

한글 자음과 모음결합을 이용한 학습용 퍼즐게임 구현

조재영*, 김윤호**

요약

본 논문에서는 한글의 자음과 모음을 별도로 분류 한 후, 자음과 모음을 실시간으로 조합하여 단어를 만드는 퍼즐게임을 구현하였다. 단어 조합기는 API 에서 지원하는 에디터를 이용하여 구현하였고, 효율적인 조합단어의 검색을 위하여 초기 합성소 자음기반 방식을 이용하였다. 구현된 한글 조합 퍼즐게임 은 아동들의 단어 학습 능력의 향상과 한글과 친해질 수 있는 기대 효과를 갖는다.

Implementation of Learning Puzzle Game by using Combination of Korean Alphabet

Jae-young Jo*, Yoon-ho Kim**

Abstract

In this paper, learning oriented puzzle game which based on combination of consonant and vowel of Korean alphabet is implemented. Firstly, consonants and vowels of Korean alphabet are classified separately, and then reconstructed a word in real time. Word combinator is utilized by API based edit window and, in order to effective retrieve, initial combined syllable consonant based method is involved. Implemented Korean puzzle game can be used for improving the words learning capability for children.

Key words : puzzle game, combination of Korean alphabet, consonants and vowels

1. 서론

한국의 인터넷 인프라 발전으로 인하여 다양한 온라인 네트워크 게임의 제작이 나날이 늘고 있다. 과거에는 일본을 중심으로 하는 아케이드 게임시장이나 가정용 콘솔 게임시장이 게임시장의 대부분을 차지하였지만 국내 게임 제작 업체들의 다양한 분야의 네트워크 게임의 성공으로 인해 국내에서의 게임시장 또한 세계 미디어 시장에서 가장 큰 비중을 차지하는 대표적인 산업으로 부상하고 있다. 이같이 한국의 게임 산업이 발전할 수 있게 된 계기는 고속 인터넷

인프라를 갖추고 있는 인터넷 PC방의 확산에 따른 것으로 위성, 케이블TV망, 고속모뎀 등을 이용한 초고속 서비스가 확대되었기 때문이다. 더구나 1가구 1PC 시대에 살고 있는 한국의 게임 유저들은 어려서부터 쉽게 컴퓨터로 게임을 다 운받고 플레이 할 수 있다. 하지만 자극적이며 폭력적인 최근의 게임들은 유년기의 아이들에게 나쁜 영향을 미칠 수도 있다. 본 논문에서는 한글의 자음과 모음을 조합하여 명사에 해당 하는 단어를 만드는 퍼즐게임을 제작함으로써 유년기의 아이들이 즐길 수 있는 교육적인 게임의 구현을 목적으로 한다. 본 논문에서의 시스템은 향후 구축될 가상공간 아바타 생성 및 행동양식 데이터베이스 구축 과정을 위해서도 필요하다.

2. 퍼즐 게임의 유형

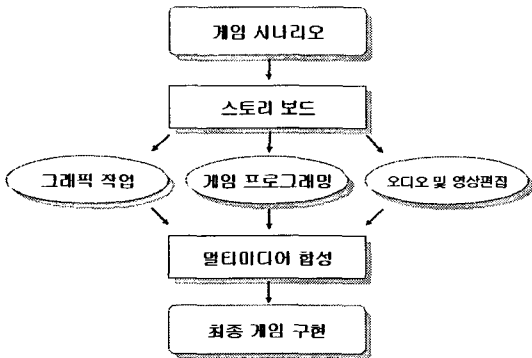
게임의 장르는 크게 롤플레이팅, 액션, 대전전략, 슈팅, 스포츠, 시뮬레이션, 어드벤처, 퍼즐 등으로 나눌 수 있다. 퍼즐게임은 논리적 사고에 기

※ 제일저자(First Author) : 조재영
접수일자:2006년09월22일, 심사완료:2006년11월18일
* 목원대학교 대학원 IT공학과 박사과정
racemaster@mokwon.ac.kr
** 목원대학교 컴퓨터공학부 교수
□ 본 연구는 산업자원부의 지역혁신 인력양성 사업의 연구결과로 수행되었음

반 한 문제 해결이 원칙으로, 컴퓨터게임 이전부터 있던 게임의 명칭이 그대로 사용되어져 현재의 퍼즐게임이라는 장르로 불리어진다. 실제 퍼즐게임의 조건으로는 컴퓨터가 제시하는 문제를 플레이어가 풀고, 문제를 풀면 다음 문제가 제시되는 형태의 게임을 의미하지만 실제 최근의 퍼즐게임의 의미로는 한 화면 내에서 특정 조건을 만족시키기 위해 플레이어가 혹은 플레이어끼리 플레이하는 형태의 게임을 의미하는 경우가 많다. 대표적인 게임으로는 여러 형태의 블록을 결합시켜 최종적으로 그 블록들을 소거시키는 테트리스라는 게임을 예로 들 수 있다. 본 논문에서 제작한 게임은 한글이 자음과 모음으로 구성되어 있다는 것에 착안하여 자음과 모음을 따로 구성하고 그것들을 조합하여 단어를 만드는 게임으로 퍼즐장르의 게임으로 구분 지었다.

3. 자음과 모음 결합형 퍼즐게임

3.1 게임 시나리오

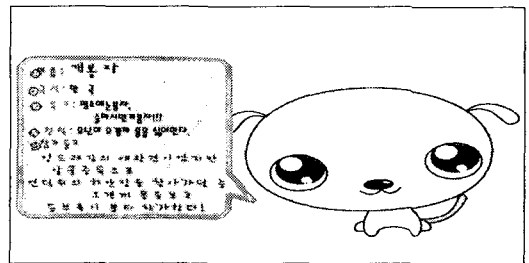


(그림 1) 게임 제작 과정

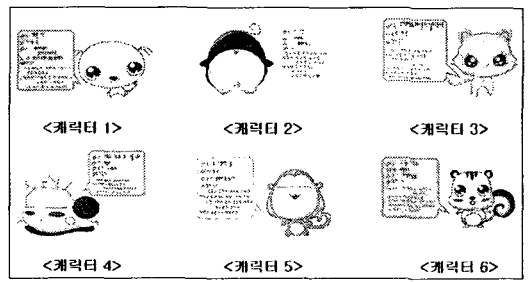
한글 조합 퍼즐게임의 제작 과정은 그림 1 과 같다. 먼저 게임 기획 후 게임의 전반적인 게임 인터페이스와 디자인에 대한 아웃라인을 작성한다. 게임 프로그래밍과 그래픽, 오디오는 따로 작업되어지며 최종적으로 작업되어진 그래픽과 오디오 소스들을 프로그래밍 되어있는 게임에 통합하여 최종 결과물을 산출하게 된다.

3.2 그래픽, 오디오 작업 및 프로그래밍

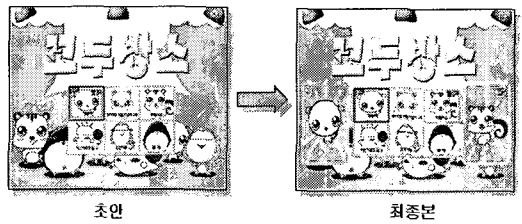
본 게임에서의 그래픽 분야 작업은 캐릭터, 인터페이스, 메뉴화면, 오프닝, 로고화면의 5가지 단계를 가진다. 사용 툴로는 포토샵, 일러스트, 프리미어, 플래시 의 4가지 툴이 사용되었다. 캐릭터는 A4 용지에 초안을 그린 후 회의를 거쳐 여러 캐릭터 중 6종류를 선택하였으며 선택되어진 6종류의 초안은 스캐너로 이미지 스캔 후 일러스트와 포토샵 툴을 이용하여 다시 그리게 된다[1][2][8][9]. 오디오 작업은 Sonar2.0 프로그램을 통한 미디작곡의 결과물을 게임내에 삽입하는 방식으로 게임 음악을 구현하였으며 배경 효과음은 샘플링 된 효과음을 사용하였다[6].



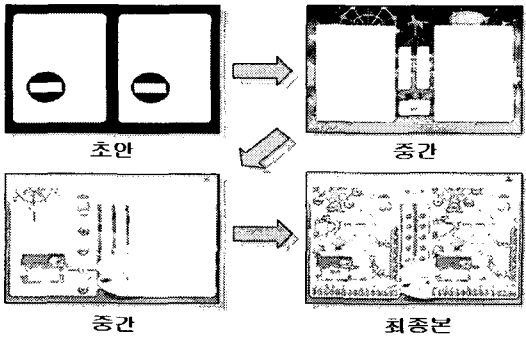
(그림 2) 캐릭터 작업



(그림 3) 캐릭터 종류



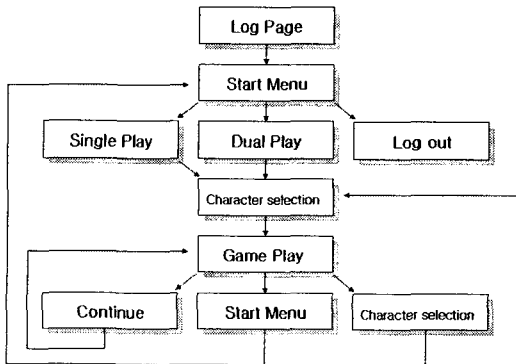
(그림 4) 캐릭터 선택화면 발전과정



(그림 5) 게임 인터페이스 발전과정

(그림 2)와 (그림 3)은 이미지 스캔 된 6가지의 캐릭터를 일러스트와 포토샵을 이용해 다시 작업한 캐릭터 화면을 보여주고 있으며 그림4와 그림5는 캐릭터 선택화면과 게임 인터페이스 화면의 발전과정을 보여주고 있다.

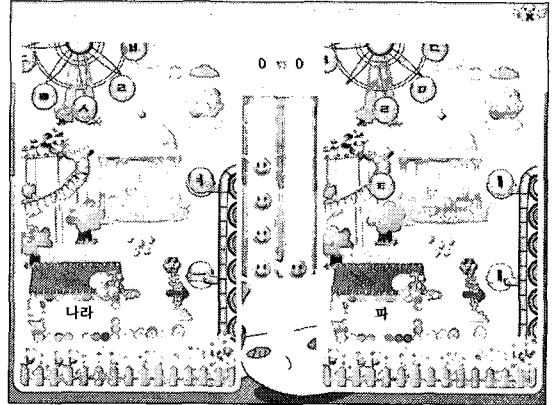
한글 퍼즐 게임은 19개의 자음과 14개의 모음들 중 원하는 자음과 모음을 선택하여 단어를 조합하는 게임이며 window 환경에서 Visual C++, MFC, Direct X 등의 개발 프로그램을 이용하여 구현된다. 전체적인 게임의 흐름은 (그림 6)과 같다.



(그림 6) 전체적인 게임 로직

(그림 7)은 한글 퍼즐 게임의 구동화면이다. 자음부에 해당하는 볼(이하 자음볼)은 상단 좌측부에서 원운동을 하며 이동하고 모음부에 해당하는 볼(이하 모음볼)은 게임화면 우측 벽에서 위에서 아래로 이동한다. 자신이 원하는 자음볼이나 모음볼이 판정위치에 도달 했을 때 특정 버튼을 눌러 해당 볼들을 합성소로 이동시키고

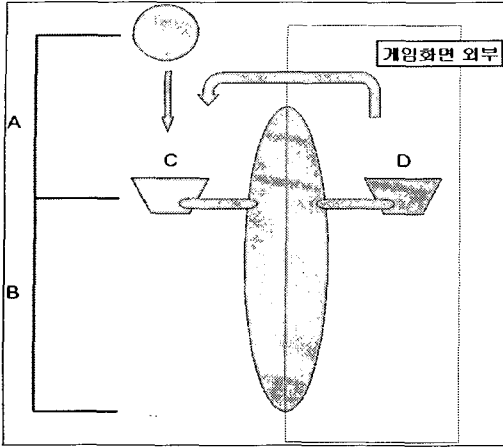
합성소에서 조합시킨 단어가 플레이어가 원하는 단어일 경우 확인 버튼을 눌러 승리하는 방식의 게임이다.



(그림 7) 한글 퍼즐 게임 구동화면

3.2.1 게임플레이

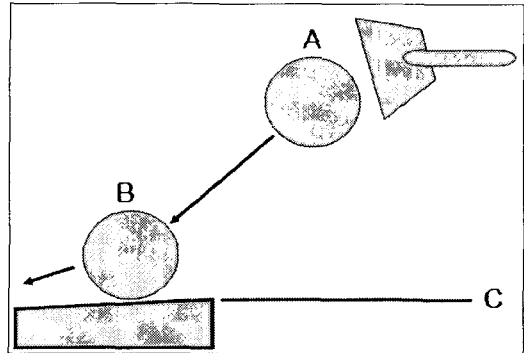
자음볼이 위치하는 판람차는 여러장의 그림을 애니메이션 처리하여 움직이는듯하게 처리하였으며 그 판람차에 달려있는 자음볼들은 등속 원운동 공식을 이용하여 움직이도록 따로 구현하였다. 이유는 플레이어가 선택한 자음볼의 경우 합성소로 떨어지는 움직임이 표현되어야 하기 때문에 판람차의 이미지와 따로 구현시킬 필요가 있다[3][4]. 자음볼과 모음볼의 경우는 각각 랜덤하게 출현하게 된다. 전체적으로 랜덤화 시켰을 경우 같은 자음과 모음이 중복되어 출현하는 현상이 발생할 수 있기 때문에 1주기(자음 19개, 모음 14개)의 범위 내에서 랜덤하게 출현하고 다시 다음 주기에서 랜덤하게 출현하게 구현하였다. 최악의 경우 주기와 주기의 사이에서 한번정도 중복되어 나오는 경우가 발생하였지만 실제 게임 플레이시에 그런 경우는 게임 플레이에 크게 지장을 주지 않았다. 각각의 볼들은 판정 구역에 들어오면 자음볼의 경우 구슬 색깔이 변하고 모음볼의 경우 표지판의 색깔이 변해서 현재 합성소로 이동시킬 수 있는 판정의 볼이 어떤 것인지 확인 가능하게 한다. 판정이 있는 볼을 선택하였을 경우 자음볼은 합성소로 등속 운동으로 떨어지고 모음볼은 레일의 경로를 통해 합성소로 이동한다.



(그림 8) 레일위로 낙하하는 볼 구현

(그림 8)은 모음볼 생성기에서 생성된 모음볼이 레일에 달려있는 바구니에 결합되는 원리를 표현한 그림이며 다음과 같이 요약된다.

1. 공과 레일에 달려있는 바구니가 만나는 부분을 기준으로 A 구간은 15, B 구간은 5로 속도를 나눈다.(속도의 기준은 1 프레임당 이동하는 픽셀수. 즉, 15의 속도는 1프레임당 15픽셀을 이동하는 속도를 의미)
2. 공이 떨어지는 주기와 바구니 곡선 회전 주기는 같다.
3. 바구니는 화면상에 보이지 않는 D의 위치에서 생성되어 화면 안쪽으로 화살표 방향으로 이동한다.
4. 볼이 A와 B의 경계점에 왔을 때 바구니는 C의 위치에 있어야 한다.
5. 볼의 속도가 A 구간에서 15의 속도이고 A 구간의 거리가 약 300픽셀, 초당프레임 속도를 30프레임 이라고 가정하면 $300/(30*15) \approx 0.66$ 초 이므로 약 0.66초 후에는 공은 C의 위치에 도달하게 된다. 즉, 바구니는 D에서 C의 위치까지 0.66초 안에 이동해야한다.



(그림 9) 선택한 볼의 지면상 이동

(그림 9)는 판정위치에서 선택된 모음볼이 바구니에서 지면으로 떨어져서 합성소까지의 이동 원리를 표현한 그림이며 다음과 같이 요약된다.

1. C를 기준으로 볼이 낙하 할 때와 지면에 닿은 후 지면의 각도대로 이동하는 것을 표현.
2. 바구니를 떠난 볼(A)은 C의 위치에 있는 지면까지 대각선 방향으로 빠른 속도로 낙하한다.
3. 지면에 도달한 볼(B)은 지면까지의 낙하 속도와는 다른 속도로 합성소까지 이동한다.

```

void GameScreen::mix_tp()
{
    editmix_tp->enableChar(true); //edit항 활성화
    editmix_tp->empty(); //전 판정 삭제
    editmix_tp->setFocus(); // edit항 치움

    POSITION pos;
    pos = listJoin.GetHeadPosition(); // 리스트 헤더 포인터

    keybd_event(UK_HANGEUL, MapVirtualKey(UK_HANGEUL, 0), 0, 0); //한글사용 요청
    keybd_event(UK_HANGEUL, MapVirtualKey(UK_HANGEUL, 0), KEYEVENTF_KEYUP, 0);

    while( pos != 0 ) //리스트 끝까지
    {
        Join temp = listJoin.GetNext( pos ); //다음 리스트로동

        if(temp.shift==true) keybd_event(16, 0, 0, 0); //shift키 눌림

        keybd_event(temp.symbol, 0, 0, 0); // 리스트에 저장된 단어를 edit창에 입력
        keybd_event(temp.symbol, 0, KEYEVENTF_KEYUP, 0);
        keybd_event(16, 0, 2, 0);
    }

    keybd_event(UK_HANGEUL, MapVirtualKey(UK_HANGEUL, 0), 0, 0); //한글사용 요청 취소
    keybd_event(UK_HANGEUL, MapVirtualKey(UK_HANGEUL, 0), KEYEVENTF_KEYUP, 0);

    editmix_tp->enableChar(false); //edit항 활성화
}
    
```

(그림 10) 한글 조합기 소스 일부

3.2.2 한글조합

한글은 초성 중성 종성으로 나뉘어져 있기 때문에 본 게임을 구현하기 위해서 우선 조합기의

구현이 필요하다. 단어 조합기는 API에서 지원 하는 편집창을 이용하여 구현하였으며 편집창을 이용하여 구현 할 경우 플레이어가 직접 그 창에 커서를 위치시키고 단어를 입력할 수 있는 문제점이 있을 수 있으므로 게임 화면 밖에 편집창을 위치시키고 그 창에서 문자열만 가지고 와서 단어 합성소에 디스플레이 하는 방식을 택하였다[5][8].

3.2.3 한글 검색

본 게임의 목표가 자음과 모음을 조합하여 명사에 해당하는 한글 단어를 만드는 것이므로 플레이어가 조합한 단어가 실제로 존재하는 단어인지 검색하는 기능을 가지는 검색기의 구현이 필요하다. 효율적인 검색을 위하여 조합이 완료된 단어를 명사 단어 데이터베이스에서 검색하는 방식이 아니라 최초에 합성소로 들어온 자음 불이 무엇이냐에 따라 그에 해당하는 단어들만 검색기준에 포함시키는 방식으로 검색기는 구현되었다[4]. 예를 들어 플레이어가 "기차"라는 단어를 조합했을 경우 최종 조합된 "기차"라는 단어를 명사 데이터베이스에서 검색할 경우 방대한 양의 명사 데이터베이스 내에 "기차"라는 단어가 있는지 검색하기 위해 게임상의 약간의 딜레이가 생기므로 "기차"의 초성인 "ㄱ"이 입력되는 순간 "ㄱ"으로 시작되는 단어를 제외한 나머지 단어들은 검색 대상에서 제외되는 방식으로 검색기를 구현하였다.

4. 결론 및 향후 과제

본 논문에서는 자음과 모음을 조합하여 단어를 만드는 한글 조합 퍼즐게임을 구현하였다. 하지만 실제 플레이를 해 본 결과 쌍자음에 해당하는 단어는 실제로 거의 선택되지 않는 경우가 많았으며 모음에 있어서도 거의 선택되지 않는 모음이 존재하였다. 실제 잘 선택되지 않는 특정 자음이나 모음도 자주 선택되는 자음과 모음과 같이 랜덤하게 출현하게 프로그래밍 되어져 있으므로 최악의 경우에는 기다리는 자음이나 모음이 너무 늦게 출현하여 게임 진행이 지루해지는 경향이 있었다. 향후 연구 과제로는 실제로 잘 사용되지 않는 자음과 모음을 제외시켜 보다

더 긴박한 게임 플레이가 가능하도록 게임을 수정하는 것이며 네트워크 기능을 적용시켜 유저들이 인터넷을 통해 멀티 플레이가 가능하게 하는 것과 모바일 게임 서비스 또한 가능하게 하는 것을 목적으로 한다.

참고 문헌

- [1] 한형만, "플래시 MX 2004 디자인 스쿨", 한빛미디어, 2006.
- [2] 김상중, "Premiere 6.5", 영진닷컴, 2002.
- [3] 신동훈, "온라인 게임 네트워크 프로그래밍", 대림출판, pp.156~265, 2004.
- [4] 김용성, "Visual C++ 6 완벽가이드 2nd Edition", 영진닷컴, pp.211~287, 2006.
- [5] 김상형, "API 정복", 다남사, pp.484~528, 2004.
- [6] 박운영, "Sonar 2.x Cakewalk11", 해지원, 2004
- [7] Ralph Davis, 김승태 역, "Win32 네트워크 프로그래밍", 도서출판 대원, 1997
- [8] Adobe systems, 박상화 역, "Adobe Illustrator CS2", 비비컴, 2002
- [9] 강화영, "Color Illust and Using Photoshop", 태학원, 2002



조재영

2002년 : 목원대학교 컴퓨터 공학부(공학사)

2004년 : 목원대학교 컴퓨터 공학부(공학석사)

2007년 ~ 현재 : 목원대학교 컴퓨터 공학부 박사과정

관심분야 : 디지털오디오 시스템, Engineering Art, 영상편집, 게임제작



김윤호

1987년 : 경희대학교 전자공학과(공학석사)

1992년 : 청주대학교 전자공학과(공학박사)

2005-2006 : Univ. of Auckland NZ Research Fellow

2007년 ~ 현재 : 목원대학교 컴퓨터 공학부 교수/기술사

관심분야 : 영상처리, 컴퓨터비전, 뉴로퍼지응용, 정보통신 정책