SIGGRAPH2005

Using OpenGL ES

Jani Vaarala

Nokia

Using OpenGL ES

- Simple OpenGL ES example
- Fixed point programming

“Hello OpenGL ES”

“Hello OpenGL ES”


```
/* ============================================================
 *  "Hello OpenGL ES" OpenGL ES code.
 *
 *  Siggraph 2005 course on mobile graphics.
 *
 *  Copyright: Jani Vaarala
 * ============================================================
*/

#include <e32base.h>
#include "SigTriangleGL.h"

static const GLbyte vertices[3 * 3] =
{
     -1,    1,    0,
      1,   -1,    0,
      1,    1,    0
};
```

“Hello OpenGL ES”


```
static const GLubyte colors[3 * 4] =
{
        255,    0,      0,      255,
        0,      255,    0,      255,
        0,      0,      255,    255
};

/*****************************************************
 * Initialize OpenGL ES context and initial OpenGL ES state *
 *****************************************************/

void CSigTriangleGL::Construct(RWindow aWin)
{
   iWin = aWin;
```

“Hello OpenGL ES”

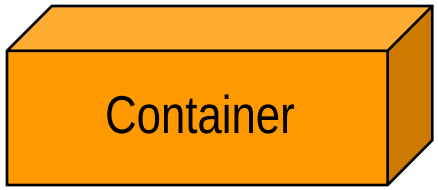

```
static void initGLES(void)
{
   glClearColor        (0.f,0.f,0.1f,1.f);
   glDisable           (GL_DEPTH_TEST);
   glMatrixMode        (GL_PROJECTION);
   glViewport          (0,0,176,208);
   glFrustumf          (-1.f,1.f,-1.f,1.f,3.f,1000.f);
   glMatrixMode        (GL_MODELVIEW);
   glShadeModel        (GL_SMOOTH);
   glVertexPointer     (3,GL_BYTE,0,vertices);
   glColorPointer      (4,GL_UNSIGNED_BYTE,0,colors);
   glEnableClientState (GL_VERTEX_ARRAY);
   glEnableClientState (GL_COLOR_ARRAY);
}
```

“Hello OpenGL ES”


```
TInt CSigTriangleGL::DrawCallback( TAny* aInstance )
{
   CSigTriangleGL* instance = (CSigTriangleGL*) aInstance;

   glClear            (GL_COLOR_BUFFER_BIT);
   glLoadIdentity     ();
   glTranslatef       (0,0,-5.f);
   glDrawArrays       (GL_TRIANGLES,0,3);

   eglSwapBuffers     (instance->iEglDisplay,instance->iEglSurface);

   /* To keep the background light on */
   if (!(instance->iFrame%100))          User::ResetInactivityTime();

   instance->iFrame++;
   return 0;
}
```

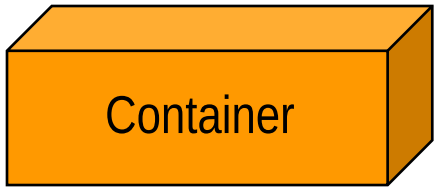
“Hello OpenGL ES”


```
void CSigTriangleContainer::ConstructL(const TRect& /* aRect */)
{
   iGLInitialized = EFalse;

   CreateWindowL();
   SetExtentToWholeScreen();
   ActivateL();

   CSigTriangleGL* gl = new (ELeave) CSigTriangleGL( );
   gl->Construct(Window());

   iGLInitialized = ETrue;
}

CSigTriangleContainer::~CSigTriangleContainer()
{
}
```



“Hello OpenGL ES”


```
void CSigTriangleContainer::SizeChanged()
{
   if(iGLInitialized)
   {
        glViewport(0,0,Size().iWidth,Size().iHeight);
   }
}

TInt CSigTriangleContainer::CountComponentControls() const
{
   return 0;
}

CCoeControl* CSigTriangleContainer::ComponentControl(TInt /* aIndex
   */) const
{
   return NULL;
}
```



““Hello OpenGL ES”


```
/*****************************************************
 * Initialize OpenGL ES context and initial OpenGL ES state *
 *****************************************************/
void CSigTriangleGL::Construct(RWindow aWin)
{
   iWin = aWin;

   iEglDisplay = eglGetDisplay(EGL_DEFAULT_DISPLAY);
   if(iEglDisplay == NULL )    User::Exit(-1);

   if(eglInitialize(iEglDisplay,NULL,NULL) == EGL_FALSE)
                               User::Exit(-1);

   EGLConfig  config,colorDepth;
   EGLint     numOfConfigs = 0;
```

““Hello OpenGL ES”


```
switch( iWin.DisplayMode() )
{
     case (EColor4K):   { colorDepth = 12; break; }
     case (EColor64K):  { colorDepth = 16; break; }
     case (EColor16M):  { colorDepth = 24; break; }
     default:
                          colorDepth = 32;
}

EGLint attrib_list[] = {    EGL_BUFFER_SIZE, colorDepth,
                            EGL_DEPTH_SIZE,  15,
                            EGL_NONE                 };

if(eglChooseConfig(iEglDisplay,attrib_list,&config,1,
     &numOfConfigs ) == EGL_FALSE) User::Exit(-1);
```

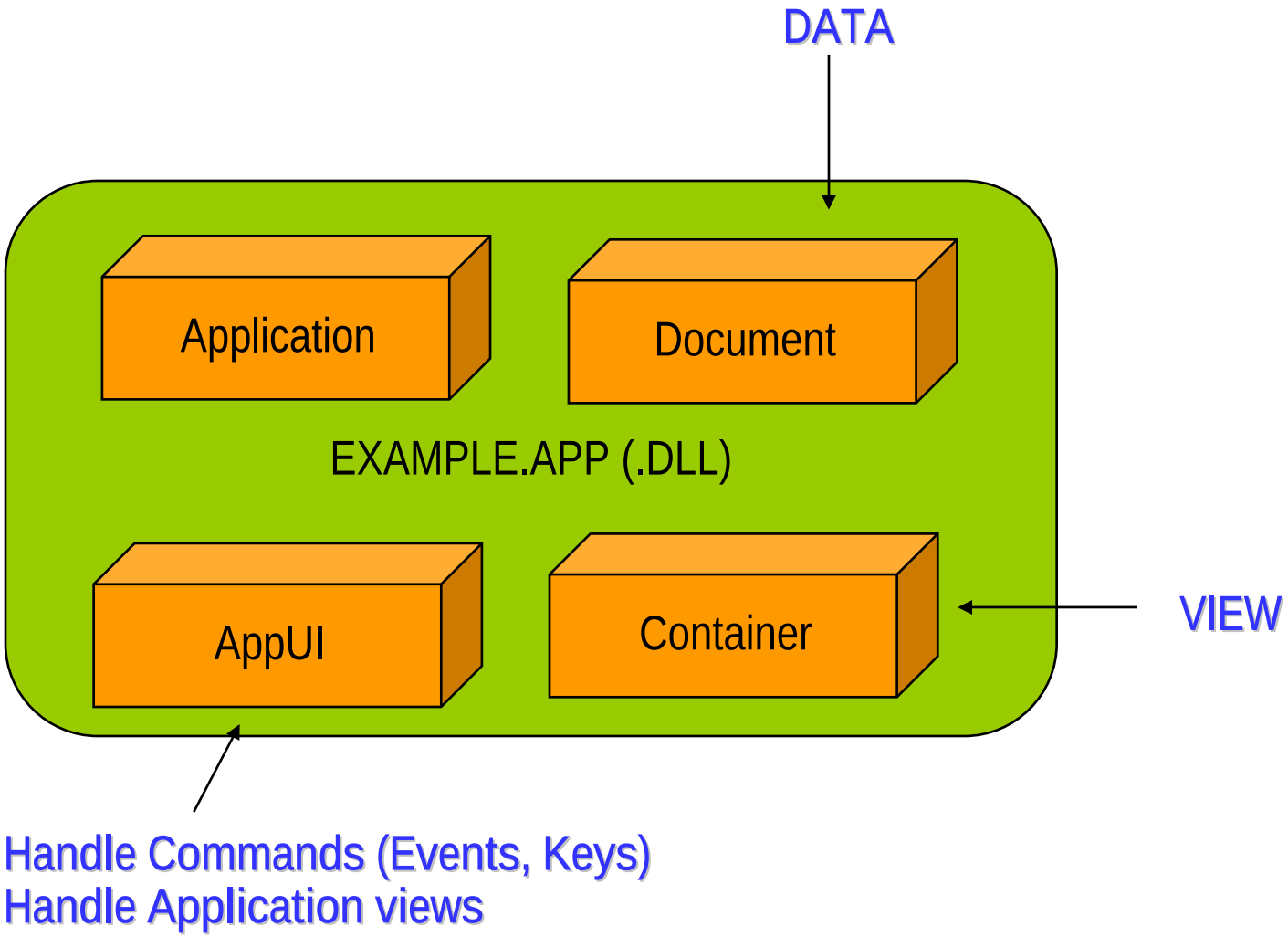
“Hello OpenGL ES”


```
iEglSurface = eglCreateWindowSurface(iEglDisplay, config, &iWin, NULL );
if( iEglSurface == NULL )                User::Exit(-1);

iEglContext = eglCreateContext(iEglDisplay,config, EGL_NO_CONTEXT, NULL );
if( iEglContext == NULL )                          User::Exit(-1);

if( eglMakeCurrent( iEglDisplay, iEglSurface, iEglSurface,
              iEglContext ) == EGL_FALSE )         User::Exit(-1);
```

“Hello OpenGL ES”


```
/* Create a periodic timer for display refresh */
iPeriodic = CPeriodic::NewL( CActive::EPriorityIdle );

iPeriodic->Start( 100, 100, TCallBack(
                            SigTriangleGL::DrawCallback, this ) );

initGLES();
```

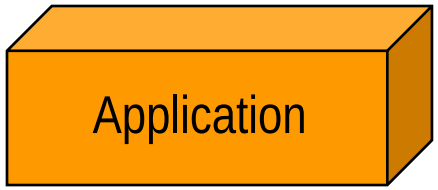
Symbian App Classes



Application class

- Application class starts the application
- Application framework first calls here

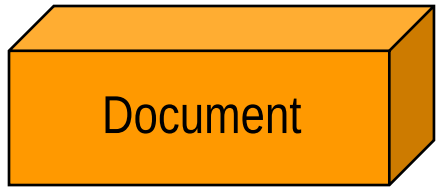
```
CApaDocument* CSigTriangleApp::CreateDocumentL()
{
    return CSigTriangleDocument::NewL( *this );
}

EXPORT_C CApaApplication* NewApplication()
{
    return new CSigTriangleApp;
}
```



Document class

- Document class holds the data (model)

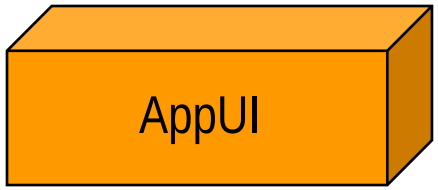
```
CEikAppUi* CSigTriangleDocument::CreateAppUiL()
{
    return new (ELeave) CSigTriangleAppUi;
}
```



AppUI class

- AppUI class handles input events (controller)

```
void CSigTriangleAppUi::HandleCommandL(TInt aCommand)
{
    switch ( aCommand )
    {
        case EAknSoftkeyBack:
        case EEikCmdExit:
         . . .
```

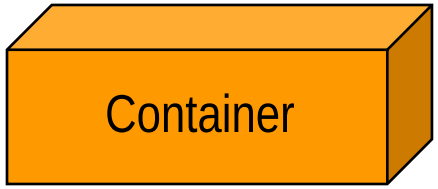



```
TKeyResponse CSigTriangleAppUi::HandleKeyEventL(
    const TKeyEvent& /*aKeyEvent*/,TEventCode /*aType*/)
{
    return EKeyWasNotConsumed;
}
```

Container class

- Container class creates a view from model

```
void CSigTriangleContainer::Draw(const TRect& /* aRect */) const
{

}

void CSigTriangleContainer::ConstructL(const TRect& /* aRect */)
{
   iGLInitialized = EFalse;

   CreateWindowL();
   SetExtentToWholeScreen();
   ActivateL();
```



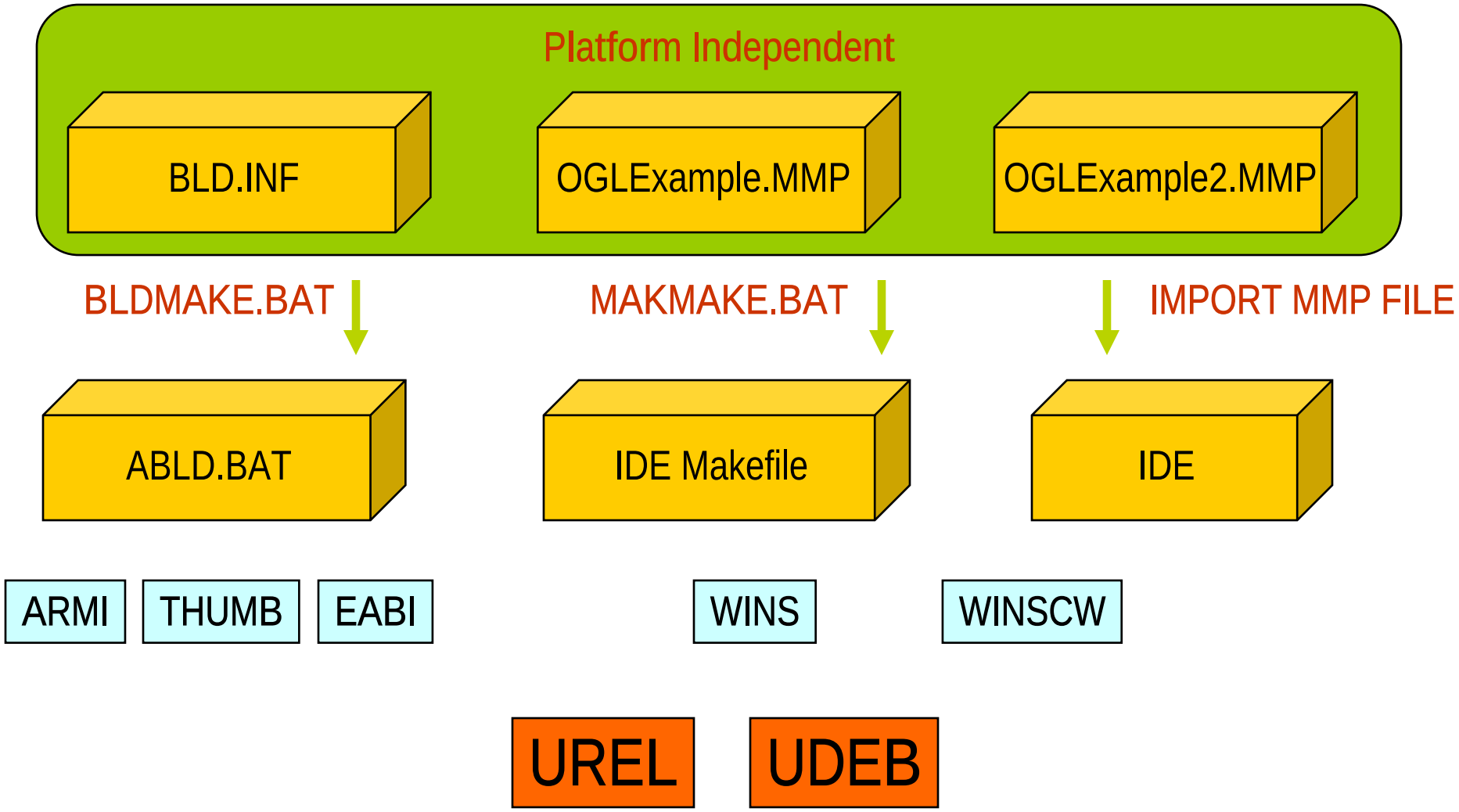
Everything in one slide...

```
/* ========================================================================
 *  "Hello OpenGL ES" Application class.
 *
 *  Siggraph 2005 course on mobile graphics.
 *
 *  Copyright: Jani Vaarala
 * ========================================================================
*/

#include    "SigTriangleApp.h"
#include    "SigTriangleDocument.h"

TUid CSigTriangleApp::AppDllUid() const
{
    return KUidSigTriangle;
}

CApaDocument* CSigTriangleApp::CreateDocumentL()
{
    return CSigTriangleDocument::NewL( *this );
}

EXPORT_C CApaApplication* NewApplication()
{
    return new CSigTriangleApp;
}

GLDEF_C TInt E32Dll( TDllReason )
{
    return KErrNone;
}
/* ========================================================================
 *  "Hello OpenGL ES" Document class.
 *
 *  Siggraph 2005 course on mobile graphics.
 *
 *  Copyright: Jani Vaarala
 * ========================================================================
*/

#include "SigTriangleDocument.h"
#include "SigTriangleAppUi.h"

CSigTriangleDocument::CSigTriangleDocument(CEikApplication& aApp)
: CAknDocument(aApp)
{
}

CSigTriangleDocument::~CSigTriangleDocument()
{
}

void CSigTriangleDocument::ConstructL()
{
}

CSigTriangleDocument* CSigTriangleDocument::NewL(
        CEikApplication& aApp)
{
    CSigTriangleDocument* self = new (ELeave) CSigTriangleDocument( aApp );
    CleanupStack::PushL( self );
    self->ConstructL();
    CleanupStack::Pop();

    return self;
}

CEikAppUi* CSigTriangleDocument::CreateAppUiL()
{
    return new (ELeave) CSigTriangleAppUi;
}

/* ========================================================================
 *  "Hello OpenGL ES" Application UI class.
 *
 *  Siggraph 2005 course on mobile graphics.
 *
 *  Copyright: Jani Vaarala
 * ========================================================================
*/

#include "SigTriangleAppUi.h"
#include "SigTriangleContainer.h"
#include <SigTriangle.rsg>
#include "SigTriangle.hrh"

#include <avkon.hrh>

void CSigTriangleAppUi::ConstructL()
{
    BaseConstructL();

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

CSigTriangleAppUi::~CSigTriangleAppUi()
{
                    if (iAppContainer)
    {
        RemoveFromStack( iAppContainer );
        delete iAppContainer;
    }
}

void CSigTriangleAppUi::DynInitMenuPaneL(
    TInt /*aResourceId*/,CEikMenuPane* /*aMenuPane*/)
{
}

TKeyResponse CSigTriangleAppUi::HandleKeyEventL(
    const TKeyEvent& /*aKeyEvent*/,TEventCode /*aType*/)
{
    return EKeyWasNotConsumed;
}

void CSigTriangleAppUi::HandleCommandL(TInt aCommand)
{
    switch ( aCommand )
    {
        case EAknSoftkeyBack:
        case EEikCmdExit:
        {
            Exit();
            break;
        }
        default:
            break;
    }
}
```

```
/* ========================================================================
 *  "Hello OpenGL ES" Application View (Container) class.
 *
 *  Siggraph 2005 course on mobile graphics.
 *
 *  Copyright: Jani Vaarala
 * ========================================================================
*/

#include "SigTriangleContainer.h"
#include "SigTriangleGL.h"
#include <GLES/gl.h>

void CSigTriangleContainer::ConstructL(const TRect& /* aRect */)
{
    iGLInitialized = EFalse;
    CreateWindowL();
    SetExtentToWholeScreen();
    ActivateL();

    CSigTriangleGL* gl = new (ELeave) CSigTriangleGL( );
    gl->Construct(Window());

    iGLInitialized = ETrue;
}

CSigTriangleContainer::~CSigTriangleContainer()
{
}

void CSigTriangleContainer::SizeChanged()
{
    if(iGLInitialized)
    {
        glViewport(0,0,Size().iWidth,Size().iHeight);
    }
}

TInt CSigTriangleContainer::CountComponentControls() const
{
    return 0;
}

CCoeControl* CSigTriangleContainer::ComponentControl(TInt /* aIndex */) const
{
    return NULL;
}

void CSigTriangleContainer::Draw(const TRect& /* aRect */) const
{
}

void CSigTriangleContainer::HandleControlEventL(
    CCoeControl* /*aControl*/,TCoeEvent /*aEventType*/)
{
}
```

```
/* ========================================================================
 *  "Hello OpenGL ES" OpenGL ES code.
 *
 *  Siggraph 2005 course on mobile graphics.
 *
 *  Copyright: Jani Vaarala
 * ========================================================================
*/

#include <e32base.h>
#include "SigTriangleGL.h"

static const GLbyte vertices[3 * 3] =
{
    -1,    1,    0,
     1,   -1,    0,
     1,    1,    0
};

static const GLubyte colors[3 * 4] =
{
    255,0,0,255,
    0,255,0,255,
    0,0,255,255
};


static void initGLES(void)
{
    glClearColor        (0.f,0.f,0.1f,1.f);
    glDisable           (GL_DEPTH_TEST);
    glMatrixMode        (GL_PROJECTION);
    glViewport          (0,0,176,208);
    glFrustumf          (-1.f,1.f,-1.f,1.f,3.f,1000.f);
    glMatrixMode        (GL_MODELVIEW);
    glShadeModel        (GL_SMOOTH);
    glVertexPointer     (3,GL_BYTE,0,vertices);
    glColorPointer      (4,GL_UNSIGNED_BYTE,0,colors);
    glEnableClientState (GL_VERTEX_ARRAY);
    glEnableClientState (GL_COLOR_ARRAY);
}

/*****************************************************
 * Initialize OpenGL ES context and initial OpenGL ES state *
 *****************************************************/

void CSigTriangleGL::Construct(RWindow aWin)
{
    iWin = aWin;

    iEglDisplay = eglGetDisplay(EGL_DEFAULT_DISPLAY);
    if(iEglDisplay == NULL )
                                                    User::Exit(-1);

    if(eglInitialize(iEglDisplay,NULL,NULL) == EGL_FALSE)
                                                    User::Exit(-1);

    EGLConfig  config,colorDepth;
    EGLint     numOfConfigs = 0;

    switch( iWin.DisplayMode() )
    {
        case (ECOlor4K):   { colorDepth = 12; break; }
        case (ECOlor64K):  { colorDepth = 16; break; }
        case (ECOlor16M):  { colorDepth = 24; break; }
        default:
                           colorDepth = 32;
    }

    EGLint attrib_list[] = {
                        EGL_BUFFER_SIZE, colorDepth,
                        EGL_DEPTH_SIZE,  16,
                        EGL_NONE                    };

    if(eglChooseConfig(iEglDisplay,attrib_list,&config,1,&numOfConfigs ) == EGL_FALSE)
        User::Exit(-1);

    iEglSurface = eglCreateWindowSurface( iEglDisplay, config, &iWin, NULL );
    if( iEglSurface == NULL )
                                                    User::Exit(-1);

    iEglContext = eglCreateContext( iEglDisplay, config, EGL_NO_CONTEXT, NULL );

    if( iEglContext == NULL )
                                                    User::Exit(-1);

    if( eglMakeCurrent( iEglDisplay, iEglSurface, iEglSurface, iEglContext ) == EGL_FALSE )
                                                    User::Exit(-1);

    /* Create a periodic timer for display refresh */
    iPeriodic = CPeriodic::NewL( CActive::EPriorityIdle );
    iPeriodic->Start( 100, 100, TCallBack( CSigTriangleGL::DrawCallback, this ) );

    initGLES();
    iFrame = 0;
}

CSigTriangleGL::~CSigTriangleGL()
{
}

/*****************************************************
 * Draw callback executed regularly by a timer callback
 *****************************************************/

TInt CSigTriangleGL::DrawCallback( TAny* aInstance )
{
    CSigTriangleGL* instance = (CSigTriangleGL*) aInstance;

    glClear          (GL_COLOR_BUFFER_BIT);
    glLoadIdentity   ();
    glTranslatef     (0,0,-5.f);
    glDrawArrays     (GL_TRIANGLES,0,3);

    eglSwapBuffers   (instance->iEglDisplay, instance->iEglSurface);

    /* To keep the background light on */
    if (!(instance->iFrame%100))        User::ResetInactivityTime();

    instance->iFrame++;
    return 0;
}
```

Compiling SW



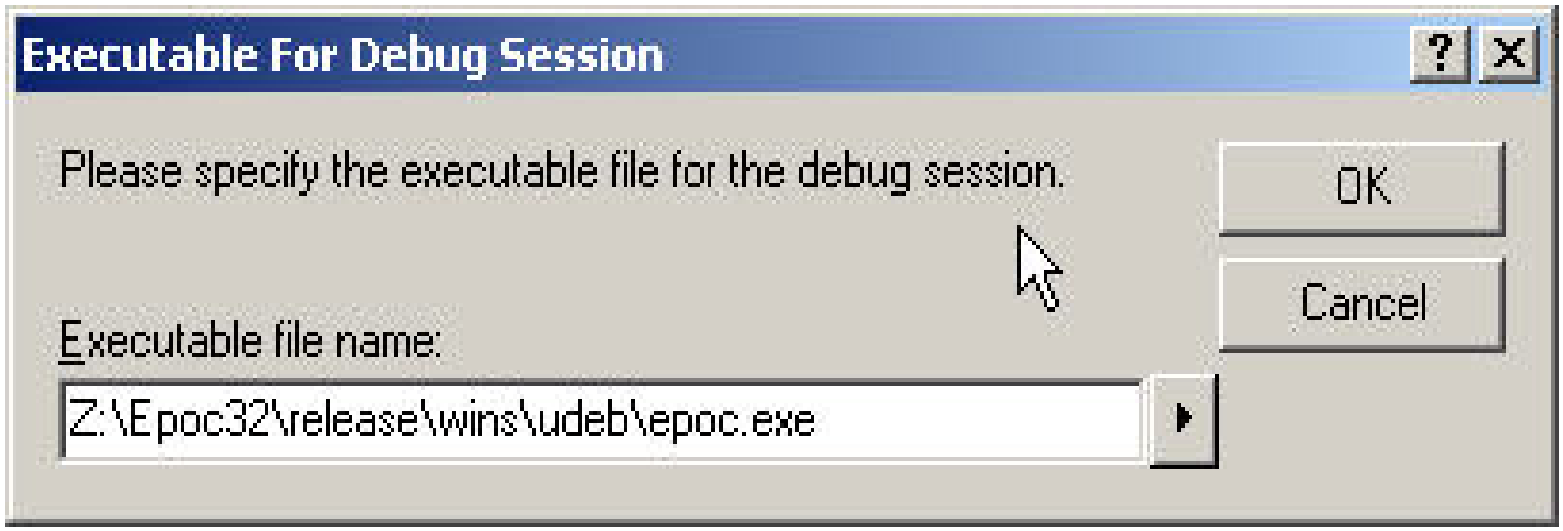
MMP file contents


```
TARGET                  SigTriangle.app
TARGETTYPE              app
UID                     0x100039CE 0x01CC1856
TARGETPATH              \system\apps\SigTriangle

SOURCEPATH              ..\src
SOURCE                  SigTriangleApp.cpp
SOURCE                  SigTriangleGL.cpp
 . . .

USERINCLUDE             ..\inc
 . . .

LIBRARY                 libgles_cm.lib ws32.lib
```

Compiling in Windows

SIGGRAPH2005

Just press F5 to build'n'run (Visual Studio)



Step-by-step guide in the course notes:

Sig2005/SigTriangle/readme.txt

Emulator

SIGGRAPH2005

Emulator

SIGGRAPH2005

Compiling for ARM Target

- Target build is done in the group folder:

  bldmake.bat bldfiles

  ./abld.bat build thumb urel OR

  ./abld.bat build armi urel

- Install file (SigTriangle.sis) is created (sis folder)

  makesis.bat SigTriangle.pkg

- PKG-file is a list of files to be included in the .sis file
- Installation file can be sent to phone

Fixed point programming

- Why to use it?
  - Most mobile handsets don’t have a FPU
- Where does it make sense to use it?
  - Where it makes the most difference
  - For per-vertex processing: morphing, skinning, etc.
  - Per vertex data shouldn’t be floating point
- OpenGL ES API supports 32-bit FP numbers

Fixed point programming

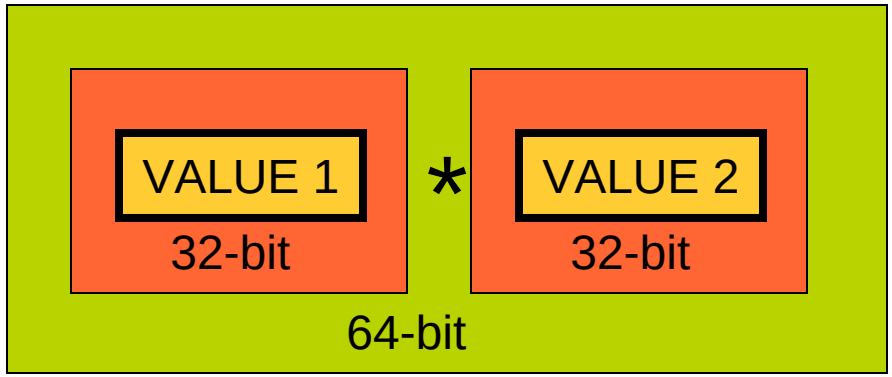
- There are many variants of fixed point:
  - Signed / Unsigned
  - 2’s complement vs. Separate sign
- OpenGL ES uses 2’s complement
- Numbers in the range of [-32768, 32768 [
- 16 bits for decimal bits (precision of 1/65536)
- All the examples here use .16 fixed point

Fixed point programming

- Examples:

```
0x0001 0000 =  “1.0f”
0x0002 0000 =  “2.0f”
0x0010 0000 =  “16.0f”
0x0000 0001 =  1/0x10000(0x10000 = 2^16)
0xffff ffff = -1/0x10000(-0x0000 0001)
```

Fixed point programming



>> 16 = RESULT

Intermediate overflow
- Higher accuracy (64-bit)
- Downscale input
- Redo range analysis

Result overflow
- Redo range analysis
- Detect overflow, clamp

Fixed point programming

- Convert from floating point to fixed point

```
#define float_to_fixed(a)  (int)((a)*(1<<16))
```

- Convert from fixed point to floating point

```
#define fixed_to_float(a)  (((float)a)/(1<<16))
```

- Addition

```
#define add_fixed_fixed(a,b) ((a)+(b))
```

- Multiply fixed point number with integer

```
#define mul_fixed_int(a,b) ((a)*(b))
```

Fixed point programming

- MUL two FP numbers together

```
#define mul_fixed_fixed(a,b) (((a)*(b)) >> 16)
```

  - If another multiplier is in] -1.0, 1.0 [, no overflow
- Division of integer by integer to a fixed point result

```
#define div_int_int(a,b) (((a)*(1<<16))/(b))
```

- Division of fixed point by integer to a fixed point result

```
#define div_fixed_int(a,b) ((a)/(b))
```

- Division of fixed point by fixed point

```
#define div_fixed_fixed(a,b) (((a)*(1<<16))/(b))
```

Fixed point programming

- Power of two MUL & DIV can be done with shifts
- Fixed point calculations overflow easily
- Careful analysis of the range requirements is required
- Always try to use as low bit ranges as possible
  - 32x8 MUL is faster than 32x32 MUL (some ARM)
  - Using unnecessary “extra bits” slows execution
- Always add debugging code to your fixed point math

Fixed point programming


```
#if defined(DEBUG)
int add_fix_fix_chk(int a, int b)
{
  int64 bigresult = ((int64)a) + ((int64)b);
  int smallresult = a + b;
  assert(smallresult == bigresult);
  return smallresult;
}
#endif

#if defined(DEBUG)
#  define add_fix_fix(a,b) add_fix_fix_chk(a,b)
#else
#  define add_fix_fix(a,b) ((a)+(b))
#endif
```

Fixed point programming

- Complex math functions
  - Pre-calculate for the range of interest
- An example: Sin & Cos
  - Sin table between [0, 90°]
  - Fixed point angle
  - Generate other angles and Cos from the table
  - Store as fixed point ((short) (sin(angle) * 32767))
  - Performance vs. space tradeoff: calculate for all angles

Fixed point programming

- Sin
  - 90 ° = 2048 (our angle scale)
  - Sin table needs to include 0° and 90°

```
INLINE fp_sin(int angle)
{
   int phase                    = angle & (2048 + 4096);
   int subang                   = angle & 2047;

   if( phase == 0 )             return sin_table  (subang);
   else if( phase == 2048 )     return sin_table  (2048 - subang);
   else if( phase == 4096 )     return –sin_table (subang);
   else                         return –sin_table (2048 – subang);
}
```

Example: Morphing

- Simple fixed point morphing loop (16-bit data, 16-bit coeff)

```
#define DOMORPH_16(a,b,t) (TInt16)(((((b)-(a))*(t))>>16)+(a))

void MorphGeometry(TInt16 *aOut, const TInt16 *aInA, const TInt16
   *aInB, TInt aCount, TInt aScale)
{
   int i;

   for(i=0; i<aCount; i++)
   {
        aOut[i*3+0] = DOMORPH_16(aInB[i*3+0], aInA[i*3+0], aScale);
        aOut[i*3+1] = DOMORPH_16(aInB[i*3+1], aInA[i*3+1], aScale);
        aOut[i*3+2] = DOMORPH_16(aInB[i*3+2], aInA[i*3+2], aScale);
   }
}
```

Example: Morphing

SIGGRAPH2005

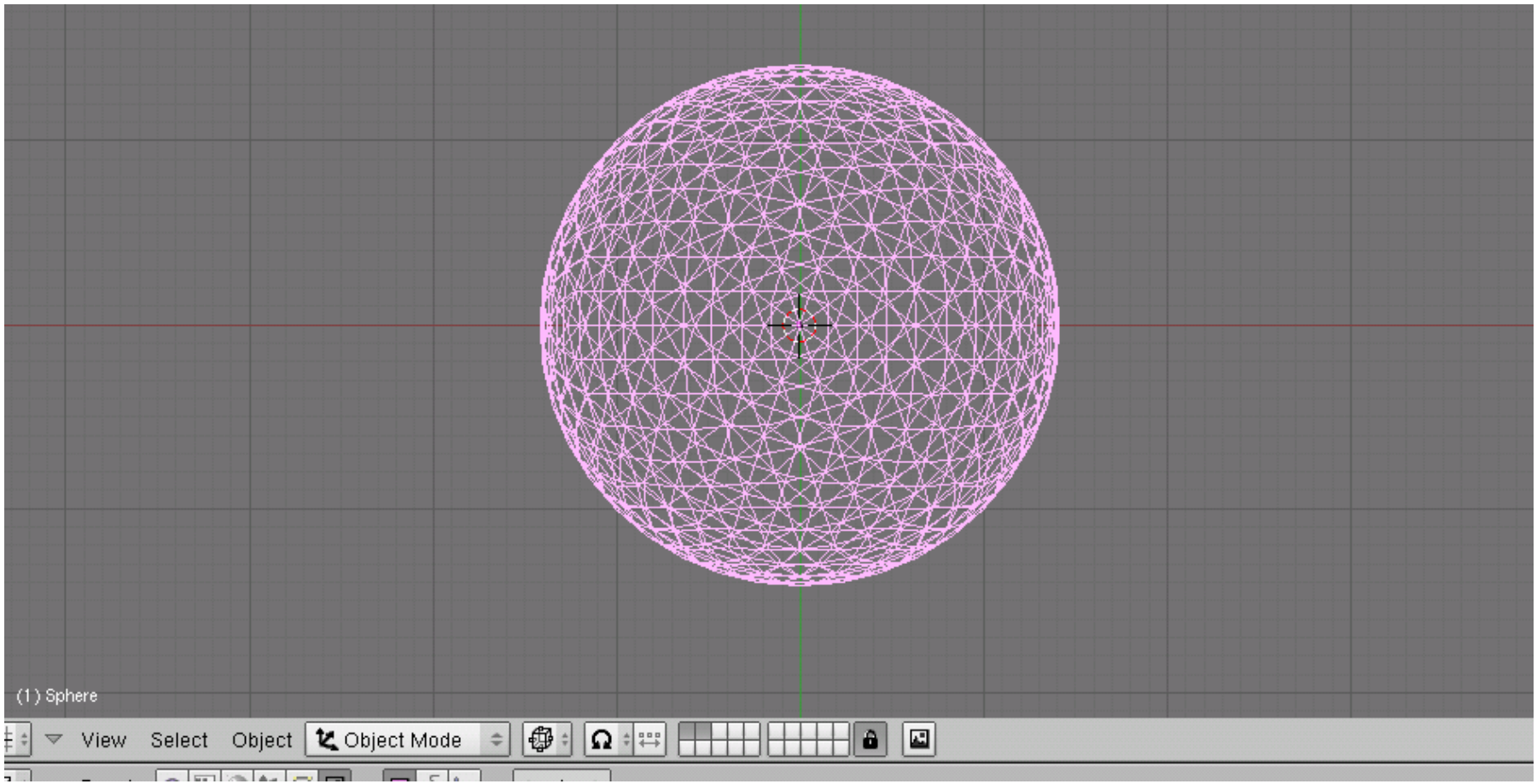


Image from Blender (http://www.blender3d.org)

Example: Morphing

SIGGRAPH2005

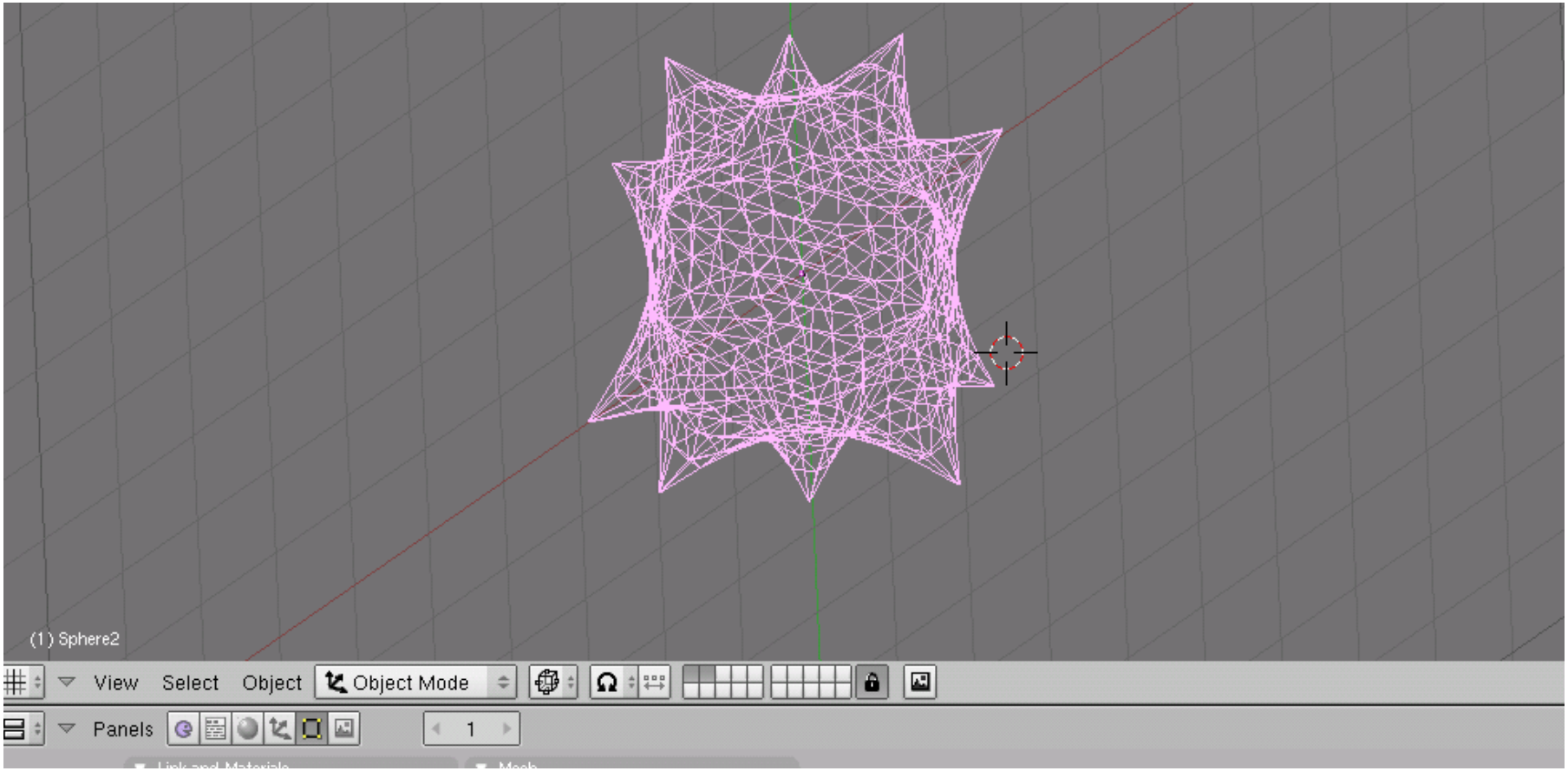


Image from Blender (http://www.blender3d.org)

Example: Morphing

SIGGRAPH2005

- Source code in the course notes
- Blender exporter in course notes
- 2 Blender models
- Exported vertices and normals
- Fixed point morphing for both

Questions?