A Case Study of Mobile Application Development

Wei Dong
Samsung Electronics
Content

- Tizen Application Development
- Practices of Tizen Application Development
  - Performance optimization
  - Memory usage
  - Database usage
Tizen Application Development
Application Development - Start from Here

- Platform
- Application Framework
- UI
- Functionalities
EFL Architecture

Applications, Libraries, Services

Elementary

Edje
Efreet Eio Eeze Ethumb Eldbus Emotion

Embryo
Eina

Ecore

Evas
Eet

BASE OPERATING SYSTEM / DISPLAY ETC.
Case Study: Contact Application
Application Lifecycle

- **Ready**
  - app_create_cb()
  - app_resume_cb()

- **Created**
  - app_control_cb()
  - app_resume_cb()
  - app_terminate_cb()

- **Running**
  - app_control_cb()
  - app_pause_cb()
  - app_resume_cb()
  - app_control_cb()

- **Paused**
  - app_control_cb()

- **Terminated**
  - app_terminate_cb()
Sample Code of main()

```c
int main(int argc, char *argv[])
{
    struct appdata ad;
    app_event_callback_s event_callbacks;
    event_callbacks.create = ct_create;
    event_callbacks.terminate = ct_terminate;
    event_callbacks.pause = ct_pause;
    event_callbacks.resume = ct_resume;
    event_callbacks.app_control = ct_app_control;
    ...
    return app_efl_main(&argc, &argv, &event_callbacks, &ad);
}
```
UI Analysis

- Layout
- Widget selection
Edje

- A complex graphical design and layout library
- Create visual layouts from compiled EDC script
- Support relative and absolute positioning

```plaintext
collections { - list the groups.
    group { - the list of parts and programs that compose a given edje object.
        images { } – list each image file that will be used in the edc.
        parts {
            part { - the most basic design elements of the group.
                (i.e. RECT, TEXT, IMAGE, SWALLOW, TEXTBLOCK, GROUP, BOX, TABLE, EXTERNAL)
                }
        }
    }
    programs {
        program { - manipulates the different interface elements and serve as a bridge between
                     the application and interface.
        }
    }
}
```
Scalability

• Relative position
Sample Code of UI Layout

group {
    name: "item";
    parts {
        part {
            name: "icon";
            type: SWALLOW;
            description {
                state: "default" 0.0;
                min: 72 0;
                align: 0.0 0.5;
                fixed: 1 0;
                rel1 { relative: 0.0 0.0 ;}
                rel2 { relative: 0.0 1.0 ;}
            }
        }
    }
    part {
        name: "text";
        type: TEXT;
        description {
            state: "default" 0.0;
            rel1 { relative: 1.0 0.0 ; to: "icon";}
            rel2 { relative: 1.0 1.0 ;}
        }
    }
    ...
}

...
Application View

- View control with naviframe
Building Application

- **Separation of layout and logic**
  - A graphical part: GUI layout binary (edj)
  - A functionality: executable logic binary (C)
Practices of Performance Optimization
Launching Performance Analysis

- **Synchronized procedure**
  - Poor performance for mass data

<table>
<thead>
<tr>
<th>App</th>
<th>Condition</th>
<th>Launching Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>10000 records</td>
<td>2.27s</td>
</tr>
</tbody>
</table>

Create main window

Connect DB service

Create list view

Get all records from DB

Show list view
Processing in Parallel

- Multi-threads processing
  - Retrieving data asynchronously

<table>
<thead>
<tr>
<th>App</th>
<th>Condition</th>
<th>Launching Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>10000 records</td>
<td>1.53s</td>
</tr>
</tbody>
</table>

Create main window
Connect DB service
Create list view
Get all records from DB
Show list view
Update list
Further Optimization

- **Splitting data loading**
  - Retrieving necessary data for display
  - Loading rest data in idle state

<table>
<thead>
<tr>
<th>App</th>
<th>Condition</th>
<th>Launching Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>10000 records</td>
<td>1.04s</td>
</tr>
</tbody>
</table>
List Items Loading

- Create list item with `elm_genlist_homogeneous_set()`

```c
Evas_Object *ct_create_genlist(...) {
    ...
    genlist = elm_genlist_add (parent);
    elm_genlist_homogeneous_set(genlist, EINA_TRUE);
    ...
    elm_genlist_item_append(...);
    ...
}
```

The height of list items should be same
Practices of Memory Usage
Memory Tool in Tizen

- **Valgrind**
  - For memory debugging, memory leak detection, and profiling

```c
==2741== Invalid write of size 4
==2741== at 0x57700B0: ct_input_push_navi_content (ct-input-view.c:7694)
==2741== by 0x5752A29: ct_input_launch_preloaded_view (ct-detail-main.c:833)
==2741== by 0x4953579: ct_list_create_contact_btn_clicked_cb (ct-list-contact-view.c:539)
==2741== by 0x4D2E83F: evas_object_smart_callback_call (in /usr/lib/libevas.so.1.7.99)
==2741== Address 0x0 is not stack'd, malloc'd or (recently) free'd
```

- Memory leak

```c
==11097== 1,024 bytes in 1 blocks are definitely lost in loss record 144 of 196
==11097== at 0x4834FBC: malloc (vg_replace_malloc.c:291)
```
Widget Object Release

• Lifecycle of widget object
User Data Release

- Event in object destroy
  - EVAS_CALLBACK_DEL

Entry destroy procedure
Practices of Database Usage
Database Engine in Tizen

• **SQLite 3**
  - A popular choice as embedded database
  - No wrapper in Tizen platform

• **Practices of SQLite3**
  - Concurrency control
    - Use mutex in application side
  - API usage in application
    - sqlite3_exec()
    - sqlite3_prepare_v2() & sqlite3_step()
Media Content Framework

- CRUD media information from/to database
- Managed content: image/video/audio files
Case Study: Music Player with Media Content

Get media info

... /* initialize */
media_content_connect(...);
/* create filter */
filter = media_filter_create(...);
/* get audio file count */
media_info_get_media_count_from_db(...);
/* set filter property (SQL prepare) */
media_filter_set_order(...);
media_filter_set_condition(...);
media_filter_set_offset(...);
/* query audio file list from media content */
media_info_foreach_media_from_db(...);
... /* monitor audio file update */
media_content_set_db_updated_cb(...);
... /* Finalize */
media_content_disconnect(...);

Get Metadata

... /* initialize */
media_content_connect(...);
media_info_get_media_from_db(...);
media_info_get_audio(...);
audio_meta_get_sample_rate(...);
audio_meta_get_duration(...);
...
/* Finalize */
media_content_disconnect(...);