Graphics on Raspberry Pi

Pavol Rusnak @pavolrusnak

Hardware

Broadcom Videocore IV (BCM2835)

- Features:
  - H.264/MPEG-4 AVC
  - MPEG-2, VC-1 (license required)
  - OpenGL ES 1.1/2.0

- Outputs:
  - Composite RCA (PAL+NTSC)
  - HDMI (rev 1.3 & 1.4)
  - DSI

OpenGL ES 2.0

- subset of OpenGL 2.0
- no GLU or GLUT
- no fixed-function pipeline
  - similar to 3.0->3.1 move
  - not compatible with ES 1.1
- Android 2.0+, iOS, Blackberry, NaCl, WebGL, Raspberry Pi, ...

OpenGL (fixed pipeline)

```
#include <GL/gl.h>

glMatrixMode (GL_PROJECTION);
glLoadIdentity();
glFrustum(-400, 400, -300, 300, 0.1, 5000.0);

glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glTranslatef(10, 10, 0);
glRotatef(45, 0, 0, 1);

glBegin(GL_TRIANGLE_STRIP);
glColor4f(1,0,0,1); glTexCoord2f(0, 0); glVertex2f(-100.0, +100.0);
glColor4f(0,1,0,1); glTexCoord2f(0, 1); glVertex2f(-100.0, -100.0);
glColor4f(0,0,1,1); glTexCoord2f(1, 0); glVertex2f(+100.0, +100.0);
glColor4f(1,1,0,1); glTexCoord2f(1, 1); glVertex2f(+100.0, -100.0);
glEnd();
```

OpenGL ES (programable pipeline)

fragment shader:

```
varying vec4 v_col;
varying vec2 v_texcoord;
uniform sampler2D tex;
void main() {
   gl_FragColor = texture2D(tex, v_texcoord);
   gl_FragColor *= v_col;
}
```

vertex shader:

```
uniform mat4 m_modelview;
uniform mat4 m_projection;
attribute vec4 pos;
attribute vec4 col;
attribute vec2 texcoord;
varying vec4 v_col;
varying vec2 v_texcoord;
void main() {
   gl_Position = m_projection * m_modelview * pos;
   v_texcoord = texcoord;
   v_col = col;
}
```

OpenGL ES (programable pipeline)

```
#include <GLES2/gl2.h>
#include <GLES2/gl2ext.h>

matProjection = mat4::Frustum(-400, 400, -300, 300, 0.1, 5000.0);
glUniformMatrix4fv(u_Projection, 1, GL_FALSE, matProjection.Pointer());

matModelView =
        mat4::Translate(10.0, 10.0, 0.0) *
        mat4::Rotate(45, 0.0, 0.0, 1.0);
glUniformMatrix4fv(u_ModelView, 1, GL_FALSE, matModelView.Pointer());
```

matrix.h: https://github.com/prusnak/armap/blob/master/armap/matrix.h

OpenGL ES (programable pipeline)

```
static const GLfloat verts[4][2] = {
    { -100, 100 },
    { -100, -100 },
    { 100, 100 },
    { 100, -100 }
};
static const GLfloat texcoords[4][2] = {
    {0, 0},
    {0, 1},
    {1, 0},
    {1, 1}
};
static const GLfloat colors[4][4] = {
    {1,0,0,1},
    {0,1,0,1},
    {0,0,1,1},
    {1,1,0,1}
};
```

OpenGL ES (programable pipeline)

```
glVertexAttribPointer(attr_pos, 2, GL_FLOAT, GL_FALSE, 0, verts);
glVertexAttribPointer(attr_col, 4, GL_FLOAT, GL_FALSE, 0, colors);
glVertexAttribPointer(attr_tex, 2, GL_FLOAT, GL_FALSE, 0, texcoords);

glEnableVertexAttribArray(attr_pos);
glEnableVertexAttribArray(attr_col);
glEnableVertexAttribArray(attr_tex);

glDrawArrays(GL_TRIANGLE_STRIP, 0, 4);

glDisableVertexAttribArray(attr_pos);
glDisableVertexAttribArray(attr_col);
glDisableVertexAttribArray(attr_tex);
```

Build

“normal” hardware:

```
-lGLESv2
```

Raspberry Pi:

```
-I/opt/vc/include -I/opt/vc/include/interface/vcos/pthreads
-L/opt/vc/lib
-lGLESv2
```

VideoCore

```
#include <bcm_host.h>

bcm_host_init();

bcm_host_deinit();
```

Make it portable

Makefile:

```
LIBS+=-lGLESv2
ifeq ($(shell test -d /opt/vc/include && echo 1),1)
CFLAGS+=-I/opt/vc/include -I/opt/vc/include/interface/vcos/pthreads -DRPI=1
LDFLAGS+=-L/opt/vc/lib
endif
```

code:

```
#ifdef RPI
#include <bcm_host.h> /// bcm_host_init(); /// bcm_host_deinit();
#endif
```


GK2.5K