Assignment 2 - Report

CSE 333/533

Name:- Aayush Kumar

Roll No.:- 2020008

Date:- 29/09/2022

Introduction

This assignment is all about the concepts of Modeling, Viewing and Projections in course CSE 333/533.

Q1

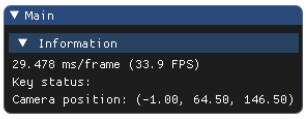
a)

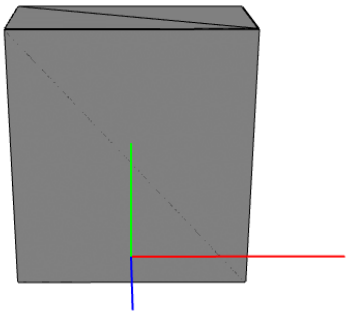
```
//setupModelTransformation(shaderProgram);
// Modelling transformation is setup in the display loop

setupProjectionTransformation(shaderProgram);

unsigned int cube_VAO, axis_VAO;
createCubeObject(shaderProgram, cube_VAO);
createAxesLine(shaderProgram, axis_VAO);

while (!glfwWindowShouldClose(window))
{
        glfwPollEvents();
        setupViewTransformation(shaderProgram);

                        // Get key presses
if (ImGui::IsKeyDown(ImGui::GetKeyIndex(ImGuiKey_LeftArrow))) {
                camPosition.x -= 3.5f;
  strcpy(textKeyStatus, "Key status: Left");
}
else if (ImGui::IsKeyDown(ImGui::GetKeyIndex(ImGuiKey_RightArrow))) {
                camPosition.x += 3.5f;
  strcpy(textKeyStatus, "Key status: Right");
}
else if (ImGui::IsKeyDown(ImGui::GetKeyIndex(ImGuiKey_UpArrow))) {
  if(io.KeyShift){
                camPosition.z += 3.5f;
    strcpy(textKeyStatus, "Key status: Shift + Up");
          }
  else{
                camPosition.y += 3.5f;
    strcpy(textKeyStatus, "Key status: Up");
          }
}
else if (ImGui::IsKeyDown(ImGui::GetKeyIndex(ImGuiKey_DownArrow))) {
  if(io.KeyShift){
                camPosition.z -= 3.5f;
    strcpy(textKeyStatus, "Key status: Shift + Down");
          }
  else{
                camPosition.y -= 3.5f;
    strcpy(textKeyStatus, "Key status: Down");
          }
}
else
  strcpy(textKeyStatus, "Key status:");
```

In the above code, on pressing the required keys, the position of the object is changing in view/camera space by a constant value with every press.The function setupViewTranformation() is being called in order to get the position in the camera view by the glm:: lookAt().

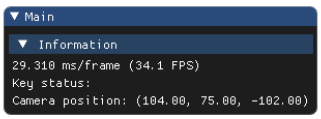
b)

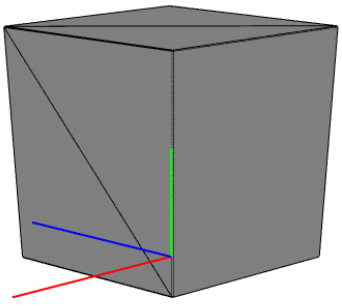
One point perspective:-





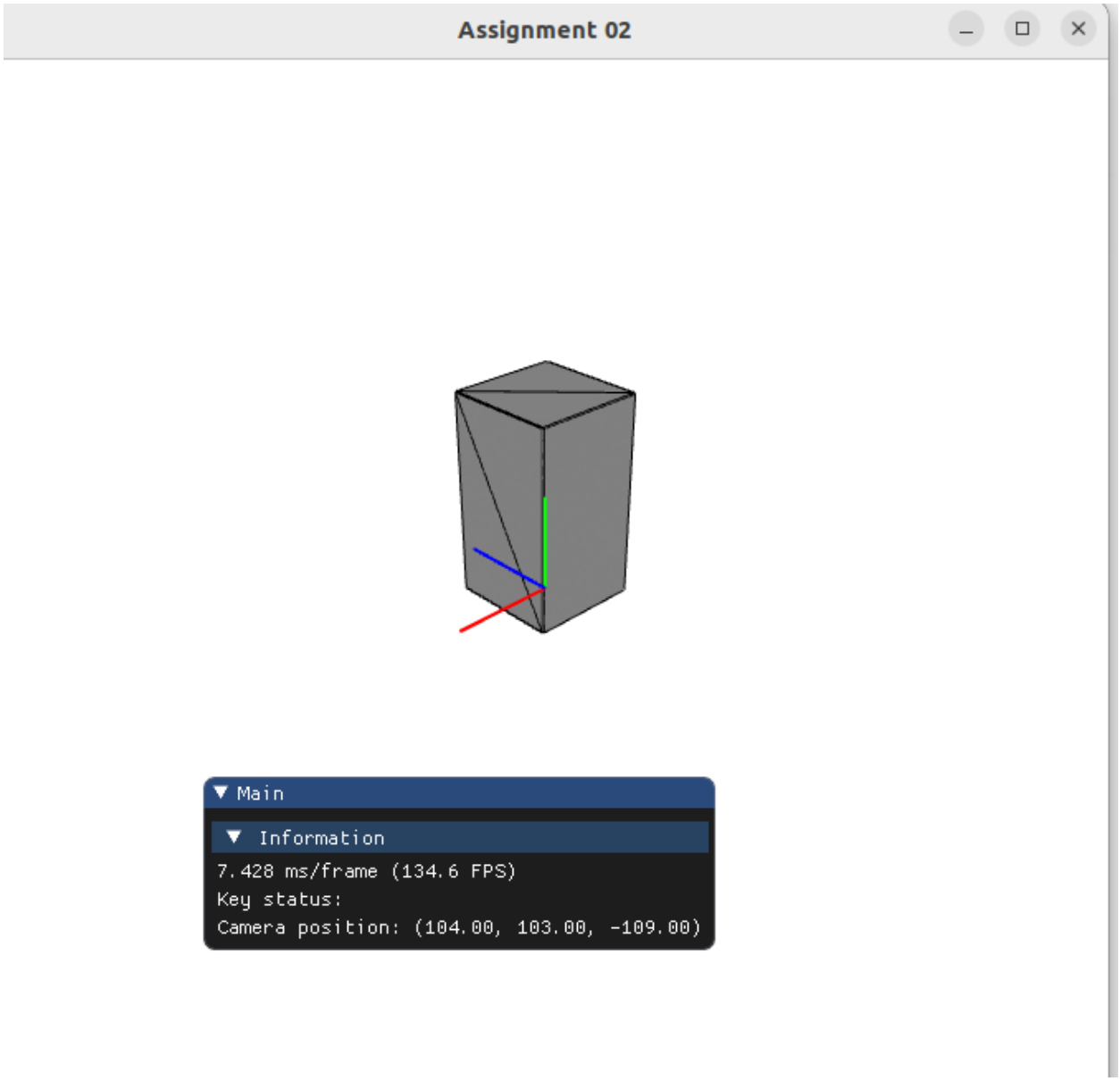
Two point perspective:-



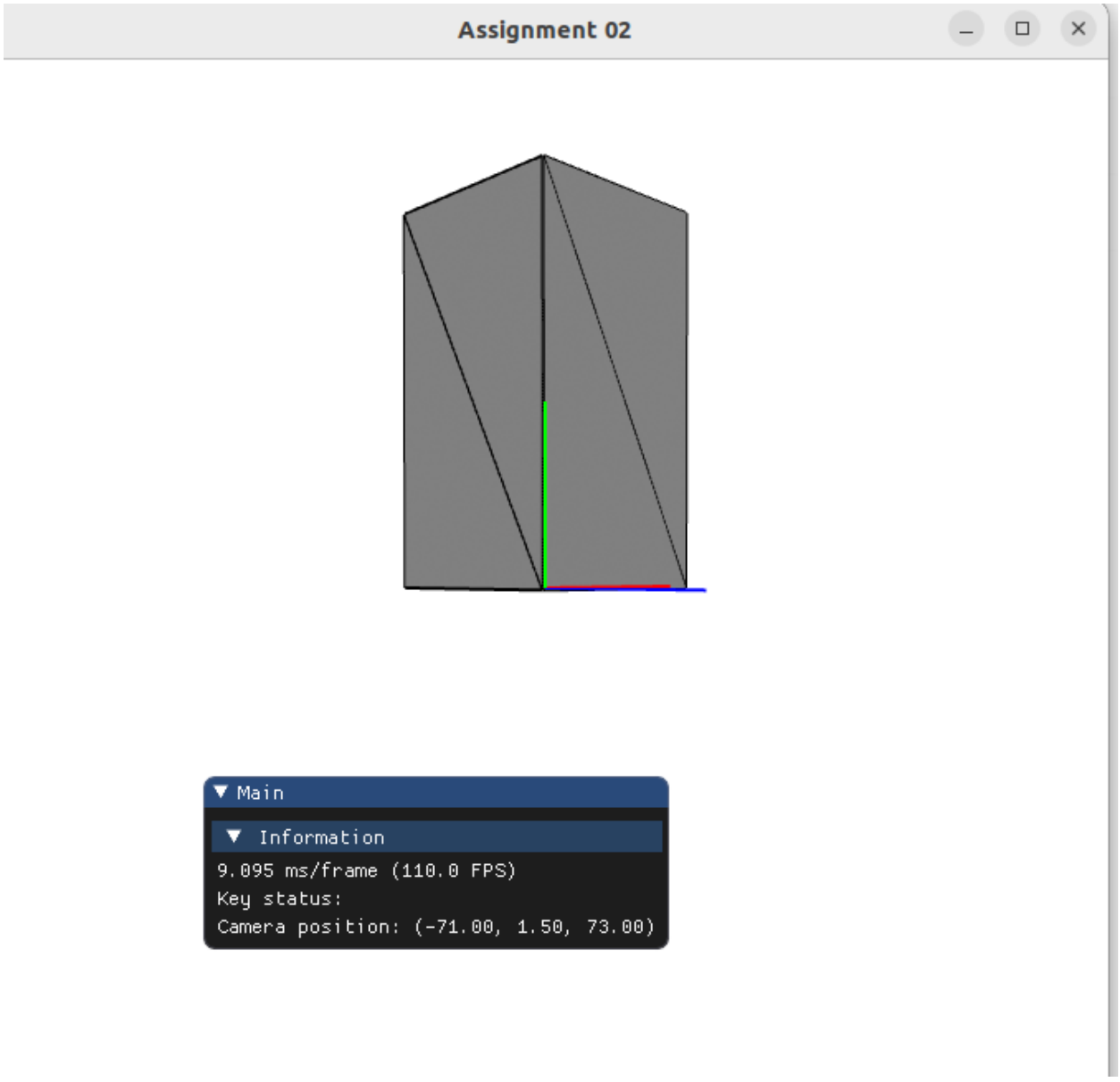


Three Point perspective:-

Bird's eye



Rat's eye



Q2

Solution of Both 1 and 2

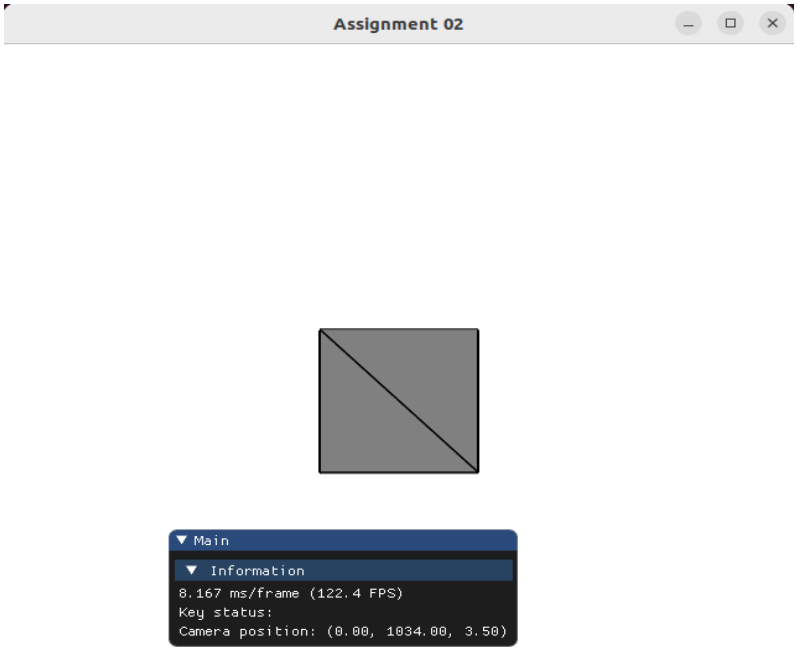
```
void setupProjectionTransformation(unsigned int &program, char temp)
{
        //Projection transformation
        if(temp == 'P'){
                projectionT = glm::perspective(45.0f, (GLfloat)screen_width/(GLfloat)screen_height, 0.1f, 1000.0f);
        }
        else if (temp == 'O'){
                projectionT = glm::ortho(-50.0f,50.0f,-50.0f,50.0f,0.1f, 1000.0f); //left,right,bottom,top,near,far
        }
```

```
                                // Get key presses
        if (ImGui::IsKeyDown(ImGui::GetKeyIndex(ImGuiKey_LeftArrow))) {
                        camPosition.x -= 3.5f;
          strcpy(textKeyStatus, "Key status: Left");
        }
        else if (ImGui::IsKeyDown(ImGui::GetKeyIndex(ImGuiKey_RightArrow))) {
                        if(io.KeyCtrl){
                        camPosition = glm::vec4(1007.0f, 0.0, 0.0f, 1.0);
                }
                else{
                        camPosition.x += 3.5f;
          strcpy(textKeyStatus, "Key status: Right");}

        }
        else if (ImGui::IsKeyDown(ImGui::GetKeyIndex(ImGuiKey_UpArrow))) {
          if(io.KeyShift){
                        camPosition.z += 3.5f;
            strcpy(textKeyStatus, "Key status: Shift + Up");
                }
                else if(io.KeyCtrl){
                        camPosition = glm::vec4(0.0, 1034.0, 3.5f, 1.0);
                }

          else{
                        camPosition.y += 3.5f;
            strcpy(textKeyStatus, "Key status: Up");
                }
        }
        else if (ImGui::IsKeyDown(ImGui::GetKeyIndex(ImGuiKey_DownArrow))) {
          if(io.KeyShift){
                        camPosition.z -= 3.5f;
            strcpy(textKeyStatus, "Key status: Shift + Down");
                }
                else if(io.KeyCtrl){
                        camPosition = glm::vec4(0.0, 0.0, 80.0, 1.0);
                }
          else{
                        camPosition.y -= 3.5f;
            strcpy(textKeyStatus, "Key status: Down");
                }
        }
                else if (ImGui::IsKeyPressed(ImGui::GetKeyIndex(ImGuiKey_A))) {
                        setupProjectionTransformation(shaderProgram,'P');
                // strcpy(textKeyStatus, "Key status: Left");
        }
                else if (ImGui::IsKeyPressed(ImGui::GetKeyIndex(ImGuiKey_Z))) {
                        setupProjectionTransformation(shaderProgram,'O');
                // strcpy(textKeyStatus, "Key status: Left");
        }
```

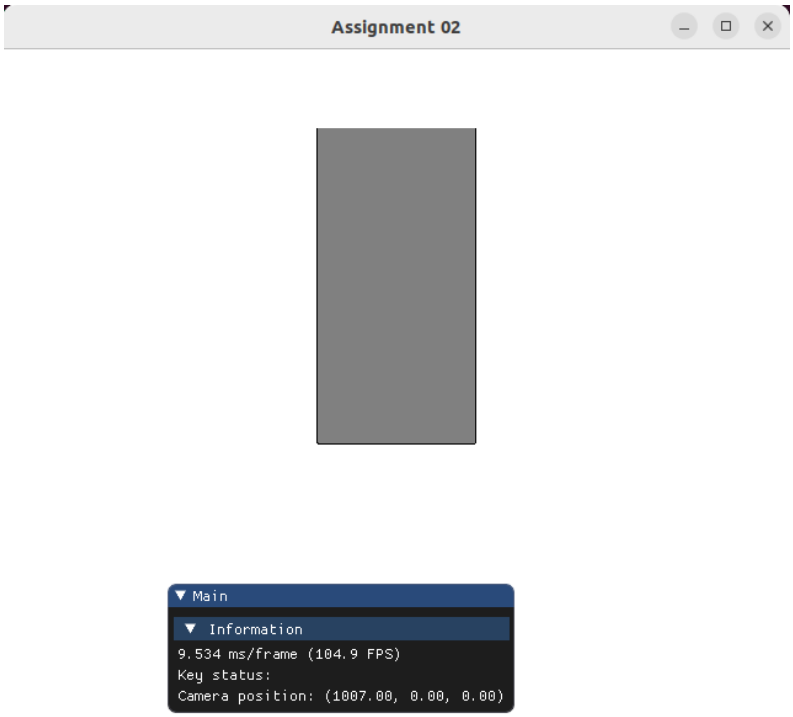
The keys A and Z represent the perspective and orthogonal view respectively using the IsKeyPressed().

The Modifier key (Ctrl Key)is being used to generate top view,front elevation and side elevation.

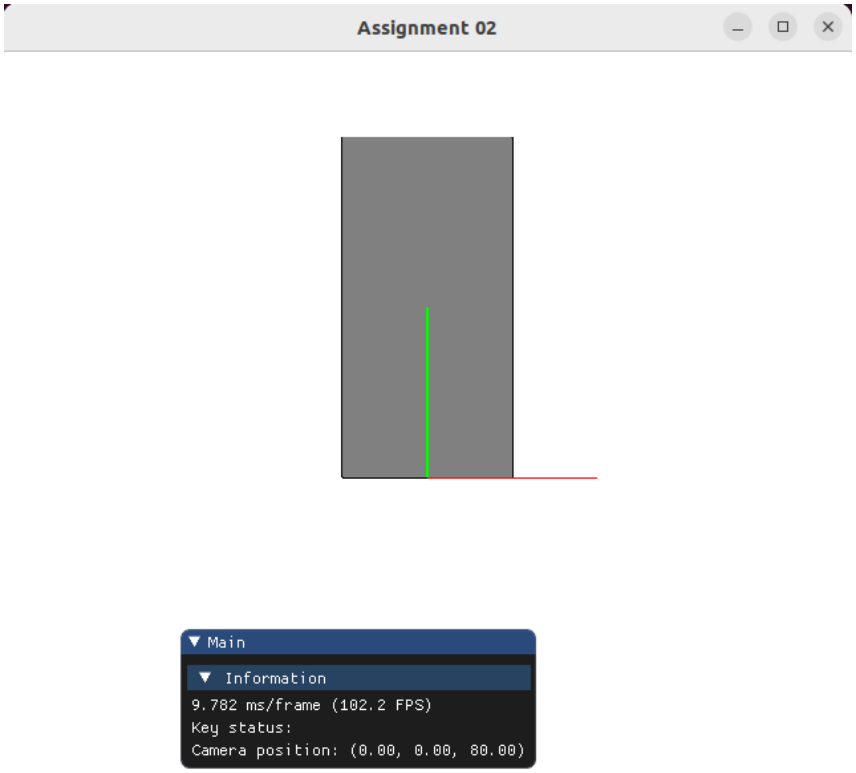
For Top View:- Ctrl + Up arrow

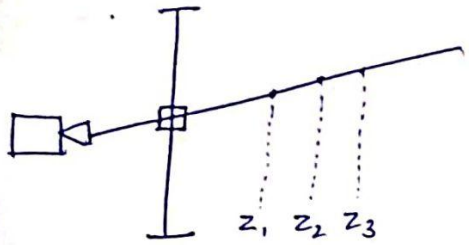


For Side elevation:- Ctrl + right arrow


Assignment 02
Main
Information
9.534 ms/frame (104.9 FPS)
Key status:
Camera position: (1007.00, 0.00, 0.00)

For front elevation:- Ctrl + down arrow





To prove :- The mapping z' leaves the order of z values intact post projection.

We know that, the size of an object on the screen is proportional to l/z for an eye at origin looking up to the -ve z axis. (Ref. Lec-9).

$z'(z)$ is such that $z'(n) = n$ for $z_1 < z_2$,
$z'(f) = f$. $(z_1)' < (z_2)'$

Now, in the given image,

$\rightarrow z_1 < z_2 < z_3$

$\rightarrow z_1' < z_2' < z_3'$

consider

$$z' = n + f - \frac{nf}{z}$$

$$z' \propto -\frac{1}{z}$$

$$z_1' < z_2'$$

$$\frac{-1}{z_1} < \frac{-1}{z_2}$$

$$z_1 < z_2 \qquad \text{— (1)}$$

similarly,

$$z_2 < z_3 \quad \& \quad z_1 < z_3 \qquad \text{— (2)}$$

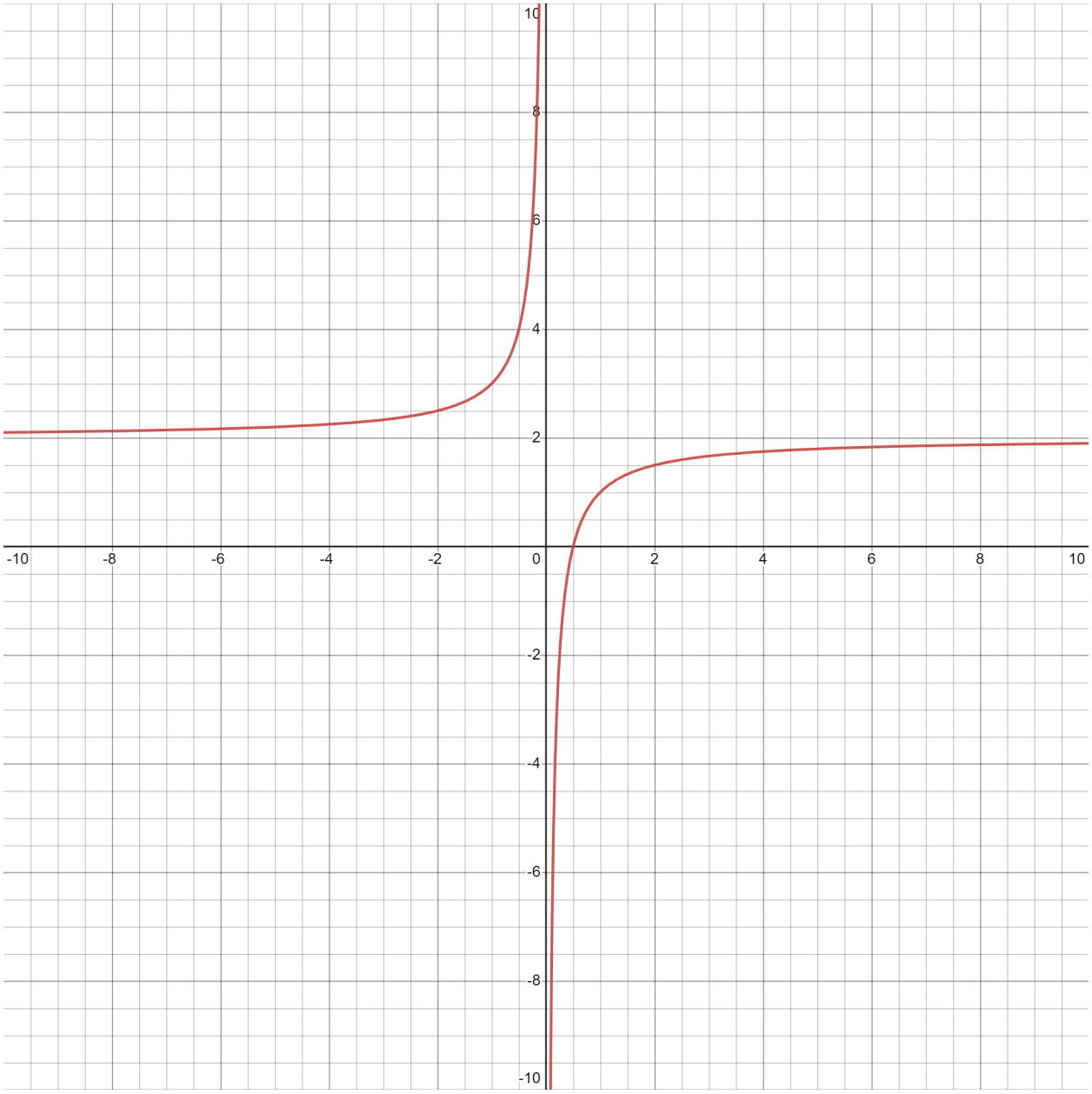
$\therefore$ From (1) & (2),

$$\boxed{z_1 < z_2 < z_3}$$

Hence, it is concluded that z' leaves the order of z-values intact post projection.

Below is the graph plotting of Z' and Z

$$z'(z) = n + f - \frac{nf}{z}.$$



$$y = n + f - \frac{nf}{x}$$

for Z' = n + f - nf/z