# Документация проекта: Minisynth QT

## 1. Введение

Minisynth QT — это приложение для синтеза звука, разработанное с использованием языка программирования C++ и библиотеки Qt. Программа предоставляет пользователю простой и удобный графический интерфейс для генерации звуков. Основные возможности включают управление частотой, длительностью и формой звуковой волны.

## 2. Цели и задачи

Основные цели проекта:

* - Изучение алгоритмов синтеза звуковой волны.
* - Реализация взаимодействия с графическим интерфейсом Qt.
* - Демонстрация навыков работы с C++ и событийной моделью.

## 3. Основная функциональность

Программа предоставляет следующие функции:

* - Генерация звуковой волны на основе частоты.
* - Регулировка параметров через графический интерфейс.
* - Интерактивный слайдер для изменения частоты звука.
* - Воспроизведение сгенерированного звука.

## 4. Архитектура проекта

Программа состоит из следующих модулей:

1. 1. main.cpp: Точка входа в программу.
2. 2. mainwindow.cpp и mainwindow.h: Логика графического интерфейса.
3. 3. tonesynth.cpp и tonesynth.h: Генерация звуковой волны.
4. 4. mainwindow.ui: Макет графического интерфейса.

## 5. Подробное описание кода

### 5.1. Точка входа: main.cpp

Файл main.cpp инициализирует приложение и отображает главное окно.

#include "mainwindow.h"  
#include <QApplication>  
  
int main(int argc, char \*argv[]) {  
 QApplication app(argc, argv); // Создание объекта приложения  
 MainWindow mainWindow; // Создание главного окна  
 mainWindow.show(); // Отображение главного окна  
 return app.exec(); // Запуск цикла обработки событий  
}

### 5.2. Логика интерфейса: mainwindow.cpp

Файл mainwindow.cpp реализует взаимодействие пользователя с элементами интерфейса.

#include "mainwindow.h"  
#include "ui\_mainwindow.h"  
  
MainWindow::MainWindow(QWidget \*parent)  
 : QMainWindow(parent), ui(new Ui::MainWindow) {  
 ui->setupUi(this); // Установка интерфейса  
 connect(ui->playButton, &QPushButton::clicked, this, &MainWindow::playTone); // Связывание кнопки Play с функцией playTone  
}  
  
MainWindow::~MainWindow() {  
 delete ui; // Удаление интерфейса  
}  
  
void MainWindow::playTone() {  
 int frequency = ui->frequencySlider->value(); // Получение частоты с слайдера  
 ToneSynth synth; // Создание объекта генератора звука  
 synth.generateTone(frequency); // Генерация звуковой волны  
}

### 5.3. Генерация звука: tonesynth.cpp

Класс ToneSynth отвечает за математические вычисления и генерацию звуковой волны.

#include "tonesynth.h"  
#include <cmath>  
#include <vector>  
  
void ToneSynth::generateTone(int frequency) {  
 int sampleRate = 44100; // Частота дискретизации (Гц)  
 double duration = 1.0; // Длительность сигнала (1 секунда)  
 std::vector<double> samples; // Вектор для хранения данных звуковой волны  
  
 for (int i = 0; i < sampleRate \* duration; i++) {  
 double sample = sin(2 \* M\_PI \* frequency \* i / sampleRate); // Генерация синусоиды  
 samples.push\_back(sample);  
 }  
  
 play(samples); // Воспроизведение звукового сигнала  
}  
  
void ToneSynth::play(const std::vector<double>& samples) {  
 // Реализация воспроизведения (на текущий момент не завершена)  
}

## 6. Основные вычисления

Программа основана на математическом вычислении звуковой волны с использованием функции синуса. Основная формула: y(t) = sin(2πft), где:

* - y(t): значение сигнала в момент времени t;
* - f: частота сигнала в Гц;
* - t: текущее время, выраженное в секундах.

Каждая выборка (sample) вычисляется на основе частоты дискретизации (44100 Гц), что позволяет получить гладкую синусоидальную волну.

# 7. Работа графического интерфейса (GUI)

Графический интерфейс программы разработан с использованием библиотеки **Qt** и представляет собой минималистичный синтезатор с функциональными элементами:

1. **Выбор устройства вывода звука:**

• Выпадающий список для выбора устройства воспроизведения (например, динамики или наушники).

2. **Управление громкостью:**

• Ползунок для изменения уровня громкости.

3. **Клавиатура синтезатора:**

• Набор клавиш для игры нот, включая:

• Белые клавиши: C, D, E, F, G, A, B, C’.

• Черные клавиши: C#, D#, F#, G#, A#.

4. **Настройки звука:**

• Поле выбора задержки (в миллисекундах) для настройки отклика.

• Поле выбора октавы для изменения диапазона нот.

8.Заключение  
Проект Minisynth QT продемонстрировал возможности языка C++ и библиотеки Qt для создания интерактивных приложений. Программа обладает базовыми функциями синтеза звука, включая генерацию синусоидальных волн, настройку частоты через интерфейс и воспроизведение звука.

Ключевые достижения команды: структурированная архитектура кода, удобный интерфейс и оптимальная реализация математических вычислений.

Благодаря четкому распределению задач, проект достиг высокой степени завершенности и может быть представлен для обсуждения.