```cpp
////////////////////////////////////////////////////////////
// Headers
////////////////////////////////////////////////////////////
//#include <SFML/Audio.hpp>
#include <SFML/Graphics.hpp>
#include<SFML/Window.hpp>
#include<Windows.h>
#include <iostream>
#include <string>
#include <list>
#include <fstream>
#include <cstdlib>
#include <ctime>



//this seems to be getting the sprites and then returning the sprite

sf::Sprite displayImage(std::string input, sf::Texture &texture)
{


    if (!texture.loadFromFile( input  + ".bmp"))
    {
        std::cout << "Image " << input << ".bmp failed to load\n";


        texture.loadFromFile("error.bmp");




    }


    sf::Sprite sprite;
    sprite.setTexture(texture);

    return sprite;




}

void  Listmaker(std::list <std::string> &flags)
{
```

```
50
51
52
53      std::string filename = "country_names.txt"; //gonna have to change to
          ktp
54
55
56      std::string line;
57      std::ifstream file(filename);
58
59      if (file.is_open())
60      {
61
62          while (std::getline(file, line))
63          {
64
65              flags.push_back(line);
66
67
68          }
69
70      }
71      else
72      {
73
74          std::cout << "error unable to read\n";
75
76      }
77
78
79      file.close();
80
81
82      /*for (auto i : flags)
83          std::cout << i << std::endl;*/
84
85
86
87
88
89  }
90
91  void textSetter(std::string input, sf::Text& text, sf::Font& font, int
      textposXChoice, sf::Color color)
92  {
93
94
95
96
```

```
 97      // select the font
 98      text.setFont(font); // font is a sf::Font
 99
100      // set the string to display
101      text.setString(input);
102
103      // set the character size
104      text.setCharacterSize(24); // in pixels, not points!
105
106      // set the color
107      text.setFillColor(sf::Color::Red);
108
109      // set the text style
110      text.setStyle(sf::Text::Bold | sf::Text::Underlined);
111
112
113
114      if (textposXChoice == 1)
115      {
116
117          text.setPosition(20.f, 200.f);
118
119
120      }
121      else if (textposXChoice == 2)
122      {
123
124
125          text.setPosition(20.f, 300.f);
126
127
128      }
129      else if (textposXChoice == 3)
130      {
131
132          text.setPosition(20.f, 400.f);
133
134
135
136      }
137      else if (textposXChoice == 4)
138      {
139
140          text.setPosition(20.f, 500.f);
141
142
143
144      }
145      else
```

```
146     {
147
148         // set the string to display
149         text.setString("error");
150
151         // set the character size
152         text.setCharacterSize(24); // in pixels, not points!
153
154         // set the color
155         text.setFillColor(sf::Color::Yellow);
156
157         // set the text style
158         text.setStyle(sf::Text::Bold | sf::Text::Underlined);
159
160         text.setPosition(20.f, 200.f);
161
162
163     }
164
165
166
167
168
169
170 }
171
172
173
174
175
176 void wrongAnswers(std::string input, sf::Text& text, sf::Font& font,
177     int textposXChoice, std::list <std::string> flags,
178     std::list <std::string> correctanswers, int& wrongtextpos1, int&
      wrongtextpos2, int& wrongtextpos3)
179 {
180
181
182
183     while (wrongtextpos1 == textposXChoice)
184     {
185         wrongtextpos1 = rand() % 4 + 1;
186
187         if (wrongtextpos1 != textposXChoice)
188         {
189
190             break;
191
192         }
193
```

```
194     }
195
196     while (wrongtextpos2 == textposXChoice || wrongtextpos2 ==
          wrongtextpos1)
197     {
198         wrongtextpos2 = rand() % 4 + 1;
199
200         if (wrongtextpos2 != textposXChoice && wrongtextpos2 !=
              wrongtextpos1)
201         {
202
203             break;
204
205         }
206
207     }
208
209     while (wrongtextpos3 == textposXChoice || wrongtextpos3 ==
          wrongtextpos1 || wrongtextpos3 == wrongtextpos2)
210     {
211         wrongtextpos3 = rand() % 4 + 1;
212
213         if (wrongtextpos3 != textposXChoice && wrongtextpos3 !=
              wrongtextpos1 && wrongtextpos3 != wrongtextpos2)
214         {
215
216             break;
217
218         }
219
220     }
221
222
223
224
225 }
226
227
228
229
230
231
232
233
234
235
236
237 int main()
238 {
```

```
239
240     std::string input = "test2";
241     sf::Texture texture;
242     sf::Color color = sf::Color::Red;
243
244     sf::RenderWindow window(sf::VideoMode(1000, 1000), "FlagGuesser");
245
246     int prevtextpos = 0;
247
248     int points = 100;
249
250     while (window.isOpen())
251     {
252         sf::Event event;
253         while (window.pollEvent(event))
254         {
255             if (event.type == sf::Event::Closed)
256                 window.close();
257
258         }
259
260         window.clear();
261
262
263
264
265         std::list <std::string> flags;
266         std::list <std::string> correctanswers;
267
268         Listmaker(flags);
269
270
271
272         srand(time(0));
273
274         //auto jk = rand() % flags.size() + 1;
275
276         std::list<std::string>::iterator iter = flags.begin();
277         std::string correctanswerholder;
278
279
280
281         for (int uppy = 0; uppy < 20; uppy++)
282         {
283
284             if (uppy == 20)
285             {
286
287                 window.close();
```

```
288                 break;
289
290             }
291
292             auto jk = rand() % flags.size();
293
294
295
296
297
298
299             advance(iter, jk);
300
301
302             window.clear();
303
304             window.draw(displayImage(*iter, texture));
305
306             correctanswers.push_back(*iter); //loads the correct answer
                  onto separate list
307
308
309
310             sf::Font font;
311             if (!font.loadFromFile("arial.ttf"))
312             {
313                 std::cout << "font error\n";
314             }
315
316             sf::Text text;
317             sf::Text wrong1, wrong2, wrong3;
318
319
320
321             int textposXChoice = rand() % 4 + 1;  //between 1 and 4
                  inclusive
322
323             while (prevtextpos != textposXChoice)
324             {
325
326                 int textposXChoice = rand() % 4 + 1;  //between 1 and 4
                      inclusive
327
328                 if (prevtextpos != textposXChoice)
329                 {
330                     break;
331
332                 }
333
```

```
334
335             }
336
337
338
339
340             textSetter(*iter, text, font, textposXChoice, color);
341
342             window.draw(text);
343
344             correctanswerholder = *iter;
345
346
347             //wrong answers don't trust iter after this line till end of
                  for loop
348
349             int wrongtextpos1 = 1;
350             int wrongtextpos2 = 1;
351             int wrongtextpos3 = 1;
352
353
354
355
356
357
358
359
360             wrongAnswers(*iter, text, font, textposXChoice, flags,
361                 correctanswers, wrongtextpos1, wrongtextpos2,
                      wrongtextpos3);
362
363
364             color = sf::Color::Green;
365             iter = flags.begin();
366             jk = rand() % flags.size();
367             advance(iter, jk);
368             textSetter(*iter, text, font, wrongtextpos1,color);
369             window.draw(text);
370
371
372             color = sf::Color::Blue;
373             iter = flags.begin();
374             jk = rand() % flags.size();
375             advance(iter, jk);
376             textSetter(*iter, text, font, wrongtextpos2, color);
377             window.draw(text);
378
379
380             color = sf::Color::Magenta;
```

```
381             iter = flags.begin();
382             jk = rand() % flags.size();
383             advance(iter, jk);
384             textSetter(*iter, text, font, wrongtextpos3, color);
385             window.draw(text);
386
387
388
389
390
391
392
393
394
395             iter = flags.begin();
396             prevtextpos = textposXChoice; //stores previous
397             textposXChoice = 0;
398
399
400
401
402         }
403
404         window.display();
405
406         std::string repuesta;
407
408
409         std::cout << "What country do you think this is? \n";
410         getline(std::cin, repuesta);
411
412         std::cout << "\n the correct answer was:  " << correctanswerholder
              << std::endl;
413
414
415         if (repuesta == correctanswerholder)
416         {
417
418             std::cout << "\n YOU WON FIFTEEN POINTS! \n";
419
420             points += 15;
421             std::cout << "\n Your score: " << points << std::endl;
422
423
424
425
426         }
427         else
428         {
```

```
429
430             std::cout << "\n You lost 10 points! \n";
431
432             points -= 10;
433             std::cout << "\n Your score: " << points << std::endl;
434
435
436
437
438         }
439
440
441         std::cout << "\n ____________________________________ \n";
442
443
444         if (points < 0)
445         {
446
447             std::cout << "\n You have ran out of points! Thank's for
                  playing! \n";
448             exit(0);
449
450         }
451
452
453
454
455
456
457
458
459         Sleep(4000);
460
461
462     }
463
464
465
466     return 0;
467
468
469 }
```