CBitmapRotator

Overview

`CBitmapRotator` helps when have to rotate the bitmaps.

MMP file

Following libraries are needed

```
LIBRARY bitmaptransforms.lib
```

Header

```
#include <e32base.h>
#include <BitmapTransforms.h>

class CBitmapRotator;

class MBitmapRotatorObserver
    {
    public:
        virtual void RotateComplete( TInt aError ) = 0;
    };

class CBitmapRotatorHandler : public CActive
    {
    public:
        static CBitmapRotatorHandler* NewL(MBitmapRotatorObserver& aObserver);
        virtual ~CBitmapRotatorHandler();

    private:
        CBitmapRotatorHandler(MBitmapRotatorObserver& aObserver);
        void ConstructL();

    private: // From CActive
        void DoCancel();
        void RunL();
        TInt RunError(TInt aError);

    public: // New functions
        // Rotates a bitmap by a specified angle
        void Rotate(CFbsBitmap& aBitmap,
        CBitmapRotator::TRotationAngle aAngle);

    private: // Data
        MBitmapRotatorObserver& iObserver;
        CBitmapRotator* iBitmapRotator;
    };
```

Source

```
#include "CBitmapRotatorHandler.h"

CBitmapRotatorHandler* CBitmapRotatorHandler::NewL(MBitmapRotatorObserver& aObserver)
    {
    CBitmapRotatorHandler* self = new (ELeave) CBitmapRotatorHandler( aObserver );
    CleanupStack::PushL( self );
    self->ConstructL();
    CleanupStack::Pop();
    return self;
    }

CBitmapRotatorHandler::~CBitmapRotatorHandler()
    {
    if (IsActive())
        {
        Cancel();
        }

    if (iBitmapRotator)
        {
        delete iBitmapRotator;
        iBitmapRotator = NULL;
        }
    }

CBitmapRotatorHandler::CBitmapRotatorHandler(MBitmapRotatorObserver& aObserver)
    :CActive( CActive::EPriorityHigh ),
    iObserver( aObserver )
    {
    }

void CBitmapRotatorHandler::ConstructL()
    {
    CActiveScheduler::Add( this );
    iBitmapRotator = CBitmapRotator::NewL();
    }

void CBitmapRotatorHandler::DoCancel()
    {
```


(C) Copyright Nokia 2013. All rights reserved.

```
    if( iBitmapRotator )
        {
        iBitmapRotator->Cancel();
        }
    }

void CBitmapRotatorHandler::RunL()
    {
    iObserver.RotateComplete( iStatus.Int() );
    }

TInt CBitmapRotatorHandler::RunError( TInt aError )
    {
    return KErrNone;
    }


// Rotates a bitmap by a specified angle.
// This variant of CBitmapRotator::Rotate() schedules a rotate operation
// by a specified number of degrees on the bitmap specified in aBitmap
// and overwrites the original bitmap with the new rotated image.
// This operation is asynchronous, the result is stored in a
// location pointed to by iStatus.
void CBitmapRotatorHandler::Rotate(CFbsBitmap& aBitmap,
CBitmapRotator::TRotationAngle aAngle)
    {
    if( iBitmapRotator )
        {
        if( !IsActive() )
            {
            SetActive();
            iBitmapRotator->Rotate( &iStatus, aBitmap, aAngle );
            }
        }
    }
```

Using CBitmapRotatorHandler

```
iBitmapRotator = CBitmapRotatorHandler::NewL(*this);
iBitmapRotator->Rotate(*iCurrentImage,
CBitmapRotator::ERotation90DegreesClockwise);
```

Postconditions

Bitmap is rotated.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.