Agenda

- Overview
  - Key Navigation
  - Scrollable Grid & List
  - Demo
What is CAPH?

CAPH: a Web UI Framework
Easy and quick development of web apps for Samsung Tizen TV

- Custom made for TV
- High performance
- Rich UI components
CAPH Web UI Framework

CAPH Modules
- Key Navigation
- Scrollable Grid & List
- UI Components (Button, Dialog, Checkbox, Input, Dropdown Menu, ...)

Tizen TV SDK
- CAPH Package
- Sample & Tutorial
- API Documents
- TC & Test suite
- Web App Scaffolding

jQuery
AngularJS
Velocity.js (Fast Animation)
Lodash (Utility)
With Open Source Frameworks

- Based on popular frameworks (Presently jQuery and Angular JS)
- Support more frameworks in the future
CAPH Modules

- All modules support AngularJS and jQuery
- All modules work with Key Navigation
- All UI Components are customizable
- Overview
- **Key Navigation**
  - Scrollable Grid & List
  - Demo
Key Navigation

- Handle the focus of UI elements easily by arrow keys of remote controller
- An attribute ‘focusable’ which represents the focusable element
- Support Mouse, Keyboard as well

<div focusable></div>

Find the nearest focusable element automatically
Nearest Focusable Finder

- Find the nearest focusable element automatically

- Support custom distance calculation strategy
Properties

- Properties: depth, group, name
- By default, depth: 0, group: 'default', name: 'focusable-0'
- Can be changed as below

```html
<div focusable="{depth: 1, group: 'test', name: 'focus1'}"></div>
```

or

```html
<div focusable data-focusable-depth="1"
    data-focusable-group="test"
    data-focusable-name="focus1">""</div>
```
Property - depth

- Used when moving focus to another focusable element
- Focus only changed between the same depth

- Change the depth by calling focus controller's 'setDepth' API
Property - group

- Separate focusable area semantically
- Focus can be changed if each depth is same

- Change the group easily by concatenating 'group:' prefix and option value

```html
div focusable data-focusable-group="test1" data-focusable-next-focus-down="group:test2"></div>
<dir focusable data-focusable-group="test2" data-focusable-next-focus-up="group:test1"></div>
```

You can manage each group's previous focusable element history by calling focus controller’s 'setGroup' API
Property - name

- Used when setting the next focus to the specific element
- Each name should be an unique value

```html
<div focusable="{name: 'focus1', nextFocus: {right: 'focus3'}}"></div>
<div focusable="{name: 'focus2'}}"></div>
<div focusable="{name: 'focus3'}}"></div>
```
Event Handler

- on-focused
- on-blurred
- on-selected

```html
<div focusable on-focused="focus($event, $originalEvent)"
  on-blurred="blur($event, $originalEvent)"
  on-selected="select($event, $originalEvent)"></div>
```

or

```javascript
$('[focusable]').on('focused blurred selected', function(event) {
  // your code here
});
```
Focus Controller Provider

- initial depth
- initial group of each depth
- key map reset
- keydown handler (before/after)
- disabled element focusing
- Overview
- Key Navigation
- **Scrollable Grid & List**
- Demo
High performance scrollable component.
Setting options & template view
Basic usage - AngularJS

- Create by caphList directive, add list data on the scope

```
<cap-list items="item in items" direction="vertical">
  <div class="item" focusable>{{$index}}</div>
</cap-list>
```

**items** - variable in expression
- variable: the user defined loop variable
- expression: a scope expression giving the collection to enumerate

```
{{$index}}
```
- $first $last $even $odd are also available

**direction**
- By default, the scroll direction is 'horizontal'
Basic usage - jQuery

- Define a template view to decorate the items in the list

```html
<!-- container element -->
<div id="list1"></div>

<!-- template view -->
<script id="template1" type="text/template">
  <div class="item" focusable><%= index %></div>
</script>
```

- Create a 'caphList' using the prepared resources

```javascript
$('#list1').caphList({
  items: items,
  template: 'template1'
});
```
<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resize</td>
<td>If you change the list size manually, you should send this event. The window's resize event is processed automatically.</td>
</tr>
<tr>
<td>reload</td>
<td>Update the entire item views. By default, when appending new data to collection or removing the existing data from the tail of collection, the visible item views will be updated only if necessary. So if you change the all or some(excluding default case) data of collection, you should call this method to update item views.</td>
</tr>
<tr>
<td>moveup</td>
<td>Move to the up item.</td>
</tr>
<tr>
<td>moveright</td>
<td>Move to the right item.</td>
</tr>
<tr>
<td>moveleft</td>
<td>Move to the left item.</td>
</tr>
<tr>
<td>movedown</td>
<td>Move to the down item.</td>
</tr>
</tbody>
</table>
Properties

Object[] items
string direction
string container-class
string wrapper-class
string duration
string ease
number delay
boolean loop
number initial-item-index
number mouse-scroll-area-size

Function on-decorate-item-view
Function on-focus-item-view
Function on-reach-start
Function on-reach-end
Function on-scroll-start
Function on-scroll-finish
- Overview
- Key Navigation
- Scrollable Grid & List
- Demo