모바일 게임 콘텐츠의 터치스크린 인터페이스 자동 생성 기법

(Automatic Generation Technique of Touch-Screen Interface for Mobile Game Contents)

고 석 훈 † 손 윤 식 ††
(Seokhoon Ko) (Yunsik Son)
박 지 우 †† 오 세 만 †††
(Jiwoo Park) (Seman Oh)

요 약 최근 터치스크린을 탑재한 휴대폰의 출시가 늘어나면서 기존의 모바일 게임 콘텐츠를 터치스크린 콘텐츠로 변환하여 재사용하고자 하는 요구가 증가하고 있다. 하지만, 기존에 개발된 콘텐츠는 터치스크린 인터페이스를 갖고 있지 않기 때문에 터치스크린 콘텐츠로 자동 변환하는 것은 매우 어려운 문제이다. 본 논문에서는 터치스크린 인터페이스가 없는 모바일 게임 콘텐츠에 대하여 키보드 인터페이스 정보를 이용하여 자동으로 터치스크린 인터페이스를 생성하는 방법을 제안한다. 이 방법은 콘텐츠에서 사용되는 키 이벤트의 종류를 분석하여 최적의 스크린 키보드 레이아웃을 구성하고, 터치스크린 이벤트에 대해 자동으로 대응되는 키 이벤트를 발생시켜 키보드 이벤트 핸들러가 실행되는 터치스크린 인터페이스를 생성한다. 끝으로, 실험을 통해 콘텐츠에 최적화된 터치스크린 인터페이스가 생성됨을 확인한다.

키워드 : 모바일 게임, 콘텐츠 변환, 스크린 키보드, 터치스크린 인터페이스

Abstract As touch-screen mobile phones pour into the market, demands for reusing existing mobile game contents by adding a touch-screen interface are increasing. But, the contents which are developed in existing does not have touch-screen interface and in order not to be automatic is difficult with touch- screen contents to convert. From the present paper uses keyboard interface information about the mobile game contents which does not have touch-screen interface and creates touch-screen interface with automatic the method which proposes. This method analyzes the type of key events which are used from the contents the screen keyboard layout of optimum and construct. And, key event where is confronted with automatic occurs about touch screen event and creates the screen interface which executes the keyboard event handler. In the end, experiment leads and the screen interface which has become optimum anger to the contents creating, confirms.

Key words : Mobile Game, Contents Conversion, Screen Keyboard, Touch-Screen Interface

1. 서 론

모바일 게임 시장은 2009년 기준으로 전세계 게임 시장에서 점유율 8.4%, 매출액 71억 달러가 예상되는 유망한 분야이다[1]. 모바일 게임 시장은 서비스 사업자별로 다양한 종류의 모바일 플랫폼을 사용하고 있기 때문에, 한 플랫폼에서 개발된 콘텐츠를 다른 플랫폼에서 재사용하도록 변환하는 작업은 서비스 대상 고객 규모를 확대하는 매우 중요한 작업이다.

각 모바일 플랫폼은 사용 언어가 다르고, 실행 모델이 다르더라도 동일한 휴대폰 기기를 바탕으로 제작되었기 때문에 기능적으로는 매우 유사한 구성요소를 갖는다. 이러한 특징을 바탕으로, 플랫폼 간에 서로 대응되는 구성요소의 맵핑을 찾아 콘텐츠를 자동 변환하는 방법들이 활발히 연구되고 있다.

최근 터치스크린을 탑재한 새로운 휴대폰의 출시가 늘어나고 있다[2]. 터치스크린 휴대폰에는 터치 플랫폼이 탑재되어 있고, 터치 플랫폼에서는 터치스크린 인터페이스를 갖고 있는 콘텐츠만 서비스가 가능하다. 새로운 터치 플랫폼에 대해서도 기존 콘텐츠를 재사용하기 위한 변환 작업이 필요한데, 기존에 개발된 콘텐츠는 터치스크린 인터페이스를 갖고 있지 않기 때문에 터치스크린에 대응되는 맵핑을 찾을 수 없으므로 기존의 변환 방법으로는 터치스크린 이벤트를 처리할 수 없다.

본 논문에서는 터치스크린 처리루틴이 없는 모바일 게임 콘텐츠에 대하여, 키보드 인터페이스를 분석하여 자동으로 터치스크린 인터페이스를 생성하는 방법을 제안한다. 2장에서는 콘텐츠 변환에 관련된 기존 연구를

· 이 논문은 2009 한국컴퓨터종합학술대회에서 '모바일 게임 콘텐츠의 터치스크린 인터페이스 자동 생성기법'의 제목으로 발표된 논문을 확장한 것임

† 정 회 원 : 동국대학교 컴퓨터공학과
shko99@gmail.com
†† 학생회원 : 동국대학교 컴퓨터공학과
sonbug@dongguk.edu
jojaryong@dongguk.edu
††† 종신회원 : 동국대학교 컴퓨터공학과 교수
smoh@dongguk.edu

논문접수 : 2009년 8월 13일
심사완료 : 2009년 9월 23일

정보과학회논문지 : 컴퓨팅의 실제 및 레터 제15권 제11호(2009.11)

살펴보고, 3장에서는 터치스크린 인터페이스 자동 생성 기법을 이용한 터치스크린 콘텐츠 변환기에 대해 설명한다. 4장에서는 제시한 기법의 결과를 실험 콘텐츠를 이용하여 확인하고, 마지막으로 5장에서는 결론과 함께 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 콘텐츠 변환 기술

상이한 플랫폼에 대한 콘텐츠 변환 방법은 콘텐츠를 실행하는 가상기계(virtual machine)를 포팅하는 방법과 소스코드 레벨에서 콘텐츠를 변환하는 방법으로 구분할 수 있다. 가상기계를 포팅 하는 방법은 콘텐츠를 변환하지 않고 재사용 하는 장점이 있지만, 가상기계를 실행하는데 많은 연산을 필요로 하여 실행 성능이 저하되는 단점이 있다[3,4].

콘텐츠의 소스코드를 직접 변환하는 소스레벨 변환방법은 표 1과 같이 언어 변환, 커널 변환, API 변환의 세가지 작업으로 구성되는데, 시스템 구성이 복잡해지지만 범용적으로 적용이 가능한 장점이 있다[5-7].

표 1 소스레벨 변환기의 작업 구성

변환 작업	비 고
언어 변환	대상 플랫폼에서 사용하는 언어로 콘텐츠 언어를 변환. 예) C → Java
커널 변환	대상 플랫폼에 맞춰 콘텐츠의 실행 모델을 변경. 예) 이벤트 처리기 변경
API 변환	대상 플랫폼이 제공하는 API를 사용하여 콘텐츠에서 사용하는 API Wrapper 구현

본 논문에서 다루는 터치스크린 인터페이스 자동 생성기법은 소스레벨 변환의 전처리 단계에 적용되어, 터치스크린 기능이 없는 기존 콘텐츠에 터치스크린 인터페이스를 생성하여, 터치스크린 인터페이스가 필요한 타겟 플랫폼용 콘텐츠로 변환이 가능하도록 한다.

2.2 스크린 키보드

GVM 플랫폼에서 개발된 콘텐츠를 iPhone OS용으로 변환한 마법천자문의 경우에는 그림 1과 같이 디스플레이 영역의 일부에 게임보이 스타일의 버튼을 그려 넣고, 버튼 부분을 터치하면 대응되는 키 이벤트가 발생하도록 하였다[8]. 이 방법은 터치스크린 인터페이스 구현을 손쉽게 자동화할 수 있지만, 디스플레이 영역의 일부를 사용할 수 없다는 제약을 가진다.

최근 출시되는 터치폰에서는 터치 인터페이스가 없는 콘텐츠를 위해 그림 2와 같은 스크린 키보드를 제공한다[9]. 이 방법 역시 콘텐츠 변환 없이 터치스크린을 통한 입력이 가능하지만, 콘텐츠 내용에 관계없이 고정된 형태의 키보드 이미지를 사용하기 때문에 콘텐츠 내용이 가려지고, 사용하지 않는 키까지 모두 나타나는 단점이 있다.

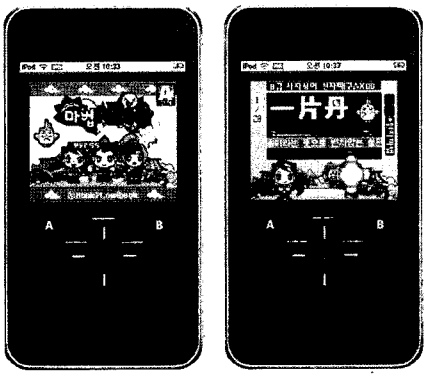


그림 1 iPhone OS용으로 변환된 마법천자문

그림 2 Qbric 터치폰의 스크린 키보드

3. 본 론

3.1 시스템 구성

본 논문에서 제안하는 터치스크린 인터페이스 생성기법은 터치스크린 인터페이스가 없는 콘텐츠 소스코드 C를 입력으로 받아 콘텐츠에서 사용하는 키 이벤트의 종류를 파악하는 키보드 인터페이스 분석 단계, 분석된 키 이벤트의 종류에 따라 알맞은 터치스크린 레이아웃을 선택하는 레이아웃 디자인 단계, 선택된 레이아웃에 따라 대응되는 템플릿코드를 삽입하여 새로운 소스코드 C'을 생성하는 소스코드 생성 단계로 구성된다. 그림 3은 본 논문에서 제안하는 방법을 적용한 콘텐츠 변환기의 시스템 구성도 이다.

3.2 시스템의 동작

(1) 키보드 인터페이스 분석기: 콘텐츠에서 사용 가능한 모든 키코드 집합 $K = \{c_1, c_2, c_3, ..., c_n\}$의 모든 원소에 대하여, 콘텐츠 소스코드 C의 키 이벤트 처리루틴에

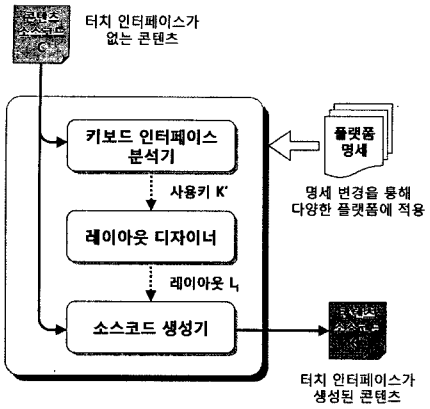


그림 3 터치스크린 콘텐츠 변환기

서 참조하는 키코드를 검색하여, 콘텐츠에서 사용되는 키코드 집합 $K' = \{c_p, c_q, ..., c_r\}$을 생성한다.

콘텐츠에서 사용 가능한 키코드 집합 K는 플랫폼 명세를 통해 터치스크린 콘텐츠 변환기에 전달되며, 소스코드에 사용되는 키코드의 검색은 어휘분석을 통해 키코드와 일치하는 심볼을 찾는 방법을 사용한다.

(2) 레이아웃 디자이너: 개발자에 의해 플랫폼 명세에 정의되어있는 터치스크린 키보드 레이아웃 집합 $S = \{(K_1, L_1),(K_2, L_2),...,(K_m, L_m)\}$에서 $K' \subset K_i$을 만족하는 가장 작은 (K_i, L_i)을 찾는다.

터치스크린 레이아웃 집합 S의 원소는 표 2와 같이 키코드 조합 K_i와 그에 대응되어 스크린을 분할하여 대응되는 키코드의 위치를 나타내는 스크린 레이아웃 L_i의 쌍으로 이루어진다. 키코드 조합은 이론적으로 $2^n - 1$가지가 존재하지만, 실제 콘텐츠는 주로 사용되는 몇 가지 종류의 전형적인 키코드 조합을 사용하므로 모든 조합에 대해 레이아웃을 정의할 필요는 없다.

표 2 터치스크린 키보드 레이아웃 S의 예

키코드 집합 K_i	터치스크린 레이아웃 L_i	키코드 집합 K_i	터치스크린 레이아웃 L_i
{ OK }	OK	{↑, ↓, OK }	↑ / OK / ↓
{←, →}	← | →	{←, →, OK, CL }	← | OK / CL | →
{↑, ↓}	↑ / ↓	{←, →, ↑, ↓, OK }	↑ / ← | OK | → / ↓
{←, →, OK }	← | OK | →	{←, →, ↑, ↓, OK, CL }	↑ / ← | OK / CL | → / ↓

(3) 터치스크린 인터페이스 생성: 플랫폼 명세에 정의되어 있는 템플릿 코드 중 앞 단계에서 선택된 레이아웃 L_i에 대응되는 코드를 콘텐츠에 적용하여 터치스크린 인터페이스 기능이 구현된 콘텐츠 소스코드 C'을 완성한다.

터치스크린 인터페이스는 콘텐츠 실행 중에 터치스크린 이벤트가 발생하면 자동으로 대응되는 키 이벤트를 발생시켜 키보드 이벤트 핸들러 루틴을 실행하도록 한다(그림 4).

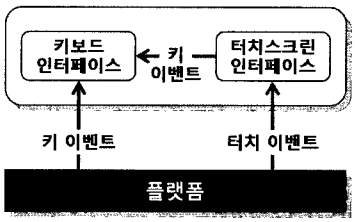


그림 4 터치스크린 인터페이스가 추가된 콘텐츠

4. 실험 및 분석

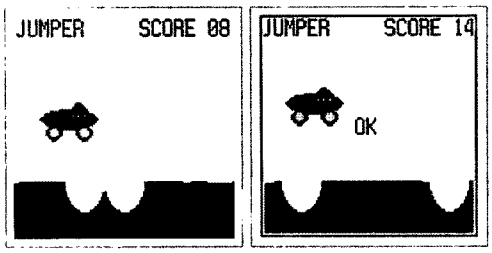
실험은 MobileC 규격을 사용하는 GVM 플랫폼에서 구현된 4개의 테스트 콘텐츠에 대해 터치스크린 콘텐츠 변환기를 적용하여 그 결과를 관찰하였다. 사용 가능한 모든 키코드 집합 K = {←, →, ↑, ↓, OK, CLR}이며, 터치스크린 레이아웃은 표 2에 정의된 집합 S를 적용하였다.

각 테스트 콘텐츠는 다양한 레이아웃이 선택되도록 서로 다른 종류의 키코드 조합을 갖는 콘텐츠를 선택하였다(표 3). Jumper, Submarine과 Puzzle은 실험을 위해 제작된 콘텐츠이며, Aiolos는 GNEX SDK에서 제공하는 상용 수준의 시험 콘텐츠이다[10].

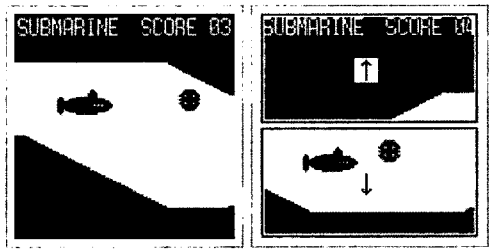
콘텐츠 변환 결과 4개의 시험 콘텐츠는 프로그래머의 개입 없이 모두 자동으로 변환되었다. 실험에서 사용한 터치스크린 인터페이스에는 콘텐츠 사용자의 편의를 위한 부가기능으로 터치스크린 레이아웃을 보여주는 기능이 포함되어 있다. 그림 5는 실험 콘텐츠의 실행 화면이다. 그림에서 보는 바와 같이 콘텐츠에서 사용하는 키의 종류에 최적화된 터치스크린 레이아웃이 생성되었음을

표 3 테스트 콘텐츠에서 사용하는 키의 종류

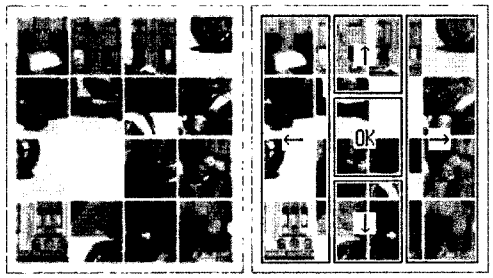
테스트 콘텐츠	사용하는 키
Jumper	{ OK }
Submarine	{ ↑, ↓ }
Puzzle	{ ←, →, ↑, ↓ }
Aiolos	{ ←, →, ↑, ↓, OK }



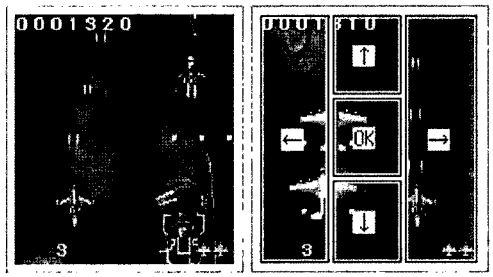
(1) Jumper



(2) Submarine



(3) Puzzle



(4) Aiolos

그림 5 테스트 콘텐츠 실행화면
(좌: 일반 실행화면, 우: 레이아웃 안내화면)

확인할 수 있다.

Puzzle의 경우, 콘텐츠에서 네 개의 방향키만 사용하지만, OK키가 포함된 레이아웃이 선택되었다. 그 이유는 집합 S에 네 개의 방향키 만으로 구성된 레이아웃은 존재하지 않기 때문에, 네 개의 방향키가 포함된 가장 작은 레이아웃이 선택된 결과이다. 역시 주어진 명세를 기준으로 최적의 결과가 선택되었다.

변환된 콘텐츠에는 공통적으로 터치스크린 인터페이스 기능을 수행하는 6개의 함수가 구현되어있는 템플릿 소스코드가 삽입되었다. 템플릿 소스코드의 크기는 약 255라인 정도인데, 선택된 레이아웃의 종류에 따라 #ifdef 문으로 구분되어 있으므로 실제 컴파일 되는 소스코드의 크기는 그보다 훨씬 적다.

콘텐츠의 종류에 관계없이 추가되는 함수의 개수가 같고, 삽입된 소스코드의 양이 적다는 것은 변환된 콘텐츠에 생성된 터치스크린 인터페이스가 콘텐츠의 복잡도와 관계없음을 의미한다. 다시 말해 콘텐츠의 크기 증가 문제 및 속도 저하문제를 발생하지 않도록 효과적으로 구현되었음을 의미한다.

5. 결론 및 향후 연구

본 논문에서는 터치스크린 인터페이스가 구현되지 않은 모바일 게임 콘텐츠에 대해 자동으로 터치스크린 인터페이스를 생성하는 기법을 제안하였다. 이 방법은 콘텐츠 내용에 따라 최적의 레이아웃을 선택하고, 효율적인 소스코드를 생성하는 장점을 갖는다.

본 논문에서 제시한 자동화된 터치스크린 인터페이스 생성기법을 이용하면 기존에 개발된 많은 수의 모바일 게임 콘텐츠를 손쉽게 터치폰 용으로 변환하여 서비스할 수 있으므로 모바일 게임 산업 발전에도 큰 도움이 될 것으로 생각한다.

현재 다양한 플랫폼 규격의 콘텐츠에 대한 명세를 적용해 나가는 연구가 진행되고 있으며, 향후에는 단순한 터치뿐만 아니라 제스처가 포함된 인터페이스를 생성하는 방법과 콘텐츠 내부의 상태 변화를 파악하여 상황에 따라 동적으로 레이아웃을 변경하는 방법에 대한 연구를 진행할 예정이다.

참 고 문 헌

[1] Korea Creative Content Agency, *2007 Korea Game White Paper*, http://www.gitiss.org, pp.652-684, 2008.

[2] M. J. Kim, J. H. Yoon, "Implementation of Mobile Game Interface through an Operating Interface Analysis of Touchscreen Devices," *Journal of Korean Society of Design Science*, vol.22, no.1, pp.231-244, 2009. (in Korean)

[3] J. Cho, C. Hong, Y. Lee, "Implementation of Automatic Translation Tools of GVM to BREW Contents in Mobile Environment," *Korea Multimedia Society*, vol.9, no.2, pp.38-49, 2005. (in Korean)

[4] Script Creation Utility for Maniac Mansion Virtual Machine, http://www.scummvm.org.

[5] S. Park, H. Kwon, Y. Kim, Y. S. Lee, "Design and Implementation of GVM-to-MIDP Automatic

Translator for Mobile Game Contents," *Journal of Game Society*, vol.3, no.1,2, pp.5-12, 2006. (in Korean)

[6] K. B. Kang, D. H. Kang, C. P. Hong, J. M. Ryu, J. H. Lee, J. H. Yoon, J. W. Jwa, "Development of Conversion Solutions for Interoperability of Applications on Different Mobile Internet Platform," *Journal of Korea Contents Society*, vol.7, no.4, pp.1-9, 2007. (in Korean)

[7] S. M. Oh, S. L. Yoon, Y. S. Son, J. W. Park, *Development of Mobile C-WIPI C Source Converter*, Project Report, Industry-Academy Cooperation Foundation of Dongguk Univ., 2008.

[8] C. J. Lee, *The Proposal which gives at the Mobile Game Enterprise and the Case of the Magic Thousand-Character Text Game*, http://blog.dreamwiz.com/chanjin/9646413, 2009.

[9] Pantech, iSKY IM-R470S Qbric, http://www.isky.co.kr

[10] Sinjisoft, http://www.gnexclub.com