Subject: FEATURE_REQUEST: SliderCtrl to have WhenClick
Posted by nixnixnix on Tue, 20 Oct 2009 21:24:59 GMT
View Forum Message <> Reply to Message

Hi,

I would like the user to be able to move a slider very accurately using the cursor keys. I realise only one slider can use the WantFocus method in the toolbar and so I was looking to be able to switch focus depending on which slider the user had last clicked on.

Is there a reason why all Ctrls can't throw all the WhenX methods?

Also, it would be cool (for me at least) if the slider control could have a text control readout of its current position perhaps in place of either the top or bottom half of its handle.

Also, it would be useful for me if instead of a step size, I could program the slider to have a series of discrete values.

I realise these maybe should all be separate feature requests but they are all connected for me and I didn't want to spam the forum.

Cheers,

Nick

Subject: Re: FEATURE_REQUEST: SliderCtrl to have WhenClick
Posted by mrjt on Thu, 22 Oct 2009 13:58:51 GMT
View Forum Message <> Reply to Message

I can't comment on why Ctrl don't have more default actions, but everything you want can be done easily using a composite control:

```
class KeySlider : public Ctrl
{
 typedef KeySlider CLASSNAME;
private:
 struct ClickSlider : public SliderCtrl {
  Callback WhenClick;
  virtual void LeftUp(Point p, dword keyflags) {
   SliderCtrl::LeftUp(p, keyflags);
   WhenClick();
  }
 };
 ClickSlider ctrl;
 Label label;
 Vector<Value> keys;

 void UpdateSlider();
```

```
 void Click() const       { WhenClick(); }
 virtual void Updated()       { label.SetLabel(AsString(GetData())); }
public:
 Callback WhenClick;

 KeySlider();

 virtual void LeftUp(Point p, dword keyflags) { Click(); }

 KeySlider &Add(Value v) { keys.Add(v); UpdateSlider(); return *this; }
 void Remove(int i) { keys.Remove(i); UpdateSlider(); }
 int  GetCount() const;

 void SetLabelWidth(int cx);

 virtual Value GetData() const { return keys.GetCount() ? keys[(int)~ctrl] : Value(); }
 virtual void SetData(const Value& data);
};

KeySlider::KeySlider()
{
 Transparent(true);
 SetLabelWidth(32);
 UpdateSlider();
 ctrl.WhenClick = THISBACK(Click);
 ctrl <<= THISBACK(UpdateAction);
}

void KeySlider::SetLabelWidth(int cx)
{
 Ctrl::Add(ctrl.VSizePosZ().HSizePosZ(0, cx+2));
 Ctrl::Add(label.VSizePosZ().RightPosZ(0, cx));
}

void KeySlider::UpdateSlider()
{
 ctrl.Enable(keys.GetCount());
 if (!keys.GetCount())
  return;
 ctrl.MinMax(0, keys.GetCount()-1);
 if (IsNull(~ctrl))
  ctrl <<= 0;
 Updated();
}

void KeySlider::SetData(const Value& data)
{
 int ix = FindIndex(keys, data);
```

```
 if (ix >= 0)
  ctrl <<= ix;
}
```
Because it uses key values you can do things like:

slider.Add("A").Add("B").Add("C").Add("D").Add("E").Add("F").Add("G");

And when I want to add WhenClick or WhenKey events to Ctrls I usually just use a simple template that overloads the event handling function.

Subject: Re: FEATURE_REQUEST: SliderCtrl to have WhenClick
Posted by mirek on Thu, 22 Oct 2009 20:14:49 GMT
View Forum Message <> Reply to Message

I have looked into the issue and came to conclusion that SliderCtrl needs a call to SetWantFocus in LeftDown (I have also improved focus visuals in the process).

-> All you need to do now is to call WantFocus for your slider.

Mirek

Subject: Re: FEATURE_REQUEST: SliderCtrl to have WhenClick
Posted by nixnixnix on Wed, 11 Nov 2009 02:54:38 GMT
View Forum Message <> Reply to Message

Thanks James and Mirek,

I'll download the latest SVN and see how I get on.

Cheers,

Nick
