

14.09.2024

- Программные средства
- Процесс построения изображения
- Модель графического конвейера

# Терминология

**Спецификация (стандарт, API)** — документированные наборы правил, рекомендаций и параметров, определяющие способы взаимодействия, описание, единое понимание и совместимость между аппаратной и программной частями.

**Драйвер (видеодрайвер)** — программный интерфейс между операционной системой и графическим аппаратным обеспечением компьютера или устройства.

**Графическая библиотека** — набор программных инструментов, функций и ресурсов, предназначенных для упрощения создания графических элементов в компьютерных приложениях.

**Графический фреймворк** — комплексная структура, предоставляющая базовую архитектуру и инструменты для разработки графических приложений.

**Графический движок** — программное обеспечение, которое предоставляет инфраструктуру и инструменты для разработки интерактивных графических приложений, игр и визуальных симуляций.

**Графические редакторы** — программное обеспечение, предназначенное для создания, редактирования и манипулирования графическими изображениями, включая различные эффекты, анимацию и многое другое.

**Графический профилировщик** — программное обеспечение, используемое для анализа и оптимизации производительности графических приложений или систем.

**GPU benchmark (бенчмарк графического процессора)** — методика тестирования и оценки производительности графического процессора, которая позволяет измерить его способность обрабатывать графику и выполнение вычислительных задач.

2024-10-04

Построение изображений  
└ Программные средства  
└ Терминология

Терминология  
Спецификация (стандарт, API) — документированные наборы правил, рекомендаций и параметров, определяющие способы взаимодействия, описание, единое понимание и совместимость между аппаратной и программной частями.  
Драйвер (видеодрайвер) — программный интерфейс между операционной системой и графическим аппаратным обеспечением компьютера или устройства.  
Графическая библиотека — набор программных инструментов, функций и ресурсов, предназначенных для упрощения создания графических элементов в компьютерных приложениях.  
Графический фреймворк — комплексная структура, предоставляющая базовую архитектуру и инструменты для разработки графических приложений.  
Графический движок — программное обеспечение, которое предоставляет инфраструктуру и инструменты для разработки интерактивных графических приложений, игр и визуальных симуляций.  
Графические редакторы — программное обеспечение, предназначенное для создания, редактирования и манипулирования графическими изображениями, включая различные эффекты, анимацию и многое другое.  
Графический профилировщик — программное обеспечение, используемое для анализа и оптимизации производительности графических приложений или систем.  
GPU benchmark (бенчмарк графического процессора) — методика тестирования и оценки производительности графического процессора, которая позволяет измерить его способность обрабатывать графику и выполнение вычислительных задач.

Графическое API (Application Programming Interface) — это набор интерфейсов и функций, предоставляемых операционной системой или программной средой, который позволяет программам взаимодействовать с аппаратным обеспечением для рендеринга графики. Графические API позволяют разработчикам управлять отображением 2D- и 3D-графики на экране, взаимодействовать с графическим процессором (GPU), управлять ресурсами (например, текстурами, шейдерами) и другими элементами графики.

# Спецификации (API) и графические библиотеки

## OpenGL (1992, Silicon Graphics)

Открытая кросс-платформенная спецификация для работы с 2D и 3D графикой

## Mesa (1995, Brian Paul)

Свободная реализация графических API OpenGL (позже Vulkan и др.) с открытым исходным кодом

## DirectX (1995, Microsoft)

Пакет графических API для работы с играми и мультимедийными приложениями на платформе Windows

## WebGL (2011, Khronos Group)

Графический API для веб-браузеров

## Mantle (2013, AMD)

Спецификация низкоуровневого API

## Vulkan (2016, Khronos Group)

Низкоуровневый (требуется явное управление памятью и ресурсами) высокопроизводительный кроссплатформенный API для работы с 2D и 3D графикой

2024-10-04

## Построение изображений

### └ Программные средства

### └ Спецификации (API) и графические библиотеки

Спецификации (API) и графические библиотеки  
**OpenGL (1992, Silicon Graphics)**  
Открытая кросс-платформенная спецификация для работы с 2D и 3D графикой  
**Mesa (1995, Brian Paul)**  
Свободная реализация графических API OpenGL (позже Vulkan и др.) с открытым исходным кодом  
**DirectX (1995, Microsoft)**  
Пакет графических API для работы с играми и мультимедийными приложениями на платформе Windows  
**WebGL (2011, Khronos Group)**  
Графический API для веб-браузеров  
**Mantle (2013, AMD)**  
Спецификация низкоуровневого API  
**Vulkan (2016, Khronos Group)**  
Низкоуровневый (требуется явное управление памятью и ресурсами) высокопроизводительный кроссплатформенный API для работы с 2D и 3D графикой

MESA — это открытый проект, который предоставляет реализацию различных графических API (таких как OpenGL, OpenGL ES, Vulkan, OpenCL и др.) для Linux и других операционных систем с открытым исходным кодом.

MESA поддерживает реализацию OpenGL для устройств, разработанных Intel, AMD, NVIDIA, Qualcomm и др., включая виртуальные графические процессоры VMware и VirGL. Также есть несколько программных рендереров: Softpipe (драйвер Gallium) и LLVMpipe (высокоскоростной растеризатор на основе LLVM/JIT).

2024-10-04

Построение изображений

└ Программные средства

└ MESA

MESA

MESA — это открытый проект, который предоставляет реализацию различных графических API (таких как OpenGL, OpenGL ES, Vulkan, OpenCL и др.) для Linux и других операционных систем с открытым исходным кодом.

MESA поддерживает реализацию OpenGL для устройств, разработанных Intel, AMD, NVIDIA, Qualcomm и др., включая виртуальные графические процессоры VMware и VirGL. Также есть несколько программных рендереров: Softpipe (драйвер Gallium) и LLVMpipe (высокоскоростной растеризатор на основе LLVM/JIT).

Проект начал разрабатывать Брайн Пол в 1993. Основная цель заключалась в реализации OpenGL подобной IRIS GL. Первая версия вышла в 1995 с разрешения SGI. Названа в честь языка программирования MESA.  
<https://docs.mesa3d.org/history.html>

# Структура современной графической библиотеки

## Графический движок (движок рендеринга 2-х или 3-х мерной КГ)

- Должны работать в реальном времени
- Поддержка шейдеров

## Анимация

- Кинематика(компьютерный фильм)

## Физический движок (физика)

- Динамика жидкости, газа, взаимодействия тел и т.д.

## Игровой ИИ (game artificial intelligence)

- Боты (bots), моды (mods) и неигровые персонажи (non-player characters)

## Звук, система скриптов (система I/O), сетевой интерфейс и т.д.

2024-10-04

## Построение изображений

### Программные средства

### Структура современной графической библиотеки

Структура современной графической библиотеки

Графический движок (движок рендеринга 2-х или 3-х мерной КГ)

- Должны работать в реальном времени
- Поддержка шейдеров

Анимация

- Кинематика(компьютерный фильм)

Физический движок (физика)

- Динамика жидкости, газа, взаимодействия тел и т.д.

Игровой ИИ (game artificial intelligence)

- Боты (bots), моды (mods) и неигровые персонажи (non-player characters)

Звук, система скриптов (система I/O), сетевой интерфейс и т.д.

# Распределение вычислений между CPU и GPU

Central and Graphical Processor Unit

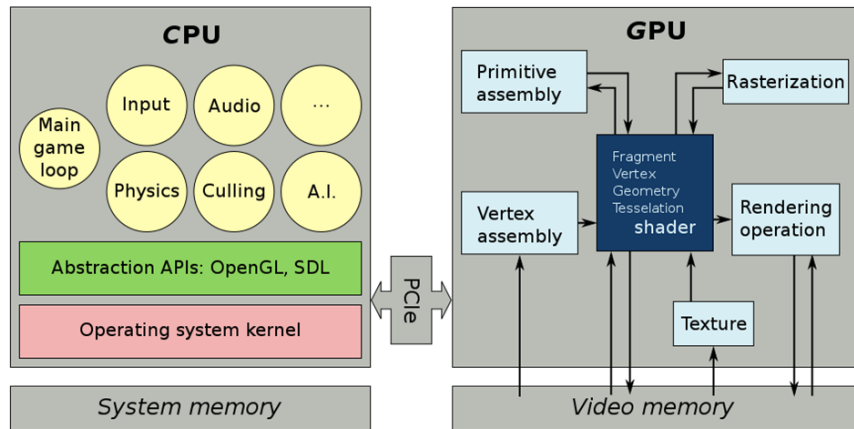


Рис. 1: Принципиальная схема распределения вычислений

2024-10-04

Построение изображений

Программные средства

Распределение вычислений между CPU и GPU

Распределение вычислений между CPU и GPU  
Central and Graphical Processor Unit

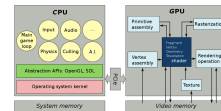


Рис. 1: Принципиальная схема распределения вычислений

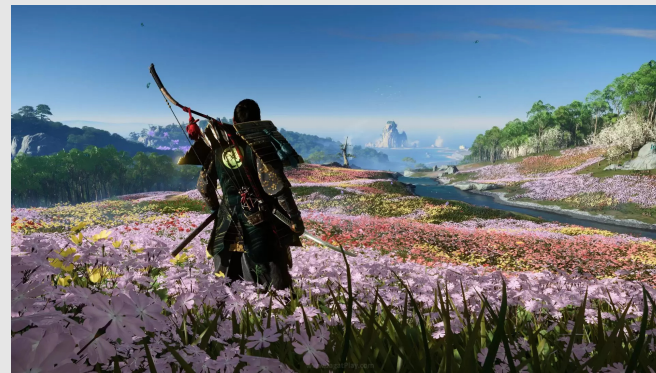


Рис. 2: Ghost of Tsushima (2024, Призрак Цусимы)

# Растровая графика

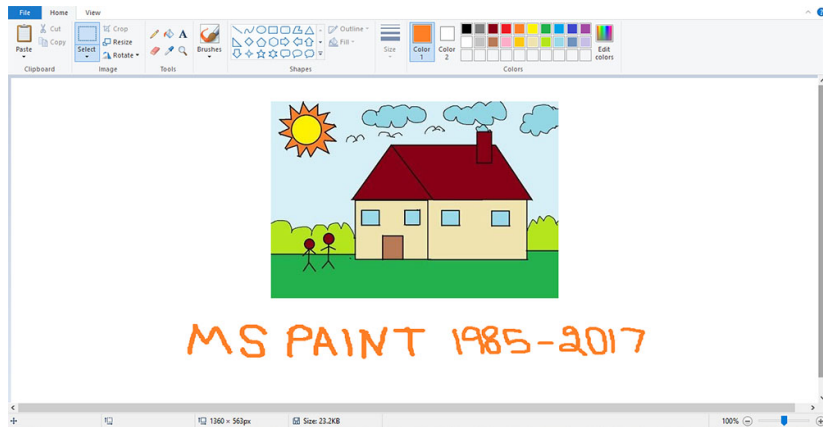


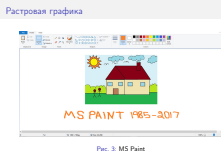
Рис. 3: MS Paint

2024-10-04

## Построение изображений

- Процесс построения изображения

- Растровая графика



Растровая графика (bitmap или pixel-based) представляет изображение в виде массива пикселей. Каждый пиксель хранит информацию о цвете, расположенные в определенном порядке формируют изображение.

Качество изображения зависит от количества пикселей. Чем выше разрешение (количество пикселей на единицу площади), тем детализированнее изображение.



# Векторная графика

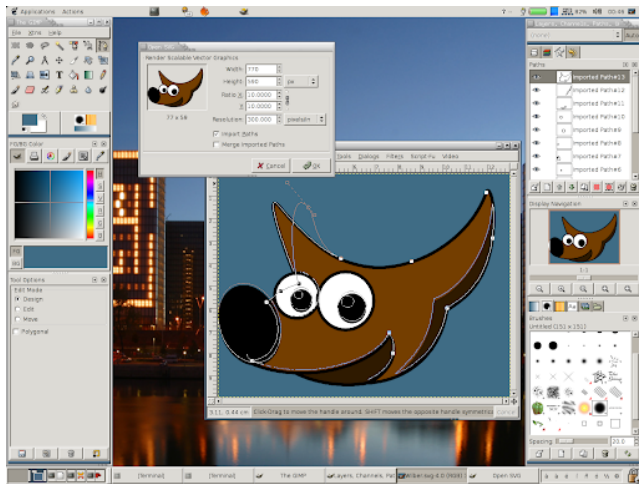


Рис. 4: GIMP

2024-10-04

## Построение изображений

- Процесс построения изображения

- Векторная графика

Векторная графика

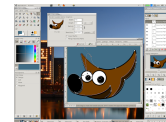


Рис. 4: GIMP

Векторная графика представляет изображение состоящее из примитивов и/или математических объектов (линии, точки, треугольники, кривые, поверхности и др.). Эти объекты описываются с помощью набора параметров и/или уравнений, что позволяет бесконечно масштабировать изображение без потери качества.

# Воксельная графика

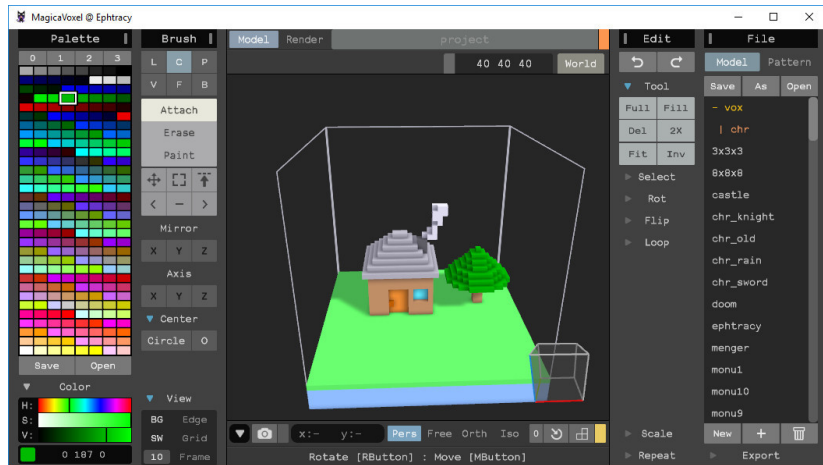


Рис. 5: Виды преобразований и системы координат

2024-10-04

Построение изображений

└─ Процесс построения изображения

└─ Воксельная графика

Воксельная графика

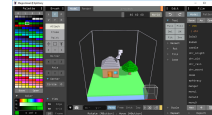


Рис. 5: Виды преобразований и системы координат

Воксельная графика — это трехмерный аналог растровой графики, где изображение состоит из объёмных пикселей, называемых вокселями (volume elements, объёмные элементы). Воксели представляют собой кубические блоки, которые формируют объёмные объекты в 3D-пространстве.

# Процесс построения изображения

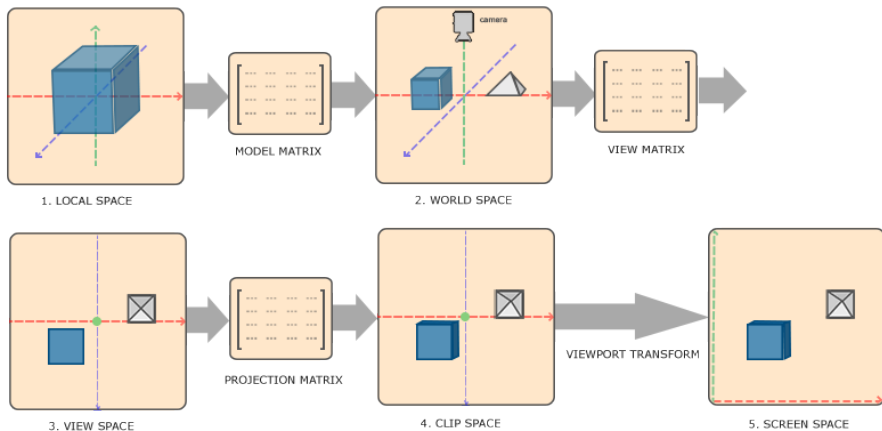


Рис. 6: Виды преобразований и системы координат

2024-10-04

Построение изображений

└ Процесс построения изображения

└ Процесс построения изображения

Процесс построения изображения

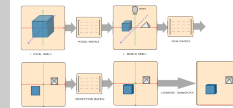


Рис. 6: Виды преобразований и системы координат

Модельные координаты (МК)

Преобразование моделирования

Внешние координаты (ВК)

Преобразование наблюдения

Координаты наблюдения (КН)

Преобразование проецирования

Координаты проекции (КП)

Преобразование нормировки и отсечение

Нормированные координаты (НК)

Преобразование поля просмотра

Координаты устройства (КУ)

# Процесс построения изображения

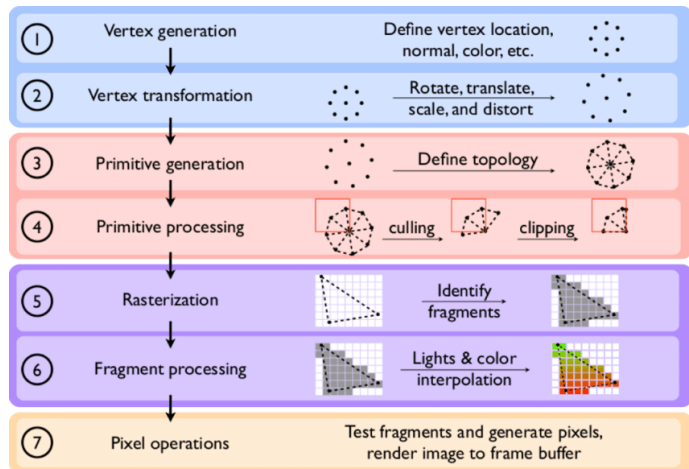


Рис. 7: Схема графического конвейера

2024-10-04

Построение изображений

└ Процесс построения изображения

└ Процесс построения изображения

Процесс построения изображения

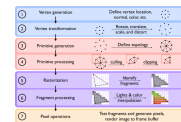


Рис. 7: Схема графического конвейера

## Конвейер рисования в OpenGL

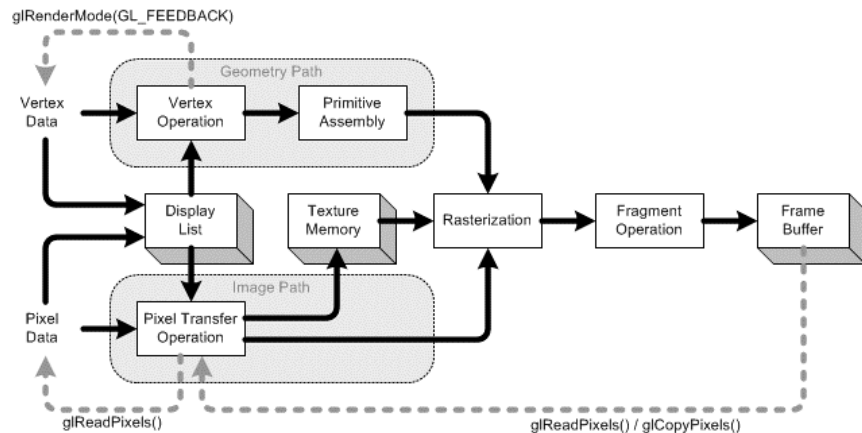


Рис. 8: Принципиальная схема распределения вычислений

2024-10-04

## Построение изображений

└─ Процесс построения изображения

└─ Конвейер рисования в OpenGL

Конвейер рисования в OpenGL

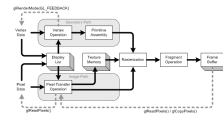
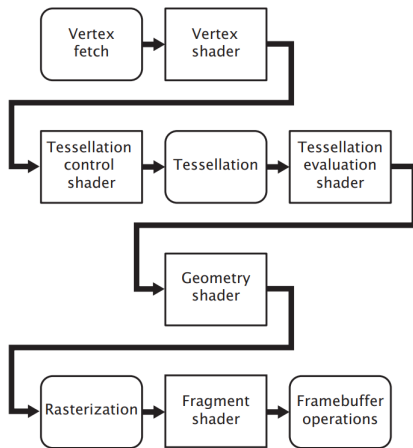


Рис. 8: Принципиальная схема распределения вычислений

# Упрощенная схема графического конвейера

Shaders



Загрузка данных

Вершина (vertices)

Вершинный шейдер

Группа вершин (primitives/patches)

Шейдер управление тесселяцией

Тесселяция

Шейдер определяющий тесселяции

Примитивы (primitives)

Геометрический шейдер

Примитивы (primitives)

Растреризация и интерполяция

Пиксели (fragments)

Пиксельный (фрагментный) шейдер

Пиксели (fragments)

Операции с буферами кадров

Пиксели (Pixels)

Рис. 9: Порядок вычисления шейдеров

2024-10-04

Построение изображений

Модель графического конвейера

Упрощенная схема графического конвейера



# Упрощенная модель графического конвейера OpenGL API

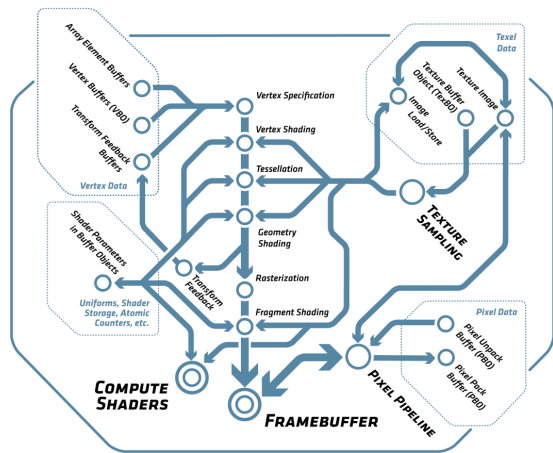


Рис. 10: Принципиальная схема распределения вычислений

2024-10-04

## Построение изображений

└─ Модель графического конвейера

└─ Упрощенная модель графического конвейера



Рис. 10: Принципиальная схема распределения вычислений

# Упрощенная модель графического конвейера Vulkan API

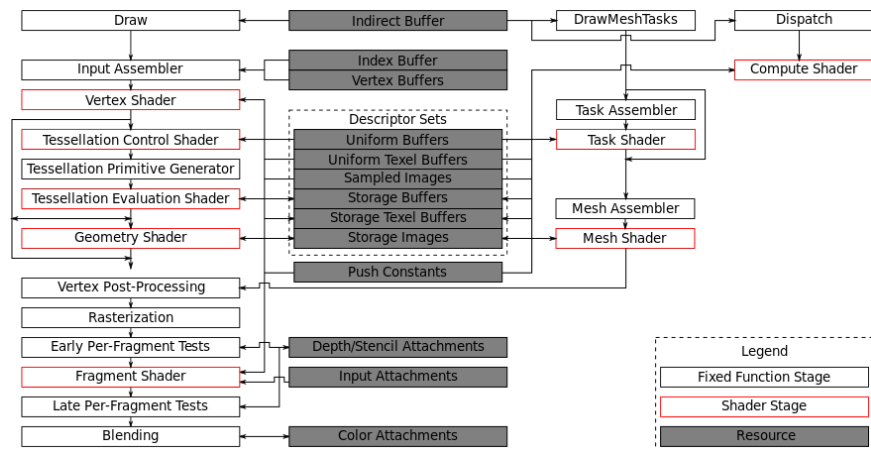


Рис. 11: Блок-схема конвейера Vulkan API

2024-10-04

## Построение изображений

### Модель графического конвейера

Упрощенная модель графического конвейера



Рис. 11: Блок-схема конвейера Vulkan API