Using screen with S60 and Maemo Platforms

Introduction

Both S60 and Maemo Platform have a large set of user interface components that the application can use to create its look.

Hildon reference

Hildon widgets gallery

The S60 or Maemo application can also implement the interface look fully.

Comparing S60 and Maemo Platforms

S60 Platform

The Draw function is called by window server. It is used for both window server initiated redrawing of the control and for some application-initiated drawing. All controls, except blank controls, should implement this function. The default implementation draws a blank control.

Draw function should be implemented by each derived control. The function is only called from within CCoeControl's member functions, and not from the derived class. For this reason it is a private member function of CCoeControl.

```
// -----------------------------------------------------------
// CBrCtlSampleAppContainer::Draw(const TRect& aRect) const
// -----------------------------------------------------------
//
void CBrCtlSampleAppContainer::Draw(const TRect& aRect) const
    {

    CWindowGc& gc = SystemGc();
    gc.SetPenStyle(CGraphicsContext::ESolidPen);
    gc.SetBrushColor(KRgbGray);
    gc.SetBrushStyle(CGraphicsContext::ESolidBrush);
    gc.DrawRect(aRect);

    gc.UseFont(iFont);

    gc.DrawText(iText, iPoint);

    gc.DiscardFont();
    }
```

How to make a full-screen application using Symbian C++

Direct Screen Access

Maemo Platform

A drawing_area widget allows a complete control for drawing the contents. If our window is obscured then uncovered, we get an exposure event and must redraw what was previously hidden. A drawing_area widget is essentially an X window and nothing more. It is a blank canvas in which we can draw whatever we like.

```
/* Create the drawing area */
void create_drawingarea( MainView *main )
{
    /* Scrolled window */
    main->scrolled_window = gtk_scrolled_window_new(NULL, NULL);
    gtk_widget_show(main->scrolled_window);
    gtk_scrolled_window_set_policy(GTK_SCROLLED_WINDOW(main->scrolled_window),
                                   GTK_POLICY_AUTOMATIC,
                                   GTK_POLICY_AUTOMATIC);

    /* Drawing area */
    main->drawing_area = gtk_drawing_area_new ();
    gtk_widget_set_size_request (main->drawing_area, 100, 100);

    /* Put drawingarea under scrolledwindow and show it */
    gtk_container_add(GTK_CONTAINER(main->scrolled_window), main->drawing_area);
    gtk_widget_show(main->drawing_area);

    /* Connect signals */
    g_signal_connect (G_OBJECT (main->drawing_area), "expose_event",
                      G_CALLBACK (callback_redraw), main);
}
```

```
/* redraw */
void callback_redraw (GtkWidget *widget, GdkEventExpose *event, gpointer data)
{
    GdkFont* font = NULL;
    GdkGC* gc = NULL;

    ...

    gc = widget->style->fg_gc[GTK_WIDGET_STATE (widget)];
```


(C) Copyright Nokia 2013. All rights reserved.

```
    font = gdk_font_from_description(
               pango_font_description_from_string ("Monospace Regular 22"));

    ...

    gdk_draw_string(widget->window,font,gc,10,30, "No text file");

    ...

}
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.