```cpp
/*Copyright Albara Mehene*/
#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 "RingBuffer.hpp"
#include "GuitarString.hpp"

#define SAMPLES_PER_SEC 44100.0
#define SAMPLE 37

std::vector<sf::Int16> makeSamplesFromString(GuitarString &gs)  {
  std::vector<sf::Int16> samples;
  int duration = 8;
  gs.pluck();
  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(300, 200), "SFML Guitar Hero Lite");
  sf::Event event;

  std::vector < std::vector<sf::Int16> > sample(SAMPLE);
  std::vector <sf::Sound> sound(SAMPLE);
  std::vector <sf::SoundBuffer> buffer(SAMPLE);
  std::string keyboard = ("1234567890qwertyuiopasdfghjklzxcvbnm,");

  // inserts all sounds in the buffer
  for (int i = 0; i < SAMPLE; i++) {
    GuitarString GStemp(440.0 * pow(2, (i - 24)/12.0));
    sample[i] = makeSamplesFromString(GStemp);
    if (!(buffer[i].loadFromSamples(&(sample[i][0]), sample[i].size(), 2 , 44100.0))) {
     throw std::runtime_error(" sf::SoundBuffer: failed to load from sample. ");
    }
    sound[i].setBuffer(buffer[i]);
  }

  while (window.isOpen()) {
    while (window.pollEvent(event)) {
      switch (event.type) {
        case sf::Event::Closed:
          window.close();
          break;
        default:
        if (sf::Event::KeyPressed && event.key.code != -1) {
          int Key = keyboard.find(event.key.code);
          sound[Key].play();
        }
          break;
      }
      window.clear();
      window.display();
    }
  }
  return 0;
}
```