Overview
Graphics and Input in Tizen
Seokjae Jeong, PhD.
seok.j.jeong@samsung.com
Contents

• Overall Architecture
• Window System
• Enlightenment Foundation Libraries & Graphics
• Input Service Framework
• Voice Framework
• Elementary Widgets
• References
Overall Architecture
Graphics Rendering Architecture

3D App
OpenGL ES/EGL
OpenGL ES/EGL
EFL App.
EFL EVAS
X server
Composite Window Manager (E17)
EFL EVAS
OpenGL ES/EGL

Video App.
EFL MM FW

EFL win
OpenGL ES win
Video win
Textures

Legend
Process  Texture Mem.  Rendering pipeline path; rendering data flow  Texture from Pixmap

Frame Buffer
Window System (X11)
X Window System

- X Window System History
  - X Window System is commonly called as X11 or X.
  - X originated at MIT in 1984.
  - X was specially designed to be used over network connections rather than on an integral or attached display device.
  - At present, the X.Org Foundation leads the X project with the current reference implementation which is X.Org server.

- Tizen 1.0 Larkspur includes X.Org R7.6 and xserver 1.9.3.
X Software Architecture (1/2)

- X has the client-server model architecture.
- X logically be thought to be two part, the client linked with Xlib called X client and the server which is generally called X server.
- The X client can communicate with X server through the X protocol translated by Xlib.
- The X client sends graphical request and gets user input.
- The X server accepts request for graphical output (windows) and sends back user input (keyboard, mouse).
X Software Architecture (2/2)

**Xlib**
Provides functions for connecting to a display server, creating windows, drawing graphics, responding to events, and so on.

**Application**
A program written with Xlib.

**Window Manager**
It controls the placement and appearance of application windows.

**Toolkit**
Implements a set of user interface features such as menus or command buttons. Ex. GTK+, Qt, EFL

**Display Server**
A program that runs on each system that supports a display.

**X server**
Provides functions for connecting to a display server, creating windows, drawing graphics, responding to events, and so on.

**Display**
A workstation consisting of keyboard(s), pointing devices, and one or more screens.

**Network**

- **X window system protocol requests** are sent from clients.
- **Events and replies** are passed back to clients.

**Legend**
- Green: X client
- Green: Toolkit
- Purple: Device drivers
- Blue: Xlib
- Purple: X Server

- Network
- Display
- X server

- X window system protocol requests are sent from clients.
- Events and replies are passed back to clients.
X Server (1/2)

X window system

Xlib (libX11 and extension libs)

X protocol

X server

Extension Layer

DDX

DIX

OS

Mouse

Keyboard (Keypad)

Touch

FrameBuffer

Legend

- Xlib
- X Window System
- X server
- Linux kernel / Frame Buffer/ Device Drivers
X Server (2/2)

• DIX(Device Independent X) layer
  – Is the part of server dealing with clients
  – Gets requests from clients and sends input events to clients
  – Unpacks request and sends them to the appropriate DDX or Extensions
  – Enables X server to get common functionality in respect of X clients where the X server is ported on any device.

• DDX(Device Dependent X) layer
  – Is the part of server dealing with hardware related in graphics and input

• OS(Operating System) layer
  – Manages client connections
  – Schedules work to be done for clients
  – Provides routines for memory allocation and free

• Extension layer
  – Enables to get additional functionalities for X server
Enlightenment Foundation Libraries & Graphics
What is EFL? (1/2)

• “Enlightenment” is not just a window manager for Linux/X11 and others, but also a whole suite of libraries to help you create beautiful user interfaces with much less work than doing it the old fashioned way and fighting with traditional toolkits, not to mention a traditional window manager. It covers from small mobile devices like phones all the way to powerful multi-core desktops (which are the primary development environment).
What is EFL? (2/2)

- These provide both a semi-traditional toolkit set in Elementary as well as the object canvas (Evas) and powerful abstracted objects (Edje) that you can combine, mix and match, even layer on top of each other with alpha channels and events in-tact. It has 3D transformations for all objects and more.

A simple overview of the EFL stack

www.enlightenment.org
Enlightenment Foundation Libraries

- **Elementary**: widget set library
- **Edje**: complex graphical design & layout library
- **Ecore**: main loop, timing, event, connection, IPC, Jobs, Text encoding, Xlib wrapper...
- **Evas**: a highly optimized canvas and rendering library
- **Eina**: providing data structures (list, array, hash, tree, shared string...)
- **Embryo**: tiny interpreter for small program compiled by embryo_cc
- **Emotion**: a media object lib to play audio/video
- **Ethumb**: thumbnail generation lib.
- **Eet**: data encoding and decoding to and from files or memory
- **E_dbus**: dbus(msg bus system) wrapper
- **Efreet**: helper lib. for Freedesktop.org standards
Immediate vs. Retained mode graphics sys. (1/2)

• Immediate mode graphics system (GTK+, GDI, GDI+)
  – Most commonly used drawing mechanism in toolkit library
  – Application is responsible for repainting the portion of the client area that is invalidated
  – Issuing drawing commands and having them processed by the display system immediately; after the draw is done, it appears in the destination
Immediate vs. Retained mode graphics sys. (1/2)

- Retained mode graphics system (Evas, Scene graph)
  - Application defines a set of drawing objects with their properties
  - The graphics system is responsible for responding to all repaint requests for rendering the application objects.
  - Application can modify or create objects at run time.
  - Drawing information is always persisted in a serialized state by the application
  - Rendering responsibility left to the system
EFL Application on X Window System

Diagram showing the components involved in an EFL application:
- EFL application
- Elementary
- Emotion
- Efreet
- Ethumb
- Ecore
- Evas
- Efl fopen
- X Protocol
- X server
- Embryo
- Eet
- freetype2
- fontconfig
- xlib
- Eina
- E_dbus
- process library
EFL back-ends (SW, OpenGL ES)

- Elementary
- Edje
- Evas
- Ecore

X Server

Frame buffer

Update window areas

EFL Software Drawing

- Xlib
- X shared Mem.

X Server

Back

Front

Double buffer

EGL

OpenGL ES

OpenGL ES H/W

controlled by

xlib

X Server

Update window areas

Frame buffer

EFL OpenGL ES Backend Drawing
Advanced Features (1/2)

- Graphics rendering path for 3D scene on an Evas object
Advanced Features (2/2)

• Graphics rendering path for Video decoding on an Evas object
Input Service Framework (ISF)
Input Service Framework

\[
\text{ISF} = \boxed{\text{SCIM v1.4.7}} + \boxed{\text{Mobile Requirements}}
\]

- Based on SCIM-1.4.7 (Smart Common Input Method)
  - Not enough to satisfy mobile requirements because it was developed for Desktop hardware keyboard.
  - Not support touch-based Soft-keyboard
  - No interactive interfaces between IM engine and Soft-keyboard
  - Insufficient interface between application and Soft-Keyboard (e.g. Changing layout)

- Enhancement
  - Support interfaces between application and Soft-keyboard for mobile environment
  - Optimized for soft-keyboard (e.g. candidate word window)
**Input Service Framework (ISF)**

- **Application**
  - Editable Widget
  - ECoreIMFContext
  - ECoreIMFContextISF

- **Input Service Manager (ISF)**
  - Panel
  - Config
  - Input Service Manager

- **Soft-Keyboard**
  - A
  - B

- **IPC**

**Legend**
- Application
- Framework
- Daemon
- Plug-in Module
- IPC Flow
- Data Flow
- Process
- Plug-in

**Panel**
1. Adaption layer for various Applications
2. Provides the visual feedback to the user, such as candidates, pre-edit word, etc.

**Input Service Manager**
1. Registration/Load/Unload/Running/Pause/Resume mutually exclusive for ISEs
2. Interfaces among Applications/ISE
3. Offer specific interface with highly modular for each ISE can be plugged in
Voice Framework
Speech-to-Text (STT)

- Client/Server model
- Components
  - STT Daemon
    - Running in background for STT service
    - Recording sound (voice)
    - Recognizing the speech with STT engine
    - Returning the text recognized to the client
  - STT Client lib
    - Providing STT APIs to application
    - Requesting STT service to the STT Daemon
    - Returning the text to the application
Text-to-Speech (TTS)

- Client/Server Model
- Components
  - TTS Daemon
    - Running in background for TTS service
    - Playing the input text from the client
  - TTS Client lib
    - Providing TTS APIs to application
    - Requesting TTS service to the TTS Daemon with text input from the application

TTS state diagram
Elementary Widgets
Elementary Widgets

- Elementary is a simple widget toolkit set
  - Fast, Finger friendly, Scalable, Theme changeable, ...

- Widgets
  - Containers - Box, Conformant, Grid, Layout, Panes, Scroller, Table, ...
  - Non-Containers - Bg, Button, Check, Entry, Label, List, Icon, Win, ...

A single line entry with some

element  
button  
icon, image

naviframe  
genlist
Elementary Widgets (desktop looks)

- clock
- segment control
- flipselector
- slider
- check
- actionslider
- frame
- notify
- radio
- photo, photocam
- menu
- colorselector
- index
- map
- separator
- bg
- Label
- bubble
- diskselector
- fileselector
- glview
- panel
- ctxpopup
- calendar
- toggle
- multibuttonentry
- toolbar
References

• Tizen Community
  – [https://www.tizen.org/community](https://www.tizen.org/community)
  – IRC
    • Primary Channel
      #tizen at irc.freenode.net: 24/7 open discussion.
    • #tizen IRC Logs
      [https://www.tizen.org/irclogs/](https://www.tizen.org/irclogs/)
  – Mailing Lists
    • **General**: general@lists.tizen.org: General questions about the project, the community, or events.
    • **Application-dev**: application-dev@lists.tizen.org: Questions and feedback related to application development.
    • **Product-dev**: product-dev@lists.tizen.org: Development of the Tizen distribution itself.
    • **IVI**: ivi@lists.tizen.org: Discussions and questions about the In-Vehicle Infotainment (IVI) project for Tizen.
    • **Announce**: announce@lists.tizen.org: This is a low volume list containing only announcements about the Tizen project.

• Enlightenment Community (EFL)
  – [http://www.enlightenment.org](http://www.enlightenment.org)
  – IRC: #e at irc.freenode.net: 6667
Thank You.