**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 2](#_Toc104623039)

[1 ПОСТАНОВКА ЗАДАЧИ И ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc104623040)

[2 МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ 7](#_Toc104623041)

[3 ПРОЕКТИРОВАНИЕ ЛОГИКИ 9](#_Toc104623042)

[4 ИСПОЛЬЗУЕМЫЕ ИНСТРУМЕНТЫ 13](#_Toc104623043)

[ЗАКЛЮЧЕНИЕ 14](#_Toc104623044)

# **ВВЕДЕНИЕ**

В истории человечества существовал такой вид искусства, как мозаика. Из маленьких кусочков разной текстуры и фактуры люди создавали произведения искусства. С ходом времени и развитием технологий начали появляться первые компьютеры, однако их технические характеристики не позволяли показывать изображения в том виде, в каком мы видим их в реальной жизни. Выходом были пиксельные изображения – новый вид мозаики. Изображение представляло собой сетку, в которой отдельный квадратик мог быть закрашен определенным цветом. Первые компьютеры не позволяли делать сетку большого масштаба, однако со временем этот показатель увеличивался. От сетки 32 на 32 пикселя мы перешли на сетки размером 1920 на 1080 пикселей, однако, как раньше, так и сейчас существуют люди, которые в специальном программном обеспечении могут создавать пиксельные рисунки высокого качества. Пример такого рисунка можно увидеть на рисунке 1.



Рисунок 1 – Пример пиксельного рисунка.

Сначала пиксельные рисунки не пользовались популярностью, но со временем их создание выделилось в отдельное течение – PixelArt. Это направление захватило многие сферы – геймдизайн, комиксы, диджитал-арт, киноиндустрия. Как говорилось ранее, все пиксельные рисунки создаются в специальных программах – графических редакторах. Актуальность данного курсового проекта заключается в том, что потребность в удобных графических редакторах довольно высока.

Целью курсовой работы является получение навыков проектирования посредством создания графического редактора пиксельной графики.

Для достижения данной цели необходимо выполнить следующие задачи:

1) описать предметную область и сопоставить аналоги;

2) построить модели предметной области;

3) спроектировать программу;

4) реализовать программу.

# **1 ПОСТАНОВКА ЗАДАЧИ И ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ**

Целью курсовой работы является получение навыков проектирования посредством создания графического редактора пиксельной графики для устройств с ОС Android.

В соответствии с заданием программа должна выполнять следующие задачи:

* создавать рисунки;
* изменять рисунки;
* сохранять рисунки;
* рисовать пикселями на холсте;
* стирать пиксели на холсте;
* заливать область холста одним цветом;
* копировать цвет с холста;
* производить выбор цвета из цветового круга.

После запуска программы пользователю отображаются две кнопки: создать новое изображение, открыть изображение.

После создания или открытия изображения приложение переносит пользователя на слой для рисования.

Слой для рисования состоит из трех частей: холст, кнопка меню и панели с кнопками.

Холст представляет из себя белое поле, на котором сеткой отображаются пиксели.

Кнопка меню располагается в правой части экрана и при нажатии на нее открывается панель меню, где можно выбрать один из пунктов: создать новое изображение, открыть существующее изображение, сохранить изображение, посмотреть информацию о приложении и др.

Панель с кнопками содержит 5 кнопок для пяти функций: заливка области с одинаковыми пикселями одним цветом, копирование цвета с холста, стирание пикселей, открытие палитры цветов и др.

Для рисования на холсте пользователь должен будет выбрать цвет из палитры (предустановленный цвет – чёрный) и нажать в любом месте на экране телефона в области холста. После этих действий на холсте появится квадратная точка, выбранного цвета. Для рисований линий пользователю нужно будет нажать на экран в области холста и провести по экрану.

Для выбора цвета пользователь должен нажать на кнопку выбора цвета, после чего выбрать наиболее подходящий цвет в открывшейся палитре цветов.

Для заливки области пикселей одним цветом пользователь должен выбрать цвет, затем нажать на кнопку заливки и нажать на холсте, после чего область цветом одинаковых пикселей зальётся выбранным.

Для копирования цвета с холста пользователю необходимо нажать на кнопку копирования цвета и нажать на экран в области холста, после чего установится цвет палитры, на который нажал пользователь.

Для стирания пикселей с холста пользователю необходимо нажать на кнопку стирания цвета и нажать на экран в области холста, после выбранный пиксель сотрется.

**Графический** **редактор пиксельной графики** - программа, позволяющая создавать и редактировать изображения на уровне пикселей на экране компьютера или телефона: рисовать линии, раскрашивать области экрана, создавать надписи, обрабатывать изображения и т.д. Графические редакторы состоят из холста и вспомогательного меню. В таблице 1 представлены аналоги приложения.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Функции/Аналоги | Pixel Studio | PixelArt | Pixel Art editor | PixLab |
| Создание своего рисунка | **+** | **-** | **+** | **+** |
| Рисование пикселями на холсте | **+** | **+** | **+** | **+** |
| Копирование цвета с холста | **+** | **-** | **+** | **+** |
| Стирание пикселей с холста | **+** | **-** | **+** | **+** |
| Заливка области холста одним цветом | **+** | **-** | **-** | **+** |
| Выбор цвета из цветового круга | **-** | **+** | **+** | **+** |
| Анимирование | **+** | **-** | **-** | **-** |

Таблица 1 – Таблица аналогов

**Функциональные требования:**

* приложение позволяет создавать или открывать и сохранять созданные пиксельные рисунки;
* приложение позволяет рисовать пикселями на холсте;
* приложение позволяет заливать область холста одним цветом;
* приложение позволяет копировать цвет с холста;
* приложение позволяет использовать для рисования заранее заготовленные цвета, хранящиеся в цветовой палитре;
* приложение позволяет просмотреть информацию о программе.

# **2 МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ**

Для понимания, какой функционал разрабатываемого приложения будет доступен пользователям, была построена диаграмма вариантов использования, представленная на рисунке 2.

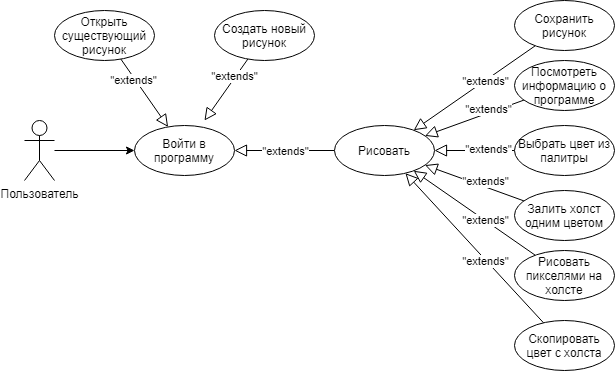


Рисунок 2 – Диаграмма вариантов использования

Для демонстрации общей структуры системы была построена общая диаграмма классов, изображенная на рисунке 3.

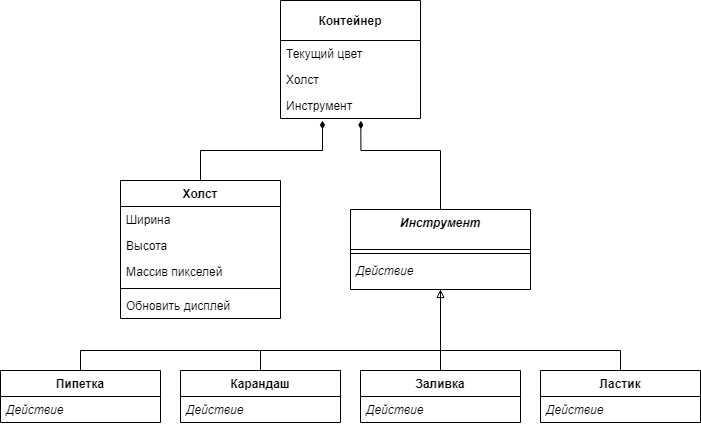


Рисунок 3 – Общая диаграмма классов

Чтобы описать поведение системы в целом, была построена диаграмма состояний, изображенная на рисунке 4.

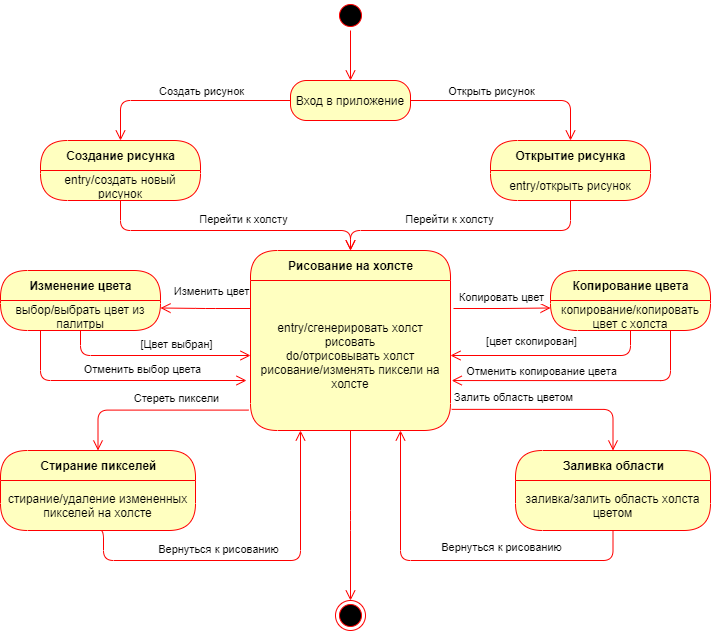


Рисунок 4 – Диаграмма состояний

На диаграмме можно увидеть основные состояния системы и переходы между ними. Основным состоянием является «Рисование на холсте». Во время этого состояния происходит отрисовка холста, а также вносятся основные изменения в изображения.

# **3 ПРОЕКТИРОВАНИЕ ЛОГИКИ**

Для демонстрации общей структуры иерархии классов, атрибутов, методов, интерфейсов и взаимосвязей между ними была построена диаграмма классов, изображенная на рисунке 5.

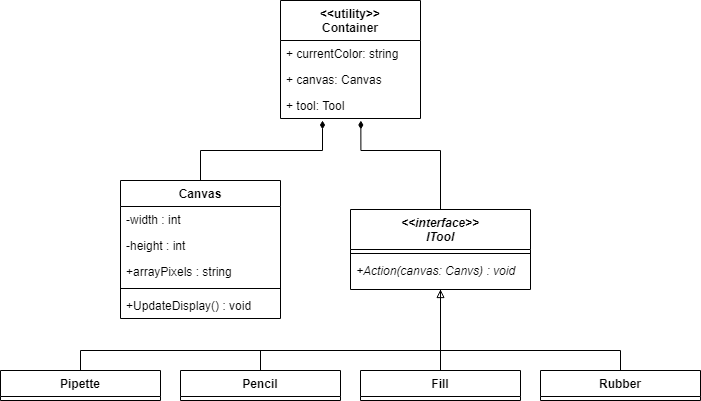


Рисунок 5 – Диаграмма классов

Для описания логики программы ниже представлены следующие алгоритмы:

* алгоритм работы инструмента «карандаш»;
* алгоритм работы инструмента «ластик»;
* алгоритм работы инструмента «пипетка»;
* алгоритм работы инструмента «заливка».

Алгоритм работы инструмента «Карандаш»:

1. Считать номер строки и столбца, где был произведён клик;
2. Присвоить полю arrayPixels объекта Canvas значение выбранного цвета по координатам считанных из первого пункта;
3. Вызвать у объекта Canvas метод UpdateDisplay.



Рисунок 5 – Блок-схема алгоритма работы инструмента «Карандаш»

Алгоритм работы инструмента «Ластик»:

1. Считать номер строки и столбца, где был произведён клик;
2. Присвоить полю arrayPixels объекта Canvas пустую строку по координатам считанных из первого пункта;
3. Вызвать у объекта Canvas метод UpdateDisplay.

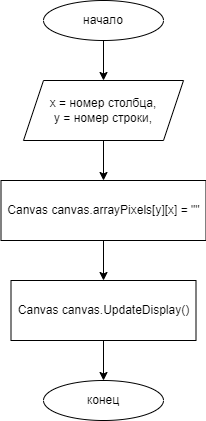


Рисунок 6 – Блок-схема алгоритма работы инструмента «Ластик»

Алгоритм работы инструмента «Пипетка»:

1. Считать номер строки, столбца и цвет, где был произведён клик;
2. Присвоить свойству currentColor объекта Canvas значение цвета, считанного из первого пункта.

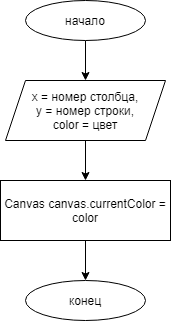


Рисунок 7 – Блок-схема алгоритма работы инструмента «Пипетка»

Алгоритм работы инструмента «Заливка»:

1. Считать номер строки, столбца и цвет, где был произведён клик;
2. Проверить верхнюю, нижнюю, левую и правую клетку на равенство с цветом выбранной клетки;
   1. Если цвета равны, то добавить номер строки и столбца клетки в конец буферного списка.
3. Закрасить текущую клетку цветом, который хранится в свойстве currentColor объекта Canvas;
4. Взять координаты последней клетки из буферного списка, заменить координаты текущей клетки новыми координатами и удалить один элемент из конца списка;
5. Повторять пункты со 2 по 4 пока длина буферного списка не будет равна 0.

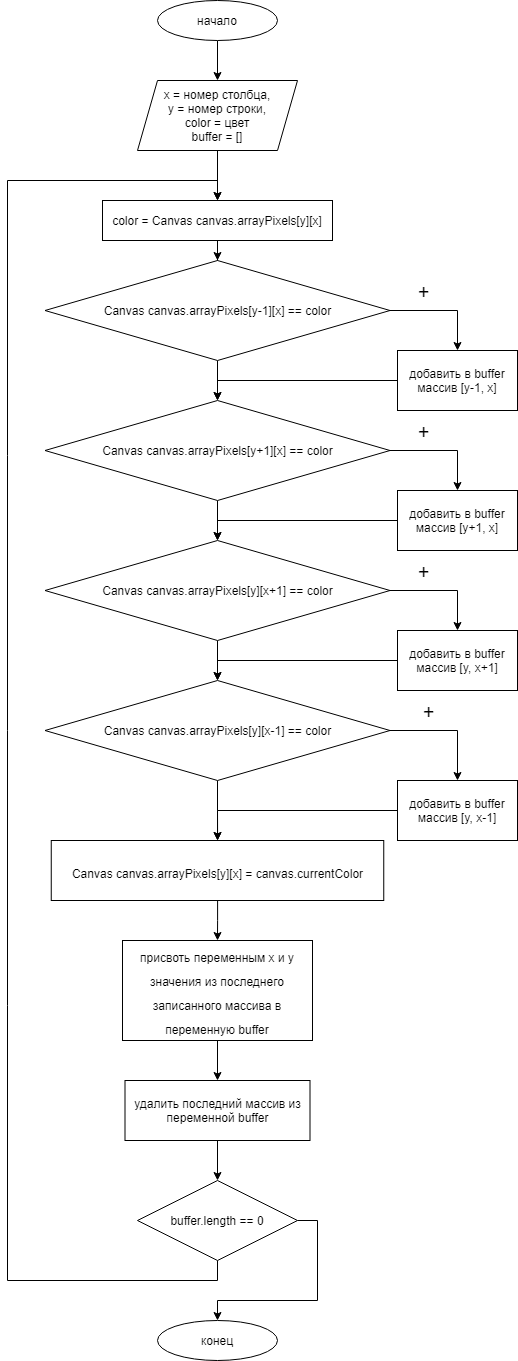


Рисунок 8 – Блок-схема алгоритма работы инструмента «Заливка»

# **4 РЕАЛИЗАЦИЯ ПРОГРАМНОГО ОБЕСПЕЧЕНИЯ**

В процессе реализации программы использовались следующие приложения и веб-сервисы:

* Android Studio;
* Draw.io;
* Genymotion.

Draw.io – онлайн-сервис, позволяющий из готовых объектов составлять диаграммы. С помощью данного сервиса были созданы диаграмма вариантов использования, диаграмма классов, диаграмма состояний, а также блок-схемы.

Android Studio – это интегрированная среда разработки для работы с платформой Android. В этом приложении были реализованы алгоритмы на языке программирования Java, а также проводилась отладка приложения.

Genymotion – эмулятор Android, используемый совместно с Android Studio. В эмуляторе проводилась проверка логики приложения.

# **ЗАКЛЮЧЕНИЕ**

В процессе написания курсового проекта были спроектирован интерфейс и разработано приложение графического редактора пиксельной графики.

В результате выполнения работы были выполнены следующие задачи:

* описана предметная область и сопоставлены аналоги;
* построены модели предметной области;
* спроектирована логика приложения;
* реализована программа.

При достижении поставленной цели, были получены навыки проектирования посредством создания графического редактора пиксельной графики.