Constructing a Symbian container control

Overview

This code snippet shows how to create a container control that owns custom controls. The example shows how to focus the first component with a rectangle and how to change focus.

The example extends the existing code snippet How to create a custom control in Symbian C++.

Container controls store their controls into `CCoeControlArray`.

Header

```
class CMyContainerControl : public CCoeControl
{
public:
    static CMyContainerControl* NewL(const TRect& aRect);
    static CMyContainerControl* NewLC(const TRect& aRect);
    virtual ~CMyContainerControl();

private: // from CCoeControl
    void Draw(const TRect& aRect) const;
    void SizeChanged();

public: // own methods
    // NOTE: Transfer ownership to CMyContainerControl
    void AddControlL(CCoeControl* aControl,TInt aControlId);
    void UpdateControls();

private: // own methods
    CMyContainerControl();
    void ConstructL(const TRect& aRect);
};
```

Source

```
CMyContainerControl* CMyContainerControl::NewL(const TRect& aRect)
    {
    CMyContainerControl* self = CMyContainerControl::NewLC(aRect);
    CleanupStack::Pop(self);
    return self;
    }

CMyContainerControl* CMyContainerControl::NewLC(const TRect& aRect)
    {
    CMyContainerControl* self = new(ELeave) CMyContainerControl();
    CleanupStack::PushL(self);
    self->ConstructL(aRect);
    return self;
    }

CMyContainerControl::CMyContainerControl()
    {
    }

CMyContainerControl::~CMyContainerControl()
    {
    }

void CMyContainerControl::ConstructL(const TRect& aRect)
    {
    // No parent owner, so create an own window
    CreateWindowL();

    // Initialize component array
    InitComponentArrayL();

    SetRect(aRect);

    ActivateL();
    }

void CMyContainerControl::SizeChanged()
    {
    UpdateControls();
    }

void CMyContainerControl::UpdateControls()
    {
    TPoint position;

    // Goes through all components of this container control
    CCoeControlArray::TCursor cursor = Components().Begin();
    CCoeControl* ctrl = NULL;
    while ((ctrl = cursor.Control<CCoeControl>()) != NULL)
        {
        // If control is not visible, do not set it's position
        if (!ctrl->IsVisible())
            {
            cursor.Next();
            continue;
```


(C) Copyright Nokia 2013. All rights reserved.

```
            }

        // Set position
        ctrl->SetPosition(position);

        // Set size
        TSize size = ctrl->MinimumSize();
        size.SetSize(Rect().Width(),size.iHeight);
        ctrl->SetSize(size);

        // Calculate position
        position.iY += size.iHeight;

        // Does control fit to view?
        if (position.iY >= Rect().iBr.iY)
            {
            ctrl->MakeVisible(EFalse);
            }
        else
            {
            ctrl->MakeVisible(ETrue);
            }

        cursor.Next();
        }
    }

void CMyContainerControl::Draw(const TRect& /*aRect*/) const
    {
    CWindowGc& gc = SystemGc();
    gc.SetBrushColor(KRgbBlack);
    gc.Clear(Rect());
    }

void CMyContainerControl::AddControlL(CCoeControl* aControl,TInt aControlId)
    {
    // NOTE: Transfer ownership of CCoeControl to CMyContainerControl
    // Add control into container control
    Components().AppendLC(aControl,aControlId);
    CleanupStack::Pop(aControl);

    // Focus first component
    if (Components().Count()==1)
        {
        aControl->SetFocus(ETrue);
        }

    // Update control's position
    UpdateControls();
    }
```

SizeChanged

When `CMyContainerControl`'s size changes (when, for example, SetRect() is called), components' positions have to be calculated again.

```
void CMyContainerControl::SizeChanged()
    {
    // Sets new position of the components
    UpdateControls();
    }
```

Focusing component

Make the following changes to the `CMyControl` component.

- `CMyControl::Draw()` must call `DrawFocusFrame()` for drawing the focus rectangle
- `CMyControl::DrawFocusFrame()` is a new method where focus rectangle is drawn

```
void CMyControl::Draw(const TRect& aRect) const
    {
    CWindowGc& gc = SystemGc();
    gc.SetBrushColor(KRgbBlue);
    gc.Clear(Rect());

    DrawFocusFrame(aRect);
    }

void CMyControl::DrawFocusFrame(const TRect& aRect) const
    {
    // Nothing to draw if not focused
    if ( IsFocused() == EFalse )
        return;

    // Prep for draw
    CWindowGc& gc = SystemGc();
    gc.SetPenStyle( CGraphicsContext::ESolidPen );
    gc.SetPenSize( TSize(KFocusFrameWidth,KFocusFrameWidth) );
    gc.SetBrushStyle( CGraphicsContext::ENullBrush );
    gc.SetPenColor( KRgbDarkGray );

    // Draw the rounded rectangle
    gc.DrawRoundRect( aRect, TSize( KFrameRoundRadius, KFrameRoundRadius ) );
    }
```

Postconditions

`CMyContainerControl` container has some `CMyControl`custom controls in a list.

See also

(C) Copyright Nokia 2013. All rights reserved.

Custom Control Series:

- How to create a custom control in Symbian C++ Defining a custom control
- Constructing a Symbian custom control from a resource Creating a control from a resource
- Changing the focus of a Symbian custom control Handling key events and changing active custom control focus
- Archived:Using scrollbars in Symbian container control Adding scroll bar to custom control
- Using a Symbian custom control in a dialog Adding a custom control into CAknDialog
- File:CustomControl.zip Example code patch

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.