Looking Glass Modifications to Xorg

Deron Johnson
Version 1.1
Fri Aug 25 13:53:46 PDT 2006

1 Introduction

This document describes the changes which have been made to the Xorg window system server by the Project Looking Glass team. These changes have been made to integrate Xorg into the 3D Looking Glass (LG) window system.

Several changes have been made. A new extension has been added and the input subsystem has been modified. In addition, certain window tree operations have been modified to work in a 3D window system and the handling of certain override redirect windows has been changed. The display of the X cursor has been disabled and QueryPointer has been modified to work better in a 3D window system. Finally, miscellaneous bug fixes (for X server, client, and DDX bugs) have been implemented.

The LG Xorg modifications are based on the source code for Xorg 6.8.2. All of these changes are currently enclosed in `#ifdef LG3D`. The latest LG source code resides in the `lg3d-dev-0-7-1` branch of the Xorg CVS.

The file path names in this document are relative to `xc/programs/Xserver` unless otherwise specified.

2 New Extension: LGE

LG adds to Xorg a new extension called LGE which is used to put Xorg into *Looking Glass mode*. In this mode, input device events are redirected through the LG Display Server (DS). In addition, other behavior of Xorg is altered to function in a 3D window system.

The LGE extension is used exclusively by the DS. It is used to inform Xorg that the DS is running and that Xorg should start operating in LG mode.

Refer to the the document `xc/doc/specs/LG/lge_protocol.pdf` and the man page `xc/lib/Xlg3d/Xlg3d.man` for more information.

(In the following list of new files added by LG, all of the path names except `Xext` are relative to the top of the Xorg CVS tree, `xc`. `Xext` is a subdirectory of `xc/programs/Xserver`).

New files:

```
include/extensions/lgewire.h
lib/Xlg3d/AUTHORS
lib/Xlg3d/autogen.sh
```

```
lib/Xlg3d/ChangeLog
lib/Xlg3d/configure.ac
lib/Xlg3d/COPYING
lib/Xlg3d/CVS/
lib/Xlg3d/Imakefile
lib/Xlg3d/INSTALL
lib/Xlg3d/Makefile.am
lib/Xlg3d/NEWS
lib/Xlg3d/README
lib/Xlg3d/Xlg3d-def.cpp
lib/Xlg3d/Xlg3d.man
lib/Xlg3d/xlg3d.pc.in
lib/Xlg3d/Xlge.c
lib/Xlg3d/Xlge.h
Xext/lge.c
Xext/lgeint.h
```

(In the following list files modified by LG `config`, `include/extensions`, and `lib` are relative to the top of the Xorg CVS tree, `xc`. All other path names are relative to `xc/programs/Xserver`).

Modified files:

`config/cf/X11.tmpl`
  Standard changes for the addition of a new extension library.

`include/extensions/Imakefile`
  Standard changes for the addition of a new extension include file.

`lib/Imakefile`
  Standard changes for the addition of a new extension library.

`Xext/Imakefile`
  Addition of new file `lge.c`

`mi/miinitext.c:`
  Standard changes for the addition of a new extension:
  - LG3D code is disabled in `Xprt`
  - Added new global variable: `Bool noLgeExtension`
  - Add LGE to `ExtensionToggleList` and `staticExtensions`
  - Add `LgeExtensionInit` call to `InitExtensions`

`include/globals.h`
  Added new global variable: `Bool noLgeExtension`

`os/utils.c`
  Added new global variable: `Bool noLgeExtension`

3 Event Subsystem Modifications

The event subsystem of Xorg has been modified to redirect device input events through the DS so that the events can be associated with objects in the 3D scene graph.

These modifications are discussed in detail in the file `xc/doc/specs/LG/lg_event_trip.pdf`.

New files:
```
      dix/xytosubwin.c
```

Modified files:
```
      dix/events.c
      xkb/xkbPrKeyEv.c
```

4 QueryPointer

The semantics of the `QueryPointer` X request in a 3D window system are problematic. For example, what values should be returned by QueryPointer when the avatar[1] for the argument window is slanted in the Z dimension and the cursor is inside the window? And what should be returned when the cursor is in some other window or not in any window at all? Because 2D X11 clients have no idea they are running in a 3D window system it is hard to define a set of semantics that produce meaningful values in all cases.

In a 3D window system, the sprite location has 3 coordinates (x, y, and z). One option would be to project the sprite location onto the image plane and then take the resulting 2D screen absolute coordinates and use them to calculate the `QueryPointer` return values. However, if the window avatar for the argument window is slanted in the Z dimension, then this would return values that the 2D client would not be expecting, and the values could be misinterpreted.

A better approach is to have the X server maintain the notion of a *virtual sprite* position. The virtual sprite position is only defined when the sprite is inside an X11 window. The virtual sprite position consists of a window reference and the coordinates of the sprite **relative to that window**. `QueryPointer` uses this virtual sprite position to calculate the return values. But these values are only meaningful in the case where the argument window is inside the current virtual sprite's top-level window, or is the top-level window itself. Fortunately, this is the most frequent case we have encountered during LG testing.

However, when the QueryPointer argument window is some other window which is outside the current sprite's top-level window, or not in any window at all, the return values will probably not make sense to the client. We have not yet found a good solution to this problem. Fortunately, this case happens infrequently.

Modified files:
```
   dix/events.c
```

5 Disable Incompatible Extensions

Certain X extensions have not yet been upgraded to be compatible with LG. Specifically, Xvideo and XVideo-MotionCompensation are not compatible because their rendering is not redirectable via the

1 A window *avatar* is the 3D object onto which the image of a 2D window is texture mapped.

composite extension. And XINPUT is not compatible because the LG modifications to the Xorg event subsystem do not yet support XINPUT devices.

Programs which use these extensions cannot run in LG without causing problems. So these extensions are explicitly disabled while Xorg is in LG mode. To perform this sort of run-time extension configuration, the following changes have been made:

- `dix/extension.c:ProcListExtensions` has been modified to not include incompatible extensions in the list it sends to the client, and
- Using `LgeControlLgMode` to enable LG mode will result in the dispatch functions for incompatible extensions to be replaced with routines that print an error message to the server log and return `BadAccess`.

Note: These changes are sufficient to prevent XVideo and XINPUT applications from running. But they are not sufficient to prevent OpenGL applications from running, because these applications do not check for the existence of the GLX (or NV-GLX) extension, nor does the OpenGL library itself. We have not yet found a suitable method for preventing OpenGL apps from running in LG.

New files:
```
    Xext/lge.c
```

Modified files:
```
    dix/extension.c
```

6 Solaris-Specific Work Arounds

This section describes certain bugs which are known to occur only on the Solaris platform and how they were worked around for LG.

6.1 Solaris Nvidia Driver Initialization Bug Work Around

Problem:

When LG was first ported to Solaris, invalid drawing occurred when the user scrolled individual lines of a StarOffice or Mozilla window. This problem was caused by an interaction between Xorg's kludgey way of initializing the Damage extension and the Nvidia driver (DDX). Only the Solaris x86 Nvidia driver (version 1.0-7663) seems to have a problem with this kludgey Xorg code; the Linux Nvidia driver (version 1.0-7174) doesn't have this problem.

Xorg has some kludgey code in `miSpriteInitialize` which calls `DamageSetup` during `InitOutput` (i.e. the DDX initialization routine). Since `miSpriteInitialize` calls `DamageSetup`, `DamageSetup` is getting called before `InitOutput` returns. On Linux, after `InitOutput` calls `miSpriteInitialize` it doesn't change any screen ops. But on

Solaris, `InitOutput` **does change** screen ops (such as `CreateGC`) after calling `miSpriteInitialize`. This messes up the wrapping of the Damage extension.

LG Change:

LG removes the `DamageSetup` call from `miSpriteInitialize` and adds calls to `DamageSetup` for all screens to the start of `InitExtensions`.(Perhaps there is a better way to handle this, such as cleaning up the kludgey initialization of the Damage extension).

Modified Files:

```
mi/miinitext.c
mi/misprite.c
```

7 Acknowledgements

The following individuals provided invaluable assistance in the design of the LG Xorg Modifications:

Paul Byrne
Amir Bukhari
Alan Coopersmith
Jay Cotton
Jim Gettys
James Gosling
Hideya Kawahara
Stuart Kreitman
Keith Packard

The author would also like to gratefully acknowledge the members of the Project Looking Glass community for their rigorous testing of the LG system, which brought many of these issues to light.