Manage Scenes

Published 2014-10-27 | (Compatible with SDK 2.5,3.5,4.5,5.0,5.1 and 2011,2012,2013,2014 models)

> Basics of building the Smart TV applications using scenes mechanism.

Contents

Scene HTML File

Scene CSS File

Scene JavaScript File

**** This class will not be supported in 2015.**

*All functionalities of AppsFramework are more improved, integrating with CAPH. Therefore Appsframework is not supported since 2015 Smart TV. To use functionalities of Appsframework, refer to **here**.*

A scene is a kind of layer - it is what the TV user sees and interacts with to perform a single task or a group of tasks.

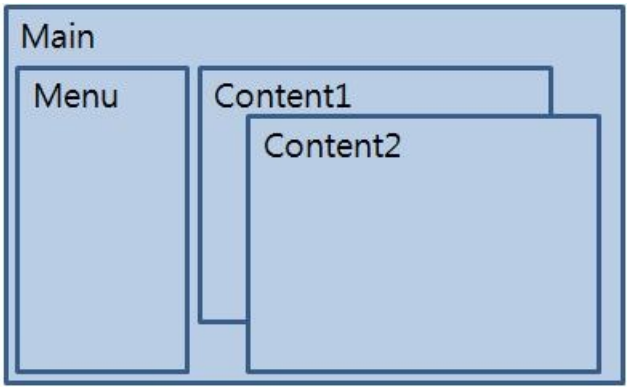

Figure: AppsFramework scene architecture

Each scene has show/hidden and focused/blurred states. A focused scene controls the key input. You develop scene-based applications by managing scenes and scene states.

Each scene has its own set of files:

- HTML
- CSS
- JavaScript

In order to add use scene-based architecture in your application, you must add the scenes to your application's project files.

Samsung Smart TV provides the following tutorial to demonstrate different aspects of working with Scenes: Creating a News Application.

Scene HTML File

The Scene HTML file name must be in the form like this: [SceneID].html. It is located in the app/html directory. Let's assume we are preparing the Main scene so our HTML file will be named Main.html. Add HTML content to the body by wrapping the Scene[SceneID] ID's div tag as shown below.

```
<div id="SceneMain" style="visibility: visible;">
  <!-- Scene HTML contents -->
</div>
```

Scene CSS File

The CSS file used for the scene HTML setting, must have the name formed like this: [SceneID].css so in our case it is Main.css. It is located in the app/stylesheets directory. All the scene css files are loaded when the application starts.

> Important
>
> Note that if the scene is not fullscreen style, the position (left, top) and size (width, height) could change.

```
#SceneMain {
  position: absolute;
  left: 0px;
  top: 0px;
  width: 960px;
  height: 540px;
  background-color: #fff;
}
```

Scene JavaScript File

The scene JavaScript file name must be in the form like this: [SceneID].js so in our case it is Main.js. It is located in the app/scenes directory. This file includes the definition of the Scene[SceneID] class (SceneMain in our case). When the application starts an instance of this class is created and managed by the Scene Manager API (sf.scene) and its methods.

The example below shows a basic scene class.

```
function SceneMain(options) {
}

SceneMain.prototype.initialize = function () {
    alert("SceneMain.initialize()");
    // this function will be called only once, when the scene manager shows this scene for the first time
    // initialize the scene controls and styles, and initialize your variables here
    // scene HTML and CSS will be loaded before this function is called
}

SceneMain.prototype.handleShow = function (options) {
    alert("SceneMain.handleShow()");
    // this function will be called when the scene manager shows this scene
}

SceneMain.prototype.handleHide = function (options) {
    alert("SceneMain.handleHide()");
    // this function will be called when the scene manager hides this scene
}

SceneMain.prototype.handleFocus = function (options) {
    alert("SceneMain.handleFocus()");
    // this function will be called when the scene manager sets the focus on this scene
}

SceneMain.prototype.handleBlur = function () {
    alert("SceneMain.handleBlur()");
    // this function will be called when the scene manager moves the focus to another scene from this scene
}

SceneMain.prototype.handleKeyDown = function (keyCode) {
    alert("SceneMain.handleKeyDown(" +  keyCode + ")");
    // TODO: write a key event handler when this scene gets the focus
    switch (keyCode) {
        case sf.key.LEFT:
            break;
        case sf.key.RIGHT:
            break;
        case sf.key.UP:
            break;
        case sf.key.DOWN:
            break;
        case sf.key.ENTER:
            break;
    }
}
```