

# 초등학교 교과서 음악과 연계한 구구단 학습법

이현정  
서울교육대학교 컴퓨터교육과  
hardrain99@hotmail.com

## Multiplication table Study related to Elementary Music

Hyunl-jung Lee  
Dept. of Computer Education, Seoul National University of Education

### 요약

수학은 생활의 기본이 되는 도구 교과이다. 그 중에서 아동들이 가장 중요하지만 어려워하고 후속 학습에까지 영향을 미치는 것은 구구단이다. 고학년이 되어서까지 곱셈 구구단을 못 외우는 학생들은 지도가 어려울 뿐만 아니라 이후에도 지속적으로 수학 학습에서 뒤떨어지게 된다. 하지만 암기가 중심이 되는 부분이라 다른 것들보다 학습의 흥미가 떨어지고 시기를 놓치면 학습이 점점 더 어려워지는 단점이 있다. 이에 지금까지 어려웠던 어렵고 암기위주로 흘러가던 구구단을 음악과 연결시켜 보는 것은 어떨까란 생각을 하게 되었다. 여기에서 쓰이는 음악들은 모두 초등학교 교과서에 있는 음악으로 아동들이 친숙하게 느낄 수 있는 음악들이다. 이 소프트웨어는 2학년 뿐만 아니라 구구단을 어렵게 느끼는 고학년의 학습에도 도움이 되도록 제작하였다. 프로그램은 전체 구구단인 2~9단을 다루도록 하였으며, 각 단은 4개의 파트로 구성되어 있다. 1단계에서는 1-각 수까지 반복해서 세는 연습을 하였고, 2단계에서는 반복했던 수를 연결하여 띄어세기 연습, 3단계에서는 2단계에서 했던 띄어세기를 확장하여 구구단 연습을 해 볼 수 있도록 하였다. 정리단계에서는 각단의 구구단을 랜덤으로 배치하여 구구단의 복습이 될 수 있도록 하였다. 2,4,5,6,8단은 각각 박자에 맞춰서 구구단을 외우는 것이고, 5,7 단은 박자에 맞추기 어려우므로 글자수에 맞춰서 구구단을 외우도록 하였다. 박자에 맞추는 것이 주가 아니라 수학을 배우는 것이 중심이므로 틀린 박자에 맞는 숫자를 누르는 것은 반응하지 않도록 하였다. 구구단 중간에 있는 수까지를 전부 제시함으로써 큰수와 띄어세기 그리고 구구단과의 관계를 이해할 수 있도록 하였다.

### 1. 서론

수학은 생활의 기본이 되는 도구 교과이다. 그 중에서 아동들이 가장 중요하지만 어려워하고 후속 학습에까지 영향을 미치는 것은 구구단이다. 고학년이 되어서까지 곱셈 구구단을 못 외우는 학생들은 지도가 어려울 뿐만 아니라 이후에도 지속적으로 수학 학습에서 뒤떨어지게 된다. 하지만 암기가 중심이 되는 부분이라 다른 것들보다 학습의 흥미가 떨어지고 시기를 놓치면 학습이 점점 더 어려워지는 단점이 있다. 이에 지금까지 어려웠던 어렵고 암기위주로 흘러가던 구구단을 음악과 연결시켜 보는 것은 어떨까란 생각을 하게 되었다. 여기에서 쓰이는 음악들은 모두 초등학교 교과서에 있는 음악으로 아동들이 친숙하게 느낄 수 있는 음악들이다. 이 소프트웨어는 2학

년 뿐만 아니라 구구단을 어렵게 느끼는 고학년의 학습에도 도움이 되도록 제작하였다.

### 2. 이론적 배경

음악이 갖는 특징을 생각해 보면 음악은 우리를 둘러싸고 있는 환경의 일부분이다. 우리는 주위에서 항상 음악을 듣는다. 즉, 음악은 우리 삶에서 떨어질 수 없는 부분이다. [1]

또한 구구단은 초등학교 전 과정과 중, 고등학교 일상생활에서까지 크게 영향을 미치는 과목이다. 사소한 어느 것 하나 곱셈과 연결되지 않은 것은 없다.

### 3. 프로그램 분석

#### 3.1. 사용된 음악 분석

2단 : 3학년 1. 꼬까신 (2/4 박자)

3단 : 3학년 9. 고기잡이 (2/4박자)

4단 : 3학년 3. 구슬비 (2/4박자)

5단 : 6학년 21. 풍년가 (12/8박자)  
 6단 : 5학년 6. 파란마음 하얀 마음  
 (3/4박자)

7단 : 2학년 3. 꼭꼭 숨어라 (12/8박자)

8단 : 4학년 7. 방울꽃 (4/4박자)

9단 : 4학년 11. 산바람 강바람 (3/4박자)

2학년이 중심이 되는 수업이므로 노래는 3,4 학년 음악 위주로 골랐으며 불가피하게 찾기가 어려운 경우에만 고학년 음악을 선택하였다. 생각보다 박자를 맞추는 것이 빨라 3박자 음악이라고 해도 3박자에 맞추는 것이 아니라 강박에 맞춰 음악을 천천히 흘러가도록 하였다. 2,3,4박의 박자는 무리가 없었지만 5,7박의 음악은 박자에 맞추기가 어려워 가사에 맞추도록 하였다. 속도가 너무 빠르지 않도록 몇 개의 박자는 생략하고 중첩하여 학습을 진행하였다. 음악의 박자 학습이 중심이 아니라 수학적 학습이 중심이므로 너무 박자에 연연하지는 않도록 하였다.

### 3.2. 교수 및 학습 전략 및 내용구성

- 1) 전체 구구단인 2~9단을 다루도록 하였다.
- 2) 각 단은 4개의 파트로 구성되어 있다.

1단계에서는 1-각 수까지 반복해서 세는 연습을 하였고, 2단계에서는 반복했던 수를 연결하여 띄어세기 연습, 3단계에서는 2단계에서 했던 띄어세기를 확장하여 구구단 연습을 해 볼 수 있도록 하였다. 정리단계에서는 각단의 구구단을 랜덤으로 배치하여 구구단의 복습이 될 수 있도록 하였다.

3) 2,4,5,6,8단은 각기 박자에 맞춰서 구구단을 외우는 것이고, 5,7 단은 박자에 맞추기 어려우므로 글자수에 맞춰서 구구단을 외우도록 하였다.

4) 박자에 맞추는 것이 주가 아니라 수학을 배우는 것이 중심이므로 틀린 박자에 맞는 숫자를 누르는 것은 반응하지 않도록 하였다.

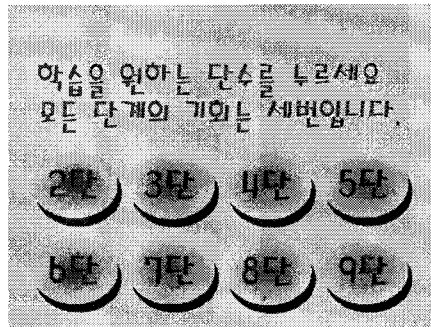
5) 구구단 중간에 있는 수까지를 전부 제시함으로써 큰수와 띄어세기 그리고 구구단과의 관계를 이해할 수 있도록 하였다.

6) 음악은 저작권의 문제상 초등학교 수준의 교과서 음악을 활용하였다.

### 3.3. 장점 및 특성

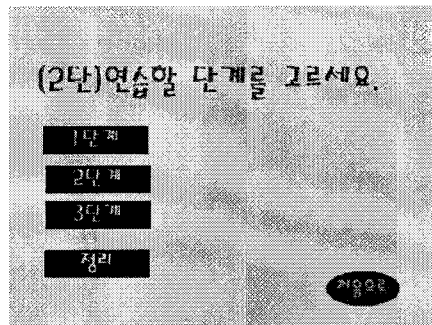
음악과 관련하여 지루하지 않게 구구단을 학습할 수 있게 하였다는 것이 가장 큰 장점이며, 구구단 뿐만 아니라, 띄어세기의 기본적인 연습을 할 수 있도록 하여 단계적인 학습이 가능하도록 하였다. 그리고 이미 다루었던 큰 수의 개념과도 연결 시켜 수에 대한 양감을 기를 수 있도록 하였다.

### 3.4 실제 프로그램 분석



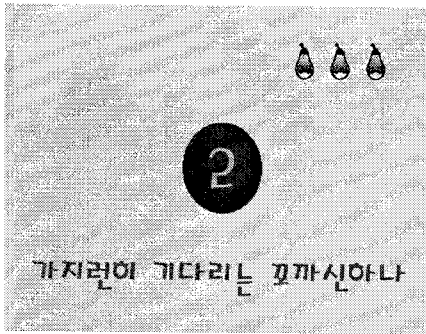
[그림1] 초기 선택 화면

초기화면에서 시작을 누르면 들어갈 수 있으며 2단부터 9단까지 원하는 학습을 선택하여 학습할 수 있습니다.

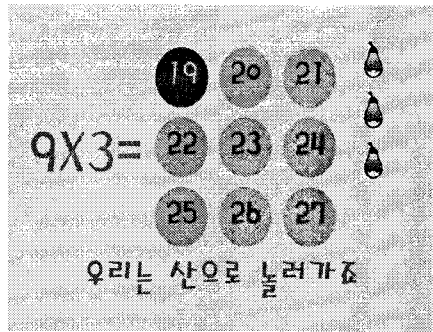


[그림2] 각 단수의 하위 선택 화면

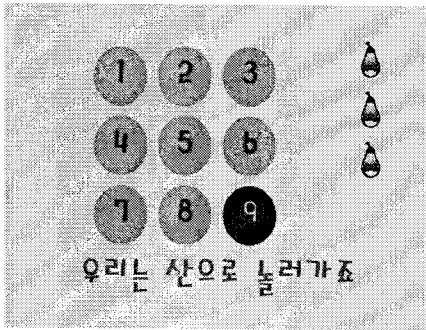
각 단수를 선택했을 때의 화면입니다. 각 단수는 3개의 단계를 통하여 음악으로 학습하고 마지막으로 정리를 할 수 있도록 합니다.



[그림3] 2~4단의 1단계



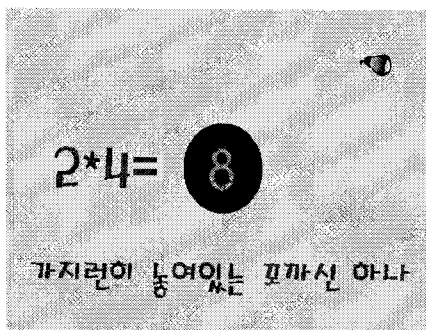
[그림6] 3~9단의 2단계



[그림4] 5~9단의 1단계

1) 1단계의 그림입니다. 2~4단까지는 하나씩 숫자가 바뀌고 각 단의 숫자가 등장할 때 누르면 맞는 개수가 더해집니다. 5~9단은 각 단수까지를 한 화면으로 제시해서 맞는 숫자에 색깔이 들어오면 누르면 됩니다. 연습이므로 1~각 단수까지의 숫자만 등장하고 구구단도 제시되지 않습니다. 우측 상단의 가지 그림은 목숨을 나타내는 것으로 3번까지 틀릴 수 있습니다. 틀리면 가지는 하나씩 없어집니다.

2) 띄어세기 앞 과정으로 박자에 맞춰 정해진 수까지 세어보는 연습을 하는 것입니다. 아직 학습의 이전 단계라고 할 수 있습니다.

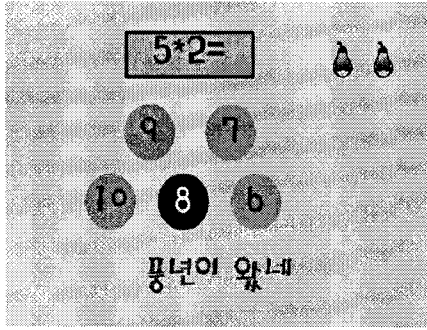


[그림5] 2~4단의 1단계

1) 동그라미 안에 1~숫자가 차례로 변하고 앞에는 구구단이 제시 됩니다. 구구단에 맞는 수에 누르면 됩니다. 처음부터 외우는 것이 아니라 무의식중에 연습하면서 이해할 수 있도록 하였습니다. 앞의 1단계에서 띄어 세는 것을 기초로 띄어세기의 숫자 변화 추이를 이해할 수 있도록 합니다. 2~3단은 각각 하나씩 제시하였고 맞는 숫자의 색깔이 변하므로 쉽게 알 수 있습니다. 3~9단은 모든 숫자의 색깔이 변하지만 위치를 이해하면 쉽게 학습이 될 수 있습니다. 1~각수를 제시하므로 큰수의 이해 학습을 도울 수 있습니다. 각 숫자의 띄어세고 묶어세기의 개념을 알 수 있도록 구성하였습니다.



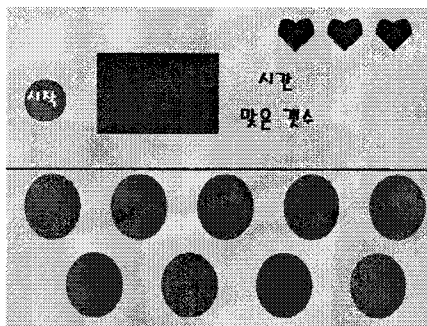
[그림7] 2~4단의 3단계



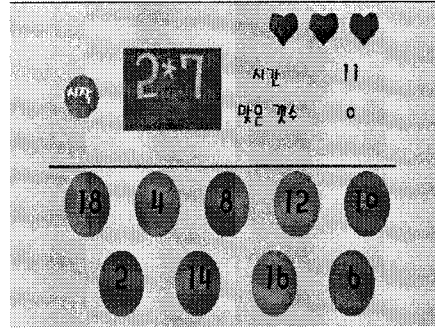
[그림8] 5\*8단의 3단계

1) 앞의 두 단계는 위치와 색깔로 구분하는 경향이 있어서 개념적 이해가 중심이 되었다면 실질적 학습이 제대로 되었는지 확인하기 이전에 보충 단계가 필요합니다. 3단계는 이를 위해 전체 수를 제시하였습니다. 위의 구구단은 박자에 맞춰 변하고 제시된 수가 변화하여 있는 도중에 맞는 답을 누르면 됩니다. 정답이 맞으면 제시된 버튼의 색깔이 변화합니다. (2~4단)

2) 5\*9단은 1부터 숫자가 차례로 변화합니다. 그러나 위치는 정해져 있지 않습니다. 맞는 수를 찾아서 누르면 됩니다. 혼동을 피하기 위해서 수는 무작위 배열이 아니라 바로 옆에 수와 연결되어 언제쯤 색깔이 들어올지 예상할 수 있도록 일정하게 순서가 정해져 있습니다.

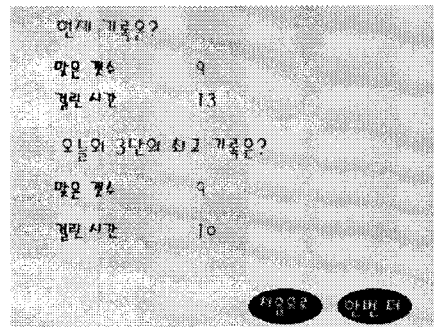


[그림9] 정리 초기 화면



[그림10] 시작을 눌렀을때의 화면

정리 화면입니다. 처음에는 빈 공간으로 있다가 시작 버튼을 누르면 숫자가 들어오고 지나간 시간이 표시 됩니다. 숫자 배치는 랜덤이며 9개의 문제가 출제됩니다.



[그림11]정리를 마쳤을 때의 결과 화면

끝나면 결과 화면이 표시되고 여러 번 하면 최고 기록이 계산 됩니다.

### 3.4. 사용된 주요 Action Script

#### 1) 변수

<표1>

변수의 역할	표시
틀린 개수를 나타내는 무비 클립의 번호를 지정하기 위한 초기 변수	i=1;
눌러야 할 시간에 눌렀는지 확인	p=0;
게임마다 맞은 개수 확인 변수	j=0;

#### 2) 정답 버튼에 작성된 Script

```
on (press) {if(p==0){
    p=1;
    j++;
    _root.yes.play();
    _root.gagil.gotoAndPlay(16);
    _root.gagi2.gotoAndPlay(16);
}}
```

```

    _root.gagi3.gotoAndPlay(16);
  })

```

주어진 버튼이 한번 눌린 것인지 혹은 두 번 눌린 것인지 확인 하는 변수 p가 있습니다. p가 0일 때는 눌리지 않은 것이므로 본문을 수행하고 p를 1로 만들어 주어 두 번 수행되지 않도록 합니다. 정답의 개수가 여러번 늘었다고 반복 더해지는 것을 방지 하기 위한 변수입니다. 여기서 j는 정답을 누른 횟수를 확인 하기 위한 변수입니다.

yes라는 무비클립의 1행에는 stop() action이 들어있고, 2행부터는 정답일때 나오는 소리가 들어있습니다.

gagi 무비 클립에서는 정답일때 16행으로 이동하게 되어 있고 정답일때 가지가 춤을 추는 동작을 하게 들어 있습니다. 버튼을 누르지 않았을 경우 무비 클립은 1~15까지 이동하여 옆으로 몸을 흔드는 동작만 합니다.

```

3) 오답 버튼에 작성된 Script
on(press){
    _root.no.play();
    if(i==4){gotoAndPlay(2);}
    else
    {_root["gagi"+i].unloadMovie();
    i++;}}

```

no라는 무비클립의 1행에는 stop() action이 들어있고, 2행부터는 오답일때 나오는 소리가 들어있습니다.

오답일때는 i를 기준으로 하여 각 무비 클립을 하나씩 제거하는 동작을 합니다. i가 4일 때는 주어진 기회 3번을 모두 소진 한 것이므로 초기 화면으로 이동하게 되어 있습니다.

#### 4)정리 화면의 시작에 작성된 Script

```

on(Release){gotoAndPlay(다음행);}

```

버튼 자체에서 변수를 모두 초기화 시킬 수도 있으나 이럴 경우 다른 버튼들이 초기화 되지 않은 상태일 경우 null값 자체에서 프로그램이 가동되는 것을 막기 위하여 다음행으로 이동하는 script를 짜 줍니다.

#### 5) 정리 화면의 초기화

```

sta=["5*1","5*2","5*3","5*4","5*5","5*6","5*7","5*8","5*9"];

```

```

_root.dan=sta;
sta2=["5","10","15","20","25","30","35","40","45"];

```

```

_root.dan2=sta2;
Array.prototype.mixArray = function(){
    var len=this.length;
    for(var i=0;i<len;i++){
        var x=random(len);
        var y=random(len);
        var z=this[x];
        this[x]=this[y];
        this[y]=z;});

```

```

_root.dan.mixArray();
_root.dan2.mixArray();
sta=["5*1","5*2","5*3","5*4","5*5","5*6","5*7","5*8","5*9"];

```

```

sta2=["5","10","15","20","25","30","35","40","45"];

```

```

_root.txtResult2=0;
_root.txtResult="";
for(i=0;i<9;i++){["result"+i]="";}
j=0;
i=1;
txtResult4=0;

```

stop();

5단의 예입니다. 다른 단수는 sta 와 sta 2의 숫자만 변화시켜 주면 됩니다. 시작 버튼과 같이 있는 메인 화면에서 초기화 하는 것입니다. dan과 dan2는 이 기초적인 자료를 랜덤으로 선택하여 재배치 하여 저장해 줍니다.

#### 6) 각 정답에 있는 버튼의 Script

```

on (press) {for (k2=0; k2<9; k2++) {
    if (txtResult == sta[k2]) {
        kk2 = k2;}
    if (result0 == sta2[k2]) {
        k1 = k2; }
    }

```

```

    if (k1 == kk2) {
        _root.yes.play();
        j++;
        txtResult2 = j;}

    else {_root.no.play();
    if (i == 4) {gotoAndPlay(정리 초기
화면프레임 번호); }
    else {_root["heart"+i].gotoAndPlay(2);
    i++;}}
if(txtResult==dan[8]){gotoAndPlay(정리 결과
화면 프레임 번호);}
    for (k=0; k<8; k++) {
        if (txtResult == dan[k]) {
            answer = dan[k+1];
            gotoAndStop(정리 화면 계속 수행
프레임 번호);}
        }txtResult = answer;}

```

sta에 있는 값과 화면에 표시된 값을 비교하여 몇 번째 위치에 있는지 확인 하고, 지금 누른 버튼에 있는 답의 값과 sta2에 있는 값의 위치를 확인 합니다. 위치가 같은 경우 맞는 답이므로 정답 사운드를 발생하고 정답의 개수를 하나 증가 시킵니다. 그리고 화면에 맞는 개수를 표시합니다.

틀린 경우 몇 번 틀렸는지 i를 통하여 확인하고 정리 초기화면으로 넘어가거나 무비럽을 삭제합니다.

다음 문제로 넘어가기 위하여 dan배열을 하나 증가시키고 마지막 값인지 확인하여 마지막 값이 아닐 경우 계속 수행하고 마지막 값일 경우 정리의 결과 화면을 표시합니다.

7) 흐른 시간을 나타내는 Script

```

txtResult4=0;
    i=1;
    txtResult=dan[0];
for(ku=0;ku<9;ku++)
{_root["result"+ku]=dan2[ku];
_root.txtResult2=0;}
j=0;
var countNum = 1;

```

```

function countDown() {
    txtResult4=countNum;
    countNum++;
}
countObj = setInterval(countDown, 1000);
stop();
setinterval함수는 흐른 시간을 표시하는 것
입니다. 1초에 한번씩 불러와서 화면상에 표시
합니다.
8) 정리 화면의 결과 출력
clearInterval(countObj);
txtResult5=j;
txtResult6=countNum;
if(fivebestj<=j){
    fivebestj=j;
if(fivecountNumbest>=countNum)
{fivecountNumbest=countNum;}}
txtResult7=fivebestj;
txtResult8=fivecountNumbest;
stop();

```

흐른시간을 나타내는 함수를 제거하고 맞은 개수를 표시합니다. best 함수에는 맞은 개수가 저장되고 countNumbest에는 걸린 시간을 저장하여 맞은 개수가 이전보다 많고 걸린 시간이 적을 경우 최고기록으로 저장하여 보여 줍니다.

#### 4. 결론 및 제언

초등학교에서 가장 중요한 연산 기능의 하나인 곱셈은 음악과 관련된 학습으로 인해 향상됨을 기대할 수 있다.

본 프로그램을 통하여 우리는 수학적 측면에서 여러 가지 것들을 익힐 수 있다.

첫째, 80까지의 큰 수를 다시 한번 익힐 수 있는 기회를 가질 수 있다. 1~각 단의 마지막 수까지 가보는 연습을 통해 우리는 수의 개념과 양감을 자연스럽게 익힐 수 있다.

둘째, 큰 수의 띄어세기를 자연스럽게 익힐 수 있다. 2,3,4.....식으로 이어지는 수의 띄어세기가 단순히 딱딱한 일정한 규칙이 없는

수의 모임이 아니라는 것을 많은 연습을 통하여 익힐 수 있다.

셋째 이를 토대로 간단한 구구단을 외울 수 있다. 리는 구구단이라고 하면 어렵고 외어서 이해해야 하는 것이라고 생각을 하고 있었던 것을 이 프로그램을 통하여 깰 수 있으며 자연스럽게 음악을 통한 정서 함양 까지 도모하려고 한다.

그리고, 음악을 위해 기본적으로 요구되는 소양인 박자 감각을 익히면서 구구단을 외울 수 있다.

앞으로 많은 부분에서 음악을 활용한 학습의 보편화가 이루어졌으면 하는 바람이다.

## 5. 참고 문헌

[1]교육부, “초등학교 교사용 지도서”, 2001