**Лабораторная работа 5: Алгоритмы отсечения отрезков и многоугольников**

***Цель работы:***

Целью данной лабораторной работы является разработка приложения, иллюстрирующего работу алгоритмов отсечения отрезков и многоугольников. В качестве языка программирования используется C++ с использованием фреймворка Qt (версия 10.0.0 Community).

***Реализованные алгоритмы:***

***-*** Алгоритм Сазерленда-Коэна (через явное задание прямой, содержащей отрезок)

***-*** Алгоритм отсечения выпуклого многоугольника

***Исходные данные***

Программа принимает входные данные из текстового файла

***Требования:***

1. Вывести систему координат в соответствующем масштабе.

2. Отобразить отсекающее окно одним цветом, исходные отрезки (многоугольники) – другим цветом.

3. Выполнить отсечение соответствующими алгоритмами.

4. Визуализировать видимые части отрезков (многоугольников).

***Реализация:***

*Главное окно (MainWindow)*

Главное окно содержит элементы управления для выбора папки с изображениями и отображения результатов. Реализованы функции отрисовки системы координат, отсекающего окна и результатов отсечения.

*Метод `paintEvent`*

Отрисовывает систему координат, отсекающее окно, исходные отрезки и результаты отсечения.

*Метод `cutSegments`*

Выполняет отсечение отрезков с использованием алгоритма Коэна-Сазерленда.

*Метод `cohenSutherlandClip`*

Выполняет отсечение отдельного отрезка методом Коэна-Сазерленда.

*Метод `getCode`*

Определяет код точки относительно отсекающего окна.

*Метод `cutPolygon`*

Выполняет отсечение многоугольника методом Сазерленда-Ходжмана.

*Метод `intersectionPoint`*

Определяет точку пересечения отрезка с границей отсекающего окна.

***Более подроная реализация алгоритмов:***

**1) Алгоритм Сазерленда-Коэна (Cohen-Sutherland):**

*Метод `cutSegments`:*

Этот метод принимает вектор отрезков и выполняет их отсечение с использованием алгоритма Коэна-Сазерленда. Важно отметить следующие моменты:

1. Перед началом отсечения каждого отрезка, метод определяет коды конечных точек отрезка с использованием метода `cohenSutherlandClip`.

2. Если обе конечные точки отрезка видимы, то отрезок полностью видим и добавляется в `clipSegments[0]`.

3. Если обе конечные точки отрезка невидимы, то отрезок полностью отсекается и не добавляется в результат.

4. Если одна из конечных точек невидима, то метод определяет точку пересечения с окном и добавляет два новых отрезка в соответствующие массивы `clipSegments[0]` и `clipSegments[1]`.

*Метод `cohenSutherlandClip`:*

Этот метод принимает две конечные точки отрезка (`P1` и `P2`) и определяет их коды относительно окна отсечения. Далее, основываясь на кодах, метод решает, каким образом отсечь отрезок или определить точку пересечения с окном.

**2) Алгоритм отсечения выпуклого многоугольника:**

*Метод `cutPolygon`:*

Этот метод принимает выпуклый многоугольник и выполняет его отсечение методом Сазерленда-Ходжмана. Ключевые моменты:

1. Перебор каждой грани многоугольника и проверка её видимости.

2. Использование метода `intersectionPoint` для определения точек пересечения с гранями окна отсечения.

3. Построение нового выпуклого многоугольника, содержащего только видимые части исходного многоугольника.

*Метод `intersectionPoint`:*

Этот метод определяет точку пересечения грани многоугольника с границей окна отсечения.

***Интерфейс:***

Интерфейс приложения разработан с использованием фреймворка Qt, включает в себя элементы управления для выбора папки, настройки масштаба и визуализации результатов.

***Заключение:***

В ходе выполнения лабораторной работы было разработано приложение, иллюстрирующее работу алгоритмов отсечения отрезков и многоугольников. Также лабораторная работа по разработке приложения, иллюстрирующего алгоритмы отсечения отрезков и многоугольников, позволила освоить применение алгоритма Сазерленда-Коэна для отсечения отрезков и алгоритма Сазерленда-Ходжмана для отсечения выпуклых многоугольников. Полученные навыки программирования на C++ с использованием фреймворка Qt 10.0.0 Community оказываются полезными в областях, где требуется визуализация и обработка графических данных.

Эти навыки могут быть применены, например, в разработке графических редакторов, CAD-систем, визуализации данных в научных и инженерных областях. В индустрии разработки игр, где важна работа с графикой, также могут быть использованы алгоритмы отсечения для создания реалистичных и оптимизированных сцен. Таким образом, полученные навыки открывают двери для применения знаний в различных областях, где требуется обработка и визуализация графических данных.

Приложение способствует более глубокому пониманию и применению алгоритмов в области компьютерной графики.