

Create a soundboard in C#

This assumes an intermediate knowledge of C#.

Create a new Windows Form Project

Call it Soundboard

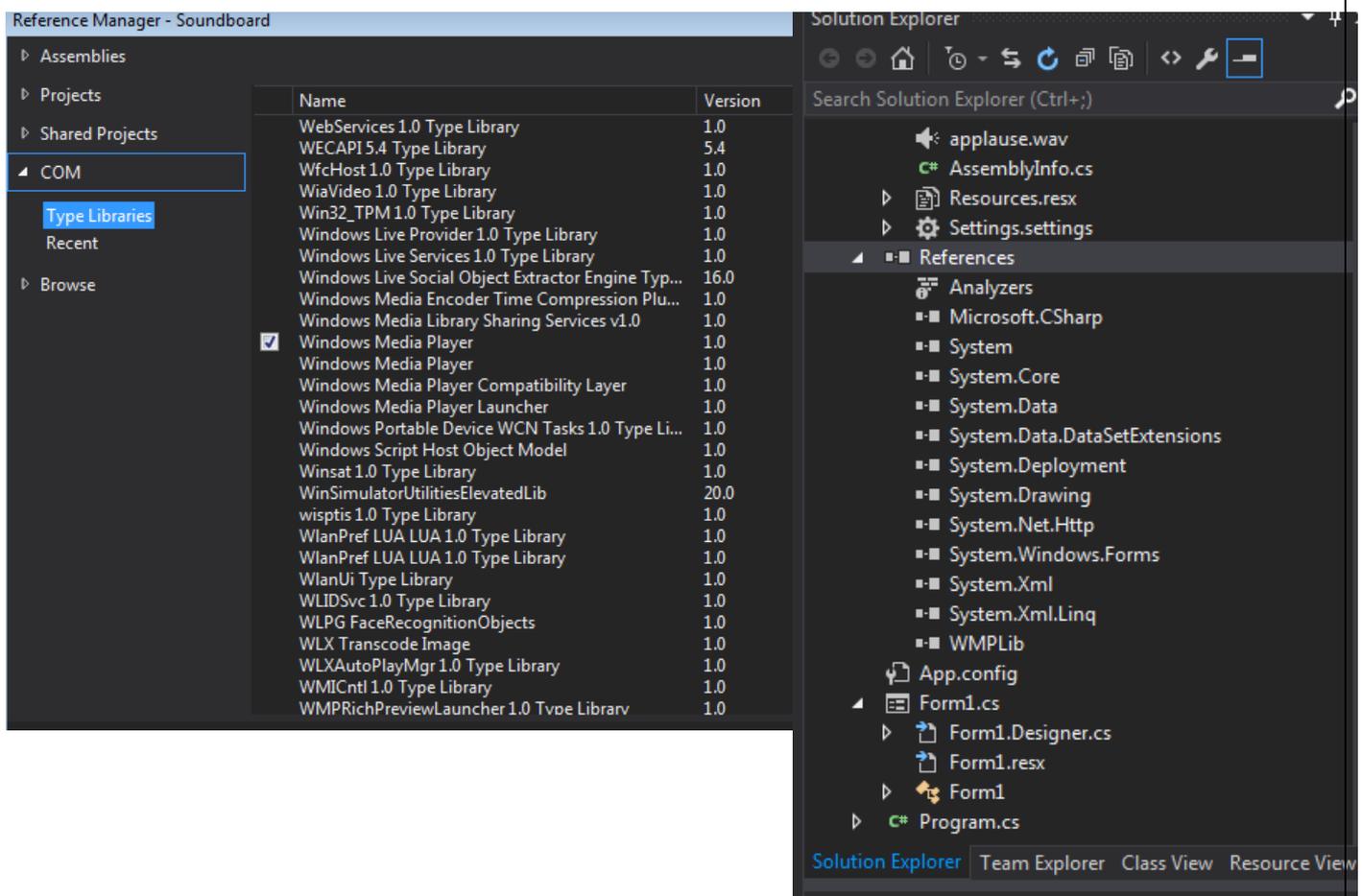
Add six new Buttons to the form

Go to opengameart or similar and download some free sound effects, the example I used is applause and evil laugh3.

In solution explorer right click References (as highlighted in below) - > Add reference

Click Com On the left then find windows media player

Click it then click in the little checkbox that appears and click OK



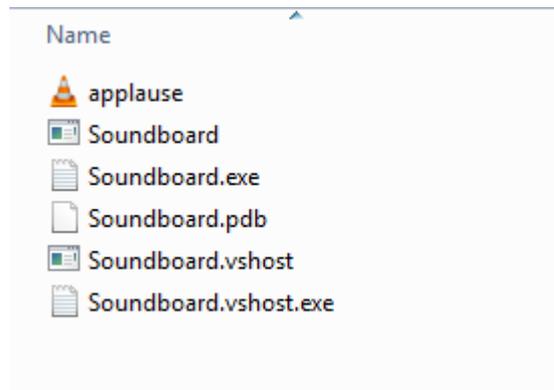
WMPLib should appear on the right when you expand references

Copy the sounds to your app folder- the usual directory is My Documents/ Visual Studio 20XX/ Projects/ Soundboard/ Soundboard/ bin / Debug for apps that we launch from Visual Studio. Apps that are built for release are / Release. Here Soundboard is the project name and Visual Studio 20XX is the visual studio year.

Rename the Text of the button to something relevant to the sound – i.e. because I used applause I changed text to applause.

Double click the button to add an event handler

Inside the event handler type the following;



```
private void button1_Click(object sender, EventArgs e)
{
    WMPLib.WindowsMediaPlayer wplayer = new WMPLib.WindowsMediaPlayer();
    wplayer.URL = @"applause.wav";
    wplayer.controls.play();
}
```

You'll need to change the *applause.wav* bit to the name AND extension of whatever you have used- for instance if the file is hello and it's an mp3 it would be @"hello.mp3".

An explanation of this;

We use the WindowsMediaPlayer type, which is from the WMPLib that we imported earlier, we create a new one and then we give it the URL of what we want to play. (This could be a web URL or a local file like this one).

We then send it the command to play.

Repeat this process for each of the other buttons adding in other sounds (change @applause to other elements). Try it out and they should now all play.

We're going to add two more things to this;

The first will be a button that when you click it, it adds all the clips to a playlist then plays the playlist. This will use a for loop, iteration, to do so and this is below.

The second will be a help menu, or you could add lots of things to it but help is what we will.

Add to your program a new button and call it "Play All", double click the button and see below.

```

private void button7_Click(object sender, EventArgs e)
{
    String[] clips = new String[2];

    clips[0] = "applause.wav";
    clips[1] = "evil laugh3.mp3";

    WMPLib.WindowsMediaPlayer wplayer = new WMPLib.WindowsMediaPlayer();

    WMPLib.IWMPPlaylist playlist = wplayer.newPlaylist("MyPlayList", "");

    for (int i=0;i<clips.Length;i++)
    {
        playlist.appendItem(wplayer.newMedia(@clips[i]));
    }

    wplayer.currentPlaylist = playlist;
    wplayer.controls.play();
}

```

Here you can see an example with two clips added, let's break this down line by line.

`String[] clips = new String[2]` - this means create a new array of Strings, an array is similar to a list, it's a collection of strings. The 2 in the box [] means it contains 2 elements, you may want more in yours.

`Clips[0] = "applause.wav"` and `clips[1] = "evil laugh3.mp3"` - These assign the string values of the file names to the spot 0 and 1 of an array. Arrays start at 0 not 1 remember this.

Similarly to before the next line creates a MediaPlayer element.

The line below similarly creates a playlist named "MyPlayList"

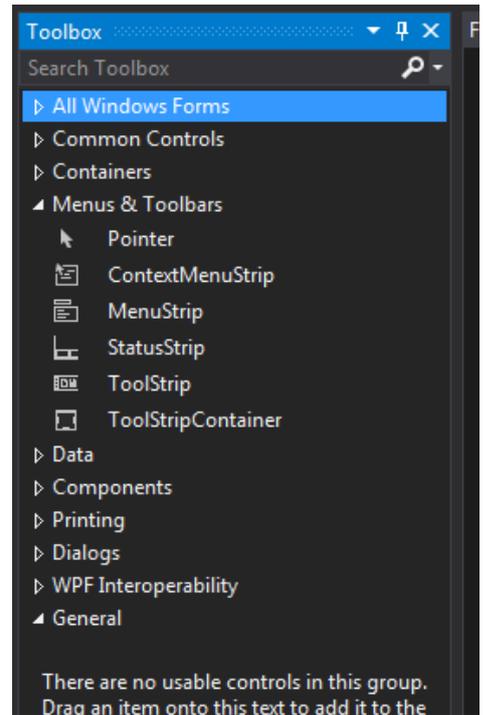
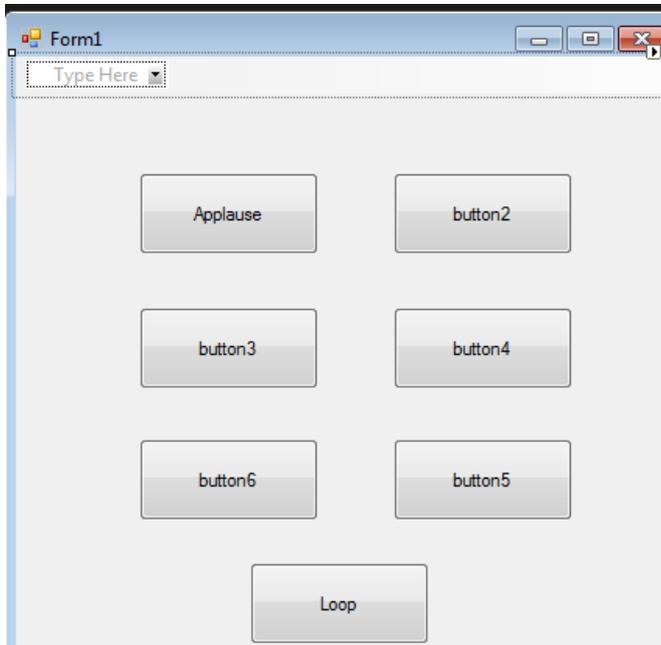
Then the loop, the for loop starts at 0, goes until it has reached the number of elements in clips and adds 1 each time (i++ means add 1 to i).

Within each time the loop goes around we add an item to the playlist, note that `wplayer.newmedia` just loads the clip into the media player. `@clips[i]` means the string represented by the ith number of clips. If I is 0 that's the first one, if 1 the 2nd and so on. We add the @ so it forces C# to ignore any special symbols and just take it as is (useful if we have things like a /n in the string for instance).

Finally, we set the current playlist in the mediaplayer to playlist and play the music. Give it a try!

Now, onto the menu.

From your toolbox choose Menus and Toolbars, a menu is an important program element, choose MenuStrip and drag it onto your program. In the section that says type here, type “Help”



Now, we’re going to add a new Form that we will switch to for a help screen.

Go to project a the top of your screen, click add new item, choose form and click ok. You should get a new form if you’re using default names called “Form2”. It should appear, add some help text for the player, don’t forget to add names.

Let’s add code for that, double click Help that you added earlier. You should have a piece of code added similar to the one below without the two lines between the { }. Add them now, we just create a new Form2 and then show it. C# and visual studio takes care of the rest.

```
private void helpToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form2 helpForm = new Soundboard.Form2();

    helpForm.Show();
}
```

That brings to an end this tutorial, I hope you enjoyed, look at adding colour and style to your app, adding more menu functions.