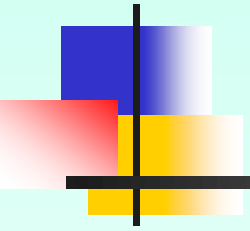
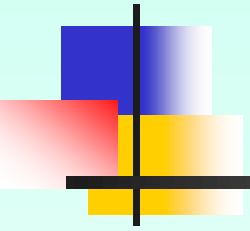


COMPUTER GRAPHICS

ЗАСОБИ ПРОГРАМУВАННЯ КОМП'ЮТЕРНОЇ ГРАФІКИ

Лек. 14 2021 ІПЗ-19



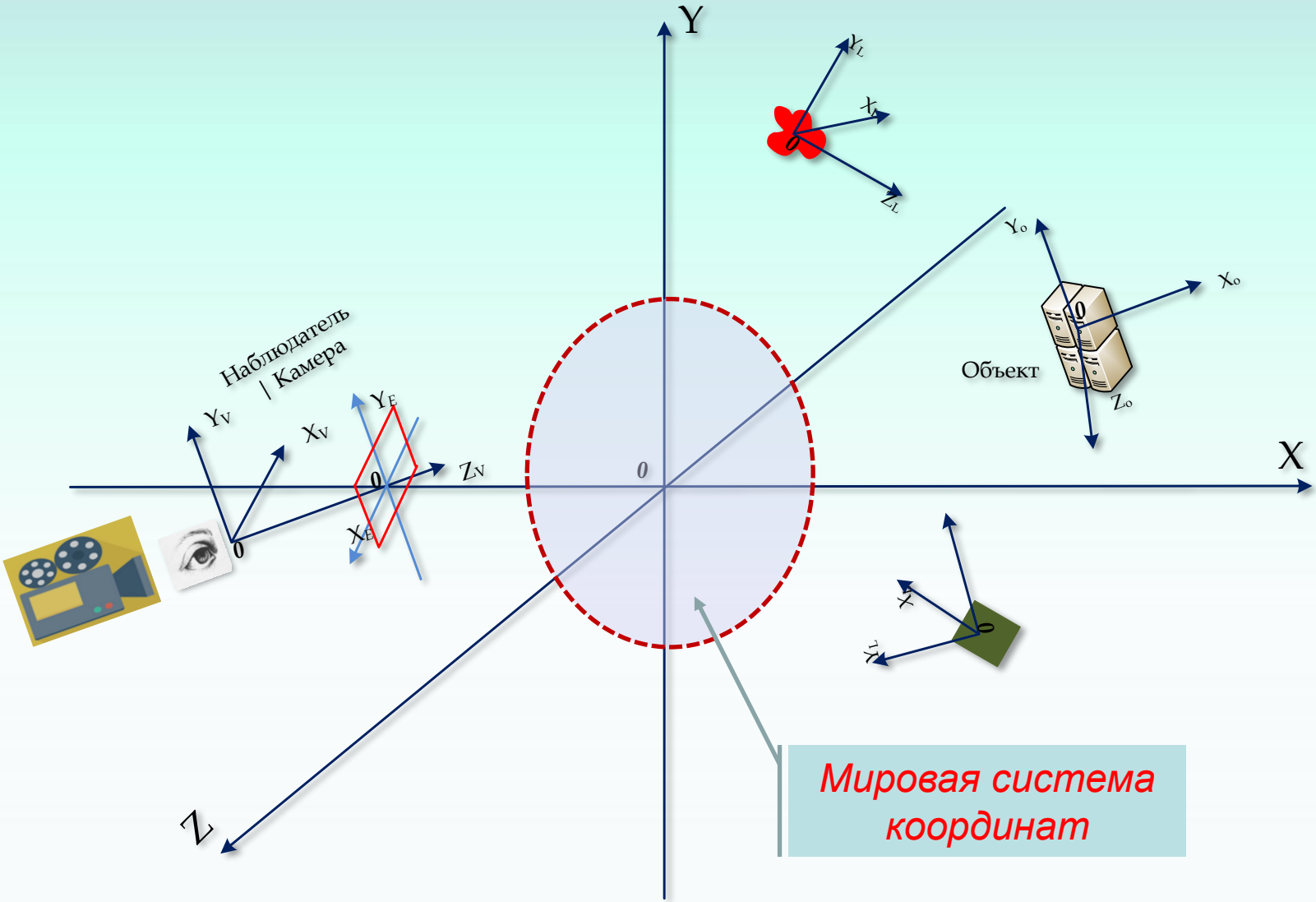
COMPUTER GRAPHICS

OPEN_GL (part 4)

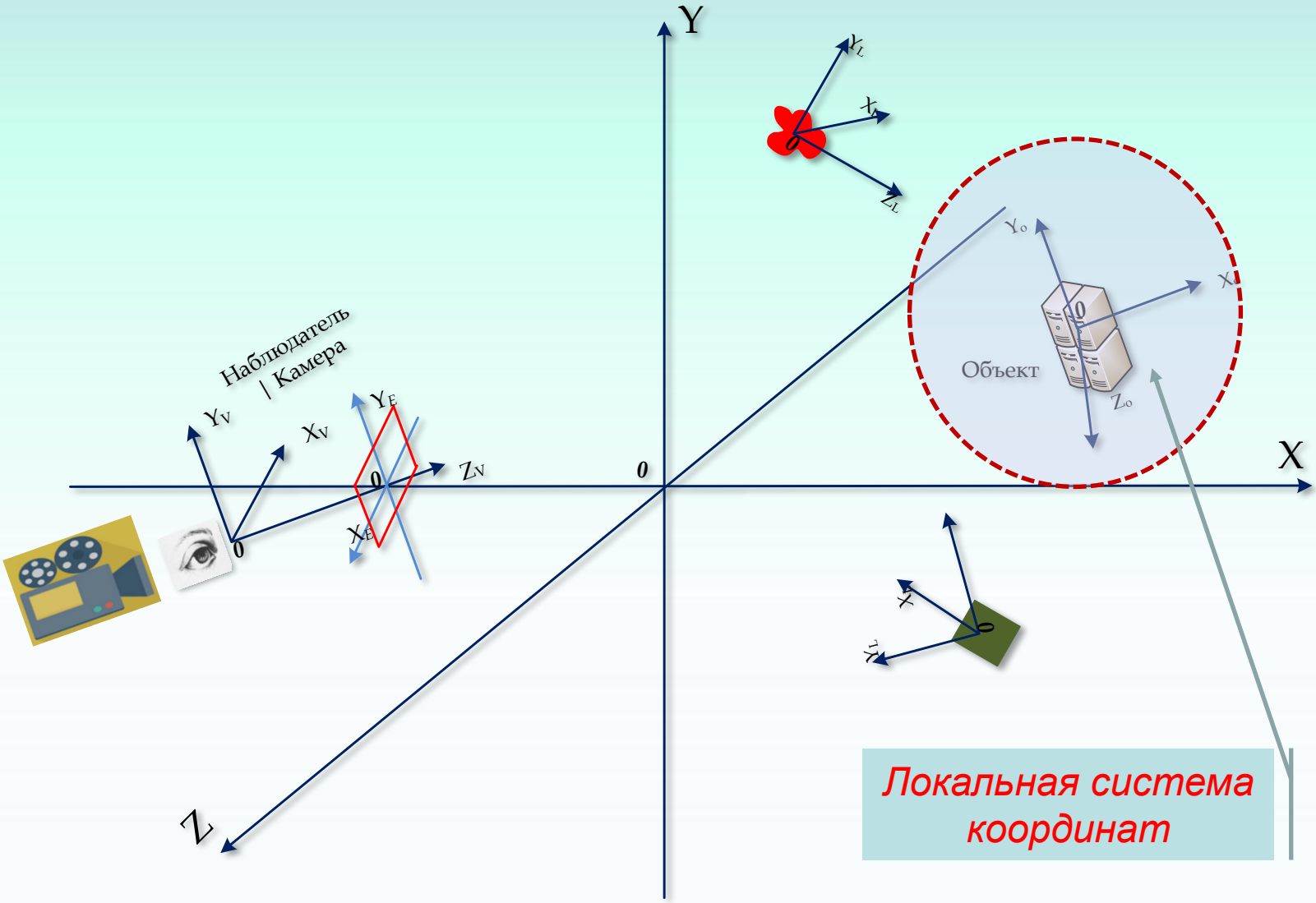
Open GL. 3D Преобразования

- Системы координат.
- Преобразование модели.
- Преобразование вида.
- Преобразование проецирования.
- Примеры.

Open GL. СЦЕНА

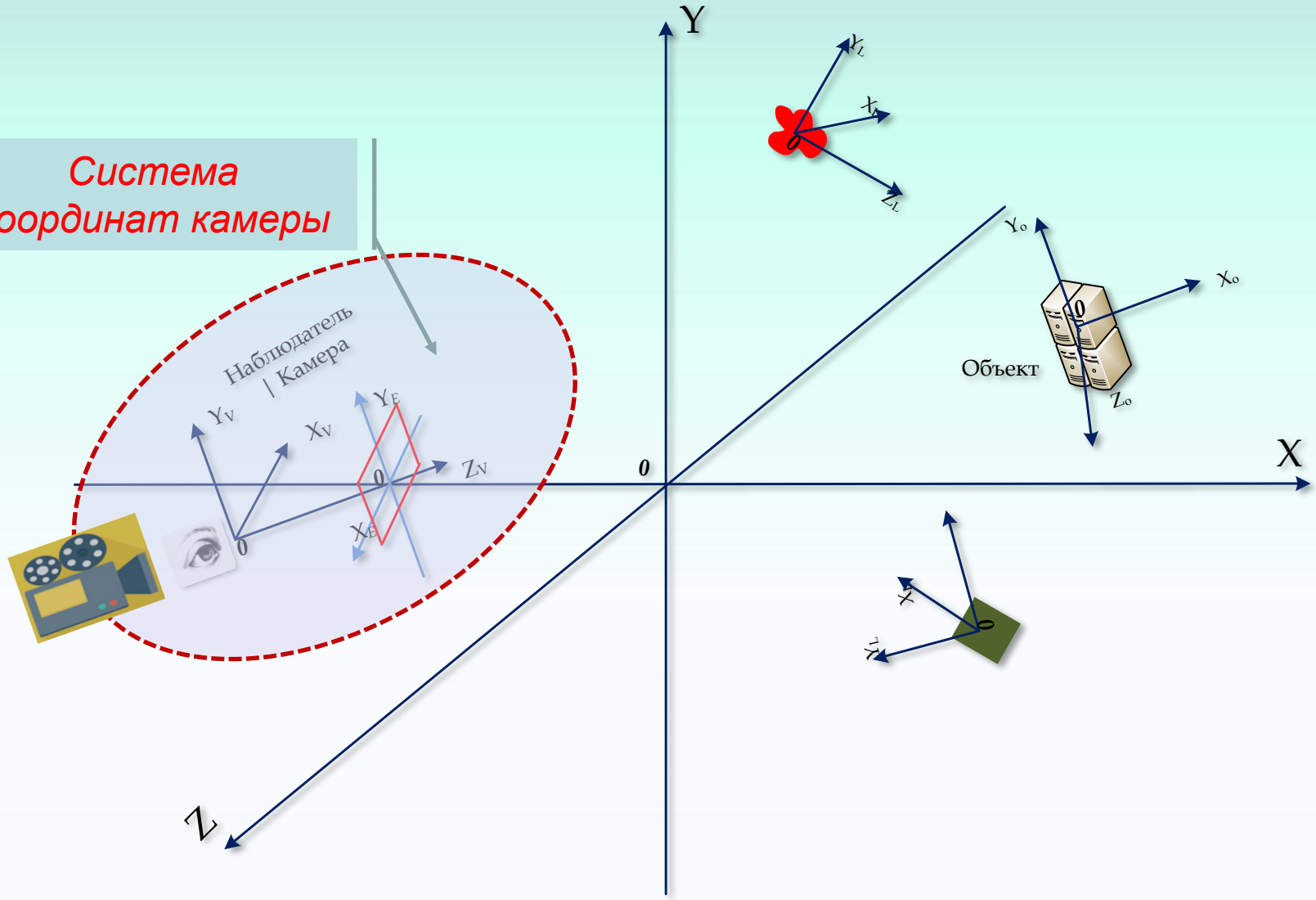


Open GL. СЦЕНА

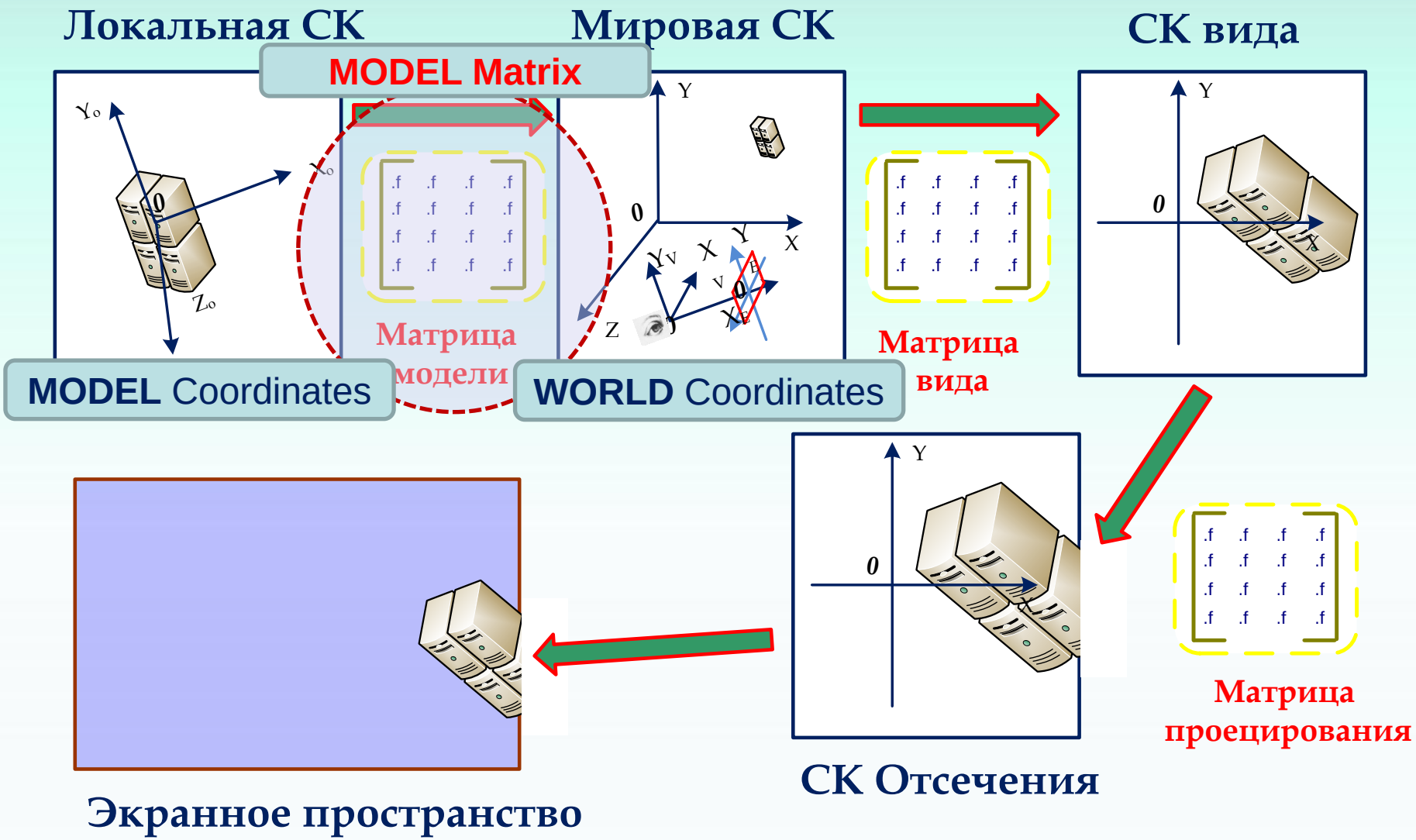


Open GL. СЦЕНА

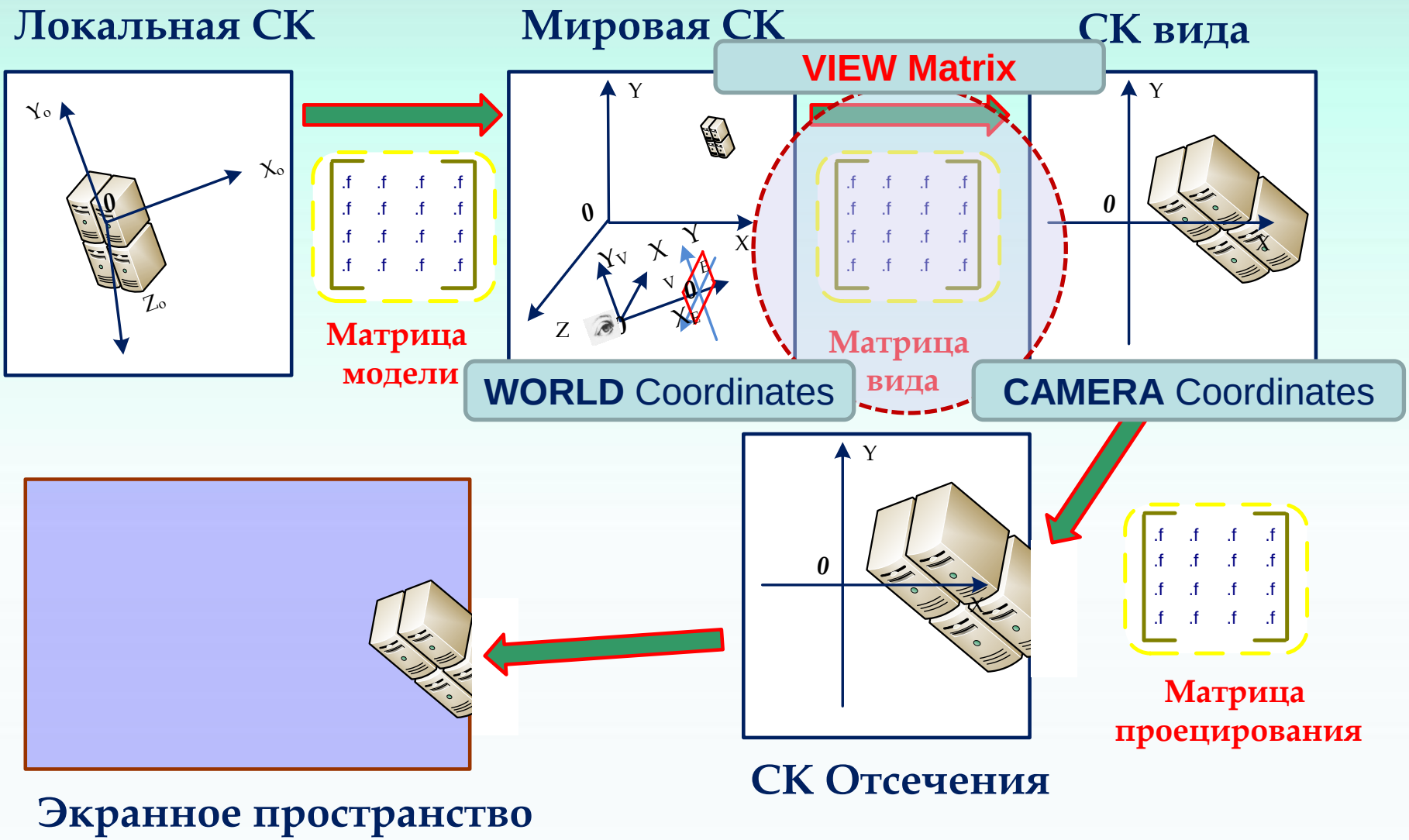
Система координат камеры



Open GL. 1-й шаг

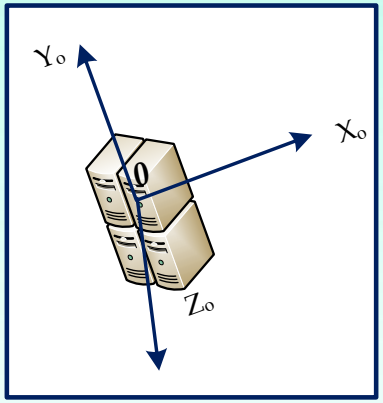


Open GL. 2-й шаг



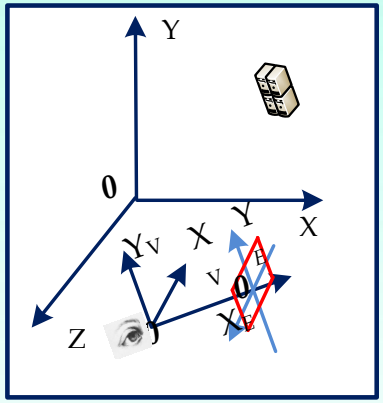
Open GL. 3-й шаг

Локальная СК



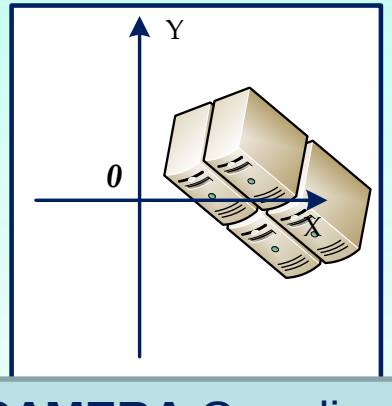
Матрица модели

Мировая СК

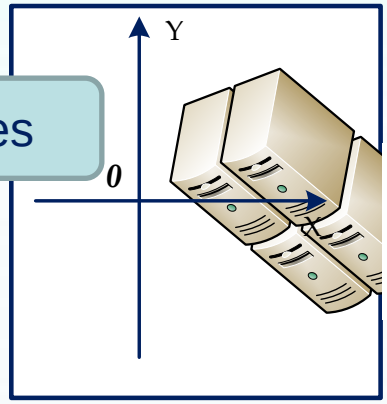


Матрица вида

СК вида



CAMERA Coordinates

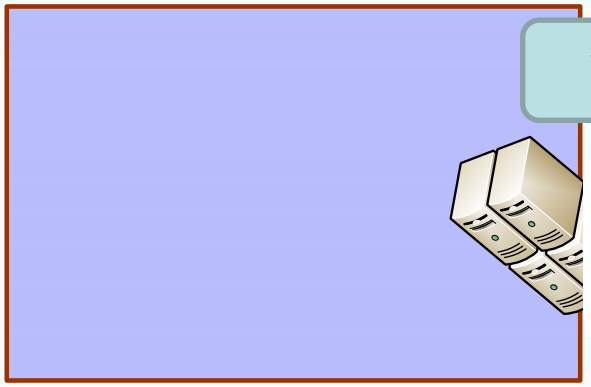


СК Отсечения

Матрица проецирования

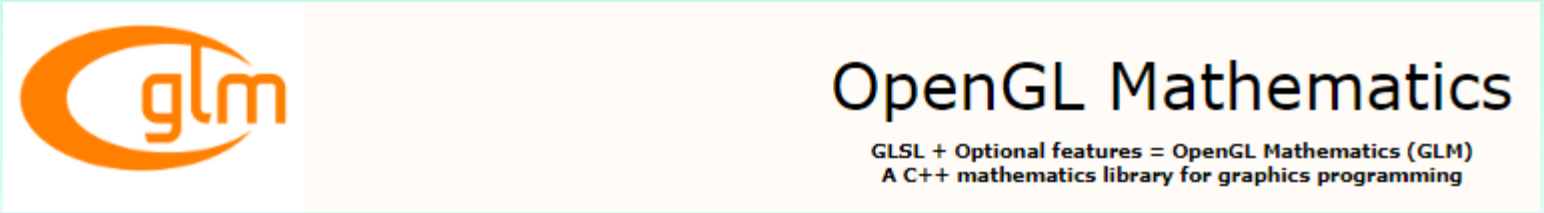
PROJECTION Matrix

WIN Coordinates



Экранное пространство

Open GL. GLM
Математическая библиотека



Основные библиотеки:

glm/glm.hpp // vec2, vec3, mat4, radians
glm/ext.hpp // perspective, translate, rotate

Поддерживаются все функции GLSL
+ матричные операции с квадратными и прямоугольными матрицами (и многое другое)

Open GL. GLM

Матрица Модели (model) ОСК→МСК

Матрица масштабирования (Scale) - S.

```
S = glm::scale(glm::mat4(1.0f), glm::vec3(1.0f, 1.0f, 1.0f));
```

коэффициенты масштабирования

Матрица вращения (Rotate) - R.

угол поворота

```
R =glm::rotate(glm::mat4(1.0f), glm::radians(35.0f),
```

ось поворота

```
glm::vec3(1.0f, 0.0f, 0.0f));
```

Матрица сдвига (Translation) - T .

```
T = glm::translate(glm::mat4(1.0f),
```

вектор смещения

```
glm::vec3(0.0f, 0.0f, -1.0f));
```

Model = R *S * T.

Open GL. GLM

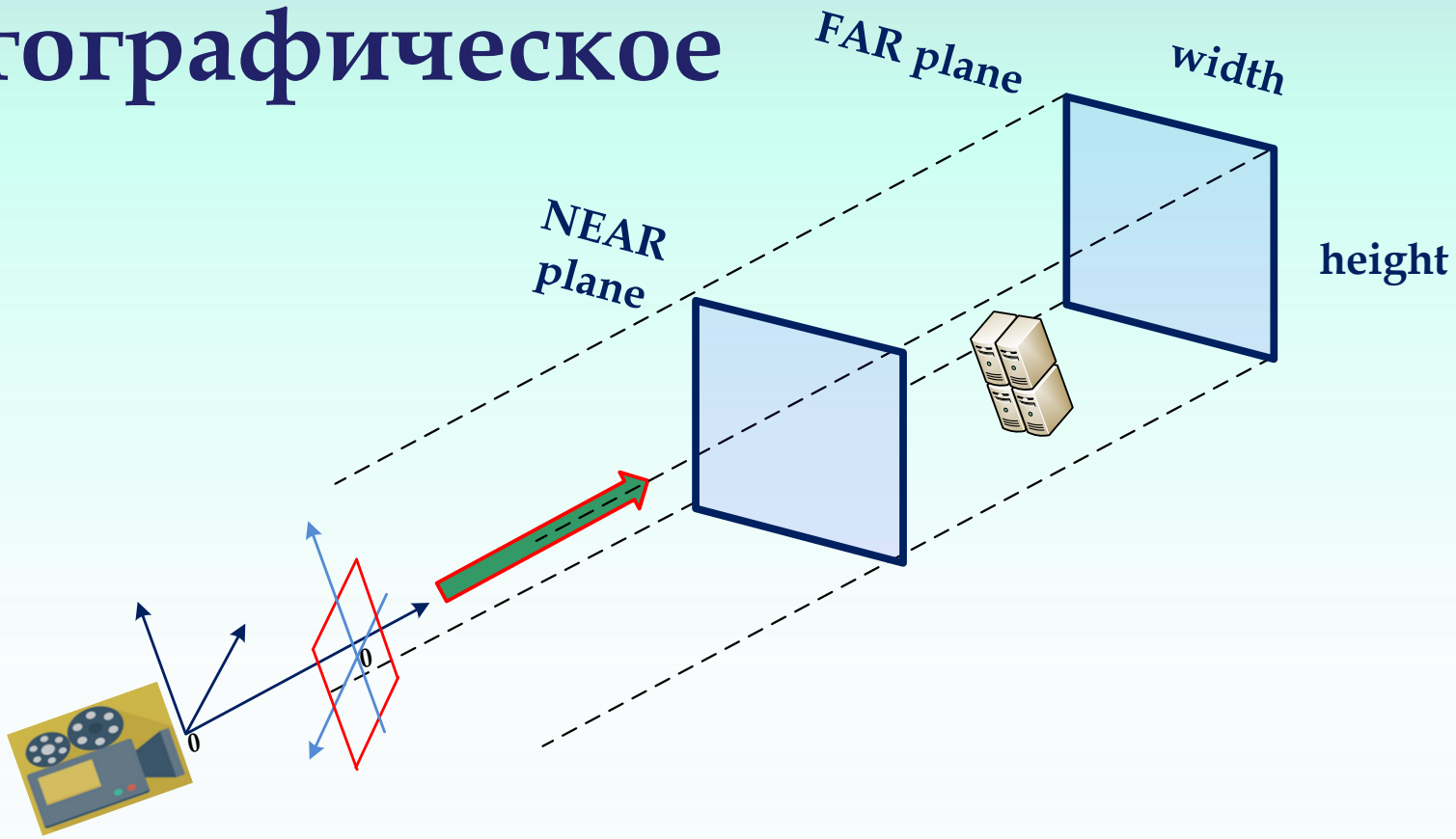
Матрица Вида (view) МСК→СКН

Матрица преобразования в СК камеры (наблюдателя)

```
View = glm::lookAt (glm::vec3(0.0f, 0.0f, 2.0f),
```
вектор положения камеры
```
                    glm::vec3(0.0f, 0.0f, -1.0f),
```
вектор направления визирования
```
                    glm::vec3(1.0f, 0.0f, 0.0f));
```
вектор ориентации камеры

Open GL. ПРОЕЦИРОВАНИЕ

Ортографическое



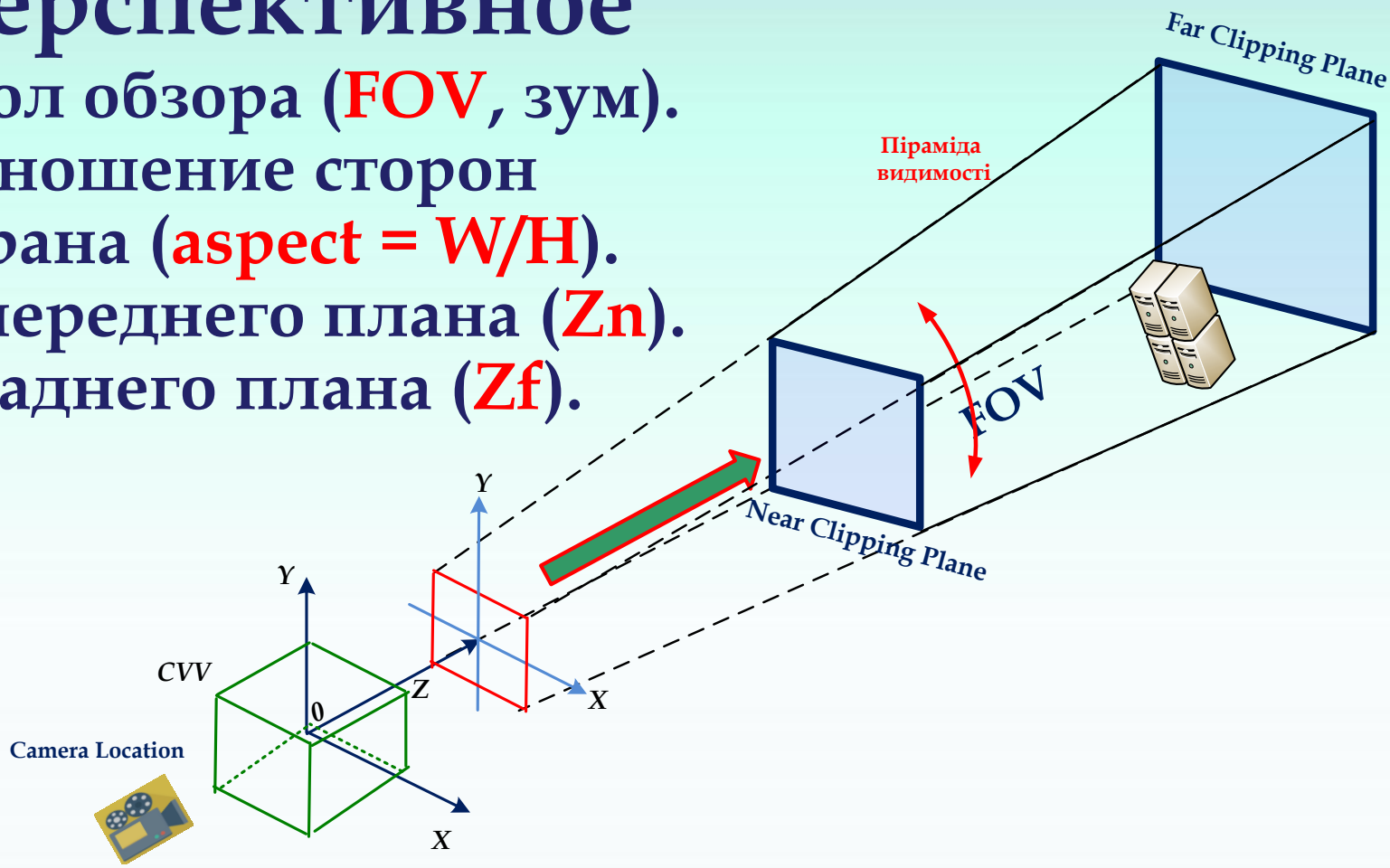
projection = glm::ortho (-1.0f, 1.0f, -1.0f, 1.0f , 2.0f , 10.0f);

левая граница экрана, правая граница экрана
нижняя граница экрана, верхняя граница экрана
Z переднего плана, Z заднего плана

Open GL. ПРОЕЦИРОВАНИЕ

Перспективное

Угол обзора (**FOV**, зум).
Отношение сторон экрана (**aspect = W/H**).
Z переднего плана (**Zn**).
Z заднего плана (**Zf**).

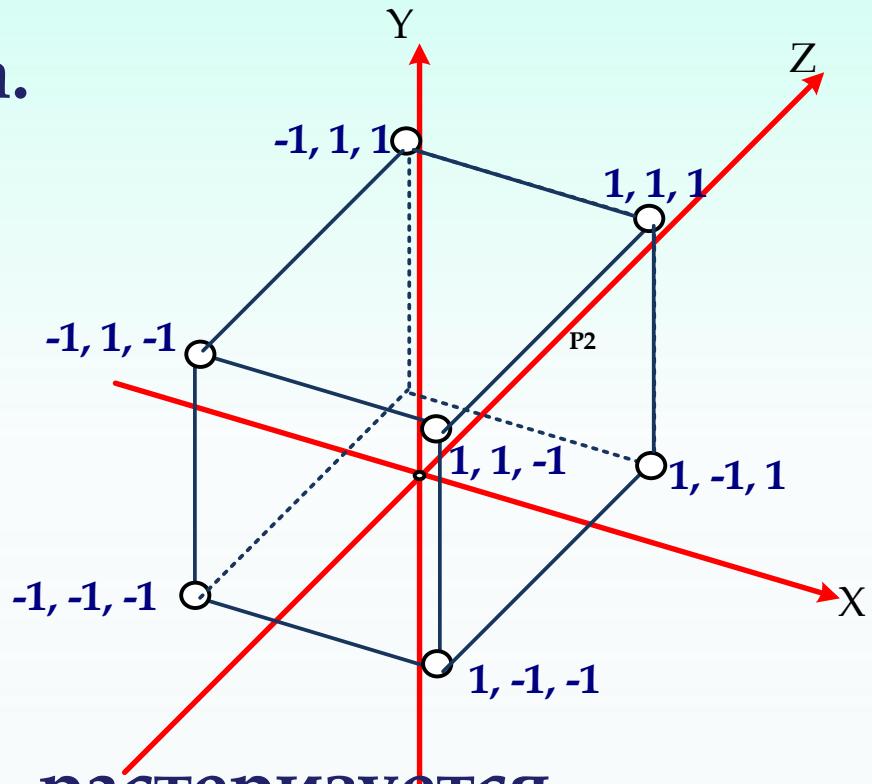


projection = glm::perspective(2.0f, WIDTH/HEIGHT, 2.0f, 10.f);

Open GL. Перспективное проецирование

Преобразует пирамиду видимости в канонический объем отсечения (CVV).

Левосторонняя система. CVV – куб, с центром в начале координат и длиной ребра = 2.



Все что попадает в куб – растеризуется.
Все что не попадает в куб – отсекается.

Open GL. Перспективное проецирование

CVV – псевдоглубина (псевдоZ): $Z \to \hat{Z}$

Псевдоглубина $\hat{Z}$:

- рассчитывается по Z ,

- у всех точек лежащих на передней плоскости $\hat{Z} = -1$,

- у всех точек лежащих на дальней плоскости $\hat{Z} = +1$,

- у всех точках внутри пирамиды видимости $\hat{Z} \in [-1, +1]$.

Open GL. Перспективное проецирование

Расчет псевдоглубины : $Z \to \widehat{Z}$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 \to a & 0 \to b \\ 0 & 0 & {}^{1}/_{Z_E} \to 1 & 0 \end{bmatrix}; \quad P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

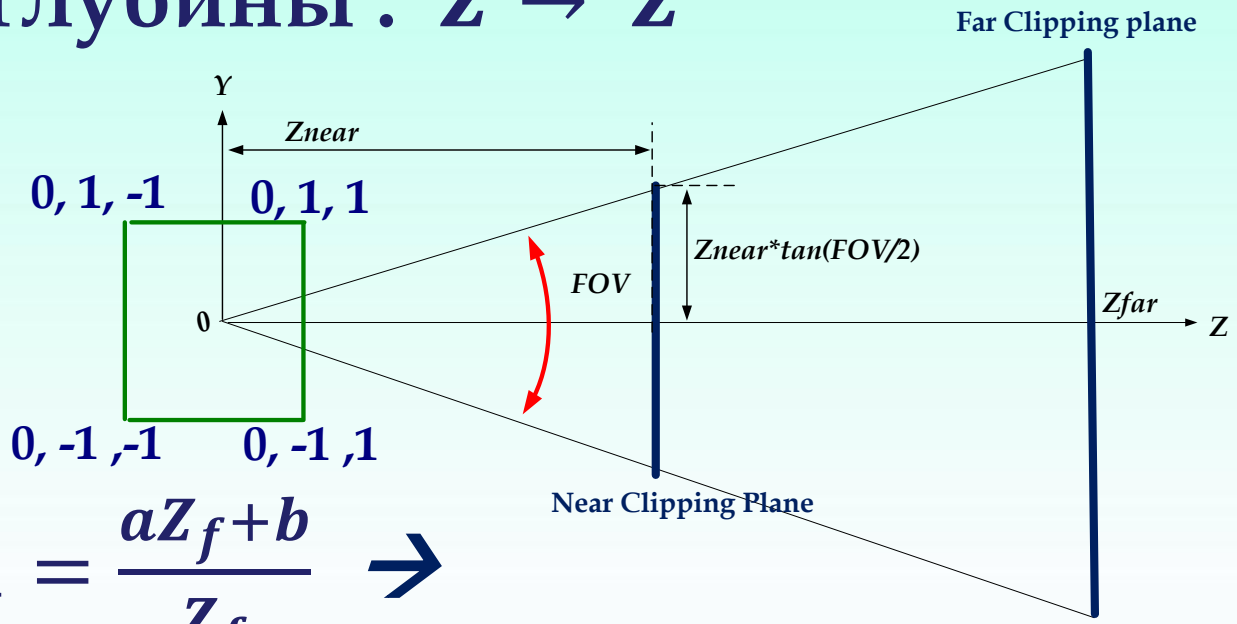
Будем искать

$$\widehat{Z} = \frac{aZ + b}{Z}$$

Open GL. Перспективное проецирование

Расчет псевдоглубины : $Z \to \hat{Z}$

Пусть $\hat{Z} = \frac{aZ+b}{Z}$



$$-1 = \frac{aZ_n+b}{Z_n}; +1 = \frac{aZ_f+b}{Z_f} \rightarrow$$

$$Z_f(aZ_n + b) + Z_n(aZ_f + b) = 0$$

$$a = \frac{Z_f+Z_n}{Z_f-Z_n}; b = \frac{-2Z_fZ_n}{Z_f-Z_n}$$

Open GL. Перспективное проецирование

Расчет псевдоглубины : $Z \rightarrow \hat{Z}$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{Z_f + Z_n}{Z_f - Z_n} & \frac{-2Z_f Z_n}{Z_f - Z_n} \\ 0 & 0 & 1 & 0 \end{bmatrix};$$

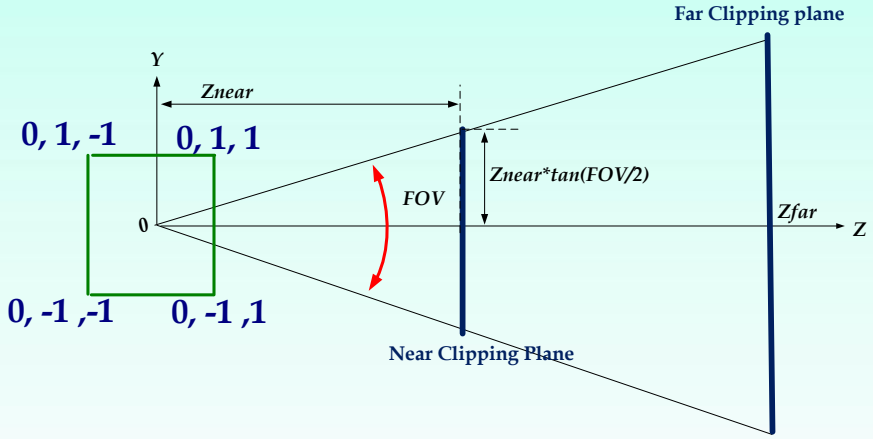
Zf=10,
Zn = 2;
a = 1.5,
b = -5.0

Z=2 → $\hat{Z}$=-1; Z=10 → $\hat{Z}$=1; Z= 5 → $\hat{Z}$=.5

Open GL. Перспективное проецирование

Преобразование X,Y

$$M = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & 1 & 0 \end{bmatrix};$$



$$\widehat{Y} = q * {}^{Y}/_{Z}; 1 = q * \left({}^{\frac{H}{2}}/_{Z_n}\right) = q * tg(fov/2)$$

$$q = ctg\left(\frac{fov}{2}\right)$$

$$s = \frac{q}{aspect} = ctg(\frac{fov}{2})/aspect$$

Open GL. Перспективное проецирование

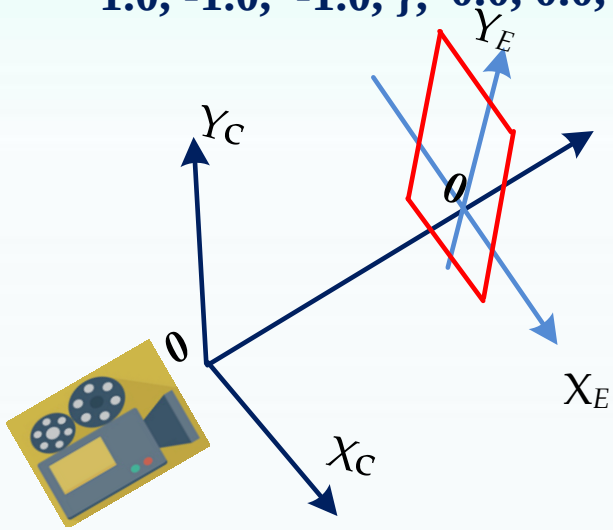
Матрица трансформации в CVV

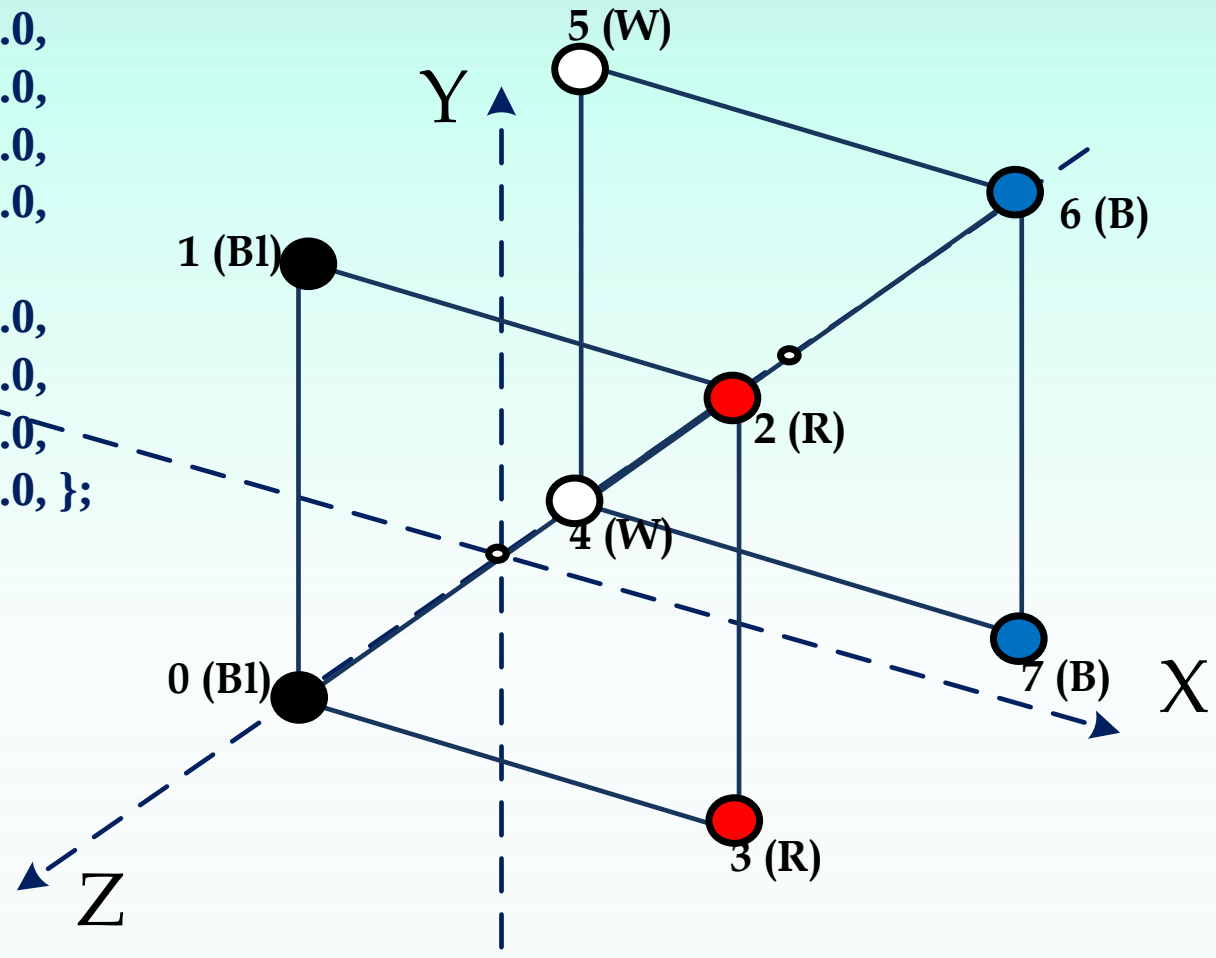
$$\begin{bmatrix} \frac{ctg\left(\frac{fov}{2}\right)}{aspect} & 0 & 0 & 0 \\ 0 & ctg\left(\frac{fov}{2}\right) & 0 & 0 \\ 0 & 0 & \frac{Z_f + Z_n}{Z_f - Z_n} & \frac{-2Z_f Z_n}{Z_f - Z_n} \\ 0 & 0 & 1 & 0 \end{bmatrix};$$

Open GL. Пример

```
vertices[] = {
-1.0, -1.0,  0.0,
-1.0,  1.0,  0.0,
 1.0,  1.0,  0.0,
 1.0, -1.0,  0.0,
//
-1.0, -1.0,  -1.0,
-1.0,  1.0,  -1.0,
 1.0,  1.0,  -1.0,
 1.0, -1.0,  -1.0, };
```

```
colors[] = {
0.0, 0.0, 0.0,
0.0, 0.0, 0.0,
1.0, 0.0, 0.0,
1.0, 0.0, 0.0,
//
1.0, 1.0, 1.0,
1.0, 1.0, 1.0,
0.0, 0.0, 1.0,
0.0, 0.0, 1.0, };
```






```
elements[] = { 0, 1, 2, 2, 3, 0, 4, 5, 6, 6, 7, 4,};
```

Open GL. Пример

Vertix Shaider:

```
#version 330 core
layout (location = 0) in vec3 position;
layout (location = 1) in vec3 color;

uniform mat4  model;
uniform mat4  view;
uniform mat4  projection;

void main() {
        gl_Position = projection*view*model*vec4(position,
                                                 1.0f);
            ourColor = color;
}
```

Open GL. Z буфер

Для проверки видимости пиксела используется тест глубины (z - буфер).

Включение

```
glEnable(GL_DEPTH_TEST);
```

Очистка

```
glClear(GL_COLOR_BUFFER_BIT |
                    GL_DEPTH_BUFFER_BIT);
```

ЛИТЕРАТУРА

Официальный сайт
https://www.opengl.org/wiki/Main_Page
Open GL programming guide
http://glprogramming.com/red/index.html
Русскоязычный сайт
http://www.opengl.org.ru/books/open_gl/index.html

- **OpenGL super bible: comprehensive tutorial and reference / Richard S. Wright Jr. …. – 5th ed., Pearson Education Inc. - 2011**
- **Рост Р.Дж. Open GL. Трехмерная графика и язык программирования шейдеров. Для профессионалов.- Спб.: Питер, 2005.- 428 с.: ил.**
- Ву М., Девис Т., Нейдер Дж., Шрайнер Д. Open GL. Руководство по программированию. Библиотека программиста. 4-е изд. – СПб.: Питер, 2006 .- 624 с.: ил.

ЛИТЕРАТУРА

GLSL:

The OpenGL® Shading Language.

The Khronos Group Inc. Language Version: 3.30, Document Revision: 6. 11-Mar-2010

https://www.khronos.org/opengl/wiki/OpenGL_Shading_Language

ЛИТЕРАТУРА

GLM:

The OpenGL Mathematics. GLSL + Optional features = OpenGL Mathematics (GLM)

https://glm.g-truc.net/0.9.9/index.html

GLM manual

http://glm.g-truc.net/glm.pdf

Вопросы для экзамена

ТЕМА: 3D

1. Какие системы координат используются в OpenGL ?
2. Как формируется и какие преобразования выполняет матрица модели?
3. Как формируется и какие преобразования выполняет матрица вида?
4. Как формируется и какие преобразования выполняет матрица проецирования?

END #14