



# Symbian OS

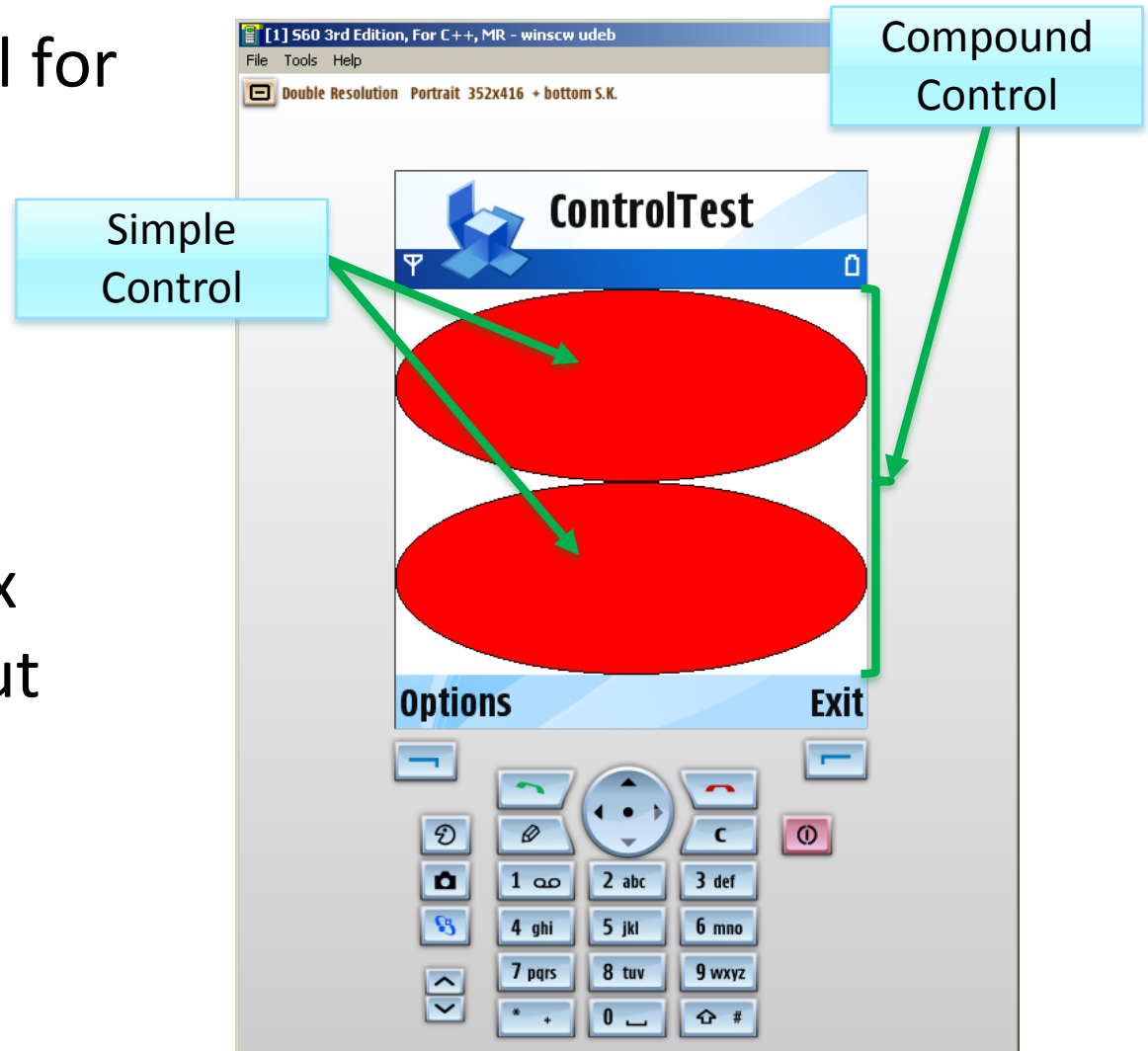
Challenge  
(Controls)

# Disclaimer

- These slides are provided free of charge at <http://www.symbianresources.com> and are used during Symbian OS courses at the University of Applied Sciences in Hagenberg, Austria ( <http://www.fh-hagenberg.at/> )
- Respecting the copyright laws, you are allowed to use them:
  - for your own, personal, non-commercial use
  - in the academic environment
- In all other cases (e.g. for commercial training), please contact [andreas.jakl@fh-hagenberg.at](mailto:andreas.jakl@fh-hagenberg.at)
- The correctness of the contents of these materials cannot be guaranteed. Andreas Jakl is not liable for incorrect information or damage that may arise from using the materials.
- Parts of these materials are based on information from Symbian Press-books published by John Wiley & Sons, Ltd. This document contains copyright materials which are proprietary to Symbian, UIQ, Nokia and SonyEricsson. “S60™” is a trademark of Nokia. “UIQ™” is a trademark of UIQ Technology. Pictures of mobile phones or applications are copyright their respective manufacturers / developers. “Symbian™”, “Symbian OS™” and all other Symbian-based marks and logos are trademarks of Symbian Software Limited and are used under license. © Symbian Software Limited 2006.

# The final application

- Develop an own control for drawing a circle
- Compound Control (= Container) should manage both controls
- **Code-Template:** S60 3.x GUI Application (without UI-Designer!)



# Strategy

- **Your circle-control:**

- derive it from CCoeControl
  - void ConstructL (const TRect& aRect, const CCoeControl\* aParent)
  - void Draw (const TRect &aRect) **const**
- Should use window from the compound control
- SDK-Doc (see below): “Creating a control”, “Window owning or not?”, “Drawing graphics”

- **Compound Control:**

- Extend the Container (ControlTestAppView.cpp) created by Carbide-wizard
- Calculate size of the circle-controls: TRect curRect = Rect();
- SDK-Doc (see below): “Creating a compound control”

- **SDK-Doc:** Symbian OS v9.1 → Symbian OS guide → Application framework → Using UI Control Framework (CONE) → How to Write Controls

# Circle Control

- Header file:

```
#include <coecntrl.h> // CCoeControl

class CCircleControl : public CCoeControl
{
public:
    CCircleControl();
    ~CCircleControl();
    void ConstructL(const TRect& aRect, const CCoeControl* aParent);
public:
    void Draw(const TRect &aRect) const;
};
```

# Circle Control

- Source file:

```
void CCircleControl::ConstructL(const TRect& aRect, const CCoeControl* aParent) {
    if (aParent) {                // Control has a parent ...
        SetContainerWindowL(*aParent); // ... reuse parents window
    } else {                      // Control should be window-owning ...
        CreateWindowL();         // ... create an own window
    }
    SetRect(aRect);              // Assign size to this control
    ActivateL();                 // Control-initialisation finished, ready to draw
}

void CCircleControl::Draw(const TRect &aRect) const {
    // We don't want to do partial redraws – area specified by aRect is ignored
    CWindowGc& gc = SystemGc(); // Get Graphics Context to draw contents
    DrawBackground(Rect());     // Let framework draw background
    // Set GC pen & brush styles
    gc.SetPenStyle(CGraphicsContext::ESolidPen);
    gc.SetPenColor(KRgbBlack);
    gc.SetBrushStyle(CGraphicsContext::ESolidBrush);
    gc.SetBrushColor(KRgbRed);
    gc.DrawEllipse(Rect());      // Draw an ellipse using the size of the control
}
```

# Container

- CControlTestAppView::ConstructL(), before ActivateL()

```
// Initialize the component array  
InitComponentArrayL();
```

New in  
Symbian OS 9

```
// Construct components
```

```
TRect curRect = Rect();
```

```
// Size of drawing area
```

```
curRect.SetHeight(Rect().Height() / 2);
```

```
// Reduce to half of the original height
```

```
// Create the first control
```

```
CCircleControl* myControl = new (ELeave) CCircleControl();
```

```
// Append the control to the component array
```

```
Components().AppendLC(myControl);
```

```
// Places control on Cleanup Stack, for further initialization
```

```
myControl->ConstructL(curRect, this);
```

```
// Do 2nd phase construction of control
```

```
CleanupStack::Pop(myControl);
```

```
// Construction finished, Cleanup Stack not needed anymore
```

```
// Reuse rectangle, but move it down
```

```
curRect.Move(TPoint(0, curRect.Height()));
```

```
// Create 2nd control (same way as 1st)
```

```
CCircleControl* myControl2 = new (ELeave) CCircleControl();
```

```
Components().AppendLC(myControl2);
```

```
myControl2->ConstructL(curRect, this);
```

```
CleanupStack::Pop(myControl2);
```