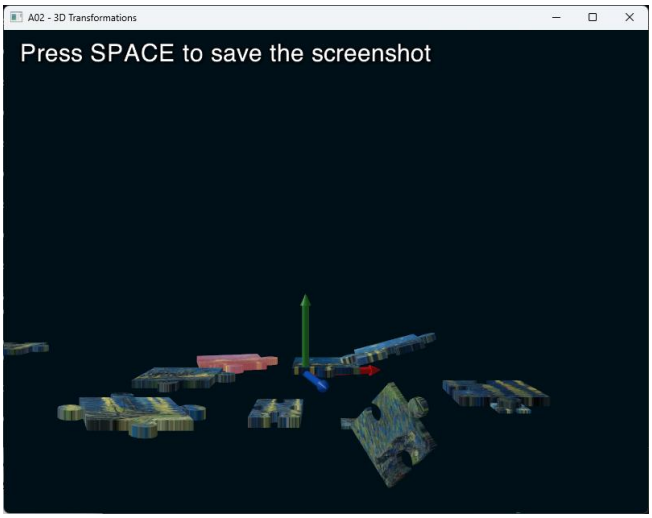
Assignment 02: 3D transformations

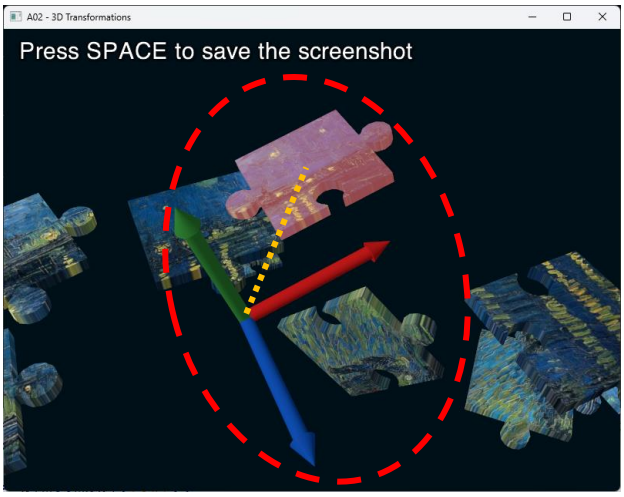
The purpose of Assignment 02 is to learn the transformation matrices with a puzzle game. When you run the application, you should see a screen with all the pieces scrambled:



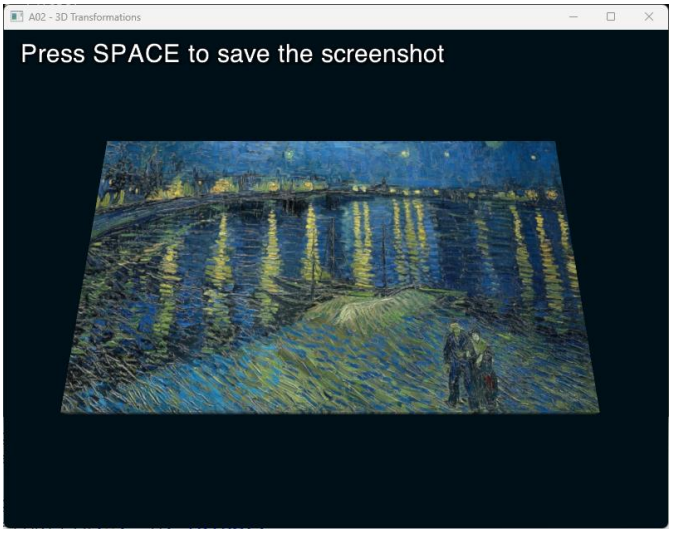
The goal is to compose the puzzle, modifying file **`transforms.hpp`** and using function:

`A->SetMatrix(int i, glm::mat4 M, bool hil)`

The first parameter **`int i`** is an integer number between 0 and 14, which defines the piece for which you want to set the matrix, **`glm::mat4 M`** is the 4x4 matrix encoding the transformation you want to set, and **`bool hil`** is a Boolean value that can be used to highlight a piece and simplify its placement. The first piece, index 0, is already in place and should not be modified. The code for piece two, index 1, is there just to show you how you can use GLM to create a transformation matrix. It also shows the effect of setting the highlight parameter to true. This piece can be set in place by simply setting its matrix to the identity, i.e. commenting lines `38` and `39` that includes the example transformation, and uncommenting line `40`. You should start working from the third piece on (index 2 to 14). Please note that most of the pieces are not centered in the origin. For this reason, when you highlight one piece, you the application also draws three arrows to highlight its origin and axes orientation.



When you succeed in assembling the puzzle, you should make sure that no piece is highlighted (that is, in all the calls to **`A->SetMatrix(…)`** the last parameter is always **`false`**), and you should see an image like this:



Once you have completed your puzzle, press SPACE: this will save a screenshot of your window in file **`A02.png`** . Please check that its content matches your window, as such file will be an important part of the final delivery of this assignment.

You can move the view using either the keyboard, the mouse or a game pad, using the controls below: