Computer Graphics

spring, 2013

Chapter 3
An Introduction to EGL

EGL

- Provides mechanisms for...
  - Communicating with the native windowing system
  - Querying the available types & configurations of drawing surfaces (eglGetConfigs, eglChooseConfig)
  - Creating drawing surfaces (eglCreateWindowSurface, eglCreatePbufferSurface)
  - Synchronizing rendering between OpenGL ES 2.0 and other graphics-rendering APIs (e.g. OpenVG) (eglWaitClient, eglWaitNative)
  - Managing rendering resources such as texture maps

EGL

- “Glue” layer between GLES2 and the native windowing system, such as X Window (GNU/Linux), MS Windows, or Quartz (OSX)

Steps

1. Create & initialize a connection with the local EGL display (eglGetDisplay)
2. Initialize EGL (eglInitialize)
3. Choose a surface configuration (eglGetConfigs, eglChooseConfig)
4. Create a rendering surface (eglCreateWindowSurface, eglCreatePbufferSurface)
5. Create a context (eglCreateContext)
6. Bind the context with the rendering surface (eglMakeCurrent)

eglGetDisplay

- Connect to the EGL display server
- EGLDisplay eglGetDisplay(EGLNativeDisplayType display_id);
  - EGLNativeDisplayType display_id
    - Defined to match the native window system's display type (e.g. HDC on MS Windows)
    - EGL_DEFAULT_DISPLAY accepted
  - Returns EGL_NO_DISPLAY if a display connection is not available

Error Checking for EGL

- Most EGL functions return EGL_TRUE/ EGL_FALSE
- Error codes should be queried by eglGetError

eglInitialize

- Initialize EGL's internal data structure and returns the version of EGL implementation
- EGLBoolean eglInitialize(EGLDisplay display, EGint *majorVersion, EGint *minorVersion);
- Error codes: EGL_BAD_DISPLAY, EGL_NOT_INITIALIZED

Determining the Available Surface Configurations

- Two ways
  - Query every surface configurations and find the best choice ourselves (eglGetConfigs)
  - Specify a set of requirements and let EGL make a recommendation for the best match (eglChooseConfig)
- EGLConfig returned

eglGetConfigs

- To query all EGL surface configurations
- EGLBoolean eglGetConfigs(EGLDisplay display, EGLConfig *configs, EGLint maxReturnConfigs, EGLint *numConfigs);
- Two ways
  - configs==NULL: numConfigs set and can be used to allocate enough memory
  - configs!=NULL: configs filled with available configurations

EGLConfig

- Contains all of the information about a surface made available by EGL
  - # of colors, additional buffers (e.g. depth & stencil buffers), type of surfaces, etc.
- An attribute can be queried by eglGetConfigAttrib

eglChooseConfig

- To let EGL make the choice of matching EGLConfig
- EGLBoolean eglChooseConfig(EGLDisplay display, const EGLint *attribList, EGLConfig *config, EGLint maxReturnConfigs, EGLint *numConfigs);
- Returned configurations are sorted properly

eglCreateWindowSurface

- Create a window
- EGLSurface
  eglCreateWindowSurface(EGLDisplay display, EGLConfig config, EGLNativeWindowType native_window, const EGLint *attribList);
- Attribute: EGL_RENDER_BUFFER
- Only double-buffering is supported on GLES2

pbuffers

- Nonvisible off-screen surfaces
- Most often used for generating texture maps (cf: use FBO for “render-to-texture”)
- EGLBoolean eglCreatePbufferSurface(EGLDisplay display, EGLConfig config, const EGLint *attribList);

Rendering Context

- Data structure internal to GLES2
- Contains all of the states required for operation
  - EGLContext eglCreateContext(EGLDisplay display, EGLConfig config, EGLContext shareContext, const EGLint *attribList);
  - Attribute: EGL_CONTEXT_CLIENT_VERSION

Binding a Context with a Rendering Surface

- EGLBoolean eglMakeCurrent(EGLDisplay display, EGLSurface draw, EGLSurface read, EGLContext context);
- Usually draw==read