


**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

СОГЛАСОВАНО
Приглашенный преподаватель департамента
программной инженерии факультета
компьютерных наук

 Г.М. Сосновский
« 8 » _____ апреля _____ 2023 г.

УТВЕРЖДАЮ
Академический руководитель
образовательной программы «Программная
инженерия»


_____ В. В. Шилов
« ____ » _____ 2023 г.

IOS-ПРИЛОЖЕНИЕ С STEREO AUDIO

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729. 06.02-01 ТЗ 01-1-ЛУ

Исполнитель
 студент группы БПИ207
_____/ М.А.Рощупкина/
« 6 » _____ апреля _____ 2023 г.

Име. № подл.	Подп. и дата	Взам. Име. №	Име. № дубл.	Подп. и дата

УТВЕРЖДЕН
RU.17701729.06.02-01 ТЗ 01-1-ЛУ

IOS-ПРИЛОЖЕНИЕ С STEREO AUDIO

Пояснительная записка

RU.17701729.06.02-01 ТЗ 01-1-ЛУ

Листов 14

<i>Инв. № подл</i>	<i>Подп. и дата</i>	<i>Взам. инв. №</i>	<i>Инв. № дубл.</i>	<i>Подп. и дата</i>

Оглавление

1. ВВЕДЕНИЕ	3
1.1. Наименование программы	3
1.2. Документы, на основании которых ведётся разработка	3
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ	4
2.1. Назначение программы	4
2.1.1 Функциональное назначение программы	4
2.1.2 Эксплуатационное назначение программы	4
2.2. Краткая характеристика области применения	4
3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	5
3.1. Постановка задачи на разработку программы	5
3.2. Описание алгоритма и функционирования программы	5
3.2.1. Общий алгоритм работы программы	5
3.2.2. Алгоритм добавления звука	6
3.2.3. Алгоритм изменения положения звука	6
3.2.4. Алгоритм изменения параметров звука	6
3.2.5. Алгоритм сохранения пространства	7
3.3. Описание и обоснование выбора метода организации входных и выходных данных	7
3.3.1. Описание метода организации входных данных	7
3.3.2. Обоснование метода организация входных данных	7
3.3.3. Описание метода организации выходных данных	7
3.3.4. Обоснование метода организации выходных данных	8
3.4. Описание и обоснование выбора метода выбора технических и программных средств	8
3.4.1. Описание метода выбора технических и программных средств	8
3.4.2. Обоснование метода выбора технических и программных средств	8
4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	9
4.1. Ориентировочная экономическая эффективность	9
4.2. Предполагаемая потребность	9
4.3. Экономические преимущества разработки по сравнению с аналогами	9
ПРИЛОЖЕНИЕ 1	11
ПРИЛОЖЕНИЕ 2	12

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.02-01 81				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы – «iOS-приложение с Stereo Audio».

Наименование программы на английском языке – «Stereo Audio iOS App».

Краткое наименование программы – «iOS-приложение “StereoBeat”».

1.2. Документы, на основании которых ведётся разработка

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем тема курсового проекта.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 508600-01 ТЗ				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Назначение программы

2.1.1 Функциональное назначение программы

iOS-приложение «StereoBeat» помогает создавать стерео-мелодии из заранее заготовленных звуков и их настройки. Приложение подойдет начинающим музыкантам и людям, заинтересованным в быстром создании мелодий или их набросках.

iOS-приложение «StereoBeat» состоит из трех частей: главного экрана, коллекции звуков и библиотеки сохраненных мелодий.

2.1.2 Эксплуатационное назначение программы

iOS-приложение «TSEVA» будет применяться в сфере дизайна и искусства.

Оно позволит людям быстро подбирать сочетания цветов в любой точке мира и делиться этими сочетаниями.

Целевой аудиторией проекта являются дизайнеры, SMM-специалисты и digital художники.

2.2. Краткая характеристика области применения

iOS-приложение «StereoBeat» будет применяться в сфере саунд-дизайна и для создания музыки.

Оно позволит людям быстро создавать наброски своих идей и экспериментировать со стерео-аудио не запуская профессиональное программное обеспечение.

Целевой аудиторией проекта являются начинающие музыканты, битмейкеры и продюсеры.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 508600-01 ТЗ				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на разработку программы

Из функционального назначения программы следует, что разрабатываемая программа должна выполнять следующие функции:

1. Главный экран:

- Возможность перемещать звук по полю
- Возможность изменить параметры звука (тональность, скорость и прочие)
- Возможность сохранить мелодию в коллекцию
- Возможность очистить поле со звуками
- Возможность слышать получившуюся мелодию

2. Коллекция звуков:

- Возможность просмотреть звуки доступные для добавления
- Возможность прослушать сэмпл выбранного звука
- Возможность выбрать звук и добавить его на главный экран

3. Сохраненные мелодии:

- Возможность просмотреть список сохраненных мелодий
- Возможность открыть сохраненную мелодию для редактирования и воспроизведения
- Возможность поделиться сохранённой мелодией в файлы

3.2. Описание алгоритма и функционирования программы

3.2.1. Общий алгоритм работы программы

Система представляет из себя IOS приложение, написанное с помощью Swift. Общая схема работы системы и главные экраны приведены ниже (рис. 1).

Главный экран написан с помощью UIKit, а вспомогательные с использованием более гибкого и современного SwiftUI. Такой подход был выбран из-за того, что экраны коллекции и сохраненных пространств состоят из достаточно стандартных сетки и списка которые быстрее и компактнее реализуются на SwiftUI. В то время как главный экран содержит написанный с нуля компонент поля, логику которого (особенно распознавание жестов) было бы тяжело переложить на новый фреймворк. Существовал вариант сделать

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 508600-01 ТЗ				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

только поле на UIKit и встроить его в экран на SwiftUI, но такой подход не является особо популярным у разработчиков.

В разработке приложения не были использованы никакие сторонние библиотеки, вся логика работы со звуком было реализована с помощью AVFoundation и CoreMotion.

Архитектора приложения в общих чертах напоминает MVVM, с некоторыми отклонениями из-за того, что пришлось связывать между собой два фреймворка. Классическую MVC невозможно было использовать из-за того, что она плохо совместима со SwiftUI, а более сложные архитектуры требуют значительных временных затрат на слишком подробную декомпозицию, которая бессмысленна в таком небольшом проекте.

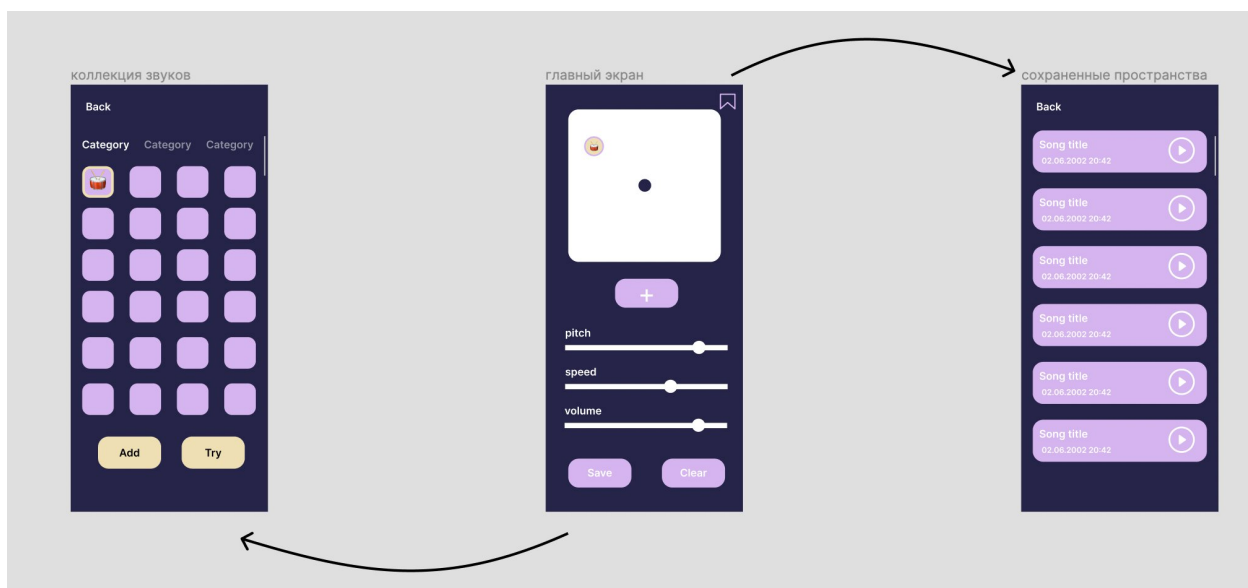


Рисунок 1. Иллюстрация схемы работы приложения.

3.2.2. Алгоритм добавления звука

Пользователь нажимает на кнопку добавления на главном экране и переходит в коллекцию звуков. Там он может прослушать доступные звуки по нажатию на соответствующую кнопку (стоит таймер на 10 секунд если звук слишком долгий) и добавить звук на главный экран. После добавления пользователя возвращает на главный экран и звук появляется на поле в случайном месте.

В настоящий момент вся коллекция звуков хранится на устройстве пользователя в виде статичного словаря со звуками, разбитыми по категориям. В дальнейшем (с увеличением количества звуков) планируется хранить ее в облаке и загружать только нужные пользователю категории.

Сетка коллекции реализована с помощью LazyVGrid. Клеточка звука является написанным отдельно компонентом.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 508600-01 ТЗ				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Звук хранится в приложении в виде модели AudioFileModel, которая содержит сам файл, название и иконку звука. Затем эта модель используется в AudioSource, который уже непосредственно отвечает за всю логику воспроизведения и изменения звука.

3.2.3. Алгоритм изменения положения звука

Пользователь нажимает на звук на поле и перемещает его пальцем на другое место. Звук начинает звучать громче с той стороны к какой он ближе.

Помимо выбранного варианта поля рассматривался компонент, напоминающий циферблат, с уже готовыми позициями для звуков. Такой вариант был отклонен из-за меньшей вариативности.

При изменении положения звука вызывается метод, который вычисляет угол между исходным вектором пространства и вектором, полученным из новой точки звука. Затем этот угол преобразуется в значение в диапазоне от -1 до 1, где -1 означает что звук играет только с левой стороны, а 1 – с правой. Полученное значение передается в плеер.

3.2.4. Алгоритм изменения параметров звука

Пользователь выбирает звук нажатием на него. Он изменяет значения первого слайдера чтобы изменить тональность выбранного звука. Чтобы увеличить или уменьшить скорость воспроизведения пользователь использует второй слайдер. Третий слайдер отвечает за громкость отдельного звука. С его помощью также можно выключить звук, не убирая его с поля.

Для работы с аудио используется AVAudioEngine, которому подключается плеер (AVAudioPlayerNode) и компоненты, отвечающие за скорость (AVAudioUnitVarispeed) и тональность (AVAudioUnitTimePitch). Воспроизведение звука по кругу реализовано с помощью AVAudioPCMBuffer. Для зацикливания аудио было бы проще использовать AVAudioPlayer, у которого есть специальный параметр для этого, но в данном плеере невозможно изменять другие параметры у аудио.

Максимальной скоростью была выбрана x4, а диапазон тональностей от -800 до +800, так как выше или ниже звук слишком искажается. От лейблов к слайдерам решено было отказаться из-за того, что они перегружают интерфейс и легко запоминаются в первые минуты использования приложения.

3.2.5. Алгоритм сохранения пространства

Пользователь нажимает кнопку сохранения на главном экране. После этого массив звуков (с их положениями и параметрами) добавляется в список сохраненных и появляется на соответствующем экране. По нажатию на сохраненное пространство в списке оно открывается на главном экране.

Данные о пространстве и звуках в нем сохраняются локально в UserDefaults, а при запуске приложения выгружаются оттуда.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 508600-01 ТЗ				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3.3. Описание и обоснование выбора метода организации входных и выходных данных

3.3.1. Описание метода организации входных данных

Для работы интерфейса приложение принимает на вход данные, отправляемые с нажатием пользователя на экран.

Для выбора звука – пользователь переходит в коллекцию и выбирает нужный нажатием.

Для смены его расположения пользователь перетаскивает его по полю на главном экране.

Для изменения звука пользователь использует различные слайдеры на главном экране.

Во всех случаях программа обрабатывает нажатия пользователя на экран.

3.3.2. Обоснование метода организация входных данных

Приведённая организация входных данных является одной из наиболее простых и удобных для мобильного приложения, а также полностью отвечает требованиям технического задания.

3.3.3. Описание метода организации выходных данных

На главном экране пользователь слышит получившуюся мелодию и видит расположение звуков на поле.

Экран коллекции звуков показывает пользователю доступные звуки и проигрывает небольшой кусок по нажатию.

На экране сохраненных мелодий пользователю показывается список мелодий, которые были сохранены ранее.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 508600-01 ТЗ				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3.3.4. Обоснование метода организации выходных данных

Приведённая организация выходных данных является одной из наиболее простых и удобных для мобильного приложения, а также полностью отвечает требованиям технического задания.

3.4. Описание и обоснование выбора метода выбора технических и программных средств

3.4.1. Описание метода выбора технических и программных средств

Для бесперебойной работы программного продукта требуется мобильное устройство с:

- операционной системой iOS не ниже версии 15.0
- объемом свободной памяти не менее 400 МБ,

3.4.2. Обоснование метода выбора технических и программных средств

Операционная система iOS версии 15.0 и выше установлена на 96% мобильных устройств с данной операционной системой. В связи с этим кажется нерациональным поддерживать более старые версии. [10]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 508600-01 ТЗ				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Ориентировочная экономическая эффективность

Приложение достигает экономической эффективности за счет предоставления большей библиотеки звуков по ежемесячной подписке.

4.2. Предполагаемая потребность

Данный программный продукт будет интересен начинающим музыкантам, битмейкерам и продюсерам, желающим сделать быстрый набросок мелодии.

4.3. Экономические преимущества разработки по сравнению с аналогами

Приложение ориентировано на мобильность и удобность, поэтому веб-аналоги рассматриваться не будут. Также не будут рассмотрены полностью платные приложения из-за того, что они очевидно уступают в доступности. Выбраны 5 основных конкурентов, которые примерно отражают ситуацию с подобными приложениями на рынке:

Приложение	Drum Pad Machine	Toca Band	Figure	Boom	Sleep Orbit
Звуки	много, но со встроенными покупками	мало	много	готовые песни	фокус на звуках природы
Цена	бесплатно	бесплатно	бесплатно	бесплатно	бесплатно
Стерео Аудио	нет	нет	нет	есть	есть, но звуки просто ездят по кругу
Дизайн	современный	детский	современный	современный	устаревший

Становится очевидно, что бесплатного приложения, сочетающего весь заявленный функционал на рынке в данный момент нет.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 508600-01 ТЗ				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. ГОСТ 19.101-77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. Share of Apple devices by iOS version worldwide from 2016 to 2023 [Электронный ресурс] / Statista. Режим доступа <https://www.statista.com/statistics/565270/apple-devices-ios-version-share-worldwide/>, свободный. (дата обращения: 04.04.2023)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729. 508600-01 ТЗ				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 1

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ

Таблица 1.1 – Описание классов в группе Sliders

Класс/файл	Назначение
PitchSlider	Класс для слайдера тональности
SpeedSlider	Класс для слайдера скорости

Таблица 1.2 – Описание классов в группе Extensions

Класс/файл	Назначение
UIImageExtension	Расширение для работы картинками
Colors	Расширение для работы с цветами

Таблица 1.3 – Описание классов в группе AudioSpace

Класс/файл	Назначение
AudioNodeView	Класс компонента звука на поле
AudioSpaceView	Класс компонента поля
Vector2D	Набор методов для работы с векторами
AudioSource	Класс с логикой звука
AudioSpace	Класс с логикой поля
AudioFileModel	Класс модели файла звука

Таблица 1.4 – Описание классов в группе Screens

Класс/файл	Назначение
MainViewController	Класс главного экрана приложения
SavedView	Класс экрана со списком сохраненных пространств
CollectionView	Класс экрана коллекции доступных звуков
CollectionViewCell	Класс ячейки звука на экране коллекции

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.02-01 81				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 2

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ МЕТОДОВ

Таблица 2.1– Описание методов класса MainViewController

Имя	Модификатор доступа	Тип	Аргументы	Назначение
refreshSliders	public	Void	-	Обновляет значения слайдеров в соответствии с значениями выбранного звука
viewDidLoad	private	Void	-	Загрузка и расположение всех элементов экрана
pitchSliderMoved speedSliderMoved volumeSliderMoved	private	Void	-	Изменяет значение выбранного параметра у звука
processData	private	Void	_ data: CMDevice Motion	Обрабатывает звуки на поле
setupAudioSpace setupAddButton setupSliderStack setupButtonStack	private	Void	-	Отвечает за расположение элементов на экране
saveSpace	private	Void	-	Сохраняет пространство
clearSpace	private	Void	-	Очищает пространство
openCollection	private	Void	-	Переход на экран коллекции звуков
openSaved	private	Void	-	Переход на экран сохраненных пространств

Таблица 2.2 – Описание методов класса SavedView

Имя	Модификатор доступа	Тип	Аргументы	Назначение
openSpace	private	Void	-	Открывает пространство на главном экране

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.02-01 81				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 2.3 – Описание методов класса CollectionView

Имя	Модификатор доступа	Тип	Аргументы	Назначение
headerOnTap	private	Void	-	Переключает на следующий раздел коллекции
playSound	private	Void	-	Проигрывает выбранный звук
addSound	private	Void	-	Добавляет звук на поле

Таблица 2.4 – Описание методов класса AudioSource

Имя	Модификатор доступа	Тип	Аргументы	Назначение
runAudio	public	Void	-	Запускает аудио
stopAudio	public	Void	-	Останавливает аудио
applyFrom	public	Void	viewPoint: CGPoint, insideSize: CGSize	Высчитывает новую позицию звука
updateVolume	public	Void	-	Обновляет громкость звука
updateAudioResult	public	Void	-	Обновляет позицию звука
normalizeAudioResultBy Volume	private	CGFloat	result: CGFloat	Обновляет громкость в соответствии с общим уровнем громкости

Таблица 2.5 – Описание методов класса AudioSpace

Имя	Модификатор доступа	Тип	Аргументы	Назначение
addSource	public	Void	audioSource: AudioSource	Добавляет звук на поле
removeSource	public	Void	audioSource: AudioSource	Удаляет звук с поля
clearSources	public	Void	-	Удаляет все звуки

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.02-01 81				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

[illegible]