

Interactive program to adjust frustum,
near, and far clipping planes.

Interactive frustum program

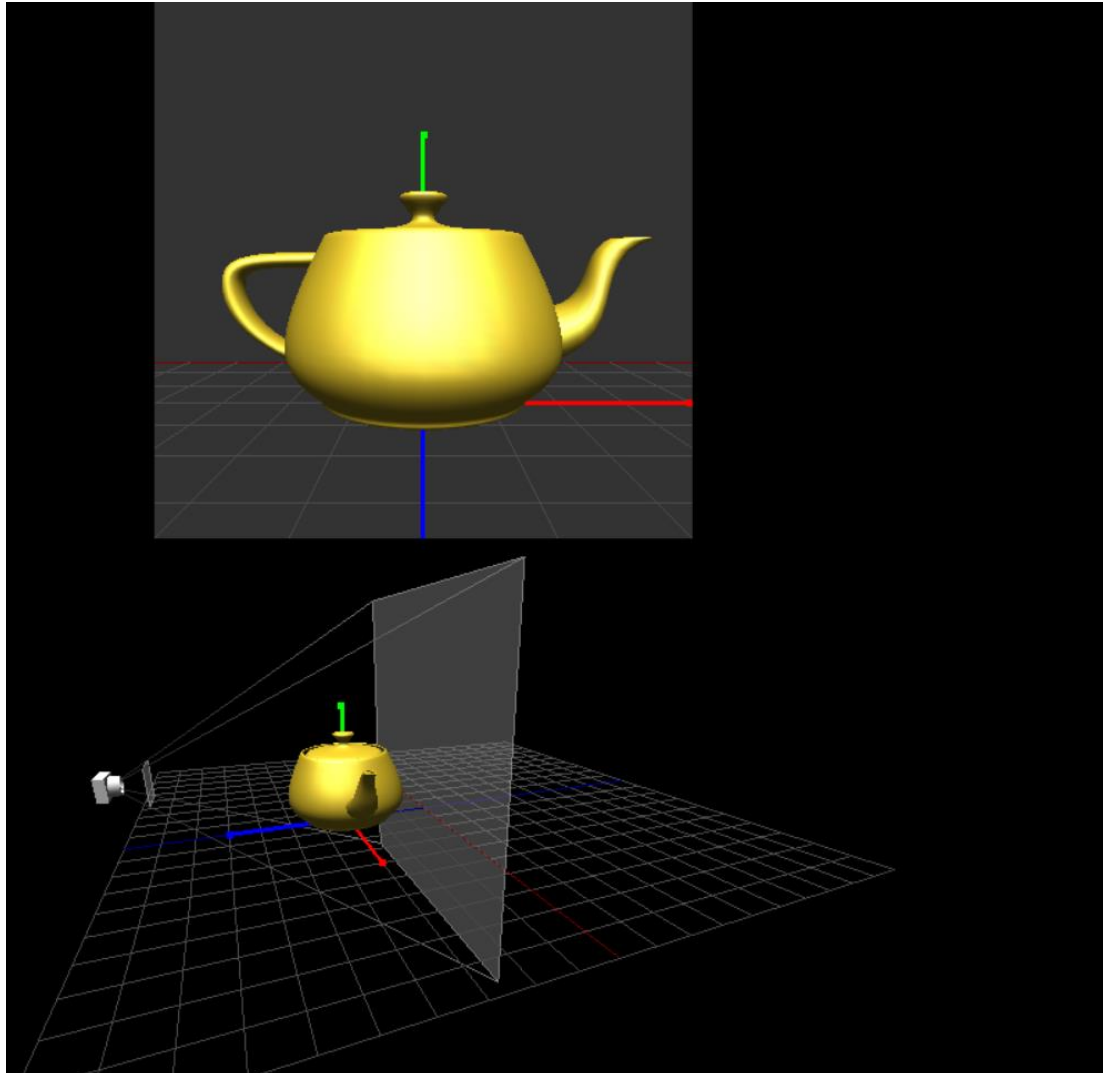
Interactive program to adjust near and far clipping planes:

<http://www.songho.ca/opengl/files/matrixModelView.zip>

Note: This program will only run under Microsoft's Windows operating system. It uses a combination of OpenGL as well as Windows API. However, you can still read the code.

The “take away” from reading the code: Linear transformations are used to adjust camera view when near or far clipping planes are adjusted. This is how we compute the view from the camera!

Interactive frustum program: screenshot



View (Camera)

X OpenGL Functions

Position Y `glRotatef(-0, 0, 0, 1);`

Z `glRotatef(-0, 0, 1, 0);`

Pitch (X) `glRotatef(-0, 1, 0, 0);`

Heading (Y) `glTranslatef(-0, -2, -10);`

Roll (Z)

Model

X OpenGL Functions

Position Y `glTranslatef(0, 0, 3);`

Z `glRotatef(0, 1, 0, 0);`

Rotation X `glRotatef(0, 0, 1, 0);`

Y `glRotatef(0, 0, 0, 1);`

Z

ModelView Matrix = View Matrix x Model Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-7.00
0.00	0.00	0.00	1.00

1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-10.00
0.00	0.00	0.00	1.00

1.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00
0.00	0.00	1.00	3.00
0.00	0.00	0.00	1.00

Interactive frustum program: screenshot

The screenshot displays an interactive 3D environment. On the left, a yellow teapot is rendered in a perspective view. A blue arrow points from a yellow box labeled "Camera perspective" to the teapot. Below the teapot, a camera frustum is visible, showing the viewing frustum and the teapot's position within it. On the right, a control panel is shown with the following sections:

View (Camera)

X: 0, Y: 2, Z: 10, Heading (Y): 0, Roll (Z): 0. Includes a "Reset" button and an "OpenGL Functions" box containing:

```
glRotatef(-0, 0, 0, 1);
glRotatef(-0, 0, 1, 0);
glRotatef(-0, 1, 0, 0);
glTranslatef(-0, -2, -10);
```

Model

X: 0, Y: 0, Z: 3, Rotation X: 0, Y: 0, Z: 0. Includes a "Reset" button and an "OpenGL Functions" box containing:

```
glTranslatef(0, 0, 3);
glRotatef(0, 1, 0, 0);
glRotatef(0, 0, 1, 0);
glRotatef(0, 0, 0, 1);
```

ModelView Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-7.00
0.00	0.00	0.00	1.00

View Matrix

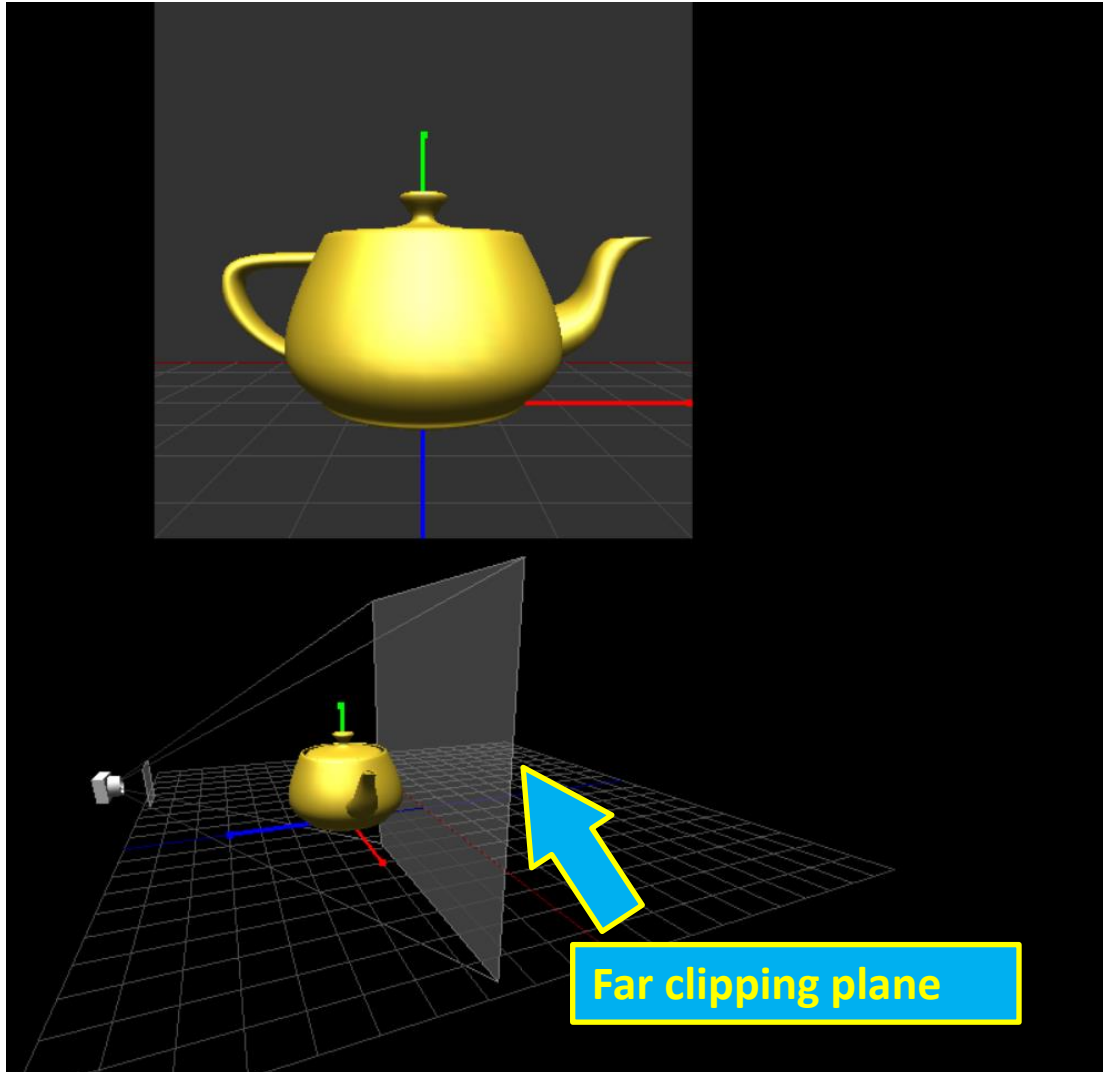
1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-10.00
0.00	0.00	0.00	1.00

Model Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00
0.00	0.00	1.00	3.00
0.00	0.00	0.00	1.00

The matrices are combined as follows: ModelView Matrix = View Matrix × Model Matrix.

Interactive frustum program: screenshot



View (Camera)

X OpenGL Functions
Position Y
Z
Pitch (X)
Heading (Y)
Roll (Z)
OpenGL Functions

```
glRotatef(-0, 0, 0, 1);  
glRotatef(-0, 0, 1, 0);  
glRotatef(-0, 1, 0, 0);  
glTranslatef(-0, -2, -10);
```

Model

X OpenGL Functions
Position Y
Z
Rotation X
Y
Z
OpenGL Functions

```
glTranslatef(0, 0, 3);  
glRotatef(0, 1, 0, 0);  
glRotatef(0, 0, 1, 0);  
glRotatef(0, 0, 0, 1);
```

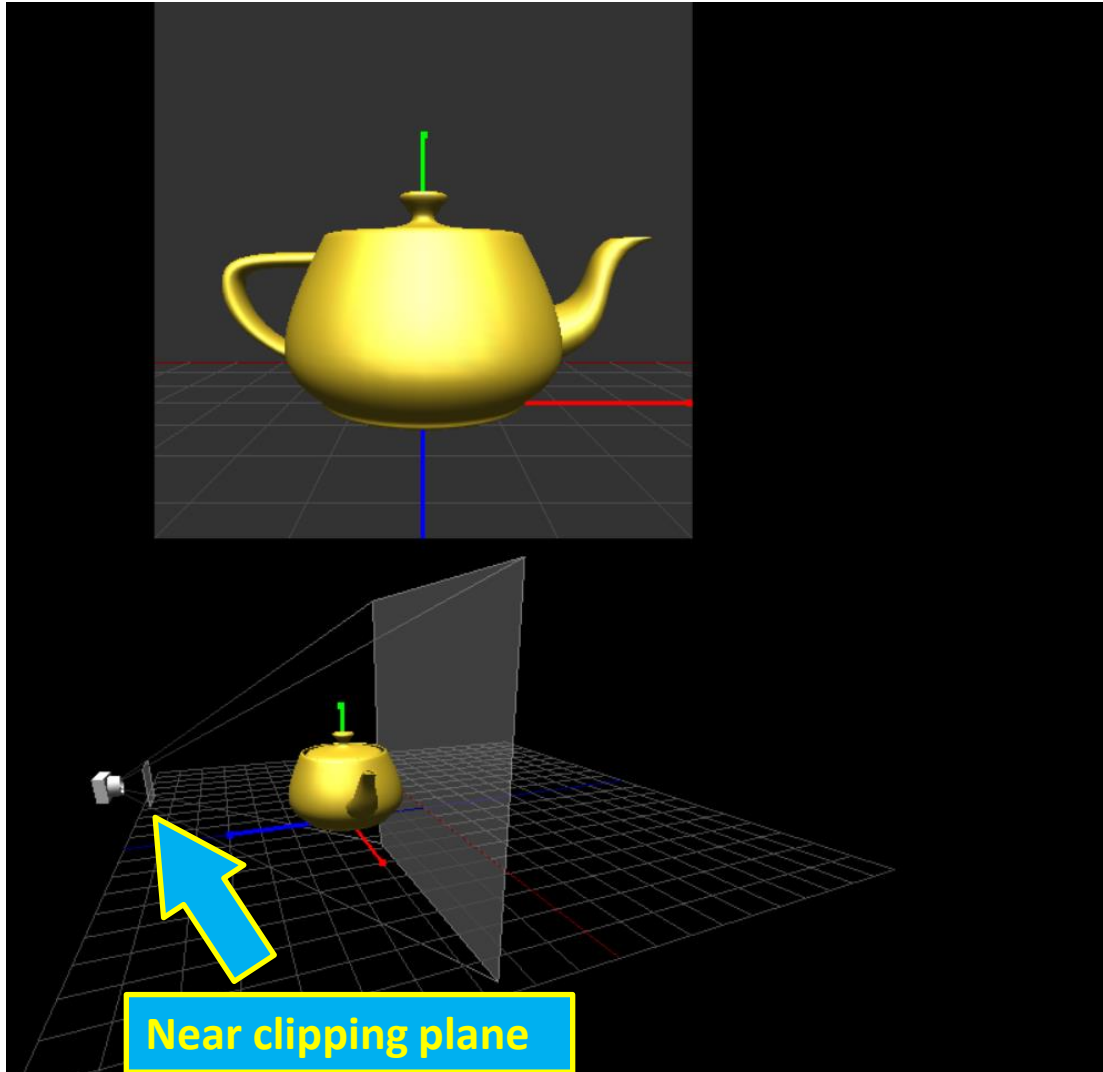
ModelView Matrix = View Matrix x Model Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-7.00
0.00	0.00	0.00	1.00

1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-10.00
0.00	0.00	0.00	1.00

1.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00
0.00	0.00	1.00	3.00
0.00	0.00	0.00	1.00

Interactive frustum program: screenshot



View (Camera)

X OpenGL Functions

Position Y `glRotatef(-0, 0, 0, 1);`

Z `glRotatef(-0, 0, 1, 0);`

Pitch (X) `glRotatef(-0, 1, 0, 0);`

Heading (Y) `glTranslatef(-0, -2, -10);`

Roll (Z)

Model

X OpenGL Functions

Position Y `glTranslatef(0, 0, 3);`

Z `glRotatef(0, 1, 0, 0);`

X `glRotatef(0, 0, 1, 0);`

Rotation Y `glRotatef(0, 0, 0, 1);`

Z

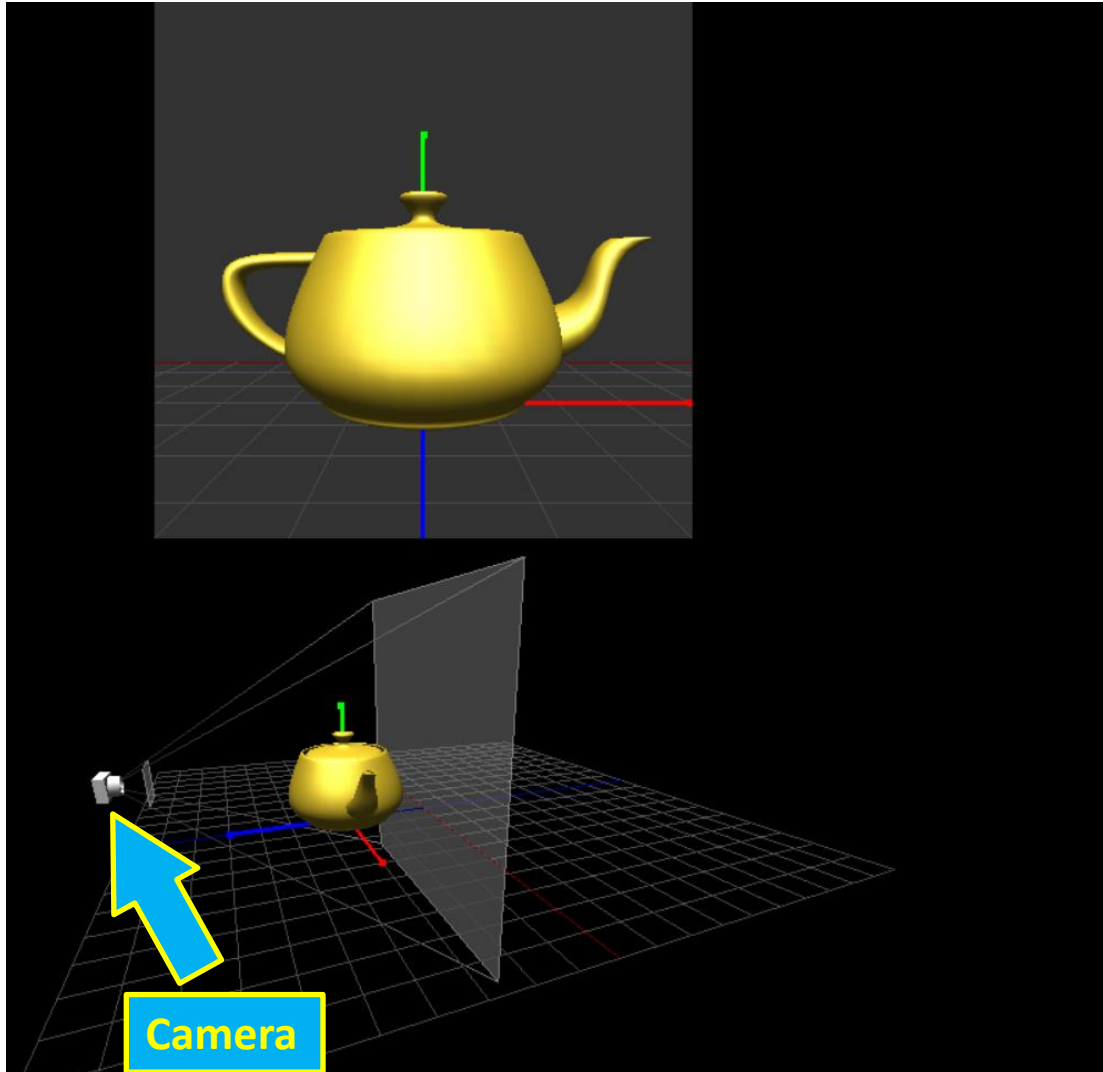
ModelView Matrix = View Matrix x Model Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-7.00
0.00	0.00	0.00	1.00

1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-10.00
0.00	0.00	0.00	1.00

1.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00
0.00	0.00	1.00	3.00
0.00	0.00	0.00	1.00

Interactive frustum program: screenshot



View (Camera)

X OpenGL Functions

Position Y `glRotatef(-0, 0, 0, 1);`

Z `glRotatef(-0, 0, 1, 0);`

Pitch (X) `glRotatef(-0, 1, 0, 0);`

Heading (Y) `glTranslatef(-0, -2, -10);`

Roll (Z)

Model

X OpenGL Functions

Position Y `glTranslatef(0, 0, 3);`

Z `glRotatef(0, 1, 0, 0);`

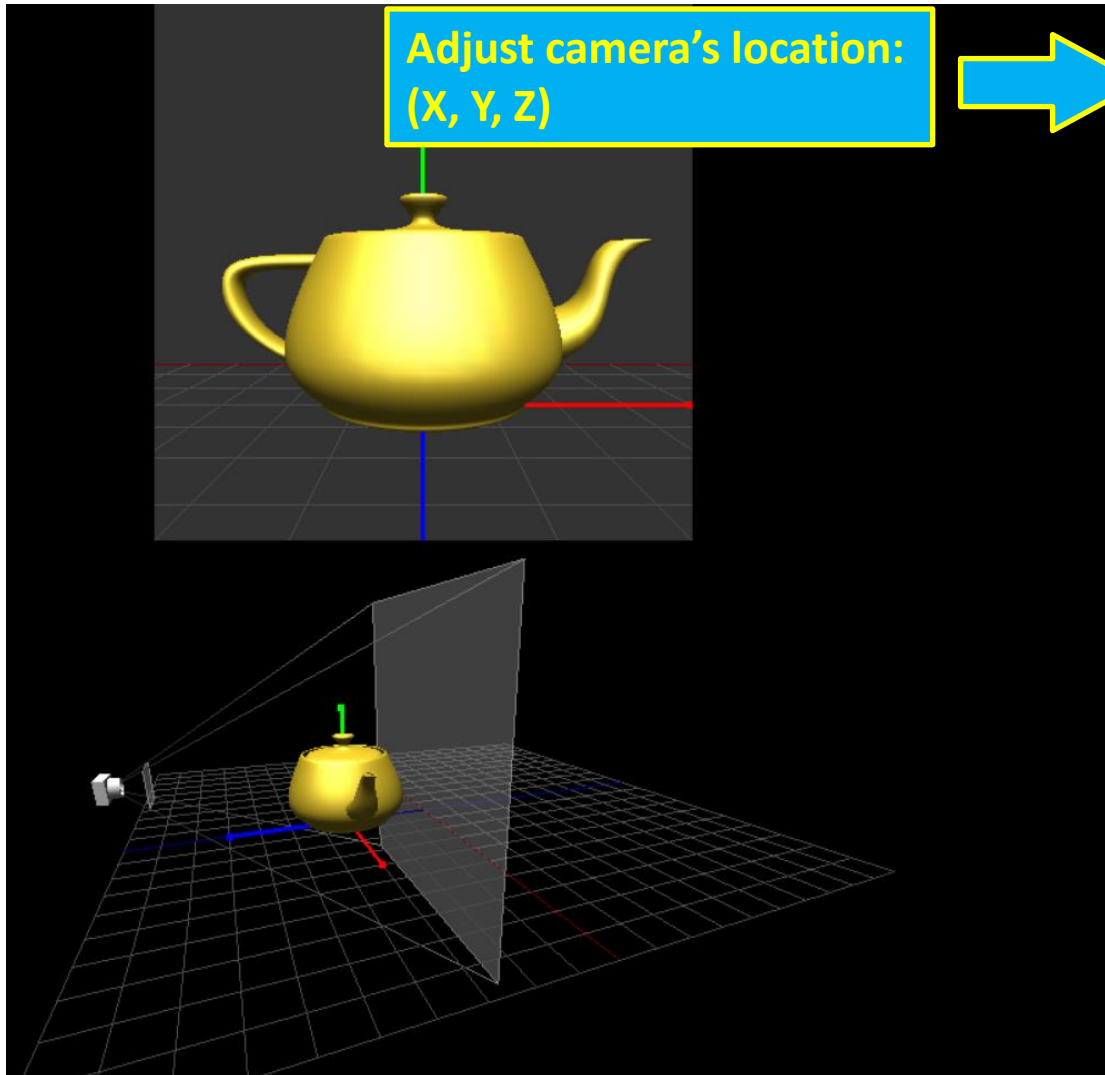
Rotation X `glRotatef(0, 0, 1, 0);`

Y `glRotatef(0, 0, 0, 1);`

Z

ModelView Matrix	View Matrix	Model Matrix
1.00 0.00 0.00 0.00	1.00 0.00 0.00 0.00	1.00 0.00 0.00 0.00
0.00 1.00 0.00 -2.00	0.00 1.00 0.00 -2.00	0.00 1.00 0.00 0.00
0.00 0.00 1.00 -7.00	0.00 0.00 1.00 -10.00	0.00 0.00 1.00 3.00
0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00	0.00 0.00 0.00 1.00

Interactive frustum program: screenshot



View (Camera)

X

Position Y

Z

Pitch (X)

Heading (Y)

Roll (Z)

OpenGL Functions

```
glRotatef(-0, 0, 0, 1);
glRotatef(-0, 0, 1, 0);
glRotatef(-0, 1, 0, 0);
glTranslatef(-0, -2, -10);
```

Model

X

Position Y

Z

Rotation X

Y

Z

OpenGL Functions

```
glTranslatef(0, 0, 3);
glRotatef(0, 1, 0, 0);
glRotatef(0, 0, 1, 0);
glRotatef(0, 0, 0, 1);
```

ModelView Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-7.00
0.00	0.00	0.00	1.00

=

View Matrix

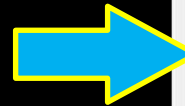
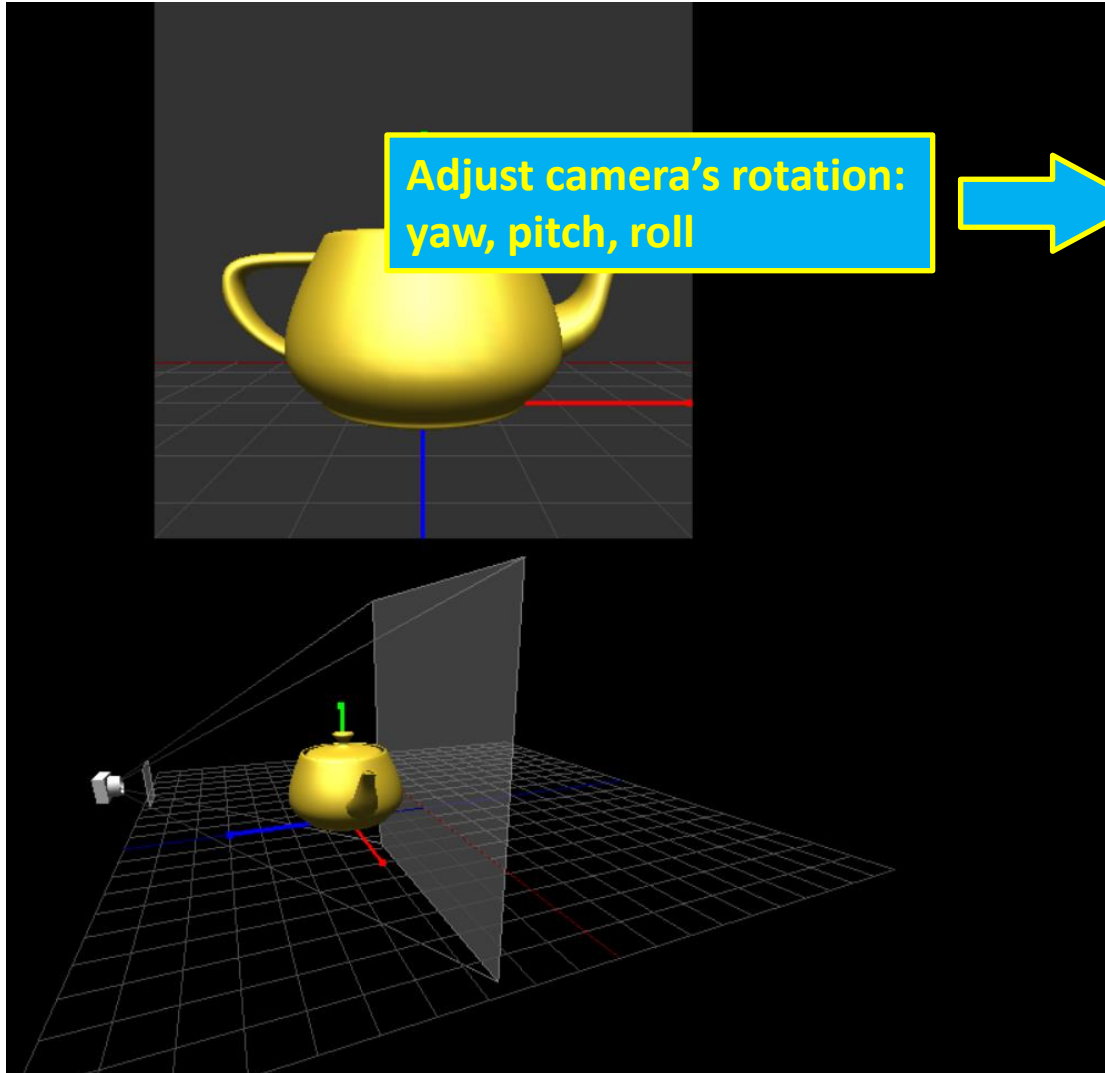
1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-10.00
0.00	0.00	0.00	1.00

x

Model Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00
0.00	0.00	1.00	3.00
0.00	0.00	0.00	1.00

Interactive frustum program: screenshot



View (Camera)

X

Position Y

Z

Pitch (X)

Heading (Y)

Roll (Z)

OpenGL Functions

```
glRotatef(-0, 0, 0, 1);  
glRotatef(-0, 0, 1, 0);  
glRotatef(-0, 1, 0, 0);  
glTranslatef(-0, -2, -10);
```

Model

X

Position Y

Z

Rotation X

Y

Z

OpenGL Functions

```
glTranslatef(0, 0, 3);  
glRotatef(0, 1, 0, 0);  
glRotatef(0, 0, 1, 0);  
glRotatef(0, 0, 0, 1);
```

ModelView Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-7.00
0.00	0.00	0.00	1.00

=

View Matrix

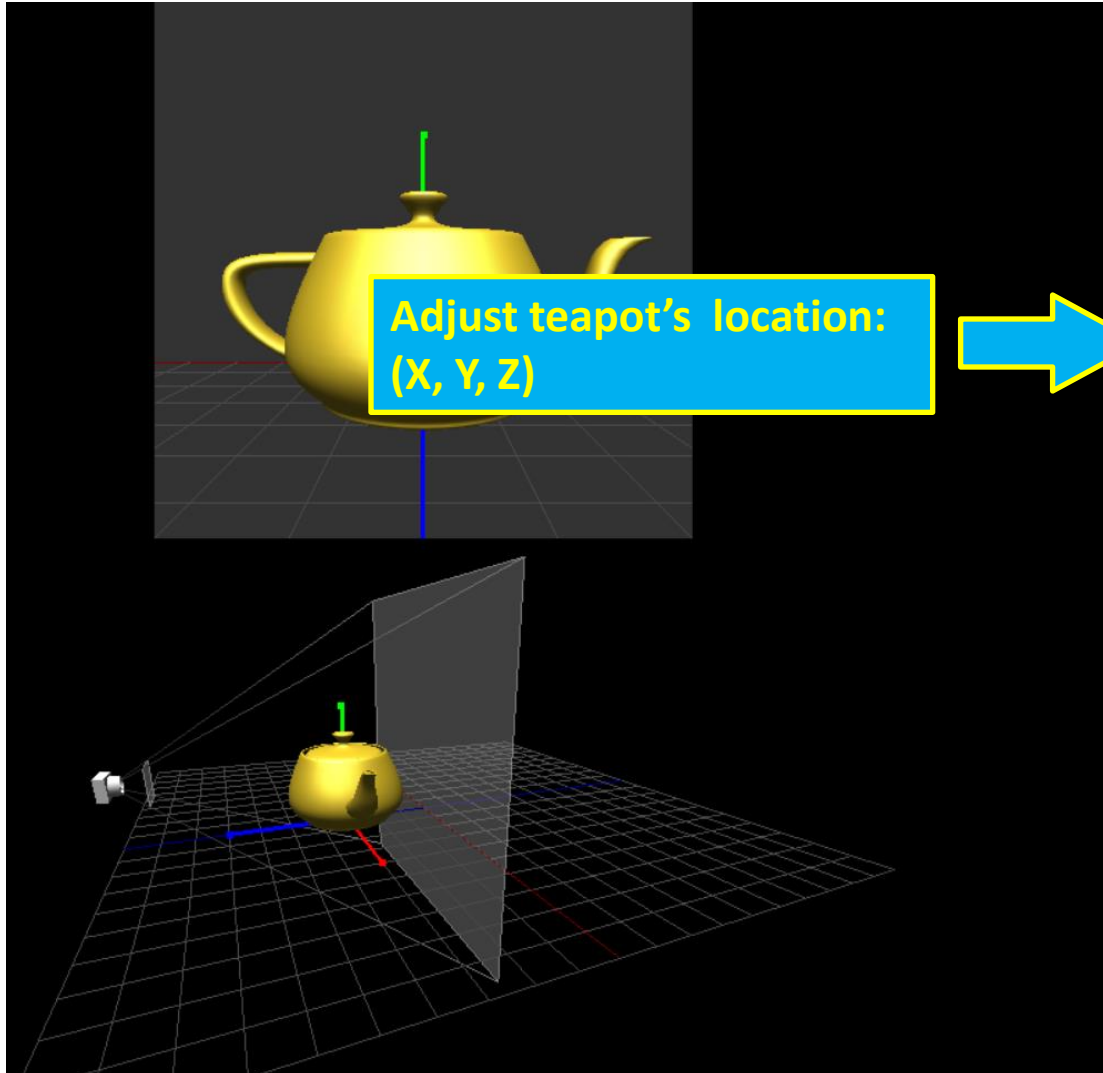
1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-10.00
0.00	0.00	0.00	1.00

x

Model Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00
0.00	0.00	1.00	3.00
0.00	0.00	0.00	1.00

Interactive frustum program: screenshot



View (Camera)

X OpenGL Functions

Position Y
 Z
 Pitch (X)
 Heading (Y)
 Roll (Z)

```
glRotatef(-0, 0, 0, 1);  
glRotatef(-0, 0, 1, 0);  
glRotatef(-0, 1, 0, 0);  
glTranslatef(-0, -2, -10);
```

Model

X OpenGL Functions

Position Y
 Z
 Rotation X
 Y
 Z

```
glTranslatef(0, 0, 3);  
glRotatef(0, 1, 0, 0);  
glRotatef(0, 0, 1, 0);  
glRotatef(0, 0, 0, 1);
```

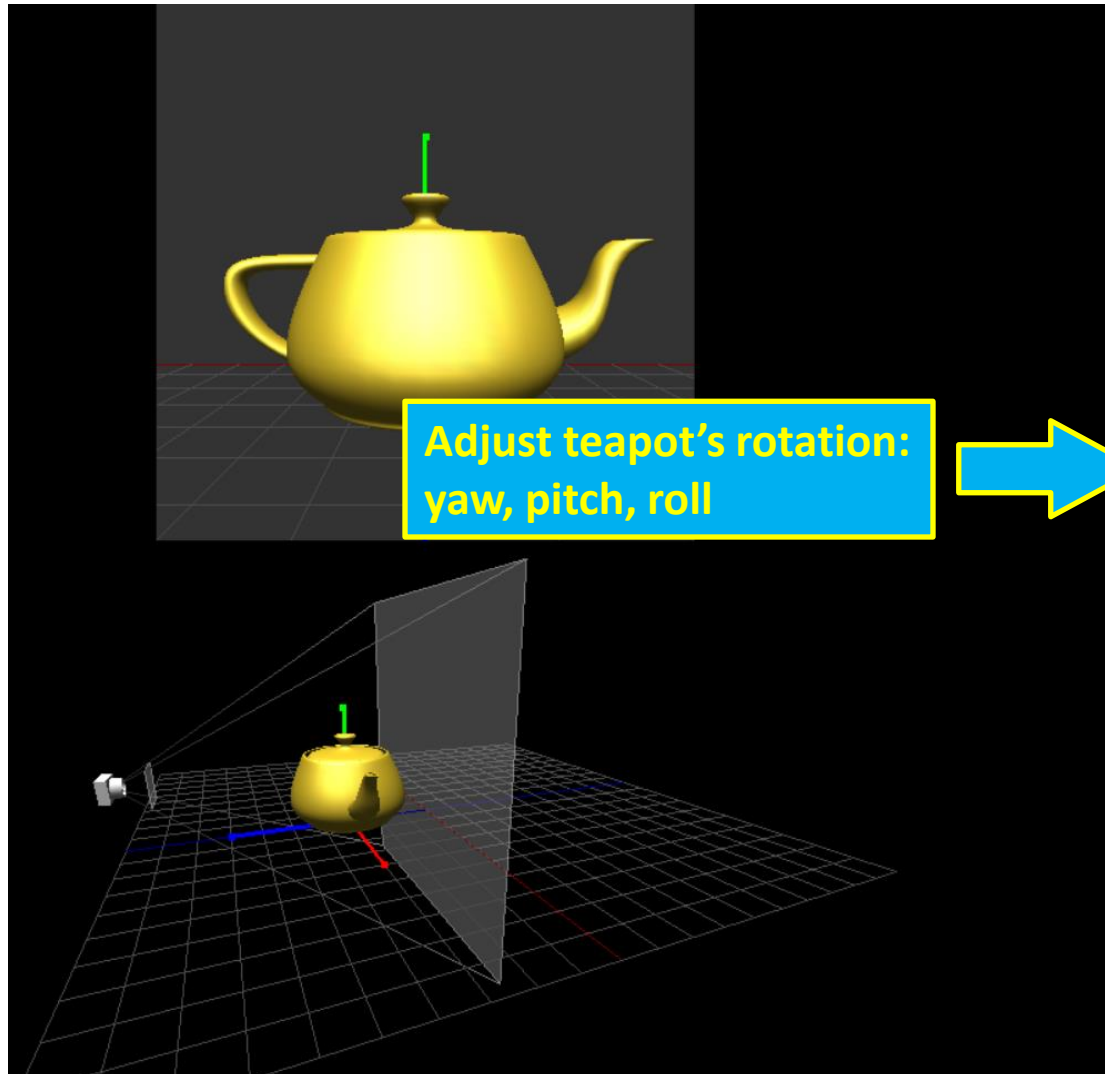
ModelView Matrix = View Matrix x Model Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-7.00
0.00	0.00	0.00	1.00

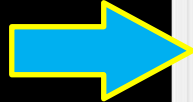
1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-10.00
0.00	0.00	0.00	1.00

1.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00
0.00	0.00	1.00	3.00
0.00	0.00	0.00	1.00

Interactive frustum program: screenshot



Adjust teapot's rotation:
yaw, pitch, roll



View (Camera)

X

Position Y

Z

Pitch (X)

Heading (Y)

Roll (Z)

OpenGL Functions

```
glRotatef(-0, 0, 0, 1);  
glRotatef(-0, 0, 1, 0);  
glRotatef(-0, 1, 0, 0);  
glTranslatef(-0, -2, -10);
```

Model

X

Position Y

Z

Rotation X

Y

Z

OpenGL Functions

```
glTranslatef(0, 0, 3);  
glRotatef(0, 1, 0, 0);  
glRotatef(0, 0, 1, 0);  
glRotatef(0, 0, 0, 1);
```

ModelView Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-7.00
0.00	0.00	0.00	1.00

=

View Matrix

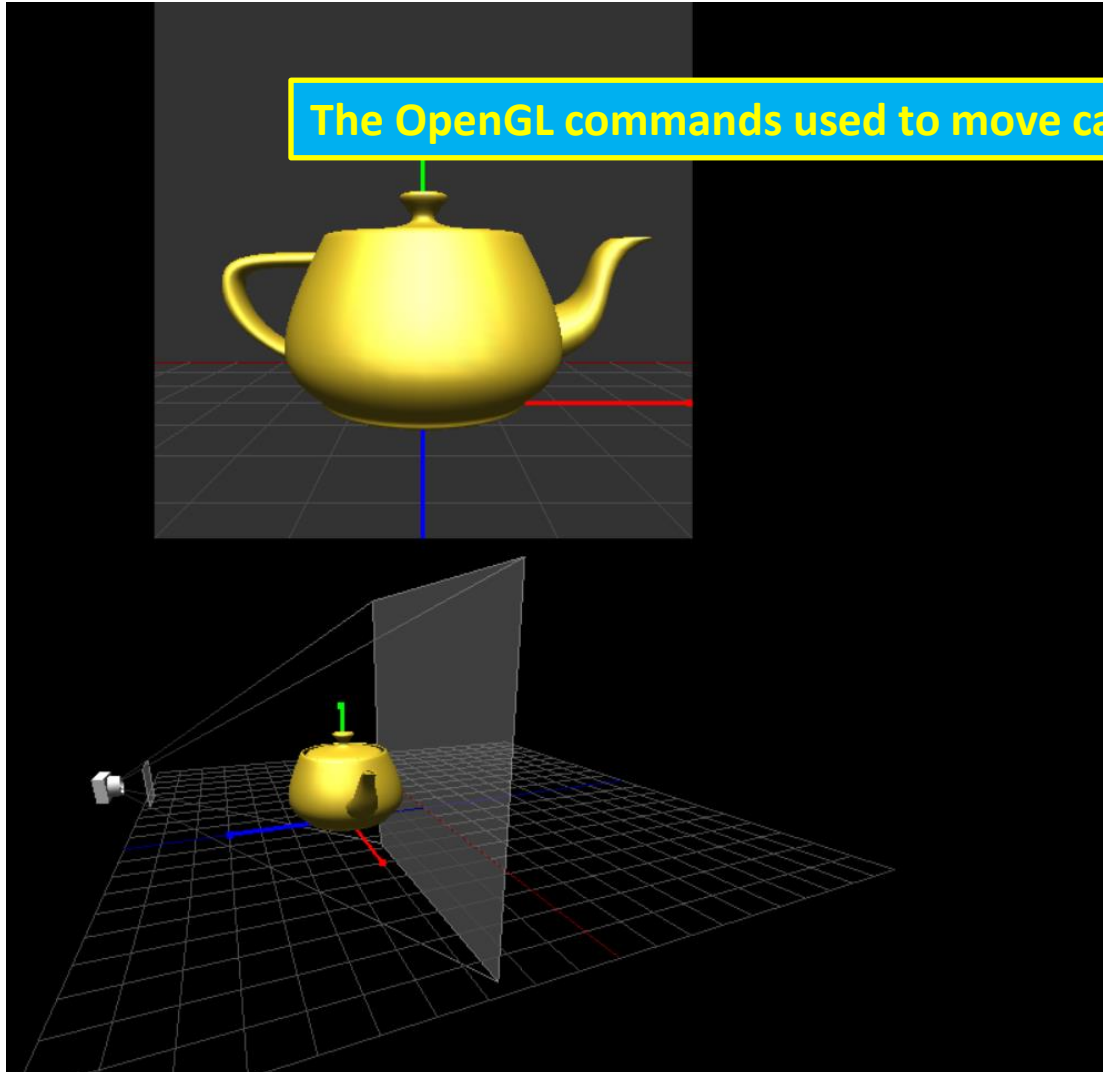
1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-10.00
0.00	0.00	0.00	1.00

x

Model Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00
0.00	0.00	1.00	3.00
0.00	0.00	0.00	1.00

Interactive frustum program: screenshot



View (Camera)

X

Pitch (X)

Heading (Y)

Roll (Z)

OpenGL Functions

```
glRotatef(-0, 0, 0, 1);
glRotatef(-0, 0, 1, 0);
glRotatef(-0, 1, 0, 0);
glTranslatef(-0, -2, -10);
```

Model

X

Position Y

Z

X

Rotation Y

Z

OpenGL Functions

```
glTranslatef(0, 0, 3);
glRotatef(0, 1, 0, 0);
glRotatef(0, 0, 1, 0);
glRotatef(0, 0, 0, 1);
```

ModelView Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-7.00
0.00	0.00	0.00	1.00

=

View Matrix

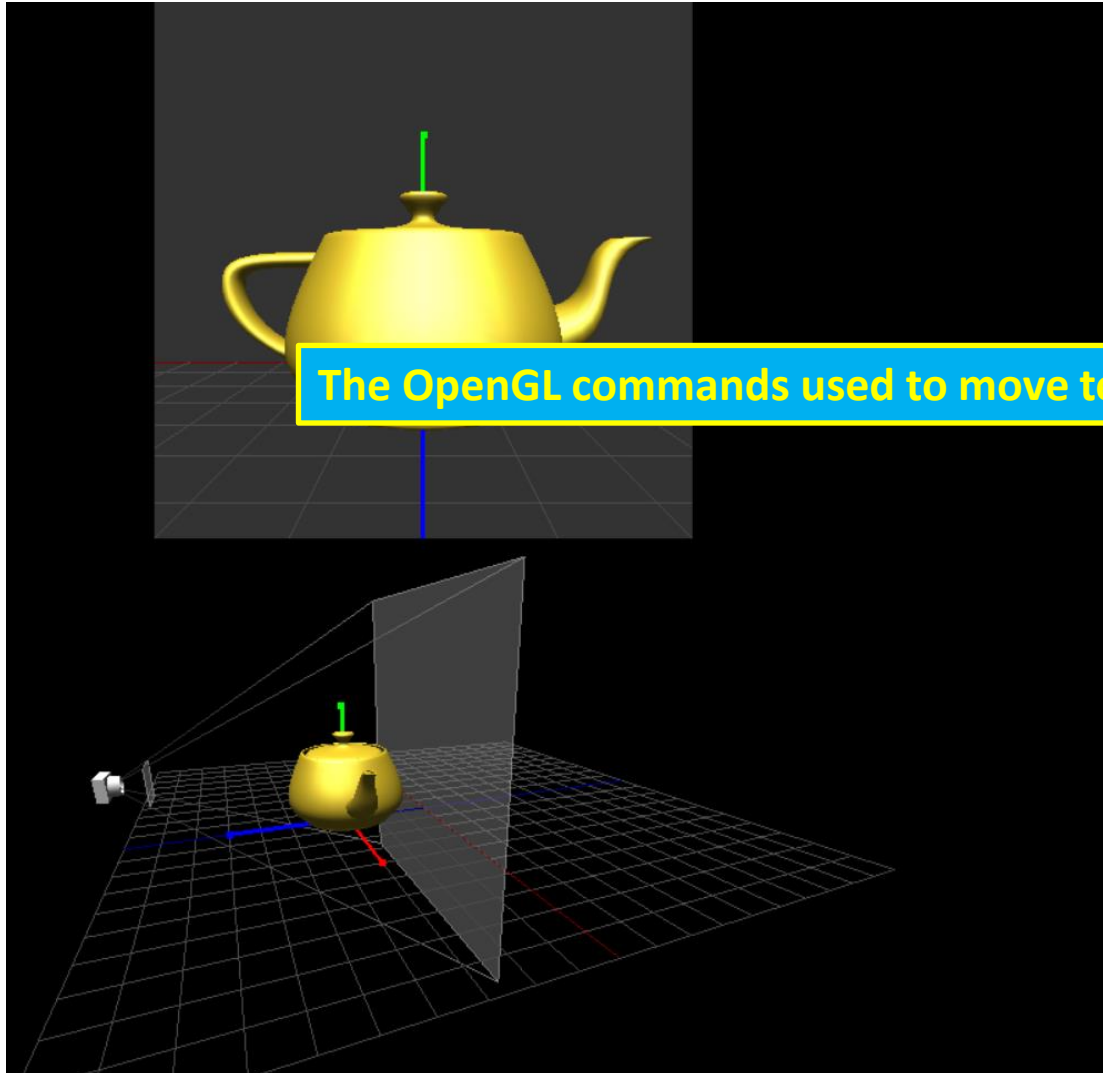
1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-10.00
0.00	0.00	0.00	1.00

x

Model Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00
0.00	0.00	1.00	3.00
0.00	0.00	0.00	1.00

Interactive frustum program: screenshot



The OpenGL commands used to move teapot.

View (Camera)

X OpenGL Functions

Position Y
 Z

Pitch (X)
 Heading (Y)
 Roll (Z)

```
glRotatef(-0, 0, 0, 1);  
glRotatef(-0, 0, 1, 0);  
glRotatef(-0, 1, 0, 0);  
glTranslatef(-0, -2, -10);
```

Model

X OpenGL Functions

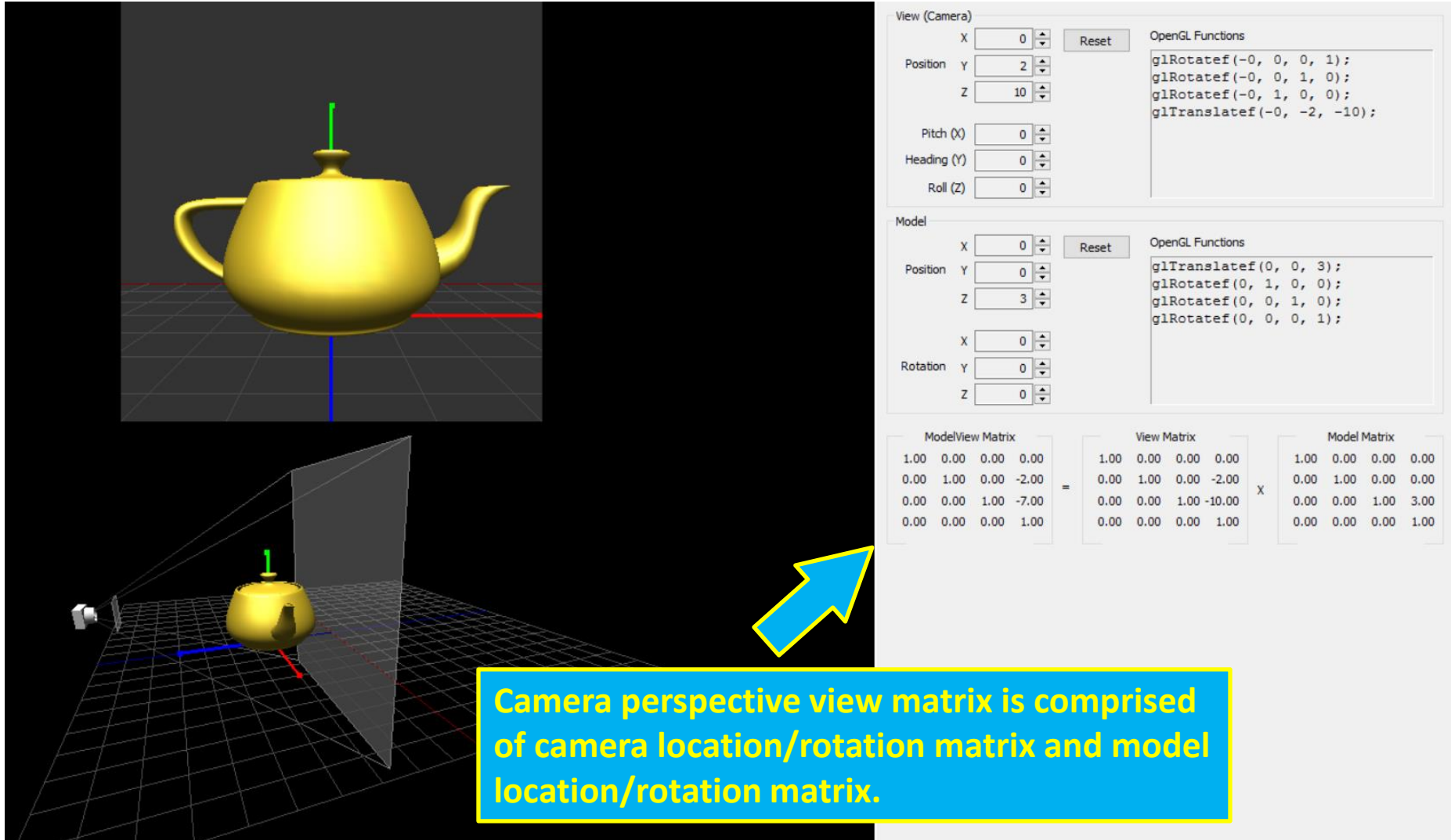
Position Y
 Z

Rotation Y
 Z

```
glTranslatef(0, 0, 3);  
glRotatef(0, 1, 0, 0);  
glRotatef(0, 0, 1, 0);  
glRotatef(0, 0, 0, 1);
```

ModelView Matrix	View Matrix	Model Matrix		
1.00 0.00 0.00 0.00 0.00 1.00 0.00 -2.00 0.00 0.00 1.00 -7.00 0.00 0.00 0.00 1.00	=	1.00 0.00 0.00 0.00 0.00 1.00 0.00 -2.00 0.00 0.00 1.00 -10.00 0.00 0.00 0.00 1.00	x	1.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 1.00 3.00 0.00 0.00 0.00 1.00

Interactive frustum program: screenshot



The screenshot displays an interactive 3D environment. On the left, a yellow teapot is shown in a perspective view. Below it, a camera frustum is visible, illustrating the viewing frustum. On the right, a control panel is shown with the following sections:

View (Camera)

Position: X: 0, Y: 2, Z: 10
Pitch (X): 0
Heading (Y): 0
Roll (Z): 0

OpenGL Functions

```
glRotatef(-0, 0, 0, 1);  
glRotatef(-0, 0, 1, 0);  
glRotatef(-0, 1, 0, 0);  
glTranslatef(-0, -2, -10);
```

Model

Position: X: 0, Y: 0, Z: 3
Rotation: X: 0, Y: 0, Z: 0

OpenGL Functions

```
glTranslatef(0, 0, 3);  
glRotatef(0, 1, 0, 0);  
glRotatef(0, 0, 1, 0);  
glRotatef(0, 0, 0, 1);
```

ModelView Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-7.00
0.00	0.00	0.00	1.00

View Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	-2.00
0.00	0.00	1.00	-10.00
0.00	0.00	0.00	1.00

Model Matrix

1.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00
0.00	0.00	1.00	3.00
0.00	0.00	0.00	1.00

The matrices are combined as follows: ModelView Matrix = View Matrix x Model Matrix.

Camera perspective view matrix is comprised of camera location/rotation matrix and model location/rotation matrix.