

# 패밀리 팩을 이용한 게임에서 애니메이션의 구현\*

류 동 항, 정 민 수, 이 중 동  
경남대학교 컴퓨터공학과

## Implementation of Animation for Family Pack Games

Dong-Hang Ryu, Min-Soo Jung, Jong-Dong Lee  
Dept. of Computer Engineering, Kyungnam University

### 요 약

현재 컴퓨터를 이용한 교육용 게임과 관련 프로그램들이 다양하게 개발되어 있다. 그리고 프로그램의 교육적 효과와 흥미 유발을 위한 여러 가지 애니메이션 기법들이 발표되었다. 교육용 프로그램에서 애니메이션을 통한 사용자와의 상호작용은 교육 효과를 높이는데 매우 이상적인 방법이다. 본 논문에서는 기존의 컴퓨터가 아닌 가정용 패밀리 컴퓨터를 위한 몇 가지 애니메이션 기법을 구현하여 패밀리 게임을 개발하였다.

### 1. 서 론

컴퓨팅 기술의 발달과 함께 애니메이션 관련 기술들도 빠른 속도로 발전하고 있다. 산업 분야 뿐만 아니라 교육용 소프트웨어 분야에서 애니메이션은 매우 중요한 부분을 차지한다. 특정 단원에 대한 이해를 돕고, 지루하지 않고 몰두 할 수 있게 하며, 올바른 행위에 대한 보상 효과를 애니메이션화 한다면 교육적 효과는 더욱 커지게 된다[1,2]

본 논문에서는 패밀리 컴퓨터를 이용한 교육용 게임의 애니메이션 기법을 연구하였다. 본 논문의 구성은 다음과 같다. 2 장에서는 패밀리 컴퓨터의 하드웨어 특성에 대해 기술하고, 3 장에서는 애니메이션 기법을 소개를 하며, 4 장에서는 구현된 내용을 기술하며 마지막으로 5 장에서는 결론을 기술하였다.

### 2. 패밀리 컴퓨터의 하드웨어 특성

패밀리 컴퓨터는 그림 1 과 같이 CPU의 명령어를 펌치(fetch)하기 위한 하나의 디코드(Decode)와 타이밍 제어를 위해 사용되는 타이밍 제너레이터(Generator), 사운드와 비디오 출력부, 그리고 조종기의 입력을 위한

I/O 인터페이스로 제공되며, 6502 CPU와 비디오 출력을 위한 PPU(Picture Process Unit), 사운드 출력을 위한 PSG(Programmable Sound Generator)등으로 구성된다.

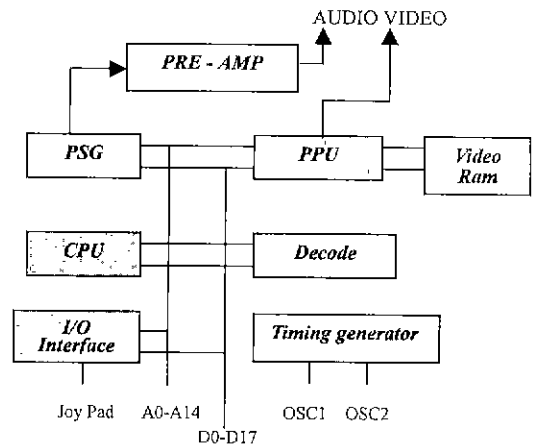


그림 1 패밀리 게임기의 하드웨어 구조

\* 본 연구는 중소기업청 산 학.연 지역 컨소시엄 과제 연구비의 지원으로 수행되었습니다.

이 시스템에서 가장 중요시 해야 할 점이 타이밍 문제인데 처리해야 할 일이 많다고 해서 갑자기 전체 속도가 느려지거나, 반대로 처리해야 할 일이 없을 때 빨라지면 안되기 때문이다. 이 타이밍 문제를 적절히 조절하기 위해서 인터럽트 개념을 도입해서 사용하고 있다. 이 인터럽트중 타이밍에 필요한 것이 주로 CRT 칩에서 나오는 주사선 인터럽트이다. 주사선 인터럽트는 화면을 주사할 때가 되었을 때 CRT 칩에서 CPU 쪽으로 보내는 요구로서 보통 60분의 1초에 1번씩 들어오게 된다 [6]

이 주사선을 사용하여 타이밍을 조절하는 방법은 크게 2가지로 나뉘는데, 첫째는 메인 프로그램은 인터럽트 외부에서 동작하고, 타이밍 계산만 인터럽트를 이용하는 방법이고, 둘째는 메인 프로그램 자체가 인터럽트에서 동작하는 방법이 있다. 여기서는 두 번째 방법을 사용하는데, 일정 시간을 주기로 들어오는 주사선 인터럽트를 사용하여 프로그램이 수행된다[6]

여기서 중요한 것은 위에 설명한 60분의 1초 내에 원하는 해당 모듈을 실행하고 새롭게 구성된 화면을 TV로 전송해야만 한다. 그렇지 못할 경우에는 화면의 떨림 현상이나 화면이 깨어지는 현상이 나타나게 된다.

패밀리 컴퓨터에서는 화면을 TV로 전송할 순간에 화면 출력을 중지시킨다. 그리고 화면이 모두 전송되면 다시 화면 출력을 재개시키게 된다.

### 3. 애니메이션 기법

패밀리 컴퓨터에서 화상의 애니메이션은 스프라이트(Sprite)와 뱅킹(Banking)을 이용한 두 가지로 구분할 수 있다. 스프라이트에 의한 애니메이션 기법은 주로 작은 크기의 화상을 고속으로 움직이고자 할 때 사용하며, 뱅킹에 의한 애니메이션 기법은 배경 화면의 그림 데이터가 저장되어 있는 메모리의 일부분을 일정한 시간 간격으로 교체함으로써 배경 화면의 움직임을 조절 하는 방법이다.

#### 3.1 스프라이트

캐릭터(Character)란 개성을 뜻하며 게임에서 사용될 때는 개성을 가진 등장인물을 가리킨다 일반적으로 캐릭터를 스프라이트 또는 오브젝트라고도 하며 배경과는 관계 없이 스스로 움직이는 것들을 지칭한다. 패밀리 컴퓨터에서 기본 스프라이트 사이즈는 8 x 8 이며, 3 가지 색상의 컬러 팔레트가 4 개까지 지원된다.

하드웨어적으로 스프라이트 기능이 없는 일반 컴퓨터에서는 소프트웨어적으로 스프라이트 기능을 구현한다. 그 과정을 살펴 보면 먼저 스프라이트가 표시될 배경공간을 보존시킨 후 스프라이트를 출력하고 이동할 때는 보존해둔 원래의 배경을 다시 출력한 후에 이동할 지점의 데이터를 저장하는 것을 반복한다. 하지만 하드웨어적인 스프라이트 기능에 비해 속도가 현저하게 떨어지게 된다 반면에 패밀리 컴퓨터에서는 하드웨어적인 스프라이트 기능이 지원된다.

패밀리 컴퓨터에서 스프라이트를 이용하여 화상에 대해 애니메이션을 처리하는 방법은 다음과 같다. 첫째, 스프라이트를 정의한다. 스프라이트를 정의 한다는

것은 스프라이트 패턴 데이터를 스프라이트 체너레이터 테이블에 등록시키는 것을 의미한다. 둘째, 스프라이트를 출력한다. 스프라이트를 출력한다는 것은 정의된 스프라이트를 스프라이트 속성 테이블의 속성 비트를 조작하여 실제로 화면에 출력한다는 의미이다.

스프라이트 체너레이터 테이블은 일반적으로 비디오(Vram) 램상에 저장되지만 패밀리 기종은 롬(Rom)에 저장된다. 그림 2는 스프라이트 체너레이터 테이블의 구조이다

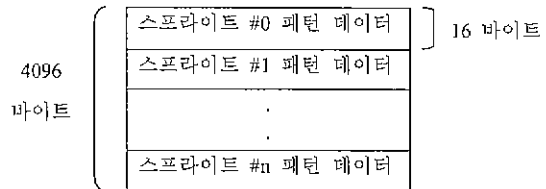


그림 2 스프라이트 체너레이터 테이블의 구조

스프라이트 속성 테이블은 스프라이트가 표시되기 위한 X/Y 좌표, 속성, 팔레트, 컬러와 같은 여러가지 정보가 저장된다. 속성에는 상/하 또는 좌/우 반전과 스프라이트의 출력 우선 순위등을 지정할 수 있다

스프라이트의 애니메이션은 이러한 속성 비트 중에서 출력될 좌표를 적절히 조정하여 움직임을 처리하는 경우와 여러 장의 스프라이트 데이터를 구성하여 각 스프라이트를 교체하는 방법으로 나눌 수 있다.

#### 3.2 뱅킹

맵 버퍼는 프로그램에서 사용하는 배경 화면의 정보가 저장되는 부분이다. 이 영역은 각각의 맵을 구성하는 데이터의 인덱스가 나열되며 필요할 때 마다 배경 화면을 교체할 수 있다. 배경 화면은 스프라이트와 달리 동적인 움직임 처리에는 부적합하지만 영역의 일부분을 교체하는 방법을 통해 부분적으로는 각 형상에 대한 애니메이션이 가능하다.

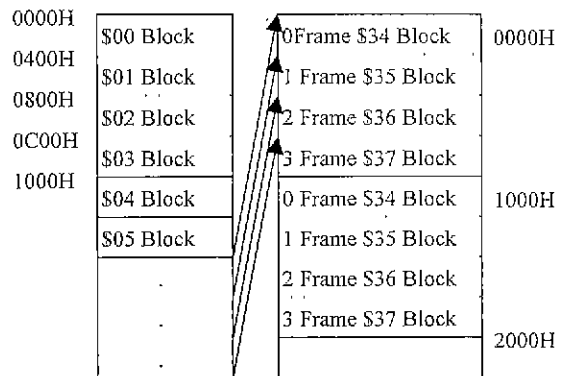


그림 3 폰트 롬과 메모리 뱅크의 관계

그림 3은 폰트 롬과 메모리 맵과의 관계를 나타낸다. 게임 팩은 PCB에 따라 블록의 분할 크기가 달라지지만 보통 400H 또는 800H 단위로 분할된다. 일반적으로 4개의 프레임으로 나뉘어 펌핑되는 경우에 마지막 프레임은 맵의 일부 데이터를 교체하기 위해서 사용된다.

#### 4. 구 현

구현된 프로그램은 윈도우 95 상에서 6502 CPU를 위한 x6502 크로스(Cross) 어셈블러를 이용하여 작성하였다.

그림 4는 스프라이트를 이용하여 화상을 출력하는 코드이다. 스프라이트 출력 모듈은 스프라이트의 출력 속성과 좌표, 그리고 스프라이트 페턴 데이터를 전송하고 실제로 화면 출력하는 모듈을 호출한다[3, 4, 5].

```

SP1:
    LDA    #0
    STA    SPRITE_ATTR
    LOADM  SPRITE_XX, BORAMI_XX
    LOADM  SPRITE_YY, BORAMI_YY

    LDA    SIDE01_CMT
    ASL    A
    TAX
    LOADM  PUT_SPRITE_DATA, BORAMI_TBL, X
    LDY    SPRITE_CMT
    JSR    PUT_SPRITE
    STX    SPRITE_CMT
    RTS
    
```

그림 4 스프라이트 출력 모듈

이 모듈에서 스프라이트의 좌표를 계속적으로 변경시키면 화상의 애니메이션을 조정할 수 있다. 실제 전송되는 스프라이트 페턴 데이터는 그림 5와 같다.

```

BORAMI_TBL:
    DW    BORAMI_0
    DW    BORAMI_1

BORAMI_0:
    DB    0, 0
    .DB   2, 3
    DB    $07, $08
    DB    $17, $18
    .DB   $27, $28

BORAMI_1
    .DB   -8, 0
    .DB   3, 3
    .DB   $0A, $0B, $0C
    DB    $1A, $1B, $1C
    DB    $2A, $2B, $2C
    
```

그림 5 스프라이트 페턴 데이터

그림 5는 두개의 스프라이트 페턴 데이터를 나타내고 있으며 이러한 두개의 페턴 데이터에 대한 인덱스를 이용하여 출력시에 화상을 일정한 시간 간격으로 출력함으로써 두 화면이 교차하는 애니메이션 효과를 얻을

수 있다. 그리고 좀더 부드러운 애니메이션 동작을 원한다면 페턴 데이터를 추가하여 화상의 개수를 증가시키면 된다.

그림 6은 बैं킹에 의한 애니메이션을 구현한 예제이다. 위의 예에서와 마찬가지로 일정한 시간 간격을 두어 출력되는 배경이 되는 폰트 बैं크의 번호를 교체하는 방식을 사용한다. 이 모듈은 실제 주 실행 모듈에서 다른 모듈과는 상관 없이 60분의 1초마다 페턴 불러게 된다. 그리고 적절한 시기가 되면 बैं킹이 이루어진다.

```

BANCKING:
    LDA    INT_CNT
    AND    #$1F
    BNE    BC_EXIT

    INC    FLAG61

    LDA    FLAG61
    AND    #3
    STA    FLAG61

    CLC
    ADC    #$E8
    JSR    SET_FONT_BANK_B0

BC_EXIT:
    RTS
    
```

그림 6 बैं킹 모듈

#### 5. 결 론

본 논문에서는 페밀리 컴퓨터를 위한 애니메이션 기법을 구현하였다. 하지만 페밀리 상의 애니메이션은 스프라이트 출력 크기와 가로 좌표상의 출력 개수에 대한 제약 문제 때문에 원활한 애니메이션 효과를 얻기가 힘들다. 따라서, 페밀리 컴퓨터에서 보다 향상된 애니메이션 기법에 대한 연구가 필요하다.

#### 참 고 문 헌

- [1] Wortham, Sue Clark, Early Childhood Curriculum, Simon & Schuster, p 146-351(1993)
- [2] Merrill, P, Computers in Education, Allyn & Bacon, p74-134(1994)
- [3] Lemone, K, Assembly Language Computer Organization, Harper-Collins, p.128-249(1988)
- [4] Yu, Y, Assembly Language Programming and Organization of the I, McGraw-Hill, p 88-105(1989)
- [5] Fernandez, JN, Assembly Language Programming, McGraw-Hill, p165-245(1990)
- [6] 김도우, 정민수, 이현순. "페밀리 게임기상의 유아 교육 프로그램 구현", 정보과학회지, 25(1), pp 739-741(1998)