

# Мультимедиа

## Лекция №5

Рябинин Константин Валентинович

e-mail: [icosaeder@ya.ru](mailto:icosaeder@ya.ru)

jabber: [icosaeder@jabber.ru](mailto:icosaeder@jabber.ru)

Пермь, 2013

## ● Для типов существуют квалификаторы точности

- lowp – низкая точность
- mediump – средняя точность
- highp – высокая точность

## ● Скалярные (базовые типы)

### ● float – вещественное число:

- lowp float:  $[-2; 2]$    mediump float:  $[-2^{14}; 2^{14}]$    highp float:  $[-2^{62}; 2^{62}]$

### ● int – целое число

- lowp int:  $[-2^8; 2^8]$    mediump int:  $[-2^{10}; 2^{10}]$    highp int:  $[-2^{16}; 2^{16}]$

### ● bool – логическое значение

### ● Особенности:

#### ● Отсутствие неявных приведений типа:

```
float a = 1; // порождает ошибку  
float b = 1.0, c = float(1);
```

#### ● Тип int не всегда поддерживается аппаратно (в общем случае – обёртка над float), поэтому результат переполнения, вообще говоря, не определён

#### ● Отсутствуют побитовые операции

#### ● Тип bool – обёртка над int, то есть, в общем случае, так же обёртка над float

## Векторы

- `vec2`, `vec3`, `vec4` – вещественные вектора на 2, 3 и 4 компоненты

- `ivec2`, `ivec3`, `ivec4` – целочисленные вектора на 2, 3 и 4 компоненты

- `bvec2`, `bvec3`, `bvec4` – логические вектора на 2, 3 и 4 компоненты

- Особенности:

- Реализована перегрузка операций сложения и вычитания векторов, причём код оптимизируется и выполняется GPU за  $O(1)$

- Инициализация вектора может быть осуществлена при помощи конструкторов вида

```
vec3 a = vec3(0.1, 0.2, 0.3);
```

```
vec4 b = vec4(a, 0.4);
```

```
vec2 c = vec2(a); // будут взяты первые две компоненты
```

- Для доступа к компонентам можно использовать индекс:

```
vec3 a; a[1] = 0.5;
```

либо мнемонические поля (`x`, `y`, `z`, `w` ~ `r`, `g`, `b`, `a` ~ `s`, `t`, `p`, `q`):

```
vec3 a; a.y = 0.5;
```

- Существуют вспомогательные поля, предоставляющие доступ к любому подмножеству значений в любой последовательности:

```
vec3 a, b;
```

```
a.xy = b.zy = vec2(0.5, 0.8);
```

```
vec3 c = vec3(0.9, a.xy);
```

- Мнемоника полей существует лишь для пользователя, представляя собой обёртку доступа к данным

## ● Матрицы

- `mat2` – вещественная матрица 2x2

- `mat3` – вещественная матрица 3x3

- `mat4` – вещественная матрица 4x4

- Особенности:

- Реализована перегрузка операций сложения, вычитания и умножения матриц

- Реализована перегрузка операции умножения матрицы на вектор

- Матрицы хранятся по столбцам и могут быть рассмотрены как массив векторов-столбцов

- Как правило, матрицы приходят в шейдер из основной программы и используются для произведения аффинных преобразований

## ● Дискретизаторы – специализированные структуры данных для доступа к текстурам

- sampler1D – предоставляет доступ к одномерной текстуре
- sampler2D – предоставляет доступ к двумерной текстуре
- sampler3D – предоставляет доступ к трехмерной текстуре
- samplerCube – предоставляет доступ к кубической текстуре

### ● Особенности:

- Дискретизатор приходит в шейдер извне через *uniform*-переменную и не может быть изменён внутри шейдера
- Используется для доступа к текстуре
- Для извлечения данных из дискретизатора используются специализированные функции, например:

```
// fragment shader
uniform sampler2D tex;
void main()
{
    vec4 color = texture2D(tex, gl_TexCoord[0].st);
    gl_FragColor = color;
}
```

## ● Структуры

● `struct Light { vec3 position; vec3 color; };`

### ● Особенности:

- Структуры, фактически, полностью идентичны структурам в C
- *union* и *enum* зарезервированы в качестве ключевых слов, но пока не поддерживаются

## ● Массивы

● `float a[10];`  
`vec4 points[5];`

### ● Особенности:

- Можно объявлять массивы любых типов
- Массивы являются статическими

## ● Void – тип для функций, не возвращающих значения

→ Более никаких типов в GLSL нет; динамическое выделение памяти (указатели) не поддерживается; строки и абстрактные типы не предусмотрены

- Перегрузка операций для векторных и матричных типов данных
- Функции над векторами:
  - `dot` – скалярное произведение
  - `normalize` – нормирование вектора
  - `reflect` – отражение вектора относительно вектора
  - `refract` – преломление вектора относительно вектора с коэффициентом преломления
  - `length` – длина вектора
  - `distance` – расстояние между двумя точками
  - ...

- **Функции над матрицами:**
  - **determinant** – определитель матрицы
  - **transpose** – транспонированная матрица
  - **inverse** – обратная матрица
  - ...
- **Тригонометрические функции**
  - **sin, cos, tan** – функции
  - **asin, acos, atan** – аркфункции
  - **radians, degrees** – перевод из градусов в радианы и обратно
  - ...
- **Гиперфункции**
  - **sinh, cosh, tanh** – функции
  - **asinh, acosh, atanh** – аркфункции
  - ...



- Математические функции

- pow – возведение произвольную в степень

- exp – экспонента

- log – натуральный логарифм

- sqrt – квадратный корень

- clamp – ограничение значения

- abs – модуль

- ceil, floor, round – округление в разные стороны

- sign – сигнум

- min, max – минимум, максимум

- ...

- ...