X on GL

Adam Jackson
Wild Open Source

ajax@nwnk.net

Introduction

- X servers typically targetted to an output device
- For hardware, performance matters
- Current hardware servers are slow
- Why not target an API?

Motivation

- Open standard
- Widely supported
- Extensible (both directions)
- Mandatory anyway
- ... and very very fast

OpenGL Architecture

- Rendering core
  - OpenGL, ES profiles
- Versions
  - 1.0 through 2.0
  - Extensions
- Window system binding
  - GLX, WGL, AGL

First Effort: Xglx

- Cairo has multiple output backends
- glitz backend outputs to GL
  - Actually, GLX/WGL/AGL
- glitz-glx + X input events = Xglx
- But still nested

Leaving the Nest

- miniglx from Tungsten Graphics
- OpenGL|ES project's EGL API
  - `glXCreateContext(Display *, XVisualInfo *, GLXContext, Bool);`
  - `eglCreateNewContext(EGLDisplay, EGLConfig, EGLContext, const int *);`
- EGL_MESA_screen_surface
- EGL_MESA_choose_display

Code Duplication

- Input drivers
- Mode setting
- Configuration
- Historical baggage needs to be dropped
- Factor useful stuff up to DIX layer

X Without Root

- Requires kernel support
- DRM API
- Mode setting needs verification
- PCI support into the kernel

Future GL Work

- Memory management needs work
- pbuffers and FBOs are unimplemented
- Shared textures
  - GL_APPLE_client_storage
- Composite integration
- Accelerated indirect GLX

Questions?