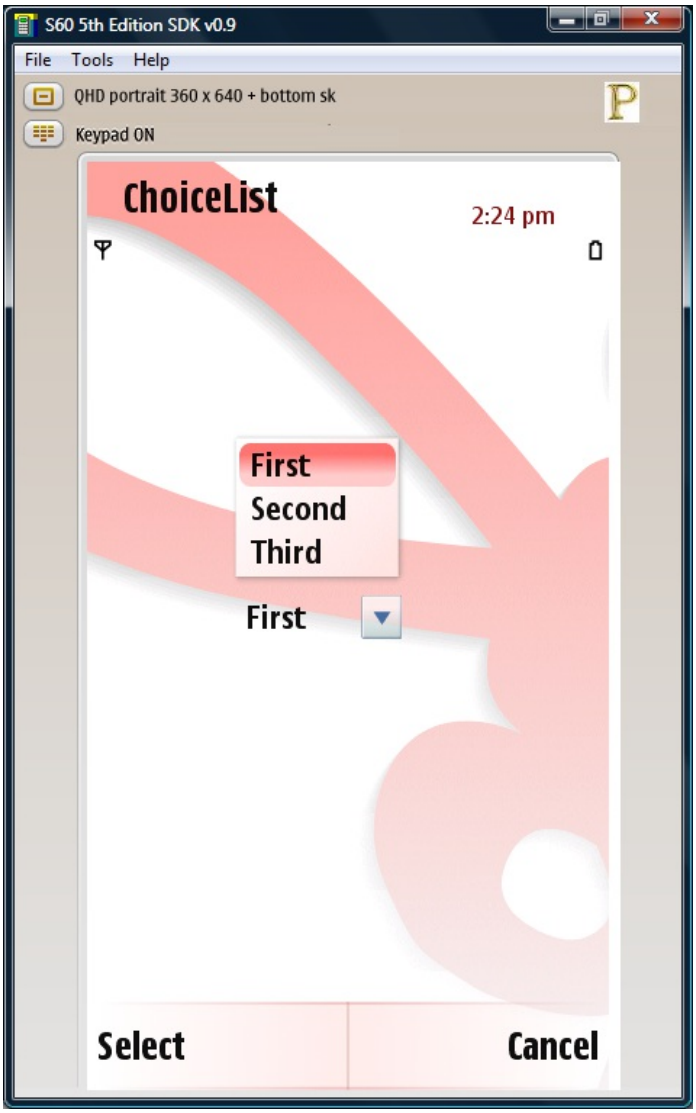
Working with ChoiceList API - S60 Touch UI

Overview

ChoiceList API is a new component introduced in S60 5th edition Touch UI concept. In this article we will learn how to play with ChoiceList API with different methods.

What is ChoiceList API

- ChoiceList API is a new Touch UI component available with S60 5th Edition platform.
- In the construction of ChoiceList, you can define it with "default selection enabled" or "without selection enabled" for the items in the list.
- You can use Buttons as items on ChoiceList.
- You can fill the items of ChoiceList with array elements.
- ChoiceList API provides various function to manipulate items residing on ChoiceList.



Implementing ChoiceList component

ChoiceListAppView.h

- We will inherit our class from `MCoeControlObserver` interface and implement its pure virtual method `HandleControlEventL()` to receive and handle events for ChoiceList.
- Declare object of `CAknChoiceList` and use it to set observer for this class and handle events on ChoiceList.

```
...
...
#include <aknchoicelist.h>

// CLASS DECLARATION
```


(C) Copyright Nokia 2013. All rights reserved.

```
class CChoiceListAppView : public CCoeControl, public MCoeControlObserver
 {
  .....
  .....
  //From MCoeControlObserver
  void HandleControlEventL(CCoeControl *aControl,TCoeEvent aEventType);

  //Application specific helper function
  void CreateChoiceListL();

  //From CCoeControl
  TInt CountComponentControls() const;
  CCoeControl* ComponentControl(TInt aIndex) const;
  .....
 private:
  CAknChoiceList* iChoiceList;
  ....
 };
```

ChoiceListAppView.cpp

- Call to create choice list function from `ConstructL()` as shown below.

```
// ----------------------------------------------------------------------------
// CChoiceListAppView::ConstructL()
// Symbian 2nd phase constructor can leave.
// ----------------------------------------------------------------------------
//
void CChoiceListAppView::ConstructL(const TRect& aRect)
 {
 // Create a window for this application view
 CreateWindowL();

 CreateChoiceListL(); //We will create our choice list in this function
 ....
 ....

 // Set the windows size
 SetRect(aRect);

 // Activate the window, which makes it ready to be drawn
 ActivateL();
 }
```

- Create choice list and fill the default items from array.
- Set "this" as an observer for ChoiceList and eventually we will receive callbacks in `HandleControlEventL()`

```
void CChoiceListAppView::CreateChoiceListL()
 {
 CDesCArray* itemArray = new (ELeave) CDesCArrayFlat(5);
 CleanupStack::PushL(itemArray);
 itemArray->AppendL(_L("First"));
 itemArray->AppendL(_L("Second"));
 itemArray->AppendL(_L("Third"));

 iChoiceList = CAknChoiceList::NewL(this, itemArray);
 CleanupStack::Pop(itemArray);
 iChoiceList->SetContainerWindowL(*this);
 iChoiceList->SetObserver(this);
 iChoiceList->SetRect(TRect ( TPoint(100,200), TSize(120,10) ) );
 }
```

- We will handle callback events on choice list control as shown below in the `HandleControlEventL`

```
void CChoiceListAppView::HandleControlEventL(CCoeControl *aControl,TCoeEvent aEventType)
 {
 if(iChoiceList == aControl)
  switch(aEventType)
  {
   case EEventStateChanged:
   {
    TInt index = iChoiceList->SelectedIndex();
    _LIT(KMenuItem,"%d is selected");
    TBuf<20> buf;
    buf.Format(KMenuItem(), index+1);
    CAknInformationNote* info = new (ELeave) CAknInformationNote;
    info->ExecuteLD(buf);
   }
   break;
  }
 }
```

- Definition for the following functions to return ChoiceList control.

```
TInt CChoiceListAppView::CountComponentControls() const
{
 return 1; // return number of controls inside this container
}

CCoeControl* CChoiceListAppView::ComponentControl(TInt aIndex) const
{
    switch ( aIndex )
    {
        case 0:
            return iChoiceList;

        default:
            return NULL;
    }
}
```

- Make sure to delete **iChoiceList** in the destructor of the class.

```
// ----------------------------------------------------------------------------
// CChoiceListAppView::~CChoiceListAppView()
```


(C) Copyright Nokia 2013. All rights reserved.

```
// Destructor.
// ---------------------------------------------------------------------------
//
CChoiceListAppView::~CChoiceListAppView()
{
 ....
 ....
 if(iChoiceList)
 {
  delete iChoiceList;
  iChoiceList = NULL;
 }
}
```

Useful functions

CAknChoiceList

- SelectedIndex()
- AddItemL()
- RemoveItem()
- HideChoiceList()
- InsertItemL()

MCoeControlObserver

- HandleControlEventL()

Keywords

Headers

- #include <aknchoicelist.h>

Classes

- CAknChoiceList
- MCoeControlObserver

Libraries

- eikcoctl.lib

Example Application

- ChoiceList.zip

Related links

- A tour to the S60 Touch UI components
- Working with Toolbar API - S60 Touch UI
- Working with LongTapDetector API - S60 Touch UI
- Working with Stylus Pop-up Menu API - S60 Touch UI
- Working with Tactile Feedback Client API - S60 Touch UI
- Working with Adaptive Search feature - S60 Touch UI
- Working with Generic Button API - S60 Touch UI
- Working with SingleStyleTreeList with Hierarchical Lists API - S60 Touch UI
- Working with SingleColumnStyleTreeList with Hierarchical Lists API - S60 Touch UI

Reference list

- S60 5th edition SDK help
- S60 5th Edition C++ Developer's Library v1.0

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.