

논문 2008-4-13

객체지향 소프트웨어 공학 기법을 이용한 리듬 게임의 개발

Development of Rhythm Game based on Object-Oriented Software Engineering

오영석*, 임준석**, 황성운***

Young Seok Oh*, Joon Suk Lim**, Seong Oun Hwang***

요 약 최근에 컴퓨터 기술이 음악, 게임 등과 같은 우리 일상 생활에 활발히 적용되고 있다. 대표적인 예가 리듬 게임이라고 할 수 있다. 본 논문에서는 PC 및 북 인터페이스를 갖춘 사용하기 쉬운 리듬 게임을 객체지향 소프트웨어 기법을 적용하여 개발한다.

Abstract Recently computer technologies have been actively applied in our daily life such as music and game, etc. One representative example is rhythm game. In the paper, we develop a new rhythm game using PC with two drums attached based on object-oriented software engineering.

Keywords : Rhythm Game, Software Engineering, Direct X, Object-Oriented

1. 서 론

리듬 게임이란 음악에 맞춰서 손 혹은 발 등 몸의 특정 부위를 사용해 조작하는 게임을 총칭하는 말이다. 음악 게임이라고도 한다. 1996년 소니 컴퓨터 엔터테인먼트사의 파라파 더 래퍼를 시작으로 다양한 리듬 게임이 출시되었는데, 특히 1998년 코나미사에서 아케이드 게임 장용으로 출시한 비트매니아를 시작으로 음악 게임 열풍이 불기 시작해 현재는 다양한 음악 게임 시리즈가 출시되었다.

현 시점에서 제공되는 리듬 게임의 경우 한정된 장소(게임장)나 콘솔 게임기(닌텐도, psp, ps 등)를 구매한 후 해당 소프트웨어를 구매하여 플레이 또는 인터넷에서 서비스하고 있는 게임을 가입하여 게임을 하는 경우가 대부분이다. 현재의 게임들은 대부분 해당 프로그램에서 제공하는 노래만으로 게임을 해야 한다는 제약점이 있다.

또한 특별한 장소 나 기기를 이용해야만 한다는 점도 큰 제약점이라고 할 수 있다. 본 논문에서는 PC 및 북 인터페이스를 활용한 새로운 리듬 게임을 개발함으로써 이러한 제약점을 극복하고자 한다. 즉, 본 리듬 게임을 통해 사용자들은 자신이 원하는 노래에 원하는 타격을 북을 통해 적용할 수 있다.

현재 이용 가능한 리듬 게임들을 간단히 소개하면 다음과 같다[1].

비트 매니아[2]는 1997년 일본의 코나미사에서 개발한 음악 게임이다. 게임방식은 음악에 맞춰 화면에 표시되는 채보에 따라 5개의 건반과 턴테이블을 이용해 플레이 한다. 조작을 잘 하면 화면 하단에 표시되는 게임포인트가 올라가는데, 곡이 끝났을 때 이 게임포인트가 일정량 이상이면 다음 스테이지로 넘어갈 수 있고 일정량 이하면 그 스테이지에서 게임이 끝나는 방식이다. 오락실 아케이드용 beatmania 오리지날 시리즈와 플레이스테이션 용인 beatmania APPEND 를 13편씩 발매 발표했으며 그 외에 콘솔게임기용으로 4개를 발매하였다.

키보드 매니아[3]는 일본의 코나미사에서 개발한 음악

*홍익대학교 컴퓨터정보통신공학과 학생

**홍익대학교 컴퓨터정보통신공학과 학생

***홍익대학교 컴퓨터정보통신공학과 교수

접수일자:2008.7.15, 최종수정일자:2008.8.2

게임으로, 오락실 아케이드용만 발매되었다. 이름 그대로 전자 키보드(24개의 키)를 연주하는 시뮬레이션 게임이다. 악기를 다루는 감각을 잘 살려냈다는 평가를 받지만, 높은 난이도로 일부 사용자층에게만 호응을 얻었다. 3rd MIX를 마지막으로 개발이 중단되고 말았다.

디제이맥스(DJMAX)[4]는 한국의 펜타비전에서 개발한 음악 게임이다. 온라인 게임판과 플레이스테이션 포터블판으로 발매되었다. 게임방식과 키입력 방식이 비트매니아와 같다. 온라인판은 현재 서비스가 중지되었다.

드럼매니아[5]는 일본의 코나미사에서 개발한 음악 게임으로, 오락실 아케이드용만 발매되었다.

드럼 컨트롤러를 이용해서 곡의 드럼 파트를 연주하는 게임이다. 드럼 파트는 아마하에서 제작한 전자 드럼 킷을 개조한 형태이며, 게임 방식은 비트매니아 시리즈와 동일하다.

테고의 달인[6]은 일본의 남코가 개발한 게임으로, 오락실용과 콘솔게임용으로 발매되었다. 커다란 북이 설치된 기기를 통해 게임을 플레이하며 11종류의 게임이, 가정용 게임기(플레이스테이션2)용의 경우 10종류의 게임이 나왔으며 타타콘이라 불리는 전용 컨트롤러로 북을 두드리는 느낌을 살리고 있다. 그리고 닌텐도 DS 판은 터치 스크린 기능을 이용한 전용 컨트롤러인 파치펜을 통해 아케이드 판의 느낌을 살리려 한 점이 특징이다. 그러나, 플레이스테이션 포터블 판은 하드웨어 구조의 한계로 전용 컨트롤러가 없다.

II. 본 론

본 연구에서는 기존 리듬 게임들을 벤치 마킹을 한 결과, 사용자에게 간단한 입력장치를 통해 함께 배포함으로써 일반 가정집에서도 PC를 통하여 편하게 즐길 수 있고 위의 제품에서 장점을 추려서 리듬 게임을 개발하기로 결정하였다.

개발 프로세스 모델로는 프로토타이핑 모델 (Rapid Prototyping Model)을 사용함으로써 사용자의 요구를 더 정확히 추출하고, 구현 기술의 타당성 검증, 인터페이스 시험 제작 등을 할 수 있었다. 아래에서 각종 UML (Unified Modeling Language) [7] 형태의 다이어그램 작성을 위해서 HiMEM ver 1.0 [8]이라는 UML 툴을 사용하였다. 본 연구에서 UML을 이용한 객체 지향 소프트웨어

어 공학 기법을 사용하여 개발하게 된 이유는 다음과 같다. 기존의 게임들을 분석한 결과, 기능의 추가, 삭제, 확장이 자주 일어나기 때문에 전통적인 소프트웨어 방법론(구조적 방법, 자료구조 중심 방법 등)으로 개발 및 유지보수를 하게 되면 많은 시간과 비효율성을 초래한다고 생각하였다. 하지만 객체 지향 방법론의 경우 사용사례, 클래스 다이어그램을 통해 정보은닉화, 모듈간 응집력, 재사용성(차후 다른 장르의 게임 개발시 유용함)을 증대하는 장점이 있어 본 게임 개발에 적합하다고 판단하였다.

1. 연구 절차

가. 문제 정의: 벤치 마킹을 통하여 이미 나와 있는 게임들을 플레이 해보고 리듬 게임이 제공할 기능 및 범위를 정하였다.

나. 요구 사항 분석 및 추출: 요구 사항 추출을 위해서 액터 (actor), 시나리오 등을 찾고 이를 바탕으로 사용사례 (use case) 다이어그램 및 기술서를 작성하였다. 또한 객체 후보들 및 이들 사이의 연관 관계들을 모두 찾은 다음 이들을 정제하여 클래스 (class) 다이어그램을 생성하였다. 또한 사용사례별로 순서 (sequence) 다이어그램을 작성하여 시스템의 동작을 정형화하고 객체들 사이의 메시지 교환을 표현하였다.

다. 설계: 분석 단계의 클래스 다이어그램에서는 주로 응용 객체들과 그들 사이의 속성, 메소드를 파악하였다. 설계 단계에서는 이들 응용 객체를 좀더 상세화 (속성 및 메소드를 추가하고 각각 가시성 및 파라미터를 명시)하고, 응용 객체 이외에도 시스템을 완성시키는데 필요한 솔루션 객체들을 찾아 클래스 다이어그램에 반영하였다. 또한 시스템을 구성할 하드웨어 및 소프트웨어 플랫폼, 재사용할 라이브러리 등을 결정하였다.

라 구현: 작성된 클래스 다이어그램을 바탕으로 C++ 언어를 이용해 프로그램을 구현했다.

마. 테스트: 완성된 시스템을 일반 사용자들이 주로 기능 사양에 근거하여 테스트하는 블랙박스 (blackbox) 테스트 기법을 사용했다.

2. 문제 정의서

이미 시판되고 있는 리듬 게임의 기능을 분석함으로써 다음과 같은 기능 위주의 문제 정의서를 도출하였다.

가. 시스템이 제공하는 악보를 보고 사용자가 북을 쳐서 점수를 계산하는 게임을 개발한다.

나. 사용자는 노래를 선택하여 게임을 한다

다. 노래와 함께 나오는 화면에서 버튼을 누를 수 있다. 버튼을 누르면 소리가 나온다.

라. 북을 쳤을 경우 게임 화면에서 나오는 악보의 위치가 시스템이 정한 위치에 50%이상 일치했을 때 '지화자'를 출력하고 점수를 올린다.

마. 북을 쳤을 경우 게임 화면에서 나오는 악보의 위치가 시스템이 정한 위치에 10~50% 일치했을 때 '얼 수'를 출력하고 점수를 올린다.

바. 그 외에 북을 쳤을 경우 '아이구'를 출력하고 점수를 내린다.

사. 노래가 끝났을 경우 점수를 출력한다.

아. 해당 점수가 최고 점수일 경우 결과를 저장한다.

자. 노래를 조회할 수 있다. 조회 중 각 노래에서 최고의 점수를 기록했던 점수를 보여준다.

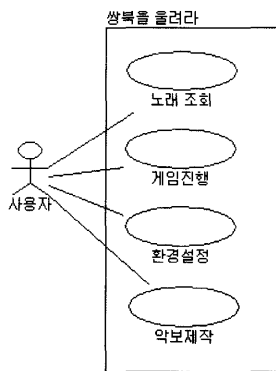
차. 게임 도중에 게임을 종료할 수 있다.

카. 환경 설정 화면을 통해 게임 속도 및 북소리를 변경할 수 있다.

타. 사용자는 노래를 제작 및 추가할 수 있다.

3. 사용사례 다이어그램

문제정의서를 바탕으로 분석한 결과, 게이머라는 액터와 네 가지의 사용사례 (노래조회, 게임진행, 환경설정, 악보제작)를 도출하였다.



4. 클래스 다이어그램

아래는 클래스 다이어그램 및 각 클래스들의 기능을 나타낸다. 리듬 게임의 특성상, 반응이 느려서는 안되다는 제약 조건과 일반 PC에서 구동되어야 하는 조건을 만족시키기 위해 멀티미디어와 게임에 강한 다이렉트 X를 사

용 하였다. 재사용 라이브러리로 DIRECT X 9.0 (2007. 8) SDK를 사용하였으며 이 라이브러리로 Cinput, CSound, CSurface 객체를 생성하여 사용하였다.

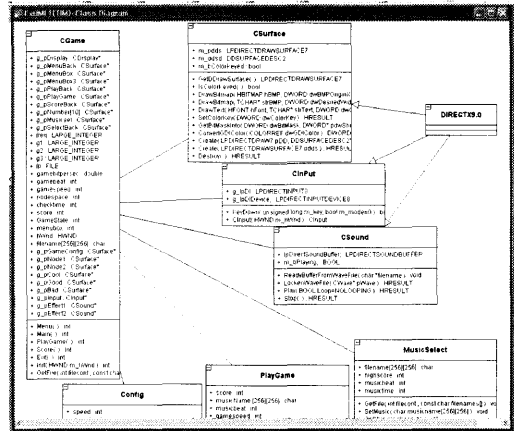


그림 2. 클래스 다이어그램
Fig. 2. Class Diagram

표 1. 클래스 이름 및 기능

Table . Descriptions of Classes

클래스	기능
CGame	게임을 하기 위한 메인 클래스
Config	환경설정 관련 클래스
PlayGame	게임 플레이 관련 클래스
MusicSelect	음악 선택 관련 클래스
CSurface (DIRECTDRAW)	DIRECTX->DRIEICTDRAW를 이용하여 표면을 생성하는 클래스
CInPut (DIRECTINPUT)	DIRECTX->DRIEICTINPUT를 이용하여 입력을 위한 클래스
CSound (DIRECTSHOW)	DIRECTX->DRIEICTSOUND를 이용하여 사운드 출력을 위한 클래스
DIRECTX 9.0	재사용 라이브러리

5. 순서 다이어그램

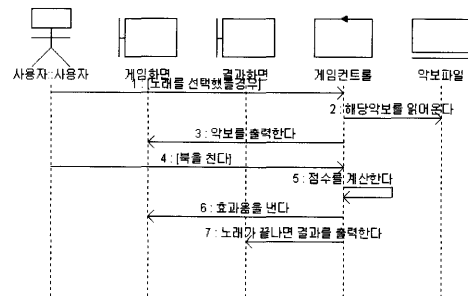


그림 3. 순서 다이어그램
Fig. 3. Sequence Diagram

위에서 도출된 사용사례별로 각각의 순서 다이어그램을 작성하였다. 그림3은 '게임진행'을 나타내는 순서 다이어그램을 나타낸 것이다. 사용자가 노래를 선택했을 경우, 게임 컨트롤은 해당 노래의 악보와 노래를 읽은 후 악보를 화면에 출력한다. 사용자가 북을 쳤을 경우, 게임 컨트롤은 해당되는 입력에 대하여 점수를 계산하고 해당 효과음과 점수를 출력한다. 노래가 종료되었을 경우 점수결과를 화면에 출력한다. 보다 자세한 설명은 구현 부분의 그림 9번과 10번을 참고한다.

6. 자료 저장소

자료 저장소로 악보 및 점수 파일은 텍스트 파일, 음악 파일은 mp3, 사운드 파일은 wav를 사용하였다.

표 1. 자료 저장소
Table 1. Data Storage

종류	자료 저장소
노트(악보)	txt
점수	txt
배경음	mp3
효과음	wav

III. 구현 및 결과

1. 개발 환경

시스템은 Windows XP SP2에서 개발되었으며 입력 장비로는 일반 키보드 외에도 북 인터페이스를 사용할 수 있다. 북 인터페이스로 저가의 모형 북에 진동센서를 북의 헤드안쪽 부분에 부착하고 타격이 있을 때마다 진동을 감지하여 지정한 아스키코드를 USB포트로 전송한다.



그림 4. 북 인터페이스
Fig. 4. Drum Interface

악보 제작 과정은 그림5 와 같이 악보제작 프로그램을

실행시켜 사용자가 직접 악보를 그린다. BPM(Beat Per Minute)과 시간(초단위)을 설정하고 한 노트당 16분음표로 보고 노래에 맞는 박자를 클릭하여 추가하고 저장하면 텍스트 파일로 저장된다.

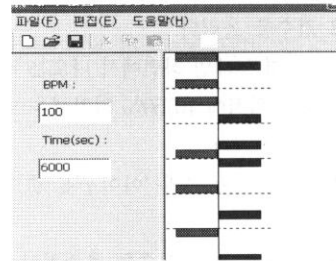


그림 5. 악보 제작 프로그램
Fig 5. Music Composition Program

2. 사용자 인터페이스

가. 메인화면

메인화면에서 게임조회, 환경설정, HowToPlay,나가를 선택할 수 있다.



그림 6. 메인화면
Fig. 6. Main Menu

나. 환경설정

환경설정 화면에서 게임속도와 북칠 때 나는 효과음을 설정할 수 있다.



그림 7. 환경설정
Fig. 7. Configure

다. 음악 선택

악보와 노래가 포함된 폴더에서 정보를 읽어와 노래를 조회하고 정보를 표시한다.

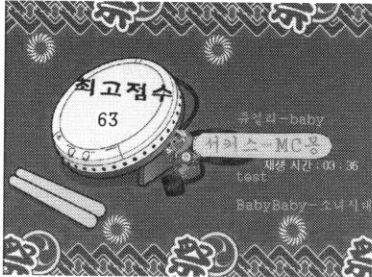


그림 8. 음악 선택
Fig. 8. Music Select

라. 게임 진행 화면

현재 스코어와 노래이름 그리고 악보가 재생되고 키보드 입력에 따라 해당 노트의 점수를 계산하여 출력한다.

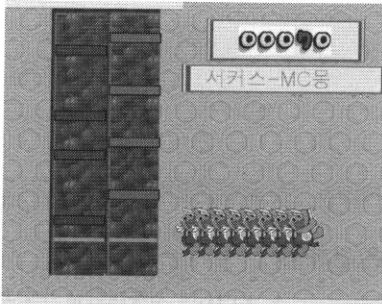


그림 9. 게임 진행
Fig. 9. Game Play

마. 점수 계산화면

현재까지 플레이한 점수와 그 음악의 최고점수를 출력해준다.



그림 10. 점수 출력
Fig. 10. Display Score

3. 테스트

BPM를 계산함에 있어서 시간적인 오류가 없는지 테스트하기 위하여 게임 플레이 동안 설정된 시간으로 잘 수행되는지 시간을 체크하였다. 속도가 100일 경우 0.024초에 노트의 움직임이 1픽셀씩 움직여야 한다는 계산이 나와서 해당속도가 잘 나오고 있는지 측정하였다.

시간의 차 * 10 ⁸	실제 실행 시간	노트의 움직임
2399997.765079	시간 : 10.798451	노트 : 70 키보드 : 409
2400053.638102	시간 : 10.822452	노트 : 70 키보드 : 410
2400025.701591	시간 : 10.846454	노트 : 70 키보드 : 411
2399969.828568	시간 : 10.870456	노트 : 70 키보드 : 412
2400053.638102	시간 : 10.894456	노트 : 70 키보드 : 413
2400025.701591	시간 : 10.918458	노트 : 70 키보드 : 414
2399969.828568	시간 : 10.942460	노트 : 70 키보드 : 415
2399997.765079	시간 : 10.966461	노트 : 70 키보드 : 416
2400025.701591	시간 : 10.990462	노트 : 70 키보드 : 417
2399969.828568	시간 : 11.014464	노트 : 70 키보드 : 418
2399997.765079	시간 : 11.038464	노트 : 70 키보드 : 419
2399969.828568	시간 : 11.062466	노트 : 70 키보드 : 420
2399997.765079	시간 : 11.086467	노트 : 70 키보드 : 421
2400081.574614	시간 : 11.110468	노트 : 70 키보드 : 422

그림 11. 테스트
Fig. 11. Test

노트가 1픽셀씩 움직일 때 $\pm(100 \times 10^{-8})$ 초 오차가 발생하는 것을 확인할 수 있다. 이 결과로 플레이시간이 긴 노래에서는 노래 후반부에 가서 어느 정도 지연이 생성될 수도 있다는 결론이 나왔다.

4. 상용 게임과의 비교

표 2. 상용 게임과의 비교
Table 2. Comparison

	비트 매니아	DJ MAX	드럼 매니아	태고의 달인	쌍복을 울려라
사용된 악기	전용 키보드	전용 키보드	고무 드럼모형	큰 북 모형	북 모형 (키보드, 마우스도 지원)
PC용 프로그램 존재유무	0	0	X	X	0
종류별 인터페이스 존재	오락실, 콘솔, PC	PC (비트매니아 것을 사용)	오락실	오락실, 콘솔	PC
가정용 (콘솔, PC용)의 가격	5'6만원	5'6만원	X	4'6만원	만원 이하
확장성	중간	중간	낮음	낮음	높음
재사용성	X	X	X	X	0

다음은 상용 리듬 게임들과 본 연구에서 개발된 게임을 가격 측면에서 비교한 것이다. 본 연구에서 개발한 계

임의 경우 가정용 PC 및 저가의 북을 사용함으로써 사용자가 적은 비용으로 게임을 즐길 수 있다는 것을 알 수 있다. 물론 본 게임은 북 뿐만 아니라 마우스, 키보드 등의 기존의 PC 인터페이스도 지원할 수 있다. 또한 사용자의 취향에 맞게 북의 개수를 늘리거나 기능 추가가 용이하다. 기능별로 CSurface, CInput, CSound 등 클래스를 세분화하여 개발하였기 때문에 차후에 전혀 다른 장르의 게임을 개발한다고 하여도 개발된 모듈의 재사용을 통해 빠르게 개발할 수 있다.

IV. 결 론

본 연구에서는 조작과 제작이 간단한 리듬 게임을 객체 지향 소프트웨어 공학 기법을 적용하여 개발하는 것이 목적이었다. 현재 간단하게 악보를 제작할 수 있고 이를 간단한 조작으로 게임을 즐길 수 있다.

향후 연구 계획으로, 현재의 악보 제작 과정에서 악보 중간에 노래 속도를 변경할 수 있게 하거나 앞에서 언급한 BPM 오차를 수정이 가능하도록 악보 제작 과정을 보

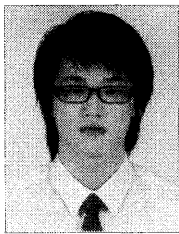
완할 예정이다. 또한 북의 개수를 1~5개까지 지원할 수 있도록 추가하고, 특히 프로그램이 스스로 노래를 분석하고 해당 노래를 즐길 수 있는 악보를 자동적으로 생성하는 부분을 구현하여 이를 게임에 적용시키면 어느 누구라도 자신이 원하는 노래를 쉽게 추가하여 게임을 즐길 수 있게 되지 않을까 한다.

참 고 문 헌

- [1] 음악게임 (본문,목차), <http://ko.wikipedia.org>
- [2] 비트매니아, <http://www.konami.jp/am/bm>
- [3] 키보드매니아, <http://www.konami.jp/am/keyboard>
- [4] DJMAX, <http://www.djmax.co.kr>
- [5] 드럼 매니아, <http://dmv4.uniana.com>
- [6] 태고의 달인, <http://taiko.namco-ch.net>
- [7] UML ver 2.1.2, www.uml.org
- [8] HiMEM ver 1.0, <http://selab.hongik.ac.kr>

저자 소개

오 영 석(정회원)



• 2009년 2월 홍익대학교 컴퓨터정보통신공학과 학사 졸업 예정
 <주관심분야 : 임베디드 프로그래밍, 소프트웨어 프로그래밍>

• 황 성 운(정회원)



• 1993년 서울대학교 수학과 학사 졸업
 • 1998년 포항공과대학교 정보통신학과 석사 졸업
 • 2004년 한국과학기술원 전자전산학과 박사 졸업
 • 2008년 현재 홍익대학교 컴퓨터정보통신공학과 교수

<주관심분야 : 정보보호, 계산이론>

임 준 석(정회원)



• 2009년 8월 홍익대학교 컴퓨터정보통신공학과 학사 졸업 예정
 <주관심분야 : 임베디드 프로그래밍, 게임프로그래밍, 데이터베이스>