

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

**Лабораторная работа №3
по курсу “Компьютерная графика”**

Основы 3D графики

Выполнил: А. Н. Гуляев
Группа: М8О-311Б-23
Преподаватель: В. Д. Бахарев

Москва, 2026

Условие

В этой лабораторной работе должны быть загружены изображения из файла, созданы текстуры из изображения и, используются текстурные координаты, а также сэмплеры для наложения текстур на объекты. Должны быть:

- Созданы текстуры из изображений с использованием `veekay::graphics::Texture`.
- Созданы объекты сэмплера и заданы разумные параметры сэмплирования
- Записаны в набор дескрипторов новые привязки (дескриптор – изображение+сэмплер), чтобы его увидел фрагментный шейдер.

Вершины объектов должны содержать текстурные координаты.

Фрагментный шейдер должен сэмплировать текстуру, используя текстурные координаты, которые были переданы из вершинного шейдера.

Метод решения

Загрузка и создание текстур

JPG текстуры декодируются библиотекой `stb_image` в формат 32-bit RGBA. `VkImage` создаётся с флагами `VK_IMAGE_USAGE_TRANSFER_DST_BIT` | `VK_IMAGE_USAGE_SAMPLED_BIT`. Данные загружаются через `staging buffer` и копируются на GPU.

UV-координаты и интерполяция

UV-координаты задаются в структуре `Vertex`:

```
struct Vertex {  
    veekay::vec3 position;  
    veekay::vec3 normal;  
    veekay::vec2 uv; // UV-координаты текстуры  
};
```

UV-координаты задаются в диапазоне `[0, 1]` для каждой вершины:

`createCubeGeometry()`: `{0.0f, 0.0f}` — левый нижний угол, `{1.0f, 0.0f}` — правый нижний, `{1.0f, 1.0f}` — правый верхний, `{0.0f, 1.0f}` — левый верхний

В vertex shader UV интерполируются между вершинами треугольника и передаются в fragment shader:

```
f_uv = uv;
```

Combined Image Sampler в Descriptor Set

Descriptor binding 3 объединяет:

VkDescriptorImageInfo + VkSampler:

```
VkDescriptorImageInfo image_info {mat.sampler, tex→view,  
VK_IMAGE_LAYOUT_SHADER_READ_ONLY_OPTIMAL};
```

В fragment shader:

```
layout(binding = 3) uniform sampler2D albedoTexture;
```

Множественные текстуры и Material

Структура Material:

```
struct Material {  
    veekay::graphics::Texture* albedo_texture = nullptr;  
    VkSampler sampler = VK_NULL_HANDLE;  
    VkDescriptorSet descriptor_set = VK_NULL_HANDLE;  
};
```

Фоновая текстура

Background плоскость (6×4 единиц) с текстурой land.jpg, рендерится после основных объектов для эффекта заднего плана.

Результаты

На рисунках ниже показана трехмерная сцена, полученная в результате выполнения лабораторной работы:



Выводы

Я научился загружать изображения из файла, создавать текстуры из изображений и использовать текстурные координаты, а также сэмплеры для наложения текстуры на объекты.