**Введение**

Приложение представляет собой интерактивную программу для создания, редактирования и анализа треугольников на двумерной плоскости. Выбранный подход основан на объектно-ориентированном программировании с использованием графического интерфейса пользователя.

Реализованное приложение позволяет пользователям создавать треугольники, перемещать их, поворачивать вокруг выбранной вершины, изменять цвет и определять пересечения между треугольниками. Интерфейс построен с использованием библиотеки tkinter.

**1. Проектная часть**

**1.1. Постановка задачи на разработку приложения**

Определяется заданием на курсовую работу. Детализируется в разработанном техническом задании (приложение 1).

**1.2. Математические методы**

В приложении используются следующие математические методы и модели:

Векторная алгебра: для представления и манипуляции координатами вершин треугольников используется класс Vec2, реализующий базовые операции с двумерными векторами.

Геометрические преобразования: для поворота треугольников применяется матрица поворота, реализованная в методе rotate класса Shape.

Пересечение прямых: для нахождения точек пересечения треугольников используется алгоритм, основанный на решении системы линейных уравнений, реализованный в методе intersection\_lines класса Util.

Выбор этих методов обусловлен их эффективностью и простотой реализации для решения поставленных задач.

**1.3. Архитектура и алгоритмы**

**1.3.1 Архитектура**

Архитектура приложения основана на объектно-ориентированным подходе. Классы для представления геометрических объектов:

* Vec2: представляет двумерный вектор
* Path2D: представляет последовательность точек
* CanvasObject: базовый класс для объектов на холсте
* Shape: представляет геометрическую фигуру
* Triangle: представляет треугольник

Классы для управления графического интерфейса:

* App: основной класс приложения
* Toolbar: панель инструментов
* Sidebar: боковая панель с элементами управления
* RotationControlFrame: элементы управления поворотом
* Классы для управления данными и состоянием:
* AppContext: хранит глобальное состояние приложения
* TriangleManager: управляет коллекцией треугольников

Вспомогательные классы:

* Util: содержит вспомогательные методы для геометрических вычислений

**1.3.2 Алгоритм поворота треугольника**

1. Начало

2. Получить угол поворота в радианах

3. Получить координаты точки вращения (выбранной вершины)

4. Для каждой вершины треугольника:

4.1. Если вершина совпадает с точкой вращения, пропустить

4.2. Вычислить разницу координат между вершиной и точкой вращения

4.3. Применить матрицу поворота: x' = x \* cos(угол) - y \* sin(угол) y' = x \* sin(угол) + y \* cos(угол)

4.4. Обновить координаты вершины

5. Обновить координаты треугольника на холсте

6. Конец

**1.3.3 Алгоритм определения пересечения треугольников**

1. Начало

2. Получить выбранный треугольник

3. Для каждого другого треугольника на холсте:

3.1. Для каждой стороны выбранного треугольника:

3.1.1. Для каждой стороны текущего треугольника:

3.1.1.1. Вычислить точку пересечения отрезков

3.1.1.2. Если точка пересечения существует, добавить её в список

4. Удалить все предыдущие точки пересечения с холста

5. Отобразить новые точки пересечения на холсте

6. Конец

**1.4. Тестирование**

**1.4.1 Описание отчета о тестировании**

Отчет о тестировании предназначен для оценки функциональности и надежности приложения. Он содержит описание проведенных тестов, их результаты и рекомендации по улучшению приложения.

**1.4.2 Цель тестирования**

Целями тестирования являются:

* Проверка корректности работы основных функций приложения
* Выявление возможных ошибок и несоответствий техническому заданию
* Оценка удобства использования пользовательского интерфейса
* Проверка стабильности работы приложения при различных сценариях использования

**1.4.3 Методика тестирования**

Для тестирования приложения используются следующие методы:

* Функциональное тестирование: проверка соответствия функциональности требованиям технического задания
* Тестирование пользовательского интерфейса: проверка удобства использования и отзывчивости интерфейса

**1.4.4 Проведенные тесты**

1. Создание треугольника

* Ожидаемый результат: Треугольник появляется на холсте в указанной точке
* Фактический результат: Соответствует ожидаемому

2. Перемещение треугольника

* Ожидаемый результат: Треугольник перемещается за курсором мыши
* Фактический результат: Соответствует ожидаемому

3. Поворот треугольника

* Ожидаемый результат: Треугольник поворачивается вокруг выбранной вершины на заданный угол
* Фактический результат: Соответствует ожидаемому

4. Изменение цвета треугольника

* Ожидаемый результат: Цвет заливки треугольника меняется на выбранный
* Фактический результат: Соответствует ожидаемому

5. Определение пересечений

* Ожидаемый результат: Точки пересечения треугольников отображаются на холсте
* Фактический результат: Соответствует ожидаемому

6. Сохранение и загрузка данных

* Ожидаемый результат: Данные о треугольниках корректно сохраняются в файл и загружаются из него
* Фактический результат: Соответствует ожидаемому

**1.4.5 Выводы**

Основная функциональность приложения работает корректно и соответствует техническому заданию. Пользовательский интерфейс интуитивно понятен и отзывчив. Приложение стабильно работает при различных сценариях использования.

Рекомендации по улучшению:

Добавить примитивные геометрические фигуры. Добавить возможность изменения размеров фигуры. Реализовать возможность отмены и повтора действий пользователя. Добавить палитру основных цветов в панель инструментов. Добавить возможность сохранения холста в графический файл.