같이 중간에 거치는 스택(call stack)이 많아지면 상대적으로 속도의 손실이 있게 된다. 그러나 하드웨어에 직접 포팅 레이어를 연결하는 방법을 사용하여 속도 향상을 꽤 할 수 있을 것으로 생각된다.

앞으로의 연구 과제로서 언제, 어디서나, 어느 기기든지 미디어에 구매받지 않고 게임 서비스를 이용할 수 있도록 유비쿼터스란 목적에 보다 적합하도록 연구되고 개발이 이루어져야 할 것이다.

References

- [1] <http://sparcs.org/~dash/case/>
- [2] W. S. Hong, H. W. Lee, Requirement and Priority for Activation of Mobile Game Industry. (2003), pp.25-28.
- [3] S. H. Bae, Mobile Standard Platform Specification (TTAS.KO-06.0036), Telecommunications Technology Association of Korea. (2002), Vol.82, pp.59-66.
- [4] Global Game Market Status and Forecast, IITA, pp.210-221, (2002).
- [5] Worldwide Video-game Forecast and Analysis, IDC 2002-2007, (2003).
- [6] Consumer Warm to Enhanced Video-Game Consoles, Yankee Group, LA, (2003).
- [7] H. M. Kim, C. H. Lee, T. Y. Kim, W. H. Lee, HCI Korea 2004, pp.1906-1911.
- [8] IITA, Global Video Game Status and Outlook. (2003), pp.150-163.
- [9] A. SILBRCHATZ, P. B. GALVIN, G. GAGNE, OPERATING SYSTEM CONCEPTS,, WILEY, USA, (2003).
- [10] S. H. Bae, Telecommunications Technology Association of Korea. (2002), Vol.82, pp.20-30.
- [11] D. S. Joe, J. S. Kim, M. J. Lee, H. S. Kwak, The Research Institute of Industrial Technology Chonbuk National University. (2002), Vol.33, pp.135-141
- [12] H. M. Kim, W. H. Lee, Middleware Design for Porting Game, Review of Korean Society for Internet Information in Autumn. (2004), May, Seoul, Korea.
- [13] B. J. Koo, K. C. Kim, OS - platform structure proposal by development of Mobile Operating System and platform, Korean Institute of Information Scientists and Engineers in Autumn. (2003), October 24-25; Seoul, Korea.