

```
1: // Copyright 2022 Anson Cheang
2:
3: // This class is used to create the sound used by KSGuitarSim.cpp
4:
5: #include "StringSound.h"
6: #include <time.h>
7: #include <iostream>
8: #include <cmath>
9: #include <vector>
10: #include <iterator>
11: #include <random>
12: #include <chrono> //NOLINT
13:
14: #define SAMPLING_RATE 44100
15:
16: StringSound::StringSound(double frequency) {
17:     if (frequency <= 0) {
18:         throw std::invalid_argument(
19:             "StringSound(double frequency) : cant accept values of 0 or 1
ower");
20:     }
21:     size_t _size = ceil(SAMPLING_RATE/frequency);
22:     _cb = new CircularBuffer(_size);
23:     _time = 0;
24: }
25:
26: StringSound::StringSound(std::vector <sf::Int16> init) {
27:     _cb = new CircularBuffer(init.size());
28:     std::vector<sf::Int16>::iterator p;
29:     p = init.begin();
30:     while (p != init.end()) {
31:         _cb->enqueue(*p);
32:         p++;
33:     }
34:     _time = 0;
35: }
36:
37: StringSound::~StringSound() {
38:     delete _cb;
39:     _cb = nullptr;
40: }
41:
42: void StringSound::pluck() {
43:     if (_cb->size() <= 0) {
44:         while (!_cb->isEmpty()) {
45:             _cb->dequeue();
46:         }
47:     }
48:
49:     unsigned int secret = std::chrono::system_clock::now().time_since_epoch().count(); //NOLINT
50:     std::mt19937 gen(secret);
51:     while (!_cb->isFull()) {
52:         // generate random number
53:         std::uniform_int_distribution<int16_t> dist(-32768, 32767);
54:         _cb->enqueue(dist(gen));
55:     }
56: }
57:
58: void StringSound::tic() {
59:     if (!_cb->isFull()) {
60:         throw std::runtime_error("Tic() : The list isn't full");
61:     }
62:     // size_t size = _cb->size();
63:     sf::Int16 val1 = _cb->dequeue();
```

```
64:     sf::Int16 val2 = _cb->peek();
65:     sf::Int16 new_val = .996 * ((val1 + val2) / 2);
66:
67:     _cb->enqueue(new_val);
68:
69:     _time++;
70: }
71:
72: sf::Int16 StringSound::sample() {
73:     if (_cb->isEmpty()) {
74:         throw std::runtime_error("sample() : _cb is empty");
75:     }
76:     sf::Int16 sample = _cb->peek();
77:
78:     return sample;
79: }
80:
81: int StringSound::time() {
82:     return _time;
83: }
```