

Разработка библиотеки визуализации трехмерной графики для симулятора дронов

Цырендашиев Сультим Баиржапович, 371

Научный руководитель: ст. преп. Пименов А.А.

Введение

- Гражданские беспилотные летательные аппараты, их тестирование
- CoreCVS
- Необходимо повышение переносимости и удобства использования визуализации трехмерной графики

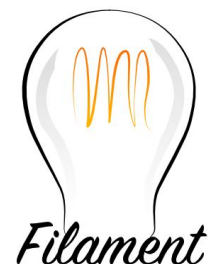
Цель и постановка задачи

Цель курсовой работы – разработка графической библиотеки с высокой конфигурируемостью и простотой использования.

- Реализация обертки над графическим API
- Реализация высокоуровневой системы визуализации трехмерной графики
- Интеграция в подсистему отображения инженерной графики библиотеки CoreCVS

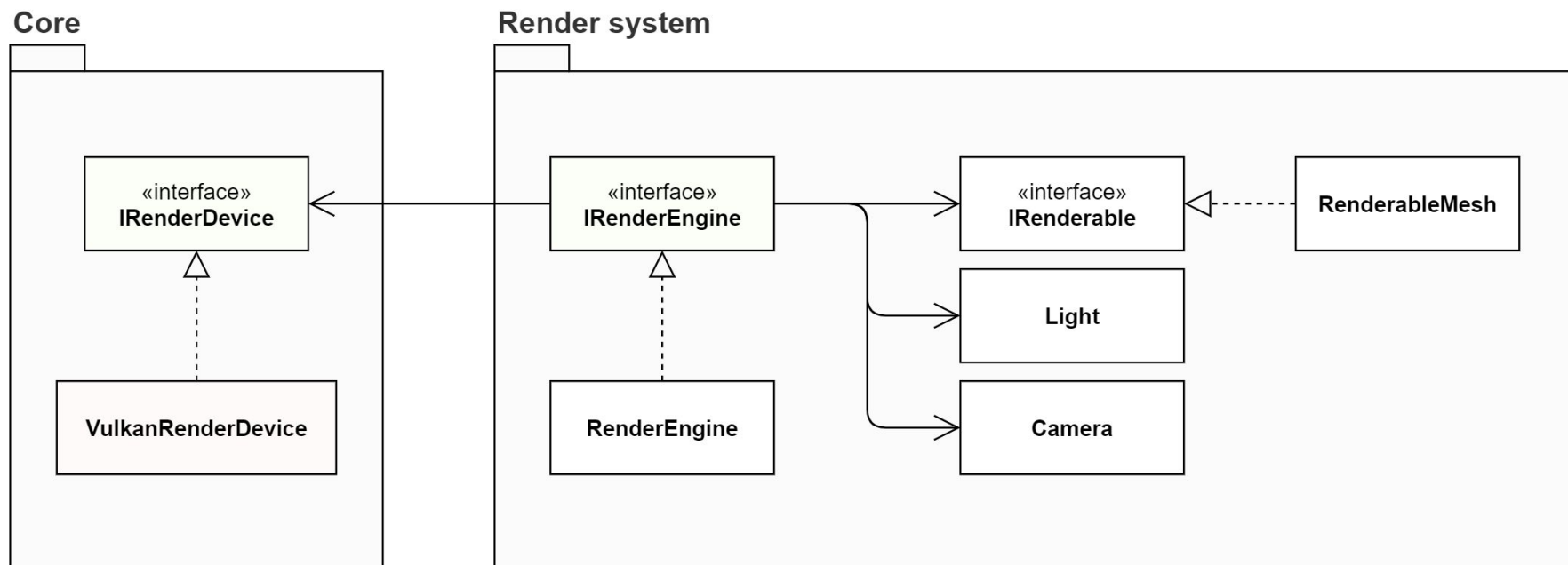
Существующие решения

- Google Filament
 - произвольные шейдеры?
 - произвольный формат вершин?
- Object-oriented Graphics Rendering Engine
 - множество зависимостей
 - сложность интеграции с Qt
- Симуляторы AirSim, Carla — Unreal Engine, Unity Engine

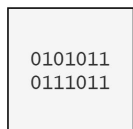


OGRE

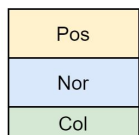
Обзор



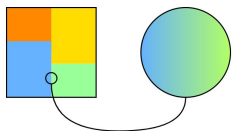
Реализация обёртки



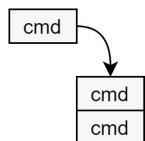
- Вершинные, индексные буферы



- Регистрация структур вершин

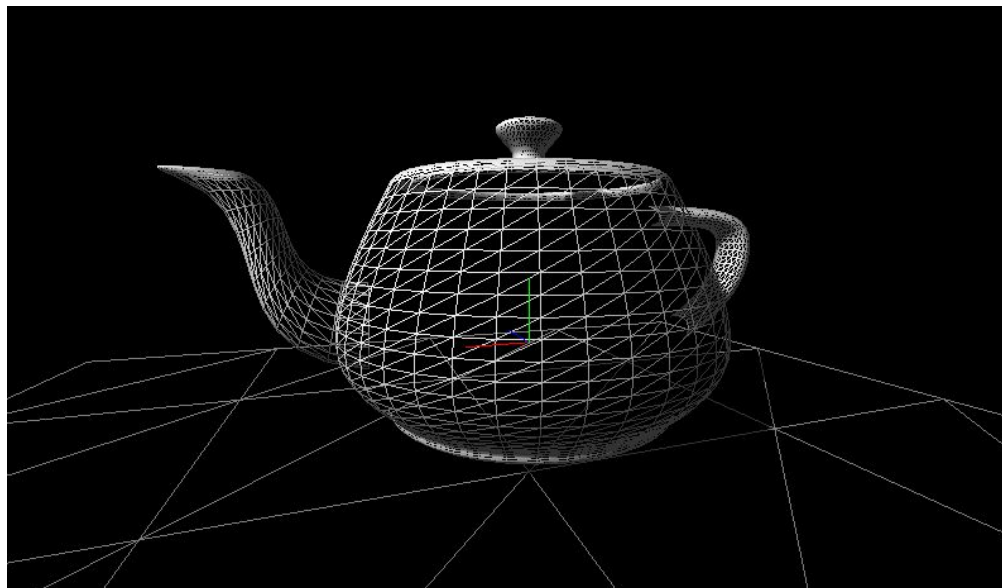


- Текстуры, пробники



- Работа с командными буферами

Вершинные, индексные буферы



Полигональные сетки

Динамические

Статические

Структуры вершин

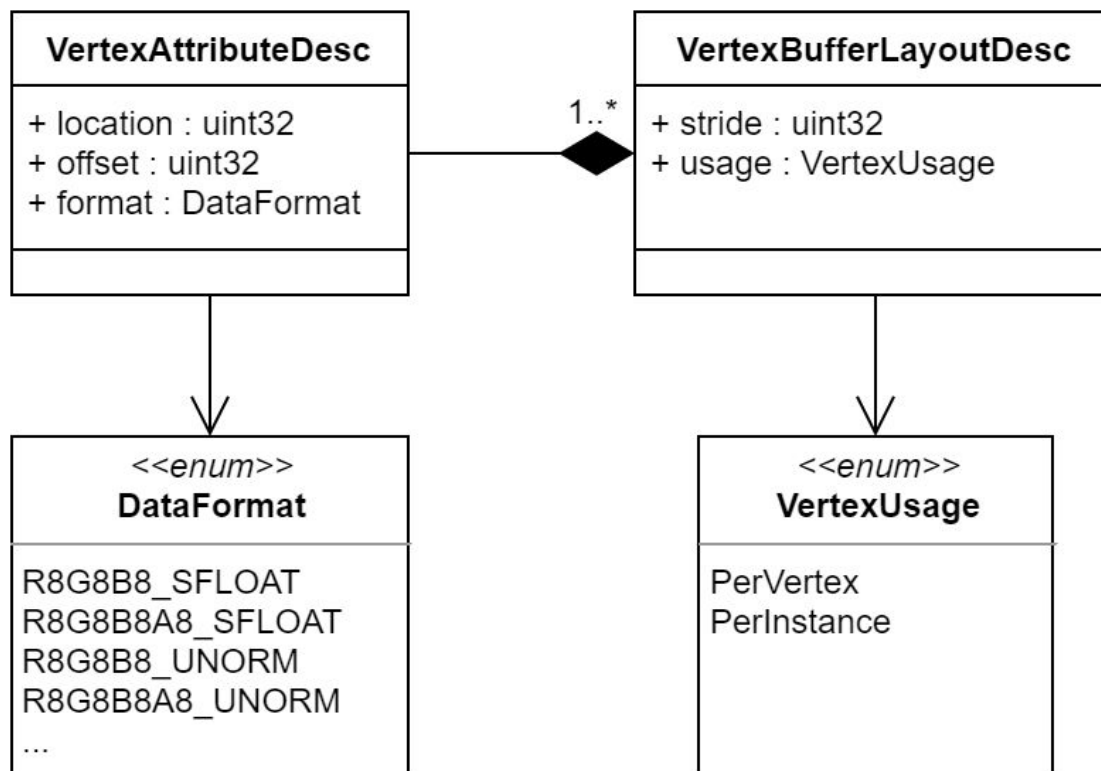
Полигональная сетка 1

Вершина 1		Вершина 2	
Позиция 3 float	Не используется 5 float	Позиция 3 float	Не используется 5 float

Полигональная сетка 2

Вершина 1			Вершина 2		
Позиция 3 float	Нормаль 3 float	Цвет 4 float	Позиция 3 float	Нормаль 3 float	Цвет 4 float

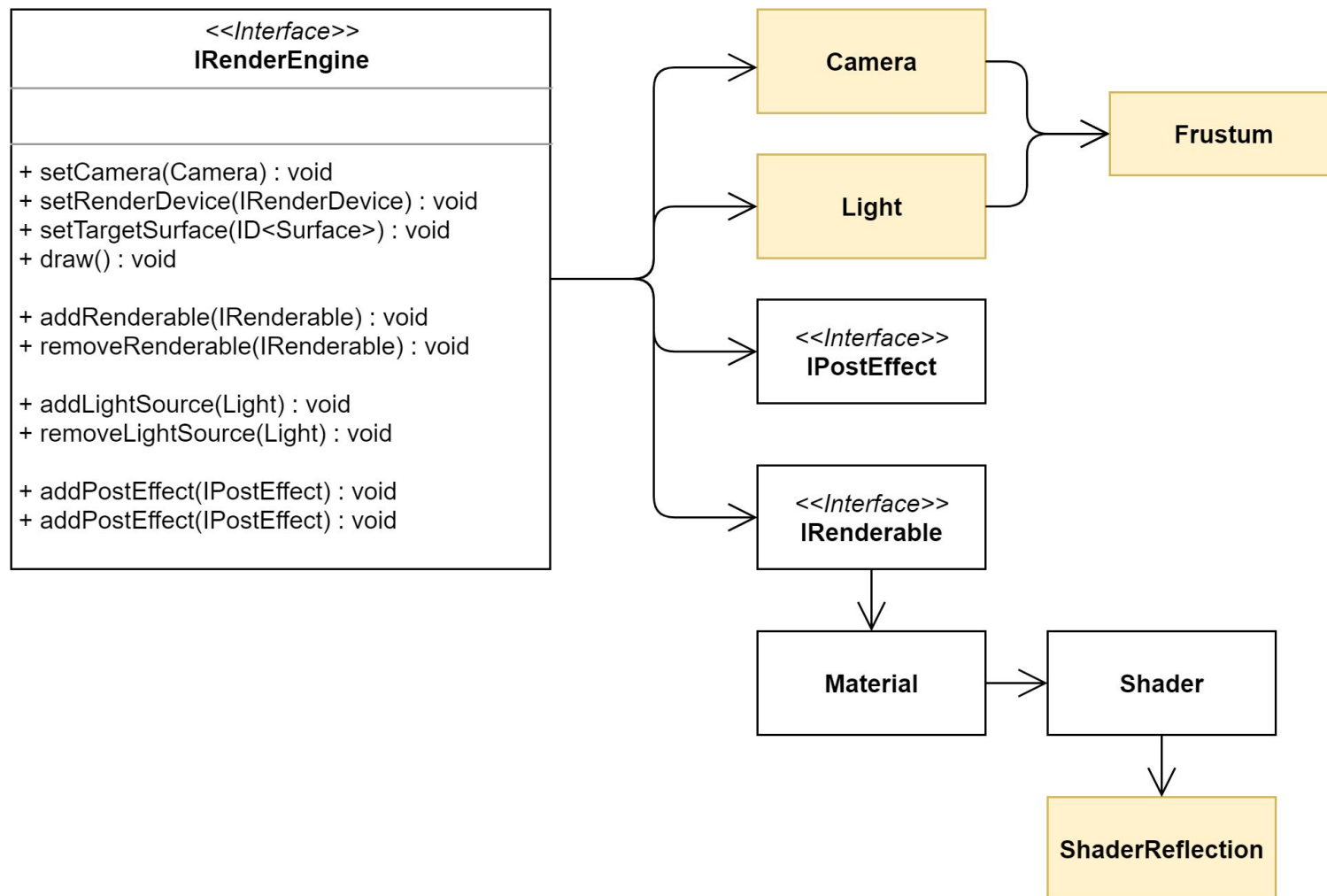
Структуры вершин



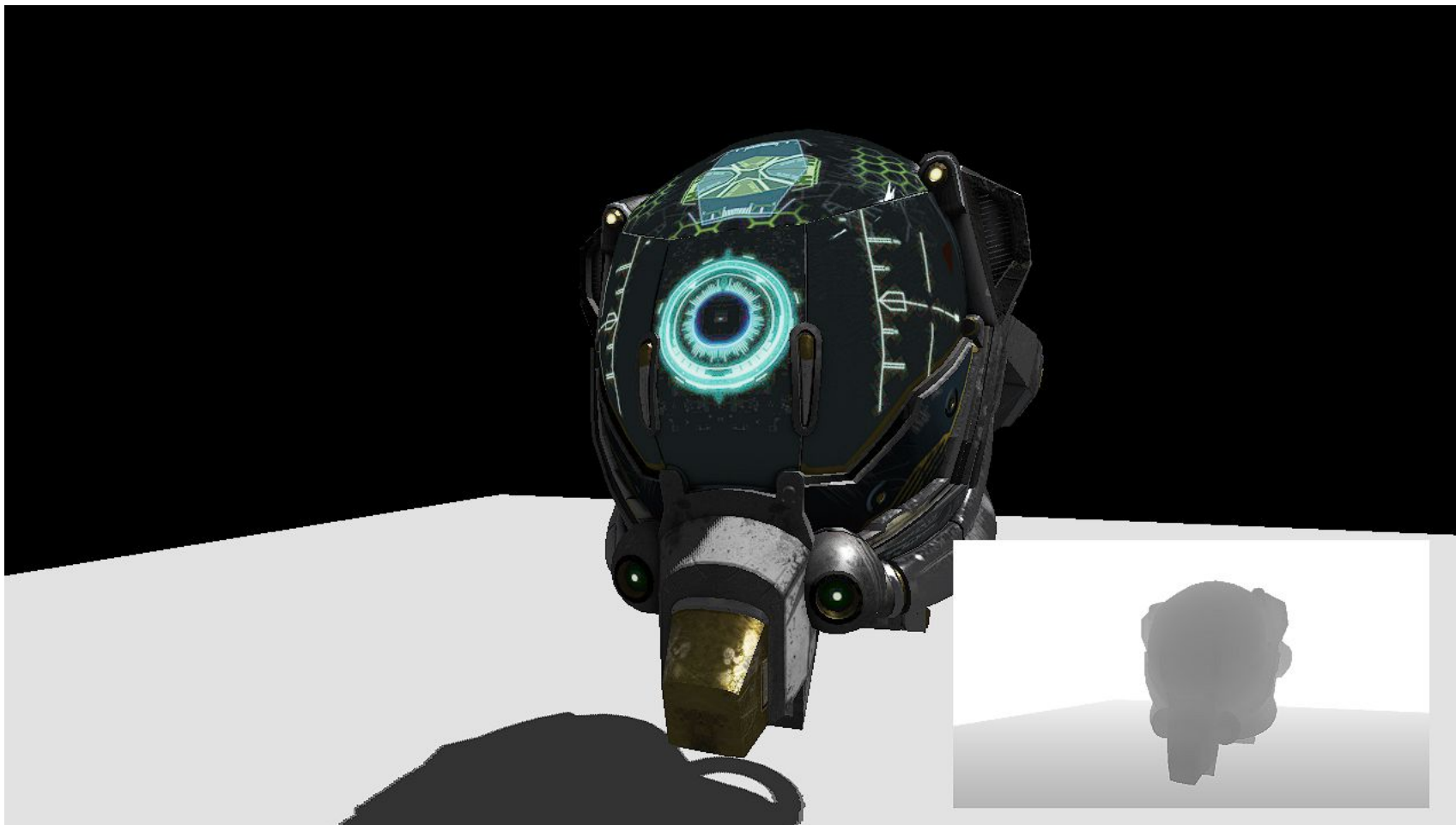
Интеграция Vulkan Memory Allocator

- Изначально: каждому `VkBuffer` — свой `VkDeviceMemory`
- Количество одновременно существующих `VkDeviceMemory` ограничено
- Vulkan Memory Allocator решает эту проблему, оптимально используя количество аллокаций `VkDeviceMemory`

Высокоуровневая система отображения



Примитивы



Рефлексия шейдеров с помощью SPIRV-Cross

Параметры:

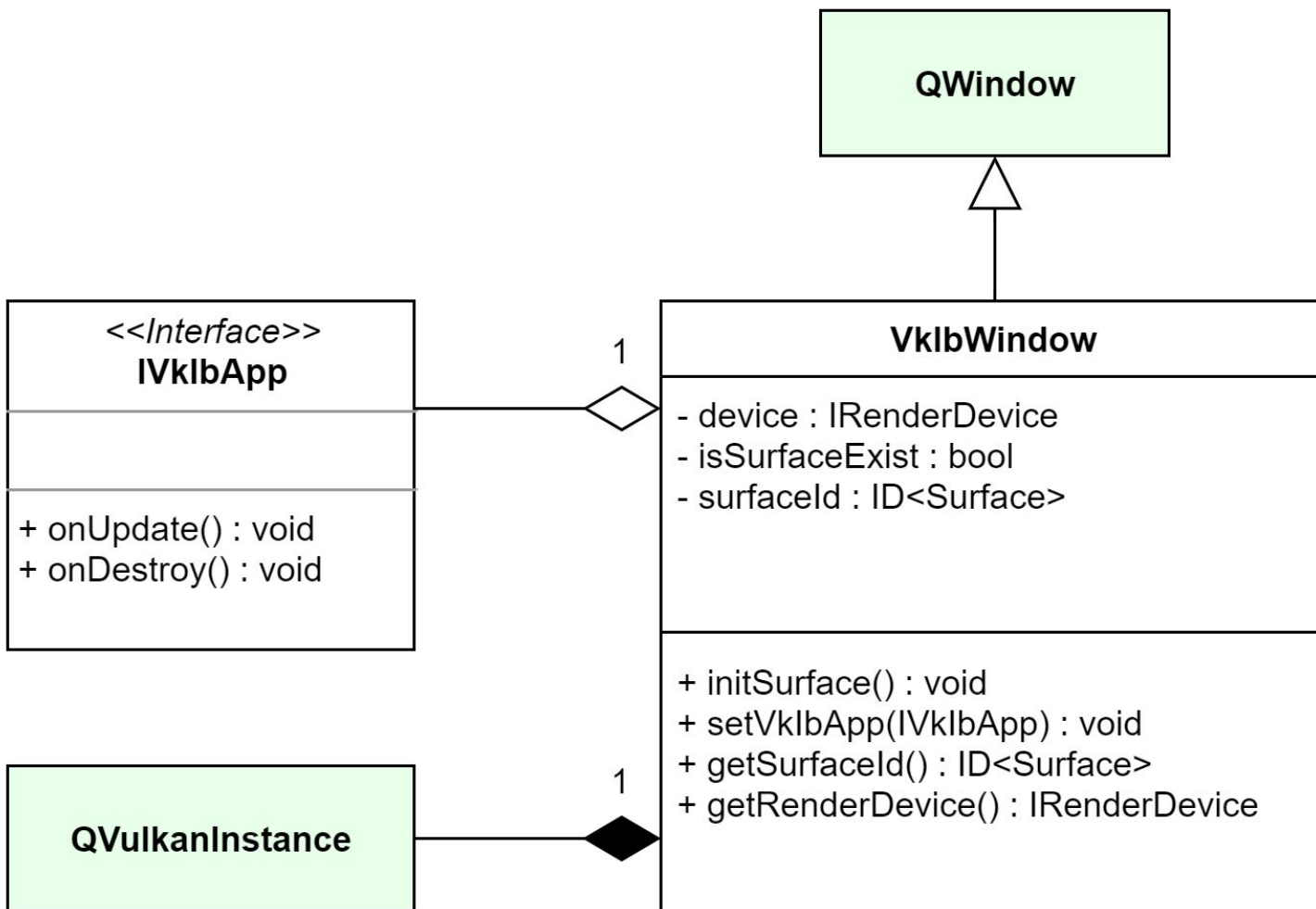
- входные и выходные данные
- буферы параметров шейдера
- текстуры

```
layout (location = 0) in vec4 inColor;  
layout (binding = 0) uniform UBO {  
    mat4 mat;  
    vec3 w;  
} ubo;  
layout (binding = 1) uniform sampler2D texture;
```

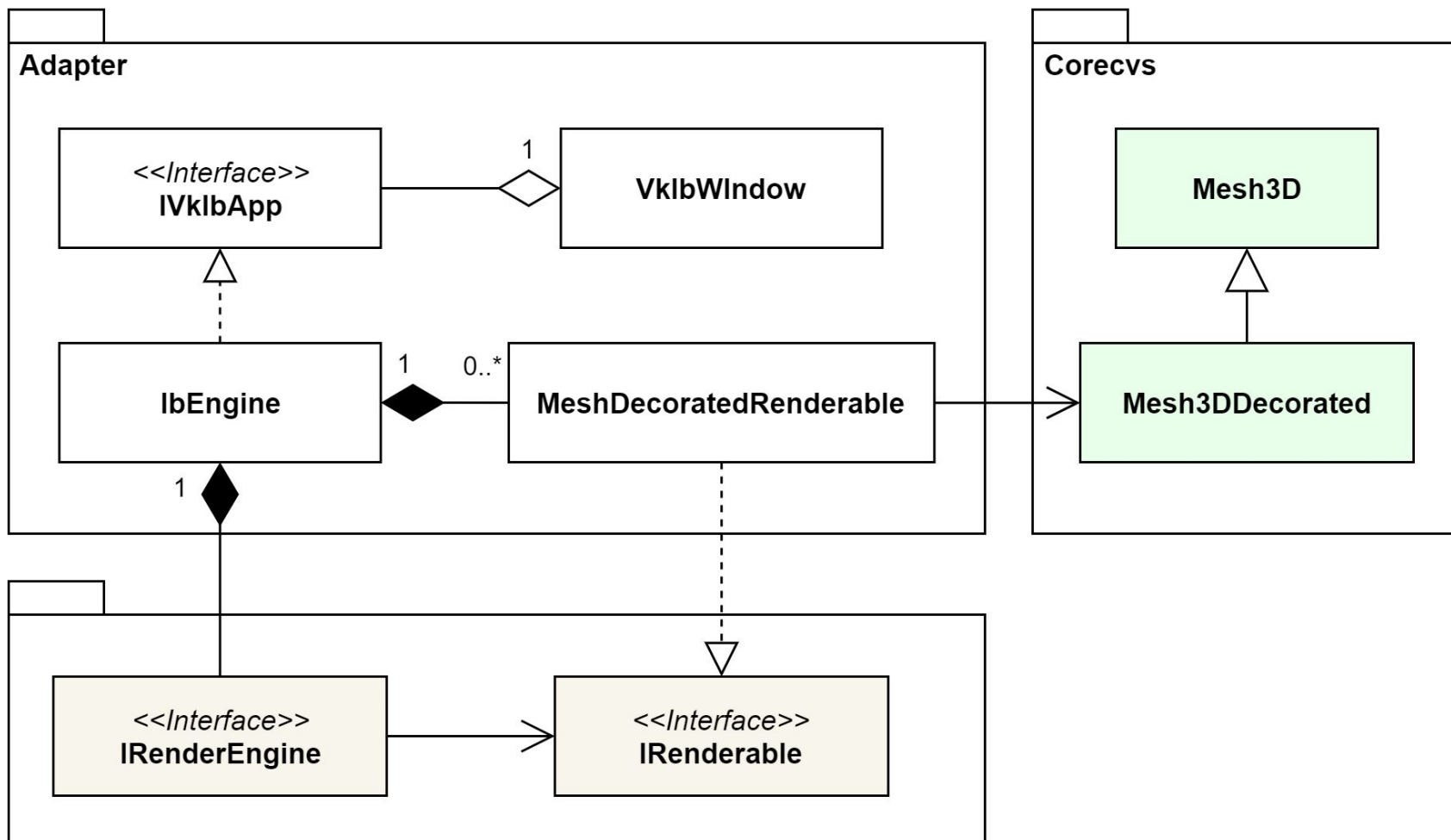
Интеграция в CoreCVS

- CoreCVS использует фреймворк Qt
- Для Vulkan необходим [VkSurfaceKHR](#) для отрисовки в окно
- Qt с версии 5.10 предоставляет возможность получения [VkSurfaceKHR](#)

Интеграция в CoreCVS



Интеграция в CoreCVS



Итоги

- Реализована графическая библиотека:
 - с оберткой над низкоуровневым графическим API
 - с высокоуровневой системой отображения
- Библиотека интегрирована в подсистему отображения библиотеки CoreCVS