Subject: [Feature+Patch] CodeEditor: Text Highlighting
Posted by omari on Mon, 13 Jan 2020 20:28:34 GMT
View Forum Message <> Reply to Message

CodeEditor.h:

```
class CodeEditor : public LineEdit,
public HighlightSetup
{

..

 static Array<CodeEditor*> instances;

..
 void   StdBar(Bar& menu);
..
```

CodeEditor.cpp:

```
Array<CodeEditor*> CodeEditor::instances;

VectorMap<String, int>& highlightList()
{
 static VectorMap<String, int> map;
 return map;
}

#define CUSTUM_Hi_COUNT 11

static Color highlightColors[CUSTUM_Hi_COUNT] =
{
  Color(255, 105, 30),  // 01
  Color(0xC0, 0xC0, 0xC0), // 02
  LtMagenta(),    // 03
  LtCyan(),     // 04
  Color(198, 255, 50),  // 05
  Color(255, 0xE4, 0xE1),  // 06
  Color(0x32, 0xCD, 0x32), // 07
  Color(0x5E, 0x90, 0xFF), // 08
  Color(0x7F, 0xFF, 0xD4), // 09
  Color(0xFF, 0xD7, 0x00), // 10
  Color(0xAF, 0xEE, 0xEE)  // 11
};
```

```
static Color highlightColorsDark[CUSTUM_Hi_COUNT] =
{
  Color(120, 50, 20),   // 01
  Color(0x60, 0x60, 0x60), // 02
  Magenta(),     // 03
  Cyan(),      // 04
  Color(99, 126, 25),   // 05
  Color(126, 0x72, 0x71),  // 06
  Color(0x16, 0x70, 0x16), // 07
  Color(0x30, 0x45, 0x88), // 08
  Color(0x40, 0x88, 0x79), // 09
  Color(0x88, 0x66, 0x00), // 10
  Color(0x55, 0x77, 0x77)  // 11
};

static Image mkImg(int i, int cx = 16, int cy = 16)
{
 ImageDraw w(cx,cy);
 w.DrawRect(Size(cx,cy), IsDarkTheme()? highlightColorsDark[i] : highlightColors[i]);
 return w;
}

void CodeEditor::StdBar(Bar& menu)
{
 static auto RefreshInstances = []{
   for(auto e: instances)
    e->Refresh();
  };
 menu.Sub(t_("Highlight"), [&](Bar& bar) {
  String a = GetSelection();
  if(a.GetCount()){
   if(highlightList().Find(a) >= 0){
       bar.Add(t_("UnHighlight"), [a, this]{highlightList().RemoveKey(a); RefreshInstances();});
    bar.Separator();
   }
   for(int i = 0; i < CUSTUM_Hi_COUNT; i++){
    bar.Add( t_ ("style ") + AsString(i + 1), [a, i, this]{highlightList().GetAdd(a) = i;
RefreshInstances();}
     ).Image(mkImg(i));
   }
   bar.Separator();
  }
  bar.Add(t_("Clear Highlighting"), [this]{highlightList().Clear(); RefreshInstances();});
 });
 LineEdit::StdBar(menu);
}
```

```
void CodeEditor::HighlightLine(int line, Vector<LineEdit::Highlight>& hl, int64 pos)
{
...

 if(highlightList().GetCount() > 0){
  HlStyle st = hl_style[PAPER_SELWORD];
  for(int i = 0; i < highlightList().GetCount(); i++)
  {
   String sel = highlightList().GetKey(i);
   int n = highlightList().Get(sel);
   st.color = IsDarkTheme()? highlightColorsDark[n] : highlightColors[n];
   int q = 0;
   for(;;) {
    q = l.Find(~sel, q);
    if(q < 0)
     break;
    int h = q + sel.GetCount();
    for(int i = 0; i < sel.GetCount() && i + q < hl.GetCount(); i++) {
     hl[i + q].paper = st.color;
     if(st.bold)
      hl[i + q].font.Bold();
    }
    q = h;
   }
  }
 }
}


CodeEditor::CodeEditor() {
...
 instances.Add(this);
}



CodeEditor::~CodeEditor() {
 for(int i = 0; i < instances.GetCount(); i++){
  if(instances[i] == this){
   instances.Remove(i);
   break;
  }
 }
}
```

File Attachments

1) highlighting.PNG, downloaded 622 times