
Remoxly Documentation

Release 0.0.1

roxlu

March 24, 2014

Contents:

Getting started

While working on interactive simulations and installations for museums and conferences we've used many different GUIs to help us to get just the correct settings for the simulation. In fact it often feels like most of our time goes into tweaking the parameters to get the exact result we like. Our projects involve using projectors too. Controlling a GUI on something which gets projected on a non-flat surface is not so easy. Creating this GUI is a step towards a GUI that can be directly controlled inside your application but also through through the web, without the need of configuring a router. This means that you can control your application from anywhere and anytime, all in a safe way.

After creating several complete GUIs we have enough experience to create a new one with all experience from the previous times. It will be a basic and blazingly fast GUI.

The goal of this project is to create a couple of GUIs. One will be directly used inside your OpenGL/DirectX* application. Another one can be used in a browser to directly control the parameters of your (C/C++) application over the internet, through firewalls. This also means that the GUI will provide a way to use your mobile phone to tweak parameters. And, if this isn't enough, a local networked GUI will be created too.

For this project we develop a couple of separate classes which can be useful for other projects. We've tried to keep these reusable pieces separate from the GUI, although the GUI needs them. Therefore we have several sub-projects:

Sub Projects

- BitmapFont: A BitmapFont renderer using BMFont files.
- FontBaker: Bakes C++ headers with pixel and font data.
- TextInput: Uses BitmapFont to create a text input type.
- Gui: This is -the- gui for you OpenGL/DirectX apps.

1.1 Getting the code

We're using github to keep track of our changes and all related projects. To get the code you should clone the latest version from Github which can be done using the following steps:

```
mkdir remoxly
cd remoxly
git clone git@github.com:roxlu/remoxly.git
```

1.2 High Level Description

This page will describe how the code for this project is structured, starting with a high level description followed some more in depth paragraphs describing the inner working of the GUI.

Widgets Every interactive element of the GUI is called a Widget. The widget is the bass type for e.g. the Slider, Toggle and Button. But also the Gui element and the Panel element inherit from Widget. The widget class handles all things like mouse presses, mouse releases, mouse movenent, key presses, creates the elements that get drawn etc. See the Widget Code Info page for more info.

Gui The Gui element extends the Widget class and is a container for elements like Sliders, Buttons, Toggles, etc.. It's a pure container type which takes also care of aligning the added sub-widgets.

Panel The Panel is another container type. But now it's not a container for types like Sliders, Toggles, Buttons, etc. but it's a container for other Guis. The Panel element allows you to stack multiple Guis on top of each other. A Panel is given a certain height. When the contained elements extend the height you can use the scrollbar to scroll to the desired element.

Render Render is the interface for the graphics API. Using a Render interface allows us to use this GUI in OpenGL, DirectX etc.. At this moment we focus on getting everything ready for OpenGL. After that an implementation for DirectX will follow.

Storage The Storage interface is used to save and load the settings of a Panel, Gui or multiple Panels, Guis. For now Remoxly ships with a basic XML storage class that uses the fantastic and uber fast rapidxml parser.

Programmers Guide

This page describes what you need to get started with the Remoxly. Remoxly is a GUI that you can use directly in your C/C++ OpenGL applications.

Indices and tables

- *genindex*
- *modindex*
- *search*