

Lab 3

- 2D Transformations
- 3D Transformations
- Defining models using triangular faces
- Removing hidden faces using Back-Face Culling

1.1 2D Transformations – Concatenating successive transformations

Analyze the incomplete example **WebGL_example_13.html**. Notice how the contents of the file are organized.

Identify the main changes regarding the previous examples:

- Auxiliary functions that allow instantiating and multiplying transformation matrices.
- The definition of a global transformation matrix – Model-View Matrix – and the application of successive elementary transformations.
- Passing the global transformation matrix as an argument to the vertex-shader: note how the global transformation matrix is defined and applied to each vertex.

Tasks:

- Implement and test the functionalities that allow applying the **missing elementary transformations**.
- In addition to interacting using buttons, allow the user to use the keyboard to apply successive transformations.

Suggestion:

- Select an appropriate set of keys that allow controlling, in an intuitive way, the application of elementary transformations.

1.2 2D Transformations – Updating the parameters defining the three elementary transformations being applied

Analyze the incomplete example **WebGL_example_14.html**. Notice how the contents of the file are organized.

Identify the main changes regarding the previous example:

- The global transformation matrix – Model-View Matrix – results from concatenating **three** elementary transformations: scaling, 2D rotations and translation, in that order.
- The parameters defining each transformation are directly updated.

Tasks:

- Implement and test the functionalities that allow applying the **missing elementary transformations**.
- In addition to interacting using buttons, allow the user to use the keyboard to apply successive transformations.

1.3 3D Transformations– Rotation around each of the coordinate axis

Analyze the incomplete example **WebGL_example_15.html**. Notice how the contents of the file are organized.

Identify the main change regarding the previous example:

- The global transformation matrix – Model-View Matrix – results from concatenating **five** elementary transformations, to allow for rotations around each of the coordinate axis.

Tasks:

- Implement and test the functionalities that allow applying the **rotation transformations** around the coordinate axis.
- Complete, step-by-step, the definition of the model representing a **cube centered at the coordinate axes origin**, by defining each of the cube's faces using two triangles.
- **Pay attention** to the order of the successive vertices of each triangle.
- Assign a different color to each one of the triangles.
- Allow setting/unsetting hidden face removal using **Back-Face Culling**.

Questions:

- What happens if the model intersects the faces of the view volume?
- What would happen, if all triangles defining a model had the same color?

Tasks:

- Implement and test the functionalities that allow applying the **missing elementary transformations**.
- Define and visualize other **simple models**: triangular prism, parallelepiped, etc.
- In addition to interacting using buttons, allow the user to use the keyboard to apply successive transformations.