## **10. Reading a 3DCG model from a file**

We did a quick test of opening and closing a file in Section 02.

Here we will format the 3DCG model so that we can read that information from a file.

In this section, we will write a new program for reading data from a file and expanding it in memory.

CG does not appear in this section.

### **10.01. Opening / closing a file**

Reading a 3DCG model file is going to be integrated into the main program of Section 09.

It implies that the file-reading functions can be made into a library.

From the beginning, the source file is divided and programmed.

In Section 10.01., we will not read the content of the model file.

**[Program 10-01]**

1. 10-01-MainFunction.c
2. 10-01-ReadModel.c
3. ic2\_DefaultModel.txt

**Exercise**

**10-01-ex1**: Error trap

In the program in this section, point out the line that is error trapping. Show the error to be trapped and the countermeasures. If there is more than one, show them all.

**10-01-ex2**: Confirmation of error trap

Confirm that the error trap of 10-01-ex1 actually occurs.

**10-01-ex3**: Expression of define statement at runtime

How is the character string introduced by the #define statement expanded in memory when the program is executed? Check it using eclipse.

### **10.02. Reading model data from a file**

**3DCG model file syntax rules**

Consider representing a 3DCG model in a text file.

One line represents one triangle patch.

|  |
| --- |
| X0 Y0 Z0 X1 Y1 Z1 X2 Y2 Z2 R G B |

Syntax rules:

[1] One or more spaces between elements

[2] The side that looks counterclockwise when tracing three vertices in the order 0,1,2 (to the right-hand screw) is the front side (the tip side of the right-hand screw is the front side)

[3] RGB ranges from 0.0 to 1.0

[4] Ignore blank lines

[5] Ignore lines of space only

[6] Ignore lines starting with # as comment lines

In this lesson, for simplicity, only triangle patches are used as CG primitives.

(To ensure that line figure also can be managed, how would you do? How do you extend the syntax rules?)

**Use of the structure variable**

According to the above data format, we use a structure variable with the type definition of the ic2PATCH structure.

Prepare the ic2POINT structure and the ic2COLOR structure in the ic2PATCH structure.

Linked list expression is used to manage multiple patches at once.

For this, we also have a pointer inside the ic2PATCH structure.

**Header file** (ic2-ModelHeaders.h)

The function created here will be used later when creating another program.

(The library is not created, but the usage is the same as the library.)

Therefore, prepare a header file and declare the function prototype in it.

**[Program 10-02]**

1. ic2-ModelHeaders.h
2. 10-02-MainFunction.c (difference from 10-01)
3. 10-02-ReadModel.c (difference from))
4. 10-01ic2\_DefaultModel.txt (same as 10-01)

**Exercise**

**10-02-ex1**: Linked list (insert at the beginning)

Find the program source where the ic2\_InsertPATCH() function is described. Explain that part where a new patch is inserted at the beginning of the list by showing the relationship between each variable and the address in the figure.

**10-02-ex2**: Linked list (insert at the end)

Modify the right part of the ic2\_InsertPATCH() function so that a new patch is inserted at the end of the linked list.

**10-02-ex3**: Cost comparison

Please explain which of the two algorithms, 10-02-ex1 and 10-02-ex2, has the lower calculation cost at runtime. Also, consider whether the difference in computational costs is worth considering in the program revision in this section.

**10-02-ex4how**: Check the source code

Check the syntax rules [1] to [6] of the 3DCG model file are correspond to the parts of the program source. Explain each by showing the relevant part.

**10-02-ex5**: Error trap

For each function described in 10-02-ReadModel.c, write down the contents of the error trap and how the countermeasures are taken.

**10-02-ex6**: Continue on error

Even if the model file is corrupted in the middle, improve 10-02-ReadModel.c so that it will be read halfway and continue execution. Explain how you improve the program by the comments in the source file.

**10-02-ex7**: Surface normal (normal vector)

For a surface normal vector of each triangle polygon, write the formula to find the normalized normal vector. Use the symbols that appear in the syntax rules in this section.

(It is common to formulate the surface normal by the outer product of vectors)