Mission offer

# Qt 3D: covering of realistic 3D objects with custom beautiful textures

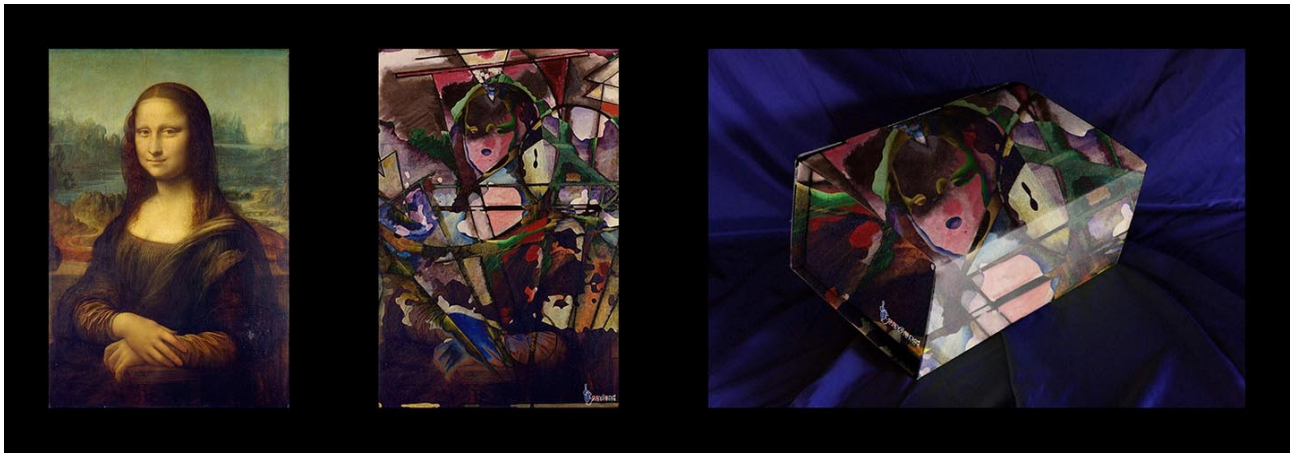
**Context**

Laoviland Experience is a french start-up dealing with graphical art creation and technology, the association between these two special worlds is our specificity. More concretely, we develop a new kind of 2D graphical creation software, called Salient Design, that helps inspiration by providing different levels of creative automation from existing pictures via tailored algorithms.
This leads in particular to a new artistic trend: the Salient Art.

To see more, visit our website : https://laoviland.com

**The mission**

Laoviland Salient Design makes you able to create stunning images, patterns, textures. These graphical elements can be used in various industries of materialized graphics, for instance textile (T-Shirt) or furniture (sofa, seat, carpet, …).
With this in mind, we would like to include in Salient Design a module that makes our users able to preview their graphical pattern in real situation. For instance, you made a beautiful pattern, and you would like to see it covering a sofa, in a well-lighted living room.
This is a 3D preview of the scene, and we provide many scenes. The mission is to provide code to load a 3D scene and replace a given texture in it, and display it to the user. Then, we need a UI to apply a transform to the texture (2D shift, rotation, 2D scale, shear…).



*Left: The original artwork by L. da Vinci*
*Middle: An example of artwork that you can realize with Laoviland Salient Design*
*Right: An example of covering with this artwork, on a postal parcel*

**Technological concerns**

Salient Design is made in Qt5 (C++17). The 3D preview module has to be made using the quite young Qt 3D library. We will provide predefined 3D graphic scenes to include in the preview (which we will choose for their different objects to cover, like canvas on the wall, furniture, and so on), but obviously a dynamic version (able to load any kind of 3D scene file, using QSceneLoader) would be a plus. The mission can be realized using either the C++ Qt API or QtQuick, even if we would prefer C++ over QtQuick because of interfacing with our full-C++ software. Platform : All non-mobile Qt-supported (we currently only support Windows for our software, but we chose Qt to be able to port to MacOS and Linux soon).

If you would like to apply to this mission offer, please contact us with this adresse: contact@laoviland.com