```
using System;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Windows.Forms;
using System.IO;
using libZPlay;
using System.Diagnostics;
using WMPLib;



namespace Jukebox
{
    public partial class Jukebox : Form
    {
        //WindowsMediaPlayer player = new WindowsMediaPlayer();
        TStreamInfo info;
        //public static int StSndBtnClickCnt;
        char[] EOLS = { '\n' };

        //public
        //statics to access them from other form

        public static string[][][] SyllSep;//where separate syllables stored
        public static int[][] syllables;//keeps # of syllables per word
        public static int[] totalSyll;//keeps number of syllables per line of
          lyrics
        public string[][] words;//individual words storage
        public static int[] numWords;//number of words per line
        public static int numBeats;//
        public static Stopwatch stopwatch;
        public static ZPlay player;//plays MP3 and WAV files (pretty versatile)
        public static string[] BeatStr;//stores the string of beats when read
          from file
        public static double[] Beats;//stored doubles of beats after read from
          file

        //private
        PopOut Karaoke;
        static string[] Lines;  //hold strings of the lines of lyrics
        static double[] beatTimes;  //holds beats
        static int maxNumBeats; //so we dont have to make new beat array every
          time we add
        string[] totalSyllstr;//holds number of syllables per line
        string musicfilename;

        //Everyone Needs a constructor
        public Jukebox()
        {
            InitializeComponent();
```

```csharp

            btnAnaMusic.Enabled = false;
            btnChooseMusic.Enabled = false; //disable buttons so you can't do things out of order
            btnChooseBeatFile.Enabled = false;
            btnTellBeat.Enabled = false;
            btnStopCount.Enabled = false;
            btnAnaMusic.Enabled = false;
            StKar.Enabled = false;


            maxNumBeats = 500;
            beatTimes = new double[maxNumBeats];
            numBeats = 0;

            stopwatch = new Stopwatch();
        }

        //Called when Open Lyrics btn pushed
        private void btnOpenFile_Click(object sender, EventArgs e)
        {
            OpenFileDialog dialog = new OpenFileDialog();
            dialog.Title = "Select the Lyrics";
            dialog.Filter = "TXT files|*.txt";
            dialog.InitialDirectory = @"C:\Users\[USER]\Documents\Visual Studio 2015\Projects\Jukebox\bin\Debug";

            if (dialog.ShowDialog() == DialogResult.OK)
            {
                string filename = dialog.FileName;
                Lines = File.ReadAllLines(filename);
                tbLyrics.Text = string.Join("\n", Lines);
                btnLyricAnalyze_Click(sender, e);
                btnChooseMusic.Enabled = true;
            }//end if
            else
            {
                MessageBox.Show("Please retry choosing a Lyrics file");
                btnOpenFile.Focus();
            }
        }

        //Doesn't do anything
        private void OFDLyrics_FileOk(object sender, CancelEventArgs e)
        {

        }

        //Called on Open Music btn press
        private void btnChooseMusic_Click(object sender, EventArgs e)
        {
            OpenFileDialog dialog = new OpenFileDialog();
            dialog.Title = "Select the Music File";
            dialog.Filter = "MP3 files|*.mp3| WAV files|*.wav";//order these are in determines which is pulled up first
            dialog.InitialDirectory = @"C:\Users\[USER]\Documents\Visual Studio 2015\Projects\Jukebox\bin\Debug";
            if (dialog.ShowDialog() == DialogResult.OK)
            {
                musicfilename = dialog.FileName;
                btnAnaMusic.Enabled = true;
                player = new ZPlay();
                player.OpenFile(musicfilename, TStreamFormat.sfAutodetect);
                player.GetStreamInfo(ref info);
                btnChooseBeatFile.Enabled = true;
                btnTellBeat.Enabled = true;             //let users use next set of buttons
                btnStopCount.Enabled = true;
                btnAnaMusic.Enabled = true;
            }//end if
            else
            {
                MessageBox.Show("Please try selecting music again");
                btnChooseMusic.Focus();
            }


        }

        //Separates word into syllables
        private string[] SeparateWord(string word, int numSylls)
        {
            string[] ret = new string[numSylls];
            if (numSylls == 1)  //if word is only one syllable, return the word
            {
                ret[0] = word;
                return ret;
            }

            string temp = "";

            int j = 0;//need to use later so not created in for loop

            for (int i = 0; i < numSylls; i++)
            {
                for (j = i * ((word.Length) / numSylls); j < ((i + 1) * ((word.Length) / numSylls)); j++)
                {                               //hardest part was figuring ^^this^^ out to be general for all words
                    temp += word[j];                            //goes through until you hit index where next syllable starts
                }

                ret[i] = temp;
                //MessageBox.Show(ret[i]);
                temp = "";
            }

            if (j != word.Length)   //see, told you we'd use it later
                ret[numSylls - 1] += word.Substring(j); //makes sure all the string is in a syllable!

            return ret;

        }

        //starts the Karaoke, transfers syllables to karaoke form/popout
        private void StSndBtn_Click(object sender, EventArgs e)
        {
            char[] vowels = new[] { 'a', 'e', 'i', 'o', 'u', 'y' };
            if (Lines.Length != 0)
            {
                SyllSep = new string[Lines.Length][][];


                numWords = new int[Lines.Length];

                for (int i = 0; i < Lines.Length; i++)
                {
                    numWords[i] = 0;
                    numWords[i] = Jukebox.Lines[i].Split(' ').Length;   //this counts the number of words per line
                    SyllSep[i] = new string[numWords[i]][];             //creates second layer of the syllables array

                    for (int j = 0; j < numWords[i]; j++)
                    {
                        SyllSep[i][j] = SeparateWord(words[i][j], syllables[i][j]); //puts the separate syllables into that array
                    }//end for

                }//end for


                Karaoke = new PopOut();             //creates the Karaoke PopOut

                numWords = new int[Lines.Length];
                for (int i = 0; i < Lines.Length; i++)
                {
                    numWords[i] = new int();
                    numWords[i] = Jukebox.Lines[i].Split(' ').Length - 1;   // gets you number of words per line, puts in array

                    for (int j = 0; j < numWords[i]; j++)
                    {
                        for (int k = 0; k < syllables[i][j]; k++)
                        {
                            Karaoke.SepLines[i][j][k].Text = SyllSep[i][j][k];      //gets the text to the karaoke popout
                        }

                    }
                }

                Karaoke.ShowDialog();
            }
            else
            {
                MessageBox.Show("Can't do Karaoke with no lyrics!");
                btnOpenFile.Focus();
            }

            // this.Hide();
        }

        //Button doesn't do anything for the karaoke but allows to find BPM
        private void btnAnaMusic_Click(object sender, EventArgs e)
        {

            ZPlay player = new ZPlay();
            if (player.OpenFile(musicfilename, TStreamFormat.sfAutodetect) == false)
            {
                MessageBox.Show("couldn't open music file");
            }
            else
            {
                TStreamInfo info = new libZPlay.TStreamInfo();
                player.GetStreamInfo(ref info);
                //player.StartPlayback();
                MessageBox.Show("Music File " + musicfilename + " opened");
                MessageBox.Show("BPM: " + player.DetectBPM(TBPMDetectionMethod.dmAutoCorrelation));

                if (musicfilename == "")
                {
                    MessageBox.Show("Silly user - you must specify a job file name at the command line!");
                    //error number!
                }

            }

        }

        //returns # of syllables in a word
        private int SyllableCount(string word)
        {
```

```csharp
            //adapted from https://codereview.stackexchange.com/questions/9972/syllable-counting-function
            word = word.ToLower().Trim();
            bool lastWasVowel = false;

            //way to improve this function:
            //maybe make an array of exceptions and their actual number of beats and go through that array
            //instead of coding individual words in
            //could go on main screen to have users add in exceptions

            var vowels = new[] { 'a', 'e', 'i', 'o', 'u', 'y' };
            int count = 0;
            foreach (var c in word)
            {
                if (vowels.Contains(c))
                {
                    if (!lastWasVowel)
                        count++;
                    lastWasVowel = true;
                }
                else
                    lastWasVowel = false;
            }
            if ((word.EndsWith("e,") || word.EndsWith("e") || (word.EndsWith("es") || word.EndsWith("ed"))) && !word.EndsWith("le"))
                count--;

            if (word.Contains("oah") || word.Contains("io") || word.Contains("dn't") || word.Contains("ded"))
                count++;

            //*********EXCEPTIONS****************
            if (word.ToUpper() == "WHILE" || word.ToUpper() == "WHILE,")
                count = 1;

            //WILL CAUSE PROBLEMS IF ME-MO-RIES instead of mem-ries, will depend on song
            if (word.ToUpper() == "MEMORIES" || word.ToUpper() == "MEMORIES,")
                count = 2;

            if (word.ToUpper() == "MAMA" || word.ToUpper() == "MAMA," || word.ToUpper() == " MAMA")
                count = 2;

            //*********END OF EXCEPTIONS**********

            if (count == 0) //fixes problem with words like "blue" and "the"
                return 1;
            else
                return count;

```

```
        }

        //makes popout to see # of syllables in the lyrics
        public void btnLyricAnalyze_Click(object sender, EventArgs e)
        {
            totalSyll = new int[Lines.Length];
            words = new string[Lines.Length][];
            syllables = new int[Lines.Length][];
            for (int i = 0; i < Lines.Length; i++)
            {
                words[i] = Lines[i].Split(' ');
                syllables[i] = new int[words[i].Length];
                for (int j = 0; j < words[i].Length; j++)
                {
                    syllables[i][j] = new int();
                    syllables[i][j] = SyllableCount(words[i][j]);
                }
            }

            if (cbSyllCount.Checked == true) //displays syllables in new form
            {
                Syllable_Count SeeSylls = new Syllable_Count();
                SeeSylls.ShowDialog();
            }

            int linesum = 0;
            totalSyllstr = new string[Lines.Length];

            for (int i = 0; i < Lines.Length; i++)
            {
                for (int j = 0; j < words[i].Length; j++)
                    linesum += syllables[i][j];

                totalSyllstr[i] = linesum.ToString();
                totalSyll[i] = linesum;
                linesum = 0;
            }
        }

        //called every time lyrics change in the text box
        private void tbLyrics_TextChanged(object sender, EventArgs e)
        {
            Lines = tbLyrics.Text.Split(EOLS);
            btnLyricAnalyze_Click(sender, e);
        }

        //starts process of creating beat file
        private void btnTellBeat_Click(object sender, EventArgs e)
        {

            if (numBeats == 0)
                stopwatch.Start();
            //if (StSndBtnClickCnt == 0)
            player.StartPlayback();

            if (numBeats >= maxNumBeats - 1)
            {
                double[] temp = new double[maxNumBeats + 2];        //probably could be more efficient but dont want
                beatTimes.CopyTo(temp, 0);                          // file to get too long without having data in lines
                maxNumBeats = maxNumBeats + 10;
                beatTimes = new double[maxNumBeats];
                temp.CopyTo(beatTimes, 0);
            }

            beatTimes[numBeats] = stopwatch.Elapsed.TotalMilliseconds - 150;    // -150 is correct for time to click button
            numBeats++;
        }

        //called when Stop Beat Count btn pressed (ends creating beat file)
        private void btnStopCount_Click(object sender, EventArgs e)
        {
            Array.Copy(beatTimes, 1, beatTimes, 0, beatTimes.Length - 1);
            if (tbBeatFile.Text != "")
                File.WriteAllLines(tbBeatFile.Text + ".txt", beatTimes.Select(d => d.ToString()));
            else
                File.WriteAllLines(musicfilename.Remove(musicfilename.Length - 4) + "Beat.txt", beatTimes.Select(d => d.ToString()));
        }

        //Used as accessor for Karaoke PopOut
        public static string[] GetLines()
        {
            return Lines;
        }

        //accessor for syllable array(converted to strings)
        public static string[] GetSyll()
        {
            string[] newsyll = new string[Lines.Length];

            string temp = "";

            for (int i = 0; i < Lines.Length; i++)
            {
                for (int j = 0; j < syllables[i].Length; j++)
                    temp += syllables[i][j];

                newsyll[i] = temp;
                temp = "";
            }
```

```
                return newsyll;
            }

            //onClick of Choose Beat File button
            private void button1_Click(object sender, EventArgs e)
            {
                OpenFileDialog dialog = new OpenFileDialog();
                dialog.Title = "Select the Beat File";
                dialog.Filter = "TXT files|*.txt";
                dialog.InitialDirectory = @"C:\Users\[USER]\Documents\Visual Studio
                  2015\Projects\Jukebox\bin\Debug";

                if (dialog.ShowDialog() == DialogResult.OK)
                {
                    string filename = dialog.FileName;
                    BeatStr = File.ReadAllLines(filename);  //get beats from file
                    Beats = new double[BeatStr.Length];
                    for (int q = 0; q < BeatStr.Length; q++)
                    {
                        Beats[q] = Convert.ToDouble(BeatStr[q]);   //convert to
                         doubles
                    }
                    StKar.Enabled = true;       //let the people sing!


                }//end if
                else
                {
                    MessageBox.Show("Couldn't open file, please try again.");
                    btnChooseBeatFile.Focus();
                }

                return;

            }

        }
    }

```