

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

« \_\_\_\_\_ » \_\_\_\_\_ 2023г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.01 Информатика и вычислительная техника

**String Art: android-приложение с голосовым помощником**

Руководитель

\_\_\_\_\_

подпись, дата

доцент кафедры ВТ, канд. тех. наук

\_\_\_\_\_

должность, ученая степень

Л.И. Покидышева

Выпускник

\_\_\_\_\_

подпись, дата

Е.В. Федченко

Красноярск, 2023

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

« \_\_\_\_\_ » \_\_\_\_\_

2022г.

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

Красноярск 2022

Студенту Федченко Егору Витальевичу  
фамилия, имя, отчество

Группа КИ19-06Б Направление (специальность) 090301  
номер код

Информатика и вычислительная техника  
наименование

Тема выпускной квалификационной работы: String Art:  
android-приложение с голосовым помощником

Утверждена приказом по университету № 4765/С от 23.03.2023

Руководитель ВКР: Л.И. Покидышева, доцент кафедры ВТ ИКИТ СФУ,  
инициалы, фамилия, учёная степень, должность, место работы

канд. тех. наук

Исходные данные для ВКР:

Задание на ВКР.

Перечень разделов ВКР:

1) Анализ предметной области;

2) Архитектура и основные технические решения;

3) Реализация приложения.

Перечень графического материала: Презентация в формате

MicrosoftPowerPoint.

Руководитель ВКР Л.И. Покидышева  
подпись инициалы, фамилия

Задание принял к исполнению Е.В. Федченко  
подпись инициалы, фамилия

«16» 11 2022 г.  
Дата

## РЕФЕРАТ

Выпускная квалификационная работа по теме «String Art: android-приложение с голосовым помощником» содержит 41 страницу текстового документа, 12 рисунков, 8 использованных источников, 2 приложения.

STRING ART, ГЕНЕРАТОР, ГОЛОСОВОЙ ПОМОЩНИК, ГВОЗДИК, ИНСТРУКЦИЯ, ЛИНИЯ, ИЗОБРАЖЕНИЕ

Объект исследования – String Art приложение.

Цель работы: разработать android-приложение для помощи в создании String Art картин. Были поставлены следующие задачи:

- провести анализ исследуемой предметной области;
- провести проектирование android-приложения: продумать дизайн страниц, построить диаграммы вариантов использования, диаграммы последовательности, общую структуру;
- разработать алгоритм обработки изображений в String Art;
- реализовать спроектированное android-приложение.

В результате работы над ВКР было разработано и реализовано android-приложение для создания String Art изображений.

## СОДЕРЖАНИЕ

Введение .....	4
1 Анализ предметной области .....	6
1.1 Постановка задачи .....	6
1.2 Алгоритм построения изображения .....	7
1.3 Обзор аналогов .....	8
1.3.1 String Art Generator .....	8
1.3.2 StringFace .....	10
1.3.3 ThreadArt .....	11
1.4 Формулировка требований к продукту .....	11
1.5 Голосовой помощник .....	12
1.6 Вывод по первой главе .....	13
2 Архитектура и основные технические решения .....	14
2.1 Анализ технологий .....	14
2.1.1 Python Kivy .....	15
2.1.2 Python Pillow .....	15
2.1.3 Python SoundDevice .....	16
2.2 Разработка интерфейса системы .....	17
2.3 Разработка UML модели проекта .....	18
2.3.1 Диаграмма вариантов использования .....	19
2.3.2 Диаграмма последовательности .....	22
2.3.3 Диаграмма классов .....	23
2.4 Модуль «Преобразователь изображений String Art» .....	25
2.5 Модуль «Голосовой помощник» .....	26
2.6 Вывод по второй главе .....	27
3 Реализация приложения .....	29
3.1 Начальная страница .....	29
3.1.1 Функциональность кнопки «Загрузить изображение» .....	30
3.2 Страница просмотра сгенерированного изображения .....	31

3.3 Страница просмотра инструкции.....	33
3.4 Разработка модуля «Преобразователь изображений String Art».....	35
3.4.1 Алгоритм работы модуля.....	36
3.5 Разработка голосового помощника.....	36
3.6 Вывод по третьей главе.....	37
Заключение.....	38
Список использованных источников.....	39
ПРИЛОЖЕНИЕ А Листинг кода функций draw_line() и draw_poins().....	40
ПРИЛОЖЕНИЕ Б Листинг функции text2int().....	41

## ВВЕДЕНИЕ

Развитие информационных технологий оказывает влияние на все аспекты жизни человека, в том числе и на art-искусство. Информационные технологии приносят новые методы получения людьми изучаемого хобби, а также методы реализации его задумки. На сегодняшний день снова становится популярной ручная работа, как в коммерческих целях, так и для личного пользования. В связи с этим набирают популярность как приложения и сайты, которые могут подать идею для творчества, так и те, которые могут пошагово объяснить, как это делается. Существуют мастер-классы в offline и online режимах, где люди могут попробовать что-то новое, а также могут открыть в себе талант.

String art – вид искусства создания изображений путём специального переплетения нитей. Обычно на каком-нибудь материале, например, деревянной дощечке, вбиваются пины, гвозди или закручиваются винты, за которые цепляется нить и, последовательно цепляясь за них, создаётся какой-нибудь рисунок [1].

На сегодняшний день не существует бесплатных приложений и сайтов для создания алгоритмов, по которому можно сделать полноценную string art картину с фото или каким-либо рисунком. Таким занимается только одна компания, но для получения доступа нужно купить набор с уже обработанной дощечкой и другими материалами для создания такой картины, а так же её диаметр и количество точек(гвоздиков) нельзя менять. Поэтому возникла идея создать бесплатное приложение, скачав которое, человек может с нуля создать своё панно.

Цель создания String Art приложения состоит в предоставлении пользователю удобного и интуитивно понятного инструмента для создания красивых и уникальных произведений искусства из нитей и гвоздей. Это приложение ставит своей задачей вдохновить и поддержать творческий процесс, а также сделать его доступным для широкой аудитории.

Такой модуль может стать полезным инструментом для художников, дизайнеров и любителей искусства, которые хотят создавать интересные и красивые изображения с помощью техники String Art.

Цель работы – разработать android-приложение для помощи в создании String Art картин (рукоделие).

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ исследуемой предметной области;
- провести проектирование android-приложения: продумать дизайн страниц, построить диаграммы вариантов использования, диаграммы последовательности, общую структуру;
- разработать алгоритм обработки изображений в String Art;
- реализовать спроектированное android-приложение.

Приложение будет актуальным и интересным для всех, кто ищет творческую самореализацию, расслабление и возможность поделиться своими работами с другими любителями ручной работы.



## **1 Анализ предметной области**

### **1.1 Постановка задачи**

Изонить (String art) – это вид рукоделия, в котором картины создаются с помощью обычных гвоздей и ниток. В буквальном переводе с английского название означает струнное искусство. Ведь действительно, если присмотреться, нитки реально напоминают натянутые струны. С помощью этой техники можно получить как небольшой и простой, так и сложный рисунок [2].

Подобные картины зачастую размещаются в доме для создания дополнительного уюта. Основным плюсом такой техники является простота реализации. Требуется только быть довольно внимательным, чтобы зацепить нить именно за нужный очередной гвоздь или винт. Создавать картины в стиле String Art можно как вручную, так и автоматически - сегодня существуют даже целые машины, которые самостоятельно подготавливают основание и выполняют натягивание нитей, однако и в том и другом случае необходимо иметь последовательность гвоздей, требующуюся для конкретной картины.

В настоящее время картины из гвоздей и ниток создают с помощью таких техник:

– угол. В этом случае нитка движется таким образом, чтобы между вертикальной и горизонтальной линией был прямой угол;

– окружность. Техника заключается в соединении гвоздей расположенных с противоположной стороны. Разделяют визуально круг на две части, в одной – движение проводят налево, а в другой – направо;

– дуга. Заполняется таким же образом, как и окружность. Начинают с края, потом движутся под прямым углом, а дальше продолжают смещение в сторону [2].

Для успешного рисования в стиле String art существует несколько настроек:

- количество гвоздей – чем больше гвоздей, чем точнее и качественнее может получиться рисунок;

- количество линий – аналогично количеству гвоздей, влияет на детализацию получаемого изображения;

- непрозрачность линий – позволяет воссоздавать оттенки за счёт наложения нескольких линий с прозрачностью.

## **1.2 Алгоритм построения изображения**

Выбирается гвоздь, с которого будет начат процесс переплетения, после чего необходимо выбрать гвоздь, в который провести линию.

Для этого среди всех возможных линий подсчитывается средняя яркость пикселей исходного изображения и выбирается линия, имеющая наименьшую яркость.

Проведённая линия «удаляется» из исходного изображения путём прибавления к значениям пикселей этой линии значения непрозрачности (если непрозрачность равна 100%, то за одну линию все соответствующие пиксели исходного изображения станут белыми, а получаемая картинка очень быстро превратится в сплошную залитую фигуру). Таким образом, использование прозрачности позволяет создавать оттенки.

Далее начальным гвоздём становится тот, который находится на конце добавленной линии, и процесс начинается сначала. Нити натягиваются между гвоздями до тех пор, пока они не сольются в различимое изображение.

На Рисунок 1 показан процесс генерации String Art изображения.

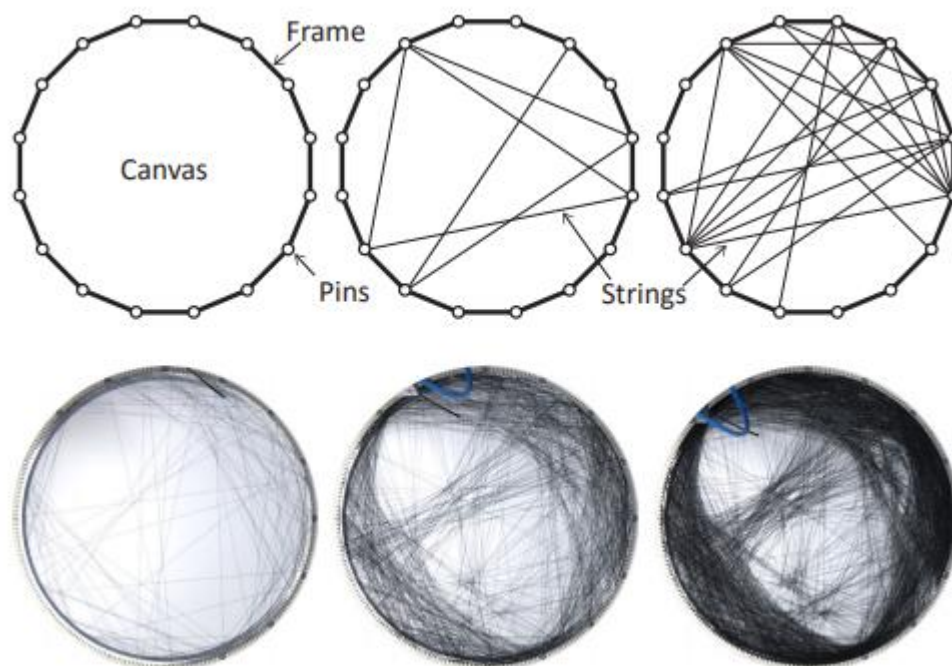


Рисунок 1– Генерация String Art изображения

### 1.3 Обзор аналогов

В сети, как уже упоминалось, достаточно скудно представлены web приложения, мобильные приложения на тему генератора string art изображений, поэтому будут изучаться аналоги, связанные с этой тематикой. Необходимо выявить, какой функционал в них реализован, и чем они отличаются друг от друга, чтобы создать наиболее удобное и эффективное android приложение.

#### 1.3.1 String Art Generator

Web-версия генератора String Art картинок. Превращает произвольные изображения в изображения в стиле string art. При этом результат работы можно сохранить в виде png и svg изображений, а также в специальном String Art формате, который можно открыть на этом же сайте [3].

Перед созданием можно ввести количество гвоздей, проводимых линий и непрозрачность линий.

Помимо самого создания генератор позволяет изменить некоторые параметры входного изображения для создания наиболее качественного рисунка:

- яркость;
- контрастность;
- инверсия яркости.

Программа сама генерирует последовательность гвоздей, через которые следует натягивать нить.

Таким образом, можно по предложенным данным и примеру итоговой картинки собрать такое изображение на полотне. Все упирается только лишь во внимательность, чтобы не пропустить, что написано в инструкции, потому что она очень монотонная, цифры расположены одна за другой, а их около 4000. И это самый главный минус.

Интерфейс и уже готовое изображение с инструкцией представлены на Рисунок 2.

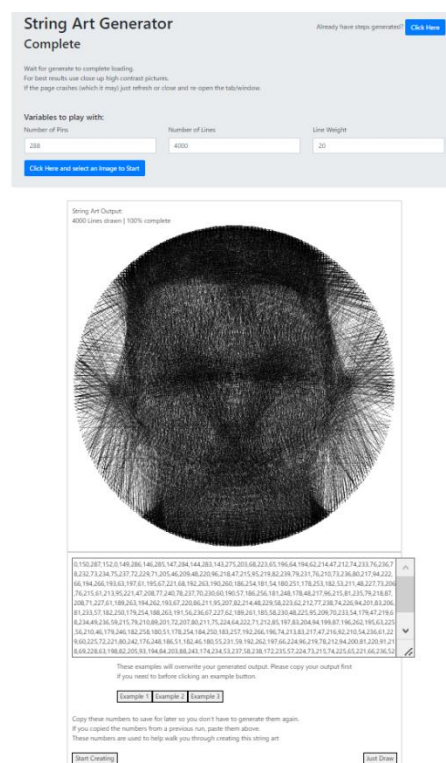


Рисунок 2 – String Art Generator

### 1.3.2 StringFace

Это уже приложение на платформе android. Также выполняет задачу преобразования изображения в картинку нитей. Можно отметить, что у программы множество возможностей:

- есть возможность выбрать форму итоговой картинки. Это может быть не только круг, но и самолет, бабочка, голова быка, ребенок;
- можно сделать фотографию и тут же её преобразовать;
- ввести параметры преобразования (количество гвоздей, линий, непрозрачность линий).

Интерфейс системы представлен на Рисунок 3.

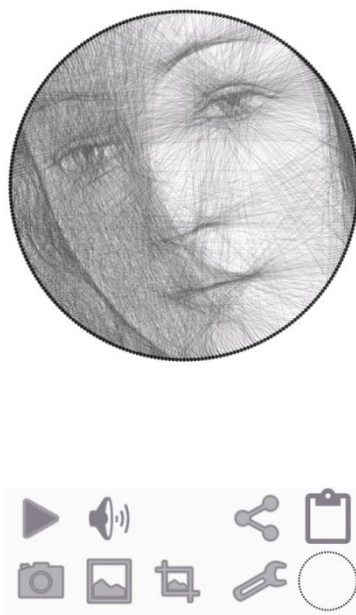


Рисунок 3 – StringFace

Приложение позиционирует себя как бесплатное, хотя за вывод инструкции платный. Возможность пошагового разбора работает некорректно, возможно это из-за того, что это не платная версия.

### 1.3.3 ThreadArt

Приложение работает на платформе android. Есть возможность выбрать контур изображения в виду квадрата и круга. Можно изменять количество гвоздиков и линий. Проект можно сохранить, им можно поделиться. Рабочее пространство увеличивается для детального осмотра (Можно разглядеть номера гвоздиков, где переплетаются линии). Интерфейс системы представлен на Рисунок 4.

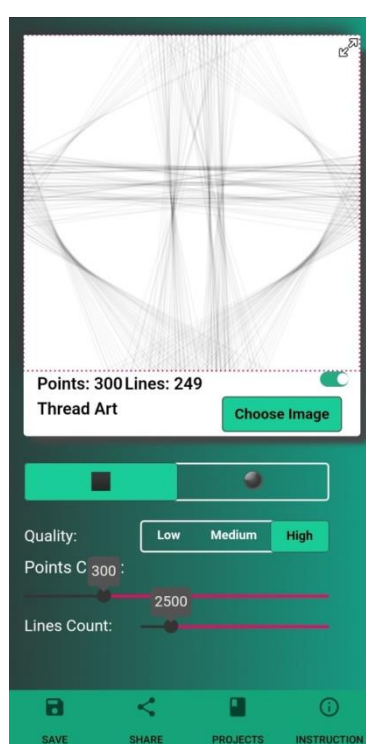


Рисунок 4 – ThreadArt

### 1.4 Формулировка требований к продукту

Проанализировав сервисы по преобразованию изображения в String art, был выявлен ряд несовершенств, таких как отсутствие удобной, читаемой инструкции, голосового помощника, удобного интерфейса.

В реализации приложения по преобразовании изображения в String art для создания картины своими руками было выявлено несколько требований:

– необходимо создать преобразователь изображения в String art с удобным и понятным интерфейсом;

– должна составляться инструкция для пользователя, чтобы на её основании он мог создать свою желаемую картину. Инструкция должна иметь печатный вид, также должна быть возможность применения голосового помощника, который бы диктовал последовательность действий.

## **1.5 Голосовой помощник**

Голосовой ассистент - это компьютерная программа, предназначенная для обработки голосовых команд пользователя и выполнения соответствующих действий. Для того чтобы голосовой ассистент выполнял свои функции эффективно и быстро, он должен соответствовать определенным требованиям. Некоторые из основных требований для голосового ассистента включают в себя:

Распознавание речи: голосовой ассистент должен иметь способность распознавать речь пользователя и понимать его команды. Для этого он должен иметь качественный механизм распознавания речи.

– надежность: голосовой ассистент должен быть надежным и не выдавать ошибочных результатов. Для этого он должен быть тщательно протестирован и отлажен;

– скорость: голосовой ассистент должен быть быстрым и отзывчивым на команды пользователя. Он должен обрабатывать команды максимально быстро, чтобы пользователь мог получить необходимую информацию или выполнить нужные действия как можно быстрее;

– голосовой вывод: голосовой ассистент должен иметь возможность воспроизводить голосовой вывод для общения с пользователем. Голосовой вывод должен быть четким и понятным;

– безопасность: голосовой ассистент должен обеспечивать высокий уровень безопасности и защиты личных данных пользователей.

В целом, голосовой ассистент должен быть удобным, быстрым, надежным и безопасным, чтобы пользователи могли использовать его без особых проблем и получать максимальную пользу от его функциональности.

## **1.6 Вывод по первой главе**

В данной главе были поставлены цели и задачи, которые должны быть выполнены при реализации данной выпускной квалификационной работы. Обзор аналогов позволил получить представление: какой должна быть система, и какой функционал она должна реализовывать.



## 2 Архитектура и основные технические решения

### 2.1 Анализ технологий

Сегодня все мобильные устройства работают на системах android и ios. Но наиболее актуальной и востребованной сегодня является именно разработка мобильных приложений под Android.

Выбор фреймворка для разработки приложения под Android зависит от ряда факторов, таких как требования проекта, цели разработки, опыт команды разработчиков и технические возможности фреймворков.

Для разработки приложения взят за основу язык программирования Python (фреймворк Kivy и библиотеки KivyMD, Pillow). Его использование, вместо языков программирования Java или Kotlin, для разработки приложений под Android может быть целесообразным по ряду причин:

- простота и удобство: Python - это язык программирования, который отличается простым и понятным синтаксисом. Он имеет меньшее количество строгих правил и требует меньше кода для достижения тех же результатов, по сравнению с Java или Kotlin;

- богатый набор библиотек и фреймворков: Python обладает огромным набором библиотек и фреймворков, которые облегчают разработку приложений;

- кроссплатформенность: Python можно использовать для разработки не только под Android, но и для других платформ, таких как iOS, Windows, macOS и Linux. Это позволяет повторно использовать код и упрощает поддержку различных платформ;

- интеграция с другими языками и системами: Python легко интегрируется с другими языками программирования и системами [4].

### **2.1.1 Python Kivy**

Kivy - это открытый кросс-платформенный фреймворк на языке Python для разработки приложений, использующих инновационные многокаскадные пользовательские интерфейсы. Целью является обеспечение быстрого и простого проектирования взаимодействия и быстрого прототипирования, при этом делая код повторно используемым и готовым к развертыванию [3].

Основные преимущества Kivy:

- кроссплатформенность: Kivy позволяет создавать приложения для разных платформ, используя единый код;
- легкость разработки: Kivy облегчает процесс разработки приложений благодаря интуитивно понятному API и обширной документации;
- гибкость: Kivy позволяет создавать приложения любой сложности с использованием различных элементов, таких как кнопки, текстовые поля, изображения и многое другое;
- большое сообщество: Kivy имеет большое и активное сообщество разработчиков, что облегчает поиск решений проблем и получение поддержки.

Идеально подходит для разработки интерфейса системы.

### **2.1.2 Python Pillow**

Python Pillow - это библиотека для обработки изображений в Python. Она имеет простой в использовании интерфейс и обширный функционал, что делает ее популярным выбором для многих задач, связанных с обработкой и манипуляцией изображениями.

Основные преимущества Python Pillow:

- имеет простой и понятный интерфейс, что делает ее легкой в освоении для новичков в программировании или обработке изображений;

– предоставляет широкий набор функций и возможностей для обработки изображений;

– поддержка множества форматов: Pillow поддерживает большое количество популярных форматов изображений, включая JPEG, PNG, BMP, GIF, TIFF и другие;

– работает на различных операционных системах, включая Windows, macOS и Linux. Это позволяет вам создавать и запускать код на разных платформах без изменений;

– оптимизирована для работы с изображениями и обладает хорошей производительностью. Может обрабатывать большие изображения и выполнение сложных операций быстро и эффективно.

Идеально подходит для обработки изображения. На его основе можно разработать преобразователь изображения в String Art.

### **2.1.3 Python SoundDevice**

Технология Sounddevice является инструментом для работы с аудио в языке программирования Python. Она предоставляет простой и удобный интерфейс для записи и воспроизведения звука, а также для обработки аудиоданных в реальном времени. Основные аспекты и возможности этой технологии:

– обеспечивает простой способ записи аудио с микрофона или других источников;

– предоставляет возможность обрабатывать аудиоданные в реальном времени. Это позволяет применять различные эффекты, фильтры и обработку звука на лету. Например, можно изменять громкость, применять эквалайзеры, добавлять эффекты реверберации или задержки, и многое другое;

– может работать на различных операционных системах, включая Windows, macOS и Linux, что делает его универсальным инструментом для разработки аудио-приложений на языке Python.

Библиотека применима для разработки голосового помощника.

## 2.2 Разработка интерфейса системы

Интерфейс системы предполагает наличие трёх основных страниц (Рисунок 5). При проектировании интерфейса может быть применен следующий подход:

### 1. Первая страница (начальная страница):

– размещение элементов управления для загрузки изображения с возможностью выбора файла или загрузки изображения из галереи устройства;

– добавление полей для настройки параметров преобразования изображения в String Art, таких как количество гвоздиков, количество линий и толщина нитей;

– добавление кнопки для загрузки уже сгенерированного проекта с возможностью выбора файла.

### 2. Вторая страница (просмотр результата):

– центрированное размещение изображения, сгенерированного на основе загруженного изображения и настроек с первой страницы;

– размещение кнопок "Назад" и "Вперед", позволяющих пользователю вернуться на предыдущую страницу для загрузки другого изображения или изменения параметров преобразования изображения.

### 3. Третья страница (инструкция и голосовой помощник):

– размещение текстовой инструкции по процессу создания String Art проекта;

– добавление кнопок и ползунков для управления функциями инструкции, такими как пауза, поиск гвоздика, изменение скорости воспроизведения;

– интеграция голосового помощника, который может обладать всей функциональностью страницы, позволяя пользователям взаимодействовать с приложением голосовыми командами для выполнения различных операций.

Общая концепция интерфейса должна быть интуитивно понятной и простой в использовании. Ясное расположение элементов управления, четкая навигация между страницами и понятные инструкции помогут пользователям легко освоить приложение и насладиться процессом создания String Art проектов.

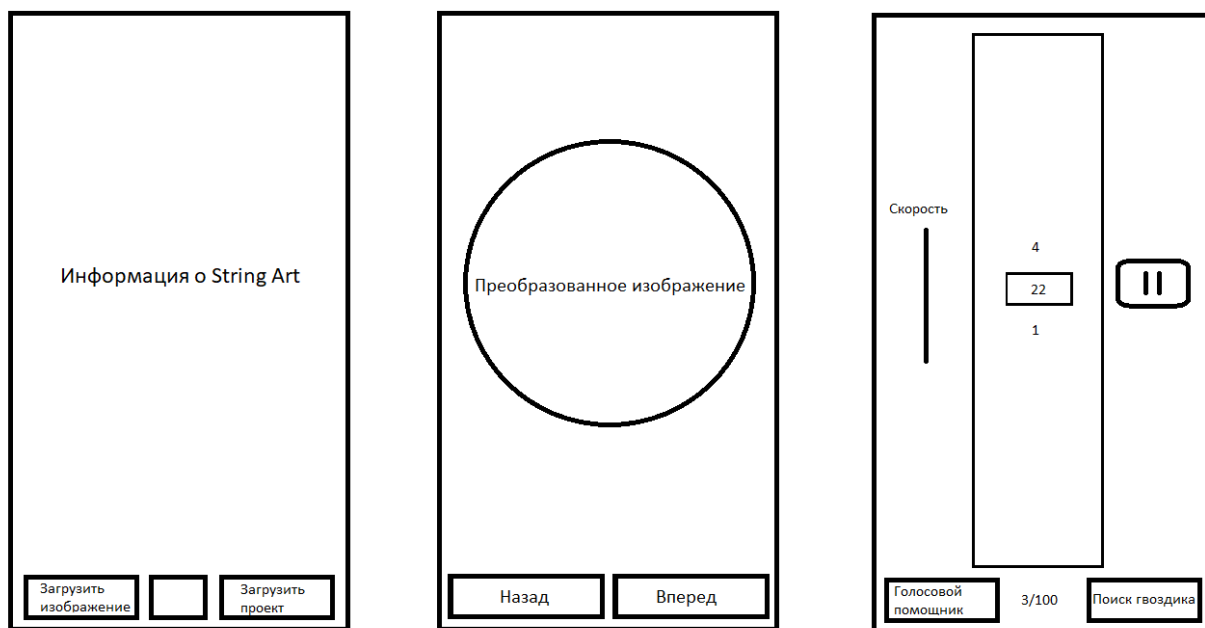


Рисунок 5 - Схематичная структура будущего приложения

### 2.3 Разработка UML модели проекта

Разработка UML модели проекта является важным этапом в жизненном цикле любого программного проекта. UML (Unified Modeling Language) - это стандарт, который используется для моделирования объектно-

ориентированных систем. Он предоставляет возможность визуального представления проекта, позволяя команде разработчиков, менеджерам и заказчикам более полно понимать его структуру и функциональность.

Основная цель разработки UML модели проекта – описать его архитектуру, компоненты, взаимодействия и зависимости. Это позволяет определить требования к проекту и убедиться, что он соответствует всем необходимым функциональным и производственным характеристикам. UML модель также упрощает коммуникацию между членами команды разработчиков, поскольку она обеспечивает единую терминологию и визуальную интерпретацию проекта.

Разработка UML модели проекта включает в себя создание диаграмм, которые представляют различные аспекты проекта. Например, диаграмма классов показывает классы, которые используются в проекте, и их связи друг с другом. Диаграмма последовательности демонстрирует, как различные компоненты взаимодействуют друг с другом в процессе выполнения определенных задач. Диаграмма состояний описывает, как объекты изменяют свое состояние в разных ситуациях.

В целом, разработка UML модели проекта - это ключевой этап, который позволяет убедиться в правильности концептуального дизайна проекта и обеспечить его успешную реализацию. Это позволяет команде разработчиков точнее определить требования к проекту, а также обеспечивает единую терминологию и понимание проекта между различными участниками команды [5].

### **2.3.1 Диаграмма вариантов использования**

Диаграмма вариантов использования - это один из наиболее часто используемых инструментов UML для моделирования поведения системы. Она позволяет описать функциональность системы из точки зрения взаимодействия с ее пользователями или другими внешними сущностями.

Диаграмма вариантов использования включает в себя актеров, сценарии использования и связи между ними.

Актеры - это внешние сущности, которые используют систему или взаимодействуют с ней. Они представляют собой роли, которые могут быть выполнены различными пользователями системы или другими системами. Сценарии использования описывают, как актеры используют систему для достижения своих целей. Они показывают последовательность шагов, необходимых для выполнения конкретной задачи, и включают в себя взаимодействие актеров с системой.

Связи между актерами и сценариями использования показывают, как актеры используют систему для выполнения своих задач. Они могут быть представлены различными способами, включая отношения включения, расширения и обобщения. Отношение включения показывает, что один сценарий использования включает в себя другой. Расширение используется для описания дополнительных шагов, которые могут быть выполнены в случае определенных условий. Обобщение используется для описания отношений наследования между сценариями использования. Диаграмма вариантов использования показана на Рисунок 6.

Варианты использования системы пользователем:

1) Открыть пособие по сборке заготовки. Пользователь открывает страницу, где подробно рассказывается, как изготовить каркас, на который можно в дальнейшем натягивать нить;

2) Создать проект. Чтобы создать проект, пользователь должен загрузить изображение, которое он хочет преобразовать, при этом у пользователя есть возможность установить количество пинов, количество линий и непрозрачность линии;

3) Загрузить файл проекта. Пользователь может загрузить файл уже ранее преобразованного изображения;

4) Ознакомиться с визуализацией итоговой картины. Пользователь может заранее посмотреть, что получится после проделанной им работы, следуя инструкции;

5) Просматривать инструкцию. На экране пользователь читает последовательность натяжений нити и в точности повторяет. При этом можно искать линию, если сбился, или продолжил после перерыва;

6) Использовать голосовой помощник. Пользователь может общаться с голосовым помощником для координации своих действий, либо для продиктовки последовательности линий, при этом голосовой помощник по команде может приостановить чтение, ускорять или замедлять темп.

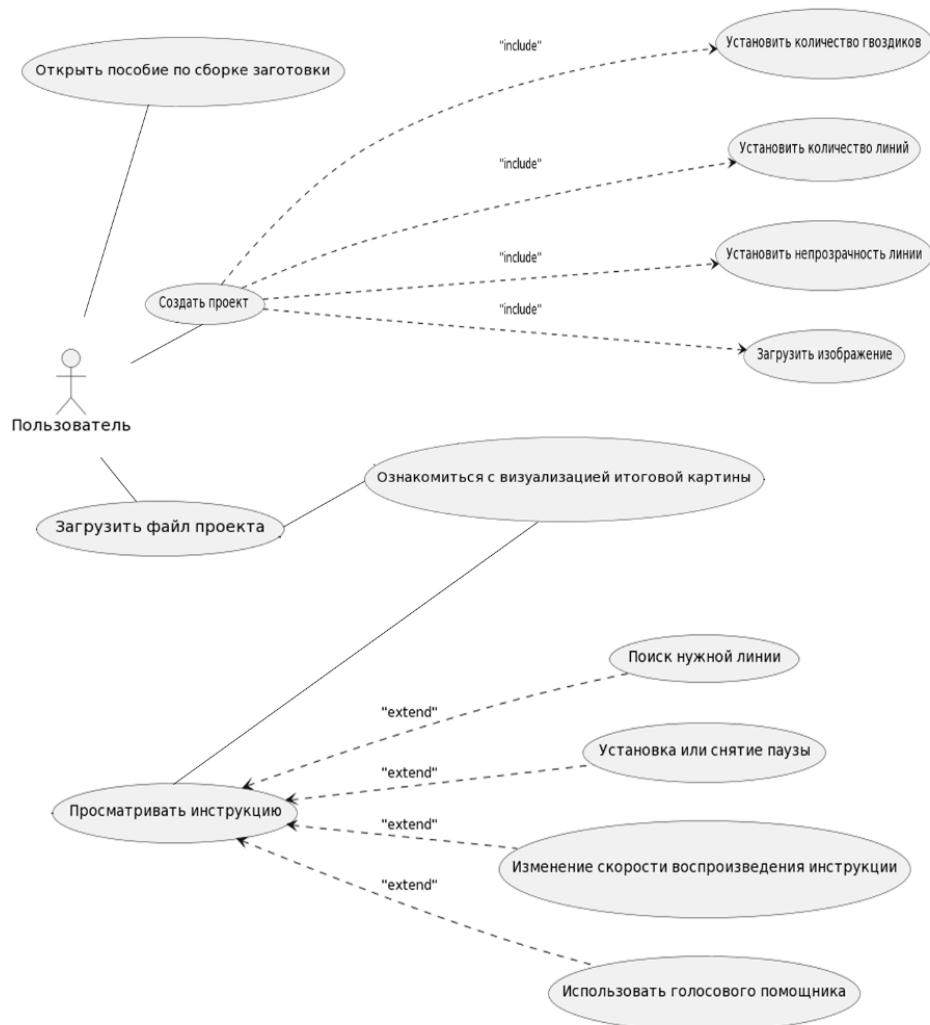


Рисунок 6 - Диаграмма вариантов использования



### 2.3.2 Диаграмма последовательности

Диаграмма последовательности - это вид диаграмм UML, который позволяет визуализировать взаимодействие объектов в системе во времени. Она представляет собой последовательность сообщений между объектами и позволяет понять, как объекты взаимодействуют между собой, какие сообщения передаются, в какой последовательности и какие операции выполняются.

Диаграмма последовательности состоит из объектов, сообщений и временных линий, которые представляют время выполнения операций и передачи сообщений между объектами. Каждый объект на диаграмме представлен вертикальной линией, которая показывает время жизни объекта, а сообщения представлены горизонтальными стрелками, которые показывают передачу информации между объектами. Диаграмма последовательности показана на Рисунок 7.

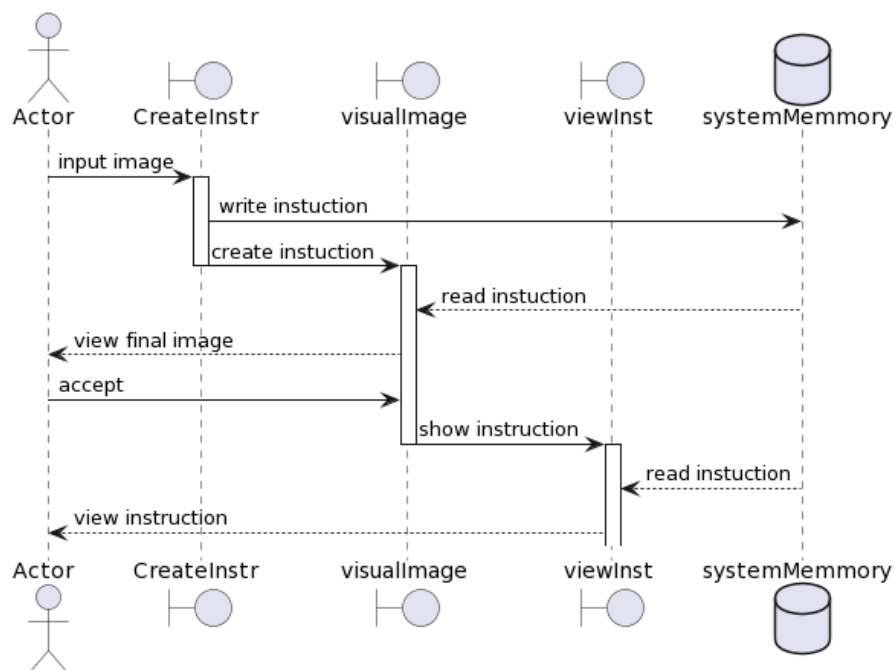


Рисунок 7 - Диаграмма последовательности

Диаграмма последовательности показывает: загрузку изображения с последующим преобразованием в инструкцию, записанную в память устройства, где в последующем она читается следующими активностями для визуализации картины и вывод, а на экран – в читаемом виде.

### 2.3.3 Диаграмма классов

Диаграмма классов используется для визуализации структуры классов и их взаимосвязей в системе. Она представляет собой графическое представление классов, их атрибутов, методов и связей между ними.

На диаграмме классов каждый класс представляется прямоугольником, внутри которого указывается имя класса. Внутри класса указываются его атрибуты (поля) и методы. Атрибуты отображаются с указанием их типа данных, а методы - с указанием их сигнатуры.

Связи между классами на диаграмме классов могут быть различными типами, такими как ассоциация, наследование, агрегация, композиция и другие. Они показывают взаимодействие и зависимости между классами. Например, ассоциация указывает, что один класс использует функциональность другого класса, наследование показывает отношение между базовым и производным классами, а агрегация и композиция указывают на отношение "часть-целое" между классами. Диаграмма классов представлена на Рисунок 8.

Основные классы **MainScreenView**, **ViewInstructionScreenView**, **ViewStrArtScreenView** представляют главный экран, экран просмотра инструкций и экран просмотра сгенерированного изображения соответственно.

Класс **ViewInstructionScreenView**. Содержит приватные атрибуты ``num_points``, ``list_inst``, ``label_references``, ``schedule_interval``, ``current_ind_list``, ``is_paused``, ``size_list_buffer_outlist``, ``buffer_outlist`` (количество точек, список инструкций, данные с label, интервал воспроизведения инструкций, текущий

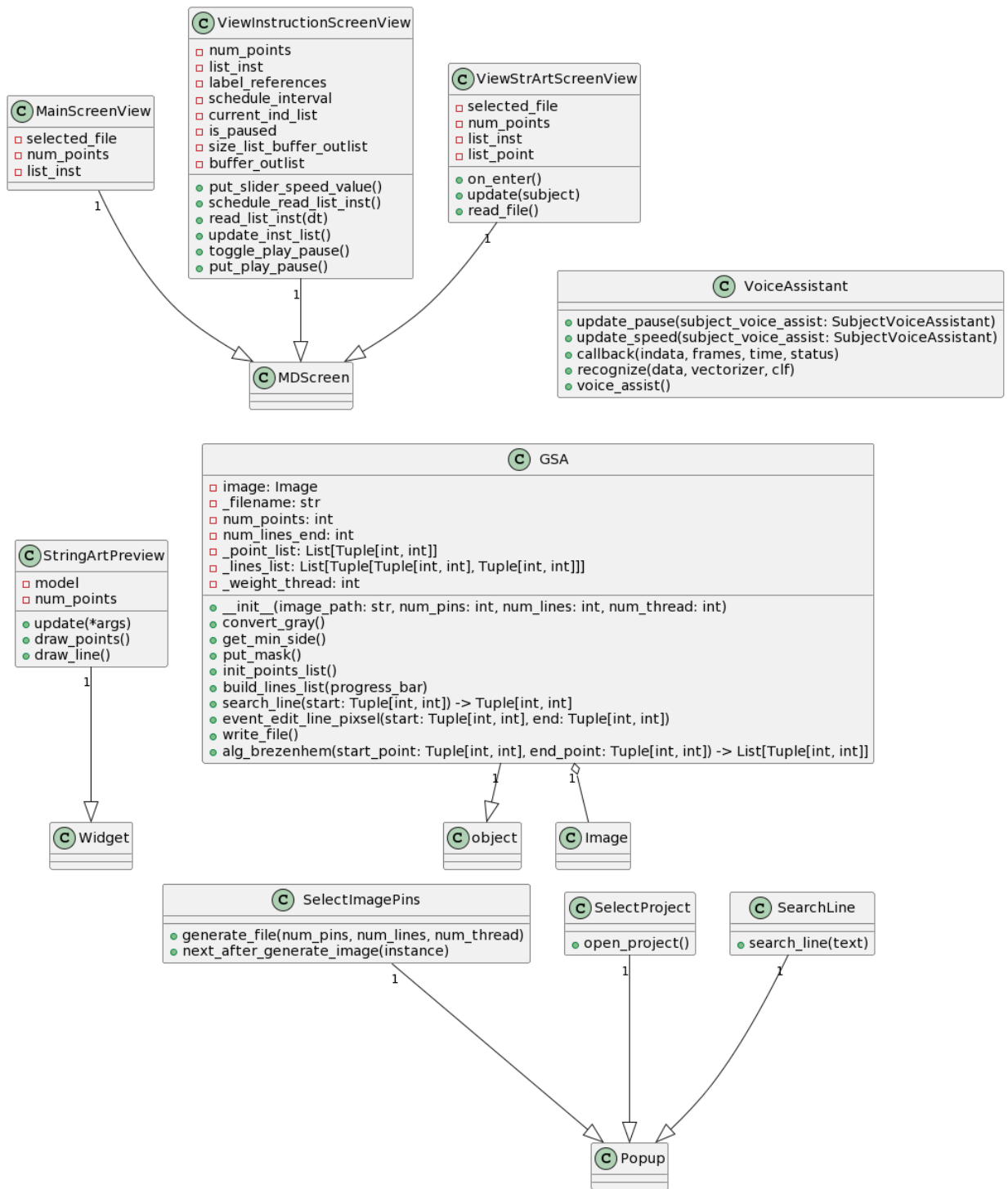


Рисунок 8 – Диаграмма классов

индекс списка, флаг паузы, размер буфера выходного списка, выходной список). Также содержит публичные методы `put\_slider\_speed\_value`, `schedule\_read\_list\_inst`, `read\_list\_inst`, `update\_inst\_list`, `toggle\_play\_pause`, `put\_play\_pause` (установка значения ползунка скорости, планирование

чтения списка инструкций, чтение списка инструкций, обновление списка инструкций, переключение между воспроизведением и паузой, установка флага воспроизведения/паузы).

Для загрузки изображения, установки параметров преобразования и загрузки файла проекта используются диалоговые окна `PopUp`. Класс **SelectImagePins** унаследован от `PopUp`. Служит для выбора изображения и задания параметров. Сам выбор и установка происходят в графическом интерфейсе. Сам класс содержит публичные методы `'generate_file'` (генерирует файл с инструкцией), `'next_after_generate_image'` (следующий шаг после генерации изображения).

Класс **GSA**. Служит для создания String Art изображения. Весь алгоритм преобразования происходит в его методах. Содержит приватные атрибуты `'image'`, `'_filename'`, `'num_points'`, `'num_lines_end'`, `'_point_list'`, `'_lines_list'`, `'_weight_thread'` (изображение, имя файла, количество точек, количество линий, список точек, список линий, вес линии). Также содержит публичные методы `'__init__'`, `'convert_gray'`, `'get_min_side'`, `'put_mask'`, `'init_points_list'`, `'build_lines_list'`, `'search_line'`, `'event_edit_line_pixel'`, `'write_file'`, `'alg_brezenhem'` (инициализация, преобразование в оттенки серого, получение минимальной стороны, добавление маски, инициализация списка точек, создание списка линий, поиск линии, удаление линии из изображения, запись файла, алгоритм Брезенхэма).

## 2.4 Модуль «Преобразователь изображений String Art»

Модуль генерации изображения String Art должен соответствовать определенным требованиям, чтобы обеспечивать высокое качество создаваемых изображений и удобство использования. Некоторые из основных требований для модуля генерации String Art изображения могут включать в себя:

– гибкость настроек: модуль должен обеспечивать гибкие настройки, которые позволят пользователю настраивать изображения с учетом разных параметров, таких как размер изображения, количество точек, количество линий, тип линий и т.д;

– легкость использования: модуль должен быть легко доступным и простым в использовании, даже для пользователей, не знакомых с техническими аспектами генерации изображений;

– высокое качество изображения: модуль должен обеспечивать высокое качество создаваемых изображений, включая четкость и детализацию линий и точек;

– поддержка разных форматов: модуль должен поддерживать различные форматы изображений, такие как PNG, JPG и другие, чтобы пользователь мог легко экспортировать и использовать изображения в различных приложениях;

– оптимизация производительности: модуль должен быть оптимизирован для высокой производительности и быстрого создания изображений, даже при большом количестве точек и линий;

– надежность: модуль должен быть надежным и стабильным, чтобы пользователи могли использовать его без проблем и не столкнуться с ошибками или сбоями в работе.

В целом, модуль генерации String Art изображения должен быть гибким, легким в использовании, обеспечивать высокое качество изображений и быть надежным и быстрым.

## **2.5 Модуль «Голосовой помощник»**

Для голосового помощника в приложении String Art, можно выделить следующие требования:

– озвучивание инструкций: Голосовой помощник должен иметь возможность читать инструкции вслух для пользователя. Он должен быть

способен передавать информацию о шагах и действиях, необходимых для создания String Art изображения;

- управление озвучиванием: Голосовой помощник должен позволять управлять озвучиванием инструкций. Это может включать функции паузы, остановки и возобновления озвучивания, чтобы пользователь мог контролировать темп и прогресс инструкций;

- натуральный голос: Голосовой помощник должен иметь натуральный и понятный голос, который будет комфортным для прослушивания пользователем. Он должен быть четким и ясным, чтобы пользователь мог легко понимать озвучиваемые инструкции;

- интеграция с приложением: Голосовой помощник должен быть интегрирован в приложение String Art, чтобы пользователь мог управлять им и получать инструкции без необходимости переключаться на другие приложения или устройства;

- поддержка ошибок и обратной связи: Голосовой помощник должен быть способен обрабатывать ситуации ошибок и предоставлять соответствующую обратную связь пользователю. Это может включать информацию об ошибках в выполнении команд или просьбу повторить инструкцию.

При этом голосовой помощник должен быть доступен только на странице просмотра инструкции. В его функционал должны входить:

- установка и снятие паузы;
- изменение скорости воспроизведения инструкции;
- поиск гвоздика, с которого начать воспроизведение.

## **2.6 Вывод по второй главе**

Во второй главе произведен выбор языка и его фреймворка. В качестве языка программирования был выбран язык Python, в качестве фреймворка для разработки интерфейса Kivy и библиотека KivyMD, в качестве

библиотеки для разработки преобразователя изображения Pillow, в качестве библиотеки для разработки голосового помощника SoundDevice.

Разработан архитектурный подход к созданию будущего android-приложения путем создания UML диаграмм. Выделены основные модули работы приложения, к каждому модулю разработаны соответствующие требования.

## 3 Реализация приложения

### 3.1 Начальная страница

При запуске приложения появляется начальная страница. Она состоит из нескольких компонентов и функциональности, которые обеспечивают удобство использования и навигацию для пользователей. Рассмотрим эти компоненты и их функциональность

– Загрузка изображения: Эта функция (Реализуется классом `SelectImagePins`) предоставляет пользователю возможность загрузить собственное изображение, которое будет использовано в качестве основы для создания произведения `String Art`. При запуске открывается диалоговое окно выбора файла, где пользователь может выбрать нужный файл с изображением.

– Продолжить проект: Эта функция (Реализуется классом `SelectProject`) предназначена для возможности продолжения работы над уже существующим проектом. При её запуске пользователю предлагается выбрать файл с сохраненным проектом, который был создан ранее. После выбора файла, приложение загружает данные проекта, позволяя пользователю продолжить работу с уже имеющимися настройками.

– Дополнительные опции: Данная кнопка предоставляет дополнительные опции и функциональность приложения. При нажатии на нее открывается всплывающее меню с различными настройками и функциями. Одной из доступных опций в меню является "Просмотр инструкции по разработке холста для натягивания нитей". При выборе этой опции, приложение отображает подробную инструкцию или руководство, которое объясняет пользователю основы и шаги по созданию холста для `String Art` проектов. Инструкция может содержать текстовую информацию, изображения, чтобы помочь пользователям понять процесс и достичь желаемых результатов.



Компоненты и функциональность на начальной странице приложения - это интуитивно понятные элементы интерфейса, которые помогают пользователям начать новый проект или продолжить работу с уже существующим. Кнопки "Загрузка изображения" и "Продолжить проект" предоставляют удобный способ импортирования изображений и сохраненных данных, а кнопка меню позволяет получить доступ к дополнительным опциям, включая просмотр инструкций для разработки холста. Интерфейс представлен на Рисунок 9.

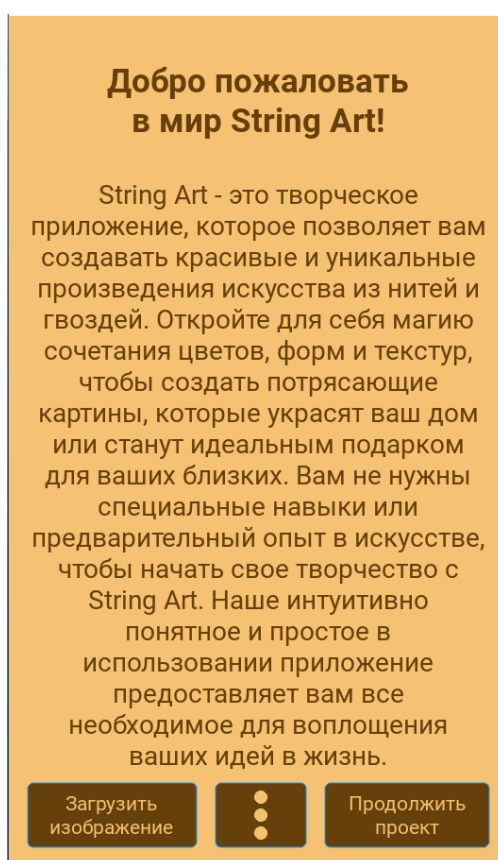


Рисунок 9 – Начальная страница

### 3.1.1 Функциональность кнопки «Загрузить изображение»

При нажатии на кнопку появляется диалоговое окно, где представляется возможным загрузить изображение из памяти устройства, также можно установить желаемые параметры для преобразованного

изображения, такие как количество гвоздиков, количество линий, непрозрачность линии (Рисунок 10). При выборе всех позиций можно запустить процесс преобразования изображения (запуске функции `generate_file()`), что визуально отображается шкалой загрузки.

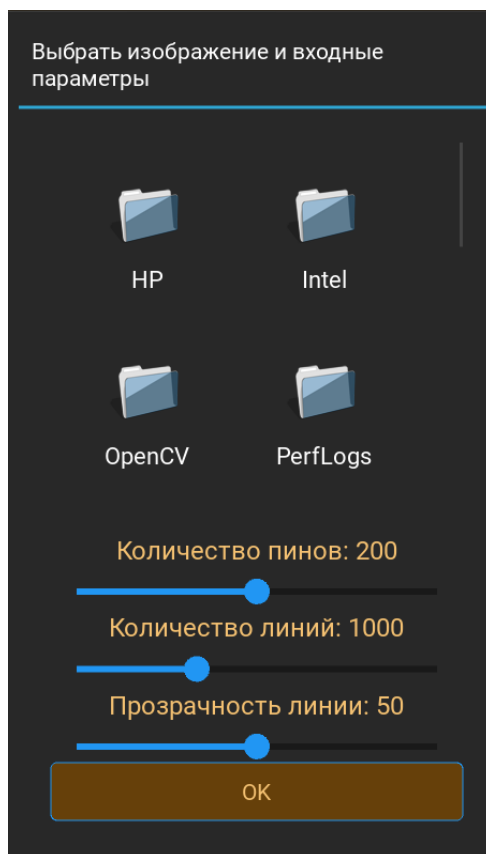


Рисунок 10 – Диалоговое окно «Выбор изображения»

### 3.2 Страница просмотра сгенерированного изображения

Страница после генерации String Art изображения предоставляет пользователю возможность оценить окончательный результат своей работы и принять решение о дальнейших действиях. Кнопки "Назад" и "Вперед" позволяют пользователю выбирать, нужно ли вернуться к настройке проекта или продолжить сразу с рисованием на основе созданного изображения. Эта страница служит важным промежуточным шагом в процессе создания String

Art произведения и позволяет пользователю принять более информированное решение о следующих действиях.

В центре страницы расположено изображение, которое представляет собой результат создания String Art произведения (Реализовано в классе `StringArtPreview`). Это изображение отражает то, как будет выглядеть полотно с натянутыми нитями и гвоздями. Пользователь может визуально оценить результат своей работы и оценить, насколько близко полученное изображение соответствует его ожиданиям. Функция `draw_points()` рисует все гвоздики, а функция `draw_line()` рисует все линии, координаты линий считываются из файла функцией `read_file()` класса `ViewStrArtScreenView`. Листинг функций `draw_line()` и `draw_point()` представлен в приложении А.

Кнопка "Назад". Эта кнопка предназначена для возврата пользователя на предыдущую страницу, в данном случае - на начальную страницу приложения. При нажатии на кнопку пользователь может вернуться к началу процесса и изменить параметры проекта, загрузить новое изображение или выполнить другие действия.

Кнопка "Вперед". Данная кнопка переносит пользователя на следующую страницу, где предоставляется функциональность для помощи в рисовании полученного String Art изображения. Нажатие на эту кнопку позволяет пользователю перейти к следующему шагу и приступить к созданию полотна на основе представленного изображения.

Интерфейс представлен на Рисунок 11.

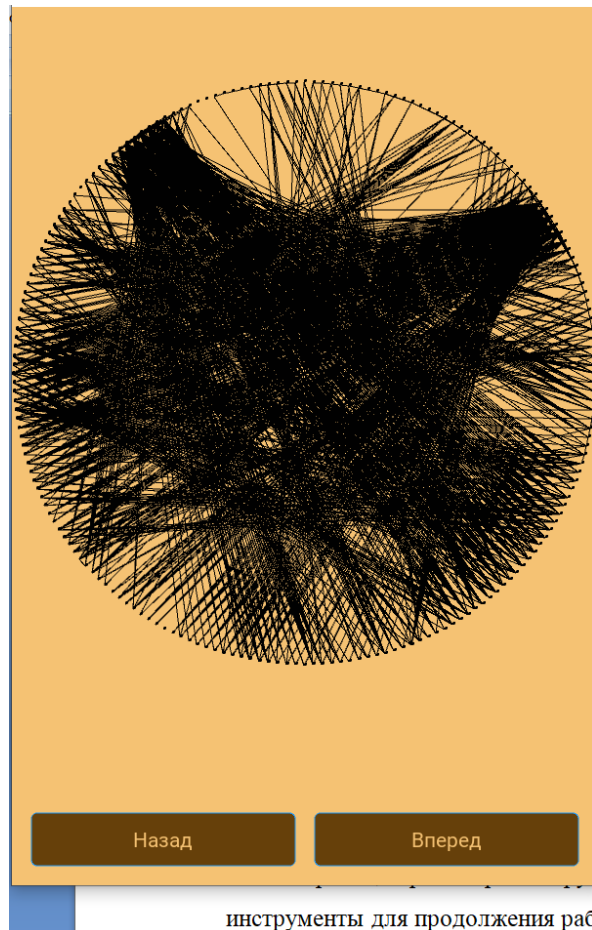


Рисунок 11 – Страница просмотра сгенерированного изображения

### 3.3 Страница просмотра инструкции

Страница просмотра инструкции предоставляет пользователю удобные инструменты для продолжения работы над проектом (Реализуется в классе `ViewInstructionScreenView`).

При отображении страницы функция `on_start()` заполняет список для вывода на экран частью инструкции из файла, после чего функцией `update_inst_list()` выводит все на экран.

В центре страницы отображается текущий гвоздик, который должен быть установлен на холсте в соответствии с инструкцией. Это помогает пользователю определить следующий шаг и продолжить работу.

Система автоматически меняет отображаемый гвоздик через определенный интервал времени, для этого используется `schedule_once` из

библиотеки `kivy.clock`. Функция `read_list_inst()` формирует списки для вывода на экран через промежуток времени задаваемый `schedule_once`. Скорость смены гвоздиков может быть настроена. Для этого на экране расположен `slider`, которым можно задавать значение для изменения скорости. При изменении значения на `slider` запускается функция `put_slider_speed_value()`, которая изменяет интервал времени для смены гвоздиков.

На странице присутствуют элементы управления, позволяющие пользователю регулировать скорость смены гвоздиков. Также присутствует кнопка "Пауза", которая приостанавливает смену гвоздиков, позволяя пользователю замедлить процесс или восстановить его в нужный момент, реализуется функцией `put_play_pause()`.

Есть возможность найти гвоздик в инструкции. Функция позволяет пользователю быстро найти гвоздик в инструкции, чтобы начать работу с него, если требуется продолжить проект с определенного момента. Пользователь может ввести номер или описание гвоздика, и приложение автоматически найдет соответствующую часть инструкции (Класс `SearchLine` функция `search_line()`).

На этой странице включен голосовой помощник, обладающий всей функциональностью страницы. Пользователь может использовать голосовые команды для управления скоростью, паузой, поиска гвоздика и других функций. Голосовой помощник помогает сделать процесс более удобным и интуитивным, особенно при работе с руками, занятыми установкой гвоздиков. Подробное описание реализации в пункте 3.5.

Интерфейс представлен на Рисунок 12.

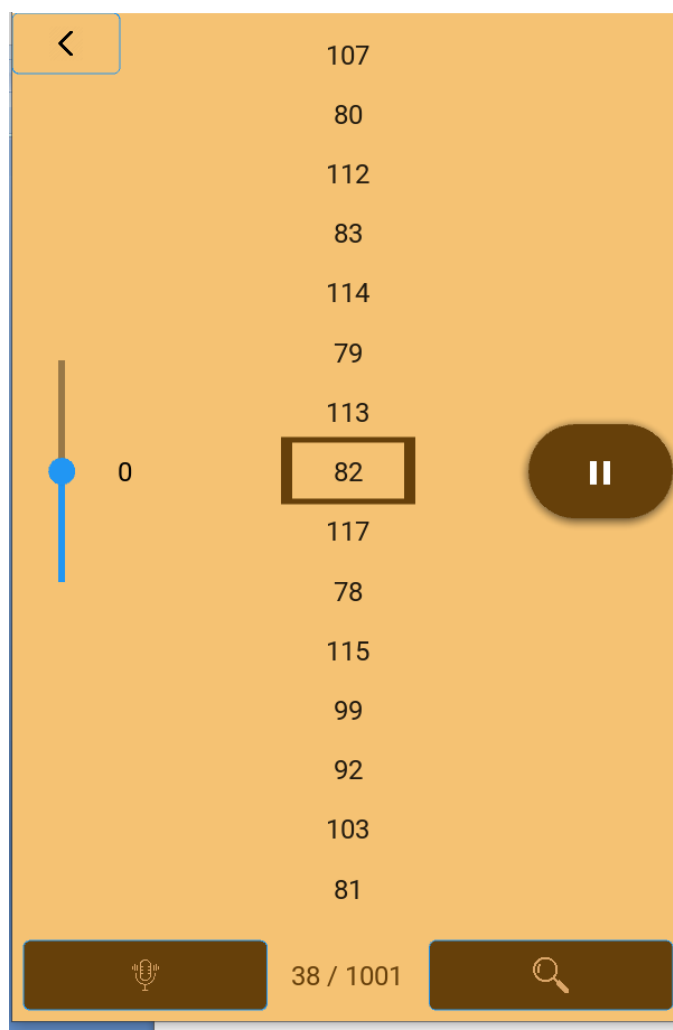


Рисунок 12 – Страница просмотра инструкции

### 3.4 Разработка модуля «Преобразователь изображений String Art»

Модуль принимает входные параметры (количество гвоздиков, количество линий, непрозрачность линии), после чего начинает преобразовывать изображение.

Модуль поддерживает форматы изображений, таких как JPEG, PNG. На изображение накладывается маска круга, чтобы придать нужную форму итоговому рисунку. На выходе в память устройства записывается файл с подробной инструкцией (функция `write_file()`).

### 3.4.1 Алгоритм работы модуля

Работа модуля осуществляется по следующему алгоритму

1) цветное изображение преобразуется в изображение с оттенками серого (функция `convert_gray()`).

2) затем изображение обрезается до круга (функция `put_mask()`).

3) помещает виртуальный круговой обод с  $P$  равноудаленными точками на нем, количество точек (гвоздиков) задаётся пользователем (функция `init_point_list()`).

4) принимает первую точку ( $p = 0$ ) в качестве отправной точки и начинаем поиск линий (функция `build_lines_list()`).

5) находит следующую точку  $p'$ , чтобы провести линию из точки  $p$ , найдя линию с наибольшей интенсивностью, используя алгоритм Брезенхэма (функция `alg_brezenhem()`) [7].

6) интенсивность линии вычисляется путем суммирования всех значений пикселей линии от  $p$  до  $p'$ .

7) новая линия записывается в список итоговых линий.

8) исходное изображение модифицируется таким образом, что пиксели под линией от  $p$  до  $p'$  становятся светлее, чтобы в следующий раз та же линия не рисовалась (функция `event_edit_line_pixel()`).

9)  $p$  устанавливается в качестве новой начальной точки  $p$ , а затем шаги с 5 повторяются до тех пор, пока не будут нарисованы  $L$  линий (функция `search_line()`).

### 3.5 Разработка голосового помощника

Голосовой помощник был интегрирован в разрабатываемое приложение, для подсказки алгоритма прохождения по гвоздикам (работает по желанию пользователя). Для этого были использованы материалы с удаленного репозитория [6]. Для распознавания и обработки речи

использовались технологии распознавания речи, такие как Google Cloud Speech-to-Text. Для распознавания голоса в отсутствие подключения интернета используется языковая модель Vosk.

Голосовой помощник взаимодействует с пользователем, предоставляя голосовую обратную связь на выполненные команды. Например, подтверждение установки паузы или информирование пользователя о текущей скорости воспроизведения.

За реализацию голосового помощника отвечает класс `VoiceAssistant`. Функция `voice_assist()` прослушивает микрофон и записывает речь, после чего запускается функция `recognize()`, которая уже расшифровывает текст.

Так как вся функциональность страницы находится в классе `ViewInstructionScreenView`, пришлось воспользоваться паттерном «Наблюдатель» для передачи данных и запуска функций вне родительского класса.

При команде поиска гвоздика в инструкции, производит распознавание числа для поиска соответствующего гвоздика. При этом для распознавания числа в речи используется функция `text2int()`, которая ищет словесные представления чисел и преобразует их в числовой вид. Листинг функции представлен в приложении Б.

### **3.6 Вывод по третьей главе**

В третьей главе представлен подход к построению приложения по преобразованию изображения в `String Art`. Также приведены результаты работы в целом.

Разработанное приложение соответствует всем поставленным требованиям, приложение преобразует изображение в набор линий, читает инструкцию, предоставляет возможности голосового помощника.



## ЗАКЛЮЧЕНИЕ

В результате работы над проектом было создано полнофункциональное приложение, которое обладает гибкостью и множеством возможностей для пользователей: оно позволяет загружать изображения, генерировать String Art варианты, рисовать и воплощать созданные проекты на холсте.

В ходе выполнения работы был проведен анализ существующих аналогов, разработаны и изучены виды и этапы разработки, выбраны средства разработки, обозначен функционал программного продукта и выполнена программная реализация.

Приложение реализовано с использованием языка программирования Python фреймворки Kivy, Pillow. Разработанное приложение отвечает всем поставленным задачам и первоначальным задумкам. В процессе работы успешно внедрен голосовой помощник. Он предоставляет дополнительные возможности управления и удобство использования, позволяя пользователям взаимодействовать с приложением с помощью голосовых команд. Одним из ключевых достижений проекта является разработка преобразователя, способного преобразовывать изображения в формат String Art.

В дальнейшем планируется продолжить улучшение и доработку приложения. Это может включать расширение функциональности, оптимизацию производительности и добавление новых инструментов и возможностей, чтобы удовлетворить потребности и желания наших пользователей.

Результаты работы были представлены на XX Всероссийской научно-практической конференции «Результаты современных научных исследований и разработок», Пенза. – 2023 [8].

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Техника Стринг Арт: создание картин из гвоздей и нитей: [Электронный ресурс]. – URL: <https://zvetnoe.ru/club/poleznye-stati/string-art/> (дата обращения: 14.12.2022).
2. Стринг-арт – что это такое, история возникновения, особенности техники и идеи: [Электронный ресурс]. – URL: <https://womanadvice.ru/string-art-cto-eto-takoe-istoriya-vozniknoveniya-osobennosti-tehniki-i-idei> (дата обращения: 14.12.2022).
3. StringArtGenerator: [Электронный ресурс]. – URL: <https://halfmonty.github.io/StringArtGenerator/> (дата обращения: 14.12.2022).
4. Kivy: [Электронный ресурс]. – URL: <https://github.com/kivy/kivy>.
5. Розенберг Д., Скотт К. Применение объектного моделирования с использованием UML и анализ прецедентов.: Пер. с англ. М.: ДМК Пресс, 2002.
6. Offline-Voice-Assistant-with-Machine-Learning-on-python: [Электронный ресурс]. – URL: <https://github.com/PythonHubStudio/Offline-Voice-Assistant-with-Machine-Learning-on-python> (дата обращения: 04.06.2023).
7. Алгоритмические основы компьютерной графики: Методическое пособие для студ. спец. 1-40 02 02 «Электронные вычислительные средства» днев. формы обуч. /М.З. Лившиц [и др.] – Минск: БГУИР, 2009. – 39 с.
8. Федченко Ф. В. Android-приложение для создания картины посредством нитеплетения и с помощью голосового помощника / Сб. статей XX Всероссийской научно-практической конференции «Результаты современных научных исследований и разработок», Пенза. – 2023. – С. 32-35.
9. Система менеджмента качества Общие требования к построению, изложению и оформлению документов учебной деятельности // Стандарт организации СТО СТУ 7.5–07–2021. – URL: <https://about.sfu-kras.ru/docs/8127/pdf/237025> (дата обращения: 20.06.2023).

## ПРИЛОЖЕНИЕ А

### Листинг кода функций draw\_line() и draw\_points()

```
# Рисует круг из точек
def draw_points(self):
    self.canvas.clear()
    angle = 2 * math.pi / self.num_points
    radius = min(self.width, self.height) / 2 - 1

    with self.canvas:
        Color(1 / 255, 1 / 255, 1 / 255)
        for i in range(self.num_points):
            x = int(radius * math.cos(i * angle))
            y = int(radius * math.sin(i * angle))
            self.model.list_point.append([x, y])
            Ellipse(pos=(self.center_x + x - 1, self.center_y + y - 1), size=(3, 3))

# Рисует линии
def draw_line(self):
    start = int(self.model.list_inst[0])
    with self.canvas:
        for i in range(1, len(self.model.list_inst)):
            Color(1 / 255, 1 / 255, 1 / 255)
            Line(points=(self.center_x - int(self.model.list_point[start][0] + 1),
                        self.center_y - int(self.model.list_point[start][1] + 1),
                        self.center_x -
int(self.model.list_point[int(self.model.list_inst[i)])[0] + 1),
                        self.center_y -
int(self.model.list_point[int(self.model.list_inst[i)])[1] + 1)))
            start = int(self.model.list_inst[i])
```

## ПРИЛОЖЕНИЕ Б

### Листинг функции text2int()

```
# Распознает текстовое представление числа и преобразует его в число
def text2int(textnum, numwords=None):
    if numwords is None:
        numwords = {}
    if not numwords:
        units = [
            "ноль", "один", "два", "три", "четыре", "пять", "шесть", "семь",
            "восемь",
            "девять", "десять", "одиннадцать", "двенадцать", "тринадцать",
            "четырнадцать", "пятнадцать",
            "шестнадцать", "семнадцать", "восемнадцать", "девятнадцать",
        ]
        tens = ["", "", "двадцать", "тридцать", "сорок", "пятьдесят", "шестьдесят",
            "семьдесят", "восемьдесят",
            "девяносто"]
        scales = ["сто", "тысяч", "миллион", "миллиард", "триллион"]
        numwords["и"] = (1, 0)
        for idx, word in enumerate(units):
            numwords[word] = (1, idx)
        for idx, word in enumerate(tens):
            numwords[word] = (1, idx * 10)
        for idx, word in enumerate(scales):
            numwords[word] = (1, 10 ** (idx * 3 or 2))
    current = result = 0
    for word in textnum.split():
        if word not in numwords:
            continue
        scale, increment = numwords[word]
        current = current * scale + increment
        if scale > 100:
            result += current
            current = 0
    return result + current
```

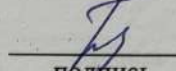
Министерство науки и высшего образования РФ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

 О. В. Непомнящий  
подпись

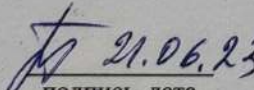
« 21. » 06 2023г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.01 Информатика и вычислительная техника

**String Art: android-приложение с голосовым помощником**

Руководитель

 21.06.23  
подпись, дата

доцент кафедры ВТ, канд. тех. наук  
должность, ученая степень

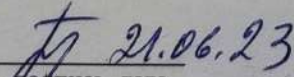
Л.И. Покидышева  
инициалы, фамилия

Выпускник

 21.06.23  
подпись, дата

Е.В. Федченко  
инициалы, фамилия

Нормоконтролер

 21.06.23  
подпись, дата

Л.И. Покидышева  
инициалы, фамилия

Красноярск 2023