Course: DT265

Lecturer: Dr. Bryan Duggan,

Module: Software Engineering

Assignment: ABC player.

Student: David Flynn D11127785

Note: I used c# 2010 express which means these programs might not open on visual studio at least not on the cs network.

Introduction:

I have included five solutions in the zip file: Assignment\_may2012, 06jun\_3, 07jun12\_2, 14jun12 & 15june12. I apologise for the names given. I also include a code directory with a sub directory for each solution just containing the cs file which can be copied directly into visual studio if need be. In fact my solution files will not run when extracted from a zip file so you might have to copy and paste the cs file from the code directory! Assignment\_may2012 is my initial attempt. 06jun\_3 is my next attempt using a string array. 07jun12\_2 is my first proper working version using a character array. 14jun12 is same as 07jun12\_2 but I use methods and finally 15june12 uses objects.

Assignment\_may2012:

This was my initial attempt at the assignment. My original thought was to encode the above frequencies into an array. Create a loop with nested if statements. The string which would be the input would be converted to a character Array. I would also have a output integer array for the sequence of frequencies to pass to the console. Each character on the string would be evaluated and if the next evaluates as a apostrophe or comma i would update the position in the output array accordingly.

I decided to create a simple test program. I selected four test frequencies: A, A a a' and get these to work in sequence and if necessary replicate in part for a more fuller example.

{

class Program

{

static void Main(string[] args)

{

///char SpecialCase="'";

int f = 293, duration = 250;

Console.Beep(f, duration);

///Console.Beep(440, 500);

string MusicInput = "A,Aa'a";

///char[] MusicInputArray = Console.Read().ToCharArray();

char[] MusicInputArray = MusicInput.ToCharArray();

int x = MusicInputArray.Length;

float[] frequency = {220F, 439.99F, 1759.97F, 879.99F};

int[] output = new int[x];

for(int i=0,j=0;i<(x-1);i++)

{

if((MusicInputArray[i]=='A') && (MusicInputArray[i+1] == ','))

{

output[j]=220;

j++;

}

else if ((MusicInputArray[i] == 'A')&& (!(MusicInputArray[i+1] == ',')))

{

output[j]=440;

j++;

}

else if ((MusicInputArray[i] == 'a') && (MusicInputArray[i + 1] == '0039'))

{

output[j] = 1760;

j++;

}

else if ((MusicInputArray[i] == 'a')&& (!(MusicInputArray[i + 1] == '\u0039')))

{

output[j] = 880;

j++;

}

}

for (int i = 0; i < (x-1); i++)

{

Console.Beep(output[i], duration);

}

}

}

}

The above is my first overall basic attempt at understanding the challenge. I convert my initial test music string to a character type array..

My Character array is MusicInputArray whose size is the length of the input string: MusicInput. The first for loop iterates through input array and modifies the output array according to the “j” integer. The next for loop with output each frequency for a fixed duration.

06jun12\_3:

This program was another approach to this assignment. I decided to try to attempt to solve the issue using string arrays with each element corresponding to a specific note such as “C,” or “c’”. Unfortunately I had to gave on this approach as the program would work for music1,music2 and music3 but it failed on music4. The reason been that this program searched the whole string for each element of the note array, however if the note was found a second time around, it would find it at the same position before. I was pondering how to fix this issue when I decided to go back my initial approach which was to use character array.

07jun12\_2:

This is version that works for all five string types plus any other combination. It is also fully commented within the code. It converts each string to a character array. There is a output array and a duration array. The output array contains the frequencies which will be called by the console.beep function. The duration array will contain the duration each note is played for. The standard is 250 ms.

The main functionality is a for loop which iterates for 10 times the length of the string. I appended a “$” character to the end of each string and there are various break statements within the for loop which activate when this character is reached. The string is converted to a Character Array. On each iteration the for loop will go through the notes array which contains each note. If the loop finds a match between the current character in musicArray with the notes array then the first element of the output array is generated by assigning the same value as the frequency array whose index is the same as the notes array. Then index is incremented which the reference for the output array. Then ‘i’ is incremented

which keeps track of each element of musicArray and is incremented on each character been correctly evaluated. ‘j’ is incremented on each iteration as musicArrray[i] is evaluated against notes[j](or musicArray[i+1] conversely).

For the special case of a ornament , i.e ‘~’; as one could relate to the comments here: Num\_ref converts the digit obtained from the string to a int value, i could have also simply evaluated musicArray[i+2] directly as either '2'or '3'. If a ~ is found in the string, this represents a roll. So e.g ~D2 becomes E50, D250, C50, D200; four different notes with different duration. So a if statement generates the corresponding entries into the output array plus duration.

When a character match is found and the next character is evaluated as a number, I simply multiply the duration by that number and ‘i’ is incremented twice. The other if statements handle the scenario whereby a letter is followed by a comma or apostrophe.

Finally when this loop terminates; I have another for loop to actually play the notes using the console.beep function.

14jun12:

This version introduces methods. From the main I called a method AssembleMusic which will play the music according to the string provided. ’AssembleMusic’ itself will call a method called ‘PlayMusic’ to play the tunes once assembled.

15june12:

This example utilises classes and object. I create a class called Music. In the main I declare the string want to test to be ‘MusicTest’ of type Music. I then call a method associated with that class and available to the object called ‘AssembleMusic’, which converts the string to a sequence of frequencies which are then called by the method MusicTest.PlayMusic() on the next line.