Subject: Re: EscPainter package, a painter extension for Esc scripting language
Posted by Oblivion on Mon, 08 May 2023 20:27:32 GMT
View Forum Message <> Reply to Message

Hi,

A new round of weekly updates:

Most of the missing Painter methods are implemented in EscPainter. Also, I have added a "headless" (console) painter example that re-creates the text on path example with EscPainter.

EscPainter package brings the power of U++'s Painter library to U++'s Esc scripting language, allowing SVG/PDF quality images/graphs via simple scripting.

Currently Implemented painter methods:

Begin()
End()
Clip()
ColorStop(pos, color)
ClearStops()
Opacity(o)
LineCap(l)
LineJoin(l)
MiterLimit(l)
EvenOdd(b)
Invert(b)
Background(color)
Stroke(...)
Fill(...)
Dash(...)
Translate(...)
Rotate(r)
Scale(...)
Move(...)
TopLeft()
TopRight()
TopCenter()
BottomCenter()
BottomLeft()
BottomRight()
Center()
Line(...)
Circle(...)
Ellipse(...)
Arc(...)
Path(x)

Cubic(...)
Quadratic(...)
BeginOnPath(...)
Rectangle(...)
RoundedRectangle(...)
GetSize()
GetRect()
GetCenterPoint()
Text(...)
Character(...)
GetTextSize(...)

Headless painter example:

```
// This example demonstrates a headless drawing example, using U++'s Esc scripting language.
#include <Core/Core.h>
#include <EscPainter/EscPainter.h>
#include <plugin/png/png.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
 StdLogSetup(LOG_COUT | LOG_FILE);

 ArrayMap<String, EscValue> global;

 StdLib(global);  // Include Esc standard library.
 PainterLib(global); // Include Esc painter library.

 try
 {
  PNGEncoder().SaveFile(
   GetHomeDirFile("EscPainterOutput.png"),
   EscPaintImage(global, LoadFile(GetDataFile("script.usc")), 1024, 1024));
 }
 catch(CParser::Error e)
 {
  RDUMP(e);
 }
}
```

The "text on path" script:

```
Paint(w) // Text on path example.
{
 M_PI = 3.14159265358979323846;
 fnt  = :Roman(100);
 text = "Hello world, this is a text on path!";
 l    = GetTextSize(text, fnt).cx;
 r    = l / (2 * M_PI);
 pos  = 0;

 w.Begin()
  .Background(:White)
  .Center()
  .Circle(0, 0, r)
  .Stroke(1, :LtRed);
 for(i = 0; i < count(text); i++) {
  c = text[i];
  x = fnt.GetWidth(c);
  w.BeginOnPath(pos + x / 2,  1)
   .Character(-x / 2, -fnt.GetAscent(), c, fnt)
   .Fill(0, -fnt.GetAscent(), :Yellow, 0, fnt.GetDescent(), :Green)
   .Stroke(1, :Red)
  .End();
  pos += x;
 }
 w.End();
}
```

Best regards,
Oblivion

---