

Лекция 3. OpenGL — векторные преобразования

Илья Макаров

ИТМО

28 сентября 2022

Санкт-Петербург

Вектора

Самые "распространенные/используемые" алгебраические структуры:

- **вектора** - направленные отрезки (размерности 2, 3 или 4);
- **матрицы** - таблицы (как правило квадратные), размерности 2×2 , 3×3 , 4×4 .

Операции:

- **скаляр-вектор**
- **вектор-вектор**
- **матрица-вектор**

Операции со скалярами

Примеры:

- Умножение на скаляр: $(x, y, z) \cdot \alpha = (\alpha \cdot x, \alpha \cdot y, \alpha \cdot z)$
- Сложение скаляра: $(x, y, z) + \alpha = (\alpha + x, \alpha + y, \alpha + z)$
- Обратный вектор: $-(x, y, z) = (-x, -y, -z)$
- ...

Операции с векторами и матрицами

Примеры:

- Сложение: $(x, y, z) + (a, b, c) = (x + a, y + b, z + c)$
- Длина: $len(x, y, z) = \sqrt{x^2 + y^2 + z^2}$
- Нормализация: $normalize(x, y, z) = \frac{(x, y, z)}{len(x, y, z)}$
- Скалярное произведение: $(x, y, z) \cdot (a, b, c) = x \cdot a + y \cdot b + z \cdot c$
- Векторное произведение:
 $(x, y, z) \times (a, b, c) = (y \cdot c - z \cdot b, z \cdot a - x \cdot c, x \cdot b - y \cdot a)$
- Умножение: $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times (x, y) = (a \cdot x + c \cdot y, b \cdot x + d \cdot y)$
- ...

Полезные преобразования

- $$\begin{bmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \cdot sx \\ y \cdot sy \\ z \cdot sz \\ 1 \end{bmatrix}$$

- $$\begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + dx \\ y + dy \\ z + dz \\ 1 \end{bmatrix}$$

- $$\begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & \cos(\alpha) & -\sin(\alpha) & dy \\ 0 & -\sin(\alpha) & \cos(\alpha) & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \cdot \cos(\alpha) - z \cdot \sin(\alpha) \\ y \cdot \sin(\alpha) + z \cdot \cos(\alpha) \\ 1 \end{bmatrix}$$

- + комбинации преобразований (**важен порядок!**)

На практике

Упрощение работы с математикой:

- Средства **QT**:

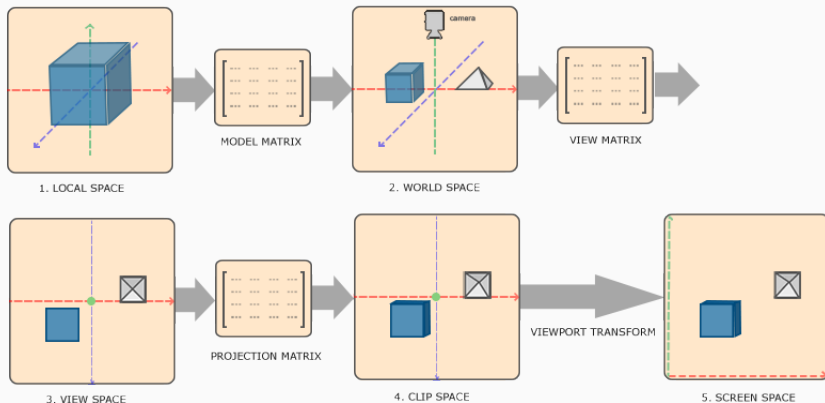
```
QMatrix4x4 matrix;  
matrix.perspective(60.0, 4.0 / 3.0, 0.1, 100.0);  
matrix.translate(0, 0, -2);  
matrix.rotate(90., rotation_axis);
```

- Библиотека **GLM**

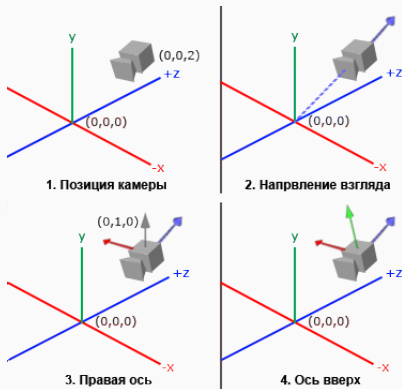
```
glm::mat4 trans;  
auto pos = glm::vec3(0.0, 0.0, 1.0);  
trans = glm::rotate(trans, 90.0, pos);  
trans = glm::scale(trans, glm::vec3(0.5, 0.5, 0.5));
```

- Другие библиотеки...

Еще раз о преобразовании координат



Камера



Задается 3 векторами, которые образуют новую СК, начало СК совпадает с положением камеры.

Как задать камеру

Положение

```
glm::vec3 cameraPos = glm::vec3(0.0f, 0.0f, 3.0f);
```

Направление взгляда

```
glm::vec3 cameraTarget = glm::vec3(0.0f, 0.0f, 0.0f);  
glm::vec3 cameraDirection = glm::normalize(cameraPos  
    - cameraTarget);
```

Вектор вправо

```
glm::vec3 up = glm::vec3(0.0f, 1.0f, 0.0f);  
glm::vec3 cameraRight = glm::normalize(  
    glm::cross(up, cameraDirection));
```

Вектор вверх

```
glm::vec3 cameraUp = glm::cross(cameraDirection,  
    cameraRight);
```

lookAt

Математика преобразований:

$$\begin{bmatrix} R_x & R_y & R_z & 0 \\ U_x & U_y & U_z & 0 \\ D_x & D_y & D_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} = [\dots] - \text{look at matrix}$$

В коде же:

```
glm::mat4 view;  
view = glm::lookAt(glm::vec3(0.0f, 0.0f, 3.0f),  
    glm::vec3(0.0f, 0.0f, 0.0f),  
    glm::vec3(0.0f, 1.0f, 0.0f));
```