Разработка системы анимации для оптимизации визуализации трехмерных сцен.

Бородулин Федор 08-1 класс

Кондаков Александр 08-1 класс

Сопина Елизавета 08-1 класс

Шемякина Елизавета 08-1 класс

Львова Диана 08-2 класс

Чугунов Андрей 09-1 класс

Смирнов Даниил 09-2 класс

Дмитриев Иван 09-3 класс

Усиков Андрей 09-3 класс

Бисеров Владислав 09-5 класс

Молостов Максим 09-5 класс

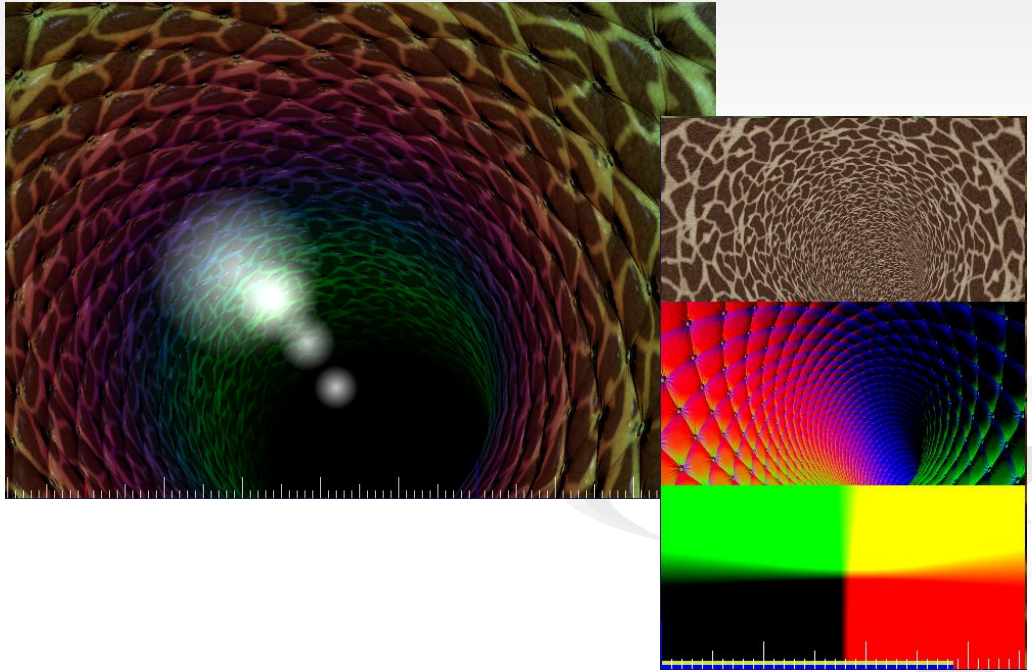
Горляков Даниил 10-5 класс

Васильев Андрей 10-6 класс

Конев Данил 10-6 класс

Лановая Александра 10-6 класс

Сабитов Кирилл 10-6 класс

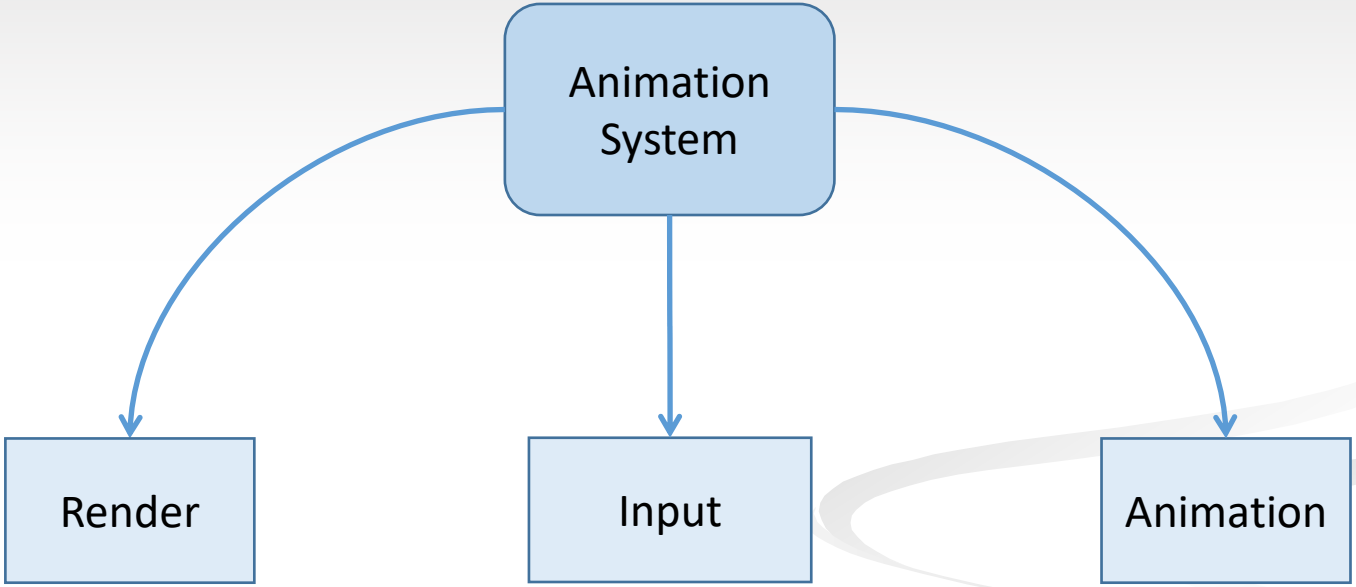


Научный руководитель: Галинский В.А., руководитель группы компьютерной графики,
преподаватель информатики и программирования СПб губернаторского физико-математического лицея № 30

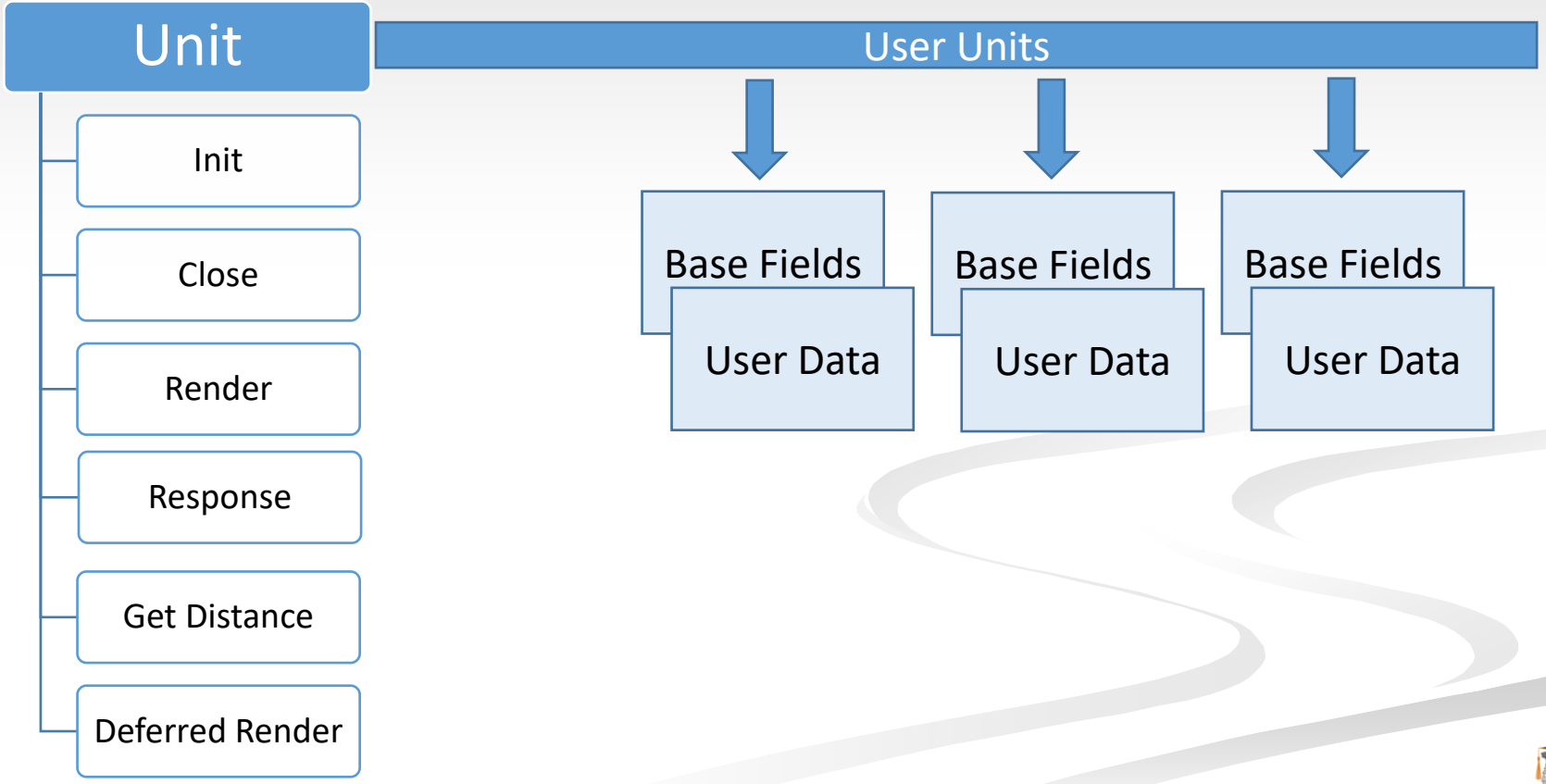
Computer Graphics Support Group

Санкт-Петербургский губернаторский физико-математический лицей № 30

Анимация



Активные элементы анимации



Конвейер анимации

Animation Initialization → Animation Render → Animation Close

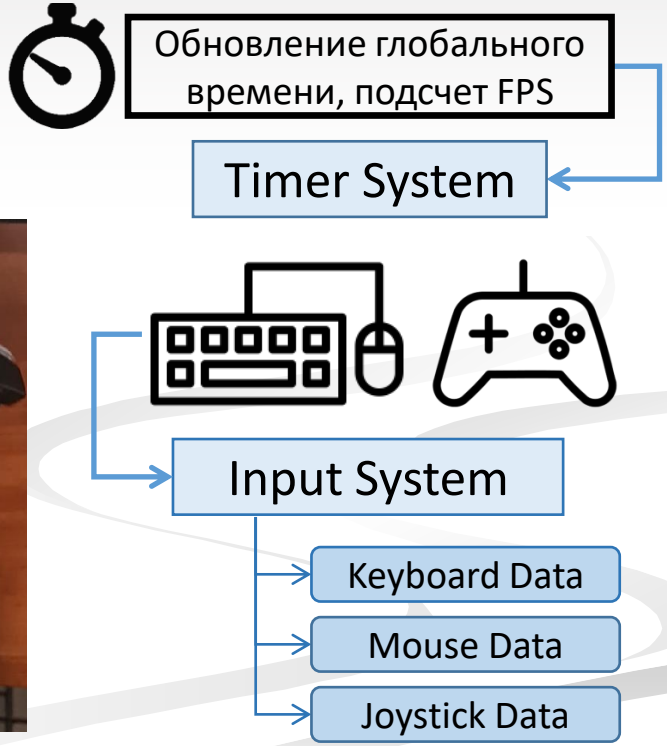
Опрос устройств ввода и таймера

Units response

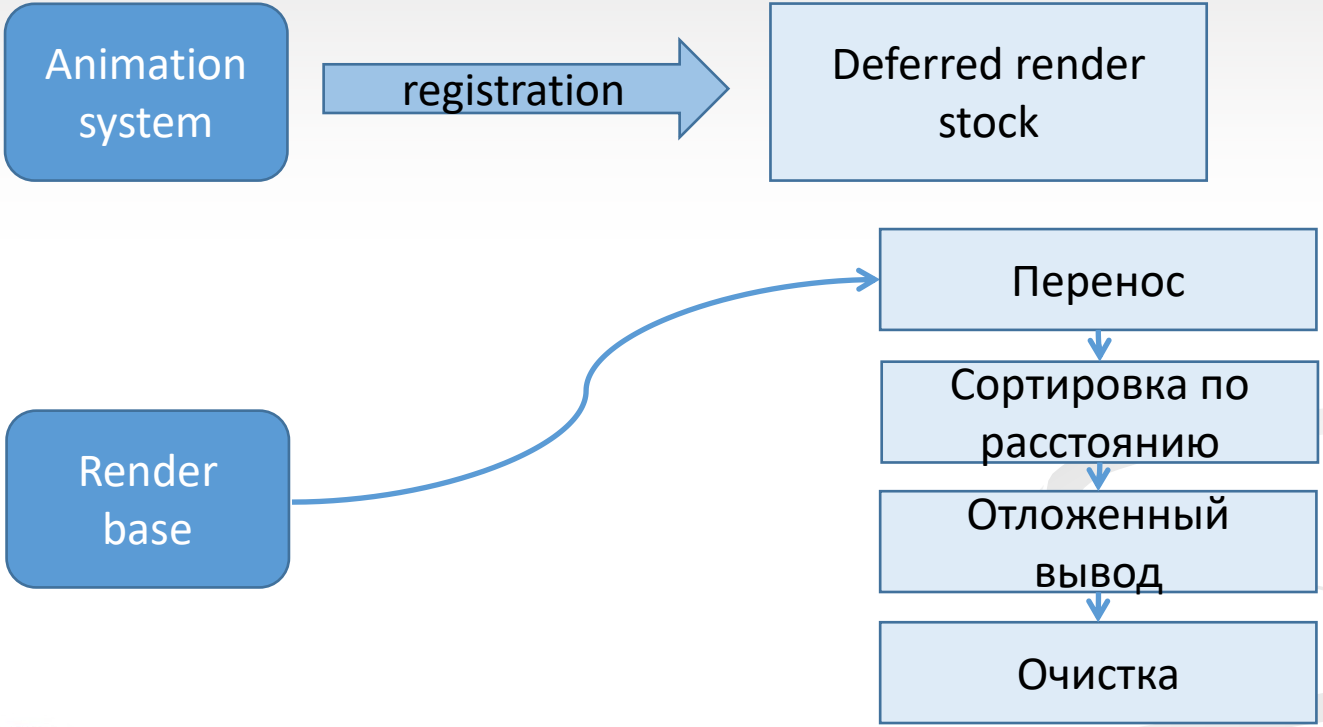
Render

Animation context structure

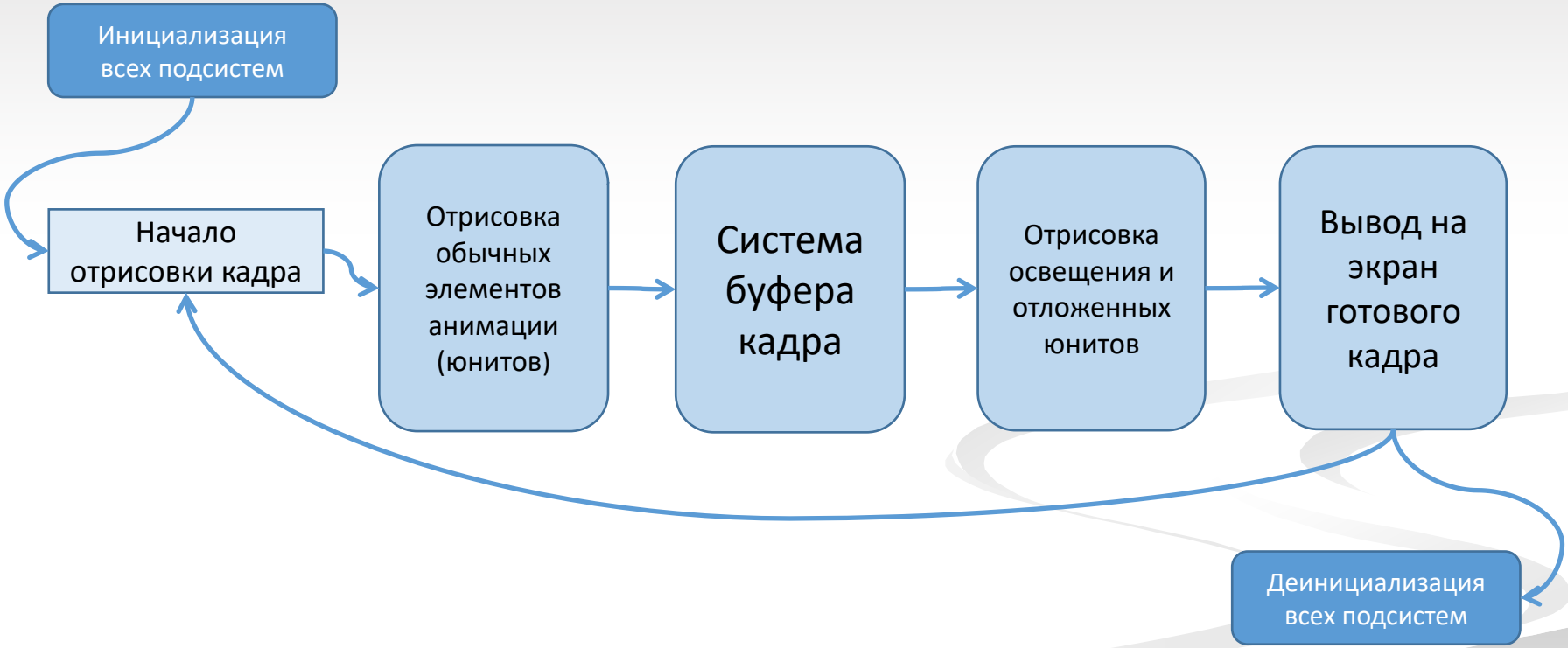
Опрос устройств и таймера



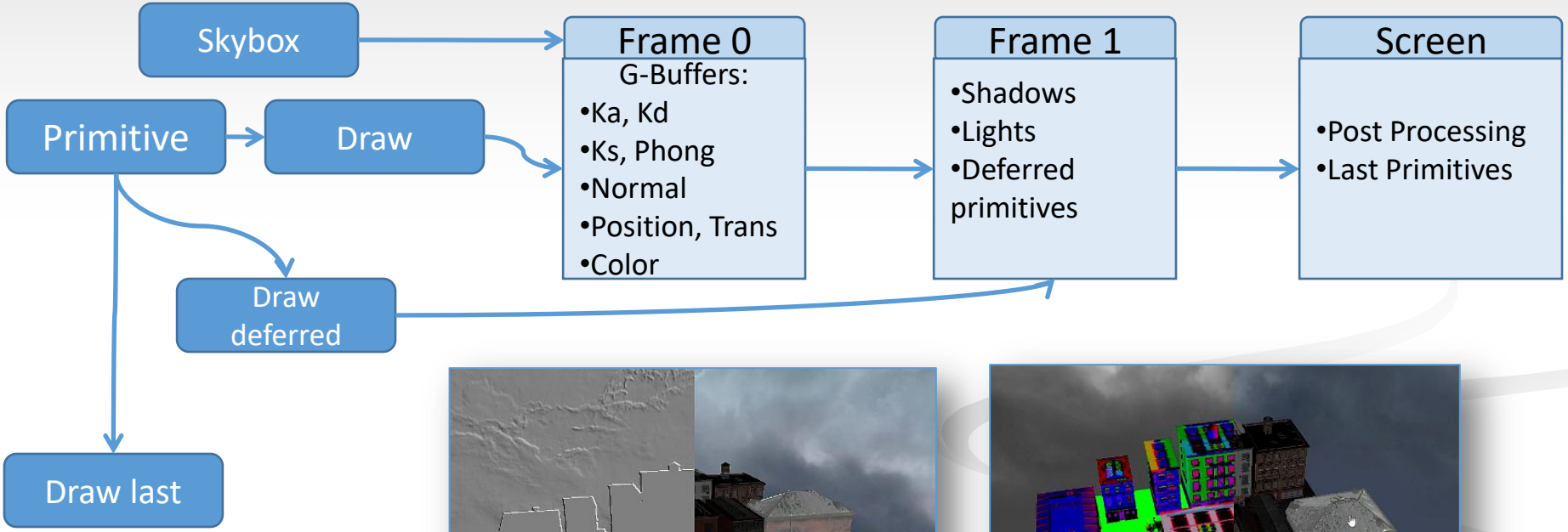
• Отложенный вывод активных элементов анимации (Units deferred rendering)

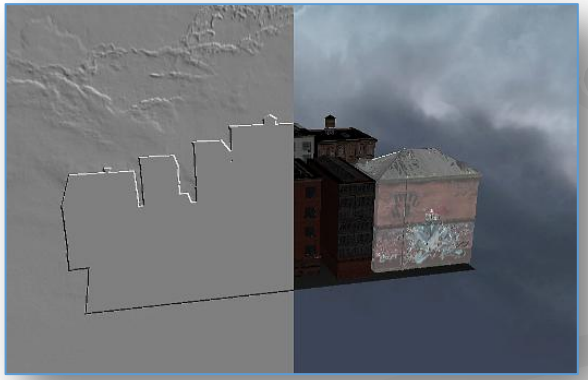


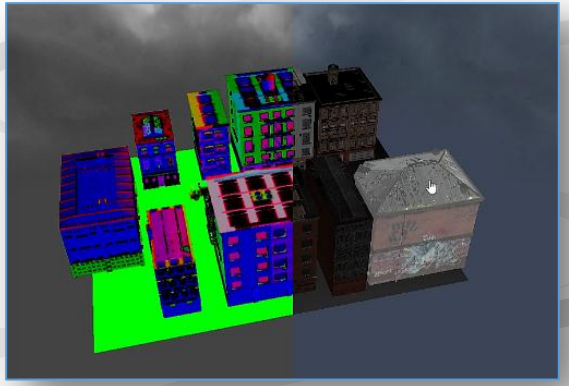
Система рендеринга



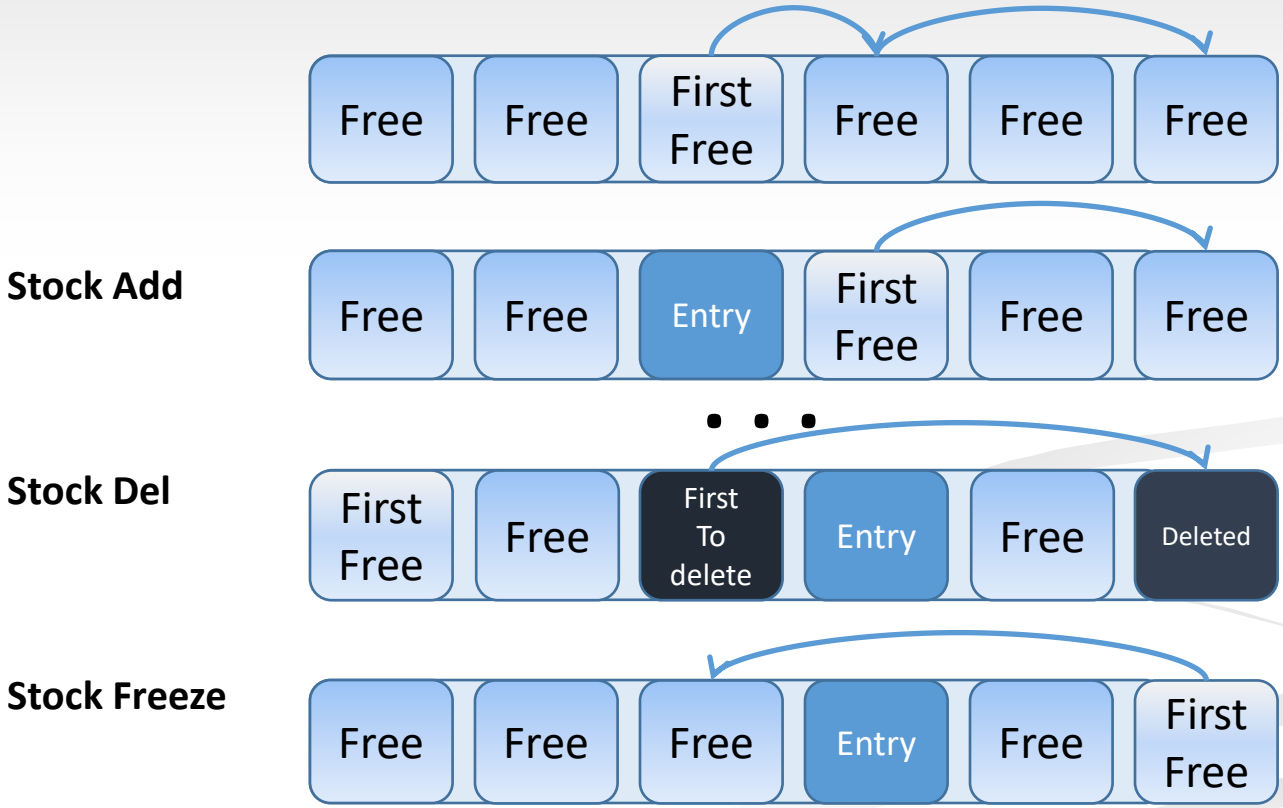
Rendering Pipeline



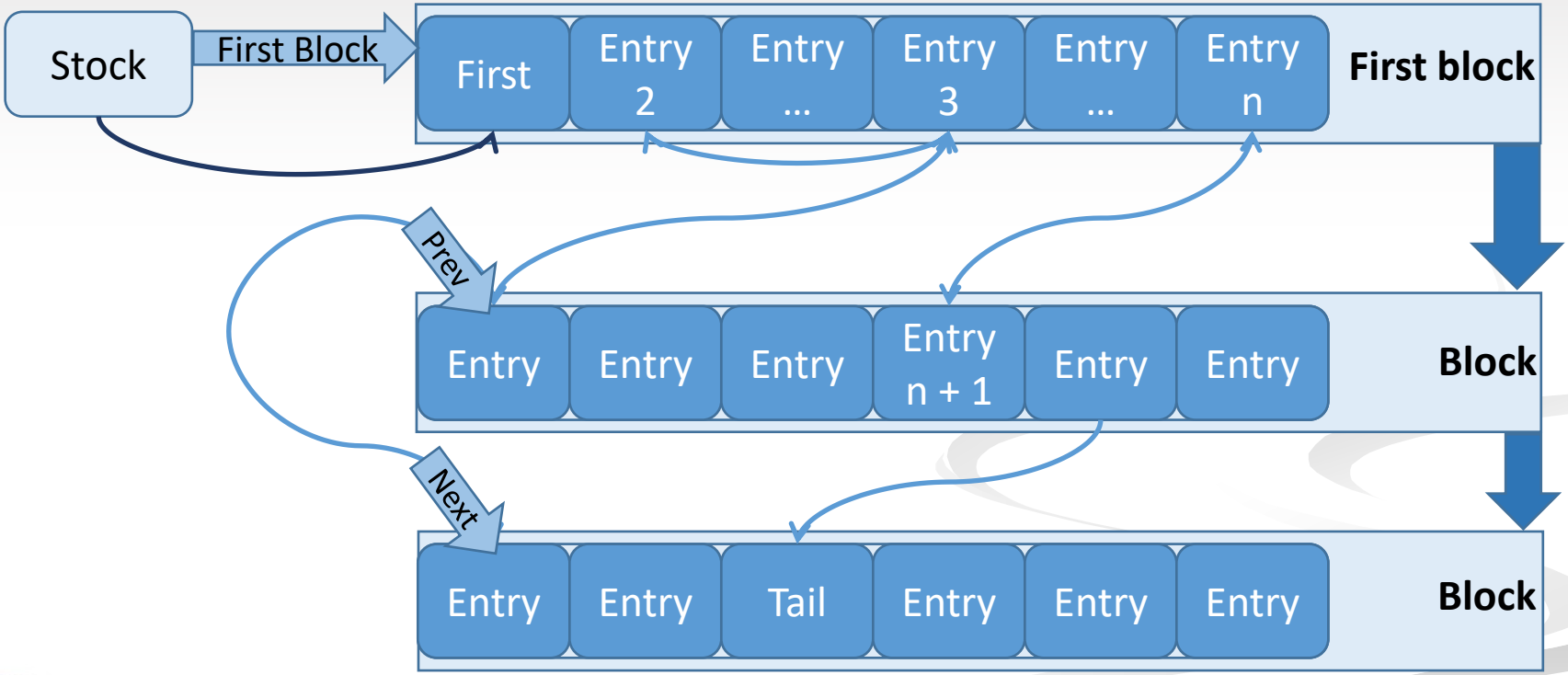




Stock system



Stock’s structure

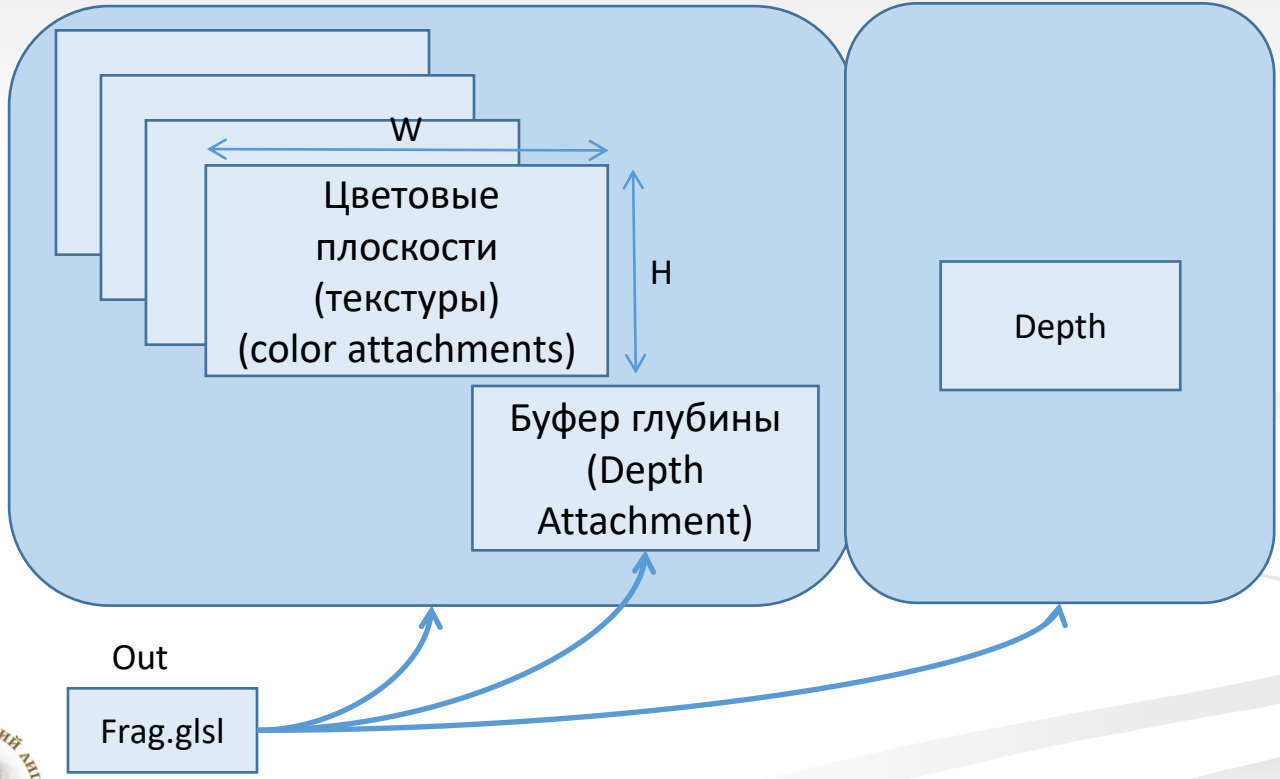


Количество entry в блоке – константа.

Буфер кадра

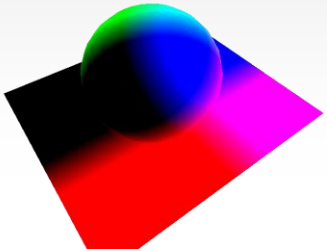
Структура буфера кадра (render target)

Размер буфера кадра (W, H)
Количество цветовых плоскостей
Флажок рендер буфера, буфера глубины, буфера трафарета
Номер рендер буфера, буфера кадра
Массив из указателей на текстуры

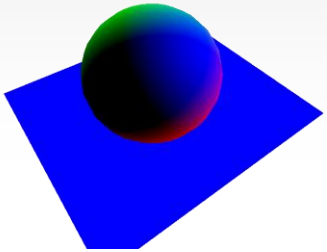


Разбиение на плоскости

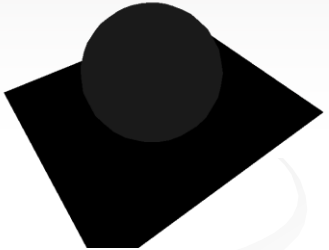
Позиция



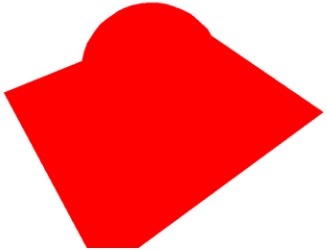
Нормали



Ka



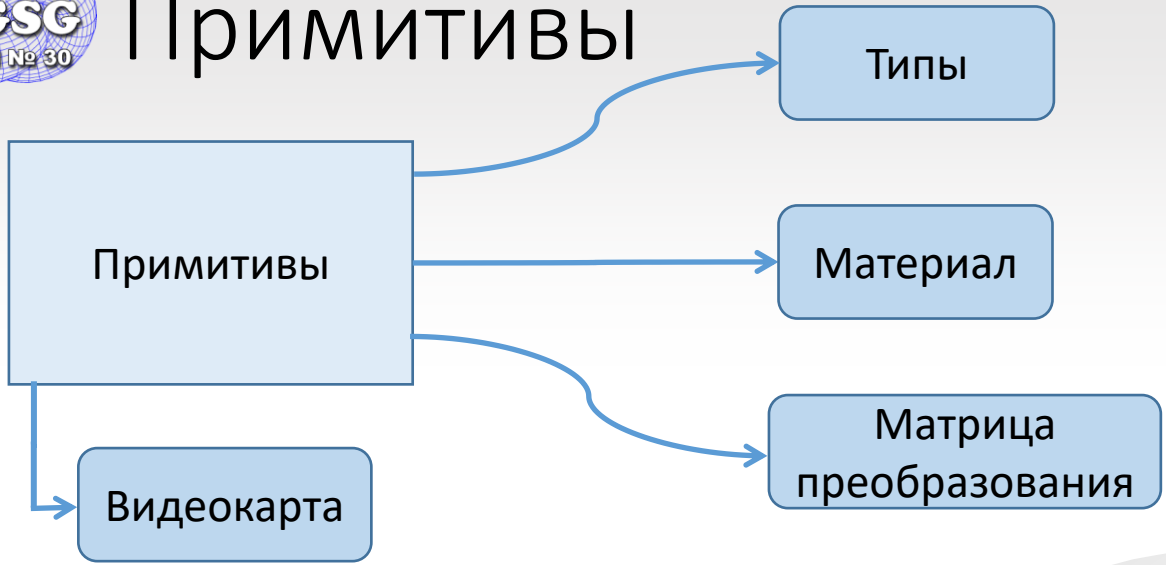
Kd



Ks

Выходной цвет

Примитивы



Вершины	(0, 0, 0)	(1, 0, 0)	(1, 1, 0)
Индексы	0	1	2

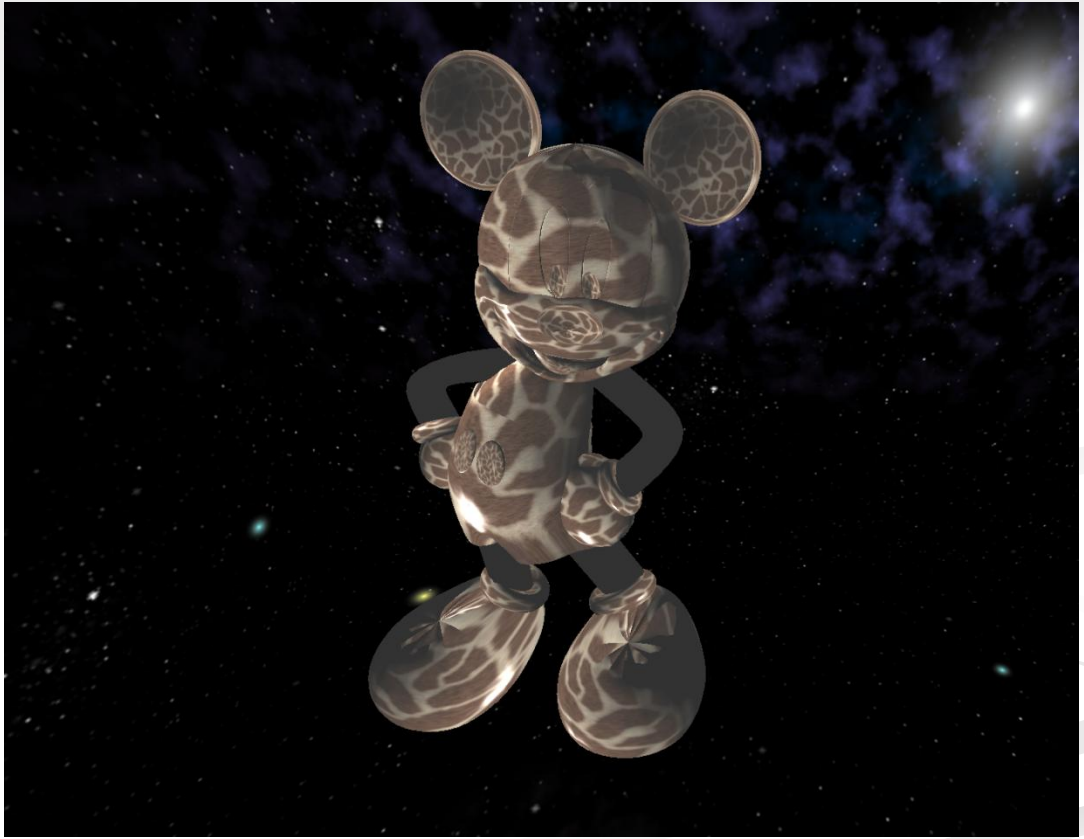
Способы вывода примитива на экран

Обычный вывод (Primitive Draw)

Отложенный вывод (Primitive Draw Deferred)

Вывод после всех остальных объектов (Primitive Draw Last)

Примитивы



Материалы

Структура материала

- Имя материала
- Указатель на шаблон материала
- Указатель на буфер материала
- Массив текстур материала

Шаблоны материалов

Структура шаблона материала

- Имя шаблона материала
- Массив шейдеров (Каждый шейдер используется при проходе рендера с таким же номером)
- Массив форматов вершин (состоит из структур, содержащих тип данных и количество)
- Количество блоков памяти для хранения коэффициентов освещения

Стандартный шаблон материала

Стандартный шаблон материала

Имя: «Стандартный шаблон»

Массив шейдеров состоит из 3 стандартных шейдеров

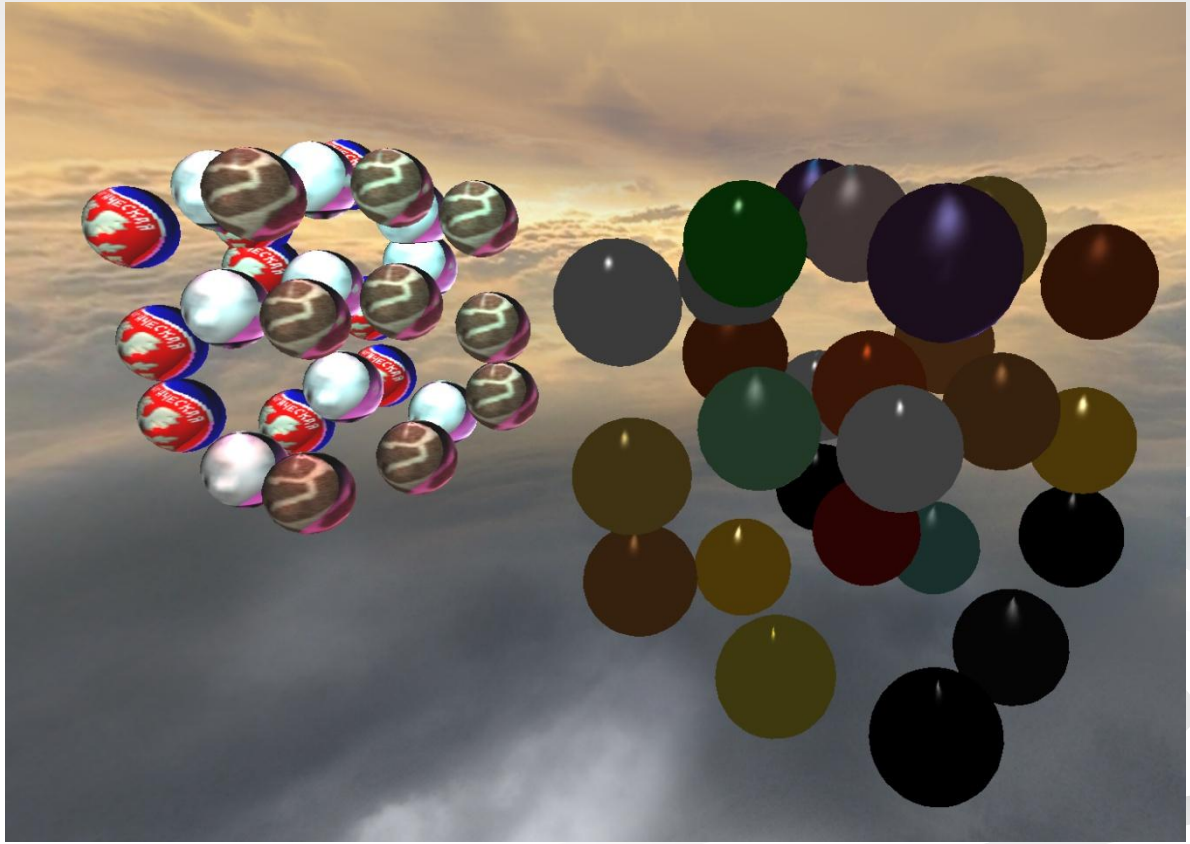
4 структуры типа vec4 (quads) для хранения данных материала

Вершины:

- Вектор с 3 координатами (vec3) для задания позиции объекта
- Вектор с 2 координатами (vec2) для задания текстурных координат объекта
- Вектор с 3 координатами (vec3) для задания нормалей объекта
- Вектор с 4 координатами (vec4) для задания цвета объекта
- Вектор с 3 координатами (vec3) для задания тангента объекта
- Вектор с 3 координатами (vec3) для задания битангента объекта

Создание материала

Библиотека материалов



Текстурирование объектов

Процесс создания текстуры

Создание изображения → Изменение изображения → Создание текстуры, ее загрузка в VRAM

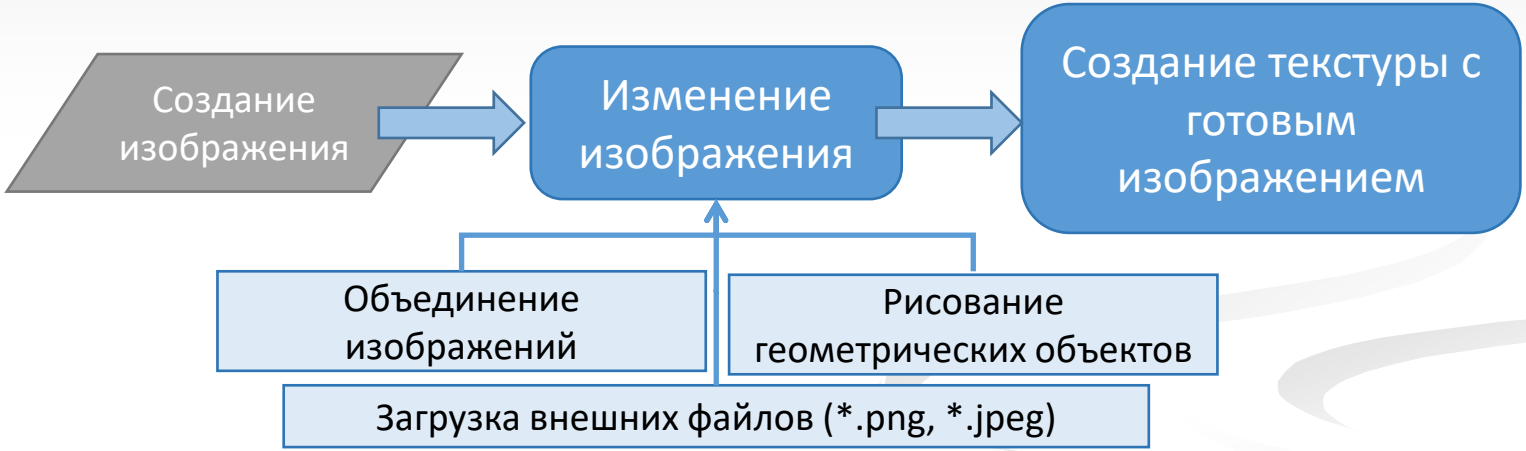
Texture Data Type

- Texture Type
- Image Size
- Image Layouts Type

Textures Stock

Texture Pointer

Изменение изображений



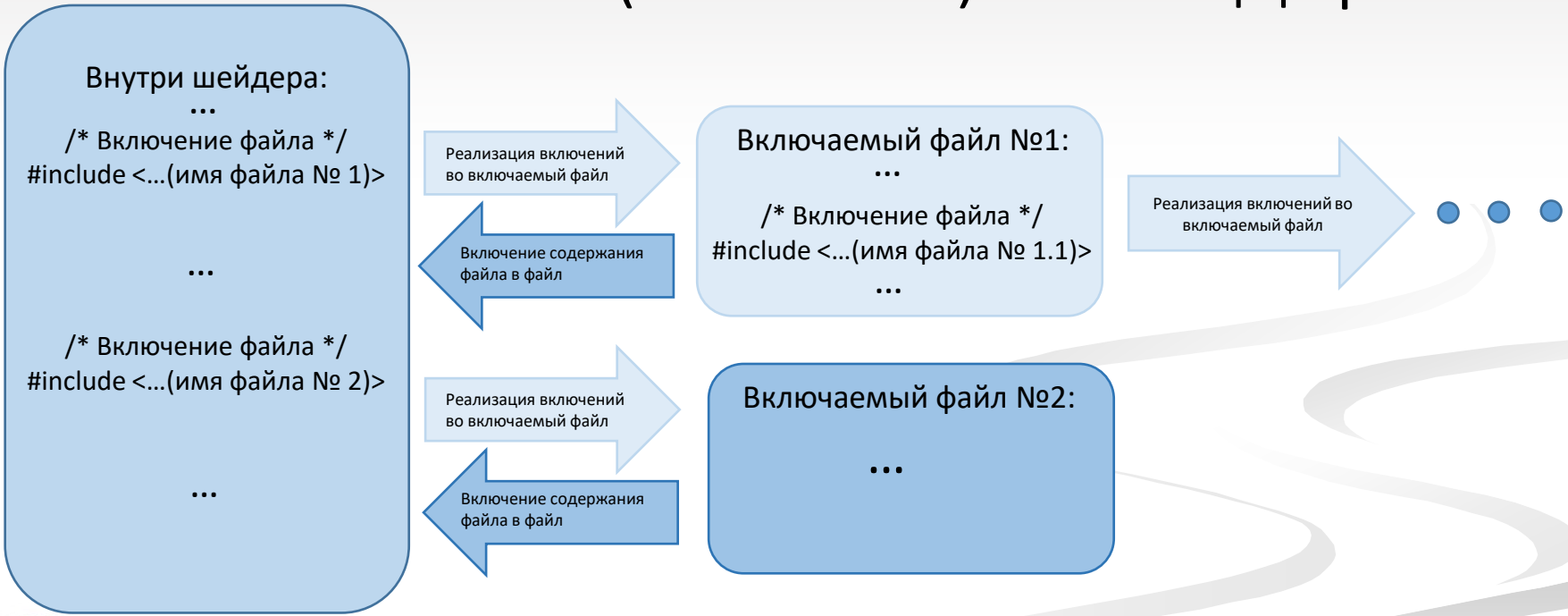
Система обработки шейдеров

Шейдер (shader)

- Шейдер нулевого прохода (pass 0)
- Шейдер первого прохода (pass 1)

- Вершинный шейдер (vertex shader – vert.glsl)
- Геометрический шейдер (geometry shader – geom.glsl)
- Шейдер управления тесселяцией (tessellation control shader – ctrl.glsl)
- Вычислительный шейдер каждой точки тесселяции (tessellation evaluation shader – eval.glsl)
- Фрагментный шейдер (fragment shader – frag.glsl)

Реализация логических включений (includes) в шейдера



Виды шейдеров: фрагментный, вершинный.

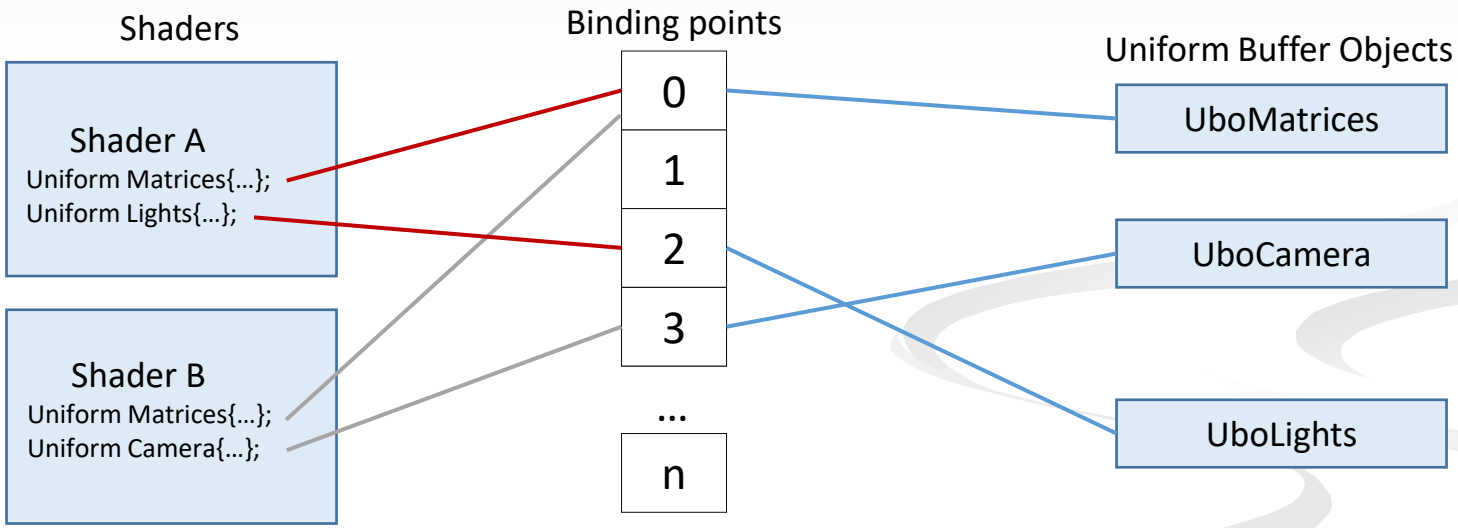
- Вершинный шейдер (vertex shader – vert.glsl)
- Фрагментный шейдер (fragment shader – frag.glsl)
- Геометрический шейдер (geometry shader – geom.glsl)
- Тесселяционные шейдера (tessellation shaders)
  - Шейдер управления тесселяцией (tessellation control shader – ctrl.glsl)
  - Вычислительный шейдер каждой точки тесселяции (tessellation evaluation shader – eval.glsl)

Виды шейдеров: геометрический.

Виды шейдеров: тесселяционный.

Uniform buffer object

Схема передачи на шейдер



Примеры использования

Примеры отправки некоторых необходимых структур на шейдер
и пример нестандартной структуры для передачи таблицы шумов

Данные камеры:
- Матрицы преобразований
- Вектора направлений камеры
- Размеры экрана

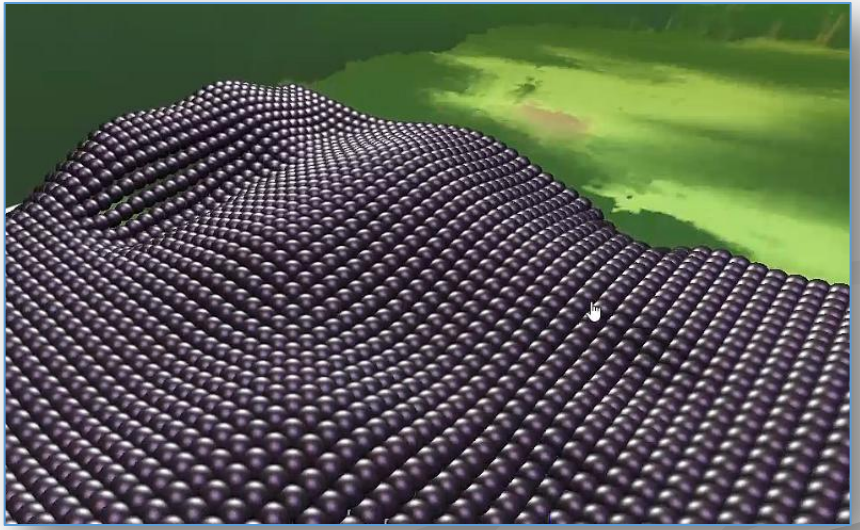
Данные синхронизации:
- Глобальное время
- Время программы
- Изменение времени

Данные материалов:
- Коэффиценты освещнения
- Коэффицент прозрачности

Данные примитива:
- Параметры рендера
- Матрицы преобразований
- Текстурные флаги

Данные для лампочек:
- Позиция лампочки
- Цвет лампочки

Данные для построения динамических гор:
- Массив с таблицей шумов
- Массив перестановок для трехмерного шума



Построение динамических гор

Топология

Тип примитива (Grid/Trimesh)

Выделение памяти под вершины и индексы

Заполнение массива вершин, с помощью параметрических уравнений

Тор, Сфера, Куб, Цилиндр/Конус

Создание примитива из топологии

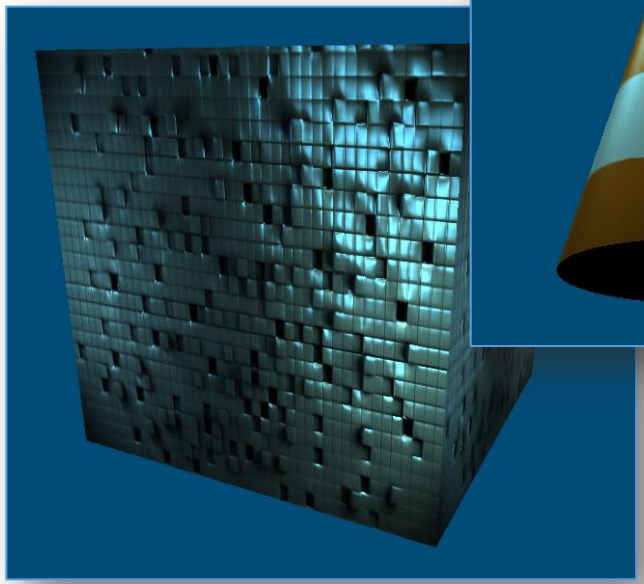
OpenGL буфер точек и индексов

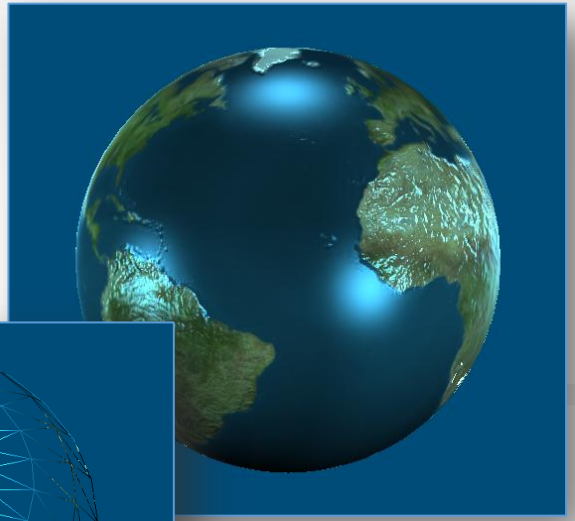
Завершение работы

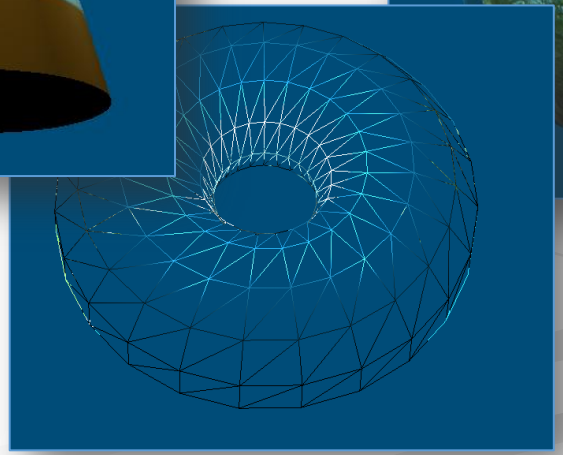
Освобождение памяти

Топология

Примеры фигур







Модели

Вершины

Модель

Материалы

Текстуры

Модуль отложенного освещения

Kp – коэффициент Фонга.
Ks – коэффициент зеркального освещения.
Kd – коэффициент диффузного освещения.
LColor – цвет источника света

$$nl = \vec{N} \cdot \vec{L}$$

$$\overrightarrow{Ks} = \overrightarrow{Lcolor} \cdot \overrightarrow{Ks} \cdot \max\left(0,\ \left(\vec{L} \cdot \vec{R}\right)^{Ph} \cdot att\right)$$

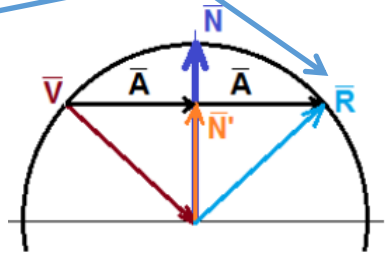
$$\overrightarrow{Kd} = \overrightarrow{Lcolor} \cdot \overrightarrow{Kd} \cdot \max\left(0,\ nl \cdot att\right)$$

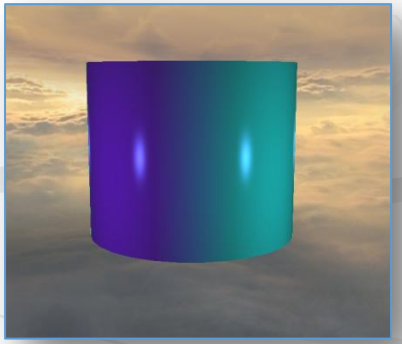
$$\overrightarrow{Color} = \overrightarrow{Ks} + \overrightarrow{Kd}$$

Итоговое наложение

Вектор отражения

$$\vec{R} = -\vec{V} + \left(\vec{V} + \vec{N} \cdot \left(-\vec{V} \cdot \vec{N}\right)\right) \cdot 2 =$$
$$= \vec{V} - \vec{N} \cdot 2 \cdot \left(\vec{V} \cdot \vec{N}\right)$$





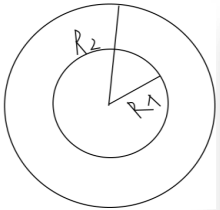
Вывод примитивов → Вывод источников освещения → Итоговый кадр

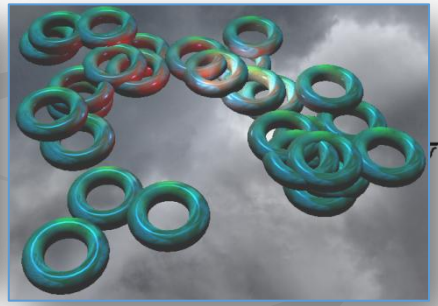
Источники света

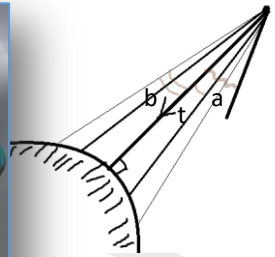
Точечный
`Att -= (Dist - R1) / (R2 -R1)`

Направленный `Att = 1`

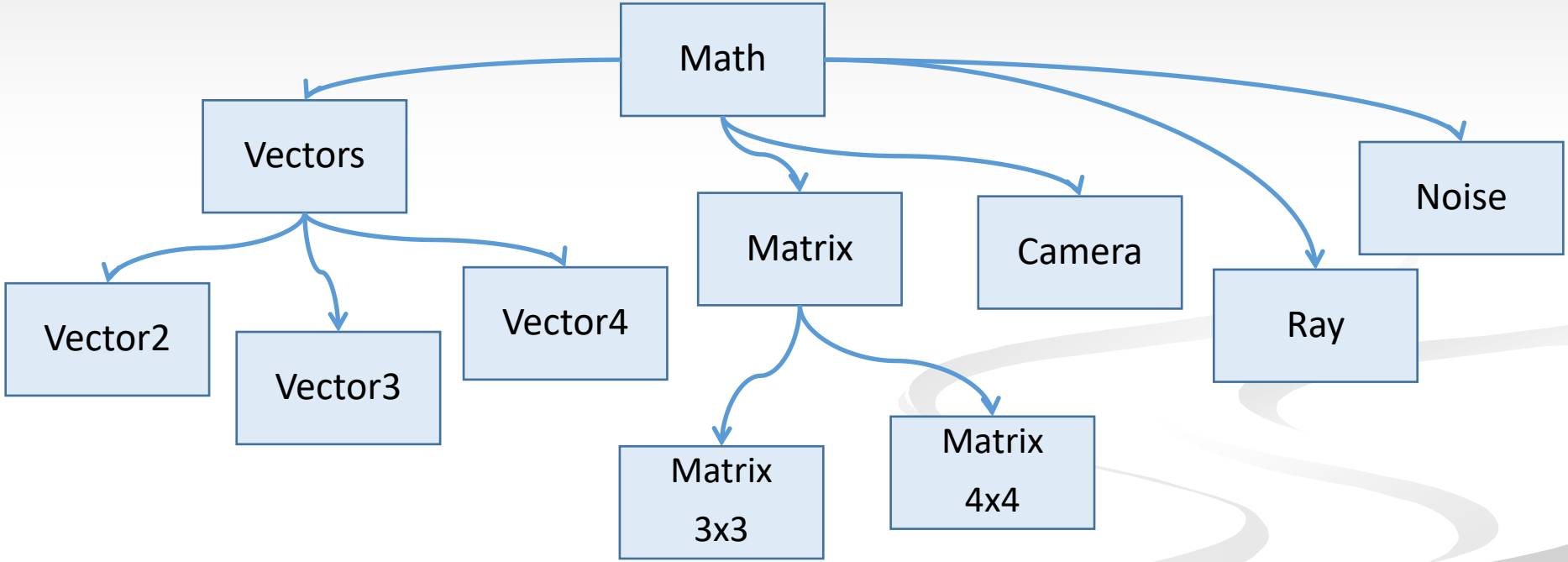
Прожекторный
`Att -= (cos(t) – cos(a)) / (cos(t) -cos(b))`







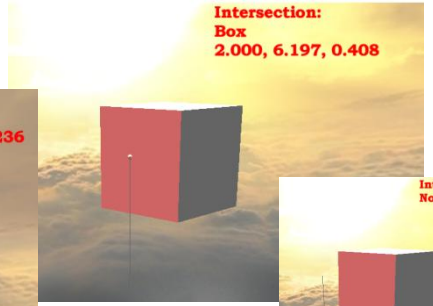
Модуль математики(Math)

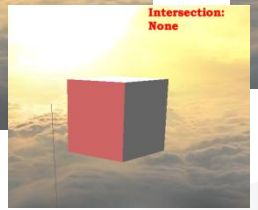


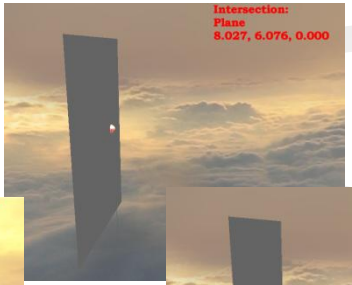
Пересечение(Intersection)

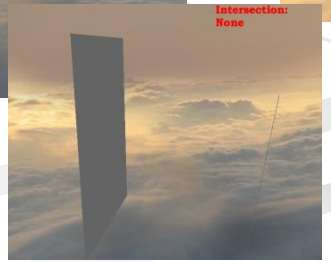
Пересечение

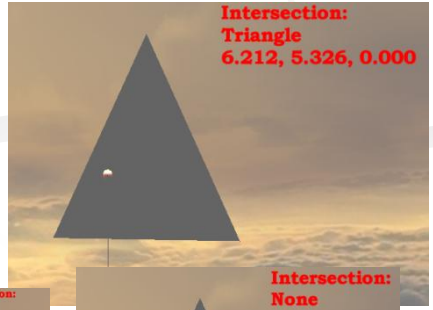
- со сферой (sphere)
- с параллелепипедом (box)
- с плоскостью (plane)
- с треугольником (triangle)

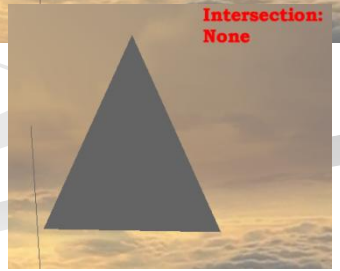












Система вывода шрифтов

Шрифт(.g3df)*

- Имя и размер
- Материал
- Массив примитивов
- Двумерный массив вершин (256 на 4)
- Параметры шрифта (например, смещение до следующей буквы)

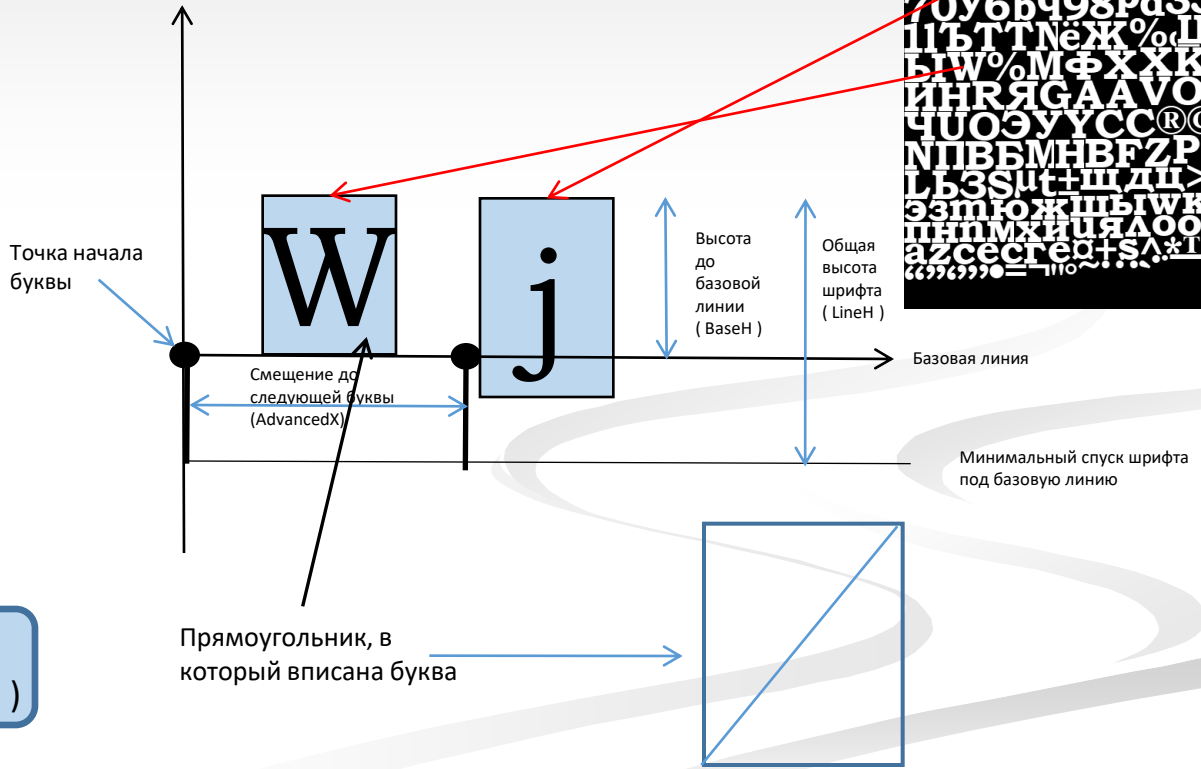
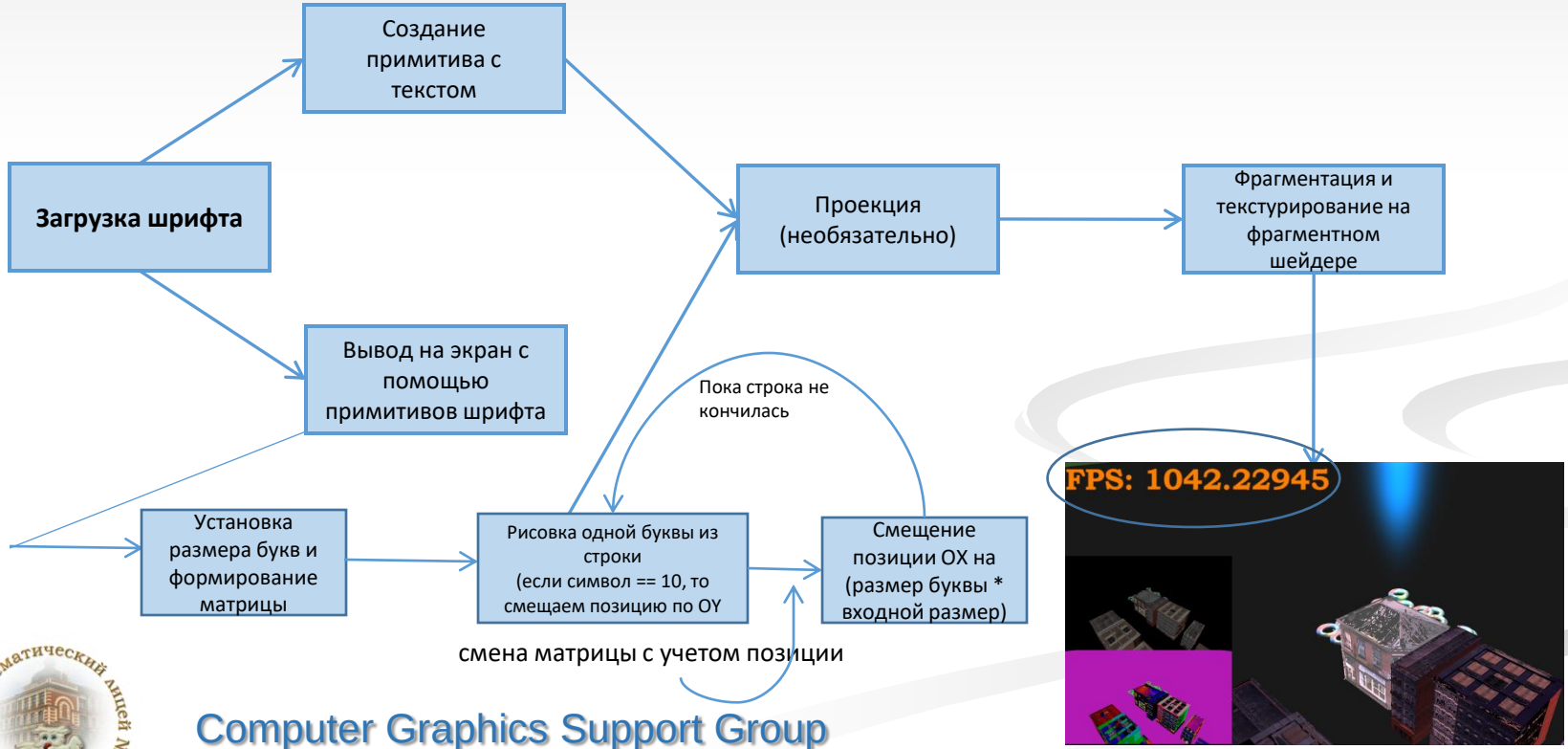


Схема вывода шрифтов

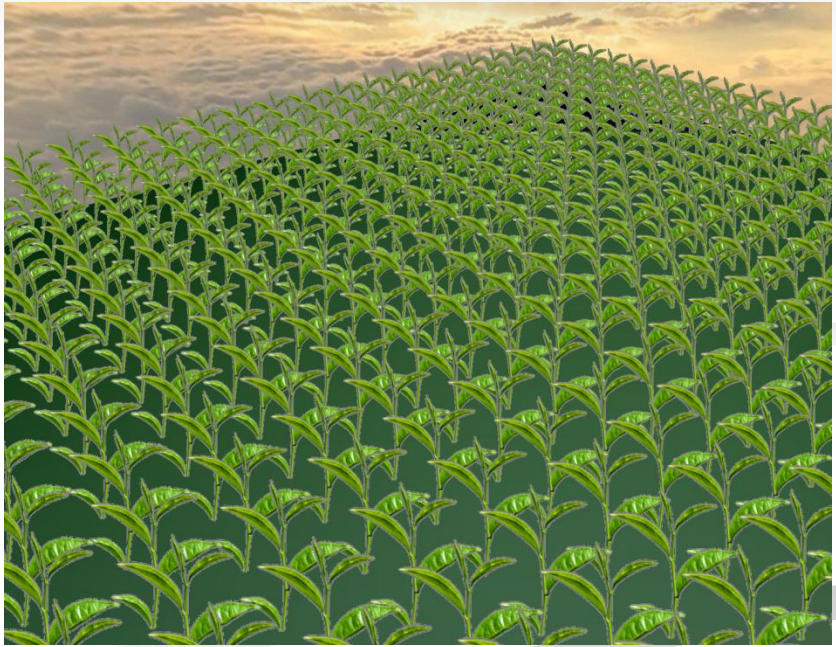


Постеры (объекты, ориентированные на камеру)

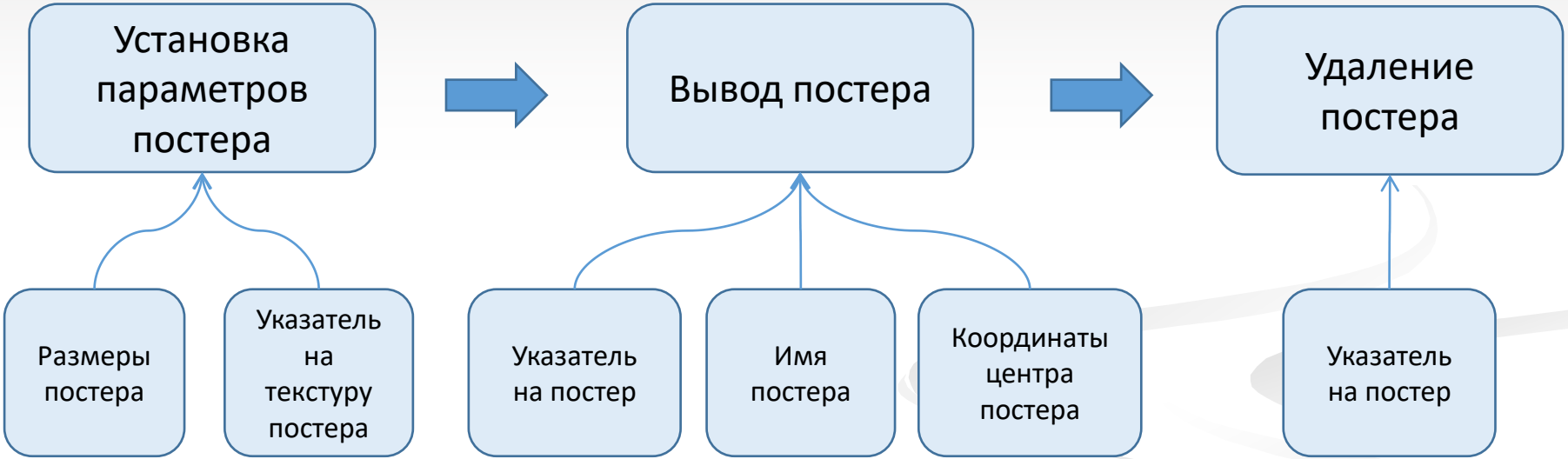
Структура постера (billboard)

Размеры постера: высота и ширина прямоугольника

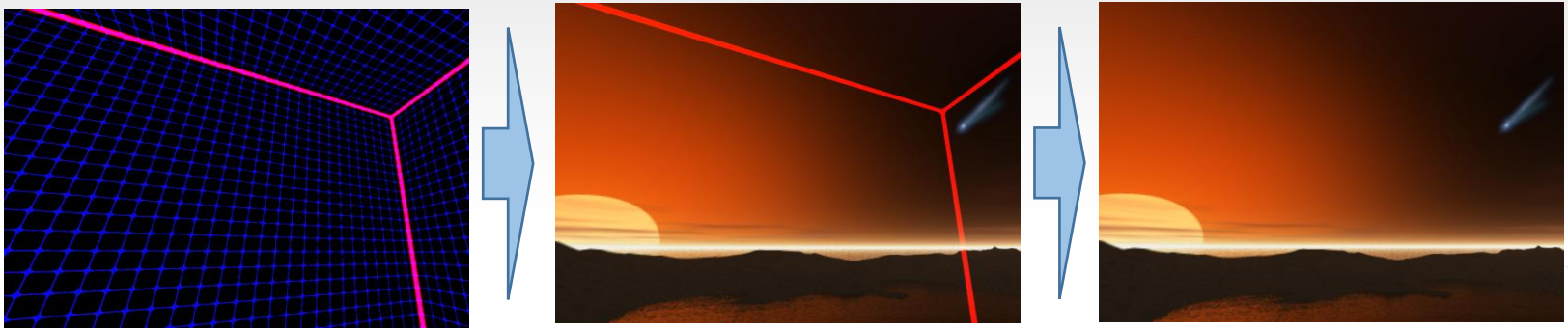
Указатель на текстуру постера

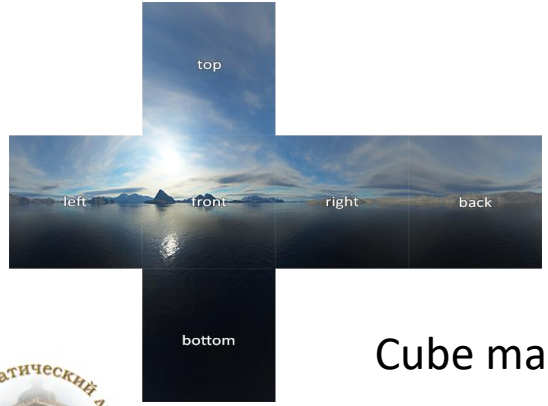


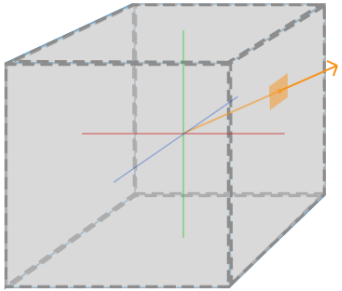
Использование постеров



Skybox







Cube map текстура

Заключение

- Спроектирована полноценная система анимации и рендеринга, система хранения данных, а также собственная библиотека математики для упрощения процесса вычисления.
- Разработано приложение на языке программирования C, работающее с языком шейдеров OpenGL GLSL для переноса вычислений на видеокарту.

Разработка кода

- **Бородулин Федор 08-1 класс**
  система загрузки и вывода фонового изображения сцены по технологии Skybox с использованием Cube Map текстуры
- **Кондаков Александр 08-1 класс**
  система хранения данных (Stock), упрощение доступа к информации в памяти
- **Сопина Елизавета 08-1 класс**
  подсистемы материалов и шаблонов материалов, библиотеки материалов, вывод объектов, ориентированных на камеру
- **Шемякина Елизавета 08-1 класс**
  подсистема работы с моделями, вывод моделей с прозрачными примитивами на экран
- **Львова Диана 08-2 класс**
  обработка использования и загрузки шейдеров на языке GLSL, поддержка логического включения внешних файлов в программы GLSL
- **Чугунов Андрей 09-1 класс**
  подсистемы построения и вывода примитивов, отложенного вывода примитивов на разных фазах рендеринга
- **Смирнов Даниил 09-2 класс**
  подсистемы рендеринга и параметров отрисовки, топологии, создание базовых фигур
- **Дмитриев Иван 09-3 класс**
  реализация системы нескольких типов экранных буферов, поддержка сохранения кадра по плоскостям и G-Buffer
- **Усиков Андрей 09-3 класс**
  функционал синтаксического разбора строк с параметрами, система обработки и вывода шрифтов
- **Бисеров Владислав 09-5 класс**
  модуль пространственной и плоской математики, пересечения с объектами, камера нижнего уровня, шум
- **Молостов Максим 09-5 класс**
  модуль отложенного освещения, визуализация наложений источников света
- **Горляков Даниил 10-5 класс**
  реализация системы анимации и ее активных элементов, создание сцены, работа с WinApi - создание окна
- **Васильев Андрей 10-6 класс**
  система рендера, инициализация и деинициализация всех подсистем программы, отрисовка кадра
- **Конев Данил 10-6 класс**
  отложенный вывод активных элементов анимации, реализация порядко-независимой прозрачности
- **Лановая Александра 10-6 класс**
  подсистема ресурсов рендера, передача необходимых структур на шейдер, с помощью создания буферов
- **Сабитов Кирилл 10-6 класс**
  опрос устройств и таймера, текстурирование объектов, обработка изображений, создание текстур

Разработка кода

- Авторов: 16
- Языки: C/GLSL
  (вершинные, тесселляционные, геометрические, фрагментные шейдера)
- Время разработки: январь 2021-апрель 2021
- Размер кода: 555 KiB
  (56 x *.c – 360 KiB, 35 x *.h – 149 KiB, 43 x *.glsl – 46 KiB)
- Строк кода: 19440
  (*.c – 12000, *.h – 5400, *.glsl – 2040)