

```
1:  /*
2:   Copyright 2015 Fred Martin,
3:   Y. Rykalova, 2020
4:  */
5:
6:  #include <SFML/Graphics.hpp>
7:  #include <SFML/System.hpp>
8:  #include <SFML/Audio.hpp>
9:  #include <SFML/Window.hpp>
10:
11:  #include <math.h>
12:  #include <limits.h>
13:
14:  #include <iostream>
15:  #include <string>
16:  #include <exception>
17:  #include <stdexcept>
18:  #include <vector>
19:
20:  #include "CircularBuffer.hpp"
21:  #include "StringSound.hpp"
22:
23:  using namespace std;
24:
25:  #define CONCERT_A 220.0
26:  #define SAMPLES_PER_SEC 44100
27:
28:  vector<sf::Int16> makeSamples(StringSound gs) {
29:      std::vector<sf::Int16> samples;
30:
31:      gs.pluck();
32:      int duration = 8; // seconds
33:      int i;
34:      for (i= 0; i < SAMPLES_PER_SEC * duration; i++) {
35:          gs.tic();
36:          samples.push_back(gs.sample());
37:      }
38:
39:      return samples;
40:  }
41:
42:  int main() {
43:      sf::RenderWindow window(sf::VideoMode(300, 200), "SFML Plucked String S
ound Lite");
44:      sf::Event event;
45:      double freq;
46:      vector<sf::Int16> samples;
47:      freq = CONCERT_A;
48:      StringSound gs1 = StringSound(freq);
49:      sf::Sound sound1;
50:      sf::SoundBuffer buf1;
51:      samples = makeSamples(gs1);
52:      if (!buf1.loadFromSamples(&samples[0], samples.size(), 2, SAMPLES_PER_S
EC))
53:          throw std::runtime_error("sf::SoundBuffer: failed to load from samp
les.");
54:      sound1.setBuffer(buf1);
55:
56:      freq = CONCERT_A * pow(2, 3.0/12.0);
57:      StringSound gs2 = StringSound(freq);
58:      sf::Sound sound2;
59:      sf::SoundBuffer buf2;
60:      samples = makeSamples(gs2);
61:      if (!buf2.loadFromSamples(&samples[0], samples.size(), 2, SAMPLES_PER_S
EC))
```

```
62:         throw std::runtime_error("sf::SoundBuffer: failed to load from samp
les.");
63:     sound2.setBuffer(buf2);
64:
65:     while (window.isOpen()) {
66:         while (window.pollEvent(event)) {
67:             switch (event.type) {
68:                 case sf::Event::Closed:
69:                     window.close();
70:                     break;
71:
72:                 case sf::Event::KeyPressed:
73:                     switch (event.key.code) {
74:                         case sf::Keyboard::A:
75:                             sound1.play();
76:                             break;
77:                         case sf::Keyboard::C:
78:                             sound2.play();
79:                             break;
80:                         default:
81:                             break;
82:                     }
83:
84:                 default:
85:                     break;
86:             }
87:
88:             window.clear();
89:             window.display();
90:         }
91:     }
92:     return 0;
93: }
94:
```