NGPS Core Documentation

1. Structure

1. Folder structure

..

scripts [ contains the core of the engine ]

plugins [ contains built-in and purchased apps ]

style [ contains global styling mechanisms, Bootstrap ]

statistics [ contains statistic information collected from the user in order to improve the experience ]

index.html [ application entry point ]

1.1 Core code structure

1.1.0 Bare core

TweenMax.min.js [ awesome 2D transforms library ]

require.js [ used for dynamically loading js files]

drivers.js [generic utility and platform specific functions]

container.js [ container code, building block upon which the system is built]

interaction.js [ extension module that adds interaction possibility to a container ]

camera.js [ extension module that turns a container into a camera]

app.js [ extension module that allows a container to host an app ]

1.1.1 Factory Wrapper

factory.js [ does initial setup, links theme with container creation, is an adapter for container & camera creation ]

init [ initial setup function, creates the factory.root object which is the root of the presetation ]

setup [ function defined by external sript for custom initial setup ]

AMS [ Anti Monotony Script dictates how every container will be styled ]

. constructors [ folder containing all the custom setup scripts ]

. themes [ folder containing all the AMS’ ]

. descriptors

containers.js [ file containing all the container styles owned by the user ]

links.js [ file containing all the link styles owned by the user ]

1.1.2 Events

gem.js [ is a generic event manager that handles event registering, dispatching, unregistering and monitoring ]

1.1.3 Saving & Loading a presentation

save.js [ handles saving the presentation in a HTML format ]

load.js [ loads the presentation from a HTML format ]

1.1.4 Applications

Every applications is identified by a unique name, for example “text\_editor” and is contained within a folder with the same name. Applications are found in the “plugins” folder:

plugins

text\_editor

main.js [ this is the application’s entry point, it is responsible for including dependencies and setting up the app]

1.1.5 CLI

cli.js [ is a command line interface that allows NGPS developers to easily test and manipulate the system ]

\*.\* Containers

\*.\* The Factory

\*.\* Apps

An apps main.js file contains all the necessary code to include dependencies, setup the app and comply with the NGPS environment. This file has a set structure so that apps can be easily loaded and managed by NGPS, therefore every app is a class ( object ) with the following structure:

var myApp = function( data )

{

this.config = { <add configuration properties here> };

this.parent = data['parent'];

this.init = function(){ } //called only one when bound to container

this.run = function(){ } //called whenever the container is triggered

this.suspend = function(){ } //called whenever the container looses focus ( or gets out of view )

this.shutdown = function(){ }//called only when app is unloaded from container

this.show = function(){ } //shows app

this.hide = function(){ } //hides app

}

//this function must be called in order to load the app into NGPS

loadAppCode("myAppName",myApp);

NGPS apps have complete control over events, therefore if an app is loaded in a container then it becomes unmovable. To solve this NGPS has a built in system that gives or takes the app control over events. By clicking once on a container with an app it will cause the app to run ( the run function is called ) and therefore all events will be diverted to the app. A round tick will be displayed in the upper left corner which if clicked will suspend the app ( the suspend function is called ) and take back event control from the app making the div movable again.