**Lab 04**

|  |
| --- |
| 그림입니다. 원본 그림의 이름: YU_UI_RGB-10.png 원본 그림의 크기: 가로 2256pixel, 세로 3047pixel 프로그램 이름 : Adobe ImageReady |

|  |  |
| --- | --- |
| 과목명 | 프로그래밍 언어 |
| 교수님 | 김영탁 교수님 |
| 이 름 | 김주환 |
| 학 번 | 21812158 |
| 일 자 | 2021.03.25.목 |

**4. 키보드 Beep() 함수를 활용한 전자피아노 설계 및 구현**

* 1. 키보드로부터 전자 피아노 음에 설정된 문자를 입력 받고, 이 문자에 해당하는 (옥타브, 음 및 주파수)를 찾아 출력하는 함수 mappingKey\_to\_Freq() 함수를 작성하라.

char ch;

while ((ch = \_getch()) != 27) {

int freq = -1, i, j;

int temp = 0;

char alt\_on = 0;

if (GetAsyncKeyState(VK\_MENU) & 0x8000) {

alt\_on = 1;

}

if (65 <= ch && ch <= 90) freq = keys\_to\_freq[alt\_on][1][ch - 65];

else if (97 <= ch && ch <= 122) freq = keys\_to\_freq[alt\_on][0][ch - 97];

else freq = -1;

for (i = 0; i < 5; i++) {

for (j = 0; j < 7; j++) {

if (ch == keys\_to\_oct\_sound[i][j]) {

temp++;

break;

}

}

if (temp == 1) break;

}

if (freq != -1) {

printf("inputkey (");

if (alt\_on == 1) printf("Alt + ");

printf("%c) : %d oct %s", ch, i + 4, sound\_k[j]);

if (alt\_on == 1) printf("#");

printf(", freq(% 3d)\n", freq);

Beep(freq, 100);

}

else printf("input key (%c) is wrong key input..\n", ch);

}

}

void mapping\_key\_to\_freq(void) {

const char sound\_k[7][3] = { "도", "레", "미", "파", "솔", "라", "시" };

const int keys\_to\_freq[2][2][26] = {

// alt\_on == 0

{

{

523, 392, 330, 660, 1323, 698, 784, 880, -1, 988, -1, -1, 494,

440, -1, -1, 1048, 1396, 588, 1568, 1976, 349, 1176, 294, 1760, 262

}, {

4186, 3136, 2637, 5274, -1, 5588, 6272, 7040, -1, 7902, -1, -1, 3951,

3520, -1, -1, -1, -1, 4699, -1, -1, 2794, -1, 2349, -1, 2093

}

}, {// alot\_on == 1

{

554, 415, 330, 660, 1323, 740, 830, 932, -1, 988, -1, -1, 494,

466, -1, -1, 1109, 1480, 622, 1661, 1976, 370, 1245, 311, 1865, 277

}, {

4435, 3322, 2637, 5274, -1, 5920, 6645, 7459, -1, 7902, -1, -1, 3951,

3729, -1, -1, -1, -1, 4978, -1, -1, 2960, -1, 2489, -1, 2217

}

}

};

const int keys\_to\_oct\_sound[5][7] = {

{

122, 120, 99, 118, 98, 110, 109

}, {

97, 115, 100, 102, 103, 104, 106

}, {

113, 119, 101, 114, 116, 121, 117

}, {

90, 88, 67, 86, 66, 78, 77

}, {

65, 83, 68, 70, 71, 72, 74

}

};

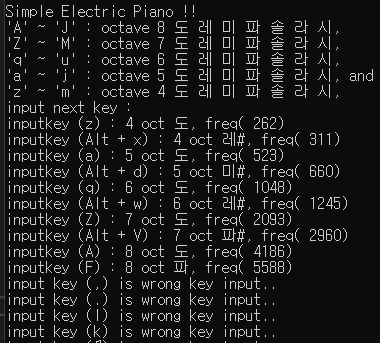
* 1. 전자 피아노 음에 설정되어 있지 않는 문자가 입력되면, 소리를 발생시키지 말고, 입력이 잘못되었다는 메시지를 출력할 것.



* 1. 만약 Alt 키가 함께 눌려졌으면 반음에 해당하는 소리가 발생되게 할 것.



* 1. 매 100ms 단위로 소리를 발생시킨 후, 문자 입력 상태를 파악하여 해당 문자에 해당하는 (옥타브, 음 및 주파수)를 찾아 출력하는 동작을 반복하게 할 것.



Oral Test

|  |
| --- |
| **(1) 표준입력장치로부터의 개별 문자 입력에서 버퍼를 사용하는 방식과 입력 버퍼를 사용하지 않는 방식의 차이점을 설명하고, 각각 어떤 함수를 사용하여야 하는지에 대하여 설명하라.**  헤더파일 stdio.h 로부터 함수를 호출한다.   * 입력 버퍼 사용 : getchar() 버퍼를 사용하며, 표준 입력장치로부터 하나의 문자를 읽어서 int 데이터 형으로 반환한다. * 입력 버퍼 미사용 : \_getch(),  버퍼를 사용하지 않고, 입력된 하나의 문자를 읽어 반환한다. 문자가 입력될 때, 화면에 표시(에코)해주지 않는다. \_getche() 를 사용하면 \_getch()와 다르게 화면에 표시(에코)해준다. |

|  |
| --- |
| **(2) 입력 문자로부터 key\_to\_freq 매핑 테이블의 인덱스 (alt\_on, shift\_on, ch\_index)를 산출하는 방법에 대하여 설명하라.**   * alt\_on GetSynchKeyState() 함수를 이용하여 인수는 VK\_MENU를 사용한다. 이 경우 0x8000(이전에 누르지 않았고 호출 시점에 눌려 있는 상태)과 bit\_AND 연산을 통해 alt 키가 입력되었는지 확인한다. * shift\_on ASCII 코드를 이용해 대소문자를 구분하여 shift 키가 입력 되었는지 확인한다. * ch\_index ASCII 코드를 이용해 몇번째 알파벳인지 판단한다. 소문자면 입력 값 - 97, 대문자면 입력값 - 65를 해준다.   (VK\_CONTROL : Control키가 입력되었는지 판단) |

|  |
| --- |
| **(3) 전자피아노 구현에서 mappingKey\_to\_Freq() 함수를 switch – case 구조로 구현하는 경우와 key\_to\_freq 매핑 테이블을 사용하는 구조로 표현하는 경우의 장단점을 비교하여 설명하라.**  이 비교는 필요한 연산의 최대, 최소, 평균 횟수를 통해 설명할 수 있다.  switch – case의 경우에는 총 35개의 알파벳 case 에 alt가 입력 되었는지 판단하는 조건문을 단다고 하면 입력 오류까지 포함해서 최대 71회 연산하고 평균적으로 35.5회 연산한다. 반대로 mapping table 의 경우에는 키 입력에 맞춰서 해당하는 항목을 바로 찾아낼 수 있기 때문에 1회 연산한다.  따라서, 두 경우의 연산 횟수로 비교해보면 mapping table을 사용하는 것이 더욱 효율적임을 알 수 있다. |

|  |
| --- |
| **(4) 키보드와 Beep() 함수를 사용하여 전자피아노를 구현하는 경우, 전자피아노 음에 설정되어 있지 않은 키가 입력되는 경우에 어떻게 오류 처리를 하는가에 대하여 설명하라.**   * switch – case 각 switch 블록 마지막에 defalt: 를 사용하여  printf(“This input is wrong . . .\n”);  break; 설정한 case 이외의 입력을 받았을 때 오류 문장을 출력하고 switch문을 탈출한다. * mapping table 생성한 배열 자체에서 지원하지 않는 키를 ‘-1’로 초기화한다. 하지만 알파벳 이외의 값을 입력 받을 경우 해당 입력은 배열에 할당 되어 있지 않으므로 쓰레기 값을 반환하기 때문에 조건문을 사용하여 알파벳 이외의 값을 ‘-1’로 초기화하고 오류 문장을 출력한다. |