

How to add a new effect to Nutcracker.

Sean Meighan

March 4, 2013

Assumptions:

You already have Code::Blocks installed and configured

You have installed the mingw C++ compiler

You have compiled and created the wxWidgets libraries.

The test of the above 3 is that you can build the xLights project. Setting up your environment is in the ToolConfig.txt document in the nutcracker_c directory.

If yes, now lets go through the steps to add a new effect into the Nutcracker tab in xLights.

- 1) Add Your effect to the wxWidgets screens of xLights.
- 2) Clone a copy of an existing effect (Use RenderTwinkle)
- 3) Modify 4 files , steps 2-6 below.
- 4) Compile project. When you select “YourEffect”, or whatever name you picked, it should create a twinkle effect.
- 5) Modify your RenderYourEffect.cpp and make the effect do what you really expect.
- 6) Run “Inno Setup” to make a setup.exe file. Open the file xLights.iss in the nutcracker_c directory. Run Compile to build a setup.exe, it will be found in ../nutcracker_c/output

Here are the 6 places that code needs to be modified to add your effect. The actual effect will be stored in step (1). The easy way to find these locations is to do a search through the project for “RenderTwinkle”.

(1) RgbEffects.cpp.h: Copy RenderTwinkle.cpp lines 1170-1208 and paste at bottom of file. Rename render name to match your new effect.

```
void RgbEffects::RenderYourEffect(int Count)
{
    ..
    ..    Copy all this as your first stub.
}
```

(2) pixelBuffer.cpp: Near line 321

```
void PixelBufferClass::RenderYourEffect(int Count)
{
    Effect[CurrentLayer].RenderYourEffect(Count);
}
```

(3) pixelBuffer.h: Near line 77

```
void RenderYourEffect(int Count);
```

(4) RgbEffects.h: Near line 64

```
void RenderYourEffect(int Count);
```

(5) TabSequence.cpp: Near line 676. Note you need to be sure you named your Slide widget with a name like:

```
"Var Name = "Slider_Twinkle1_Count"
```

```
"Identifier" = "ID_SLIDER_Twinkle1_Count"
```

So following this naming scheme, if your effect was called "YourEffect", then your slider widget would have these two variables as

```
"Var Name = "Slider_YourEffect1_Count"
```

```
"Identifier" = "ID_SLIDER_YourEffect1_Count"
```

This is the only tricky part of setting up a new effect, naming these two variables. Why a 1? Because you can have an Effect1 and an Effect2. This setup is just getting the left side, Effect1 to work. Notice in the code snippet below how the layer value gets added into the variable name. LayerStr has either the value "1" or "2" if you are clicking in the Effect1 or Effect2 space.

```
else if (effect == wxT("YourEffect"))
{
```

```
buffer.RenderTwinkle(wxAtoi(SettingsMap[wxT("ID_SLIDER_YourEffect")+LayerStr+wxT("_Count")]));
```

```
}
```

(6) TabSequence.cpp: Near line 783. Now we need to add our choice into the switch selection. This code starts with

```
switch (Choicebook1->GetSelection())
{
case 0:
    break;    // none
case 1:
```

```
..
..
```

```
case 15:    // whatever is the next new number
```

```
        buffer.RenderYourEffect(Slider_YourEffect1_Count-  
>GetValue());  
        break;
```

Enjoy your new effect in Nutcracker 3.0!