```cpp
#include <SFML/Graphics.hpp>
#include <SFML/System.hpp>
#include <SFML/Audio.hpp>
#include <SFML/Window.hpp>

#include <math.h>
#include <limits.h>

#include <iostream>
#include <string>
#include <exception>
#include <stdexcept>
#include <vector>

#include "CircularBuffer.hpp"
#include "StringSound.hpp"

#define CONCERT_A 440.0
#define SAMPLES_PER_SEC 44100
const int keyboard_size = 37;

std::vector<sf::Int16> makeSamples(StringSound gs)
{
    std::vector<sf::Int16> samples;

    gs.pluck();
    int duration = 8; // seconds
    int i;
    for (i = 0; i < SAMPLES_PER_SEC * duration; i++) {
        gs.tic();
        samples.push_back(gs.sample());
    }

    return samples;
}

int main()
{
    sf::RenderWindow window(sf::VideoMode(800, 800), "SFML KSGuitarSim");
    sf::Event event;

    double frequency;
    std::vector<sf::Int16> sample;

    std::vector<std::vector<sf::Int16>> samples(keyboard_size);
    std::vector<sf::SoundBuffer> buffers(keyboard_size);
    std::vector<sf::Sound> sounds(keyboard_size);

    std::string keyboard = "q2we4r5ty7u8i9op-[=zxdcfvgbnjmk,.;/’ ";

    for (int i = 0; i < (signed)keyboard.size(); i++) {
        frequency = CONCERT_A * pow(2, ((i - 24) / 12.0));
        StringSound tmp = StringSound(frequency);

        sample = makeSamples(tmp);
        samples[i] = sample;

        if (!buffers[i].loadFromSamples(&samples[i][0],
                                        samples[i].size(), 2, SAMPLES_PER_SEC)) {
            throw std::runtime_error("sf::SoundBuffer: failed to load from samples.");
        }

        sounds[i].setBuffer(buffers[i]);
    }

    while (window.isOpen()) {
        while (window.pollEvent(event)) {
            if (event.type == sf::Event::TextEntered) {
                if (event.text.unicode < 128) {
                    char key = static_cast<char>(event.text.unicode);

                    for (int i = 0; i < (signed)keyboard.size(); i++) {
                        if (keyboard[i] == key) {
                            std::cout << "Keyboard key is: " << keyboard[i] << "\n";
                            std::cout << "Attempting to play sound...\n";
                            sounds[i].play();
                            break;
                        }
                    }
                }
            }

            if (event.type == sf::Event::Closed) {
                window.close();
            }
        }

        window.clear();
        window.display();
    }
    return 0;
}
```