Subject: How to achieve dynamic layout scaling ?
Posted by mingodad on Tue, 05 Jan 2021 15:12:41 GMT
View Forum Message <> Reply to Message

I'm starting with Ultimate++ and I'm looking on how to achieve dynamic layout resizing (something like webrowser CTRL-Mouse-Wheel), I started with Gui15 project and added a CtrlSlider to change the StdFont with the size from the slider.

But the only button on the window only changes the label text but not it's size, can we have an example showing it ?

```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

struct MyAppWindow : TopWindow {
 Button button;
 SliderCtrl slider;

 void Slider()
 {
  //Log(String().Cat() << "-> SliderCtrlAction(" << slider.GetData() << ")");
  Font::SetStdFont(Arial(slider.GetData()));
  button.SetLabel(String() << "Button " << slider.GetData());
  Refresh();
 }

 typedef MyAppWindow CLASSNAME;
 MyAppWindow() {
  Title("My application with font-zoomed button").Sizeable();
  *this << button.SetLabel("Button").LeftPosZ(10, 64).TopPosZ(10, 24);
  slider.MinMax(10, 24);
  slider.Step(1);
  slider.SetData(14);
  slider << THISFN(Slider);
  *this << slider.TopPosZ(10, 64).LeftPosZ(180, 10);
 }

};

GUI_APP_MAIN
{
 //Font::SetDefaultFont(Arial(24));
 Font::SetStdFont(Arial(18));
 //StdFont(18);
 MyAppWindow app;
 app.Run();
```

```
}
```

Subject: Re: How to achieve dynamic layout scaling ?
Posted by mingodad on Wed, 06 Jan 2021 14:24:07 GMT
View Forum Message <> Reply to Message

The idea is to resize the window and have everything resized proportionally (widgets, fonts).

Here is a small calculator layout to use as a basis to show possible ways to achieve "Dynamic Layout Scaling".

- CalculatorDynLayout.h

```
#ifndef _CalculatorDynLayout_CalculatorDynLayout_h
#define _CalculatorDynLayout_CalculatorDynLayout_h

#include <CtrlLib/CtrlLib.h>

using namespace Upp;

#define LAYOUTFILE <CalculatorDynLayout/CalculatorDynLayout.lay>
#include <CtrlCore/lay.h>

class CalculatorDynLayout : public WithCalculatorDynLayoutLayout<TopWindow> {
public:
 CalculatorDynLayout();
};

#endif
```

- main.cpp

```
#include "CalculatorDynLayout.h"

CalculatorDynLayout::CalculatorDynLayout()
{
 CtrlLayout(*this, "Calculator");
}

GUI_APP_MAIN
{
 CalculatorDynLayout().Sizeable().Run();
}
```

- CalculatorDynLayout.lay

```
LAYOUT(CalculatorDynLayoutLayout, 160, 136)
 ITEM(Upp::Label, entry, SetFrame(BlackFrame()).HSizePosZ(12, 8).TopPosZ(8, 19))
 ITEM(Upp::Button, clear, SetLabel(t_("C")).LeftPosZ(80, 20).TopPosZ(32, 15))
 ITEM(Upp::Button, clearAll, SetLabel(t_("AC")).LeftPosZ(104, 20).TopPosZ(32, 15))
 ITEM(Upp::Button, bksp, SetLabel(t_("<-")).LeftPosZ(132, 20).TopPosZ(32, 15))
 ITEM(Upp::Button, b7, SetLabel(t_("7")).LeftPosZ(12, 20).TopPosZ(52, 15))
 ITEM(Upp::Button, b8, SetLabel(t_("8")).LeftPosZ(40, 20).TopPosZ(52, 15))
 ITEM(Upp::Button, b9, SetLabel(t_("9")).LeftPosZ(68, 20).TopPosZ(52, 15))
 ITEM(Upp::Button, bdiv, SetLabel(t_("/")).LeftPosZ(96, 20).TopPosZ(52, 15))
 ITEM(Upp::Button, bsqrt, SetLabel(t_("sqrt")).LeftPosZ(124, 28).TopPosZ(52, 15))
 ITEM(Upp::Button, b4, SetLabel(t_("4")).LeftPosZ(12, 20).TopPosZ(72, 15))
 ITEM(Upp::Button, b5, SetLabel(t_("5")).LeftPosZ(40, 20).TopPosZ(72, 15))
 ITEM(Upp::Button, b6, SetLabel(t_("6")).LeftPosZ(68, 20).TopPosZ(72, 15))
 ITEM(Upp::Button, bmul, SetLabel(t_("*")).LeftPosZ(96, 20).TopPosZ(72, 15))
 ITEM(Upp::Button, bpct, SetLabel(t_("%")).LeftPosZ(124, 28).TopPosZ(72, 15))
 ITEM(Upp::Button, b1, SetLabel(t_("1")).LeftPosZ(12, 20).TopPosZ(92, 15))
 ITEM(Upp::Button, b2, SetLabel(t_("2")).LeftPosZ(40, 20).TopPosZ(92, 15))
 ITEM(Upp::Button, b3, SetLabel(t_("3")).LeftPosZ(68, 20).TopPosZ(92, 15))
 ITEM(Upp::Button, bminus, SetLabel(t_("-")).LeftPosZ(96, 20).TopPosZ(92, 15))
 ITEM(Upp::Button, b0, SetLabel(t_("0")).LeftPosZ(12, 20).TopPosZ(112, 15))
 ITEM(Upp::Button, bsep, SetLabel(t_(".")).LeftPosZ(40, 20).TopPosZ(112, 15))
 ITEM(Upp::Button, blessplus, SetLabel(t_("-/+")).LeftPosZ(68, 20).TopPosZ(112, 15))
 ITEM(Upp::Button, bplus, SetLabel(t_("+")).LeftPosZ(96, 20).TopPosZ(112, 15))
 ITEM(Upp::Button, beq, SetLabel(t_("=")).LeftPosZ(124, 28).TopPosZ(92, 36))
END_LAYOUT
```

File Attachments

1) calculator.png, downloaded 485 times



With this I'm getting resembling what I mean by "Dynamic Layout Scaling", still missing font scaling and is a bit hack.

Does Ultimate++ provide something like this or have anyone done something like this ?

- CalculatorDynLayout.h

```
#ifndef _CalculatorDynLayout_CalculatorDynLayout_h
#define _CalculatorDynLayout_CalculatorDynLayout_h

#include <CtrlLib/CtrlLib.h>

using namespace Upp;

#define LAYOUTFILE <CalculatorDynLayout/CalculatorDynLayout.lay>
#include <CtrlCore/lay.h>

class CalculatorDynLayout : public WithCalculatorDynLayoutLayout<TopWindow> {
 Rect initSize;
 Vector<Rect> childInitSizes;
public:
 CalculatorDynLayout();
 virtual void Layout();
};

#endif
```

- main.cpp

```
#include "CalculatorDynLayout.h"

CalculatorDynLayout::CalculatorDynLayout()
{
 CtrlLayout(*this, "Calculator");
}

void CalculatorDynLayout::Layout()
{
 if(childInitSizes.GetCount() == 0)
 {
  initSize = this->GetRect();
  for(int i=0, imax=this->GetChildCount(); i < imax; ++i)
  {
   childInitSizes.Add(this->GetIndexChild(i)->GetRect());
```

```
  }
 }
 else
 {
  Rect currSize = this->GetRect();
  double cxRatio = (currSize.right-currSize.left) / ((double)(initSize.right-initSize.left));
  double cyRatio = (currSize.bottom-currSize.top) / ((double)(initSize.bottom-initSize.top));
  entry.SetLabel(String() << initSize << "::" << currSize
   << "::" << cxRatio << "::" << cyRatio
   << "::" << childInitSizes.GetCount() << "::" << this->GetChildCount());

  for(int i=0, imax=this->GetChildCount(); i < imax; ++i)
  {
   Rect newSize, childSize;
   childSize = childInitSizes[i];
   Ctrl *childCtrl = this->GetIndexChild(i);
   newSize.left = childSize.left * cxRatio;
   newSize.top = childSize.top * cyRatio;
   newSize.right = childSize.right * cxRatio;
   newSize.bottom = childSize.bottom * cyRatio;
   childCtrl->SetRect(newSize);
  }
 }
}

GUI_APP_MAIN
{
 CalculatorDynLayout().Sizeable().Run();
}
```

- CalculatorDynLayout.lay

```
LAYOUT(CalculatorDynLayoutLayout, 160, 140)
 ITEM(Upp::Label, entry, SetFrame(BlackFrame()).LeftPosZ(12, 140).TopPosZ(8, 19))
 ITEM(Upp::Button, clear, SetLabel(t_("C")).LeftPosZ(68, 20).TopPosZ(36, 15))
 ITEM(Upp::Button, clearAll, SetLabel(t_("AC")).LeftPosZ(96, 20).TopPosZ(36, 15))
 ITEM(Upp::Button, bksp, SetLabel(t_("<-")).LeftPosZ(124, 28).TopPosZ(36, 15))
 ITEM(Upp::Button, b7, SetLabel(t_("7")).LeftPosZ(12, 20).TopPosZ(56, 15))
 ITEM(Upp::Button, b8, SetLabel(t_("8")).LeftPosZ(40, 20).TopPosZ(56, 15))
 ITEM(Upp::Button, b9, SetLabel(t_("9")).LeftPosZ(68, 20).TopPosZ(56, 15))
 ITEM(Upp::Button, bdiv, SetLabel(t_("/")).LeftPosZ(96, 20).TopPosZ(56, 15))
 ITEM(Upp::Button, bsqrt, SetLabel(t_("sqrt")).LeftPosZ(124, 28).TopPosZ(56, 15))
 ITEM(Upp::Button, b4, SetLabel(t_("4")).LeftPosZ(12, 20).TopPosZ(76, 15))
 ITEM(Upp::Button, b5, SetLabel(t_("5")).LeftPosZ(40, 20).TopPosZ(76, 15))
 ITEM(Upp::Button, b6, SetLabel(t_("6")).LeftPosZ(68, 20).TopPosZ(76, 15))
 ITEM(Upp::Button, bmul, SetLabel(t_("*")).LeftPosZ(96, 20).TopPosZ(76, 15))
 ITEM(Upp::Button, bpct, SetLabel(t_("%")).LeftPosZ(124, 28).TopPosZ(76, 15))
```

```
 ITEM(Upp::Button, b1, SetLabel(t_("1")).LeftPosZ(12, 20).TopPosZ(96, 15))
 ITEM(Upp::Button, b2, SetLabel(t_("2")).LeftPosZ(40, 20).TopPosZ(96, 15))
 ITEM(Upp::Button, b3, SetLabel(t_("3")).LeftPosZ(68, 20).TopPosZ(96, 15))
 ITEM(Upp::Button, bminus, SetLabel(t_("-")).LeftPosZ(96, 20).TopPosZ(96, 15))
 ITEM(Upp::Button, b0, SetLabel(t_("0")).LeftPosZ(12, 20).TopPosZ(116, 15))
 ITEM(Upp::Button, bsep, SetLabel(t_(".")).LeftPosZ(40, 20).TopPosZ(116, 15))
 ITEM(Upp::Button, blessplus, SetLabel(t_("-/+")).LeftPosZ(68, 20).TopPosZ(116, 15))
 ITEM(Upp::Button, bplus, SetLabel(t_("+")).LeftPosZ(96, 20).TopPosZ(116, 15))
 ITEM(Upp::Button, beq, SetLabel(t_("=")).LeftPosZ(124, 28).TopPosZ(96, 36))
END_LAYOUT
```