```
 1  using System;
 2  using System.Collections.Generic;
 3  using System.ComponentModel;
 4  using System.Data;
 5  using System.Drawing;
 6  using System.Linq;
 7  using System.Text;
 8  using System.Threading.Tasks;
 9  using System.Windows.Forms;
10  using System.IO;
11  using libZPlay;
12  using WMPLib;
13  using System.Diagnostics;
14  using System.Threading;
15  //using System.Timers;
16
17
18
19  namespace Jukebox
20  {
21      public partial class PopOut : Form
22      {
23          //public
24          public Label[][][] SepLines;//labels of karaoke symbols [Line][Word]
              [Syllable]
25          public int[] numSylls;//number of syllables per line
26          //public int[][] totSyll;//number of syllables per word
27
28          //private
29          static string[] Lyrics; //holds string of lines of lyrics
30          int[] numWords; //number of words per line
31          int[][] SyllPerWord;    //number of syllables per word
32          string[] BeatStr;
33          double[] Beats;
34          //int[] fontSize;       //beginning of matching font sizes to be based on
               window size
35          public static Stopwatch CountDown;
36          private int curBeatLine;
37          private int i, j, k;
38
39          public PopOut()
40          {
41              InitializeComponent();
42              //this.FormBorderStyle = FormBorderStyle.Fixed3D;   //using these two
                   makes the popout unclosable
43              //this.WindowState = FormWindowState.Maximized;
44
45              FormClosing += PopoutClosing;   //how to add code when form closes
46              Lyrics = new string[Jukebox.GetLines().Length];
47              Beats = Jukebox.Beats;
48              BeatStr = Jukebox.BeatStr;
49              SyllPerWord = Jukebox.syllables;
```

```
50              InitLyrics();
51          }
52
53          //initializes lyrics for songs, puts first 3 lines on PopOut
54          public void InitLyrics()
55          {
56
57              Lyrics = Jukebox.GetLines();
58
59              numWords = new int[Lyrics.Length];
60              SepLines = new Label[Lyrics.Length][][];
61
62              //fontSize = new int[Lyrics.Length];    //was start of dynamically
                  making font size per window size
63
64              for (int Line = 0; Line < Lyrics.Length; Line++)
65              {
66                  numWords[Line] = 0;
67                  numWords[Line] = Lyrics[Line].Split(' ').Length;//gets number of
                      words per line
68
69                  SepLines[Line] = new Label[numWords[Line]][];
70                  for (int word = 0; word < numWords[Line]; word++)
71                  {
72                      SepLines[Line][word] = new Label[SyllPerWord[Line][word]];//
                       creating k syllables
73
74                      for (int syl = 0; syl < SyllPerWord[Line][word]; syl++)
75                      {
76
77                          SepLines[Line][word][syl] = new Label();
78                          SepLines[Line][word][syl].Text = Jukebox.SyllSep[Line]
                        [word][syl];
79                          SepLines[Line][word][syl].AutoSize = true;
80                          SepLines[Line][word][syl].ForeColor = Color.WhiteSmoke;
81
82                          Controls.Add(SepLines[Line][word][syl]);
83
84                          if (Line > 2)   //only shows first 3 lines
85                          {
86                              SepLines[Line][word][syl].Hide();
87                              SepLines[Line][word][syl].Update();
88                          }
89
90                          SepLines[Line][word][syl].Update(); //needed to update
                        the screen (VERY IMPORTANT)
91                      }//end k
92                  }
93
94
95              }//end for
96
```

```
 97
 98              //******************************Puts first 3 lines on screen at
                   beginning*********************************************
 99
100
101             for (int lineMov = 0; lineMov < 3; lineMov++)//need to fix for end of
102             {
103                 for (int wordMov = 0; wordMov < numWords[lineMov]; wordMov++)
104                 {
105                     for (int syllMov = 0; syllMov < SyllPerWord[lineMov]
                        [wordMov]; syllMov++)
106                     {
107
108                         //only documenting first if, rest are same
109                         if (lineMov == 0)//don't have to check
110                         {
111                             SepLines[lineMov][wordMov][syllMov].Font = new Font
                        ("Arial", 40);   //focus line, big font
112                             SepLines[lineMov][wordMov][syllMov].ForeColor =
                        Color.WhiteSmoke;   //words are white then turn green to show
                         singing
113                             SepLines[lineMov][wordMov][syllMov].Show();
114
115                             if (wordMov == 0 && syllMov == 0)
116                             {                       //VV first line first word
117                                 SepLines[lineMov][wordMov][syllMov].Location =
                        new Point(this.Width / 8, (this.Height / 8) - 20);//kind of
                        arbitrary but should start it somewhere closer to middle
118                                 SepLines[lineMov][wordMov][syllMov].Update();
119                             }
120                             else if (syllMov == 0)
121                             {                       //VV use location of last
                        word, to calculate position of first syllable of next word, Y
                         is same as other word
122                                 SepLines[lineMov][wordMov][syllMov].Location =
                        new Point((SepLines[lineMov][wordMov - 1][SyllPerWord
                        [lineMov][wordMov - 1] - 1].Location.X + SepLines[lineMov]
                        [wordMov - 1][SyllPerWord[lineMov][wordMov - 1] - 1].Width),
                        SepLines[lineMov][0][0].Location.Y);
123                                 SepLines[lineMov][wordMov][syllMov].Update();
124                             }
125                             else
126                             {                       //otherwise use location of
                        last syllable to put next one on the screen
127                                 SepLines[lineMov][wordMov][syllMov].Location =
                        new Point(SepLines[lineMov][wordMov][syllMov - 1].Location.X
                        + SepLines[lineMov][wordMov][syllMov - 1].Width, SepLines
                        [lineMov][wordMov][syllMov - 1].Location.Y);
128                                 SepLines[lineMov][wordMov][syllMov].Update();
129                             }
130                         }
131                         if (lineMov == 1 && lineMov < Lyrics.Length)
```

```
132                         {
133                             SepLines[lineMov][wordMov][syllMov].Font = new Font
                        ("Arial", 30);
134                             SepLines[lineMov][wordMov][syllMov].ForeColor =
                        Color.WhiteSmoke;
135                             SepLines[lineMov][wordMov][syllMov].Show();
136 
137                             if (wordMov == 0 && syllMov == 0)
138                             {           //location based on line above, same x, y
                        +150 (lower down screen)
139                                 SepLines[lineMov][wordMov][syllMov].Location =
                        new Point(SepLines[lineMov - 1][0][0].Location.X, SepLines
                        [lineMov - 1][0][0].Location.Y + 150);
140                                 SepLines[lineMov][wordMov][syllMov].Update();
141                             }
142                             else if (syllMov == 0)
143                             {
144                                 SepLines[lineMov][wordMov][syllMov].Location =
                        new Point((SepLines[lineMov][wordMov - 1][SyllPerWord
                        [lineMov][wordMov - 1] - 1].Location.X + SepLines[lineMov]
                        [wordMov - 1][SyllPerWord[lineMov][wordMov - 1] - 1].Width),
                        SepLines[lineMov][0][0].Location.Y);
145                                 SepLines[lineMov][wordMov][syllMov].Update();
146                             }
147                             else
148                             {
149                                 SepLines[lineMov][wordMov][syllMov].Location =
                        new Point(SepLines[lineMov][wordMov][syllMov - 1].Location.X
                        + SepLines[lineMov][wordMov][syllMov - 1].Width, SepLines
                        [lineMov][wordMov][syllMov - 1].Location.Y);
150                                 SepLines[lineMov][wordMov][syllMov].Update();
151                             }
152                         }
153 
154                         if (lineMov == 2 && lineMov < Lyrics.Length)
155                         {
156                             SepLines[lineMov][wordMov][syllMov].Font = new Font
                        ("Arial", 20);
157                             SepLines[lineMov][wordMov][syllMov].ForeColor =
                        Color.WhiteSmoke;
158                             SepLines[lineMov][wordMov][syllMov].Show();
159                             if (wordMov == 0 && syllMov == 0)
160                             {
161                                 SepLines[lineMov][wordMov][syllMov].Location =
                        new Point(SepLines[lineMov - 1][0][0].Location.X, SepLines
                        [lineMov - 1][0][0].Location.Y + 300);
162                                 SepLines[lineMov][wordMov][syllMov].Update();
163                             }
164                             else if (syllMov == 0)
165                             {
166                                 SepLines[lineMov][wordMov][syllMov].Location =
                        new Point((SepLines[lineMov][wordMov - 1][SyllPerWord
```

```
                        [lineMov][wordMov - 1] - 1].Location.X + SepLines[lineMov]
                        [wordMov - 1][SyllPerWord[lineMov][wordMov - 1] - 1].Width),
                        SepLines[lineMov][0][0].Location.Y);
167                             SepLines[lineMov][wordMov][syllMov].Update();
168                         }
169                         else
170                         {
171                             SepLines[lineMov][wordMov][syllMov].Location =
                        new Point(SepLines[lineMov][wordMov][syllMov - 1].Location.X
                        + SepLines[lineMov][wordMov][syllMov - 1].Width, SepLines
                        [lineMov][wordMov][syllMov - 1].Location.Y);
172                             SepLines[lineMov][wordMov][syllMov].Update();
173                         }
174                     }
175
176                 }
177             }
178         }
179
180         //*****************************END of adding 3 lines on screen at
              beginning*****************************************
181
182
183         curBeatLine = 0;
184
185         CountDown = new Stopwatch();
186         CountDown.Start();
187         Jukebox.player.StartPlayback();
188
189         i = 0; j = 0; k = 0;    //globals(sorry Prof. McVey!), used to keep
              track of Line, Word, Syllable respectively
190
191         curBeatLine = 0;
192
193         CountDown = new Stopwatch();
194         CountDown.Start();
195         Jukebox.player.StartPlayback();
196
197         curBeatLine = 0;
198         timerAdvance.Start();
199
200     }//end InitLyrics
201
202     //called as timer tick, called every 50 milliseconds
203     private void AdvanceCursor(object sender, EventArgs e)
204     {
205         if (curBeatLine < Beats.Length && CountDown.ElapsedMilliseconds >
              (long)Beats[curBeatLine] - 150 && i < Lyrics.Length)
206         {
207             //********1*******This section is what makes the words turn green
208
209             //MessageBox.Show(k + " " + j + " " + i);
```

```
210                 SepLines[i][j][k].Show();
211                 SepLines[i][j][k].ForeColor = Color.ForestGreen;
212                 //SepLines[i][j][k].Refresh();
213                 SepLines[i][j][k].Update();
214                 //SepLines[i][j][k].Invalidate();
215                 //***************End of color update section***************
216                 //**************Now we account for variable line, word, syllable
                      lengths*****************
217
218                 k++;
219                 if (k >= SyllPerWord[i][j])
220                 {
221                     k = 0;
222                     j++;
223                     if (j >= numWords[i])
224                     {
225                         j = 0;
226
227                         //hides the line that just finished
228
229                         i++;
230                         for (int q = 0; q < numWords[i - 1]; q++)
231                         {
232                             for (int w = 0; w < SyllPerWord[i - 1][q]; w++)
233                             {
234                                 SepLines[i - 1][q][w].ForeColor = Color.Black;
235                                 SepLines[i - 1][q][w].Update();
236                                 SepLines[i - 1][q][w].Hide();
237
238                                 if (SepLines[i - 1][q][w].Visible == true)
239                                 {
240                                     SepLines[1 - 1][q][w].Visible = false;
241                                 }
242                                 SepLines[i - 1][q][w].Update();
243
244                             }
245
246                         }
247
248                         //**************Still in line, word, syll update, this
                        section moves next 3 lines up(only happens if 'i' was
                        incremented*****************
249
250                         for (int lineMov = i; lineMov < i + 3; lineMov++)//need
                        to fix for end of
251                         {
252                             if (lineMov < Lyrics.Length)
253                             {
254                                 for (int wordMov = 0; wordMov < numWords
                            [lineMov]; wordMov++)
255                                 {
256                                     for (int syllMov = 0; syllMov < SyllPerWord
```

```
                    [lineMov][wordMov]; syllMov++)
257                 {
258                     if (lineMov == i)
259                     {
260                         SepLines[lineMov][wordMov]
                    [syllMov].Font = new Font("Arial", 40);
261                         SepLines[lineMov][wordMov]
                    [syllMov].ForeColor = Color.WhiteSmoke;
262                         SepLines[lineMov][wordMov]
                    [syllMov].Show();
263                         if (wordMov == 0 && syllMov == 0)
264                         {
265                             SepLines[lineMov][wordMov]
                    [syllMov].Location = new Point(SepLines[lineMov - 1][0]
                    [0].Location.X, SepLines[lineMov - 1][0][0].Location.Y);
266                             SepLines[lineMov][wordMov]
                    [syllMov].Update();
267                         }
268                         else if (syllMov == 0)
269                         {
270                             SepLines[lineMov][wordMov]
                    [syllMov].Location = new Point((SepLines[lineMov][wordMov -
                    1][SyllPerWord[lineMov][wordMov - 1] - 1].Location.X +
                    SepLines[lineMov][wordMov - 1][SyllPerWord[lineMov][wordMov -
                     1] - 1].Width), SepLines[lineMov][0][0].Location.Y);
271                             SepLines[lineMov][wordMov]
                    [syllMov].Update();
272                         }
273                         else
274                         {
275                             SepLines[lineMov][wordMov]
                    [syllMov].Location = new Point(SepLines[lineMov][wordMov]
                    [syllMov - 1].Location.X + SepLines[lineMov][wordMov][syllMov
                     - 1].Width, SepLines[lineMov][wordMov][syllMov -
                    1].Location.Y);
276                             SepLines[lineMov][wordMov]
                    [syllMov].Update();
277                         }
278                     }
279                     if (lineMov == i + 1 && lineMov <
                    Lyrics.Length)
280                     {
281                         SepLines[lineMov][wordMov]
                    [syllMov].Font = new Font("Arial", 30);
282                         SepLines[lineMov][wordMov]
                    [syllMov].ForeColor = Color.WhiteSmoke;
283                         SepLines[lineMov][wordMov]
                    [syllMov].Show();
284
285                         if (wordMov == 0 && syllMov == 0)
286                         {
287                             SepLines[lineMov][wordMov]
```

```
                    [syllMov].Location = new Point(SepLines[lineMov - 1][0]
                    [0].Location.X, SepLines[lineMov - 1][0][0].Location.Y +
                    150);
288                                     SepLines[lineMov][wordMov]
                    [syllMov].Update();
289                                 }
290                                 else if (syllMov == 0)
291                                 {
292                                     SepLines[lineMov][wordMov]
                    [syllMov].Location = new Point((SepLines[lineMov][wordMov -
                    1][SyllPerWord[lineMov][wordMov - 1] - 1].Location.X +
                    SepLines[lineMov][wordMov - 1][SyllPerWord[lineMov][wordMov -
                     1] - 1].Width), SepLines[lineMov][0][0].Location.Y);
293                                     SepLines[lineMov][wordMov]
                    [syllMov].Update();
294                                 }
295                                 else
296                                 {
297                                     SepLines[lineMov][wordMov]
                    [syllMov].Location = new Point(SepLines[lineMov][wordMov]
                    [syllMov - 1].Location.X + SepLines[lineMov][wordMov][syllMov
                     - 1].Width, SepLines[lineMov][wordMov][syllMov -
                    1].Location.Y);
298                                     SepLines[lineMov][wordMov]
                    [syllMov].Update();
299                                 }
300                             }
301
302                             if (lineMov == i + 2 && lineMov <
                    Lyrics.Length)
303                             {
304                                 SepLines[lineMov][wordMov]
                    [syllMov].Font = new Font("Arial", 20);
305                                 SepLines[lineMov][wordMov]
                    [syllMov].ForeColor = Color.WhiteSmoke;
306                                 SepLines[lineMov][wordMov]
                    [syllMov].Show();
307                                 if (wordMov == 0 && syllMov == 0)
308                                 {
309                                     SepLines[lineMov][wordMov]
                    [syllMov].Location = new Point(SepLines[lineMov - 1][0]
                    [0].Location.X, SepLines[lineMov - 1][0][0].Location.Y +
                    300);
310                                     SepLines[lineMov][wordMov]
                    [syllMov].Update();
311                                 }
312                                 else if (syllMov == 0)
313                                 {
314                                     SepLines[lineMov][wordMov]
                    [syllMov].Location = new Point((SepLines[lineMov][wordMov -
                    1][SyllPerWord[lineMov][wordMov - 1] - 1].Location.X +
                    SepLines[lineMov][wordMov - 1][SyllPerWord[lineMov][wordMov -
```

```
                        1] - 1].Width), SepLines[lineMov][0][0].Location.Y);
315                                         SepLines[lineMov][wordMov]
                    [syllMov].Update();
316                                     }
317                                     else
318                                     {
319                                         SepLines[lineMov][wordMov]
                    [syllMov].Location = new Point(SepLines[lineMov][wordMov]
                    [syllMov - 1].Location.X + SepLines[lineMov][wordMov][syllMov
                     - 1].Width, SepLines[lineMov][wordMov][syllMov -
                    1].Location.Y);
320                                         SepLines[lineMov][wordMov]
                    [syllMov].Update();
321                                     }
322                                 }
323
324                                 }
325                             }
326                         }
327                     }
328
329                 }
330             }//end if
331             curBeatLine++;
332
333         }//end milliseconds if
334
335     }//end AdvanceCursor
336
337     //called when Karaoke PopOut is closed, stops music
338     private void PopoutClosing(object sender, FormClosingEventArgs e)
339     {
340         Jukebox.player.StopPlayback();
341         //Jukebox.player.Close();//putting this in maked user have to select
              music file over again
342     }
343
344 }//end form
345 }
346
```