SOSCON
삼성 오픈소스 컨퍼런스
SAMSUNG OPEN SOURCE CONFERENCE
OPEN YOUR UNIVERSE WITH SOSCON

EFL Circular UI-Components in GearS2

Samsung Electronics
woochan Lee

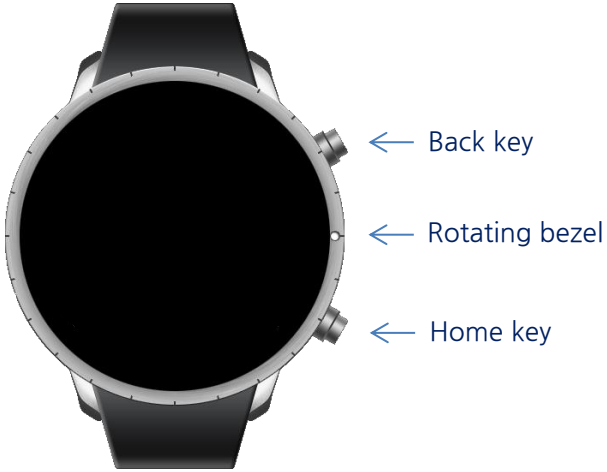
10/28/2015

Overview

1. Prerequisite
2. Basic operating principle
3. Introducing circular UI-Components

#include <efl_extension.h>

- efl_extension module is for customizing or utilizing EFL.
- Circular UI-components are only for Tizen_2.3.1 wearable platform
- EFL Extension API functions and data types are defined in the <efl_extension.h>



- There are two ways to receive the rotary events

  - Using rotary event handler
    Eina_Bool eext_rotary_event_handler_add(Eext_Rotary_Handler_Cb func, void *data);

  - Using rotary object event callback
    Eina_Bool eext_rotary_object_event_callback_add(Evas_Object *obj, Eext_Rotary_Event_Cb func, void *data);

  Data has direction and time_stamp.

More information about rotary event

Add circle surface

- What is circle surface?
  - Manages and renders circle objects. image object for cairo.

  Multiple circle objects can be connected to one circle surface as candidates of an object to be rendered.
  When one of circle objects is set visible, the surface renders the circle object and hides the others

- There are 3 APIs to add circle surface.

  - eext_circle_surface_conformant_add()
  - eext_circle_surface_layout_add()
  - eext_circle_surface_naviframe_add()

* Use 3rd API for surface zoom in/out effect when view show/hide

Ex:

```
Eext_Circle_Surface *surface;
Evas_Object *conformant;

conformant = elm_conformant_add(parent);
surface = eext_circle_surface_conformant_add(conformant);
```

eext_circle_XXX APIs

Set/get given circle object's property.

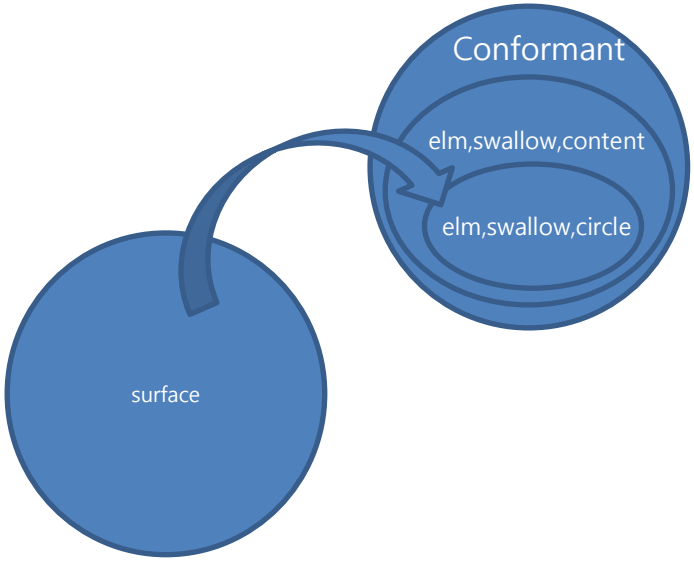
eext_circle_object_value_min_max_set()
eext_circle_object_value_min_max_get()
eext_circle_object_value_set()
eext_circle_object_value_get()
eext_circle_object_angle_min_max_set()
eext_circle_object_angle_min_max_get()
eext_circle_object_angle_offset_set()
eext_circle_object_angle_offset_get()
eext_circle_object_angle_set()
eext_circle_object_angle_get()
eext_circle_object_line_width_set()
eext_circle_object_line_width_get()
eext_circle_object_radius_set()
eext_circle_object_radius_get()
eext_circle_object_color_set()
eext_circle_object_color_get()
eext_circle_object_disabled_set()
eext_circle_object_disabled_get()

eext_circle_object_item_value_min_max_set()
eext_circle_object_item_value_min_max_get()
eext_circle_object_item_value_set()
eext_circle_object_item_value_get()
eext_circle_object_item_angle_min_max_set()
eext_circle_object_item_angle_min_max_get()
eext_circle_object_item_angle_offset_set()
eext_circle_object_item_angle_offset_get()
eext_circle_object_item_angle_set()
eext_circle_object_item_angle_get()
eext_circle_object_item_line_width_set()
eext_circle_object_item_line_width_get()
eext_circle_object_item_radius_set()
eext_circle_object_item_radius_get()
eext_circle_object_item_color_set()
eext_circle_object_item_color_get()

Basic operating principle

Surface in Conformant, Layout

- Style set as “circle” to given component
- Make elm_image and add callbacks (show, hide, del, resize)
- Put the image object in given component’s swallow part(elm,swallow,circle)
- Image size and fill set when resize callback function called.
- Cairo init using elm_image’s image object.
- Call internal function for add ecore_job
- Internal render function call cairo function for draw arc, text



Surface in Naviframe

- Add callback to naviframe’s “item,pushed,internal” signal
- Do the same things as in Conformant, Layout

Private Surface

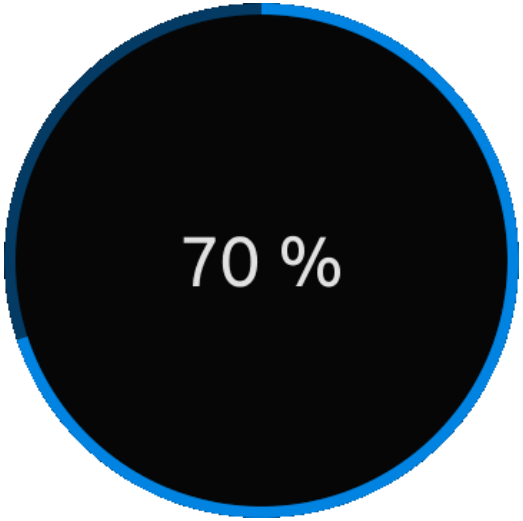
- A surface will be created internally when user calls eext_circle_object_XXX_add() with NULL value for second parameter

* It is for drawing multiple circular ui-components at the same time.
User does not have to add surface for simple circular view.

Introducing circular UI-Components

Circle Progressbar

- Circle progressbar aims to show the progress status of a given task with circular design.
- Visualizes progress status within a range.



Creating a Circle Progressbar

```
Evas_Object *circle_progressbar;

circle_progressbar = eext_circle_object_progressbar_add(parent, surface);
```

Configuring the Circle Pregressbar

- eext_circle_object_value_min_max_set(circle_progressbar, 0.0, 100.0);
- eext_circle_object_value_set(circle_progressbar, 3.0);

Using the Circle Object Property

- Circle Progressbar has the following item

  - default : Default circle item. It draws progress bar.
  - bg : Progress bar background circle item.

Introducing circular UI-Components

Circle Slider

- Circle slider changes corresponding to rotary events

Creating a Circle Slider

```
Evas_Object* slider;

circle_slider = eext_circle_object_slider_add(parent, surface);
eext_rotary_object_event_activated_set(circle_slider, EINA_TRUE);
```

Configuring the Circle Slider

- eext_circle_object_slider_step_set(circle_slider, 0.5);

Using the Circle Object Property

- Circle Slider has the following item

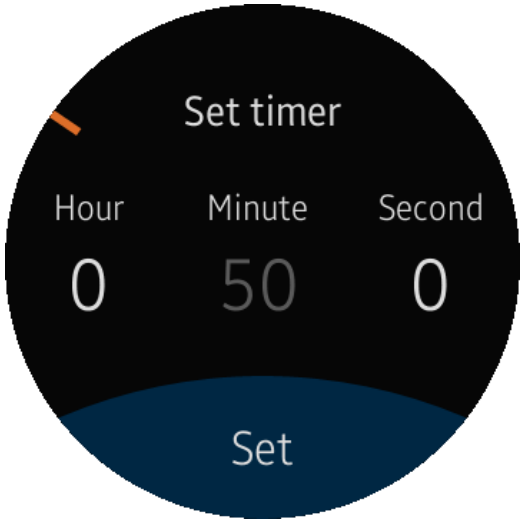
- default : Default circle item, It draws slider bar.
- bg : Background circle item.

Using Circle Slider Callback

- “value,changed” : The rotary event changes the circle slider value.

```
evas_object_smart_callback_add(slider, "value,changed", _value_changed_cb,  data);
```

Introducing circular UI-Components



Circle Spinner

- Circle spinner changes corresponding to rotary events.
- Circle spinner not only extends UI feature of elm_spinner, but also replaces functionalities of elm_spinner in a circular design.

Creating a Circle Spinner

```
Evas_Object* spinner;
Evas_Object* circle_spinner;

spinner = elm_spinner_add(parent);
circle_spinner = eext_circle_object_spinner_add(spinner, surface);
eext_rotary_object_event_activated_set(circle_spinner, EINA_TRUE);
```

Configuring the Circle Spinner

- eext_circle_object_spinner_angle_set(circle_spinner, 2.0);

  - (360/ max - min) * step -> 2.0 * step

Using the Circle Object Property

- Circle spinner has the following item

  - default : Default circle item, It draws spinner picker

Introducing circular UI-Components

Circle Datetime

- Provides circular UI which is proper to each date(or time) field of elementary datetime.
- Operates with rotary events to change its value.

Creating a Circle Datetime

```
Evas_Object* datetime;
Evas_Object* circle_datetime;

datetime = elm_datetime_add(parent);
circle_datetime = eext_circle_object_datetime_add(datetime, surface);
eext_rotary_object_event_activated_set(circle_datetime, EINA_TRUE);
```

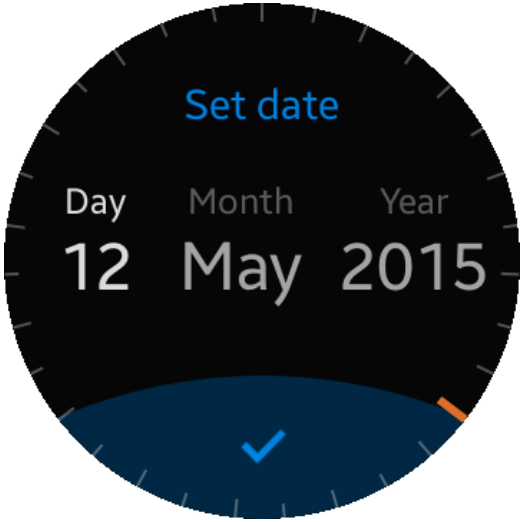
Using the Circle Object Property

- Circle Datetime has the following item

- default : Default circle item, It draws a marker

Note

- Each field has different bg image
  - bg image buffer memcpy to surface image object before render cairo arc



Introducing circular UI-Components

Circle Scroller

- Circle scroller changes corresponding to rotary events
- It shows whole scrollable area with circular scroll bar
- Only for draw circular scroll bar with edge of side circular screen

Creating a Circle Scroller

```
Evas_Object *scroller;
Evas_Object *circle_scroller;

scroller = elm_scroller_add(parent);
circle_scroller = eext_circle_object_scroller_add(scroller, surface);
eext_rotary_object_event_activated_set(circle_datetime, EINA_TRUE);
```

Configuring Circle Scroller

- eext_circle_object_scroller_policy_set(circle_scroller, ELM_SCROLLER_POLICY_OFF, ELM_SCROLLER_POLICY_ON)

- ELM_SCROLLER_POLICY_AUTO : scrollbar is made visible if it is need
- ELM_SCROLLER_POLICY_ON : turns it on all the time
- ELM_SCROLLER_POLICY_OFF : always keeps it off

Using the Circle Object Property

- Circle GenIsit has the following item

- default: Default circle item. It draws vertical scroll bar
- vertical,scroll,bg: Vertical scroll background circle item
- horizontal,scroll,bar : Horizontal scroll bar circle item
- horizontal,scroll,bg : Horizontal scroll background circle item

Introducing circular UI-Components

Circle Genlist

- Visualize and utilize the scroll effect for elm_genlist.
- Provides scrollbar with circular movement.
- Operated with rotary events to move to the next or previous item.

Creating a Circle Genlist

```
Evas_Object* genlist;
Evas_Object* circle_genlist;

genlist = elm_genlist_add(parent);
circle_genlist = eext_circle_object_genlist_add(genlist, surface);
eext_rotary_object_event_activated_set(circle_datetime, EINA_TRUE);
```

Configuring Circle Genlist

- eext_circle_object_genlist_scroller_policy_set(circle_genlist, ELM_SCROLLER_POLICY_OFF, ELM_SCROLLER_POLICY_ON)

- ELM_SCROLLER_POLICY_AUTO : scrollbar is made visible if it is need
- ELM_SCROLLER_POLICY_ON : turns it on all the time
- ELM_SCROLLER_POLICY_OFF : always keeps it off

Using the Circle Object Property

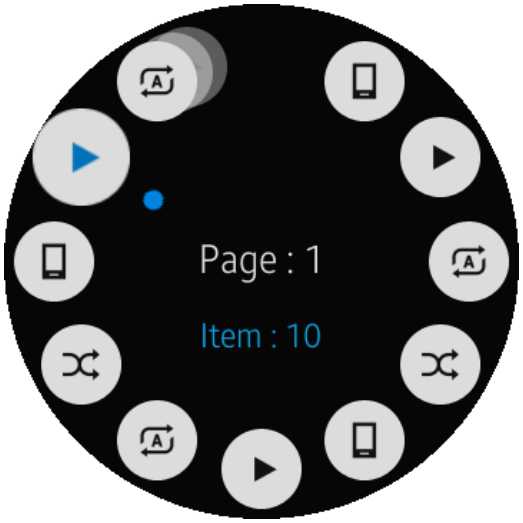
- Circle Genlsit has the following item

- default: Default circle item. It draws vertical scroll bar.
- vertical,scroll,bg: Vertical scroll background circle item.

Introducing circular UI-Components

Rotary Selector

- It is composed of Selector and multiple items which surround the Selector.
- Rotary Selector can select an item or move to next/prev page by rotary event.



Creating a Rotary Selector

```
Evas_Object *rotary_selector;

Rotary_selector = eext_rotary_selector_add(parent);
eext_rotary_object_event_activated_set(rotary_selector, EINA_TRUE);
```

Adding an Rotary Selector Item

```
/* Append item */
item = eext_rotary_selector_item_append(rotary_selector);

/* Set item icon */
image = elm_image_add(rotary_selector);
elm_image_file_set(image, "music_controller_btn_play.png", NULL);

eext_rotary_selector_item_part_content_set(item, "item,icon",
EEXT_ROTARY_SELECTOR_ITEM_STATE_NORMAL, image);
```

Using the Rotary Selector Callbacks

- item,selected: The user selected the item
- item,clicked: The user clicked the item

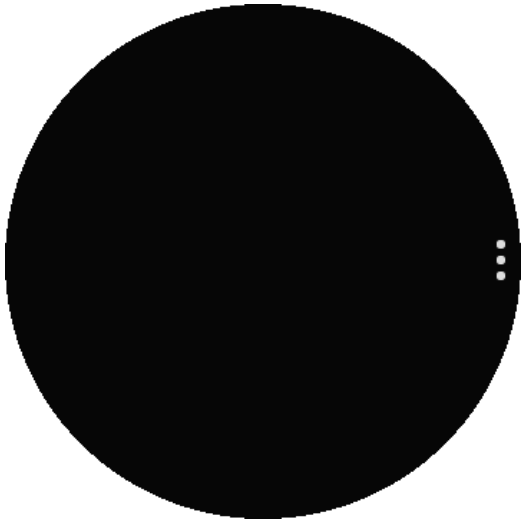
```
evas_object_smart_callback_add(rotary_selector, "item,clicked", item_clicked_cb, data);
```

More information about Rotary Selector

Introducing circular UI-Components

More Option

- More option is the widget combining elm_panel and rotary_selector.



Creating a More Option

```
Evas_Object *more_option ;
more_option = eext_more_option_add(parent);
//It doesn' t need to call eext_rotary_object_event_activated_set()
```

Configuring More Option

- eext_more_option_direction_set(more_option, EEXT_MORE_OPTION_DIRECTION_T);

  - EEXT_MORE_OPTION_DIRECTION_TOP
  - EEXT_MORE_OPTION_DIRECTION_BOTTOM
  - EEXT_MORE_OPTION_DIRECTION_LEFT
  - EEXT_MORE_OPTION_DIRECTION_RIGHT

Using the More Option Callbacks

- item,selected: The user selects the item.
- item,clicked: The user selects the already selected item again or selects a selector.
- more,option,opened: A layout in which a rotary_selector is included is seen.
- more,option,closed: A layout in which a rotary_selector is included isn`t seen.

```
evas_object_smart_callback_add(more_option, “more,option,opened”, _opened_cb, data);
```

THANK YOU!