

Гавриков А.П.

Научный руководитель: к.т.н., ведущий электроник каф. ЭиВТ Д.В. Бейлекчи Муромский институт (филиал) федерального государственного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых» 602264, г. Муром, Владимирская обл., ул. Орловская, 23 E-mail: alexgavrikov21022003@gmail.com

Исследование графических технологий для разработки пользовательского интерфейса программного обеспечения

Данное исследование проводилось в рамках курсового проектирования, в ходе которого была поставлена цель – сравнить и выбрать технологию для отображения графического интерфейса пользователя с кроссплатформенной поддержкой без использования стандартной подсистемы отображения элементов интерфейса в операционных системах Windows и Linux. Сравнение проводилось среди ряда современных графических технологий. Данные технологии представлены в рис. 1 и были разделены на три уровня по критерию взаимодействия между программой, ОС и аппаратурой.

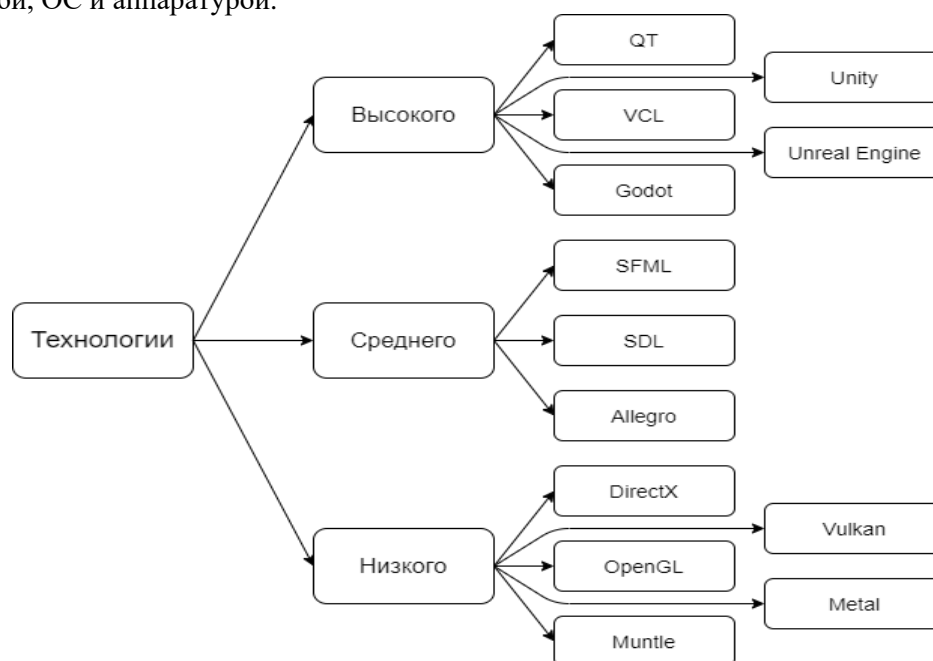


Рис. 1 – Классификация графических технологий для разработки мультимедиа приложений.

В проекте были определены следующие требования к графической технологии: поддержка для разработки приложений языка C++, удобный API для разработки небольшого приложения и кроссплатформенность между различными платформами, такими как Windows, Linux, Mac OS. При исследовании всех трёх уровней были выявлены как их недостатки, так и их преимущества.

Преимуществом технологии высокого уровня являются возможность быстрого написания приложения, наличие визуального редактора и готовые реализации различных алгоритмов и структур данных. Недостатками являются сложные взаимосвязи между компонентами технологии, большой размер программ и библиотек для работы с технологиями этого уровня, и привязка, в основном, к одному языку программирования.

У технологий низкого уровня преимуществами являются возможность точного контроля над выводом графической информации и минимальными зависимостями от других технологий. Недостатками являются: необходимость создания большого объёма кода, медленная скорость разработки из-за необходимости изучения и учета аппаратных и системных особенностей

работы и, соответственно, высокая сложность в использовании для разработки небольших приложений.

Технологии среднего уровня более просты, чем технологии низкого уровня, но не предоставляют средств для разработки интерфейса, в отличие от технологии высокого уровня. К преимуществам можно отнести малый размер, более высокую скорость разработки по сравнению с низким уровнем и возможность писать на разных языках программирования. Недостатками среднего уровня являются отсутствие каких-либо редакторов и готовых графических компонентов для разработки.

Технология SDL [1] представляет собой мультимедийную библиотеку, реализующую единый API к графической подсистеме, звуковому устройству и средствам ввода для широкого спектра платформ. Основная часть SDL обеспечивает ограниченный спектр возможностей, который расширяется путём библиотек расширений, которые обычно входят в поставку SDL. Обычно используется для работы с окном и устройствами ввода, а работу с графикой осуществляют с помощью OpenGL. Сама библиотека написана на C, но есть привязки к другим языкам программирования, например, C++, C#, Java.

Технология SFML [2] является объектно-ориентированным аналогом SDL. Разбит на пять модулей, такие как System (основной модуль), Window (управление окнами и взаимодействие с пользователями), Graphics (отображение графических примитивов и изображений), Audio (интерфейс для управлением звуком), Network (для сетевого взаимодействия). Перечисленные модули, кроме System, необязательно подключать к проекту программы. Основным языком программирования для работы с библиотекой является C++, но также есть привязки к другим языкам, например, C#, Java, Go, Rust.

Технология Allegro [3] представляет собой мультимедийную библиотеку, включающую инструменты для работы с 2D графикой и текстом, работы с изображениями, воспроизведения аудиофайлов и обработки пользовательского ввода. Библиотека написана на C и рассчитана на использование с языками C и C++, но возможна связка с Pascal, Python, Lua, Go.

Каждая технология предоставляет способ взаимодействия с C++, но лишь библиотеки SFML написаны для использования с объектно-ориентированной парадигмой. Среди трех рассматриваемых технологий среднего уровня SDL предоставляет наименее обширный API для работы с графикой в сравнении с остальными. Работа с текстом, окном и звуком присутствует во всех библиотеках на достаточном уровне для реализации проекта. Все технологии обеспечивают как динамическое, так и статическое связывание с исполняемым файлом. Лишь в SFML можно отключить неиспользуемые модули и из-за этого уменьшить итоговый объем программы. Для каждой из библиотек есть документация, уроки и примеры программ. Кроссплатформенность технологий показана в таблице 1.

Таблица 1 – Кроссплатформенность технологий

Технология	Linux	Windows	Mac OS	Android	iOS
SDL	+	+	+	+	+
SFML	+	+	+	-	-
Allegro	+	+	+	+	+

Таким образом, для реализации проекта оптимально использование библиотек SFML, так как благодаря модульному и ООП подходу, полной документации на официальном сайте, примерам на GitHub и различным урокам данная технология удобна для быстрого освоения и применения. Другие технологии можно порекомендовать, если имеются дополнительные требования, например, если нужно обеспечить вывод графики через OpenGL, то для работы с окном и устройствами ввода можно использовать SDL.

Литература

1. Описание технологии SDL [Электронный ресурс] Режим доступа: <https://www.libsdl.org>.
2. Описание технологии SFML [Электронный ресурс] Режим доступа: <https://www.sfm-dev.org>.
3. Описание технологии Allegro [Электронный ресурс] Режим доступа: <https://liballeg.org>.