

Grids - Exemplo

Grids são uma forma bi-dimensional de ListBox para representar dados (gráficos ou texto), grids são derivados da classe CAknGrid e podem ser definidos através de um RESOURCE ou diretamente em sua construção.

A disposição do Grid pode ser feita através do resource GRID_STYLE, e seus itens são compostos de um array de TBufs contendo os itens do Grid.

É preciso fazer algumas alterações no endereço do multi-bmp do resource, na função AjustarIcones e definir corretamente as enumerações.

GridExemplo.rss

```
RESOURCE GRID r_grid_exemplo
{
    flags = EAknListBoxSelectionGrid;
    style = r_grid_exemplo_style;
    array_id = r_grid_exemplo_items;
}

//Resource que contém a definição do Grid
RESOURCE GRID_STYLE r_grid_exemplo_style
{
    layoutflags = EAknGridHorizontalOrientation | EAknGridLeftToRight |
                  EAknGridTopToBottom;
    primaryscroll = EAknGridFollowsItemsAndLoops;
    secondaryscroll = EAknGridFollowsItemsAndLoops;
    itemsinprimaryorient = 6;
    itemsinsecondaryorient = 1;
    width = 21;
    height = 32;
    gapwidth = 2;
    gapheight = 2;
}

//Resource ARRAY que contém os itens do GRID
RESOURCE ARRAY r_grid_exemplo_items
{
    items = {
        LBUF { txt = "0\t"; },
        LBUF { txt = "1\t"; },
        LBUF { txt = "2\t"; },
        LBUF { txt = "3\t"; },
        LBUF { txt = "4\t"; },
        LBUF { txt = "5\t"; }
        //LBUF { txt = "6\t"; }
    };
}
//Diretório onde ser encontra o multi-bitmap com os ícones do grid
RESOURCE TBUF r_grid_exemplo_icones
{
    buf = "\\system\\apps\\nome-do- arquivo.mbm";
}
```

Grid.h

```
#include <akngrid.h>

class CGrid : public CAknGrid
{
public:
    void ConstructL(TInt aGridResource, TInt aIconResource);
public:
    void SizeChanged();
private:
    void AjustarIcones(TInt aIconResource);
    void AjustarGrid();

};

}
```

Grid.cpp

```
#include "Grid.h"
#include <akniconarray.h> // CAknIconArray
#include <aknlists.h> // AknListBoxLayouts
#include <barsread.h> // TResourceReader
#include <cartasgrid.mbg> // contains icon enumeration
#include <stringloader.h> // StringLoader
#include <akngrid.h>

void CGrid::ConstructL(TInt aGridResource, TInt aIconResource)
{
    TResourceReader reader;
    CEikonEnv::Static()->CreateResourceReaderLC(reader, aGridResource);
    ConstructFromResourceL(reader);
    CleanupStack::PopAndDestroy(); //reader
    AjustarIcones(aIconResource);
    SizeChanged();
}

//Criação dos ícones
void CGrid::AjustarIcones(TInt aIconResource)
{
    HBufC* iconFileName;
    iconFileName = StringLoader::LoadLC(aIconResource);

    CArrayPtr<CGuiIcon>* icons = new(ELeave) CAknIconArray(6);
    CleanupStack::PushL(icons);
    icons->AppendL(iEikonEnv->CreateIconL(*iconFileName,
        EMbmCartasgrid2copas, EMbmCartasgridCarta_mask));
    icons->AppendL(iEikonEnv->CreateIconL(*iconFileName,
        EMbmCartasgrid2espada, EMbmCartasgridCarta_mask));
    icons->AppendL(iEikonEnv->CreateIconL(*iconFileName,
        EMbmCartasgrid2ouro, EMbmCartasgridCarta_mask));
    icons->AppendL(iEikonEnv->CreateIconL(*iconFileName,
        EMbmCartasgrid2paus, EMbmCartasgridCarta_mask));
    icons->AppendL(iEikonEnv->CreateIconL(*iconFileName,
        EMbmCartasgrid9copas, EMbmCartasgridCarta_mask));
    icons->AppendL(iEikonEnv->CreateIconL(*iconFileName,
        EMbmCartasgrid10copas, EMbmCartasgridCarta_mask));
}
```

```
//icons->AppendL(iEikonEnv->CreateIconL(*iconFileName,
    EMbmCartasgridAsespada, EMbmCartasgridCarta_mask));
CleanupStack::Pop(icons);
CleanupStack::PopAndDestroy(iconFileName);

ItemDrawer()->FormattedCellData()->SetIconArray(icons);
}

//Definindo o layout de cada célula
void CGrid::AjustarGrid()
{
    AknListBoxLayouts::SetupStandardGrid(*this);

    CFormattedCellListBoxItemDrawer* itemDrawer = this->ItemDrawer();
    TInt cellWidth = ColumnWidth();
    TInt cellHeight = ItemHeight();

    AknListBoxLayouts::SetupFormGfxCell(*this, //the grid
        itemDrawer, // the grid's drawer
        0, // index of the graphic within item strings
        0, // left position
        0, // top position
        0, // right - unused
        0, // bottom - unused
        23, // width of graphic
        35, // height of graphic
        TPoint (0, 0), // start position
        TPoint (32 , 32)); // end position

    const CFont* KFont = LatinBold12();

    AknListBoxLayouts::SetupFormTextCell(*this, // the grid
        itemDrawer, // the grid's drawer
        1, // index of text within item strings
        KFont, // the font for the text
        215, // the color of the text
        0, // left margin
        0, // right margin - unused
        5, // Baseline
        32, // text width
        CGraphicsContext::ECenter, // Alignment
        TPoint (0, 32), // start position
        TPoint(32, 32)); // end position*

}

void CGrid::SizeChanged()
{
    CAknGrid::SizeChanged();
    AjustarGrid();
}
```

GridContainer.h

```
#include <coecntrl.h>
#include <eiklbo.h>

class CGrid;

class CGridContainer : public CCoeControl, MEikListBoxObserver
{
public: // Constructors and destructor
    static CGridContainer* NewL(const TRect& aRect);
    static CGridContainer* NewLC(const TRect& aRect);
    virtual ~CGridContainer();
    void FuncaoEspecifica();
private: // classe base
    void Draw(const TRect& aRect) const;
    void SizeChanged();
    TInt CountComponentControls() const;
    CCoeControl* ComponentControl(TInt aIndex) const;
    TKeyResponse OfferKeyEventL(const KeyEvent&
        aKeyEvent, TEventCode aType);

private:
    void HandleListBoxEventL(CEikListBox* aListBox,
        TListBoxEvent aListBoxEvent);

public:
    void ConstructL(const TRect& aRect);
private:
    CGrid* iGrid;
};

}
```

GridContainer.cpp

```
#include "GridContainer.h"
#include "Grid.h"
#include <eiklbx.h>
#include <GridExemplo_0xE14CA489.RSG>
#include <badesca.h>

void CGridContainer::ConstructL(const TRect& aRect)
{
    CreateWindowL();
    iGrid = new (ELeave) CGrid;
    iGrid->SetContainerWindowL(*this);
    iGrid->ConstructL(R_GRID_EXEMPLO, R_GRID_EXEMPLO_ICONES);
    iGrid->SetListBoxObserver(this);
    SetRect(aRect);
    ActivateL();
}
CGridContainer* CGridContainer::NewL(const TRect& aRect)
{
    CGridContainer* self = CGridContainer::NewLC(aRect);
    CleanupStack::Pop(self);
    return self;
}
```

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

CGridContainer::~CGridContainer()
{
    if(iGrid)
    {
        delete iGrid;
        iGrid = NULL;
    }
}

void CGridContainer::SizeChanged()
{
    iGrid->SetExtent(TPoint(50,80),iGrid->MinimumSize());
}

TInt CGridContainer::CountComponentControls() const
{
    return 1;
}

CCoeControl* CGridContainer::ComponentControl(TInt aIndex) const
{
    switch (aIndex)
    {
        case 0:
            return iGrid;
        default:
            return NULL;
    }
}

void CGridContainer::Draw(const TRect& aRect) const
{
    CWindowGc& gc = SystemGc();
    gc.Clear(aRect);
}

TKeyResponse CGridContainer::OfferKeyEventL( const TKeyEvent& aKeyEvent,
TEventCode aType)
{
    if (aType != EEventKey)
        return EKeyWasNotConsumed;

    if (iGrid)
        return iGrid->OfferKeyEventL (aKeyEvent, aType);
    else
        return EKeyWasNotConsumed;
}

void CGridContainer::FuncaoEspecificada()
```

```
{  
    // quantidade de itens no grid  
TInt itens = iGrid->GridModel()->NumberOfData();  
  
    // Index do item atual  
TInt itemAtual = iGrid->CurrentDataIndex();  
  
}  
void CGridContainer::HandleListBoxEventL( CEikListBox* /*aListBox*/,  
TListBoxEvent aListBoxEvent)  
{  
if ((aListBoxEvent == MEikListBoxObserver::EEventEnterKeyPressed) ||  
(aListBoxEvent == MEikListBoxObserver::EEventItemClicked))  
{  
    FuncaoEspecifica();  
}  
}
```

GridExemploAppUi.h

```
#include <aknappui.h>  
  
class CGridContainer;  
  
class CGridExemploAppUi : public CAknAppUi  
{  
public:  
    void ConstructL();  
    CGridExemploAppUi();  
    virtual ~CGridExemploAppUi();  
private:  
    void HandleCommandL( TInt aCommand );  
    void HandleStatusPaneSizeChange();  
    CArrayFix<TCoeHelpContext>* HelpContextL() const;  
private:  
    CGridContainer* iAppContainer;  
  
};
```

GridExemploAppUi.cpp

```
#include <avkon.hrh>  
#include <aknmessagequerydialog.h>  
#include <aknnotewrappers.h>  
#include <stringloader.h>  
#include <f32file.h>  
#include <s32file.h>  
#include <hlplch.h>  
  
#include <GridExemplo_0xE14CA489.rsg>  
#include <eiklbx.h>  
#include "GridExemplo_0xE14CA489.hlp.hrh"
```

```
#include "GridExemplo.hrh"
#include "GridExemplo.pan"
#include "GridExemploApplication.h"
#include "GridExemploAppUi.h"
#include "GridExemploAppView.h"
#include "GridContainer.h"
#include <GridExemplo_0xE14CA489.RSG>

_LIT( KFileName, "C:\\private\\E14CA489\\GridExemplo.txt" );
_LIT( KText, "Hello World!" );

void CGridExemploAppUi::ConstructL()
{
BaseConstructL(CAkNAppUi::EAknEnableSkin);

iAppContainer = new (ELeave) CGridContainer();
iAppContainer->SetMopParent(this);
iAppContainer->ConstructL(ClientRect());
AddToStackL(iAppContainer);

RFs fsSession;
User::LeaveIfError(fsSession.Connect());
CleanupClosePushL( fsSession );

TInt err = fsSession.MkDirAll(KFileName);
if ( (KErrNone != err) && (KErrAlreadyExists != err) )
{
CleanupStack::PopAndDestroy(1); // fsSession
return;
}

RFile file;
err = file.Replace(fsSession, KFileName, EFileWrite );
CleanupClosePushL( file );
if ( KErrNone != err )
{
CleanupStack::PopAndDestroy(2); // file, fsSession
return;
}

RFileWriteStream outputFileStream( file );
CleanupClosePushL( outputFileStream );
outputFileStream << KText;

CleanupStack::PopAndDestroy(3); // outputFileStream, file, fsSession
}

CGridExemploAppUi::CGridExemploAppUi()
{
}
CGridExemploAppUi::~CGridExemploAppUi()
{
if ( iAppContainer )
{
RemoveFromStack(iAppContainer);
delete iAppContainer;
iAppContainer = NULL;
}
```

```
}

}

void CGridExemploAppUi::HandleCommandL( TInt aCommand )
{
switch( aCommand )
{
case EEikCmdExit:
case EAknSoftkeyExit:
    Exit();
    break;

case ECommand1:
{
    HBufC* textResource = StringLoader::LoadLC(
        R_COMMAND1_TEXT );
    CAknInformationNote* informationNote;

    informationNote = new ( ELeave ) CAknInformationNote;
    informationNote->ExecuteLD( *textResource );

    CleanupStack::PopAndDestroy( textResource );
}
break;
case ECommand2:
{
    RFs fsSession;
    RFile rFile;

    User::LeaveIfError(fsSession.Connect());
    CleanupClosePushL(fsSession);

    User::LeaveIfError( rFile.Open( fsSession, KFileName,
EFileStreamText)); //EFileStreamText));
    CleanupClosePushL(rFile);

    RFileReadStream inputFileStream(rFile);
    CleanupClosePushL(inputFileStream);

    HBufC* fileData = HBufC::NewLC(inputFileStream, 32);

    CAknInformationNote* informationNote;

    informationNote = new ( ELeave ) CAknInformationNote;
    informationNote->ExecuteLD( *fileData);

    CleanupStack::PopAndDestroy(4); // filedatal,
inputFileStream, rFile, fsSession
    fsSession.Close();
}
break;
case EHelp:
{
```

```
CArrayFix<TCoeHelpContext>* buf =
    CCoeAppUi::AppHelpContextL();
HlpLauncher::LaunchHelpApplicationL(
    iEikonEnv->WsSession(), buf);
}
break;
case EAAbout:
{
CAknMessageQueryDialog* dlg = new
    (ELeave)CAknMessageQueryDialog();
dlg->PrepareLC(R_ABOUT_QUERY_DIALOG);
HBufC* title = iEikonEnv->AllocReadResourceLC(
    R_ABOUT_DIALOG_TITLE);
dlg->QueryHeading()->SetTextL(*title);
CleanupStack::PopAndDestroy(); //title
HBufC* msg = iEikonEnv->AllocReadResourceLC(
    R_ABOUT_DIALOG_TEXT);
dlg->SetMessageTextL(*msg);
CleanupStack::PopAndDestroy(); //msg
dlg->RunLD();
}
break;
default:
Panic( EGridExemploUi );
break;
}
}
void CGridExemploAppUi::HandleStatusPaneSizeChange()
{
iAppContainer->SetRect( ClientRect() );
}

CArrayFix<TCoeHelpContext>* CGridExemploAppUi::HelpContextL() const
{
#warning "Please see comment about help and UID3..."
CArrayFixFlat<TCoeHelpContext>* array = new (ELeave) CArrayFixFlat
    <TCoeHelpContext> (1);
CleanupStack::PushL(array);
array->AppendL( TCoeHelpContext( KUidGridExemploApp,
    KGeneral_Information));
CleanupStack::Pop(array);
return array;
}
```

