На каком языке пишут шейдеры?

OpenGL Shading language → Си-подобный язык.

Структура
1. Описание шейдера начинается с версии.
2. Входные, выходные и глобальные переменные.
3. Функция main

```
#version version_number        → 330 core
in type in_variable_name;      → вершинные атрибуты GL_MAX_VERTEX_ATTRIBS 16 4х компонент
out type o_variable_name;
uniform type uniform_name;     → глобальные переменные.
in                             → встроенные переменные
void main()
{
  o_variable_name = i_var_name;
  gl_Position=
}
```

type → известные из С int, float и т.д. int, float, double, uint, bool
новые для С vector и matrix

_vecn .xyzw | rgba | stpq | [i], где i = 0,n   синтаксис Swizzling
prefixes b, i, ui, d

Обращение к значениям вектора 2/3 точки, можно получить вектор xxyx
а также выполнять операции, например, сложение, скалярное умножение
транспонирование матриц

Вершинный шейдер принимает входные значения напрямую из вершинных данных. Для этого нужно указать метапозиции (с помощью спецификатора layout (location=0) или использовать вызов glGetAttributeLocation

uniform ⇔ глобальные переменные
уникальна для каждого шейдера, доступ к ней есть на любом этапе,
если вы объявляете форму.

built-in variables / встроенные перем

```
for example, vertex
  in int gl_VertexID
  out gl_PerVertex{
    vec4 gl_Position;
    float gl_PointSize;
    ~~float gl_ClipDistance~~
  }
```

Имитация камеры, двигать объекты в противоположном направлении.

FPS (First Person Shooter)

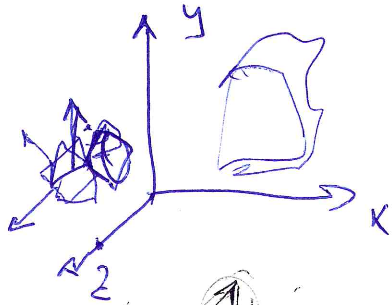
Линейная алгебра процесс Грама-Шмидта

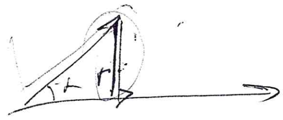
$L = LB(a_1, \dots, a_n) \to L = LB(b_1, \dots, b_k)$, где $b_i \perp b_j \ \forall i \neq j$

Процесс ортогонализации

$b_1 = a_1$, $\dfrac{\langle a_1, b_1 \rangle}{\langle b_1, b_1 \rangle} a_1$

$b_2 = a_2 - pr_{b_1}(a_1)$





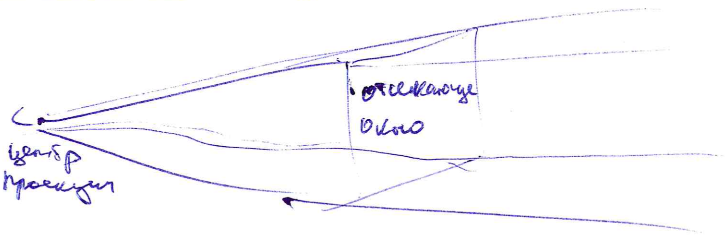
$\langle a_i, b_i \rangle = |a_i||b_i|\cos\alpha$

$pr_{b_1}(a_1) =$

Matrix LookAt

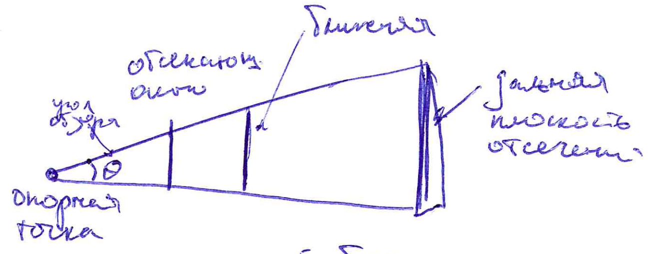
$$\begin{bmatrix} R_x & R_y & R_z & 0 \\ U_x & U_y & U_z & 0 \\ D_x & D_y & D_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & P_x \\ & 1 & & P_y \\ & & 1 & P_z \\ & & & 1 \end{bmatrix}$$

Объем наблюдения для перспективной проекции



все что вне → удаляется

~~рис.~~ пирамида зрения



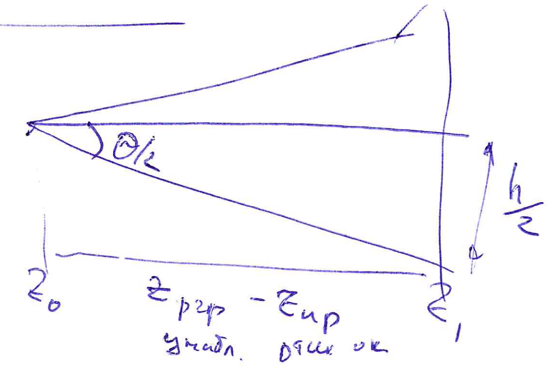
пирамидальный объем
наблюдения

Связь угла обзора с размерами отсечен. окна

$$|z_\phi - z_0| = \frac{h}{2}\cdot ctg\left(\frac{\theta}{2}\right)$$

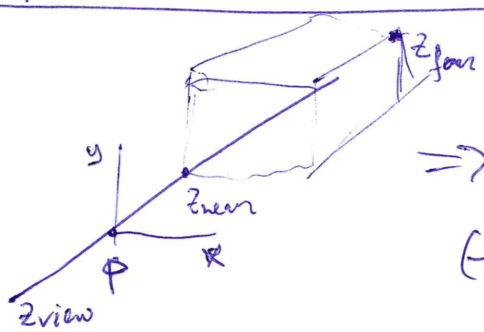
$$|\Delta z| =$$

$$\Delta z = \frac{h}{2}\, ctg\left(\frac{\theta}{2}\right)$$

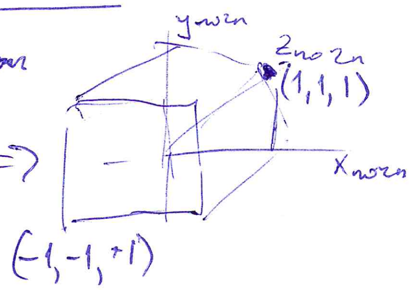


Нормированные координаты перспективной проекции

нормиров. объем наблюдения (стр. 555)



⇒



$$\begin{bmatrix} \frac{ctg\left(\frac{\theta}{2}\right)}{x_{вр.ргп}} & & \\ & ctg\left(\frac{\theta}{2}\right) & \\ & & z_{норм} \end{bmatrix}$$

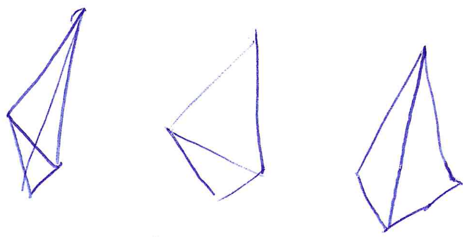
Херн
стр. 500

②

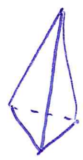
Упорядочивание по глубине (depth cueing)

частицы пыли (прозрачность, тусклость)
параметры:
1) Яркость
2) цвет объекта

Определение видимых линий и поверхностей



штриховая

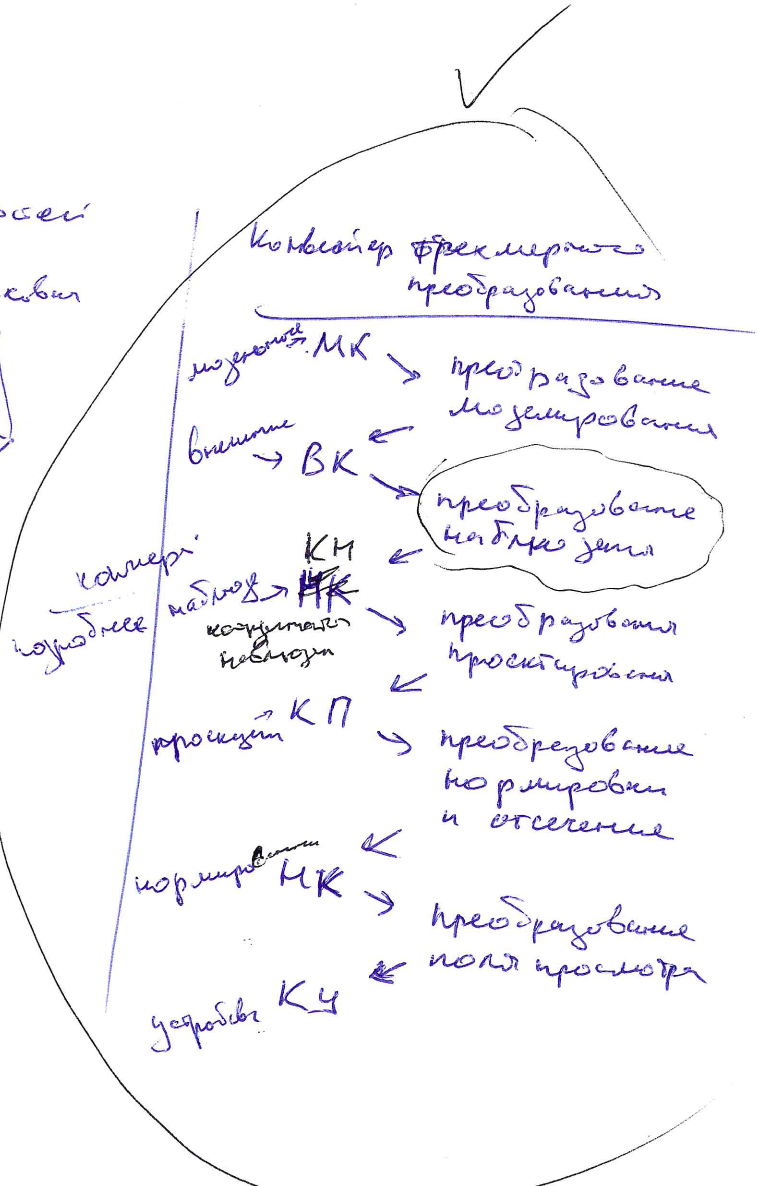


Визуализация поверхностей
1) цвет
2) ~~повер~~ источники освещения
~~3)~~ 2) фоновое освещение
точечные
3) поверхностные св-ва
) прозрачность
) шероховатость

стереоскопическая проекция

Конвейер наблюдения

Конвейер ~~трехмерного~~ преобразования



Херн (0)

двумерное поле просмотра

преобразования	результат	
Преобразование моделирования	модельные координаты	совокупность моделей
П. наблюдения	внешние	
п. проектирования	координ. наблюдения	камера
п. нормировки и отсечения	к. проекции	парал. экрану вид проекции
п. поля просмотра	нормированные к.	отсечение (отображаемый объем (наблюд.)
	к. устройства.	

МК

ВК

конвейер трехмерного преобразования

левосторонняя с-ма к-нат для отображ. глубины точек в нормализ. координатах.

важно

(левосторонняя сист. координат)

Параметры наблюдения

камера глаз look at point

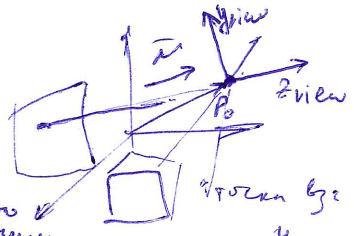
(0) выбирается точка наблюд. (вектор, глаз, положение) $P_0(x_0, y_0, z_0)$ (view up point)

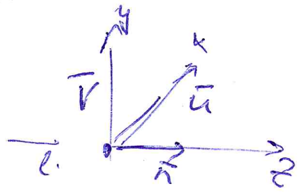
(2) задаётся вектор верха изображ. (yview) $\bar{V}$ не должно быть параллельно

остается вектор/направление наблюдения (zview) $\bar{N}$

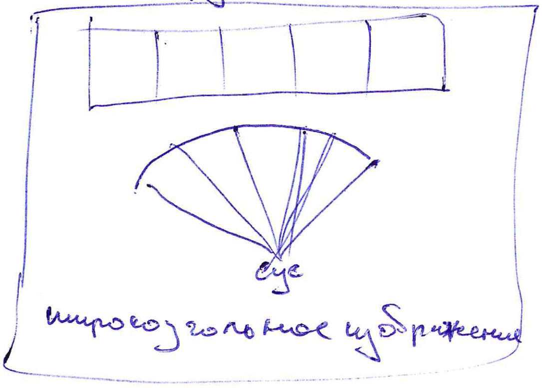
(1) сначала задается

Вектор нормали к плоскости наблюдения $\bar{N}$ обычно отрицат. -ый.





С-ма координат и КП



$$n = \frac{N}{|N|} = (n_x, n_y, n_z)$$

$$u = \frac{(V \times n)}{|V|} = (u_x, u_y, u_z)$$

$$v = (n \times u) = (v_x, v_y, v_z)$$

Общая схема преобразования

модельные координаты (МК) $\xrightarrow{\text{преобраз. моделиров.}}$ ~~внешние~~ ~~мировые~~ координаты ВК $\xrightarrow{\text{преобразов. наблюдения}}$ ~~преобразования~~ координаты наблюдения НК

Параметры наблюдения. | В компьютерной графике используют левостороннюю систему координат

1. Выбирается точка наблюдения $P_0(x_0, y_0, z_0)$ ← lookatpoint
2. Направление наблюдения $\vec{N}$ (~~точка взгляда~~) → вектор нормали к плоскости наблюдения
3. Устанавливается вектор верха прибл. V

нужно нормализовать, т.е. необх. нормировать

$\to n = \frac{N}{|N|} = (n_x, n_y, n_z)$

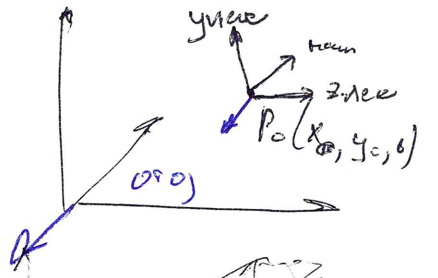
$\to U = \frac{[V \times n]}{|V|} = (u_x, u_y, u_z)$

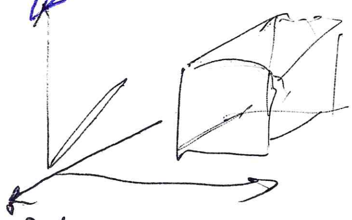
вектор верха $\to V = [n \times U] = (v_x, v_y, v_z)$

$n = 1-2$

$V =$ задан

$U - ?$



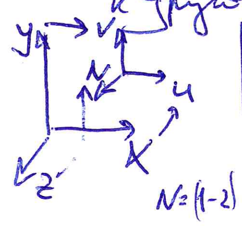


uvn-matrix

$$A_{uvn} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_{view} = A_T \times A_{uvn}$$

$$A_T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ P_x & P_y & P_z & 1 \end{bmatrix}$$

Таким образом, получаем переход от одной с-мы коорд. к другой с-ме коорд.



$N = (1-2)$

часто, также используется
1. ~~точка~~ камера
2. точка куда смотрим
3. вектор верха

нормирование