

На теореме 1 основан алгоритм нахождения кратчайшего биссектора выпуклого многоугольника:

1. Найти длины всех существующих правильных биссекторов.
2. Выбрать из них кратчайший (минимальную длину

может иметь несколько биссекторов).

Отдельно остановимся на случае треугольника.

**Теорема 2.** Если меньший угол треугольника площади  $S$  равен  $\alpha$ , то кратчайшим биссектором будет являться правильный биссектор с концами на сторонах, образующих этот угол, длиной

$\sqrt{Sa}$  и радиусом  $\sqrt{\frac{S}{\alpha}}$  (рис. 7).

Теорема 2 является следствием леммы 1 и теоремы 1. Отметим, что кратчайших биссекторов в треугольнике может быть один, два или три – по количеству углов с минимальным значением. Например, в равностороннем треугольнике три кратчайших биссектора (рис. 8).

**Заключение.** Недостатком представленного алгоритма является перебор большого количества правильных биссекторов: у  $n$ -угольника  $(n^2-n)/2$  пар сторон. Кроме того, расчет длины биссектора, стягивающего не соседние стороны, требует нахождения угла между содержащими эти стороны прямыми, а также площади фигуры, ограниченной этим углом и многоугольником. Это обуславливает значительную вычислительную сложность алгоритма.

1. Бляшке, В. Круг и шар / В. Бляшке – М. : Наука, 1967. – 233 с.
2. Пойа, Д. Математика и правдоподобные рассуждения / Д. Пойа. – 2-е изд., испр. – М. : Наука, 1975. – 462 с.

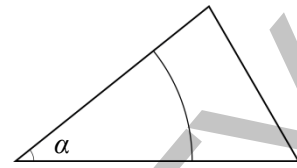


Рис. 7

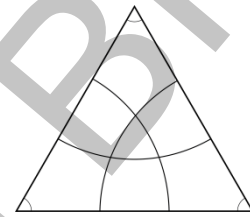


Рис. 8

## ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ВЫСОКОПРОИЗВОДИТЕЛЬНОГО МНОГОПОЛЬЗОВАТЕЛЬСКОГО ГРАФИЧЕСКОГО ПРИЛОЖЕНИЯ

**Куликов В.А.,**

выпускник ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь  
Научный руководитель – **Шпаков С.А.,** старший преподаватель

Зачастую среди небольших групп разработчиков в корпоративной среде возникает необходимость донести некоторую идею. При этом, учитывая реалии современного мира, всё больше людей переходят на удалённый режим работы. Таким образом, возникает вопрос: а как же в таких ситуациях донести свою идею наиболее эффективно и рационально? Одним из возможных решений этой проблемы может служить создание высокопроизводительного многопользовательского приложения, работающего с графикой и сетью. Такое приложение позволило бы визуально доносить свои задумки и идеи до других сотрудников, при этом позволяя им самим поучаствовать в её формировании.

Целью работы является проектирование и разработка сетевого многопользовательского векторного [1] графического редактора, создание которого позволит осуществлять визуальную коммуникацию в удалённом режиме, в том числе и посредством всемирной паутины (Internet). Для ускорения работы приложения в качестве рабочего был выбран язык программирования Rust [2] с использованием ключевой библиотеки Vulkan [3], которая позволяет взаимодействовать с API Vulkan.

**Материал и методы.** Для реализации данного приложения были использованы средства ОС Linux, текстовый редактор Neovim, который оснащён плагином для работы с LSP (Language Server Protocol) сервером. Проектирование и разработка осуществлялась на стеке достаточно новых и мало опробованных технологий – языке программирования Rust и графического API Vulkan.

**Результаты и их обсуждение.** Разработанный редактор помогает осуществлять визуальную коммуникацию по сети в режиме реального времени. Сеть может быть как локальной (корпоративной), так и глобальной (Internet).

Основной средой для реализации является, в первую очередь, язык программирования Rust [2]. Множество мощных абстракций, нацеленность на безопасность, а также тщательно продуманный дизайн языка, позволяют реализовывать достаточно сложные проекты, при этом, не теряя в производительности.

Для ускорения работы приложения используется Vulkan – открытый API. Несмотря на то, что Vulkan ещё не везде адаптирован, этот процесс продолжает нарастать. На мобильных устройствах ожидается снижение энергопотребления за счёт переноса части вычислений с процессора на видеокарту, путём использования Vulkan [3].

Таким образом, Vulkan и Rust отлично работают вместе, поскольку обе эти технологии направлены в первую очередь на производительность и скорость работы приложения.

В проекте, для реализации сетевой архитектуры, была выбрана двуранговая модель, то есть один из участников отличается от других, выступая в роли сервера. Остальные участники работают как клиенты. Схематически, данную модель можно представить как на рисунке 1:

Приложение по своей природе состоит из вечного цикла, каждая итерация которого соответствует обновлению поверхности.

Одним из важнейших примитивов векторного графического редактора является ломаная. При реализации ломаной возникают проблемы, в частности, с отображением её углов. Эта проблема решается комбинацией двух визуализаций с целью достижения наиболее приемлемого визуального эффекта. Инфраструктура передачи и приёма сообщений использует набор потоков, взаимодействующих между собой с помощью каналов.

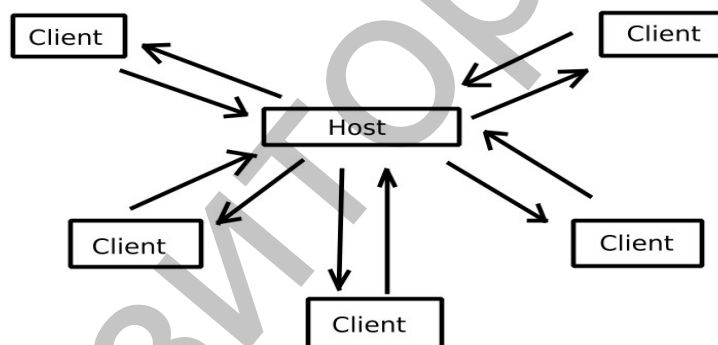


Рисунок 1 – Клиент-серверная архитектура графического приложения

Для основных частей практической реализации были использованы стейты и шейдеры. Стейты отвечают за основную полезную нагрузку приложения – векторный редактор графики, а также для управления состоянием приложения. Приложение меняет своё состояние под воздействием событий, которые реализованы как брокеры, то есть объекты, хранящие события для обработки их в некоторый более поздний момент времени. Шейдеры позволяют исполнять код на GPU, тем самым визуализируя состояние программы для пользователя.

Важный аспект работы современного приложения – это его кроссплатформенность. Несмотря на то, Rust – это компилируемый язык, библиотеки, которые используются в данном приложении, выбраны таким образом, чтобы поддерживать основные платформы, на которых это приложение необходимо запускать.

Взяв в расчёт всё вышеописанное, а также поддержку Vulkan, приложение отлично работает и на ОС Linux, и на ОС Windows.

**Заключение.** Использование данного приложения позволит свести к минимуму проблемы, в частности проблему визуальной коммуникации между разработчиками, если они находятся на удалённой работе. Технологии, с помощью которых реализовано приложение, обеспечивают его стабильность и высокую скорость работы. Разработка внедрена в производственный процесс ООО «Фабрика инноваций и решений».

1. Векторная графика [Электронный ресурс]. Режим доступа: [https://en.wikipedia.org/wiki/Vector\\_graphics](https://en.wikipedia.org/wiki/Vector_graphics) – Дата доступа: 20.08.2020.

2. The Rust Programming Language - [Электронный ресурс]. Режим доступа: <https://doc.rust-lang.org/book/> – Дата доступа: 20.08.2020.

3. Документация Vulkano [Электронный ресурс]. Режим доступа: <https://docs.rs/vulkano/0.17.0/vulkano/> – Дата доступа: 20.08.2020.

## ОСОБЕННОСТИ ОПРЕДЕЛЕНИЯ ТОЧЕК СТАРТА (ПАДЕНИЯ) МИН И АРТИЛЛЕРИЙСКИХ СНАРЯДОВ

*Леонович А.С., Свинарский М.В.,*

*адъюнкт, инженер учебной лаборатории кафедры автоматики, радиолокации и прямо-передающих устройств УО «ВА РБ», г. Минск, Республика Беларусь*

*Научный руководитель – Ярмолик С.Н., канд. техн. наук, доцент*

Анализ локальных вооруженных конфликтов, для которых характерна мобильная тактика ведения боя, подтверждает широкое применение противником артиллерийского огня. В условиях современного боя важно преждевременно подавить огневые позиции противника за максимально короткое время при условии обеспечения их оперативного вскрытия и точной корректировки собственного артиллерийского и минометного огня. Наиболее распространенным источником информации о воздушной обстановке в зоне боевых действий являются радиолокационные средства разведки [1, с. 53]. Современные образцы радиолокаторов артиллерийской разведки способны решать задачи определения координат огневых позиций противника, мест падения снарядов, а также корректировать огонь своих огневых средств. Качество функционирования средств разведки определяется эффективностью способов вычисления местоположения огневых позиций противника, а также корректировки собственного артиллерийского огня [1, с. 54].

Принцип работы РЛС КББ основывается на засечке артиллерийских средств поражения на участке траектории полета и проведении нескольких измерений текущего положения боеприпаса с целью расчета его координат и параметров движения [1, с. 56]. На основе экстраполяции точек траектории сопровождаемого снаряда осуществляется определение его возможного места падения, а на основе пролонгации точек траектории снаряда осуществляется возможное определение местоположения огневых позиций противника (рисунок 1).

Для засечки огневых позиций артиллерии противника луч диаграммы направленности РЛС сканирует пространство над линией горизонта, образуя барьер обнаружения. При получении отраженного сигнала станция сопровождает цель в течение времени, ограниченного временем нахождения объекта в анализируемом секторе обзора. После выхода наблюдаемого объекта из сектора обзора станции происходит процедура вычисления координат огневой позиции и места падения боеприпаса [1 С. 56].

В докладе рассмотрены особенности определения точек старта (падения) мин и артиллерийских снарядов для решения задач корректировки огня своей артиллерии и обнаружения огневых позиций противника.

**Материал и методы.** Значения ошибок определения точек старта (падения) являются случайной величиной, распределенной по гауссовскому закону распределения. Одной из основных величин, характеризующих нормальный закон распределения значений ошибок определения точек старта (падения) является среднеквадратическое отклонение (СКО). Методом статистических испытаний рассчитана среднеквадратическая ошибка определения точки старта (падения) анализируемого объекта. В качестве устройства вторичной обработки использовался совместный фильтр, основанный на методе наименьших квадратов [2, с. 71].