Archived:Displaying sensor data on Symbian

Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Overview

This code snippet demonstrates how to display the data produced by the accelerometer and rotation sensors of the device on the screen.

Registering listeners for the sensors is not covered in this code snippet. See Archived:Listening for accelerometer sensor data events on Symbian and Archived:Listening for rotation sensor data events on Symbian for more information on registering listeners.

Note: In order to use the code, you need to install the sensor plug-in for your SDK.

This snippet can be self-signed.

MMP file

The following libraries are required:

```
LIBRARY RRSensorApi.lib
LIBRARY gdi.lib
```

Header file

```
#include <coecntrl.h>   // CCoeControl
#include <gdi.h>        // CFont
#include <RRSensorApi.h>
```

```
// Accelerometer sensor UID
const TInt KAccelerometerSensorUID = 0x10273024;
// Rotation sensor UID
const TInt KRotationSensorUID = 0x303E;
```

```
class CMyControl : public CCoeControl, public MRRSensorDataListener
    {
    // ...

    private:  // Functions from base classes
        /**
         * From CCoeControl.
         * Draws this control to the screen.
         * @param aRect the rectangle of this view that needs updating
         */
        void Draw(const TRect& aRect) const;

        /**
         * From MRRSensorDataListener.
```


(C) Copyright Nokia 2014. All rights reserved.

```
         * Callback function for receiving sensor data events.
         *
         * @param aSensor identifies the sensor that created the event.
         * @param aEvent contains data about created event.
         */
        void HandleDataEventL(TRRSensorInfo aSensor, TRRSensorEvent aEvent);

    private:  // Data
        CRRSensorApi* iAccelerometerSensor;
        TInt iAccSensorDataX;
        TInt iAccSensorDataY;
        TInt iAccSensorDataZ;
        CRRSensorApi* iRotationSensor;
        TInt iRotSensorDataX;
        TInt iRotSensorDataY;
        TInt iRotSensorDataZ;
        CFont* iFont;
    }
```

Source file

```
#include <EIKENV.H> // CEikonEnv
#include <gdi.h>    // TFontSpec, CGraphicsDevice
#include <RRSensorApi.h>
```

```
void CMyControl::ConstructL(const TRect& aRect)
    {
    // ...

    // Reset sensor data
    iAccSensorDataX = 0;
    iAccSensorDataY = 0;
    iAccSensorDataZ = 0;
    iRotSensorDataX = 0;
    iRotSensorDataY = 0;
    iRotSensorDataZ = 0;

    // Initialize the font used in drawing onto the screen
    _LIT(KFontName, "Arial");
    const TInt KFontSize = 120;
    TFontSpec fontSpec(KFontName, KFontSize);
    CGraphicsDevice* screenDevice = CEikonEnv::Static()->ScreenDevice();
    screenDevice->GetNearestFontInTwips(iFont, fontSpec);

    // ...
    }
```

```
/**
 * Draws the display.
 */
void CMyControl::Draw(const TRect& /*aRect*/) const
    {
    // Get the standard graphics context
```


(C) Copyright Nokia 2014. All rights reserved.

```
    CWindowGc& gc = SystemGc();

    // Get the control's extent
    TRect drawRect(Rect());

    // Clear the screen with white color
    gc.SetBrushColor(KRgbWhite);
    gc.Clear(drawRect);

    // Set the font
    gc.UseFont(iFont);
    // The text is drawn in black
    gc.SetBrushColor(KRgbBlack);

    TBuf<255> buffer;

    // Draw accelerometer sensor data
    _LIT(KAccTxt, "Accel. sensor: X: %d, Y: %d, Z: %d");
    buffer.Format(KAccTxt, iAccSensorDataX, iAccSensorDataY, iAccSensorDataZ);
    gc.DrawText(buffer, TPoint(0, 20));

    // Draw rotation sensor data
    _LIT(KRotTxt, "Rotation sensor: X: %d, Y: %d, Z: %d");
    buffer.Format(KRotTxt, iRotSensorDataX, iRotSensorDataY, iRotSensorDataZ);
    gc.DrawText(buffer, TPoint(0, 40));
    }
```

```
/**
 * Gets called whenever a data event occurs.
 */
void CMyControl::HandleDataEventL(TRRSensorInfo aSensor, TRRSensorEvent aEvent)
    {
    // A data event occurred. Store new values from the sensor in question.
    switch (aSensor.iSensorId)
        {
        case KAccelerometerSensorUID:
            {
            iAccSensorDataX = aEvent.iSensorData1;
            iAccSensorDataY = aEvent.iSensorData2;
            iAccSensorDataZ = aEvent.iSensorData3;
            }
            break;
        case KRotationSensorUID:
            {
            iRotSensorDataX = aEvent.iSensorData1;
            iRotSensorDataY = aEvent.iSensorData2;
            iRotSensorDataZ = aEvent.iSensorData3;
            }
        default:
            break;
        }
    // Update the display
    DrawNow();
    }
```


(C) Copyright Nokia 2014. All rights reserved.

Postconditions

Data received from accelerometer and rotation sensors is constantly drawn onto the screen.

See also

- Archived:Listening for accelerometer sensor data events on Symbian
- Archived:Listening for rotation sensor data events on Symbian

(C) Copyright Nokia 2014. All rights reserved.