## **9. Perspective projection**

Parallel projection is easy to understand as a concept, but what is drawn often does not fit our intuition.

Here, we study perspective projection, which is the same way humans see things.

Due to the number of hours, the principles of CG drawing including perspective projection are not covered in this lesson material.

Please read Chapter 3 Viewing of the Programming Guide Book (the red book) below and study by yourself.

**Reference link**

・[OpenGL Programming Guide Book (ver1.1, commonly known as “the red book”)](http://glprogramming.com/red/index.html)

・[OpenGL References](https://www.khronos.org/registry/OpenGL-Refpages/gl4/) : OpenGL Reference Pages

・[OpenGL References / GL Utility (GLU)](https://www.khronos.org/registry/OpenGL/specs/gl/glu1.3.pdf)

・[GLUT --The OpenGL Utility Toolkit](http://www.opengl.org/resources/libraries/glut/)

・[The OpenGL Utility Toolkit (GLUT) Programming Interface API Version 3](http://www.opengl.org/resources/libraries/glut/spec3/spec3.html)

### **09.01. Introduction of perspective projection (provisional)**

**ic2\_SetUpCamera\_Perspective()**

Instead of ic2\_SetUpCamera\_Ortho(), this time we will create a function in charge of perspective projection.

This function is going to be called from the outside and used.

Prepare a header file and declare a prototype there for calling it externally.

This function manipulates the matrix of the camera projection operation.

Therefore, at the beginning of this function, specify the matrix to be operated with GL\_PROJECTION.

Actually, glMatrixMode(GL\_PROJECTION) is called.

Note that the matrix operation rule by GL\_PROJECTION is the same as that by GL\_MODEVIEW. The current matrix is ​​multiplied by the new operation from the right.

**gluPerspective()**

It is not easy to construct a matrix that realizes perspective projection only with the OpenGL library.

Therefore, GL Utility (GLU) provides [gluPerspective()](http://www.opengl.org/sdk/docs/man2/xhtml/gluPerspective.xml).

There is an explanation of gluPerspective() in [Chapter 3 Viewing in the red book](http://glprogramming.com/red/chapter03.html).

After learning how to use it, let's see how to actually write the program.

**[Program 09-01] (Gray part is unchanged)**

1. ic2-CommonHeaders.h (Difference from)
2. 08-0109-01-Projection.c (Difference from))
3. 07-0309-01-Rendering.c (08Difference from -04)
4. 07-03-EmbededObjects.c, 07-04-Initialization.c, 07-05-Callback.c, 07-05-MainFunction.c, 08-01-GLTools.c

**GLU Library Reference**

Since gluPersPective() is included in the GLU library, so add "GLU" in your project's link settings.

See 06-01-A in Section 06.01. For instructions.

(From now on, you will be required to specify both glut and GLU.)   
(Actually, we have already set GLU to use glut because glut internally uses GLU functions.)

As you may be afraid of the title "provisional", this program does not display the expected CG.

**Exercise**

**09-01-ex1**: Why can't we see?

Try to explain the reason why the program in this section does not show anything.

**09-01-ex2**: Meaning of 4 arguments of gluPerspective() function

Explain geometrically the meaning of the 4 arguments of gluPerspective() function. Explain the concept of perspective projection by drawing it on the YZ plane and words with the figure.

### **09.02. Introduction of perspective projection (formal)**

**Positional relationship between camera and object**

OpenGL perspective projection is formulated on the assumption that the focal point is at the origin of the camera coordinate system.

This is also explained in the usage of the gluPerspective() function in 09.01.

OpenGL perspective projection is formulated on the assumption that the focus is at the origin of the camera coordinate system.

In the program in Chapter 08, lines were drawn on the Z = 0.0 plane with X and Y coordinates ranging from -1.0 to 1.0.

On the other hand, in the provisional program in section 09.01., since gluPerspective() was used, the camera focus is the origin and the camera direction is the negative Z-axis direction from the explanation in [Chapter 3 Viewing in the red book](http://glprogramming.com/red/chapter03.html). The CG objects should be placed within a distance from 2.0 to 4.0.

In other words, in order to fit the drawing range of the line art within this renderable range, it is necessary to "set back the camera".

However, as explained in [Chapter 3 Viewing in the red book](http://glprogramming.com/red/chapter03.html), it is assumed that the focus of the camera is always at the origin of the camera coordinate system when projecting 3DCG onto an image plane of the camera.

It is forbidden to move the camera from the origin of the camera coordinate system.

(Strictly speaking, it is possible to do that, but in that case, you have to calculate and prepare the projection matrix all by yourself.)

Therefore, instead of setting back the camera, all the drawing targets are moved forward just before the projection. It is moved in the negative direction of the Z-axis.

**field of view**

If the drawing range and renderable range match along the Z-axis, there is still no guarantee that the CG will be drawn as you expected.

The field of view of the camera should be adjusted to the apparent size in the X-axis and the Y-axis.

By default in OpenGL, both the projection plane (image plane) used for projection and the window actually used for display are square (aspect ratio 1.0).

Since the drawing range is from -1.0 to 1.0 for each of the X and Y axes, it is also considered to be square.

Therefore, here, we will consider only the vertical angle of view when calculating. The same setting should be valid on the discussion on the horizontal angle.

- Let the distance from the camera focus (camera coordinate system origin) to the target object (logo mark drawing surface) 3.0.

- The vertical drawing range is from -1.0 to 1.0.

Due to these constraints, the vertical field of view required to fill the window with the logo is approximately 36.87 degrees. (09-02-ex3)

**field of view and focal length**

In this program, the parameters required for the perspective projection matrix were determined from the field of view, but the focal length (distance from the camera focal length to the projection image plane) is usually used as a value that represents the properties of the camera.

To master gluPerspective(), it is necessary to assume a certain image plane and find the vertical angle of view from the focal length for it.

**[Program 09-02] (Gray part is unchanged)**

1. 09-02-Projection.c (Difference from 09-01)
2. 09-02-Rendering.c (Difference from)
3. 09-0107-03-EmbededObjects.c , 07-04-Initialization.c, 07-05-Callback.c, 07-05-MainFunction.c, 08-01-GLTools.c, ic2-CommonHeaders.h

**Exercise**

**09-02-ex1**: Location of translation functions

In the above explanation, in order to “translate all drawing targets in the negative direction of the Z-axis immediately before projection", where is the translation function should be inserted in the program source file? And how the delta Z value should be? You should think of an appropriate amount of translation just before the actual drawing points are given by glVertex3f() on the program source.

**09-02-ex2**: Translate, rotate, and scale operations

In the translate, rotate, and scale operations, "fixing the camera and moving the target" and "fixing the target and moving the camera" has exactly the same result. Explain that using linear algebra.

**09-02-ex3**: Reason for the vertical field of view to be 36.87 degrees

In the explanation of this section, explain the calculation process for the vertical field of view to be 36.87 degrees.

**09-02-ex4Find**: Focal length 10

Calculate the vertical field of view when the distance from the camera center to the object center, the focal length, is changed from 3 to 10 in 09-02-ex3.

**09-02-ex5**: Focal length and vertical field of view

Suppose that the image plane is square and the height and width of the plane are from -1.0 to 1.0. Formulate the relationship between the focal length f and the vertical field of view “fovy” (Field Of View in Y axis).

**09-02-ex6**:35mm film equivalent

Explain the meaning of "35mm film equivalent focal length z mm" in the product description of digital cameras.

**09-02-ex7**: Derived from the 35mm film conversion value

Assuming that the aspect ratio of the image surface is the same as that of the 35mm film and the vertical size is +/- 1.0, and the 35mm film equivalent focal length z mm is given as your camera. Formulate the calculation procedure to obtain the focal length f and the vertical field of view fovy to be used as the variables of gluPerspective().

### **09.03. Rotate and translate in perspective projection**

You can draw images where people feel deepness with the perspective projection.

Let's actually do various exercises and feel that.

An example of moving CG is shown below.

You are encouraged to do the exercises by yourself.

**Y-axis rotation**

Like 08-04-ex3, let's use glRotatef() to create a program that slowly rotates the logo around the Y-axis, i.e., horizontal rotation.

Rotate the CG object so that the right side comes first.

Use a translation matrix to match the renderable area with the drawing area.

Prepare a variable “roty” that represents the degree of rotation and rotate it using the MODELVIEW matrix.

Now that you know what's going on, we'll speed up the CG motion.

**[Program 09-02] (Gray part is unchanged)**

1. 09-03-Rendering.c (Difference from 09-02)
2. 09-03-Callback.c (Difference from)
3. 07-0507-03-EmbededObjects.c , 07-04-Initialization.c, 07-05-MainFunction.c, 08-01-GLTools.c, 09-02-Projection.c, ic2-CommonHeaders.h

**Exercise**

**09-03-ex1**: X-axis logo rotation

Create a program so that the logo keeps rotating around X-axis. Make it look like the top is coming down.

**09-03-ex2**: Z-axis logo reciprocation

Create a program so that the logo keeps reciprocating from -2.0 to -4.0 in the Z-axis direction.

**09-03-ex3**: Z-axis logo reciprocation at Z specified value

Like the 09-03-ex2, but this time give the center Z value (09-03-ex2 is an example case of the center Z value 3.0). The moving range should be the center -Z value +/- 1.0.

**09-03-ex4**: Reciprocation with sine wave

In the program of 09-03-ex2, modify the displacement amount from -2.0 to -4.0 of the Z-axis in proportion to a sine wave with respect to the elapsed time.

**09-03-ex5**: Translate and rotate

Create a program so that the logo makes a periodic motion with translation and rotation. Make sure that reciprocation of translation and that of rotation have different cycle lengths.