РАЗРАБОТКА АЛГОРИТМА ДЛЯ ИНТЕРАКТИВНОГО ПРИЛОЖЕНИЯ ПО ИЗУЧЕНИЮ RAID-МАССИВОВ

Бараболя Илья Олегович, студент, e-mail: ahmddmamdd@gmail.com
Исакович Егор Олегович, студент, e-mail: qew1983@bk.ru
Прудников Василий Михайлович, старший преподаватель кафедры «Программное обеспечение информационных технологий», e-mail: prv58@bk.ru
Межгосударственное образовательное учреждение высшего образования «Белорусско-Российский университет» г. Могилёв, Республика Беларусь

Описана разработка алгоритма для интерактивного приложения для изучения RAID-массивов, с акцентом на визуализации логических и физических топологий различных уровней RAID. Важным аспектом визуализации является корректное отображение связей между накопителями данных и контроллером. Реализация этой задачи основана на математическом алгоритме, который определяет четыре точки и соединяет их прямыми линиями, образуя полилинию. Для разработки используется язык программирования C++ и мультимедийная библиотека SFML.

Ключевые слова: *алгоритм визуализации, интерактивное приложение, наглядность, алгоритм, визуализация линий связи.*

Разрабатывается интерактивное приложение по изучению RAID-массивов [1]. Для визуализации логических и физических топологий различных уровней RAID необходимо обеспечить наглядность процессов. В результате вопрос визуализации линий связи между накопителями данных и контроллером является важным для понимания процессов, происходящих в RAID-массивах [2].

В области компьютерной графики соединение линией двух объектов является обычной задачей. Это включает в себя рисование линии, соединяющей два объекта, обычно для обозначения какой-то связи или взаимодействия между ними.

Реализация алгоритма визуализации выполнена на языке программирования C++ с использованием мультимедийной библиотеки SFML [3]. По своей сути, реализация соединения объектов – это математическая задача, связанная с нахождением точек на двумерной поверхности. Формируемая линия должна состоять из трех перпендикулярных сегментов и не являться наклонной. Это исходные данные для дальнейшей работы.

Создано два объекта – прямоугольники разных размеров, выступающие в роли контроллера RAID и одного из жестких дисков. Из-за того, что реализуется именно алгоритмическая часть соединения объектов, абсолютно не важно какие объекты соединяются, важен лишь факт присутствия визуальной линии связи между ними.

Необходимо определить четыре точки и провести между ними прямые линии. Соединенные линии дают следующую фигуру – полилинию – геометрический объект в математике, состоящий из последовательности соединенных прямых отрезков (рисунок 1).

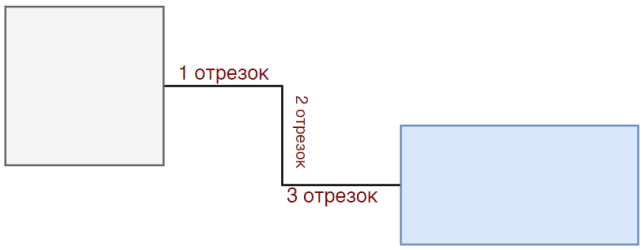

Рисунок 1 – Отрезки, соединяющие два объекта

Задача сводится к тому, чтобы отрисовать каждый отрезок отдельно.

Представим, что между объектами присутствует воображаемый прямоугольник (рисунок 2). Необходимо разделить этот прямоугольник пополам и провести через его центр прямые линии.

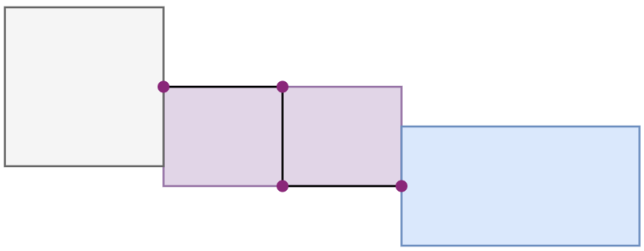

Рисунок 2 – Воображаемый прямоугольник

С целью решения задачи создана функция draw_lines.

Алгоритм начинается с определения объекта sf::VertexArray - это массив вершин, которые используются для определения графических примитивов, таких как линии, треугольники и многоугольники. В данном случае 'line' — это полилиния, содержащая четыре вершины.

Первая вершина (рисунок 3) устанавливается в центре первого прямоугольника плюс половина его ширины и с той же координатой y равной координате Y точки центра первого прямоугольника. Эта вершина представляет собой начальную точку полилинии.

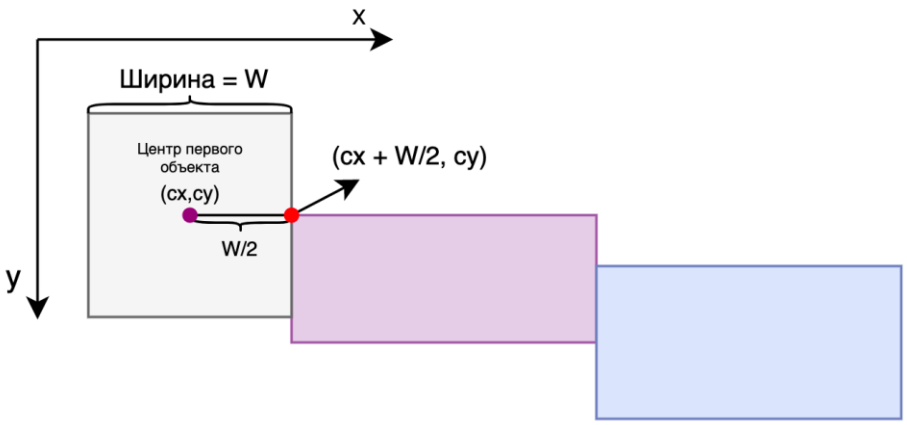

Рисунок 3 – Первая точка (красного цвета). Имеет координаты (cx + W/2, cy)

Вторая вершина (рисунок 4) устанавливается на полпути между правым краем первого прямоугольника и левым краем второго прямоугольника, с той же координатой Y, что и центральная координата Y первого прямоугольника.

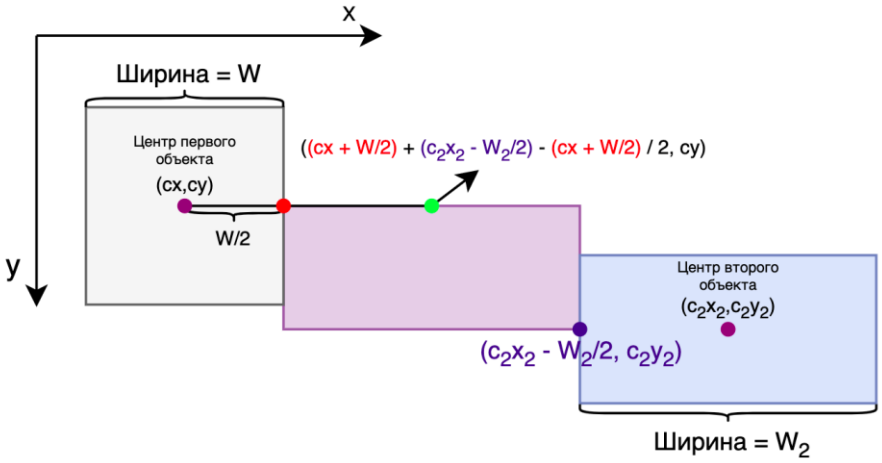

Рисунок 4 – Вторая точка (зеленая)

Эта вершина представляет собой точку, где полилиния меняет направление с горизонтального на вертикальное. ((rect2Center.x - rect2.getSize().x / 2.f) - (rect1Center.x + rect1.getSize().x / 2.f)) / 2 вычисляет половину горизонтального расстояния между правым краем первого прямоугольника (cx + W/2, cy) и левого края второго прямоугольника (c_2x_2 –

$W_2/2$, c_2y_2). В случае второго прямоугольника необходимо вычесть от центра половину ширины объекта, так как ось X направлена вправо. Это расстояние вычитается из координаты x центральной точки первого прямоугольника, чтобы получить координату x средней точки между двумя прямоугольниками.

Сложение этих двух значений вместе дает координату X второй вершины в полилинии.

Линия, середину которой необходимо найти, является прямой, параллельной оси абсцисс. Соответственно, у нее изменяется лишь одна координата – X. Мы знаем ее координату Y – это координата Y середины первого объекта (cy). Отсюда получаем, что координаты точки правого верхнего угла воображаемого прямоугольника равны ($c_2x_2 – W_2/2$, *cy*).

Третья вершина (рисунок 5) устанавливается на той же координате X, что и вторая вершина (зеленая), но изменяется координата y, тем самым сдвигая данную точку на уровень центра второго прямоугольника. Эта вершина представляет собой точку, в которой полоса линии меняет направление с вертикального на горизонтальное.

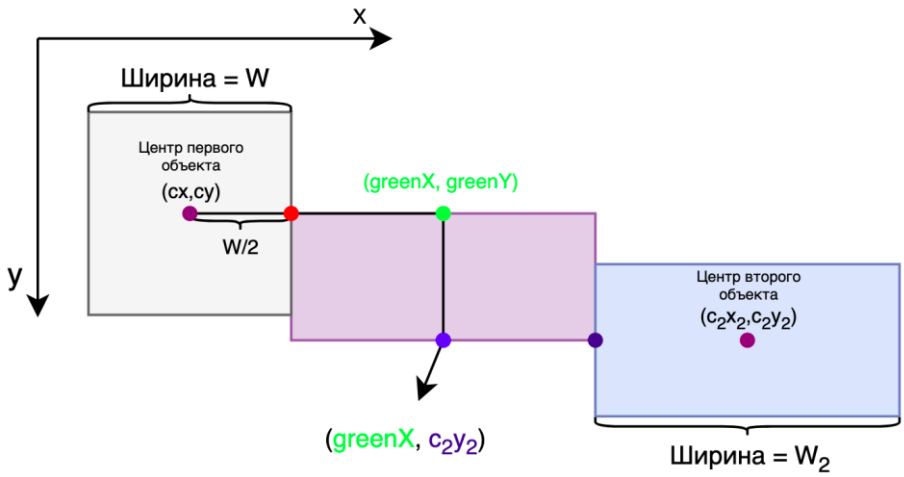

Рисунок 5 – Третья точка (синяя)

Четвертая вершина (рисунок 6) устанавливается на левом краю второго прямоугольника, на той же координате y, что и вторая и третья вершины. Эта вершина представляет собой конечную точку полилинии.

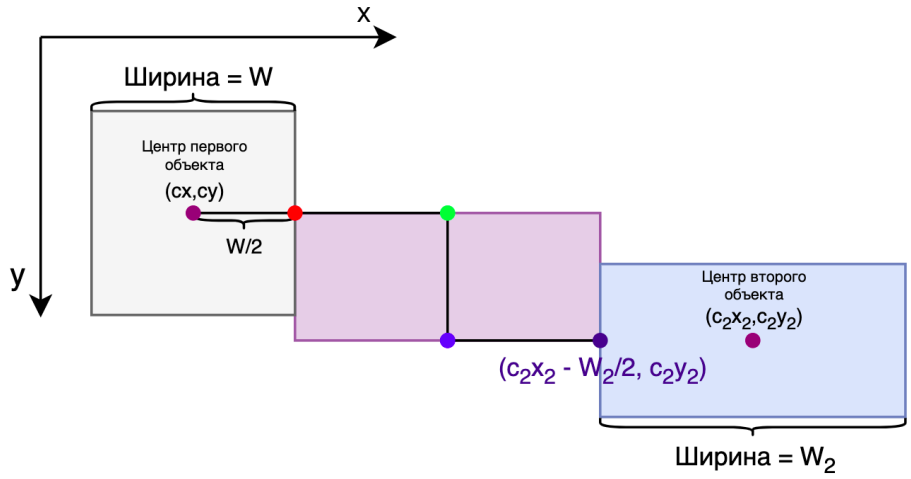

Рисунок 6 – Последняя точка (темно-синяя)

После выполнения данных операций получаем линию, которая визуализирует связь между двумя объектами. Результат можно увидеть на рисунке 7.

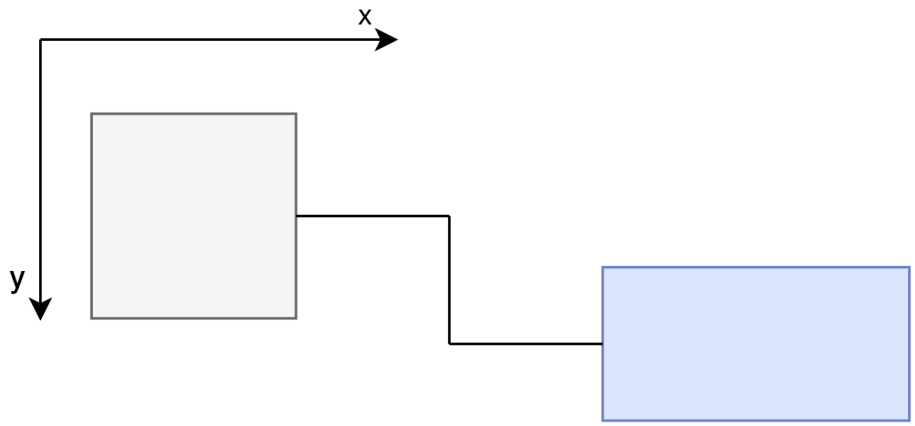

Рисунок 7 – Результат работы алгоритма

В данной статье описан алгоритм, который находит точки на поверхности с целью отрисовки полилинии, которая визуально представляет связь между объектами. Результаты работы позволяют эффективно определить необходимые точки и последовательность соединенных прямых отрезков, составляющих полилинию. Такой подход предоставляет возможность более наглядно и понятно визуализировать связи и взаимодействия между объектами, что может быть полезным в различных областях, включая компьютерную графику, архитектуру и дизайн. Дальнейшие исследования и развитие этого алгоритма могут привести к улучшению методов отображения связей и повышению качества визуальной интерпретации данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Иванов, П.С. Введение в RAID: Основы и техники / П.С. Иванов.– Москва : Издательство "Техническая литература", 2018.

2. Сидоров, А.В. Визуализация в компьютерной графике: Принципы и приложения / А.В. Сидоров.– Москва : Издательство "Наука и образование", 2019.

3. Браун, М. Реализация графических алгоритмов с использованием SFML на языке C++ / М .Браун.– Себастопол : Издательство "О'Райли Медиа", 2021.