

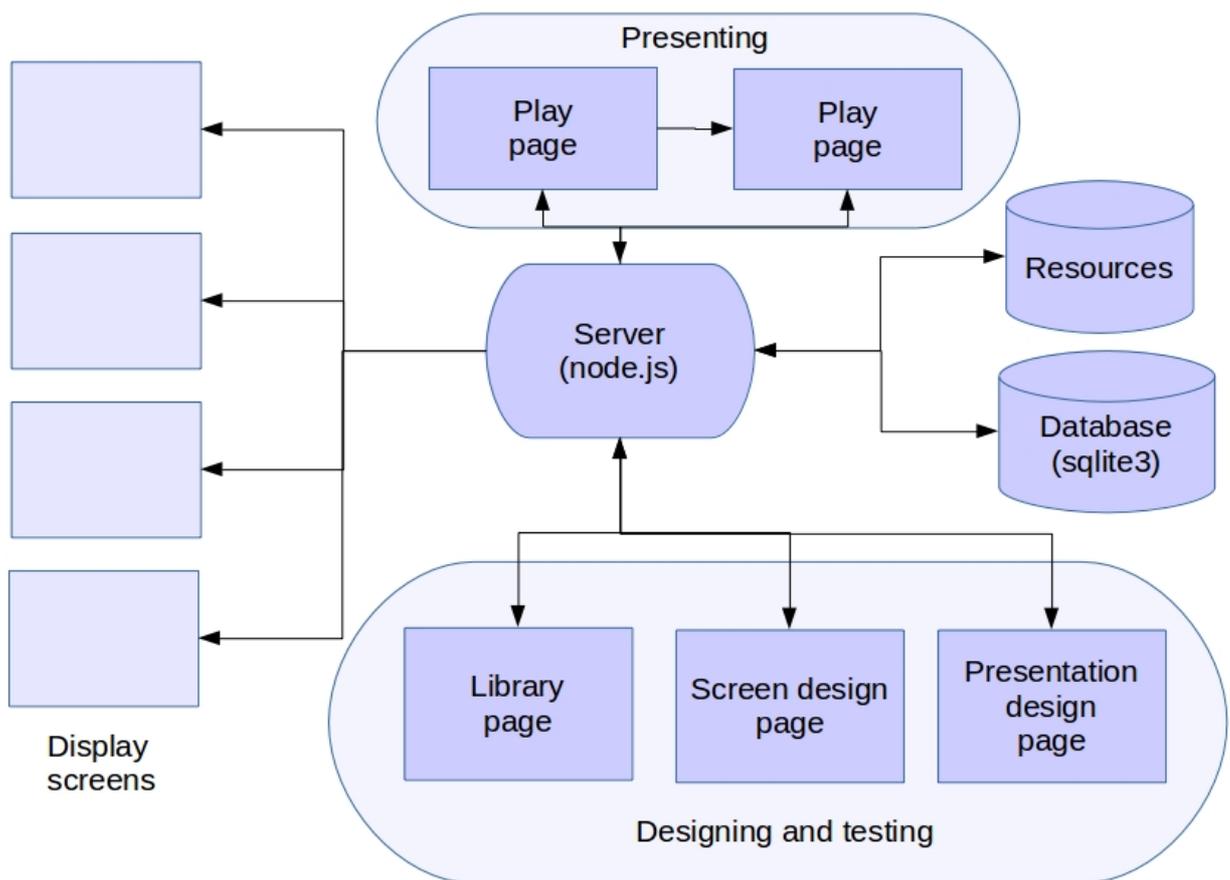
# Living Lab Screen Designer

## 1. Introduction

The Screen Designer is a browser based designer and consists of several components. The server is based on node.js and serves the pages for the library and design pages. It also saves and retrieves data to and from a sqlite3 database and the file system and handles http requests from the browser pages. The browser pages include the library page and the screen design page. The library page allows the user to upload images to a resource directory and to compose graphics objects from simple shapes, images, etc. The screen designer page enables the user to define a project with different views and view layers and to place library objects onto those layers and edit them.

## 2. How the components work together

A diagram of the various components is shown below:



The node.js server serves the pages to the browsers which connect to it via port 1337. The server also relays the transmission of content from the screen design page views to the display screens, running their own browser pages connected to the server, via the socket.io package. The server handles upload requests and stores uploaded images to a resource repository and also saves and retrieves projects and library objects to an sqlite3 database.

The library and screen design pages make use of the kineticjs HTML5 canvas framework (<http://kineticjs.com>) which provides an object oriented graphics framework for defining views and layers as well as a range of graphics primitives. In addition, jstree ([www.jstree.com](http://www.jstree.com)) is used to implement a tree structure to access and edit graphics objects in the screen designer and smartmenus ([www.smartmenus.org](http://www.smartmenus.org)) to implement dropdown menus. These make use of the jQuery library ([jquery.com](http://jquery.com)). A number of javascript modules were written to implement the HTML5 based browser pages with page styling defined in a number of css files.

### 3. Installing and running

#### Manual method:

Step 1: Create a directory for the screen design application.

Step 2: Download the node.js install file from [nodejs.org](http://nodejs.org) and follow the installation instructions for your operating system.

Step 3: Go to the directory that you created for the application and open a terminal. Install the sqlite3, formidable and socket.io modules with npm:

```
npm install formidable
```

```
npm install sqlite3
```

```
npm install --save socket.io
```

Step 4: Copy the HTML, javascript and css files into the new application directory as well as the smartmenus, jstree, images and resources folders.

Step 5: Run the server by opening a terminal in the application directory and executing 'node nodeio.js'.

Step 6: Open a browser and point it to <http://serverIP:1337/designscreen.html> for the screen design page and <http://serverIP:1337/library.html> for the library page.

#### Auto method:

For Windows there is an install package which installs the files in step 3 and 4 automatically. It also provides a batch file to start the server called noderun.bat and another batch file which opens a web page with links to the pages which form part of the package. Step 2 still has to be performed manually.

### 4. Using the library page (library.html)

The library page enables the user to upload resources and to design re-usable graphic objects which can be saved in a library database for later use in designing a screen view.

#### Resource pane:

The resource pane is on the right of the page. Images can be selected by clicking on the 'Browse' button at the top of the resource pane. Once an image has been selected, it can be uploaded to the resources directory by clicking on the 'Upload File' button.



allow the user to enter a library name for the composite object. The object is saved to the library when the 'update library' button in the property inspector is clicked.

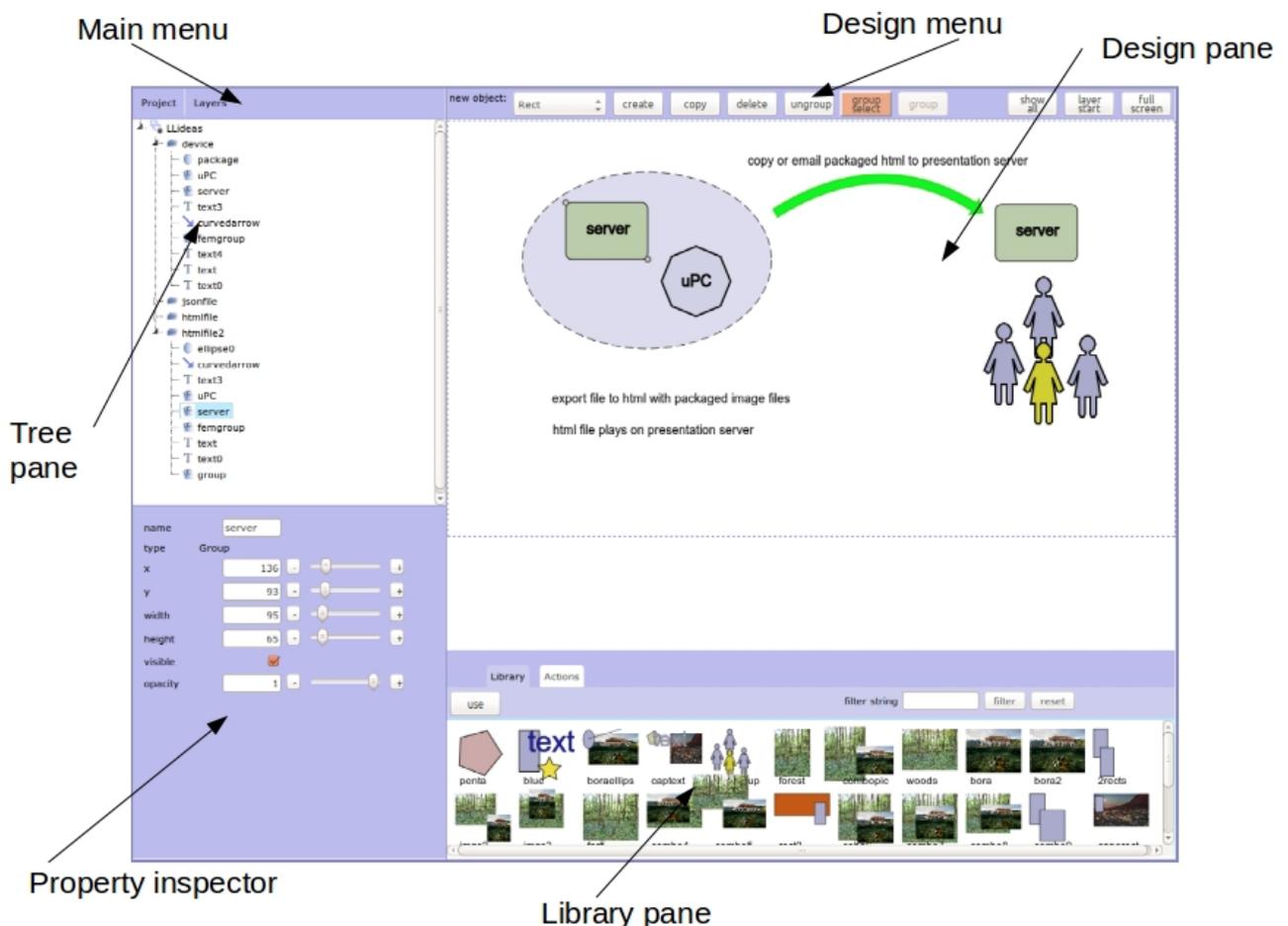
To add an image object to the design area, one has to click on the image create button in the object menu. An image icon will be displayed in the design area. The image resource can now be linked to this image object by clicking on the desired image resource while the image object is selected.

Library pane:

The library pane shows the objects saved in the object library database. These can be selected by clicking on the library object icon in the library pane. Once an object is selected, it can be added to the design area by clicking on the 'use' button. The object can also be removed from the library by clicking on the 'remove' button. Only a selection of objects can be displayed in the library pane by using the filter. By entering part of or the full name of a library object and clicking on the 'filter' button, only those objects in the library that match the filter text will be displayed in the library pane. By clicking on the 'reset' button, the pane will be refreshed to show all the objects in the library.

## 5. Using the screen layer design page (designscreen.html)

The design screen page allows the user to compose a project with different views, where each view can have several layers. Graphic objects from the library can be placed on these layers and manipulated. The design screen page consist of several menus and panes as described below.



## Main menu:

A project can be created by hovering the mouse over the 'Project' menu item and selecting 'new' from the submenu. An existing project can also be opened from or saved to the project database. When a new project is created, the project properties are displayed in the property inspector and the user can name the project and assign a creator. The project create date is automatically inserted and the last date item will be updated each time the project is saved.

When 'open' is selected, a list of projects from the database is displayed as icons in the tree pane. By hovering the mouse over a project icon, the details of that project can be viewed in the property inspector. When the user clicks on the icon, the project will be opened.

The project will be saved in the database under the name in the project name field in the property inspector when the Save option is chosen in the project drop down menu. A copy of a project can also be saved under a different name (specified by the project name field) when 'Save as new' is selected in the project drop down menu.

The 'Layers' item allows users to add and delete layers from the selected view.

## Tree pane:

The tree pane displays the structure of the project as a hierarchical tree of layers and the objects on each layer. Objects placed on the design pane are added to the selected layer in the tree. An object can be selected by either clicking on the object icon in the tree or on the object itself in the design pane. The properties of the selected object is displayed in the property inspector.

Views, layers and objects can be moved on the tree by dragging it to a new position in the tree with the mouse. In this way, an object can be moved to a different layer. The position of an object in the layer is related to its z-position on the layer. The last object in the layer list on the tree is the front object and the first object is furthest towards the back.

Several objects can be selected by holding down the ctrl button and clicking on the object in the treeview. The property inspector will only display those properties which the objects have in common. In this way the properties of several objects can be edited.

## Design pane and menu:

Objects are added to the design pane by selecting an object in the library pane and clicking on the 'use' button. The selected object's properties are displayed in the property inspector. An object's properties can be manipulated by dragging the object in the design pane, dragging the object handles or adjusting the values in the property inspector. The top left handle would normally allow an object to be resized or moved and the lower right handle will allow rotation. Objects can also be dragged by holding the mouse button down while over an object in the design pane.

A composite object from the library is a group of basic objects and will be added to the design pane as a group object. It can be disassembled into its parts by selecting the object and clicking on the 'ungroup' button. Individual objects can also be copied and deleted once they are selected.

Objects on the design pane can be grouped as a larger group object by activating the multi-select 'mselect' button. The button turns green when active. With this button active, the user can click on the objects that have to go into the new group and click on the

'group' button when done. The 'group select' button can be deactivated at any time to cancel the process.

An arrow object can be added to the design pane to show relationships between objects in the layer. This object has animation properties which will be described in the section on actions. An arrow can be added by clicking on the 'arrow' button in the design menu.

The 'full screen' button in the design menu enables the user to toggle fullscreen mode in the browser.

The 'show all' button will set the opacity of all objects on the layer to slightly transparent so that hidden and invisible objects can be viewed during the design phase.

The 'layer start' button allows the user to save the start state of all the objects on a layer.

## 6. Using the action tab on the layer design page (designscreen.html)

The designscreen also allows actions to be defined which operate on each object on a layer. By selecting the 'Actions' tab, the actions pane for each layer can be seen. The action pane will show the actions for the selected layer.

The screenshot displays the designscreen.html interface. At the top, there are buttons for 'save layer startstate', 'show all', 'layer start', and 'Full screen'. The main design area shows a diagram with a 'server' and 'uPC' connected by a green arrow labeled 'move to presentation'. Below the diagram, text reads: 'export file on uPC and moved to presentation as physical unit' and 'playscreen opens file on server'. The bottom panel is divided into 'Library' and 'Actions' tabs. The 'Actions' tab shows 'action type: appear' and buttons for 'create', 'delete', 'start', 'end', 'play', and 'play mode'. Below this are three boxes: 'actionbox', 'event1 visw0' (containing 'fadein\_package'), and 'event2 visw0' (containing 'appear\_uPC'). On the left, a property inspector shows 'descriptor: fadein\_package' and various settings like 'startstate', 'endstate', 'starttime', and 'duration'. Labels with arrows point to 'action properties', 'actionbox', 'events', and 'action controls'.

An action is created by first selecting the object on which the action has to operate. The type of action is then selected from the drop down box in the action controls and clicking on the 'create' button. The action will appear in the actionbox which acts as a holder for all actions which have not been assigned to an event. An action can be selected by clicking on it. The selected action will be highlighted and its properties will be displayed in the property inspector where properties

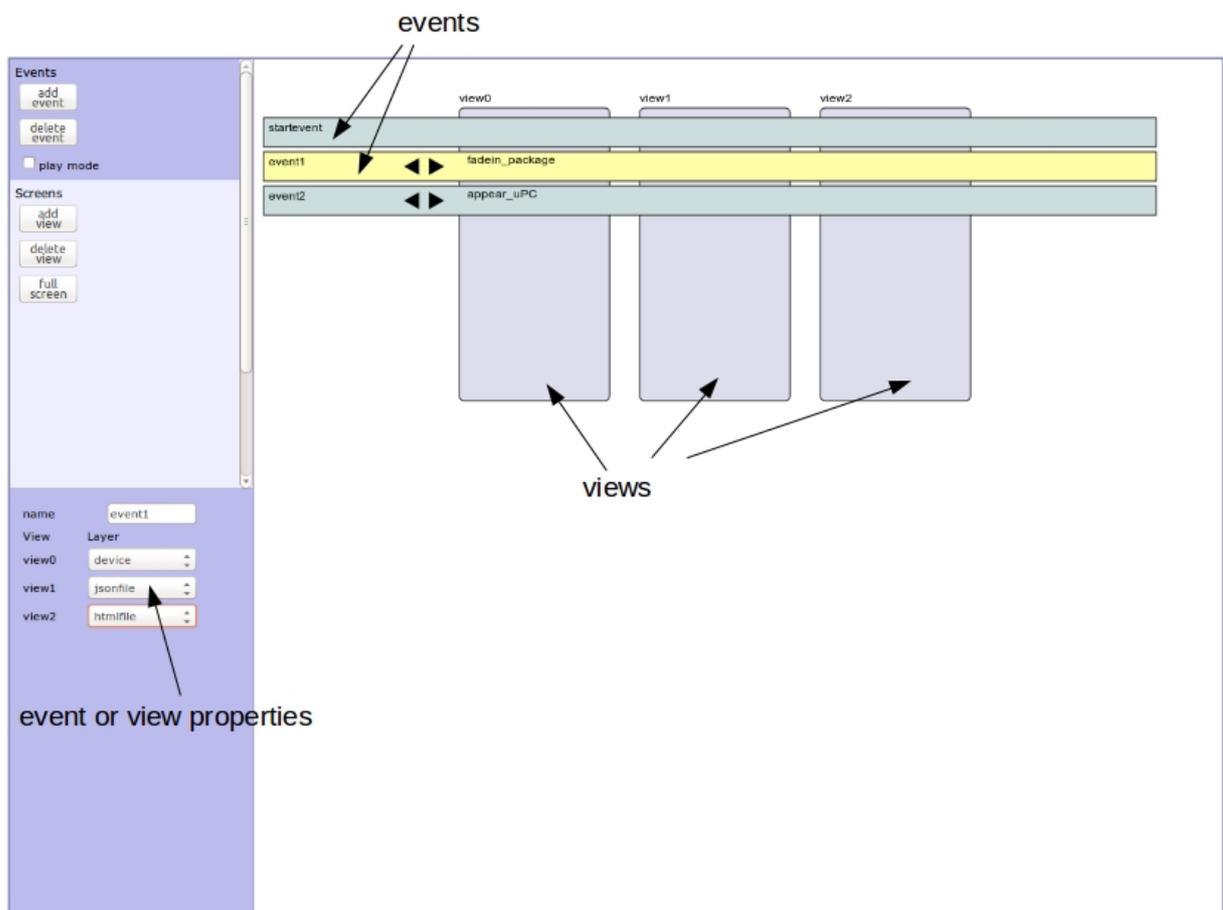
such as starttime and duration can be edited. By clicking on the 'start', 'end' or 'play' buttons in the action controls, the object property which is controlled by that action will be set to its startstate, the endstate or animated to change from the start to end state. If the the 'play mode' box is checked, the action will play when the action is clicked.

Several actions can be collected in an event box such as the actionbox. The event box can be selected by clicking inside it, but not on an action. When the event box is selected, all the actions in the box will be executed when the control buttons are clicked.

Actions can be dragged and dropped into other event boxes. The outline of an action fades when it is accepted by an event box. The next section will describe how other event boxes can be created by using the present screen page.

## 7. Using the present screen design page (presentscreen.html)

The aim of the present screen is to assign specific layers to be displayed on physical display screens and to apply actions to the objects on the displayed layer. An event is therefore defined by a layer, a view and the actions to be executed on the layer objects during the event. It serves to 'glue' the layer and its actions to the view which is associated with a physical display.



The present screen allows the user to create views and events. When the present screen is opened initially, it will only show a start event which contains the start state information that was saved by the 'layer start' button in the layer design screen. When the project on the layer design screen is renewed or re-opened, the present screen browser page has to be refreshed to enable it to load relevant information about the layers from the layer design screen.

Views can be added by clicking on the 'add view' button. A view has to be added for each physical screen. A view can be selected by clicking on it and its properties will then be displayed in the property inspector. The view can be named by editing its name property in the property inspector. The physical screen that will be accessed by this view, will then be activated by opening a browser at <http://serverIP:1337/viewname.html> where viewname is the name of the specific view.

Events are created by clicking on the add event button and can be selected by clicking on them. Once selected, an event's properties will be displayed in the property inspector. The link between a layer and a view is made by assigning a layer to a specific view in the property inspector for the event. **When a layer is assigned to a view, the intersection of that event and view will be shown in the action pane for that layer as an event box. Actions can then be dragged and dropped into the event box.** When that is done, the action name will be displayed in the event/view intersection on the present screen.

The vertical progression of events down the page describe the sequence of events during a presentation and views can be made to switch between layers and actions can be sequenced in this way. Events can be dragged and dropped in this sequence. The event sequence is also reflected in the left to right order of event boxes in the action panes of the layers. When an event is clicked or selected, it will update the display linked to the view to the startstate of that event. The two arrows on each event allow the user to play the actions or to revert to the start state of that event.. When the 'play mode' box is ticked, the actions in each event will play as soon as the event is selected. The states of the objects on each layer will be updated when the event is clicked or played.

When the project is saved, the view and event information will be saved as well and the present screen will show the saved structure after refreshing the browser page when the project is opened again.

Views and events can also be deleted by using the appropriate delete buttons on the page. When an event is deleted, the actions contained in that event will be moved back to the action box on the action panes for the different layers and the event box associated with that event, will be removed.

Once the project is ready for presentation it is necessary to package the project for transportability and speed. This is done by choosing the 'save playlist' option on the layer design screen project drop down menu. This will save the presentation package in a directory 'playlists' in a folder with the name of the project. This folder will contain the images associated with the project and an html file containing the project presentation information. The folder with the project name has to be copied to the 'playlists' folder on another Living Lab system in order to be presented there.

## **8. Using the playlist and play screen pages (playlist.html)**

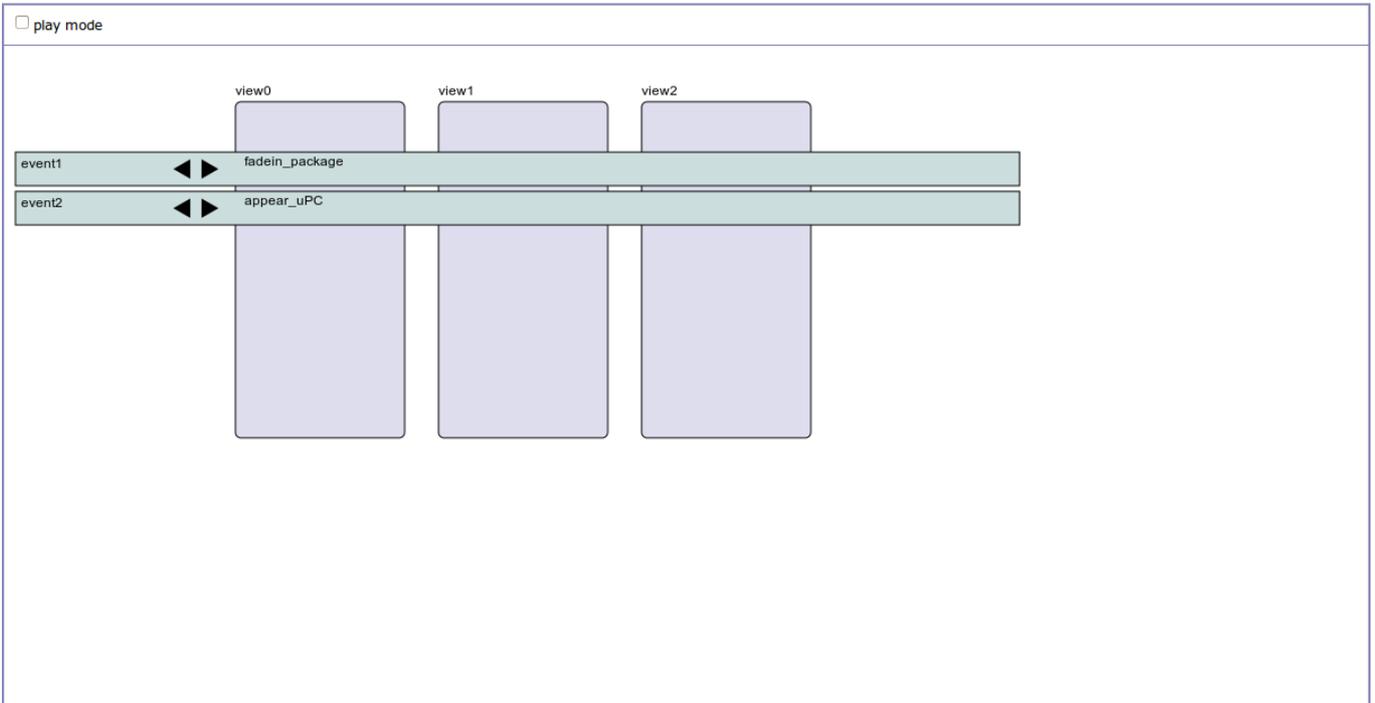
Once the project presentation folder has been copied into the 'playlists' folder of the Living lab system where it is to be presented. It can be either directly opened by opening the browser at <http://serverIP:1337/playlists/projectname/projectname.html> or more conveniently, by using the playlist utility.

When the browser is opened at <http://serverIP:1337/playlist.html>, a list of all the packaged presentation projects will be seen. By clicking on the 'open' button next to the name of the project, it will be opened in a new browser page.

## Playlists

Name	
LLIdeas	<input type="button" value="open"/>
Project123	<input type="button" value="open"/>

The presentation project page will contain all the events defined in the present screen and will also contain all the object and action information for all the layers associated with the events.



By opening browsers for the different views, the events can be displayed by simply clicking (or tapping) on the events and the arrow buttons of each event. The play mode box will allow the events to play their associated actions when clicked.