# **Module 7-1 Final Project: Reflection**

## Arturo Santiago-Rivera

## Prof. Eugenio Rodriguez M.S, MBA

## CS330 Computational Graphics and Visualization 21EW02

## Southern New Hampshire University

## December 10, 2020

## **REFLECTION: DEVELOPMENT CHOICES**

The object modeled using modern OpenGL for this final project is a furniture chair designed by the architect Gerrit Thomas Rietveld called the Zig Zag Chair in 1934.

The Zig Zag chair designed by famous Dutch furniture designer and architect Gerrit Thomas Rietveld.



Figure 1 - The Zig-Zag chair was designed in 1934 by famous Dutch furniture designer and architect Gerrit Thomas Rietveld.

The Zig-Zag chair is simple in its appearance, but it's quite a complex construction. It uses only four planes that are joined using a system of dovetailing. All joints are placed at optimum locations for load-bearing since it has no legs. The chair's supporting part is the diagonal wood plane leading to the front of the central seating plane. It's so functional and simplistic.

The chair's uniqueness is that it's perfectly lightweight and requires a minimum amount of space for the store. The Zig-Zag chair is one of the most important chairs to exist and led the way for many others of this kind. The chair really concentrated on a functional design style that revolved around standardization, inexpensive production methods, and material.

For developing the object in modern OpenGL, we consider exercising a similar code for activity four, adding some coding for automatic rotation of the object and coding for a static text with the object's name as exercised in the activity three code. I expect to model the object as shown in the image above, adding the wood's texture to make it more realistic.

## **REFLECTION: USER CAN NAVIGATE**

Through the mouse and keyboard as input devices, the user can navigate the Zig-Zag Chair 3D by controlling a virtual camera. The application has two projections, perspective and orthographic displays of the 3D chair so that the user can change the viewport display from 2D to 3D and vice versa using the tap of the keyboard keys "p" and "o" respectively.

With the perspective projection active, the user can orbit the chair by clicking the mouse left button, pressing the ALT key, and moving the mouse in a horizontal and vertical orientation. The user can zoom in/out the chair by clicking the mouse's right button and pressing the ALT key plus moving the mouse up to zoom out and move the mouse down to zoom in. With the orthographic projection active, the user can orbit the chair by clicking the mouse left button and moving the mouse in a horizontal orientation and vertical orientation.

On any of the two projections, the 3D stays static until a keyboard or mouse action is detected. For any projection, the user can pan the 3D chair up and down using the tap of the keyboard keys "u" and "d" respectively and can pan the 3D chair left and right using the tap of the keyboard keys "l" and "r" respectively. Additionally, the user can view the 3D chair in wireframe mode by using the tap of the keyboard keys "f" and "w" to disable and enable the wireframe, respectively.

When the application is executed, their initial projection is perspective, and the wireframe mode is disabled. As described above, the keyboard keys in use intuitively match the action they perform with the keyboard key assigned.

# **REFLECTION: CUSTOM FUNCTIONS**

The program uses custom functions for navigation actions, controls, and Phong light model calculations. For the Phong light mode, we use the variable method:  
***vec3 LightCalc(vec3 fragPos, vec3 objTex, vec3 norm, vec3 viewDir, vec3 lightColor, vec3 lightPos) {*…}**That can be executed for an unlimited amount of lamps. The variable method calculates the ambient, diffuse, and specular of the lamp and returns the Phong's value.

For the user to navigate the 3D, we add keyboard functions that manage the tap of a keyboard key, *UKeyboard()*, and detect the release of the key pressed, *UKeyReleased()*. Similar actions were developed for the mouse operation in perspective projection, *UMousePers()*, and for the mouse operation in orthographic projection, *UMouseOrtho()*. For any of these mouse operation functions, the *UMouseClick()* function helps to detect the click and release of the mouse buttons. For the object's wireframe mode, we use a function that enables the wireframe mode, *WireframeModeOn()*, and another function that disable it, *WireframeModeOff()*.

As a help to the user to have visual, the navigate controls of the 3D chair, a custom function, *UControls()*, help to display in the console the all the controls describe under the topic "REFLECTION: USER CAN NAVIGATE."