

# Viewing

Shaoting Zhang

May 11, 2008

1 Conclusions for HW1

2 Viewing

3 Q/A

## Conclusions for HW1, and some good demos

$$x = \sin(a\theta + \delta), y = \sin(b\theta), \theta \in [t, t + \pi]$$

- The  $\theta$  should change from  $t$  to  $t + \pi$ .
- For the "moving", the phase offset  $\delta$  is a good parameter to adjust.
- For the "fancy", the intensity of the color should still fade to black towards the trailing end of the plot.

1 Conclusions for HW1

2 Viewing

3 Q/A

# Viewing functions

## gluLookAt

- It takes three sets of arguments, which specify the location of the viewpoint, define a reference point toward which the camera is aimed, and indicate which direction is up.
- `gluLookAt(GLdouble eyex, GLdouble eyez, GLdouble eyez, GLdouble centerx, GLdouble centery, GLdouble centerz, GLdouble upx, GLdouble upy, GLdouble upz)`.
- **Cannot be used in our project 1 :-)**

# Demo

Solar system in Java, in the method "reshape".  
Change the parameters of glLookAt to see the effect.

## Notes:

The demo codes are changed from Kiet AT's "The Redbook Examples Using JOGL.

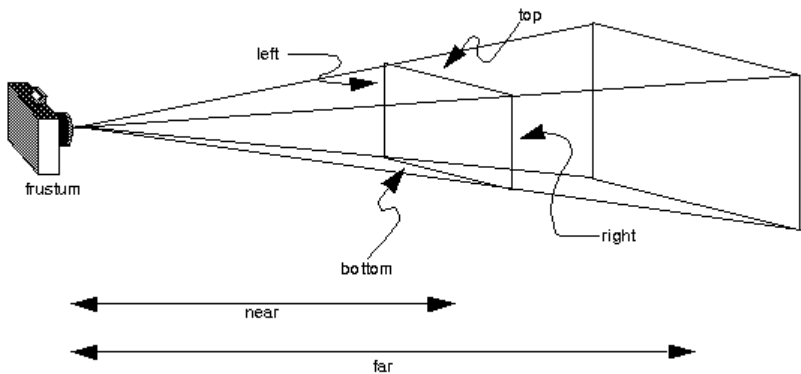
For compiling you also need glskeleton.java in the same directory.

# Viewing functions

## glFrustum

- calculates a matrix that accomplishes perspective projection and multiplies the current projection matrix (typically the identity matrix) by it.
- `void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far).`
- near and far are positive values.

# glFrustum



# Demo

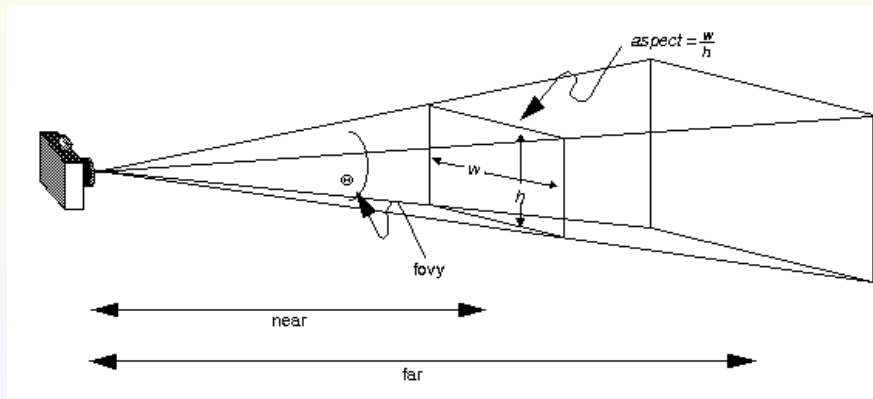
Cube.java, in the method "reshape".  
Change the parameters of glFrustum to see the effect.

# gluPerspective

## gluPerspective

- Rather than specifying corners of the near clipping plane, you specify the angle of the field of view in the x-z plane and the aspect ratio of the width to height ( $x/y$ ).
- `void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar).`

# gluPerspective



# Demo

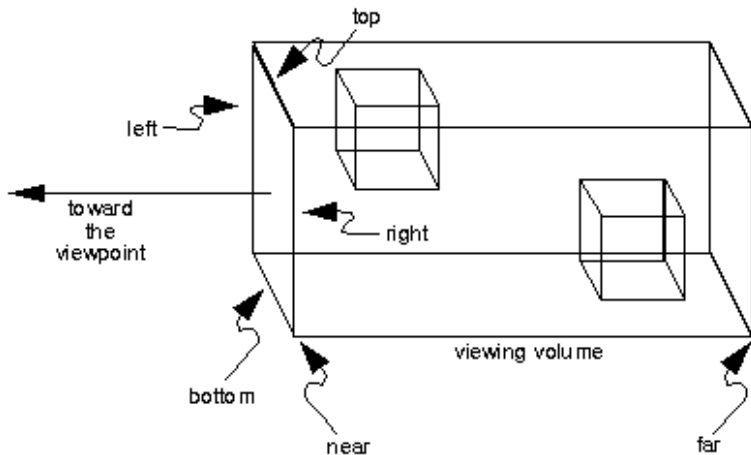
Cube.java, in the method "reshape".  
Change the parameters of gluPerspective to see the effect.

# glOrtho

## glOrtho

- Creates an orthographic parallel viewing volume.
- `glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far)`.

# glOrtho



# Demo

Cube.java, in the method "reshape".  
Change the parameters of glOrtho to see the effect.

# Aspect ratio

$(right - left)/(top - bottom) = width/height$

if  $left = -right$  and  $bottom = -top$

then  $right/top = width/height$

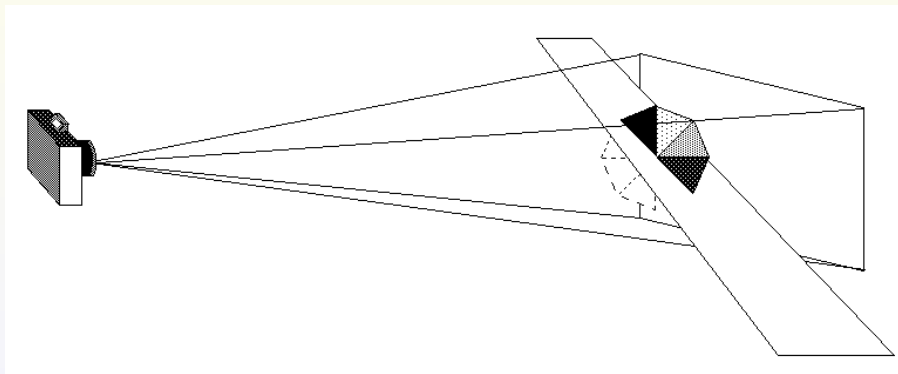
Prevent dividing integers.

# glClipPlane

## glClipPlane

- In addition to the six clipping planes of the viewing volume (left, right, bottom, top, near, and far), you can define up to six additional clipping planes to further restrict the viewing volume.
- Each plane is specified by the coefficients of its equation:  
 $Ax + By + Cz + D = 0$ .
- geometry is *\*retained\** on the side that has  
 $Ax + By + Cz + D > 0$
- `void glClipPlane(GLenum plane, const GLdouble *equation).`

# glClipPlane



# Demo

Clip.java, in the method "display".  
Change the parameters of eqn and eqn2 to see the effect.

1 Conclusions for HW1

2 Viewing

3 Q/A

Any questions?