

CEGUI – Tutorial: Creating a simple menu

There are different ways to create a simple user interface: using C++ or XML. Here we will focus on using XML files to keep the compilation time low. There are three types of XML files: .layout, .imageset and .scheme.

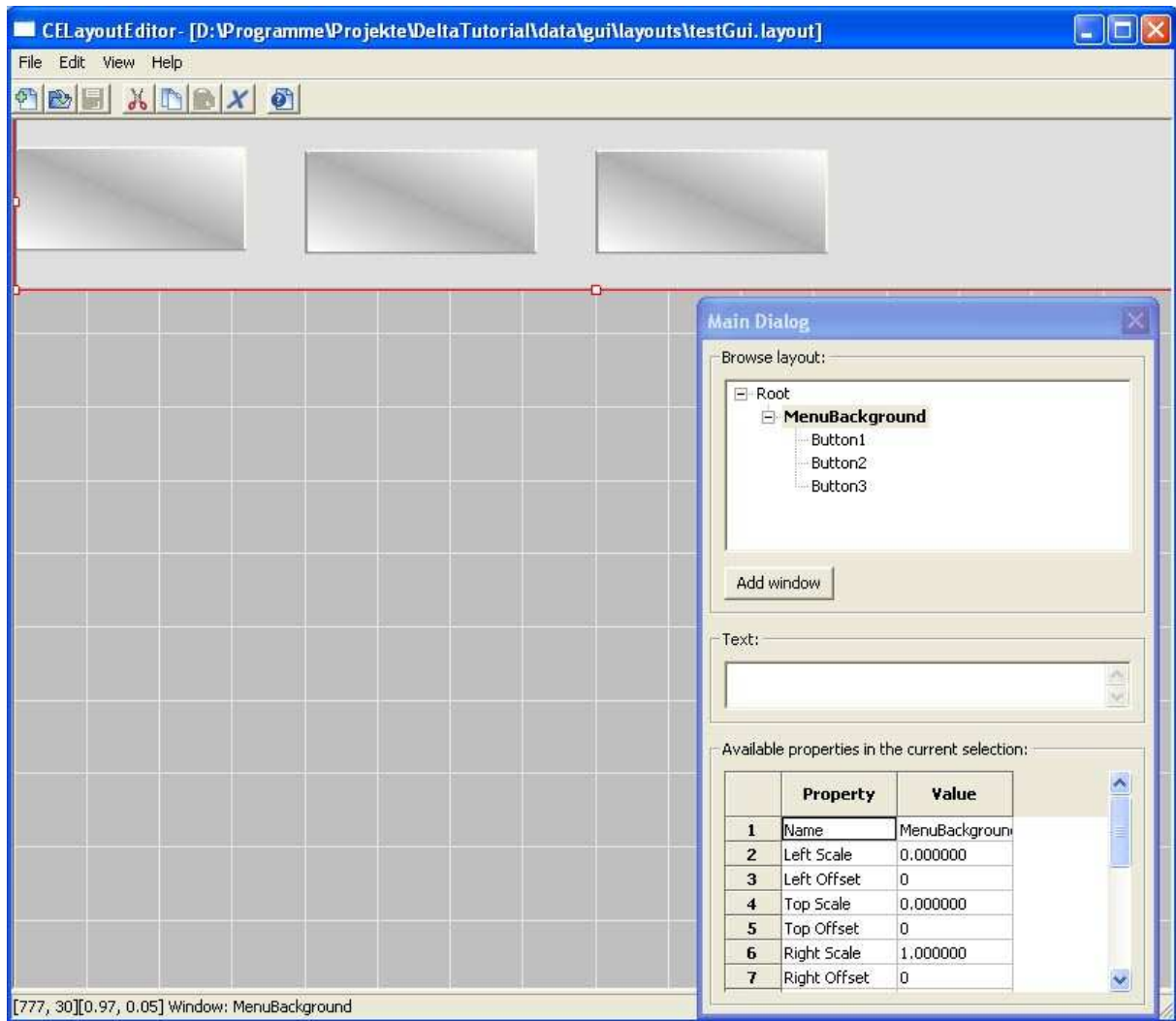
The menu will contain some buttons and a StaticImage as a background. The major steps will be explained now:

- **Creating the .layout file**
- **Creating the .imageset file**
- **Loading the Gui in the application**
- **Event Handling**
- **Links**

Creating the .layout file

You can start creating your interface using CELayoutEditor which can be downloaded here (<http://www.cegui.org.uk/wiki/index.php/Downloads>). Here are the basics steps to create the .layout file:

- Main Dialog: Add Window
 - o The name for the new window: MenuBackground
 - o Available windows look: WindowsLook
 - o Available window types: WindowsLook/StaticImage
- Change window properties
 - o The name for the new window: Button1(-3)
 - o Available windows look: WindowsLook
 - o Available window types: WindowsLook/Button
- Change window properties



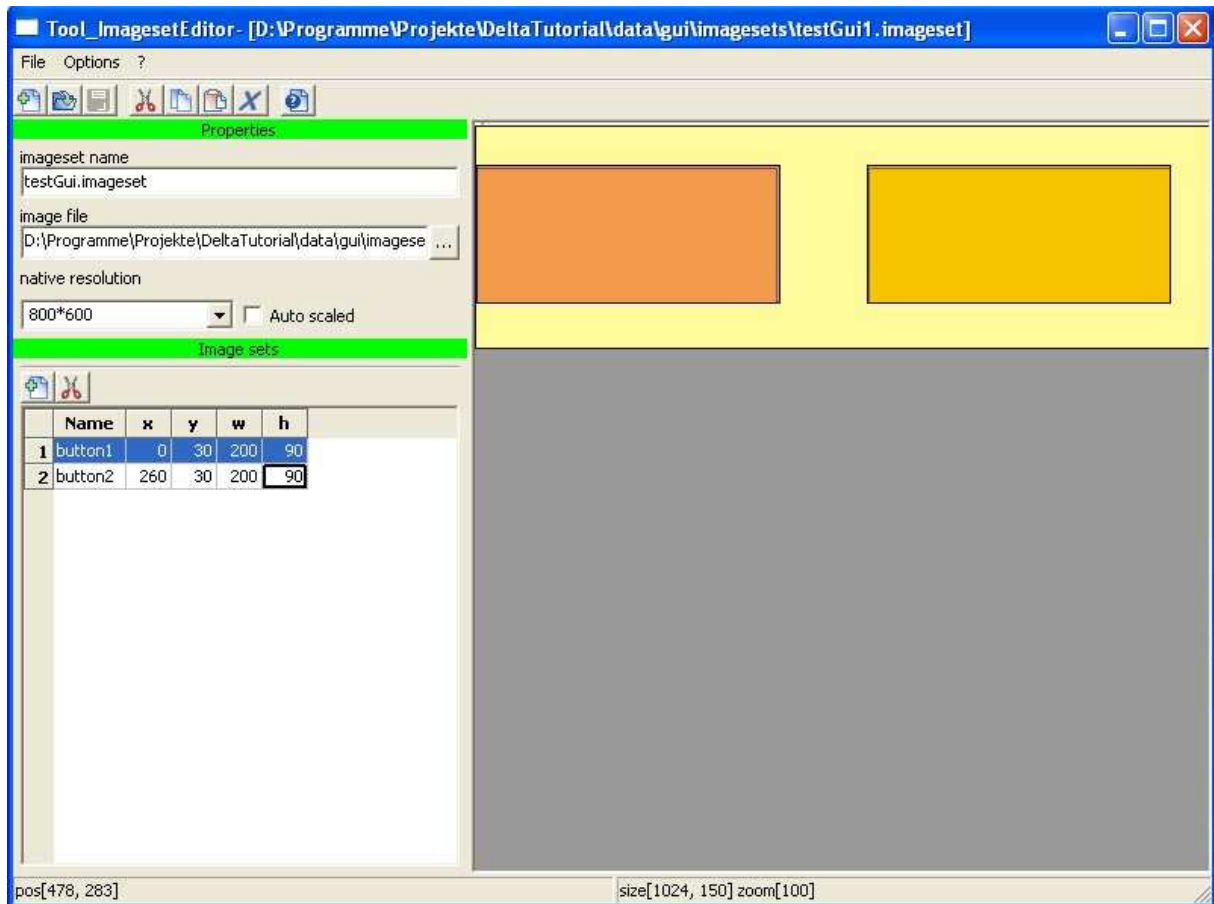
If you are finished placing the StaticImage and the buttons you save the file as testGui.layout file in the folder ../data/gui/layouts. (testGui.layout – download)

Creating the .imageset file

The next step contains the creation of an .imageset file. An imageset file is just a collection of defined regions upon the source image/texture file. Each of these defined regions has a unique name and is known within the system as an Image. An Image as defined in an imageset is the basic level of imagery used by CEGUI. By modifying the source image/texture file and also the position and size of the defined regions within the imageset files you can easily change the appearance of what gets drawn by CEGUI. One convenient way to create the .imageset file and defining the regions of the source image/texture file is using the CEImagesetEditor which can be downloaded here (<http://mfleurent.free.fr/CEGUI/CEImagesetEditor.zip>). After unzipping the executable file can be found under ../CEImagesetEditor/Tools/bin/Tool_ImagesetEditor.exe.

There you load an image and define the regions of the images which should be placed on the buttons and the background image.

If you are finished defining the regions save the file as testGui.imageset file in the folder ../data/gui/imagesets. (testGui.imageset – download)



To add the images to the button you have to make the following changes in the .layout file:

```
<Window Type="WindowsLook/Button" Name="Button1" >
    ...
    <Property Name="NormalImage" Value="set:testGui image:Button1" />
    <Property Name="HoverImage" Value="set:testGui image:Button2" />
    <Property Name="PushedImage" Value="set:testGui image:Button1" />
    <Property Name="DisabledImage" Value="set:testGui image:Button2" />
    <Property Name="Tooltip" Value="Button 1"/>
    <Event Name="Clicked" Function="HandleButtonCB"/>
</Window>
```

Loading the Gui in the application:

- load the .scheme file

```
std::string schemeFileName = osgDB::findDataFile("schemes/WindowsLookSkin.scheme");
```

- load the imageset file

```
CEGUI::ImagesetManager::getSingletonPtr()->createImageset("imagesets/testGui.imageset");
```

- load the .layout file

```
if (!mLayoutFilename.empty())
{
    //load GUI layout from file
    CEGUI::Window *w = mWnd->loadWindowLayout( mLayoutFilename.c_str() );
    mSheet->addChildWindow(w);
}
```

Event Handling:

Event handling can be realized using `dtGUI::ScriptModule`.
A way how event handling is initialized in the main-application:

```
ocsGUI::testGui testGui;
    // Event-Handling
    dtGUI::ScriptModule::HandlerFuncor callbackbtn1( &testGui,
&ocsGUI::testGui::HandleButton1CB );
    sm->AddCallback( "HandleButton1CB", callbackbtn1 );
    dtGUI::ScriptModule::HandlerFuncor callbackbtn2( &testGui,
&ocsGUI::testGui::HandleButton2CB );
    sm->AddCallback( "HandleButton2CB", callbackbtn2 );
    dtGUI::ScriptModule::HandlerFuncor callbackbtn3( &testGui,
&ocsGUI::testGui::HandleButton3CB );
    sm->AddCallback( "HandleButton3CB", callbackbtn3 );
```

The actual event handling function is defined in the testGUI class:

```
bool testGui::HandleButton1CB( const CEGUI::EventArgs &e )
{
    std::cout << "Button 1 clicked." << std::endl;
    return true;
}
```

Download the whole project testGUI.zip.

Links:

CELayoutEditor: <http://www.cegui.org.uk/wiki/index.php/Downloads>

CEImageSetEditor:

<http://www.cegui.org.uk/phpBB2/viewtopic.php?t=1472&highlight=ceimageseteditor>
<http://mfleurent.free.fr/CEGUI/CEImageSetEditor.zip>

CEGUI Main Page: <http://www.cegui.org.uk>

CEGUI Forum: <http://www.cegui.org.uk/phpBB2/index.php>

CEGUI Tutorial for a menu: http://www.cegui.org.uk/wiki/index.php/The_Main_Menu