

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О. В. Непомнящий
подпись

« _____ » _____ 2022 г.

БАКАЛАВРСКАЯ РАБОТА

VST-синтезатор

09.03.01 – «Информатика и вычислительная техника»

Руководитель _____ ст. преподаватель К. В. Пушкарев
подпись, дата

Выпускник _____ А. А. Рябошапко
подпись, дата

Нормоконтролер _____ К. В. Пушкарев
подпись, дата

Красноярск 2022

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой

_____ О. В. Непомнящий
подпись
« ____ » _____ 20 ____ г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Красноярск 2021

Студенту Рябошапко Артему Александровичу.

Группа: КИ18-06Б. Направление (специальность): 09.03.01

«Информатика и вычислительная техника».

Тема выпускной квалификационной работы: «VST-синтезатор».

Утверждена приказом по университету № 7914/с от 26.05.2022.

Руководитель ВКР: Пушкарев К. В., старший преподаватель кафедры ВТ ИКИТ СФУ.

Исходные данные для ВКР: нет.

Перечень разделов ВКР:

1. Анализ задания на выпускную квалификационную работу.

2. Проектирование и реализация приложения.

3. Инструкции.

Перечень графического материала: не требуется.

1 Задание

Разработать VST-плагин под операционную систему Windows — цифровой синтезатор звуковых сигналов.

2 Основные требования

Разработанное приложение должно работать с программой Ableton Live под управлением операционной системы Windows, иметь графический интерфейс, открытый исходный код и обладать следующими возможностями:

а) поддержка MIDI-клавиатуры;

б) генерация сигналов различной частоты и формы;

в) фильтрация частот;

г) визуализация звуковых волн;

д) сохранение и загрузка пользовательских настроек, а также загрузка предустановленных настроек (пресетов).

Руководитель ВКР

подпись

К. В. Пушкарев

Задание принял к исполнению

подпись

А. А. Рябошапко

29 ноября 2021 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «VST-синтезатор» содержит 32 страницы текстового документа, 18 иллюстраций, 2 таблицы, 21 использованный источник.

СИНТЕЗАТОР, VST-ПЛАГИН, ABLETON LIVE, JUCE, C++, MIDI, ADSR.

Цель работы: разработка VST-синтезатора для Ableton Live под управлением операционной системы Windows.

Задачи:

- анализ задания на выпускную квалификационную работу;
- проектирование VST-синтезатора;
- реализация VST-синтезатора.

При выполнении данной работы был произведен обзор предметной области, анализ задания на выпускную квалификационную работу, изучены существующие аналоги и сформированы требования, предъявляемые к программе.

В результате анализа задания на выпускную квалификационную работу были рассмотрены существующие решения, также был проведён обзор инструментов. В результате составлена диаграмма классов и прецедентов, создан макет интерфейса.

В итоге был спроектирован и реализован VST-синтезатор. Программа обладает следующими основными возможностями:

- поддержка MIDI-клавиатуры;
- генерация сигналов различной частоты и формы;
- фильтрация частот;
- визуализация звуковых волн;
- сохранение и загрузка пользовательских настроек, а также загрузка предустановленных настроек (пресетов).

СОДЕРЖАНИЕ

Введение	3
1 Анализ задания на выпускную квалификационную работу	5
1.1 Обзор существующих решений	5
1.1.1 Serum	6
1.1.2 Nexus 3.....	8
1.1.3 FabFilter Pro-Q3	9
1.1.4 Lokomotiv	10
1.1.5 Итоги обзора.....	10
1.2 Интерфейс приложения	11
1.3 Диаграмма вариантов использования	12
1.4 Выбор инструментов	14
1.4.1 Выбор библиотеки для разработки	14
1.4.2 Выбор DAW	15
1.4.3 Выбор языка программирования.....	15
1.5 Итог анализа задания	16
2 Проектирование и реализация приложения	17
2.1 Структура приложения.....	17
2.2 Алгоритм работы.....	20
3 Инструкции	25
3.1 Инструкция пользователя	25
3.2 Инструкция разработчика	27
Заключение	29
Список сокращений	30
Список использованных источников	31

ВВЕДЕНИЕ

Музыка всегда была неотъемлемой частью нашей культуры. Конечно же, Прогресс не обошел стороной эту сферу деятельности человека. По мере развития цифровых технологий, музыкантам и композиторам становился доступен все более широкий и удобный набор инструментов для создания музыки.

Термин VST расшифровывается как Virtual Studio Technology. Это технология, позволяющая писать плагины для программ обработки звука. Плагин — это программный модуль, подключаемый к основной программе, который предназначен для расширения ее возможностей. Сейчас существует большое множество плагинов, решающих различные задачи: синтезаторы, эффекты, анализаторы звука, виртуальные инструменты и так далее [1].

VST-плагины работают в DAW – Digital Audio Workstation, это электронное устройство или программа, которая используется для записи и редактирования звуковых файлов. В качестве такой программы в данной работе использована Ableton Live.

VST-плагины сделали процесс создания музыки намного проще и доступнее. В прошлом требовалось оборудование, которое было практически недоступно для большинства начинающих музыкантов из-за стоимости.

Целью выпускной квалификационной работы является разработка VST-синтезатора для Ableton Live [2] под управлением операционной системы Windows. Плагин должен позволить синтезировать собственные звуки и управлять ими.

Программа должна обладать следующими особенностями:

- поддержка MIDI-клавиатуры;
- генерация сигналов различной частоты и формы;
- фильтрация частот;
- визуализация звуковых волн;
- сохранение и загрузка пользовательских настроек, а также загрузка предустановленных настроек (пресетов).

Для достижения цели в работе решаются следующие задачи:

- анализ задания на выпускную квалификационную работу;
- проектирование VST-синтезатора;
- реализация VST-синтезатора.

Далее перечислены основные термины, использованные в выпускной квалификационной работе.

Сэмпл – относительно небольшой оцифрованный звуковой фрагмент.

Волновая таблица – это таблица сэмплов, используемая для синтеза звука путём их комбинирования во времени [3].

Пресет – файл с набором настроек.

Эквализация – управление частотами в аудиосигнале путем их усиления или ослабления.

Варпинг — это синхронизация по темпу различных аудиоматериалов: лупов, сэмплов, треков, мелодических партий и других элементов в аранжировке.

Луп — это фрагмент звуковой записи для циклического воспроизведения.

MIDI-клавиатура — устройство, которое преобразует нажатие клавиш в поток MIDI команд.

MIDI — это стандарт формата обмена данными между электронными музыкальными инструментами [4].

ADSR-огнибающая — функция, которая изменяет параметры звука во времени, используется в синтезаторах [5].

Gate — это сигнал, который пропускает или глушит другие сигналы в зависимости от установленного порогового значения. Gate пропускает сигнал только тогда, когда его уровень выше установленного порога. Используется для контроля уровня звукового сигнала, при редактировании значений ADSR-огнибающей.

1 Анализ задания на выпускную квалификационную работу

В соответствии с заданием, необходимо разработать VST-синтезатор для Ableton Live под управлением операционной системы Windows. VST-синтезатор это плагин для Ableton Live, который представляет собой цифровой синтезатор, который можно использовать для работы со звуком. Чтобы выполнить требования задания, был выполнен анализ аналогов. В ходе анализа не было обнаружено готовых решений с открытым исходным кодом.

1.1 Обзор существующих решений

На сегодняшний день существует множество VST плагинов, которые активно используются в создании современной музыки.

Помимо технологии VST, есть и другие — AU, RTAS, AAX. Технология VST была выбрана из-за её распространенности, она поддерживается большинством DAW и всеми операционными системами [6].

Технология AU (Audio Units) разработана компанией Apple. Она представляет собой набор инструкций, позволяющих операционным системам macOS и iOS генерировать, обрабатывать, воспроизводить и управлять звуковым сигналом с минимальным уровнем задержки. Серьезных отличий между AU и VST нет, но эта технология поддерживается только операционной системой от компании Apple.

RTAS (Real Time AudioSuite), AAX (Avid Audio eXtensions) это технологии, разработанные компанией Avid Technology специально для цифровой рабочей станции Pro Tools. В 2013 году RTAS перестали поддерживать и выпустили его обновленную версию AAX.

VST и AU зависят от ресурсов компьютера, они не могут работать с внешними DSP-процессорами, а технология AAX может. DSP-процессор — это специализированный микропроцессор, предназначенный для обработки оцифрованных сигналов (обычно, в режиме реального времени).

Главный минус AAX — это возможность работы только с одной программой, Pro Tools.

Ниже представлена таблица с основными сведениями о плагинах.

Таблица 1 – Основные сведения о плагинах

Название	Поддержка русского языка	Возможность синтезировать свои звуки	Возможность подключить файлы с набором настроек	Тестовое использование	Регулярные обновления в течение года
Serum	-	+	+	+	-
Nexus 3	-	+	+	+	+
Pro-Q3	-	-	+	+	+
Lokomotiv	-	+	-	+	-

1.1.1 Serum

SERUM [7] – это синтезатор волновых таблиц с двумя осцилляторами, включающий более 450 пресетов и 140 волновых таблиц. Плагин поддерживает интерфейсы VST, AU и AAX, доступные как в 32-битном, так и в 64-битном исполнении.

Данное приложение имеет множество вкладок и окон. Главное окно называется «OSC», в этом окне производятся основные настройки при работе с плагином.

Формы сигналов показаны в программе в виде двухмерной или трехмерной графики. Можно выбрать более 100 волновых таблиц, которые можно редактировать.

Данное приложение подойдет больше профессионалам, потому что без знаний синтеза звука начинающим музыкантам будет сложно создать что-то свое, а использование готовых пресетов плагина не всегда удовлетворяет все запросы пользователя. Главное окно Serum изображено на рисунке 1. Окно осциллятора изображено на рисунке 2.



Рисунок 1 – Главное окно приложения



Рисунок 2 – Окна с осцилляторами

1.1.2 Nexus 3

Nexus [8] – это один из наиболее широко используемых плагинов на рынке. 3-я версия принесла несколько технических и визуальных изменений.

Одна новая особенность — это обновленная навигация по звуковым библиотекам. У пользователя появилась возможность намного быстрее и удобней найти нужную библиотеку.

В Nexus много предварительно загруженных сэмплов, которые вы выбираете для воспроизведения. Их можно редактировать с помощью встроенных настроек плагина, но синтезировать собственный звук, как в Serum, не получится. Главный экран Nexus изображен на рисунке 3.



Рисунок 3 – Главное экран Nexus 3

1.1.3 FabFilter Pro-Q3

FabFilter Pro-Q3 [9] — это эквалайзер, инструмент, который часто используется при создании музыки. Плагин дает возможность пользователю управлять частотами в аудиосигнале в реальном времени, получая высокое качество звука на выходе.

FabFilter Pro-Q3 имеет качественный графический интерфейс и большой набор предустановленных настроек, которые упрощают работу для пользователя. FabFilter Pro-Q3 можно бесплатно получить на 30 дней, также присутствуют регулярные обновления. Главный экран FabFilter Pro-Q3 изображен на рисунке 4.



Рисунок 4 – Главный экран плагина

1.1.4 Lokomotiv

Lokomotiv [10] это VST-плагин, с его помощью пользователь может синтезировать собственные звуки. Технические возможности Lokomotiv значительно слабее, чем у Serum или Nexus, в нем нет таких расширенных настроек для синтеза звука. Зато Lokomotiv можно использовать бесплатно, это хороший вариант для начинающего музыканта. Интерфейс Lokomotiv изображен на рисунке 5.



Рисунок 5 – Главный экран плагина

1.1.5 Итоги обзора

Из обзора существующих решений, представленных в пунктах 1.1.1-1.1.4, можно сделать вывод, что каждое из рассмотренных решений позволяет работать со звуком и обрабатывать его. Во многих решениях присутствует возможность просмотра графиков и диаграмм звуковых волн.

Исходя из анализа VST-плагинов, было выяснено, что Serum имеет сильные технические способности по обработке и синтезу звука, а Nexus

больше приспособлен к работе с заготовленными сэмплами из библиотек, которые идут в комплекте с программой. FabFilter Pro-Q3 — эквалайзер с качественным графическим интерфейсом, имеющий большой набор предустановленных настроек, он был рассмотрен для того, чтобы показать разнообразие плагинов, и их технических возможностей.

В Lokomotiv реализована возможность генерации аудиосигналов, есть фильтр частот и набор предустановленных настроек для синтезатора, и в техническом плане он не перегружен дополнительными возможностями, как другие плагины, обозреваемые в пунктах 1.1.1-1.1.4. Поэтому этот плагин был принят за образец, на который можно опираться при разработке собственного VST-плагины.

1.2 Интерфейс приложения

На основе задания был сделан макет интерфейса, который должен получиться в конечном приложении. На рисунке 6 изображен этот макет.

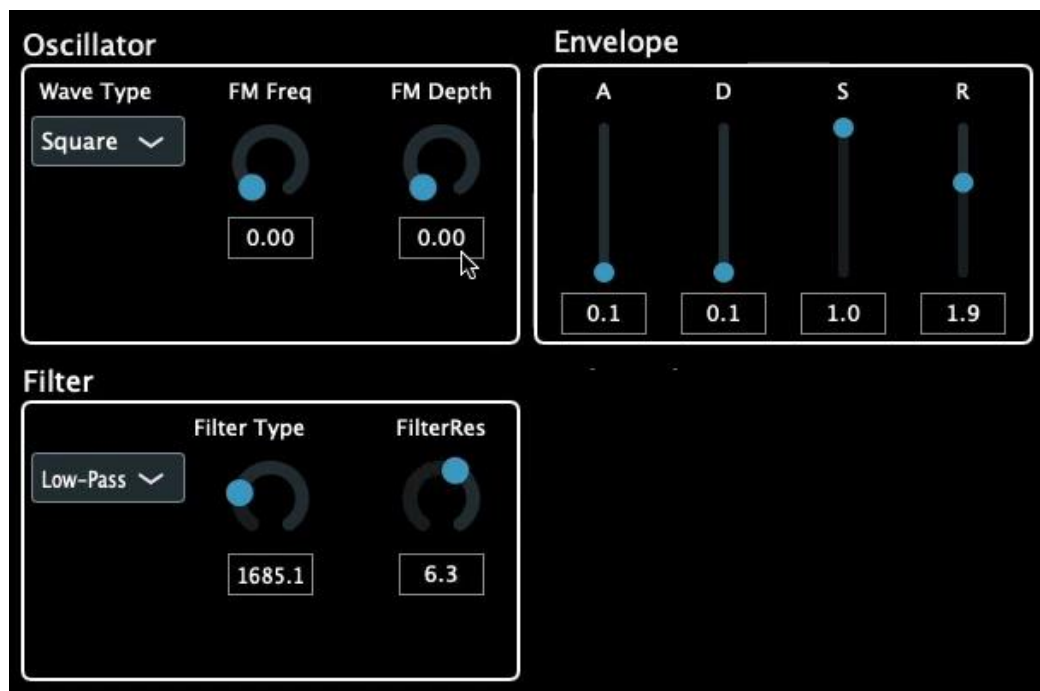


Рисунок 6 – Макет интерфейса

1.3 Диаграмма вариантов использования

На рисунке 7 изображена диаграмма вариантов использования.

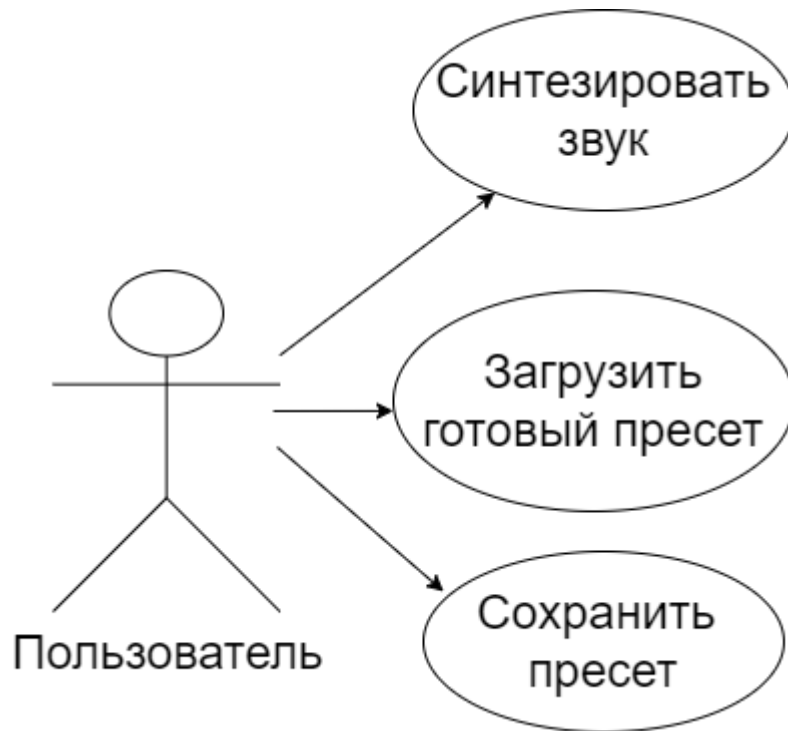


Рисунок 7 – Диаграмма вариантов использования

Название прецедента: «Синтезировать звук».

Цель сценария: Используя технические возможности программы по обработке звука, синтезировать звук, который будет удовлетворять потребности пользователя.

Пользователь, находясь на главном экране, может менять настройки в окнах: Oscillator, Envelope, Filter

В окне Oscillator пользователь может выбрать тип волны. В разделе Wave, когда пользователь выбирает Sine, при нажатии клавиши на MIDI-контроллере генерируется синусоидальная звуковая волна. Если пользователь выбирает тип Saw, при нажатии клавиши на MIDI-контроллере генерируется пилообразная звуковая волна. Если пользователь выбирает тип Square, при нажатии клавиши на MIDI-контроллере генерируется квадратная звуковая волна.

В окне Filter пользователь может выбрать способ обработки звуковой

волны. Когда пользователь выбирает Low-Pass, обрабатываются низкие частоты звуковой волны. Если пользователь выбирает High-Pass, обрабатываются высокие частоты звуковой волны.

В окне ADSR пользователь может отредактировать ADSR-огнивающую. Если пользователь изменяет значения фазы A, меняются начальная фаза Gate-сигнала. При изменении фазы D меняется фаза перехода звука в установившееся состояние. Когда пользователь изменяет значения фазы S, меняется основная фаза Gate-сигнала, которая соответствует продолжительности нажатия клавиши на MIDI-контроллере. Если пользователь изменил значения фазы R, меняется фаза затухания Gate-сигнала.

Название прецедента: «Загрузить готовый пресет».

Цель сценария: Загрузить файл с предустановленными настройками.

Находясь на главном экране, пользователь выбирает раздел загрузки пресета.

После того, как пользователь выбрал нужный пресет, в программу подгружается файл с заготовленными настройками. После этого изменяются настройки в окнах Oscillator, Envelope, Filter в соответствии загруженному пресету.

Название прецедента: «Сохранить пресет».

Цель сценария: Сохранить пресет с настройками пользователя.

Находясь на главном экране, пользователь выбирает раздел сохранения пресета.

После того, как пользователь нажал на иконку сохранения пресета, пользовательские настройки программы сохраняются в файл проекта DAW.

1.4 Выбор инструментов

1.4.1 Выбор библиотеки для разработки

Обычно VST плагины пишутся на языке программирования C++. Преимуществами этого языка являются кроссплатформенность и эффективность.

Чтобы создавать VST плагины, компания Steinberg выпустила VST SDK [11], библиотеку для разработки, написанную на C++. При анализе документации этой библиотеки, ее обучающих материалов и комментариев пользователей, был сделан вывод о том, что при настройке этой библиотеки могут возникнуть трудности. Обучающие материалы представлены в ограниченном количестве и только на английском языке. Это может увеличить сложность реализации задания на ВКР.

WDL-OL [12] — это C++ библиотека для создания кроссплатформенных плагинов. Поддерживаются технологии VST, RTAS, AAX. Удобство библиотеки состоит в том, что (при правильной настройке проекта) пишется один код, а при компиляции получается плагин в разных технологиях.

JUCE [13] — это фреймворк для языка C++ с помощью которого разрабатываются GUI приложения и плагины. JUCE похож на WDL-OL, но он уже имеет в своем составе UI-редактор и множество классов для работы с аудиоданными [14].

В качестве библиотеки для разработки VST плагина был выбран JUCE, так как он имеет большое сообщество пользователей и по нему встречается больше обучающих материалов.

1.4.2 Выбор DAW

Существует множество различных программ для работы со звуком: Logic Pro X, FL Studio, Ableton Live. Это примеры одних из самых популярных программ. Все эти DAW имеют похожий набор технических возможностей, большое количество предустановленных сэмплов и плагинов.

Logic Pro X [15] — программа от компании Apple. В Logic Pro X, как и в DAW описанных ниже, можно работать со звуком, записывать вокал и живые инструменты. Главный недостаток этой программы — работа только на macOS.

FL Studio [16] — одна из самых популярных программ для работы со звуком, разработчиком является компания Image-Line. FL Studio имеет похожий интерфейс как у Logic Pro X и Ableton Live, но не имеет удобной технологии варпинга, которая есть в Ableton Live. Эта технология значительно сокращает время пользователя на обработку сэмпла.

Тестирование плагина по заданию ВКР будет производится в Ableton Live. Плагины которые можно использовать в Logic Pro X, но нельзя в FL Studio, подходят для Ableton Live.

1.4.3 Выбор языка программирования

В пункте 1.4.1 была выбрана библиотека JUCE. Она разработана для языка программирования C++, поэтому задание на ВКР будет реализовано на этом языке программирования.

1.5 Итог анализа задания

Не было найдено ни одного приложения, которое бы удовлетворяло всем условиям задания и имело бы открытый исходный код. В качестве основы для разработки целесообразно обратиться к плагину Lokomotiv.

В качестве инструмента разработки был выбрана библиотека JUCE для языка C++.

Были сформулированы функциональные требования к разрабатываемому приложению в виде диаграммы прецедентов и их текстового описания.

2 Проектирование и реализация приложения

2.1 Структура приложения

При объектно-ориентированном проектировании были выделены основные ключевые классы. На рисунке 8 представлена диаграмма классов.

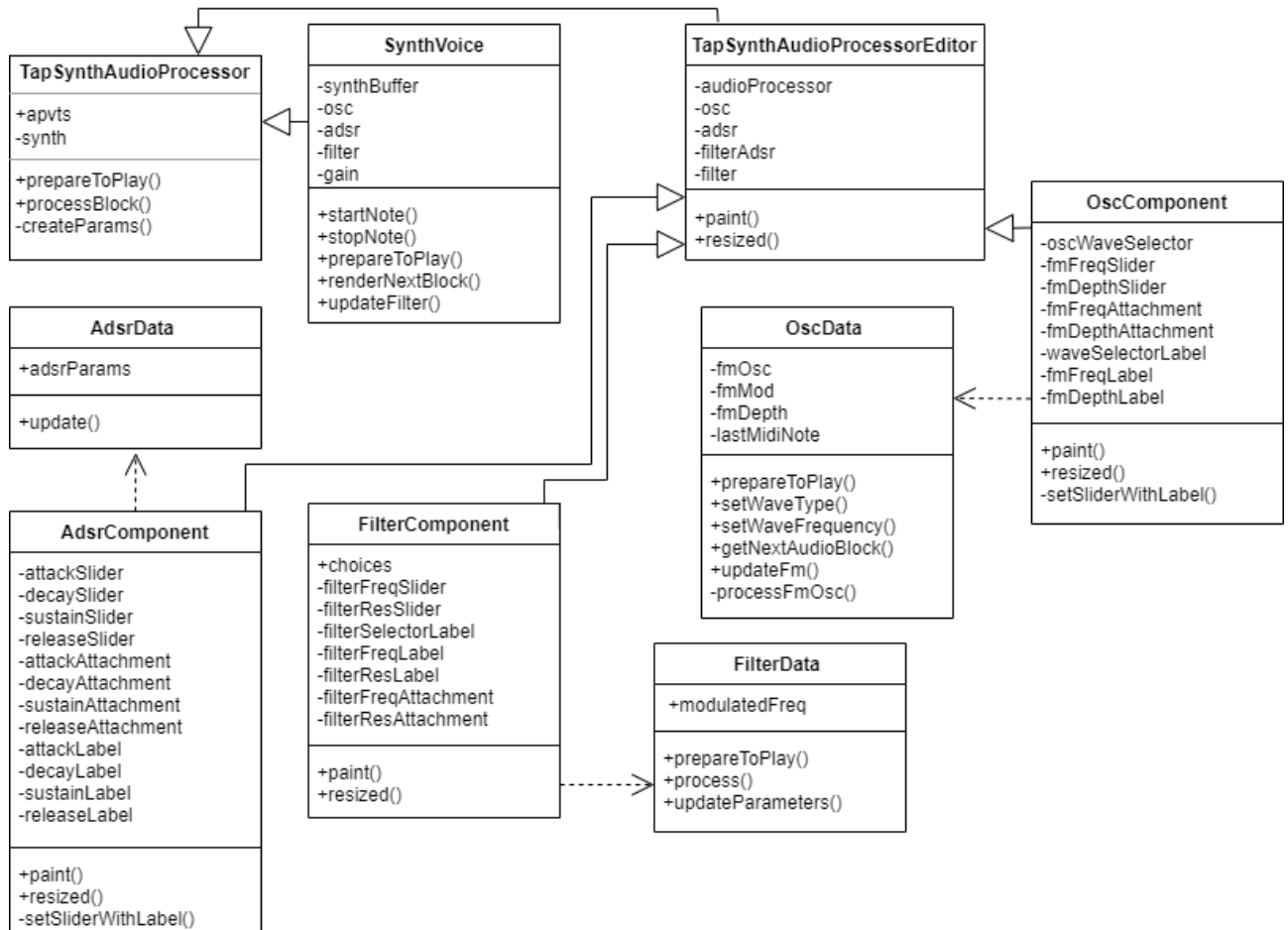


Рисунок 8 – Диаграмма классов

Был реализован класс TapSynthAudioProcessor, базовым классом которого является AudioProcessorEditor библиотеки JUCE. TapSynthAudioProcessor — это главный класс программы, который связывает графический интерфейс с классами обработки данных. В TapSynthAudioProcessor реализован метод prepareToPlay, который вызывается перед началом работы до воспроизведения

звука, чтобы позволить процессору подготовиться и загрузить все необходимые настройки. В процессе взаимодействия с программой, пользователь изменяет параметры VST-плагины, эти изменения нужно считывать и передавать в другие классы, чтобы изменения пользователя отображались. Для этого был реализован метод `processBlock`. Он считывает значение с заданных элементов графического интерфейса и передает текущее состояние классам `AdsrData`, `FilterData`, `OscData`. В методе `createParams` загружаются заранее подготовленные настройки для начала работы с плагином.

Был реализован класс `TapSynthAudioProcessorEditor`, это главный класс для графического интерфейса, в нем задаются основные настройки рабочего окна и добавляются элементы графического интерфейса. `TapSynthAudioProcessorEditor` является производным классом `AudioProcessorEditor` библиотеки JUCE. В `TapSynthAudioProcessorEditor` реализованы методы `paint`, `resized`, которые отвечают за отрисовку графического интерфейса.

Класс `SynthVoice` является производным классом `SynthesiserVoice` библиотеки JUCE. `SynthVoice` отвечает за воспроизведение звуков, в нем реализованы методы `startNote` и `stopNote`, они отвечают за передачу сигнала от MIDI клавиатуры, `startNote` передает данные о частоте от MIDI клавиатуры в осциллятор (генератор звуковой волны) и работает, когда нажата клавиша на MIDI клавиатуре, `stopNote` останавливает передачу данных от MIDI клавиатуры, когда клавиша на MIDI клавиатуре отпускается. Метод `prepareToPlay` вызывается перед началом работы до воспроизведения звука, чтобы позволить процессору подготовиться и загрузить все необходимые настройки. Метод `renderNextBlock` обрабатывает блоки аудио данных. Метод `updateFilter` обновляет параметры фильтра, которые поступают из метода `processBlock` класса `TapSynthAudioProcessor`.

Был реализован класс `AdsrData`. В этом классе находится основная логика работы ADSR-оглибающей, в нем находится метод `update`, который обновляет параметры ADSR-оглибающей, параметры поступают из метода `processBlock` класса `TapSynthAudioProcessor`. Разное состояние параметров ADSR-оглибающей

по-разному влияет на звук.

Класс `FilterData` отвечает за фильтрацию звука. В нем реализован метод `updateParameters`, в котором можно выбрать фильтрацию по низким, высоким частотам, полосовой фильтр. `Low-Pass` – фильтр нижних частот – срезает все частоты, находящиеся выше заданной. `Hi-Pass` – фильтр верхних частот – срезает все частоты, находящиеся ниже заданной. `Band-Pass` – полосовой фильтр – срезает частоты ниже и выше заданных.

Класс `OscData` содержит логику для генерации звуковых волн. В текущем состоянии программного кода в классе генерируются звуковые волны трёх типов: синусоидальная волна, пилообразная волна, прямоугольная волна.

Классы `AdsrComponent`, `FilterComponent`, `OscComponent` содержат методы, отвечающие за отрисовку элементов управления плагином в главном окне графического интерфейса.

2.2 Алгоритм работы

Алгоритм работы состоит в следующем. Сначала пользователь нажимает клавишу на MIDI-клавиатуре. Каждой клавише по умолчанию присвоена своя определенная частота, эта информация передается на осциллятор и, в зависимости от выбранного типа волны, формируется звук с частотой, соответствующей нажатой MIDI-клавише. Осциллятор перестает генерировать сигнал при отпускании клавиши. Этапы обработки сигнала представлены на рисунке 9.

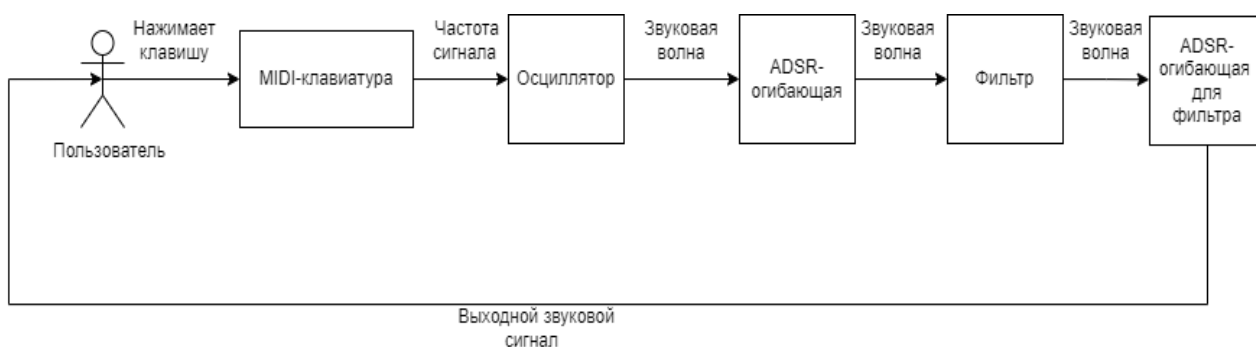


Рисунок 9 – Этапы обработки сигнала

После осциллятора сигнал идет в обработку ADSR-огибающей, которая будет менять его амплитуду. ADSR-огибающая реализована с помощью встроенного класса ADSR в библиотеке JUCE, затем сигнал поступает в фильтр. Для фильтра реализована своя ADSR-огибающая, которая изменяет настройки фильтра по такому же принципу, как и для осциллятора.

Схема работы программы представлена на рисунке 10.

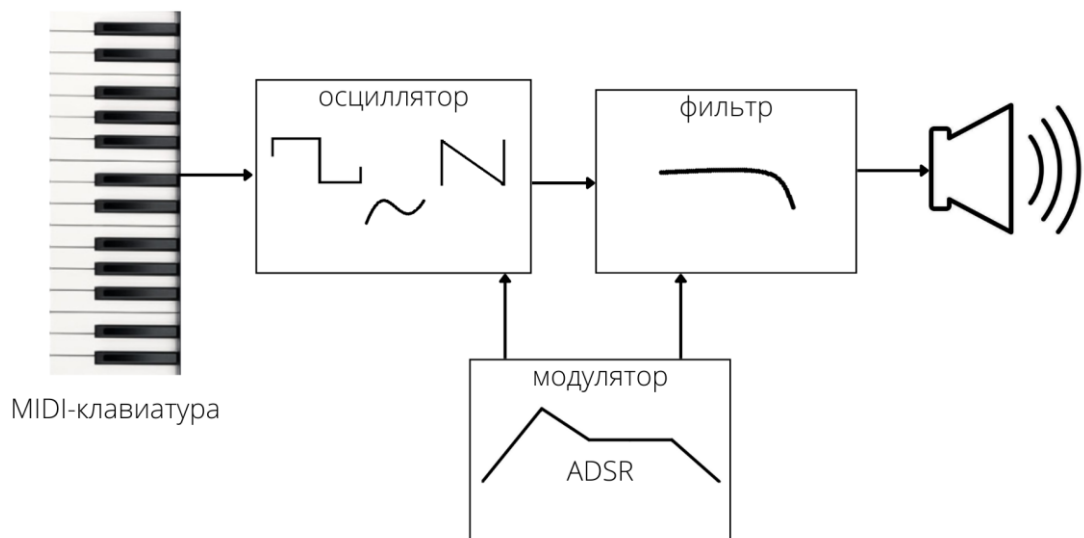


Рисунок 10 – Схема работы

Интерфейс программы представлен на рисунке 11.

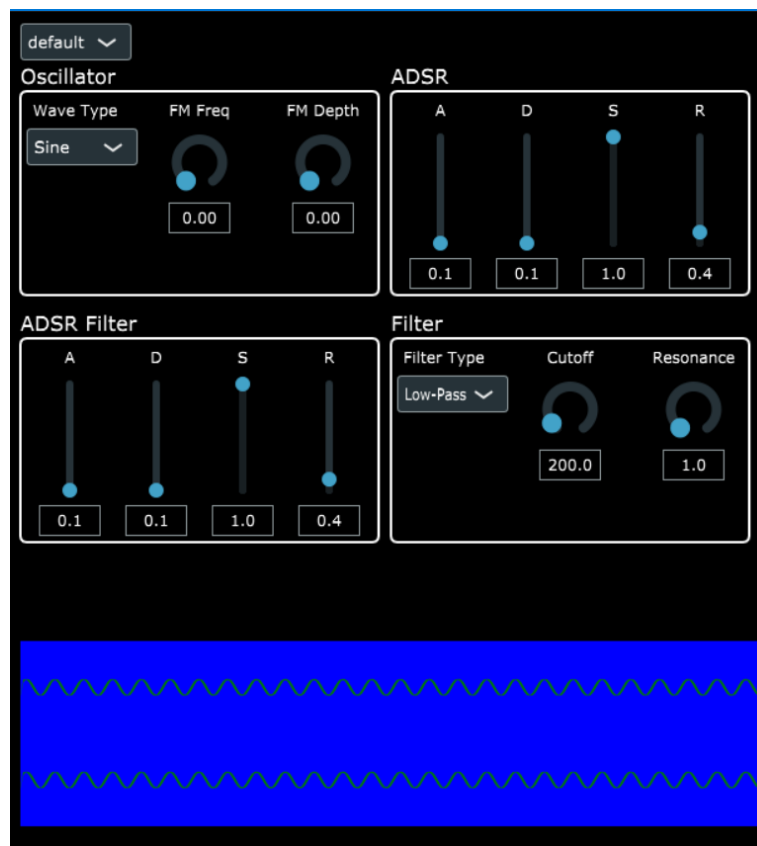


Рисунок 11 – Интерфейс программы

ADSR-огнибающая позволяет описывает изменения громкости звука как 4 последовательные фазы с параметрами:

- Attack — время, за которое уровень громкости сигнала достигла своего максимального уровня [17];
- Decay — время, за которое сигнал спадает до уровня, определенном в фазе Sustain;
- Sustain — фаза сигнала с постоянной заданной амплитудой. Sustain длится пока пользователь держит клавишу MIDI-клавиатуры нажатой;
- Release — фаза определяет время спада амплитуды сигнала до нулевого уровня. Release начинается сразу, после того, как клавиша на MIDI-клавиатуре отпущена.

Если сгенерировать синусоидальный сигнал с применением огнибающей, получим следующую волну на рисунке 12. Видно, что в фазах Attack, Decay и Release огнибающая меняется по линейному закону (аудиофайл генерировался в Ableton, а затем исследовался в открытом редакторе Audacity). График фаз ADSR-огнибающей изображен на рисунке 13.

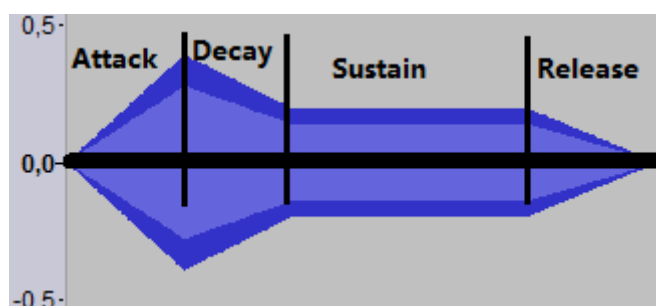


Рисунок 12 – Синусоидальный сигнал с разделением фаз ADSR [18]

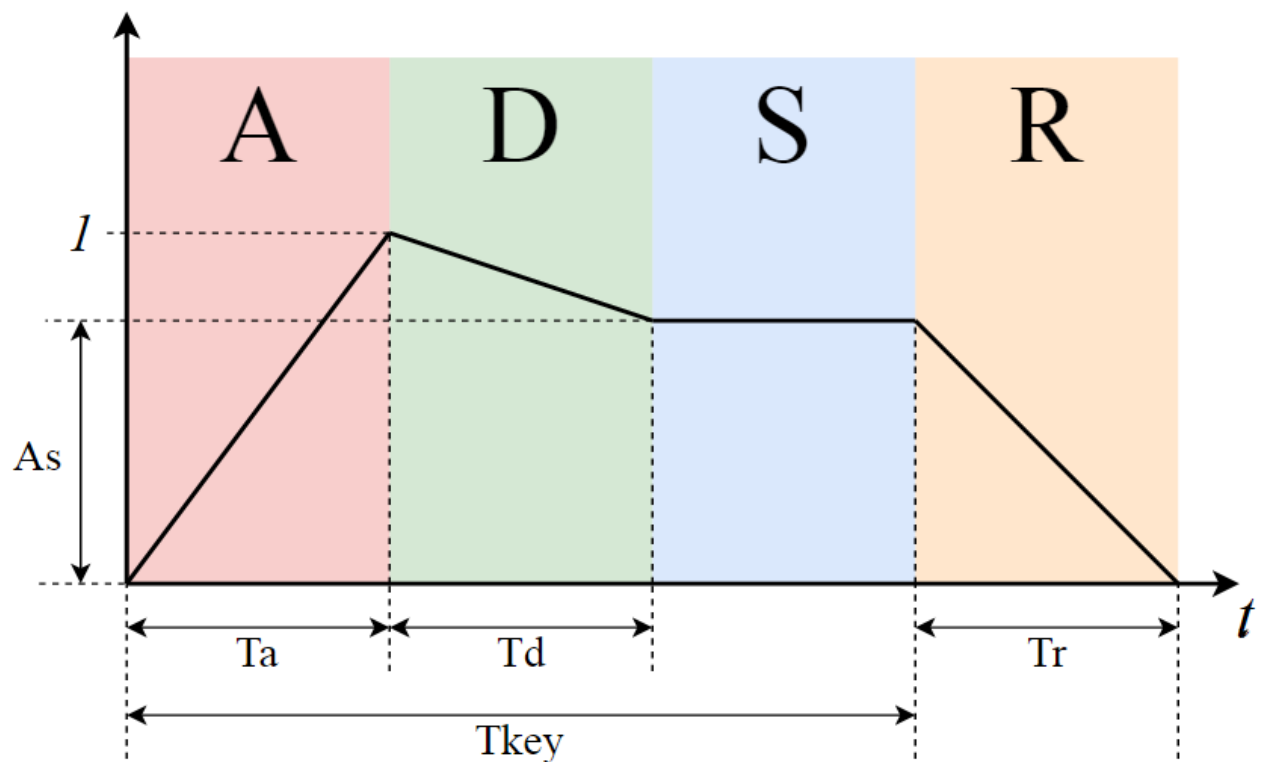


Рисунок 13 – ADSR огибающая [17]

Обозначение параметров на графике ADSR:

- T_a — время Attack;
- T_d — время Decay;
- A_s — значение амплитуды сигнала в Sustain;
- T_r — время Release;
- T_{key} — время удержания клавиши на MIDI-клавиатуре.

Фильтр работает в трёх режимах [19]:

- Low Pass – фильтр нижних частот – срезает частоты, находящиеся выше заданной;
- Hi Pass – фильтр верхних частот – срезает частоты, находящиеся ниже заданной;
- Band Pass – полосовой фильтр – срезает частоты ниже и выше заданных.

Для редактирования характеристик фильтра в программе реализованы Filter cutoff и Resonance. Filter Cutoff – это частота среза, с ее помощью мы можем отрегулировать диапазон срезаемых частот. Resonance создает пик в районе среза, тем самым еще сильнее усиливает частоты, которые нужно выделить. Графики их работы изображены на рисунке 14.

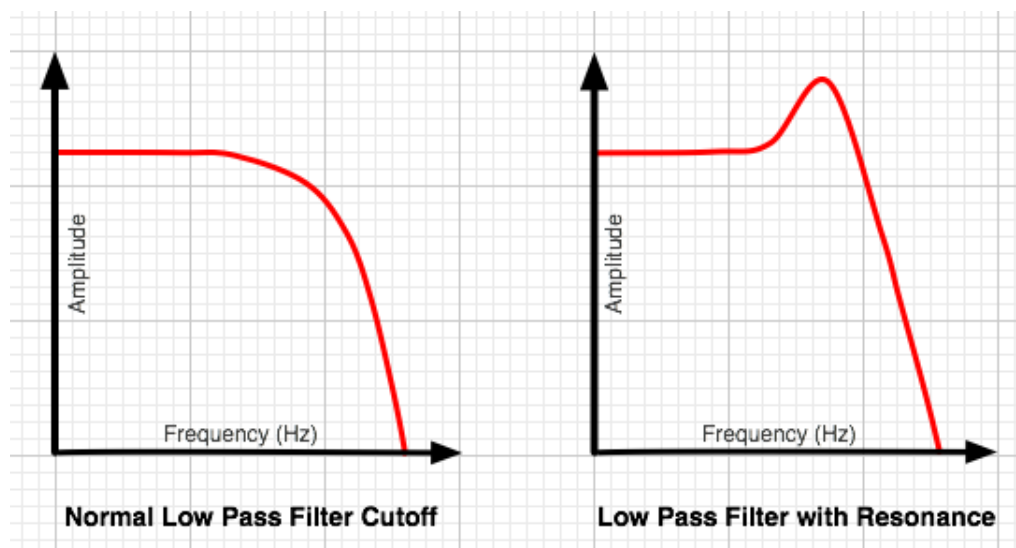


Рисунок 14 – График Filter Cutoff, Resonance [20]

3 Инструкции

3.1 Инструкция пользователя

Для начала работы с плагином необходимо открыть программу Ableton, и выбрать плагин simSynth. Ниже в разделе 3.2 будет описано, как добавить плагин в Ableton.

При открытии плагина откроется окно со всеми элементами интерфейса. В разделе Oscillator, меняя настройки Wave Type, можно выбрать тип синтезируемой волны: синусоидальная (sine), пилообразная (saw), квадратная (square). Для изменения настроек частотной модуляции, используется регулятор FM Freq, для изменения глубины модуляции (изменения амплитуды) используется регулятор FM Depth.

Для изменения настроек модуляции звуковой волны осциллятором, в разделе ADSR изменяются параметры регуляторов. Регулятор «A» изменяет время состояния Attack, регулятор «D» изменяет время состояния Decay, регулятор «S» изменяет время состояния Sustain, регулятор «R» изменяет время состояния Release.

ADSR-модуляция для фильтра может быть настроена аналогичным образом в разделе ADSR Filter.

Для изменения режима работы фильтра в разделе Filter выбирается один из режимов работы в Filter Type: Low Pass, High Pass, Band Pass. Регулятор Cutoff регулирует частоту среза фильтра, регулятор Resonance создает заданный пик в районе среза.

Для сохранения собственных настроек плагина необходимо нажать на иконку изображенную на рисунке ниже (рисунок 15). При закрытии Ableton, последнее состояние плагина сохраняется по умолчанию.

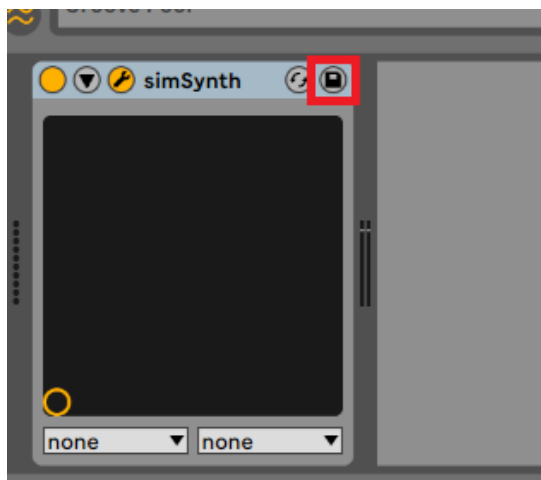


Рисунок 15 – Сохранение настроек плагина

Все сохранённые параметры пользователя сохраняются в файле проекта Ableton (рисунок 16). Для их загрузки необходимо два раза кликнуть левой кнопкой мыши по нужному пресету.

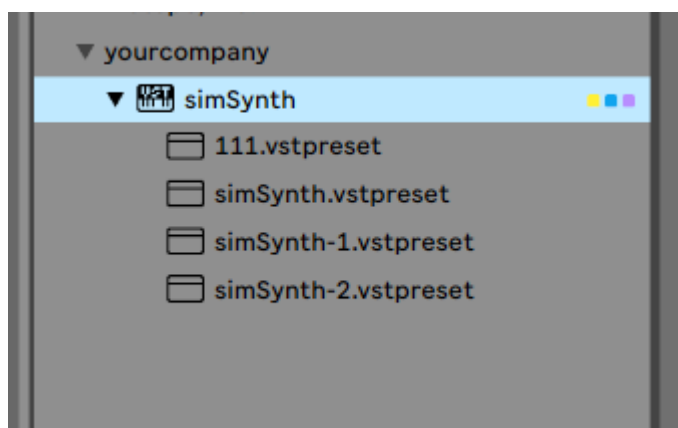


Рисунок 16 – Сохранённые пресеты пользователя

3.2 Инструкция разработчика

Для работы с исходным кодом и его компиляции необходимо:

- IDE Visual Studio 2019;
- фреймворк JUCE версия не ниже 6.1.15;
- Ableton Live 11.

Таблица 2 – Список файлов и их назначение

Список файлов	Назначение
SynthVoice.cpp и SynthVoice.h	Описание и реализация класса SynthVoice
PluginProcesor.cpp и PluginProcesor.h	Описание и реализация класса TapSynthAudioProcessor
PluginEditor.cpp и PluginEditor.h	Описание и реализация класса TapSynthAudioProcessorEditor
AdsrData.cpp и AdsrData.h	Описание и реализация класса AdsrData
FilterData.cpp и FilterData.h	Описание и реализация класса FilterData
OscData.cpp и OscData.h	Описание и реализация класса OscData.
AdsrComponent.cpp и AdsrComponent.h	Описание и реализация класса AdsrComponent
FliterComponent.cpp и FilterComponent.h	Описание и реализация класса FilterComponent
OscComponent.cpp и OscComponent.h	Описание и реализация класса OscComponent

Для открытия плагина в Ableton, в проекте Visual Studio необходимо выполнить сборку `simSynth_VST3` (рисунок 17).

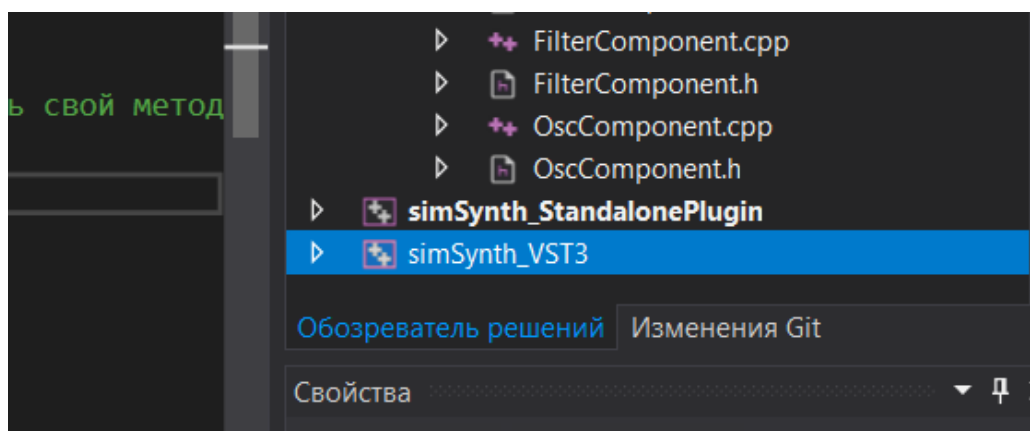


Рисунок 17 – Сборка проекта

После этого в настройках Ableton в разделе Plug-ins необходимо добавить папку в которую собрался проект (рисунок 18).

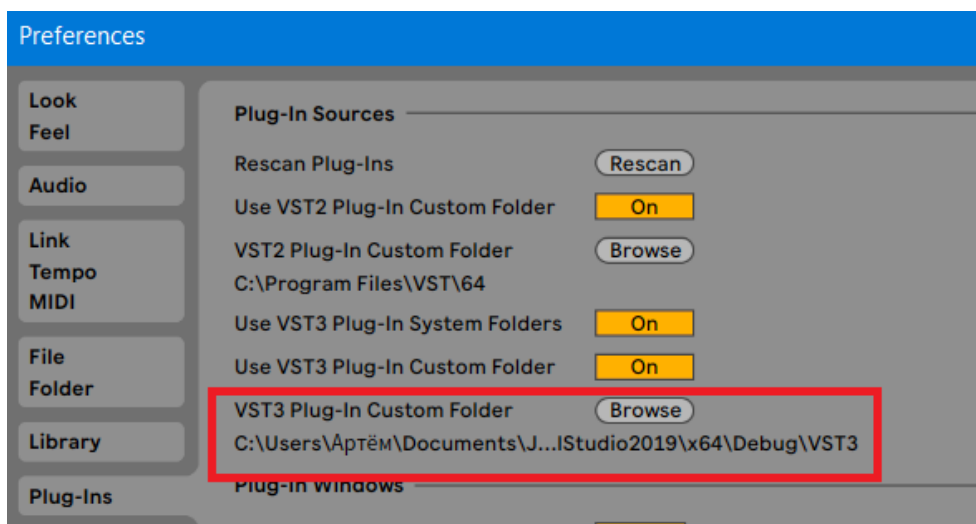


Рисунок 18 – Сборка проекта

ЗАКЛЮЧЕНИЕ

В результате работы был спроектирован и реализован VST-синтезатор.

Был разработан графический интерфейс программы, структура приложения. В плагине реализована возможность генерировать звуковые волны разной частоты и формы и обрабатывать их с помощью фильтра частот. Предусмотрена визуализация звуковой волны. Пользователь может сохранять и загружать сделанные им настройки программы, а также загружать предустановленные настройки (пресеты).

Возможные направления для дальнейшей доработки:

- добавление реверберации звука;
- добавление низкочастотной огибающей LFO (Low Frequency Oscillator).

Последняя версия приложения размещена в git-репозитории [21].

СПИСОК СОКРАЩЕНИЙ

AAX – Avid Audio eXtensions.

AU – Audio Units.

ADSR – Attack, Decay, Sustain, Release (четыре стадии огибающей).

DAW – Digital Audio Workstation (цифровая звуковая рабочая станция).

MIDI – Musical Instrument Digital Interface (цифровой интерфейс музыкальных инструментов).

OSC – Open Sound Control.

RTAS – Real Time AudioSuite.

VST – Virtual Studio Technology (технология виртуальной студии).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. What are vst plugins and how can they boost your creativity [Электронный ресурс]. – Режим доступа: <https://www.tunecore.com/ru/blog/2019/11/what-are-vst-plugins-and-how-can-they-boost-your-creativity.html> (дата обращения: 10.06.2022).
2. Ableton Live [Электронный ресурс]. – Режим доступа: <https://www.ableton.com/> (дата обращения: 10.06.2022).
3. Serum [Электронный ресурс]. – Режим доступа: <http://wikisound.org/Serum> (дата обращения: 10.06.2022).
4. MIDI [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/MIDI> (дата обращения: 10.06.2022).
5. ADSR-оггибающая (Envelope Generator) [Электронный ресурс]. – Режим доступа: <https://digitalmusicacademy.ru/lesson-adsr-envelope> (дата обращения: 10.06.2022).
6. Форматы плагинов для обработки звука: гид для начинающего музыканта [Электронный ресурс]. – Режим доступа: <https://samesound.ru/gear/32-daw-plugin-formats-comparison> (дата обращения: 10.06.2022).
7. Serum Advanced Wavetable Synthesizer [Электронный ресурс]. – Режим доступа: <https://xferrecords.com/products/serum> (дата обращения: 10.06.2022).
8. Nexus [Электронный ресурс]. – Режим доступа: <https://www.kvraudio.com/product/nexus-by-refx> (дата обращения: 10.06.2022).
9. FabFilter Pro-Q3 [Электронный ресурс]. – Режим доступа: <https://www.fabfilter.com/products/pro-q-3-equalizer-plug-in> (дата обращения: 10.06.2022).
10. Lokomotiv [Электронный ресурс]. – Режим доступа: <https://www.kvraudio.com/product/lokomotiv-by-archetype-instruments> (дата обращения: 10.06.2022).

11. 3rd party developers [Электронный ресурс]. – Режим доступа: <https://www.steinberg.net/developers/> (дата обращения: 10.06.2022).
12. WDL-OL [Электронный ресурс]. – Режим доступа: <https://github.com/olilarkin/wdl-ol> (дата обращения: 10.06.2022).
13. JUCE [Электронный ресурс]. – Режим доступа: <https://juce.com/> (дата обращения: 10.06.2022).
14. Программирование&Музыка: понимаем и пишем VSTi синтезатор на C# WPF. Часть 1 [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/311220/> (дата обращения: 10.06.2022)
15. Logic Pro X [Электронный ресурс]. – Режим доступа: <https://www.apple.com/ru/logic-pro/> (дата обращения: 10.06.2022).
16. FL Studio [Электронный ресурс]. – Режим доступа: <https://www.image-line.com/> (дата обращения: 10.06.2022).
17. Программирование&Музыка: ADSR-огибающая сигнала. Часть 2 [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/311750/> (дата обращения: 10.06.2022).
18. Audacity [Электронный ресурс]. – Режим доступа: <https://www.audacityteam.org/> (дата обращения: 10.06.2022).
19. Частотный фильтр в электронной музыке [Электронный ресурс]. – Режим доступа: <https://fierymusic.ru/rabota-so-zvukom/sintez-zvuka/frequency-filters> (дата обращения: 03.11.2021).
20. Trilian фильтры edit [Электронный ресурс]. – Режим доступа: <http://wikisound.org/Trilian> (дата обращения: 10.06.2022).
21. VST-synth [Электронный ресурс]. – Режим доступа: <https://github.com/sd-pg/VST-synth> (дата обращения: 10.06.2022).

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

О. В. Непомнящий

подпись

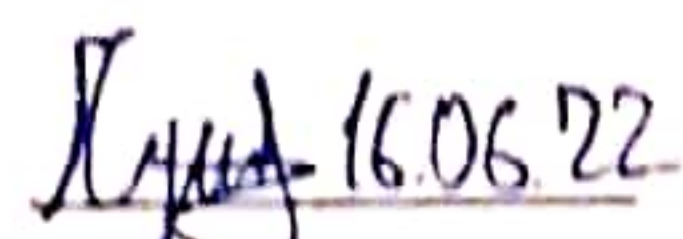
« 20 » 06 2022 г.

БАКАЛАВРСКАЯ РАБОТА

VST-синтезатор

09.03.01 – «Информатика и вычислительная техника»

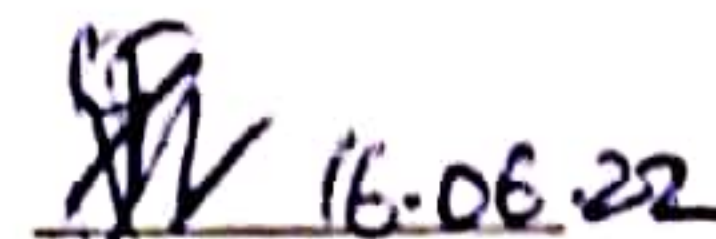
Руководитель


подпись, дата

ст. преподаватель

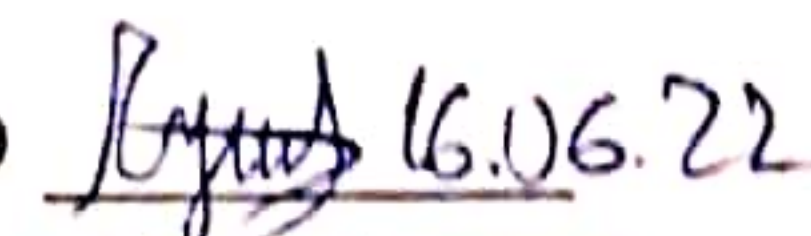
К. В. Пушкарев

Выпускник


подпись, дата

А. А. Рябошапко

Нормоконтролер


подпись, дата

К. В. Пушкарев

Красноярск 2022