Systems Mapping Platform

Specification<https://docs.google.com/document/d/1TEn-6hducdYld2OTaBCXOT6d1KxobvDWJ75d289yMmo/edit?pli=1>

[To-Do List & Bugs](#_j93yjljjot6l)

[Code Documentation!](#_rqv72u902pqv)

[Authorizing authorized Users](#_x6a0w8lt3shk)

[Version 2](#_aoiuoxrg3pfl)

[1. Single Trigger – Eliminating the ccc object](#_1frhy2wsxeuc)

[2. on/off open/close Function](#_3a4ocjmuvcqo)

[3. New Code Architecture](#_a9mqkoa21zb5)

[V2 Specification](#_46y879yx18to)

[UNOs can contain the following svg groups:](#_almd0kkt59to)

[ooo-group](#_jfv7o0kjjpxj)

[xxx-group](#_97yaxn3qwgxu)

[vvv-group](#_vbyavkf7aksk)

[sss-group](#_oh8xey3kbmb4)

[Loose / un-grouped / un-named artwork](#_3p7rp586amf8)

[URL Commands](#_kgmxhsyjvsai)

[Mouse Interactivity](#_cevzwqpg1fpd)

[Interactivity Options](#_1bi6uqvv5lr)

[Tweening: making it useable](#_b2n3me9drfis)

[Tweening: Relative Tweening Issues](#_567ozvemk1br)

[Tweening: On/Off Open/Close Functions](#_xqts11g0dnq)

[Tweening: Folding Issues](#_f63pdbqloec6)

[Toggle Modal Help](#_rv0y02jz0sgy)

[Pane-sizing and pan-zoom interaction](#_733eca2el1lg)

[Forms](#_xfbskx28lm10)

[Survey Forms](#_qx920kjlyjbb)

[Database Editing Forms](#_rknxw5h5orka)

[Export Mongo Database to Excel](#_4u28dadb13ec)

[‘all’ Command Issues](#_xli5b2cufyuf)

[Highlight Tween – tween z-index?](#_9i05jjr4r43m)

[Push URLs - to create a ‘shared screen’](#_1tf9cqoroxxe)

[Supporting Mobile Devices - Ready for testing](#_rs6ai095i0ru)

[User Controls / Preferences - saved to JSON file](#_ef8eoqqvq5r4)

[Hover Weirdness – lower priority - don’t need to fix immediately, but ...](#_34d44rifhgq1)

[Rotation Bug-Fix Bug](#_n8axuuxvzsc5)

[User Registration - status?](#_4lze0oj5k8zw)

[Qtip2 Positioning Error](#_pfh1rtpdp47d)

[What to tackle next?](#_rfdvpne3mb7s)

[Search Feature](#_h62zqqy5u5no)

[Handle Outdated Svg get Code](#_z8ocgumcodkk)

[Keyboard Shortcuts](#_wwx9lbavosr9)

[Align keystroke-zoom and button-zoom increments– MC Investigating](#_x7f6uanqqkpu)

[Pan limits – MC Investigating](#_qq4q2uazzis6)

[Dynamic SVG: Tweening: Transitions, Loops, and Custom Tweens](#_o5afcjgz9a50)

[UNO / SVG Parameters](#_ygl76tgl2nb1)

[Changing UNO / SVG Parameters](#_6wvw3jn6pdzt)

[Specifying the ‘Dynamic SVG Functions’](#_izyo19s4cbrq)

[Loops & how to stop them](#_945q8n8w8my6)

[Calling the Dynamic SVG functions](#_sywq2qylzqf)

[Dynamic SVG functions & on/off open/close](#_1dsdb6yrul0y)

[Transition UNO Commands](#_7pqzxpm2hhim)

[Loops](#_lgy62nxmeklw)

[Custom Timeline Animations](#_33p4j9o2ctsy)

[Highlight Tween – this scheme on hold.](#_yxfsucgd8m0w)

[URL Specification](#_avms37em8mfx)

[Lock-On Function](#_qud0rv5cxr6d)

[Updating the URL](#_x0f1rxhi3kn7)

[Website / Hosting Tasks](#_78myg1flr3xy)

[Perform Pre-processing on SVG at time it is uploaded](#_on8jbxo56det)

[Process for re-hosting maps](#_esap0hl1y0nv)

[Database Schema - see LM Video](#_w5ocxdtix7w8)

[\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_](#_86xuzs5q6hcn)

[Future Development](#_bdifwhgca6cn)

[Database Replace vs. Add-to](#_nx2nqxz4p77q)

[Revisit Highlighting](#_n04xwencq4wv)

[Database Editing/Table](#_eh5qnrei2m4u)

[DB/SVG Error Checking](#_h47h9j88vbga)

[On/Open by Zoom](#_7vuxzj3u48v9)

[Tear-off Panes](#_buqax3v4cyoe)

[Database Schema](#_hpm8yfdeh9cz)

[Folding](#_jecpbxn0udlg)

[SVG/Viewport Wrapping](#_oyq7q4kszdkc)

[Enable loading of local files – Low priority](#_9ew9x9d4jqau)

[Implement jjj commands ? ?](#_12m2sdd8lodf)

[Tooltips](#_6xzmw7fu94pb)

[Position-Override](#_dt4sqrpmxiib)

[Tooltip Centering](#_ct8yhji3yzr3)

[Left Tooltip Position](#_j6fp1lgzugjf)

[GSAP pan/zoom - On Hold for Now](#_nd04rh5udwfx)

[Pan/Zoom Performance](#_sobh01icq7od)

[Object Specification](#_gmnwr9l5xejj)

[UNO Commands](#_dxze73u0i2d8)

[Mouse Interactivity](#_z7vztrxu1hq1)

[Mouse Interactivity With hoverOpenClose Set to open](#_iihxneoa9yf5)

[UNO State Table](#_nqjlzzc0xegj)

[UNO Specification](#_i0c34ylvbf16)

[UNOs can contain the following kinds of things - any of which are optional:](#_x2fwwblhxx46)

[Loose artwork](#_9klj2922gd0g)

[ooo-group - open-trigger:](#_an4we6rnhf7i)

[sss-group: <id=sss>](#_1xta8kh1d8l9)

[vvv-group - visible when open:](#_p95jm97w2e7d)

[onDimming](#_ec6ijg2g5gk7)

[ccc-group - close-trigger:](#_lluibv9axaxl)

[Other UNOs:](#_a3gwyuq2ge0o)

[URL FUNCTION](#_2603vbvmezyo)

[Selective URLs +++ Command](#_goyz5d3y74iy)

[URL, CSS, and JSON Commands from the SVG file](#_ef71dxv3090t)

[rrr-commands <id=rrrS> <id=rrrH>](#_3cq7kf9gvjvl)

[CSS-commands <id=cssS> <id=cssH>](#_kzsxsxvcj6bf)

[jjj-commands <id=jjjS> <id=jjjH>](#_cwg2ksgjw60v)

[Old Specifications and Finished Tasks](#_s0a8kyw0tlab)

[Mouse tip stays on sometimes and hangs the page fixed](#_6zkwmkfs88so)

[Pane Open/Close](#_d7p75in1s9ra)

[Individually command’able UNOs while maintaining stacking order Done](#_avg3t7v2a3dv)

[Hovering over an object then off sometimes leaves the tooltip on, and hangs everything.](#_uxni7mijnc9g)

[Extra “&=” Fixed  
 - getting left in the URL after interactions.](#_xq88q9cpdyyv)

[rrr & css stuff - Done](#_1r353s6ataz0)

[Info-pane   
 - not getting filled when hovering on pane text. Fixed](#_fkfzy5za9aoa)

[Pan bug](#_s1630ynmzetg)

[- Simplify URL coordinates Fixed](#_bpbqa19rrhqk)

[Mouse Jitter](#_lv19ly64aiz3)

[Pane-text hover bug](#_v2q9wfjl4w8s)

[Restore shorter delay](#_mdzr62vtftkk)

[Design bug in vvvoooMap?](#_ly7kk36hoba)

[Master + Override Databases](#_ao9xj66kuy0s)

[Ignore UNO’s in sss’s](#_rni7whjybz3o)

[Hide UNOs in closed vvv’s](#_835uc3vzbk6o)

[Mouse events for hidden children](#_earwltpujqb4)

[InstitutionalInvestors UNO - not getting click events](#_jeuwy8d5y2ab)

[InstitutionalInvestors UNO - not getting click events](#_q79ylkk8dz8l)

[ondoubleclick filed not getting executed properly Fixed](#_w6l2ufcfgykl)

[css stuff - Working](#_orw0uyucsdqq)

[Child State](#_ate8eujystkd)

[ooo’s not being hidden on hover](#_yw03a9bc99c3)

[Intercepting mouse events - or not](#_z5mtnt29rj9b)

[FIx handling of Opacity Groups & Stacking Order](#_x8a90yuyj2k8)

[?+++ Command: use the current pan, zoom, pane-states - Fixed](#_xe8uj8ly9gav)

[‘Direct’ pane-links not getting added to URL/history - Fixed](#_e83zlekob80)

[Set URL/history on zoom controls: ‘+’, ‘-’, or ‘Reset’ fixed](#_ape2nj53gtap)

[+++&closed Bug - User (MC) Error](#_o3g5nuhj4igh)

[A hover on a pane-link can set the URL FIxed](#_ixztvtqakea2)

[Fast-clicking pane-links garbles the URL test it now. FIxed](#_bwo71m2p9n4b)

[open no longer overrides a parental close test it now - Fixed](#_8w305tvmjts1)

[manually adding a URL in address bar containing a +++ may be problematic Fixed](#_a65pr26tofim)

[Bug in clicking on pane-links? fixed](#_3k2x6io1c4pn)

[After closing a pane then hovering on link, pane re-opens… Can’t duplicate](#_mjmr3gl4kbce)

[Pane state/width in the URL Done](#_c0nmt7k141yc)

[URL update on on pane-resizing](#_cz4a9spbsqw3)

[Pan/zoom-centering for different window sizes - FIXED !!!](#_4obqsz8ypm08)

[Pane slide vs. toggle fix. – Done](#_pvtt7a4fq1sz)

[Set up node.js hosting / development site](#_ve9pdfry8ui6)

[Add password for test site access -](#_rxqwsgmze8gw)

[Pane-text and file commands](#_fr0ny04i8oyg)

[Finishing Presentation Code](#_khxkw9kpl8ic)

[Default/Initial URL – Fixed](#_18z015lhnef1)

[Per-UNO all commands: openall= and closeall=](#_liw0pcm8btuo)

[Copy URL to clipboard on shift-click](#_c89i0cwf4pty)

[Per-UNO all commands: openall= and closeall=](#_xw5yhhof3ip)

[SVG processing should ignore <symbol> tags – Fixed](#_slcckf528bi2)

[Move rrrS/rrrH to Database – Works](#_ih3u15m3dk82)

[Issues with css classes – Testing - working](#_nxpb6jku502r)

[Designer Bug](#_m2qyzem0ye94)

[Zoom/browse key overlap - Fixed](#_qg94rll087co)

[e / w pane Toggle not Slide fixed](#_2cs2er8w7fdl)

[Move rrrS/rrrH and cssS/cssH to Database](#_8699bem94hql)

[Goto Command – Working - Testing/ed - Done!](#_sfbtrlqbnw1u)

[Opening child uno’s need +++’s to retain state](#_bcwivikalsms)

[Startup URL mismatch](#_n9pg6or9pp15)

[Multiple Pane-Link Classification – Fixed](#_by6xd4qa10d9)

[Disable the Multi-function Space Bar – Fixed](#_gsuas6aa8ned)

[Handle local links from ondoubleclick – Works](#_mzuttwci6lhy)

[Collapsable Text – works](#_s5xvpkm2znko)

[Import HTML files to the panes - fix linking – Working](#_akaw7l2orsft)

[Display:none](#_reyazr8tvp7)

[Finishing the Animation Code](#_lou72yfbvsmk)

[Gotoz a bit broken - add fix to idiagram-svg.js](#_i5vr1tafce8n)

[UNO Naming Harshness - Fixed](#_5cwbbm2uyikz)

[Finishing the Animation Code - Fixed](#_9z08bdr31rpf)

[Pane-Scroll fix – Fixed](#_ark1sm7zeum6)

[g-clicks – Working](#_9iaxqc6r6lor)

[On/Open URL command bug – Fixed](#_j26k4a4rrcnq)

[Long zoom value – Fixed](#_9kw6rm97n58e)

[Option/Alt-Click should show up in the URL - Fixed](#_2ifj9xtmz0lz)

[Save as PDF – Done](#_yggv4h8d61s7)

[Database ID Format](#_whnkr0i7nvf6)

[User Registration](#_utsxhj11bx3k)

[Add tween for .classes – Working](#_h71nstlu3kke)

[JSON Preference file name – Working](#_b91tz0b01e7c)

[Compact URL Format - too tricky to implement](#_c8x88yvk8imo)

[Bugs in Webpack Version](#_74yd562c97sl)

[Draggable Panes – Fixed](#_aylqzmo7uezb)

[Custom Function Error - Resolved](#_ad5stb5upo75)

[Browsefify’ing – implemented - Working](#_jemyix8dx7ez)

[Qtip2 - working](#_a8zbf7ope58d)

[IE 11 Compatibility - Fixed](#_sel6kupflj5g)

[Help & Preferences screen – MC Implementing](#_39hzfwzdr5c1)

[Help Page / Help Key - Done](#_du03n9w0qfa5)

[Override (and Non-Override) Show/Hide – Working!](#_wtp97lxm3cex)

[Tween around center - Works great](#_1se6sji8ksxv)

[Toggle Switch - Works!](#_aigquadaeamx)

[Keyboard Focus Issue – Code commented out for now.](#_wgnzel4uawyj)

[Home & End keys - Done](#_8m97ug6a4cr3)

[importSpread UI - Done](#_gadghpxrdnrl)

[Printing Window/file Name - Done](#_nek1wud2j9se)

[Pan & Zoom Tween – working](#_ofh2skudur8a)

[Rotation Bug – Working](#_rijs49upgzqm)

[Tween Compound & Multiple Classes/IDs – Working](#_ce7ztiwoz2va)

[Animations Not Playing - use ‘segmentFile’](#_q2kcnlmkpf7y)

[Remove Duplicate URL Commands - Implemented](#_ww8iw41kbqd5)

[Accordion Reset Issue – Fixed](#_svxq1dtbmrs2)

[importSpread UI - CamelCase on the DB names!](#_qijcxhd0cz3f)

[Custom Tweens Examples - Done](#_z7hkdddn8wfn)

[Stop including all the stop commands in URL in an animated presentation. working](#_xkszk27zmt03)

[Bug: .ccc class as a child combinator in tween commands - not a bug](#_e5dnbv2lps2c)

[“On” command is disappearing from the URL address box – Working](#_9snmmntvfewx)

[Info-pane Loading Issue – Works most of the time](#_etds6u1x0g0n)

[Keyboard focus for forms with textboxes – Works](#_lzn0jn6ho5wf)

[Arcadis Interaction - Working](#_ne8a4wxkbeh3)

[Set up idiagram.com droplet – Up and running](#_6qf0s0cqlp1g)

[Password for Arcadis production folder – user registration implemented](#_cpwjbd1dee9o)

[Info-pane filling on click - FIxed](#_gzpwt73wmioh)

[Additional hover options - Working](#_bkuos9tjulk0)

[After login folder needs to change from order - Working](#_4szjajtvbdin)

[Commands without parameters - Working](#_gvqbodhdt43m)

[Hover/Tooltip timing bug – Working](#_enmnztm3l6zp)

[Info Pane Filling Bug – Fixed](#_abq97kxynsms)

[Hovering / Tooltip Issues – good enough for now](#_wlpaghab1gl3)

[Tweening: Custom Function Modifications - working](#_shjeg1eqslk)

[Tweening: Schemaless Spreadsheet Import to schemaless DB - DONE](#_nvz97b5onhur)

# To-Do List & Bugs

### Code [Documentation](#_4lx8u7us29xu)!

LM and MC keep expanding / editing - **Documentation Doc**

### Authorizing authorized Users

We need a mechanism for approving / denying user accounts to people registering for /authorized access.

## Version 2

Has 3 main changes:

### 1. Single Trigger – Eliminating the ccc object

I have very reluctantly concluded – as the basic code has been working so well, for a long time, and as it will require a huge amount of map re-building on my end – that you’re right (and I hate it when you’re right ;-): we should do away with having 2 different trigger objects, and go with a single UNO trigger object. This will simplify the code and map-building, and allow us to remove the complication of the temporary open/close rigermorole.

The other advantage of this simpler approach is that is separates the trigger object from the visible artwork in the xxx and vvv. This means that you can now tween the xxx and vvv with impunity - without moving the trigger object from underneath the pointer.

**For future reference:** If someday we want to bring back the moving-trigger functionality, we could add an ‘onClickOn’ and ‘onClickOff’ fields to the DB - which contain a URL functions to be executed. These could be used to switch between 2 different UNOs/ooo’s, using the on/off click URL commands:

off=thisUNO&on=thatUNO

on=thisUNO&off=thatUNO

### 2. on/off open/close Function

Also in this revision, we’re moving from on/off open/close being limited to show/hide (although this will still be the default). On/off open/close would now execute the specified on/off open/close function, that will do *something* with the UNO on the event. See the section **Tweening: On/Off Open/Close Functions** below.

The ooo-group will now be the single trigger. Opening/closing will then simply toggle the vvv-group and the xxx-group (xxx is a new group we now need as we’ll no longer be hiding the ccc-group on an open)

For backward (in)compatibility, the ccc-groups should just be ignored.

Because of this significant change, we should give this new version a new name - and produce an up-to-date spec / documentation.

### 3. New Code Architecture

This change could be an opportunity to rewrite the core interactivity code in a cleaner more robust promise’y way. Also - see below **Tweening: Relative Tweening Issues**  - re another timing / finishing issue that we might want to roll into new code. V2 is an opportunity to re-think the code design based on what we know now, vs. what we started with an gradually built on.

We may first want to do a quick-n-dirty ‘interim’ V2 that implements new specification outlined below, but using the existing code architecture. This will quickly get us to a version a that I can use to (re)build maps with the new spec. Once we’ve ironed-out the details of how the new spec should work, *then* we can revamp the code architecture.

I’m thinking the new (much simplified) spec - including the idea of having different on/off open/close functions - should be:

### V2 Specification

Note that the fundamentals of how things work doesn’t really change all that much. We’re just going to the one trigger, and changing which group gets hidden on an open: now the xxx-group instead of the ooo (which is now always-on).

However, we’ll need to think carefully about the consequences of going from the on/off open/close functions only showing/hiding, to allowing tween functions that might move, size, rotate, etc. For example, an on-function that uses a ‘move’ would also have to move the vvv-group, even though you’ll probably want the vvv-group to be hidden during the move…. See **Tweening: On/Off Open/Close Functions** for more.

### UNOs can contain the following svg groups:

#### ooo-group

**<id=ooo>** – the ooo-group collects the UNO-trigger artwork into a group (rather than leaving it loose). This group is always visible while the UNO is on. (Note: it should be possible to keep the pointer on the ooo and click it on/off/on/off …)

#### xxx-group

**<id=xxx>** – the xxx-group gets turned off when the UNO is open, and turned on when it’s closed. The xxx-group can contain loose artwork, other UNOs, and sss-groups.

#### vvv-group

**<id=vvv>** – the vvv-group gets turned on when the UNO is open, and turned off when it’s closed. The vvv-group can contain loose artwork, other UNOs, and sss-groups.

#### sss-group

**<id=sss>** –Contains static artwork that is ignored for mouse events. It can be used to differentiate ‘background’ artwork from the open-trigger artwork in simple UNOs.

#### Loose / un-grouped / un-named artwork

**as the trigger artwork**- we can eliminate this behavior if it’s obsolete - and only accept triggers in the ooo-group.**:** *in the absence of any other groups in the UNO*, any ‘loose’ artwork is treated as the UNO trigger. This is useful for creating simple UNOs that have no vvv-group or other groups, and are just used to trigger a tooltip and/or info-pane.

**as just more artwork** – if there are other groups, then any loose artwork, without an ID, is treated as ‘static’ by default. But it’s better practice to put any loose artwork in an sss-group.

### URL Commands

**on**=thisUNO – runs the on-function for everything in the UNO with the exception of the vvv-group.  **off**=thisUNO – runs the off-function for everthing in the UNO.

**open**=thisUNO – runs the open-function for the xxx-group (and any children of xxx) and runs the close-function for the vvv-group.

**close**=thisUNO – runs the close-function for the vvv-group (and any children of vvv) and runs the open function for the xxx-group.

**‘all’ commands -** should follow a similar logic.

### Mouse Interactivity

**Hovering vs. Clicking** - hovering on an UNO creates a ‘temporary’ open/close: hovering off the object will reverse the state, closing/opening the UNO. Clicking creates a ‘persistent’ open/close: after a click, the hover-off is ignored and thus the open/closed state will not change.

**Tooltips and info-pane**: will be loaded as specified in the DB on hover or click events.

**Hovering - on/off the ooo-group**

**If the UNO is closed and "hoverOpenClose": "open" :** a hover-on will run the open-function for the vvv-group. If no click is made, a hover-off the UNO will run the close-function for the vvv-group.

**If the UNO is open and "hoverOpenClose": "close" | “both”:** will run the close-function for the vvv-group. If no click is made, a hover-off the UNO will run the open-function for the vvv-group.

**If the UNO is open and "hoverOpenClose": “both”:** a hover-on will run the open-function for the xxx-group, and the close-function for the vvv-group. If no click is made, a hover-off the UNO will run the open-function for the vvv-group, and the close-function for the xxx-group.

**Clicking-on the ooo-group**

**If the UNO is closed and "clickAction" : "openClose":** *if not already triggered by a hover,* run the open-function for the vvv-group, and the close-function for the xxx-group. Ignore the hover-off the UNO.

**If the UNO is open and "clickAction" : "openClose":** *if not already triggered by a hover,* run the open-function for the xxx-group, and the close-function for the vvv-group. Ignore the hover-off the UNO.

### Interactivity Options

(change hoverOpenClose to hoverAction - and have it work similarly to clickAction) Set the actions that are triggered by hovering or clicking on an UNO:

“hoverAction”: “openClose” // hovering will open a closed UNO, and close an open one

“hoverAction”: “open” // hovering will open a closed UNO

“hoverAction”: “close” // hovering will close an open UNO

“hoverAction”: “none” // hovering will trigger no map actions, *but* it will show the tooltip / info-pane content if it exists.

“hoverAction”: “somethingCool” // future additions

“clickAction”: “openClose” // a click will open a closed UNO, and close an open one

“clickAction”: “gotoz” // runs the gotoz. The url is still updated the same as before. If the uno is in a close state, it will not open it before zooming into it. Pressing **c-click** will will open a closed UNO, and close an open one.

“clickAction”: “somethingCool” // future additions

## Tweening: making it useable

The items below marked **Tweening** are a set of things that all need to be implemented to make the tweening functions easily used in the map interactivity.

### Tweening: Relative Tweening Issues

Unless we can figure out how to stop and reverse relative tweens in mid-tween…

**The problem:** if a relative tween is interrupted in mid-tween, the relative values will now be in error and all subsequent relative tweens will be in error. For tween functions that use *absolute* variables, there may be no problem.

**The solution(?):** I was thinking that the solution would be to prevent any pointer events until the tween finished, but it might be much better to instead force any relative tweens that have started to finish before they’re stopped or tweened again.

(see http://greensock.com/timelinelite)

Before a tween is started (or stopped) we need to check to see if any of the objects being tweened (remeber there my be multiple selectors) is already being tweened: <http://greensock.com/docs/#/HTML5/GSAP/TweenMax/isTweening/>

If any of the selectors are in motion, then we cue-up the new tween(s) ( there may be multiple tweens triggered on an object) to execute, in order, when the previous tween finishes (using promises?) .

---------

**Thinking out loud:**

Ugh - the possibility of having multiple / nested tweens running can get really really ugly. I think the only viable solution will be prevent *any* mouse interactions while *any* relative tweening is happening. At that start of any tween function that changes things *relative,* set:

. ooo ( pointer-events: none; };

And the onComplete function will check:

If ( all tweening has stopped ) (

. ooo ( pointer-events: all; };

}

Then we’d need to ‘manually’ determine if there’s been a hover-off the UNO while it was being ignored.

Or perhaps we only need to ignore mouseleave events for that UNO’s ooo ? That way we could still process click events.

Or maybe we leave mouse events on, but only process the mouse-leave / mouse-enter events for that UNO if all tweening for that UNO has finished ?

**But** we can’t have pointer events on for any UNOs that are tweening - and an UNO could have children that could be triggered to tween. Or the openURL / closeURL could set other things tweening.

So we’re back to: if( anything is tweening) { no pointer events allowed }

During on/off tweens, pointer events should be none.

Another solution is just to ensure that the tween durations is always less than the JSON show/hide delays.

### Tweening: On/Off Open/Close Functions

We need to think carefully about the consequences of going from the on/off open/close functions only showing/hiding, to allowing tween functions that might move, size, rotate, etc. For example, an on-function that uses a ‘move’ would also have to move the vvv-group, even though you’ll probably want the vvv-group to be hidden during the move.

And

**Setting the tween starting state**

The tricky bit: Getting the UNOs into the right initial state: in the proper off or closed tween state - faded/scaled/moved/rotated properly so that when they’re turned on or opened the on/open tween will have the right starting state.

Option 1 - in the on/off open/close function, put in a ‘pre-tween’ with duration 0 before the real tween.

Option 2 - on loading the map, go through everything and run it’s off and close function - with duration 0 (so it doesn’t take forever).

We should set the map-pane to not show anything until the svg is loaded, things are turned off/on and the map is pan/zoomed to the initial URL.

------

To enable fancier transitions than an instant change in visibility, we’d like to apply different on/off open/close functions.

There are 3 ways these functions can be defined:

**Default 1 –** If the on/off open/close is not defined in the DB, then the default functions called for on/off open/close are the same they do now: setting display:none | visible and opacity:0 | 1

**------------------ Ignore for now**

**Default 2 –** In the JSON file it is possible to change the default on/off open/close functions to some other function - contained in customstuff.js - for example:

“onFunction”: “fadeOn”,

“offFunction”: “fadeOff”,

“openFunction”: “fadeOpen”,

“closeFunction”: “fadeClose”

-----------

**Custom –** The default on/off open/close functions can also be overridden on a per-UNO basis by specifying the function in the DB. There are 4 optional fields in the database (that can be easily added if needed once we have the schemaless DB working) that contain the name of a function specified in customstuff.js:

**onFunction offFunction openFunction closeFunction**

Note: For the off tween, the code should wait for the tween callback and then do a display:none (to improve display performance). For the on tween, first do a display:visible.

### Tweening: Folding Issues

**Note: these issues may be fixed by implementing the things above.**

I’ve put together a quick test of the ‘folding’ the maps (like ‘accordoning’ in html). The Arcadis ACF map is a good candidate for applying this kind of folding.

Putting GSAP move commands in the openURL, and closeURL, fields of the DB make this very easy to accomplish. However, to work properly it requires that hover/click events be handled perfectly - and we don’t quite have perfection yet.

Have a look at:

[**https://gsap.systemsagency.org/demo/fold/fold2.html**](https://gsap.systemsagency.org/demo/fold/fold2.html)

This has "hoverOpenClose": false, so that we get no hover events. The basics of it work, but there are 2 issues:

1 - if you click on something that is already open/closed it will keep firing the openURL or closeURL - and the UNOs will get moved out of place. To fix this we just need **fix up the code so the open/close URLs only get fired once on new open/closes**.

2 - if you click on something while the move tween is still moving, the GSAP records the relative position of the UNOs at that moment, so the relative positions get messed up. The solution to this (and other problems) would be to **set a promise on all GSAP tweens, and prevent events until they finish** (see the other issues below).

[**https://gsap.systemsagency.org/demo/fold/fold.html**](https://gsap.systemsagency.org/demo/fold/fold2.html)

In this test we have "hoverOpenClose": true, so we get hover events. But I have the tween duration set to 500ms, and the JSON tooltip delays set to 520ms - to prevent hovers until the tween has finished. But you’ll see that it’s still not hard to confuse the positions.

[**https://gsap.systemsagency.org/demo/fold3/fold.html**](https://gsap.systemsagency.org/demo/fold/fold2.html)

Same as above, but the JSON tooltip delays are set to 0 - and all hell breaks out. This is where careful application of promises might prevent total chaos.

### Toggle Modal Help

It would be cool if the 'h' key could toggle the help screen off as well as on.

Other than that, the modal help is working well - and is a big improvement on the old open-new-window method.

### Pane-sizing and pan-zoom interaction

I thought we might have fixed this, but there's still a problem caused by the interaction of pane resizing (which changes the map-pane size) and pan-zooming (which is done relative the the current map-pane size).

The problem occurs when you send a command like:

#/?+++&open=Collaboratory&panx=1075&pany=760&zoom=1.4&epane=500

The pan-zoom isn’t what expected because it’s relative the map-pane size *before* the epane is changed to 500.

When any URL command is received (from address bar or from a link) we should first adjust the pane sizes, *then* do the pan-zooming (now relative to the new pane sizes).

### Forms

Forms enable maps to be used as a tool to collect input from (registered) users. A form scheme would involve:

1. **Registering** users, setting them up in the Mongo DB, and determining who does or doesn’t have access to form input.   
     
   We’ll need an admin screen to set these configurations/privileges for different users to access different parts of the site.
2. **The Form**: Creating HTML / Jquery form pages to go in the info-pane (we have this working more-or-less already). Would it be helpful to use a fancier form library? Such as:  
     
   <http://fancyjs.com/> – looks like a simple library for creating nice forms.  
     
   <http://www.warski.org/blog/2014/09/introducing-supler-a-functional-reactive-form-library/> – doen more stuff - that I don’t know if we need  
     
   Or Google Forms? See:  
   <http://45.55.23.31:3000/demo/radial.html#/?on=Map&on=IssueX&openall=A&epane=400&wpane=260&panx=1000.0&pany=1000.0&zoom=1.440&story=radialStory.md&info=issueXgd.html>  
   +works great …. – Google branding … the branding can probably be removed … but it’s very difficult to modify the form appearance - and it won’t help with the DB editing..  
     
   And Bootstrap has forms:  
   <http://v4-alpha.getbootstrap.com/components/forms/>  
   <http://www.w3schools.com/bootstrap/bootstrap_forms.asp>
3. **Connecting** the form to the DB:   
   **A. Loading** the form: when the form is loaded, any existing user input should be loaded from the DB into the form: so the user can see what they’ve previously entered, and edit as needed.  
   **B.** **Editing -** we have fixed the keyboard focus issue; differentiating when we’re in a *text-input box* from *everything else.*  
   **C. Submission**: The user’s input should be submitted/stored to the DB - linked to the user’s registration.
4. **Exporting** the input data - export & download the form data in Excel format. Probably as user-per-row, input-item-per-column.

I think that’s about it.

### Survey Forms

We’ll need to set up a new DB collection to hold the answers - a DB field (column) for each answer, and a row for each user.

We specify the database collection to use …. where?

And then download the survey DB collection to excel format. See the item below **Export Mongo Database to Excel**

### Database Editing Forms

I put together a Bootstrap form using: <https://formden.com/form-builder/>

See a first-cut at the DB input form here:

<http://45.55.23.31:3000/demo/things2.html#/?on=basemap&panx=937.3&pany=720.0&zoom=1.306&epane=500&wpane=260&story=story.txt&info=dbInputForm.html>

Note that the id’s of the form elements = the DB field names.

How / where is the map form created?

-------------

As a first test of the form-input code, we should set up a form to edit the DB entries for UNOs.

To enable for editing, we set up a different .html and .json page that allow editing by setting a flag in the .json file such as:

"editing": "true",

If the ‘editing’ flag is set to true, then clicking on (but not hovering on) an UNO will load the info-pane with the DB form populated with the DB info for that UNO.

To turn on or off map db editing when a UNO is clicked, insert this command in the URL:

mapform=true

Mapformfalse

**TODO:**

1. ~~Put this on gsap.systemsagency.org~~
2. ~~Set a default database to populate -- override~~
3. ~~text boxes are not being populated. Maybe because they are markup?~~
4. This loads the override version of the database by default. If the override database has not been populated, then no error is returned. Should then try for the master database.
5. need feedback when db is finished being updated.
6. Form is too long. Need to scroll to click on submit. Maybe put submit on top (also?).
7. Changes to check boxes are not being updated in database.
8. After making the update, need to update the database var that is in memory, though there are some other vars that are not changed. The user will need to refresh the screen to see the effects of changing the database.
9. Hitting escape while the db form has focus should close the form. Now one must be focused in a text box first before pressing escape. Also make a close button.
10. ~~Put form in the info-pane.~~
11. ~~If clickAction is set to gotoz then this feature will not work. Make it always work, even when clicking on a closed uno that is on.~~
12. Some uno’s do not have click turned on in the database. This should still work.
13. mapform=false probably does not turn this feature off yet. Fix it.
14. At some point only authorized users should be able to use this feature.
15. Create an exportSpread.html for downloading the DB to a spreadsheet.
16. Set up the code / process for creating & executing survey forms.

**Documented Feature:**

The map Db Editor will only show up when the user clicks on an UNO that happens to be in a closed state. If there are those who wish to click on opened UNO’s and see a map db editor pop up, please let let us know.

And then, once we’re editing the Mongo DB directly, or collecting survey data, we’ll need the companion feature:

### Export Mongo Database to Excel

Export / download the DB from Mongo to .xls.

See:

<https://github.com/Moblox/mongo-xlsx>

http://razorsql.com/features/mongodb\_export\_tool.html

### ‘all’ Command Issues

**Documentation:**

The ‘all=’ command goes through the list of all the UNOs in the SVG file and sets each, individually, to on, off, open, or close.

It does ***not*** follow the parent-child on/off open/close hierarchy

It does ***not*** execute any onXXX commands from the database (to prevent any conflicting commands that might be contained therein).

---------------------------------

**Address Bar**: for all ‘all’ commands, the URL is not properly updated:

* all=on/open commands are not removed when an all=off/close command is issued. - WORKS
* all=off/close commands show up in address bar (they never should).
* Duplicates are not removed. - WORKS

**Restults:** The results of the commands seem to vary based on which map is being used:

<http://45.55.23.31:3000/Arcadis/arcwayv1.html>

?+++&all=on - works

?+++&all=off - works

?+++&all=open - works

?+++&all=close - works

<http://45.55.23.31:3000/demo/radial.html> - after removing all DB onXXX commands

?+++&all=on - works

?+++&all=off - works

?+++&all=open - works fine in <http://45.55.23.31:3000/demo/radial.html> but not in <https://idiagram.com/arcadis/arcwayv1.html>

?+++&all=close - works

* **Serverless Packaging**

What’s involved in packaging a map into a stand-alone thing that can be thrown onto any server?? We’re going to need this sooner or later…

Could/should we use: <http://electron.atom.io/>

Perhaps ‘packaging’ should be more like a standalone desktop app:

<http://tutorialzine.com/2015/12/creating-your-first-desktop-app-with-html-js-and-electron/>

The list of options in this thread <http://stackoverflow.com/questions/6834537/packaging-a-node-js-webapp-as-a-normal-desktop-app>

Or <http://enclosejs.com>

We don’t necessarily need it to be an off-line desktop app. However the advantage of the app would be that it takes browser dependency (and internet connectivity) out of the equation – as long as the embedded Chromium browser works well.

------------

**LM says:** With all of these extra files that we pull in -- they will need to be referenced by the full url, not just a relative path -- like the info pane files and the story panes files. Also the image files including the svg files. The data in the configuration json file (which has same name as the html file) will need to have the data show up in the html file instead. The /js/bundle.js will need to change to the full url address, like <https://idiagram.org/js/bundle.js>

### Highlight Tween – tween z-index?

For testing purposes, I added a ‘Dimming’ layer to the test file:

http://45.55.23.31:3000/demo/things2.html

You can activate dimming by running this URL command:

?+++&fade='@Dimming',0,f,0.0&on=Dimming&fade='@Dimming',2,f,0.7

To test, we just need to write a function – in things-customstuff.js – for the command:

run=tweenZ, ‘unoId’, duration, relative, zIndex

To move an UNO about the Dimming layer:

run=tweenZ, ‘@unoId’, 1, f, 0

?? Is the original zindex maintained, so that a relative ztween:

run=tweenZ, ‘@unoId’, 1, t, 0

Would put the UNO back in it’s original position ??

-----------------

New idea: use the GSAP tween z-order to pull highlight UNOs to front - just in front of the dimming layer UNO.

When highlighting is enabled, the dimming layer UNO is turned on and tween to the top of the z-index. The dimming UNO contains just a map-sized white rectangle. The dimming layer opacity can be set/changed using the fade command.

The highlighted UNO(s) are then tweened to the top of the z-index, just in front of the dimming UNO.

### Push URLs - to create a ‘shared screen’

To join a ‘webinar’, people would be invited to log into a chat session. The chat session would handle log-in, and the chat function could be used for people to ask questions, to send dial-in information, etc. ( for now we’ll assume is voice/audio is handled by a com-call, or Skype, etc.). The chat window would open a another browser-window that serves as the target for the map URLs that get pushed to the chat-group.

Similarly to how GotoMeeting works, one person in the chat-group should have the ‘control’ window: the master map from which the URLs are pushed. Optionally, control can be passed to anyone in the chat group so they can set the view.

For this to work properly, we’ll need to make sure that all necessary URL commands do indeed get passed through in the pushed URL. And we may want to disable interactivity in the ‘slave’ windows so that participants can’t mess up the view and get out out of sync - which would especially be an issue when sending +++ URLs (which assume a particular previous view).

**Got socket io working with a simple interface that we can change**. I just wanted to get it working. The way to test this out is to open two versions of the same map, such as

http://45.55.23.31:3000/Arcadis/arcwayv1.html#/?on=Map&openall=Map&wpane=300&epane=0&panx=3531.9&pany=278.81&zoom=2&story=ArcStory.html&open=Thinking

To make one of the versions as the master, send in this command: *+++&masteron=1* ( the +++& is optional). Then after that the commands that you send via the address bar updates the slave browser window. The command, *masteroff=1* will turn off this socket feature.

Bugs:

1. ~~If you send in~~ *~~masteron~~* ~~again, then it will send that command over to the slave and they will both be masters and not work.~~ Fixed
2. Right now it doesn’t matter what map you use, the master will control all other browsers that are opened to any map. Need to fix this quickly so that a user can specify if they want to be a slave (or however we want to do this), and a master only controls slaves that are using the same map.
3. ~~Need a way to send what is on the screen and not have to always update the address bar.~~ Fixed
4. ~~I notice that when the slave is updated, all the values are not duplicated exactly. I don’t know if this is because zoom is calculated based on browser window size.~~ Seem to be fixed

* **Fix the case of Arcadis folder to arcadis**

Will do when I move arcadis stuff to the Idiagram site - when there is an Idiagram site - which I need to get on soon.

----------------------------

* **Implement Rotate Bug fix for other tween commands… - waiting on MC to test**

*…* such as fade, scale, move.

Also - if an UNO’s position is tweened, then it gotoz gets the position wrong.

### Supporting Mobile Devices - Ready for testing

With Hammer included, on an ipad, gestures do not work properly.

-----------------------

Getting it working - even if slow - on iOS and Android devices.

Using <http://hammerjs.github.io> ?

Hammer seems like it should be easy to try out:

<http://hammerjs.github.io/getting-started/>

### User Controls / Preferences - saved to JSON file

The initial / defaults for these variables can be set in startup.html / idiagram.js - hardcoded as they are now. But IF there there is a preferences JSON file, and IF that file has an entry for the variable, then the hard-coded value will be replaced by the value from the JSON file.

Things might be changed on a per-map basis, and/or that user may want to change - and should be saved locally to a preferences file/cookie:

* Zoom sensitivity **- Working**
* minZoom & maxZoom **- Working**
* pan/zoom tween duration **- Working**

For future implementation:

* Search function
* Pan limits
* Font size(s) - global multiplier
* Playback Volume
* Playback speed
* Animation (turn off for slower hardware)
* Auto On/Open by Zoom - enable/disable
* Default highlight value (0 to 1.0)

They’ll need to be read/written to a cookie when the page is loaded, or anytime the user changes them.

<http://www.w3schools.com/js/js_cookies.asp>

### Hover Weirdness – lower priority - don’t need to fix immediately, but ...

In <http://45.55.23.31:3000/Arcadis/arcwayv1.html>

Controls - > Map Rotation - > rotate to something other than Default

Hover on the upper part of a step object:

* In Chrome, Safari (Mac) - the tooltip will flash once - clicking on/off will be difficult.
* In Firefox (Mac) - the tooltip will just keep flashing on/off.
* In IE and other Windows browsers ? ? ?

The bug does not seem to happen in the un-rotated view.

### Rotation Bug-Fix Bug

The new scheme of setting things from display:none to opacity:0 before animating is working well. However, there seems to be a small issue with hidden things appearing during rotation.

In <http://45.55.23.31:3000/Arcadis/arcwayv1.html> **Control > Map Rotation**

When rotating the map the labels (class: arclabel) for the closed bits of **Strategy & Financial Planning** (in the center) appear during rotation.

A timing issue? Or interaction of different opacity commands? Interesting that it’s only the labels that appear.

### User Registration - status?

Mostly working? Left to do?

**Return logged in user to correct page.** If a user tries to browse to a page that requires logging in, then after she logs in, she is sent to the page she originally tried to browse to.

Set up a registered-user-only section on the production site - as a place to start building the initial Systems Agency website.

### Qtip2 Positioning Error

The tooltip shifts with zoom. Rather than try a cludgy work-around, I think we should see if we can find a solution by asking on the Qtip2 forum. Craig Thompson - the author - seems pretty active on helping solve problems. We’d probably need a CodePen with simple demo of the problem: a simple SVG with svg-pan-zoom and Qtip2 running.

In most cases it's OK for the tooltip to be positioned on the ooo-group. But it is possible for the ooo, and ccc to be different locations/sizes, so ideally the code should locate to ooo or ccc; whichever is visible.

### What to tackle next?

Next up, chosen in order of best/lowest fruit, could be:

* **Forms - surveys & DB editing**
* **Tweening Scheme: Dynamic SVG functions: loops & on/off open/close.**
* **Push URLs / socket’ing**
* g-click, a-click, etc. on UNOs - **√ done**
* Save as PDF - **√ done**
* Qtip - **√ done**
* Search feature
* Qtip2 - find fix for the zoom and mouse tracking issues - put up a codePen showing the problems - and see if the community can help fix it.
* Revisit Highlighting
* On/Open by Zoom
* Folding
* Wrapping
* . . . ?

### Search Feature

To aid users in finding things in complicated / data-rich maps (e.g. the Arcadis map), we should add a search feature.

**Accessing** - the ‘s’ key loads the search dialog - including the last search results - into the info-pane. Or a URL command can be used, info=search.html

**Searching -** user enters a search term, on submission we do a text search of the DB: title, shortDescription, and longDescription.

**Results** - for any hits on an UNO, we return a list of links to the search dialog: gotoz=uinoid. And we save those results (as a cookie?) - so they will be loaded with the search dialog the next time the search feature is trigged.

Note that the gotoz=unoid will probably replace the contents of the info-pane with the UNOs longDescription, so it’s important to easily get back to the over-written search results.

### Handle Outdated Svg get Code

The following error messages apply to code in idiagram-svg.js in createNewEmbed()

I got the following error message while starting up a map. I need to check it out:

*Synchronous XMLHttpRequest on the main thread is deprecated because of its detrimental effects to the end user's experience. For more help, check https://xhr.spec.whatwg.org/.*

I also got this message in the console. It needs handling:

*Invoking 'send()' on a sync XHR during microtask execution is deprecated and will be removed in M54, around October 2016. See https://www.chromestatus.com/features/5647113010544640 for more details.*

## Keyboard Shortcuts

Yet to implement: can we capture a key+UNO-click?

|  |  |
| --- | --- |
| **s** | Search - bring up search dialog in the info pane. |
| **d** | Draw - annotate - at a later date |
| **l-click** | On an UNO: lock-on function for that UNO |
| **i-click** | On an UNO: toggle highlight for that UNO - hlt=thisUNO |
| **a-click** | Play the animation for that UNO - play=thisUNO |
|  |  |
|  |  |

* Full List

|  |  |
| --- | --- |
| **[** | Zoom in |
| **]** | Zoom out |
| **\** | Zoom full |
| **Left arrow** | Pan left |
| **Right arrow** | Pan right |
| **Up arrow** | Pan up |
| **Down arrow** | Pan down |
| **Comma or Page-up** | Move to previous slide link |
| **Period or Page-down** | Move to next slide link |
| **Shift + Comma** | Move to first slide link |
| **Shift + Period** | Move to last slide link |
| **Spacebar** | Pause / Resume animation |
| **f** | Skip forward one element |
| **r** | Skip backward one element |
| **Shift-click** | Anywhere in the map: save the current view in the URL |
| **Option-click** | On a UNO: open-all / close-all |
| **h** | Show/hide the help screen |
| **w** | Toggle west pane, the story-pane |
| **e** | Toggle east pane, the info-pane |
| **p** | Save current view as a PDF |
| **d** | Draw - annotate |
| **l-click** | On an UNO: lock-on function for that UNO |
| **i-click** | On an UNO: toggle highlight for that UNO - hlt=thisUNO |
| **g-click** | On an UNO: zoom in on that UNO - gotoz=thisUNO |
| **a-click** | Play the animation for that UNO - play=thisUNO |

### Align keystroke-zoom and button-zoom increments– MC Investigating

Change the code in svg-pan-zoom to use our increments? MC to look into this.

### Pan limits – MC Investigating

Add LR TB pan limits - as part of the startup.html.

Set pan limits so the user can not pan off into oblivion. Pan limits should be set in startup.html.

Pan limits can be specified between 0 and 1:

0 = no pan beyond edge of the map

1 = pan up to map width/height beyond edge of the map

The default = 0.5 ; you can pan until the center of the viewport hits a map edge.

## Dynamic SVG: Tweening: Transitions, Loops, and Custom Tweens

### UNO / SVG Parameters

The initial/default map appearance is defined by the artwork specified in the startup SVG file. But that appearance need not be fixed, as the SVG can be dynamically manipulated within the browser using CSS commands and/or GASP tweens.

The SVG properties that can be dynamically changed include:

* x, y, position
* x, y, scale
* Rotation
* Opacity
* Fill and stroke color
* Stroke width

**UNO Properties**

It may be useful for ease of coding for each UNO to have properties for the visual parameters that can be changed, eg.

* uno.xpos, uno.ypos
* uno.xscale, uno.yscale
* uno.angle
* uno.opacity
* uno.fillcolor, uno.strokecolor
* uno.strokewidth

or perhaps these are not really needed ? These could be used to store the initial values, and thus used to return the map to it’s default state? (although that’s always easy enough to do by reloading the map).

### Changing UNO / SVG Parameters

There are a two of ways that the SVG appearance can be dynamically changed in the browser:

1. **Database CSS commands** - the CSS commands in the DB allow CSS changes to be applied when the UNO is turned on/off opened/closed.
2. **‘Dynamic SVG functions’** - an extensible set of URL commands that change the SVG parameters. These function can be written to use the GASP tweens to make the changes - or not.

### Specifying the ‘Dynamic SVG Functions’

An extensible set of SVG manipulation functions are specified in a javascript file (eg.modDynomator.js). URL commands for those functions start with the function name and comma-separated parameters.

Custom function that will run user-defined functions:

**run**=functionName, p1, p2, …

e.g a custom shifting tween:

**run=**shiftBounce, unoA, 0.2 100, 100

Standard tween functions:

**move**=unoid | CSS-selectors,duration,relative,xpos,ypos done

**scale**=unoid | CSS-selectors,duration,relative,xscale,yscale done

**fade=**unoid | CSS-selectors,duration,relative,opacity done

**rotate**=unoid | CSS-selectors,duration,relative,angle done

* ***unoid*** is the id of the group.
* **CSS-selectors** are name of CSS IDs or class names (see syntax below)
* ***duration*** is duration in seconds e.g. 1.4
* ***relative*** value is t (for true) or f (for false). If true then the element is tweened relative to it’s current location or state.

The do-it-all function, when you want to do a number of things at once:

**tween** = unoID | class, duration, relative, xpos, ypos, xscale, yscale, opacity, angle

The syntax for CSS selectors in tween commands:

* Single quotes around the CSS selectors
* The @ symbol as a prefix for IDs (instead of the CSS standard #)
* Period ‘.’ as a prefix for classes
* Commas between multiple specifiers
* Greater than symbol ‘>’ between classes as a child selector
* A single space character between classes as a descendant selector

**Note:** The **child combinator,** the greater-than symbol >, separates two selectors and matches only those elements matched by the second selector that are direct children of elements matched by the first.

<https://developer.mozilla.org/en-US/docs/Web/CSS/Child_selectors>

By contrast, when two selectors are combined with the [**descendant selector**](https://developer.mozilla.org/en-US/docs/Web/CSS/Descendant_selectors), the combined selector expression matches those elements matched by the second selector for which there exists an ancestor element matched by the first selector, regardless of the number of "hops" up the DOM.

A **descendant combinator** — typically represented by a single space ( ) character in the form of selector₁ selector₂ — combines two selectors such that elements matched by the second selector (selector₂) are selected if they have an ancestor element matching the first selector (selector₁). Selectors that utilize a descendant combinator are called descendant selectors.

<https://developer.mozilla.org/en-US/docs/Web/CSS/Descendant_selectors>

**Note on selectors:** that the code prepends vvv to the uno id of a vvv class, and ccc is appended for the ccc class. For example, for the uno is = ‘A’, you could select the ccc element with '@cccA' or '@vvvA > .ccc' or '@A .ccc' (but then in this latter case you would select all .ccc descendants of #A).

The ‘unoid’ parameter passed to the tween functions can operation on the whole UNO – using just the UNOid – or it’s parts by using: oooUNOid, cccUNOid, vvvUNOid.

In addition, other artwork with a specified class can be tweened by using the class name in place of ‘unoid’.

### Loops & how to stop them

We’ll also want to add some functions that loop indefinitely, and we’ll just need a URL command to stop them, and be sure to automatically terminate them when the UNO goes off/closes.

other ideas from greensock: from and fromTo: <https://greensock.com/get-started-js>

Loop tweens would be things like:

**rotateLoop** =unoA,duration,relative,startangle,endangle

Would rock back-and-forth if endangle is < 360, if endangle=360 it should spin.

**beat**=unoA,duration,startscale,endscale

eg. beat=unoA,2,80,120 80%-120%-80% etc. scale from center in 2 sec.

### Calling the Dynamic SVG functions

There are a number of ways that the dynamic SVG parameter can be called to update the map. The current options include

* **Directly** entering the URL command in the address bar
* Calling the command from a **link**
* Using the **database  
   onURL offURL openURL closeURL**   
  fields - this is could be used for some purposes, but probably won’t work for one of the main purposes: using tweens to show/hide, open/close the UNO. So we probably really do need the new DBfields:
* **New fields in the database** - specified for each UNO:  
   **onTween offTween openTween closeTween**

In addition, we may want to add the following capabilities:

* **Click-and-drag** - would activate a vertical or horizontal slider function that would change some parameter in real-time -- e.g the UNO’s scale.
* **A form** to input new parameters, and then calling the appropriate dynamic SVG function on submitting.

### Dynamic SVG functions & on/off open/close

A key application of the tween functions will be to use them to show/hide thing when UNOs get turned on/off open/closed. So we’ll need to think-through exactly how that should happen. I believe what we’ll essentially need to do is replace the ‘default’ show/hide function we currently use ( i.e. CSS display: none and opacity: 0.0 ) with a designer specified function.

One possibility: if we get the timing right:

**Showing**: the existing CSS class method could be used to show the UNO, but the tween function (if there is one) is able to run *before* the UNO becomes visible (to the user).

**Hiding:**, the tween would need to start and finish *before* the UNO is hidden by the CSS.

Unlike some other simple tween functions, the tweens used to show/hide things must have a starting state and an end state, e.g. a fade-in function would have the hidden-state, opacity = 0.0, and the shown-state, opacity = 1.0. Likewise a fly-in would need a starting position and an end position ( the end-position would likely be as specified in the SVG). So we’ll need to think about how those start/end parameters get set.

**--------------------------------- OLDER IDEAS --------------------------------------------**

### Transition UNO Commands

One application of this feature would be to create a ‘pointer’ UNO that could be move/sized/rotated around the map to point-out different elements in presentations or animations.

The tweened transitions should start with the UNOs current values and transform to the new values.

The commands should follow inheritance, so that all children are transformed relative to the parent. For these reasons the **move** transforms should be applied by a relative shift i.e. rather than *moving* the UNO and all its children to the absolute coordinates xpos, ypos, the UNO and all it’s children should be *shifted* by (xpos- unoA.currentX), (ypos - unoA.currentY)

The **shift** command takes the UNO and all its children and shifts them by the xpos, ypos amounts.

These commands should not affect the UNO’s on/off open/closed state (so potentially you could move an object that is off, and it would appear in the new position when turned on).

Other transition functions could include:

**Draw** - reveal the (line) object(s) from the start to endpoint(s). Using the GSAP drawSVG plugin: <http://codepen.io/GreenSock/pen/jEEoyw>

<http://codepen.io/GreenSock/pen/qEdoRE>

Expand move objects from ooo center pt to their specified coordinates

Scale 0% to 100% scale from center

H-scale 0% to 100% horizontal scale from center

V-scale 0% to 100% vertical scale from center

L-scale 0% to 100% vertical scale from left edge

R-scale 0% to 100% vertical scale from right edge

T-scale 0% to 100% vertical scale from top

B-scale 0% to 100% vertical scale from bottom

The on/off transitions will be applied to the immediate content of the UNO - including the ooo, sss, and any other UNOs. The open/close transitions will be used to show/hide the ooo-group and the vvv-group (including the ccc and sss groups) - the openTween being used to show those groups, and the closeTween to hide those groups.

URL Commands could tween groups. One of these commands will roll-back a tweened group(s) to their default state (location, scale, turning off a tween loop, etc.).

### Loops

Map elements need not be static. For example, connections could be shown with a looping animation that would indicate the direction of flow. Or vvv-groups could contain subtle animated elements that help indicate they're turned on.

Loops work similarly to Transitions, except that the animation loops indefinitely. There is a library of simple loop-animations that can be applied to the map object. The set of standard loop effects:

**Name Effect**

Fade 30%-100%-30% opacity

Flow Send highlights along the path, see: <http://codepen.io/GreenSock/pen/qEdoRE>

Beat 80%-100%-80% scale from center

R-Rotate rotate 360˚ clockwise around center

L-Rotate rotate 360˚ counterclockwise around center

Rock rotate -15˚ to 15˚ to -15˚ around center

Rock-Bottom rotate -15˚ to 15˚ to -15˚ around bottom

Swing rotate -15˚ to 15˚ to -15˚ around top

Loop effects can be applied to UNOs either by specifying them in the database or through a URL command. If the UNO should always have a loop effect, if should be applied via the database. If the loop is just a temporary effect, e.g. used to temporarily highlight certain objects, then the URL command can be used.

Loops for UNOs can be set in the database fields **whileOn** and **whileOpen**.

The loop will play until the UNO is turned off or closed i.e. loop effects applied via URL are not remembered: if the UNO is opened or turned on again, it will not have the loop effect.

### Custom Timeline Animations

Any custom timeline animation could be written in modDynomator.svg and run via a dynamic svg command.

### Highlight Tween – this scheme on hold.

Let’s try LM’s idea of fading everything *except* the specified unoid or class:

**hlt** = unoid | class, duration, relative, opacity

Where opacity is the opacity of the *faded* artwork, and unoid | class gets an opacity of 1.0.

Here are some notes on what I have done. Following are two commands. One to turn on highlighting for an UNO id (classes are not implemented yet). The 2nd command unHighlights or resets the highlight. This is because the highlight command sets in-line opacity that does not go away until the unHighlight command tells it to. I may be able to highlight additional uno’s/classes with each additional highlight command, but right now calling that command resets everything and only highlights the uno sent in.

<http://45.55.23.31:3000/demo/things2.html#/?all=on&run=highlight,X>

<http://45.55.23.31:3000/demo/things2.html#/?run=noHighlight&all=on>

While something is highlighted, user interaction may not work correctly because the in-line opacity may be messing things up.

Also, I need to tweak the logic that decides what to set the opacity on. I see some elements are not set.

The default is to highlight all descendants of the element being highlighted.

----------

What if …. On reading-in the SVG file we assign each svg bit a class of .all. Then, the command to highlight unoid = thingA would just be a something like:

#/?+++&fade=.all,0.8,f,0.2&fade=thingA,0.8,f,1.0

And the unfade command:

#/?+++&fade=.all,0.8,f,1.0

Assuming that the thingA’s class is more specific than .all, I don’t see why that wouldn’t work with the existing code. And we could of course encapsulate the two calls into one URL command.

Of course this doesn’t work because it runs into the opacity non-override-ability problem.

# URL Specification

ON HOLD - until Larry has the will to face it again - and until we mull over that this is indeed how it should work.

**Development Notes:**

It’s important to get the URL working exactly right, but what we have now works OK, so this is not an immediate priority - as I’m assuming medium to high difficulty. If it’s relatively easy, we should do it sooner rather than later.

***Terminology note:*** *the term ‘parent’ is inclusive of all levels of parent (grand, great-grand, … ), and ‘child’ or ‘children’ includes all levels of children (grand, great-grand, …). To refer to a just a single generation we use ‘immediate parent’ or ‘immediate child’.*

The default for all UNOs is off/close. The map starts with a blank sheet, and the adds/subtracts whatever’s in the URL.

Hover events do not change the URL.

Internally, the code tracks (remembers) the current on/off, open/close, state of every UNO - whether the UNO is visible or not. Thus, for example, when a parent UNO is closed, any children that are open will be remembered, and those children will show as open when the parent is opened again.

**The URL always shows the list of currently visible, on or open UNOs.**

The one exception to this rule is that children UNOs (in the main body of the UNO, or in the vvv) are not explicitly listed as ‘on’ if they’ve been turned on by their parent (this makes the URL much shorter/simpler than if we listed every UNO that is on).

If an ‘on=’ command appears in the browser, it will usually be the result of a command turning on a specific UNO outside of the parental hierarchy (not using the parent to turn on the child UNO). Likewise, if an ‘off=’ command appears in the browser, it will usually be the result of a command turning off a UNO to override the parental on. Therefore, an ‘on=’ and ‘off=’ in the URL will be treated differently than an ‘open=’ when processing on/off open/close events (see “Updating the URL” below).

*At any point while browsing the map, the URL should capture the current view of the map, i.e. that URL can be opened in another browser window and will present the exact same view of the map (browser window aspect ratio aside). In addition, going back/forward in the browser history will always present the correct historical view.*

### Lock-On Function

Holding down the ‘l’ (for lock) key and clicking on a UNO will add an ‘on=’ command for that UNO to the URL (that l-click will not change the open/close state of the UNO, like a plain click would).

For example, l-clicking on the UNO “ThingA”, will add ‘&on=ThingA’ to the URL (equivalent to executing a ‘+++&on=ThingA’ command). This will ‘lock-on’ the UNO, preventing it, and any on/open children*,* from being hidden when the parent UNO is turned off or closed – as per the logic outlined below in “Updating the URL”.

### Updating the URL

To keep the URL aligned with the current state, and to maintain the proper browser history, the code needs to update the URL on any UNO on/off open/close event:

– **Direct** **On -** When a UNO is turnedon directly via a ‘+++&on=’ or ‘&on=’ command, and not from the parent being turned on or opened:

**An ‘on’ command is added to the URL** for the that UNO.

– **Parental On** – When a UNO is turnedon by a parent:

An ‘on’ command is **not** added to the URL for the that UNO.

– **Direct Off** - When a UNO is turnedoff from a direct off command, and not from the parent being turned off or closed:

**An ‘off’ command is added to the URL** for the that UNO.

If the URL contains an ‘on’ command for that UNO,that **‘on’ command is removed from the URL**. The on command *is overridden* by the *direct* off command.

Then follow the same procedure for Close.

– **Parental Off** - When a UNO is turnedoff by a parent:

An ‘off’ command is **not** added to the URL for the that UNO.

Then follow the same procedure for Close.

– **Open** - when a UNO is opened:

**‘open’ and ‘openall’ commands are added to the URL** for the UNO.

**‘close’ commands are removed from the URL** – if the URL contains an ‘off’ command for the UNO, it will be removed from the URL.

**‘off’ commands are removed from the URL** – if the URL contains ‘off’ commands for the UNO or any child UNOs, they are deleted from the URL. Thus the on/opening of a parent ***will override*** a direct ‘off’ command, as it will be assumed that when a UNO is turned on or opened we really want to see what it contains – even if some previous command hid things.

**‘on’ commands remain in the URL** – if the URL contains ‘on’ commands for children UNOs, they remain in the URL – even if they are redundant, as we assume something else turned that UNO on, and we’ll want it to remain on even if the parent UNO is turned off or closed.

– **Close** - when a UNO is closed, or turned off:

**‘open’ and ‘openall’ commands are removed from the URL -** for the UNO ***and*** any children that are open**.**

**‘on’ commands remain in the URL** – if the URL contains ‘on’ commands for the UNO or any child UNOs, they are left in the URL. Thus the closing of a parent ***will* *not* *override*** a direct ‘on’ command, as it will be assumed that the UNO was turned on by another object/command, and we want it to remain on.

**‘off’ commands are removed from the URL -** if the URL contains ‘off’ commands for any child UNOs, they are deleted from the URL – as they will be redundant, and would be overridden anyway by a subsequent on or open.

**Redundancy Checking** - for all the operations above, the URL should be checked for duplicate commands.

# Website / Hosting Tasks

### Perform Pre-processing on SVG at time it is uploaded

Do flattening and error-checking and any optimization of svg at time it is uploaded instead of every time the page is opened.

The preprocessor should check for illegal tags (eg opacity in UNOs/vvvs groups), and other potential errors: duplicate UNOs, UNOs w/o DB entries, etc.

### Process for re-hosting maps

a concise set of instructions – and a cleaned-up / cleaned-out file/folder structure – so we can hand-off the maps to others to host.

### Database Schema - see LM Video

Larry to fill-in MC on how/where the database schema gets set - it will need to be modified on a regular basis moving forward.

Also - we might want to create a separate .js file that hold the function(s) that pull data from the DB, format it, and put it in the info-pane. That is likely to change on a per-map basis. The alternative method is to do it all with Excel formulas - filling-in the long description field as needed.

The database schema is documented here. New fields still need to be documented: <https://docs.google.com/spreadsheets/d/1NKj49bkZAbVvzpuKcE_3md1vtK2_n71z1jRwGkNyiNk/edit?usp=sharing>

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Future Development

Things to potential move to the live To Do list:

### Database Replace vs. Add-to

It used to do add-to, then we changed it to replace, but it would really nice to have both. Add a radiobutton on the importSpread page to toggle between Replace vs. Add-to the DB.

### Revisit Highlighting

An UNO may be highlighted by the URL command:

**hlt=thisUNO**

Which will highlight that UNO *and* all its children. The ULR command:

**hlt=off**

will turn off all highlighting and remove of ‘hlt=’ commands from the URL (but will not show up in the URL itself).

An i-click on an UNO will toggle highlighting for that UNO, adding/removing a hlt=thisUNO to/from the URL. Multiple UNOs may be clicked on an highlighted in this way. An i-click somewhere NOT on any UNO will trigger the equivalent a hlt=off command, turning off all highlighting.

------

Use CSS to dim everything that’s not flagged as highlighted? Investigate how it’s done on kumu.io?

**NOTE:** Using opacity to dim non-highlighted things will not work because it makes the artwork transparent; which makes is not the effect we want. I tested the CSS filter { filter: brightness(2.0); } - which also doesn’t produce the right effect and is only supported for SVG by Firefox.

An non-CSS approach:

Add a topmost group ‘highlighted’ into which we can add/remove non-interactive *copies* of the UNOs specified by the commands: **hlt=thisUNO**

Just below the ‘highlighted’ group put a ‘dimming’ group that contains a map-sized white rectangle with the an opacity set the by the URL command: **hlt=0.8**

If there are any UNOs in the ‘highlighted’ group, then the highlighting groups are on. The highlight groups will be off if ‘highlighted’ is empty.

The ‘highlighted’ group can be emptied by the URL command: **hlt=off**

### Database Editing/Table

Connect Google docs spreadhseet to MongoDB?

Or use one of the js spreadsheet libraries? - <http://jspreadsheets.com>

Or react.js to link to the database.

### DB/SVG Error Checking

Agreed that we should not allow duplicate UNO IDs - but it would be good to check for SVG/DB errors like this:

* Duplicate IDs in the DB (although we allow for duplicate *overrides* in the override DB)
* Duplicate IDs in the SVG
* Notification for missing DB entries

Currently these things will cause problems with the map, but do not throw errors, so the designer has to hunt down what’s going on.

### On/Open by Zoom

**Development Notes:** This would be cool - and not too difficult to implement?

This feature allows more detailed content to appear automatically as the user zooms in.

The zoom point at which the UNOs are turned on/off, or open/close, are set in the database by the fields:

**zoomOn zoomOpen**

The feature can be enabled/disabled by a global boolean: AutoOnZoom.

Only items that are within the current view port should be turned on/opened, but they should be turned off/closed if they’re visible or not.

This code logic is something like this:

// Whenever the pan or zoom changes, use callAllGroups(?) to check each UNO with this function

// Only if the feature is enabled by the global AutoOnZoom

var AutoOnZoom;

if( AutoOnZoom == true ) {

doOpenByZoom();

}

function doOnOpenByZoom(){

// Turn the UNO on/open only if its within the current viewport - but turn off/close wherever it is

// the UNO method checkVisibility determines if the UNO is within the viewport and sets isVisible

// Only do auto on/off if the zoomOn value is a number > 0

If (thisUNO.zoomOn > 0){

thisUNO.checkVisibility();

if( thisUNO.isVisible ) {

if( thisUNO.zoomOn < currentZoom ){

if(thisUNO.on != true){

thisUNO.on = true; // or whatever needs to be done to turn the UNO on

thisUNO.onByZoom = true; // track if the UNO has been turned on by auto-on

}

}

}

}

else if(thisUNO.on == true){ // we're below the zoomOn value, and the UNO is on

if( thisUNO.onByZoom == true){ // if it was turned on by zoom, then turn it off

thisUNO.on = false; // or whatever needs to be done to turn the UNO off

}

}

}

// Do the same for open/close

If (thisUNO.zoomOpen > 0){

thisUNO.checkVisibility();

if( thisUNO.isVisible ) {

if( thisUNO.zoomOpen < currentZoom ){

if(thisUNO.open != true){

thisUNO.open = true; // or whatever needs to be done to turn the UNO on

thisUNO.openByZoom = true; // track if the UNO has been turned on by auto-on

}

}

}

else if(thisUNO.open == true){ // we're below the zoomOn value, and the UNO is on

if( thisUNO.openByZoom == true){ // if it was turned on by zoom, then turn it off

thisUNO.open = false; // or whatever needs to be done to turn the UNO off

}

}

}

}

// Elsewhere in the code: if ***anything else*** turns the UNO on set thisUNO.onByZoom = false

// This prevents something a user deliberately turned on from turning off when zooming out

thisUNO.onByZoom = false;

// Elsewhere is the code: if ***anything else*** opens the UNO, set thisUNO.openByZoom = false

// This prevents something a user deliberately opened from closing when zooming out

thisUNO.openByZoom = false;

### Tear-off Panes

Enable separate sub-windows for the story and info panes.

### **Database Schema**

to be automatically created from the imported spreadsheet’s first row.

### Folding

I’m assuming we can accomplish the folding trick with clever use of an UNO containing a jjj-group. Possibly by using CSS styles / padding ?

Fold animation - Smoothly animating as the existing artwork is moved right/left / up/down and the new artwork's width/height animates from 0 to full.

### SVG/Viewport Wrapping

Allow the SVG to wrap-around and thus provide infinite scrolling horizontally, or vertically, or possibly both. See:

<http://stackoverflow.com/questions/15560174/vertically-wrap-around-scroll-in-jquery>

<https://www.exratione.com/2011/01/a-low-tech-approach-to-wraparound-scrolling-lists/>

<http://jsfiddle.net/simsketch/aj5t2m1k/>

<https://github.com/janpaepke/ScrollMagic>

### Enable loading of local files – Low priority

Only if this is an easy modification - otherwise leave it for future work - allow the story=, map=, info=, commands to point to local files, e.g. this command would work:

story=file:///Users/mclemens/Dropbox/Systems Agency/SMP dev/dev-site/demo/story.txt

This would allow users who don’t have server access to create their own local presentation file.

### Implement jjj commands ? ?

Needed ? ?

Should this be implemented by using a database field to hold the JSON?

Do we need this for… voiceovers, animations, other tricky stuff ? Once I get the voice overs working, it will be more clear what to do here. I see if/then statements, multiple lines of command, maybe putting things all together in one are such as rrr. Tricky stuff like statistics -- where the user clicked and what order and how long they lingered on a subject.

### Tooltips

#### Position-Override

With the tooltip position fixed, and the map zoomed to fill the pane, the tooltip for objects on the edge of the map fall partially outside the map pane – and are unreadable without panning or zooming. To fix, implement a position-override, the will always keep the tooltip with the map pane.

#### Tooltip Centering

It’s currently not possible to get the tooltip to reliably center on the UNO. For example, you can set ttPosition to bottom, and ttLeftOffset to an appropriate value *when viewing the full map*. But if you zoom in or out, the tooltip will no longer be centered because the ttLeftOffset in no longer at the right scale.

#### Left Tooltip Position

- setting the tooltip to ‘left’ is still not working right. Adding an offset value to correct this is unreliable - as the tooltip will appear in various positions depending on the size of browser window on loading.   
The left position varies on zoom. Let’s not spend any more time on fixing the left position now.

### GSAP pan/zoom - On Hold for Now

Larry trying a quick ‘n dirty socking green experiment:

* How difficult is it to change over?
* Scroll wheel & trackpad works?
* Simpler? Simpler to do tweening of groups. Not simpler with Safari.
* Faster? Yes
* iOS / Android compatibility? Yes

Basics of pan/zoom working in GSAP. Single zoom value works. Not working:

* Buggy click-drag pan on Mac Safari & Chrome: you can drag if you click on the map, but clicking outside the map has iffy results - clicking LR of the map works, clicking UL of the map not so much.
* Reset key works in Chrome but not in Safari.
* Tool tips are underneath the map.
* Mouse/trackpad zoom doesn’t zoom to center on mouse arrow - the zoom tries to keep the upper-left corner of the SVG in the map-pane.
* Zoom keys do not zoom to the center of the view - again, the zoom is fixated on the map UL corner.
* Absolute coordinates not calibrated.
* Zoom in/out keys reversed.
* Arrow-to-pan keys reversed.
* Shift-click not working.
* Option-click on UNO to openall/closeall not working
* No panning physics.
* iOS - iPad/Safari - zooms usably fast. Clicking works well, pan/zoom not at all - which reinforces my suspicion that that there’s special magic need to differentiate the multi-touch gestures.

**Previous svg-pan-zoom Fork Working** - much better now with the new code that uses callbacks.

### Pan/Zoom Performance

Did a bunch of testing 04/ 09-13 /16. Our pan/zoom performance is actually OK - give the number of vectors being zoomed. Outlining text is key.

Basic zoom performance - on 1000 filled/stroked circles - is very fast.

Test done on subject pan/zoom speed in Safari (which is relatively slow), using different versions of SAMap.

Text rendering makes a huge speed hit. Outlined text is much faster - and avoid the issues with kerning etc not showing up correctly when saved to SVG.

Test files / cases:

SA Map - outlined in AI

Useable fast in Safari.

SA Map - outlined by save as SVG

For some reason, the outline text feature in the Save As SVG dialog is missing some of the text object, which makes the resulting SVG a bit slower then when the text is outlined in with shift-O.

SA Map - text outlined and ungrouped

No a noticeable improvement - maybe worse?

SA Map - text only flattened

Very fast/smooth!

SA Map - text only

SA Map - no text

SA Map - no outline optimizeSpeed

SA Map no outline

SA Map flat

This has text outlined, and then the artwork is ungrouped and flattening into ~one layer (so there are no UNOs and no interactivity). It is fastest of all; showing that the large number of groups do slow things down, although it’s not as a dramatic as the non-outlined text.

SA Map - outlined in AI - all in sss

Moving all the artwork into an SSS layer makes no difference - so we know that the interactivity code isn’t slowing down pan/zoom.

**SA Map - without Tweenlite**

Does the pan/zoom with svn-pan-zoom.js instead of GSAP. Snappier than using GSAP - desirable if performance I paramount e.g. On slow hardware or browsers.

**SA Map - display none test**

In Chrome and (especially) Safari, hiding things with display: none is better, faster. In Firefox (which is way way faster now with test outlined - and the font are with because they’re outlined), it doesn’t make any difference because Firefox cleverly treats opacity: 0 just like display: none.

Comparing zoom with Tweenlite vs. svg-pan-zoom - the svg-pan-zoom is a bit better - smoother, less-jumpy - than tweenlite (with timing set to 0.1). But tweenlite has the ability to do smooth transitions (with the timing set to ~0.4).

**Conclusions**

• Outlining Text - is essential. Using the outline text in the Save As SVG dialog may work - but only if we figure out why it’s missing some text.

• Recognizing the SVG groups as interactive items does not slow things down.

• Using display: none to hide objects does help - in Safari and Chrome - but not Firefox (as its smart)

• Using GSAP Tweenlite has the nice transitions, but it’s not as snappy as using svn-pan-zoom.js to do the pan/zoom’ing.

**Font Tests**

Simpler font shapes, when outlined, will produce fewer vectors thus smaller files and less load on the browser’s graphic engine.

With the 1000 label test file: labeltestX.ai/svg I tried a few different fonts; outlining and saving to SVG w/ 2 decimal points:

Open Sans Semibold: 9.0 mb

Myriad Pro Semibold 6.6 mb

Helvetica Neue medium 15.3 mb

Helvetica Bold 9.9 mb

Avenir Medium 14.5 mb

Source Sans Pro Semibold 6.5 mb

And as you expect, comparing pan/zoom speed of Myriad Pro vs. Helvetica Neue, in Safari, MP is better/faster.

<http://45.55.23.31:3000/demo/zlabeltest1.html>

**Testing with SDGMap**

SDG map seemed slow, even after outlining fonts, so I started to delete things – background fades and fills, text > short labels – but there doesn’t seem to be any problem items; it’s just the number of vectors slowing it down. Which still seems a bit odd give how fast the 1000 circle test is: <http://45.55.23.31:3000/demo/ztest4.html>

But perhaps that’s not too surprising as the circles file is 101K and SDGMap is 5 to 10 mb.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Documentation is here:** [**https://docs.google.com/document/d/1S96eKGRuMTMOTlJ\_hbuhQ9kccirjvhZDOYIauS6ddOA/edit?usp=sharing**](https://docs.google.com/document/d/1S96eKGRuMTMOTlJ_hbuhQ9kccirjvhZDOYIauS6ddOA/edit?usp=sharing)

# **O**bject Specification

* **Uniquely Named Object - UNO**  
    
  Objects created in Illustrator – and exported to SVG – are associated with objects in the database through the object name; which must match exactly. Illustrator may append a “\_#\_” to the id, e.g. the group named “uniqueObjectName” in Illustrator may become:   
  <g id=”uniqueObjectName\_3\_” >  
   in the SVG file. But this appended number will be ignored when the SVP parses out object names from the SVG file.  
    
  UNOs will typically have a database entry, although this is not strictly required.  
    
  UNOs can contain special svg-groups identified by the prefixes: ooo, vvv, ccc, and sss. See the UNO specification below for details.  
    
  A ‘Parent’ UNOs can contain other ‘children’ UNO’s. The parent-children relationship can be repeated/ nested an unlimited number of levels.
* **Stacking Order**

Artwork always maintains the stacking order as set in the SVG file. So, for example, the order in which UNOs get turned on will not determine the stacking order. When objects are turned on or opened it will not change their relative stacking order from what is in the original SVG file.  
  
The exception to this will be if an UNO is opened and onDimming > 0, in which case the vvv-group – *and any of the above UNO’s children UNOs* – will need to be brought up in front of the dimming layer.

### UNO Commands

- there are just 4 things that UNOs do – and 4 commands to control UNOs – **on/off** and **open/close**. The open/close command toggles between showing the vvv-group (the ccc is alway contained in the vvv, so it will always be turned on/off along with the vvv.) and the ooo-group.Open/close can be triggered by hover/click mouse events and/or via the URL. On/off can only be triggered by URL. All objects are set to off when the page is first loaded. A URL command can turn on UNO’s.

**on**=thisUNO - shows the UNO i.e. makes everything in the UNO visible *except* the vvv-group. A UNO can only be set to on or off.

**off**=thisUNO – hides the UNO, and all its children.   
  
Any UNO, anywhere in SVG, can be independently turned on/off via a URL command: from the address bar, side-pane, or by triggering an rrr. UNOs can thus be turned on/off independently of the groups that contain them.  
  
If an UNO is set to both “off” and to either “open” or “close,” then the off command gets precedence and everything is hidden.

**open**=thisUNO – hides the ooo-group (the open-UNO trigger) and shows the vvv-group. Open/close is equivalent to how we currently show the vvv-layer, so it will include the white layer if onDimming > 0.  
If an UNO is set to both “on” and “open,” then the “open” command gets precedence and the ooo group get hidden and the vvv groups gets shown.

**close**=thisUNO – hides the vvv-group (and any children) and shows the ooo-group. This essentially returns to an *on* state.

### Mouse Interactivity

**Hovering vs. Clicking** - hovering creates a ‘temporary’ open/close, and clicking creates a ‘persistent’ open/close. The key difference between the temporary and permanent is that mouse focus is only transferred between ooo/vvv on a click.  
  
• hover-on simple UNO (one with no vvv or other groups) = show tooltip and/or info pane.  
• hover-off simple UNO = hide tooltip but leave info pane displayed until it is replaced by another.  
  
• **hover-on ooo-group** = temporary open: hide ooo, show vvv and all of vvv’s other siblings, show tooltip. The ooo-group, although hidden, retains mouse focus i.e. we’re waiting for 2 possible mouse events: a hover-off the ooo, or a click on the ooo (the ccc-group will not get mouse focus unless a permanent click-open happens).  
• **hover-off ooo-group** = close, but only if this is in a temporary open state: hide vvv, show ooo, tooltip off.  
 Note that the UNO can contain other stuff: other objects, sss-groups – but its only the ooo-group that responds to mouse events (only in a really simple object with loose artwork and no other groups would it be right to to talk about hovering on the (whole) UNO.)  
  
• **Hover-on ccc-group** = temporary close: hide vvv, show ooo, show toolip. The ccc-group, although hidden, retains mouse focus i.e. we’re waiting for 2 possible mouse events: a hover-off the ccc, or a click on the ccc (the ooo-group will not get mouse focus unless a permanent click-close happens).   
• **Hover-off ccc-group** = persistent open: hide ooo, show and highlight vvv, tooltip off.  
  
• **click-on ooo-group** = persistent open: hide ooo, show vvv, turn off tooltip, prevent hovering on any other object besides ooo until ooo is hovered-off of (but do not close when hovering off ooo in this case), *then* shift mouse focus from ooo to ccc, update URL.  
  
• **click-on ccc-group** = persistent close: hide vvv, show ooo, turn off tooltip, prevent hovering on any other object besides ccc until ccc is hovered-off of, *then* shift mouse focus from ccc to ooo, update URL.

### Mouse Interactivity With *hoverOpenClose* Set to *open*

**Hovering** - hovering creates a ‘temporary’ open/close.  
  
• **hover-on ooo-group** = temporary open: show tooltip, info-pane. All visible groups retain mouse focus.  
• **hover-off ooo-group** = close, but only if this is in a temporary open state: hide vvv, show ooo, tooltip off.  
 Note that the UNO can contain other objects: sss-groups – but its only the ooo-group that responds to mouse events (only in a really simple object with loose artwork and no other groups would it be right to to talk about hovering on the (whole) UNO.)  
  
• **Hover-on ccc-group** = show tooltip and or info-pane - if set in the DB .   
• **Hover-off ccc-group** = do nothing.

### UNO State Table

The UNO can have 4 states:

**Open** or **Closed** - these are the ‘persistent’ states, where mouse control is transferred to vvv (when open) and ooo (when closed). The URL open/close commands create these states.

When the map is first loaded all UNOs are off and closed. When a UNO is turned on, it is shown with it's current open/closed state. If the UNO is turned off, it's open/closed state doesn't change. For nested objects, the open/closed states flow down the generational hierarchy.

**Temporary Open** or **Temporary Close –**these are used to handle hover events, where mouse control does not get transferred between ooo / vvv.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Action** | **state name** | **persistent?** | **What is hidden** | **What is shown** | **what is highlighted** | **Show tooltip?** | **Who has mouse control?** |
| hover-on ooo-group | temporary open | no | hide ooo | vvv and the contents of the UNO minus ooo | vvv and the contents of the UNO minus ooo | yes | ooo minus the sss. CCC has no mouse events, including clicking or hovering, only ooo, until hovering off ooo. |
| Hover-off ccc-group  click on ooo  URL open command | persistent open | yes | hide ooo | vvv and the contents of the UNO minus ooo. | vvv and the contents of the UNO minus ooo. | toggle off then on (need to simulate a hovering off event for ooo) | ccc-group  **click-on ooo-group** = persistent open: hide ooo, show vvv, turn off tooltip, prevent hovering on any other object besides ooo until ooo is hovered-off of, then shift mouse focus from ooo to ccc, update URL. |
| Hover-on ccc-group | temporary close | no | hide vvv | ooo and the contents of the UNO minus vvv. | nothing | yes | The ccc-group, although hidden, retains mouse control i.e. we’re waiting for 2 possible mouse events: a hover-off the ccc, or a click on the ccc (the ooo-group will not get mouse control unless a permanent click-close happens) |
| hover-off ooo-group  Click-on ccc-group  URL close command | persistent closed | yes | hide vvv | ooo and the contents of the UNO. | nothing | no | ooo-group – hovering on ooo or other objects is possible  **click-on ccc-group** = persistent close: hide vvv, show ooo, turn off tooltip, prevent hovering on any other object besides ccc until ccc is hovered-off of, then shift mouse focus from ccc to ooo, update URL |

• double-click on ooo or ccc or simple UNO groups = run the command that is in the UNO’s ondoubleclick database field.  
  
 If onDimming > 0, there are 2 cases:  
• no vvv-group - dimming not allowed.  
• with vvv-group - the contents of the object (except ooo) AND the vvv get put in front of the dimming layer. On a click the contents of the UNO object (except ooo) and the vvv get highlighted and the dimming are made persistent. If onDimming = 0 then the UNO object remains visible, the vvv object becomes visible but not highlighted, and the ooo group is hidden; this is made persistent until closed with a close command from click on the ccc, or from a URL close command. If onDimming = 0 the stacking order will not be changed from what is in the SVG file.  
  
Note that we now have different open-trigger (ooo) and close-trigger (ccc) groups, and the open/close-triggers get hidden on hover/clicks. Therefore, when hovering, the code will need to make the trigger-group invisible, ***but*** it must still ‘be there’ so that it will get the hover-off event. A click open/close should completely hide the (previous) trigger (from mouse events).

### UNO Specification

 a ‘UNO’ is any SVG group that has a uniquely named id. A UNO does ***not*** need a database entry to be interactive and/or receive URL commands.

A UNO must be a child of either another UNO or a vvv group. All ancestors of a UNO group must be a group element with an id. For example the following is incorrect because the parent of “foo” is a group with no id:

<g>

<g id=”foo”>

</g>

</g>  
  
The UNO may contain one - and only one of each - of special groups identified by these prefixes: **ooo**, **vvv**, and **ccc** (the ccc-group sits within the vvv-group)**.** The code needs to associate these groups with UNO that contains them. The code needs to append the immediate parent name to any ooo, vvv, or ccc groups.  
  
There can also be any number of ‘static’ **sss**-groups that are ignored by mouse events, and that are shown/hidden along with the group that contains them. The javascript should ignore sss-groups, and show/hide them with their parent groups.  
  
An open/close command essentially just toggles between showing/hiding the vvv-group and the ooo-group. Note that hiding/showing the ooo-groups means that it’s opacity is set to zero to hide it, or restored to it’s original opacity to show it. The ooo group is never highlighted. In the case of toggling ooo and vvv - hiding/showing means both visual and mouse events.

### UNOs can contain the following kinds of things - any of which are optional:

#### Loose artwork

* + - – **as the trigger artwork:** *in the absence of any other groups in the UNO*, any ‘loose’ artwork is treated as the UNO trigger. This is useful for creating simple UNOs that have no vvv-group or other groups, and are just used to trigger a tooltip and/or info-pane.
    - – **as just more artwork** – if there are other groups, then any loose artwork, without an ID, is treated as ‘static’ by default. This is how the code works now, where non-ID’d artwork is static and putting it in an sss-group is more of a formality.

#### **ooo-group - open-trigger:**

* + - **<id=ooo>** –collects the UNO-trigger artwork into a group (rather than leaving it loose) – to prevent confusion in more complicated UNOs. This group – and just this group – gets turned off when the UNO is opened, and turned on when it’s closed. An ooo group cannot contain UNO’s, vvv’s, ccc’s or other ooo’s. ooo and ccc groups can contain sss, but otherwise they're just the trigger-artwork.

#### sss-group: <id=sss>

* + - Contains static artwork that is ignored for mouse events. Used to differentiate ‘background’ artwork from the open-trigger artwork. Because it is within the ooo-group, this sss-group gets turned off when the UNO is opened, and turned on when it’s closed. In other words, If you want some static artwork to be hidden on a open-command, then it must be put in an sss-group within the ooo-group. Does not get turned off on an open command.

#### **vvv-group - visible when open:**

**<id=vvv>** – is made visible when the UNO is ‘opened’.   
  
The ooo and vvv-groups are identified merely by the 3-letter prefix, and thus the code needs to associate these groups with the containing UNO so that an open=thisUNO command will hide the correct ooo-group and open the correct vvv-group.  
  
The vvv-group can contain an sss-group: static artwork that does not respond to mouse events, but that will get hidden on an open-command.

The vvv-group is similar to just another UNO, and can contain any of the same things an UNO can. *Except* that is does not have a vvv-group (as it’s vvv equivalent is the ooo-group). And instead of an ooo-group, it’s trigger is a ccc-group, just to identify it as the close-trigger.

#### onDimming

- highlighting (adding an opacity layer) is activated only on an open-command, when the vvv-group is made visible (there must be a vvv-group for highlighting to be activated). If onDimming > 0, then the contents of the UNO, including the vvv-group, but minus the ooo-group, will be brought to the top of the stacking order and placed in front of an opacity layer (with an opacity equal to the onDimming value). If onDimming = 0, the stacking order will **not** be changed from what is in the original SVG file.

#### **ccc-group - close-trigger:**

**<id=ccc>** – artwork used to trigger a close-command. The designer needs to be sure to put this in so that UNOs can be closed. Having separate open & close trigger groups helps to push designers to create different looking triggers so users can see what’s open and what’s closed.  
  
The ccc-group can contain an sss-group: static artwork that does not respond to mouse events, but that will get hidden on a close-command. A ccc group cannot contain UNO’s or vvv’s, ooo’s or other ccc’s.   
  
Again, the code needs to associate the ccc-group with the containing UNO so that an close=thisUNO command will hide the correct vvv-group and show the correct ooo-group.   
  
When hovered-on, the ccc-group should show the same tooltip & info-pane as the ooo-group. (even though we now have 2 triggers, they should work like the current 1-trigger model re showing DB info.)

#### Other UNOs:

**<id=someName>** – a UNO can contain any number of other ‘children’ UNOs. These are just UNOs that behave like any other UNO. The preprocessing code does not have to rename the id’s of these. They're just UNOs have their own specific IDs/names by which they are referred to. Any of these UNOs can be selectively turned on/off or opened/closed.   
  
These children UNOs ***do not* get hidden** when the parent UNO is opened (only the ooo-group gets hidden).  
UNOs can be used merely to hold other UNOs, and thus need not have any trigger artwork or other groups. For example, a UNO called “basemap” could be used just to hold all the UNOs that should be shown on start-up. When opened, “/?on=basemap” would be appended to the start-up URL.

All the DB items apply equally to UNO, vvv, ooo & ccc. It all works as before, but we now have have 2 trigger objects instead of 1.

### URL FUNCTION

The following items are additions and modifications to how URLs work that will give fuller control over specifying the map state.

**Start-up URL**  will be specified in index.html, and that URL will be loaded on page startup, unless the pages is started for the first time with a URL that includes a parameter list of UNOs, i.e.,   
var defaultURL = '#/?on=basemap&panx=0&pany=0&zoom=1.0&wpane=open&epane=open';   
Done

**Browser / Map-State URLs**The the URL in the browser address bar will always contain a specification of the full map state: the full historical path of everything that's on or open, pan/zoom value, pane-state. Done  
  
This ensures that the forward/back command in the browser will work as expected, and it enables one to specify an extract the map state from an external URL.  
  
Thus a URL command sent to the map will show *only* those UNOs in the URL  
parameter list. In essence, when the map gets a URL, is starts with a blank map-window and then adds-in the items in the parameter list. If the parameter list is blank, it will populate the URL with the start-up URL specified in index.html. Note that child UNOs under vvv groups are normally off. A URL command to set them on or open will bring them to the front to display them. Then if their parent vvv is shown then hidden, the child URL will be hidden again. **Update History -**  The code needs to correctly update the browser history when populating a url that has no parameters. Currently the history gets two entries: One for the empty URL with no pan or zoom or pane states, the other with those state values populated. It should only have one entry. Not done

#### Selective URLs +++ Command

URL commands can also be used to *selectively* add/subtract items from the current map state (but otherwise not change the current state). The entire URL will be either a ‘normal’ browser URL (as above), or a selective URL (there’s no mixing of normal and selective in one URL).   
  
For a selective URL, the code will add / subtract the items listed in the selective URL from the current state (unless those items are already on/off open/closed – in which case no action would be needed). The URL will be updated accordingly, so hitting back/forward in the browser will work as expected.  
  
The selective URL is identified by the leading prefix “+++”. For example:  
 #/?+++&open=Individuals&close=AssetOwners&panx100.00&pany=100.00

would turn on Individuals (if it’s not already on), open Individuals (if it’s not already), close AssetOwners (if it’s open), and pan to 100,100. It would otherwise not change the Map state. It would update the full parameter list in the browser URL to reflect these changes.  
  
Note that this URL command:  
 #/?+++&open=Individuals  
would first turn Individuals on (if it’s not already on), and then open it.  
  
A selective URL that is triggered by a hover will not change the browser URL - a click would be needed to change the browser URL.

**Pan-zoom or pane only URLs** - A URL with a pan, zoom, or pane values but with no UNO id’s will pan, zoom, change panes, but otherwise not change the map state.

**Multiple UNOs in Pane Hover** – all UNO in the URL will be shown on a hover (this may be an obsolete issue). Show the tooltip/info-pane for the ***last*** UNO in the list ( unless tooltip=0 ).

**UNO Naming – in the URL** - UNO id names in the URL should not include the “\_#\_” that is appended many times in the SVG file. For example, a group may have, “FinancialSupplyChain\_1\_”. The preprocessing javascript code will remove all of these appended characters. *Designers must keep track of duplicate UNO names.* For example, if a designer creates a duplicate of the “id=Workers” UNO, she needs to name it Workers-2 (with a dash separator). For the purposes of tooltip and info-pane, the code will strip the “-2” and use the DB information for “Workers”. But the URL must specify on=Workers-2.

**URL Syntax**Items in the URL are separated by “&” for example: ?open=Issues&open=Goals&panx=0.00&pany=0.00&zoom=1.00

### URL, CSS, and JSON Commands from the SVG file

Any non-trigger group can contain ‘original’ SVG artwork   
***and/or***  
an rrr-group that contains a URL command  
***and/or***a css-group that contains a CSS command  
***and/or***  
a jjj-group that contains JSON statement for special commands/code

#### rrr-commands <id=rrrS> <id=rrrH>

rrr-groups enable URL commands to be executed from within a map. They are created in Illustrator by giving a simple text object an id=rrrS or an id=rrrH. The text in that group is the rrr command. The text can be given no stroke or fill so it doesn’t show in the map.

rrr commands are used in pairs, with the text object identified by an id=rrrS or id=rrrH:

**id=”rrrS”** (for Show) will be executed when the group containing the rrr is shown – by an on command (if the rrrS is in the UNO body) or an open command (if the rrrS is in the vvv-group).

**id=”rrrH”** (for Hide) will be executed when the group containing the rrr is hidden – by an off command (if the rrrH is in the UNO body) or a close command (if the rrrH is in the vvv-group).

The id can be in either a <g> tag or <text> tag.

The command in the rrr-group will follow the same syntax as browser URLs. The URLs can only target the current map-pane (and are thus path-free).

The rrr commands must be an immediate child of the UNO, or of the UNOs vvv-group. There can be no more than 2 rrr command sets per UNO: one pair (rrrS/rrrH) within the UNO, and/or one pair within the vvv-group.

rrr’s get executed whenever the group that contains them is shown or hidden - along with anything else that would be normally shown/hidden in that group.  
  
Note: The designer only needs to separate parameters with a simple ampersand (&) in Adobe Illustrator. AI will then turn that into “&amp;” when it exports it to the svg output file.

#### CSS-commands <id=cssS> <id=cssH>

CSS-groups enable CSS commands to be executed from within a map. CSS-commands are placed within a UNO and are executed when on on/off open/close events (see below).

The command in the css-group will follow normal CSS syntax. They only target the current map-pane.

The designer will typically use 2 different css commands:

**cssS** (for Show) will be executed when the group containing the is shown – by an on or open command.

**cssH** (for Hide) will be executed when the group containing the css is hidden – by an off or close command.

**Class’ifying UNOs**

Assigning classes with Illustrator can be done by adding one or more class names to the group name with the syntax:   
“objectName class:className1 class:className2 ...”  
  
For example, the UNO named in Illustrator as:  
 **Individuals class:stakeholder class:people**  
which will generate the following SVG:  
 **<g id="Individuals\_class:stakeholder\_class:people">**    
  
The code will parse that into an UNO with an id named “Individuals” (for URL command and DB purposes), with the CSS classes of “stakeholder” and “people”.

Artwork anywhere in the SVG can be assigned a class and controlled via CSS.

Classes are inherited from parent to child groups. Thus a designer to assign a css class to a number of objects by placing them within a parent group that has a css class assigned. For example, one could use CSS class ‘arrows’ to show/hide all connections by putting them in a layer named:  
 **Connections class:arrows**

The UNO’s off/close state takes precedence over CSS commands i.e. the UNO must be on or open for the results of the CSS command to be visible.

#### jjj-commands <id=jjjS> <id=jjjH>

Another special text command that contains a JSON statement. The jjj command is similar to the rrr command, and is created and executed in the same way.

the jjj / JSON commands can be used to do things you can specify with JSON - such as zooming, folding, playing voiceovers, … or other things we think of.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Old Specifications and Finished Tasks

Part of the specification is also contained in the Excel file – Object Data Schema x.x.xlsx – which contains descriptions of the database fields AND what the code needs to do with those fields

### Mouse tip stays on sometimes and hangs the page fixed

* ‘v-groups' - for visibility switchable groups can occur anywhere in the Illustrator / SVG hierarchy - have a ‘vvv’ prefix.
* ‘d-groups' - for default non-switchable groups that are always be visible - have a ‘ddd’ prefix.
* ’s-groupf - for static groups that are always visible but have no interactive elements - have a ’sss’ prefix.
* The ‘vvvName’ group itself is just a container for other objects, and thus has no visibility or interactivity itself.
* Switchable groups get parsed and treated just like any other map content: they get parsed for named IDs, DB information, should respond to mouse events etc.  There is essentially no difference in how ‘vvv’ and ‘ddd’ groups are treated, except that ‘vvv’ groups can have their visibility toggled.
* A switchable group may – or may not – have an associated ‘trigger object’: a group with the same name but no ‘vvv’ prefix.
* Switchable groups can be toggled between a visible and invisible state (using the class designations)
* **Visibility Triggers:** There are various means to toggle the group’s visibility state:
  + The initial visibility states are set when loading the page
  + hover on/off - the trigger object
  + click on/off - the trigger object
  + external events such as a hard-coded buttons in the Legend pane, or for example, by clicking on linked text in the Narrative pane
* **doDimming Variable:** In the spreadsheet / DB, each object now has a doDimming field, that specifies the dimming behavior when the object is triggered. The field can be 0 (for no dimming, 0%, False), or a number 1 to 100 that specifies the % dimming to apply for that object.
* **Duplicate Objects:** A database object - for example ‘Workers’ - may appear multiple times – in different vvv or ddd groups – and at different locations.  When Illustrator creates the SVG file it appends a number to the duplicate object e.g. id=“Workers\_2\_”.  That appended number will  be ignored when looking up the DB information, and when any one of the “Worker” object is triggered, the appropriate information from the Worker entry in the DB will be displayed in the tooltip / pane.
* vvv layer location:  vvv groups can be located on the same level as, or within the same group as, their trigger-object.   The code will flatten these layers.
* Prevent scrollbars from appearing in the map pane – is OK on the fox site.
* **Master & Override Databases** – currently, the database contains information that may be universal – can be used by many different maps – as well as information that may need to be tailored to meet the needs of a particular map. To accommodate this, we need to query 2 databases: the generic ‘master’ DB and a map-specific ‘override’ DB. The logic would be to first load the object’s information from the master DB (if it exists), and then, if there is an entry for that object in that object in the override DB, use any *non-empty* fields from the override DB instead of the master field. The override DB may also contain objects that are not in the master DB.  
    
  The user should have the option of using the master & override, or just the master, or just the override DB. So the user needs a way to specify one, or the other, or both ‘collections' - and the code shouldn’t freak out if only one collection is specified.
* **Creating Maps:**  The data that specifies a particular map will be set in a html file eg mapname.html (currently most things are in index.hbs). Javascript code that controls the map interactivity should be in a different file. For each map the following parameters will be set in mapname.html:
  + Svg file– the svg file to load
  + Story file – markdown text to load to the narrative pane (not loaded from the DB as we do now)
  + Info file – markdown text to load to the information pane (not loaded from the DB as we do now)
  + Databases – specifies the *master and override* Mongo databases to use
  + CSS – the main CSS and UI appearance specifications (currently set to mapWindow.css)
  + UI Library settings – setup for the panes (currently set to idiagram-ui.js)
  + Pane instantiation – is done with the div tags at the end of the file
* **Almost done...** **Double-Click Events –** add DB fields and code to handle double-click events. The default behavior for a double click will be to open a URL. The URL can either use the current browser window, eg to jump to another map, or \_blank to open a new window – eg to an external page. The Database will have 2 fields: “ondoubleclick” containing the URL and an optional target specifier, called “target” – if not specified, the default target will be "\_blank" (a new window). Use the value, “current” to open URL in current window. For example, given the database fields:  
   ondoubleclick: http://www.example.com  
   target: \_blank  
  The code will execute the link:  
   <a href="http://www2.deloitte.com" target= "\_blank" ></a>
* **Hover, click, and double-click events from the pane text**  – hover and double click – and add pan/zoom to make visible
  + *The ability to associate text in the panes with an object ID. The text then will act as a remote version of the map object – sending hover, click and double-click events to the map pane.*For example, we could tag the text “asset owners” in the narrative pane, and when hovering on/off that text the tooltip for the object “Asset Owner” would be shown/un-shown in the map.  And a click or double-click on the text would trigger the appropriate asset “Asset Owner” click or double-click event.
  + A hover will only highlight one object, instead of many levels deep (we may add that feature in the future). The anchor could include multiple id's, but only the last one will be highlighted. If it is clicked on, then we can make them all show up like it is doing now. We’ll usually just want to highlight the one particular object anyway.  
    Note: We are discussing highlighting all objects in the parameter list in the anchor tag. If this is too slow then we can decide if we want to return to highlighting only the last item in the parameter list. Before these objects are highlighted, we will make a copy of the current state. When hovering off, we will return to the previous state.
  + When hovering, make a copy of the object(s) and it’s associated vvv objects and stick them in front of a new white layer; when hovered off, it simply gets deleted and we return to the previous state as documented above.
  + When hovering, dimming - or not - it should follow the same behavior as the map i.e. it should look to the onDimming variable to decide if / how-much dimming to apply.
  + To do this right, we should also pan to make the object visible - only if it currently isn’t visible. Probably better not to center on the object of change the zoom level – so as not to undo the user’s preferences.
* **Mostly done (?) … External Links to the Map & Map State:** add pan & zoom values to the URL if that’s relatively quick and easy to do – otherwise, add later.  
  A link from an external page can be used to open the map page with settings for the state of the map. This will also allow us to hit the back button in the browser to go back to previous states. The ‘state’ of the map will specify the following parameters
  + **Pan** – the center point of the map display
  + **Zoom** – the zoom level
  + **vvv groups** – that will be turned on
  + **Info pane** filled for the active group.
  + **Pane settings in URL** – Capture in the URL the story and info pane settings: open/closed, open width, toggled or slid open.
* **Override tooltip variable in URL** – allow the URL to suppress/show the tooltip. -- Done.
* **Load info pane on hover** – set the info pane to load either on a hover or on a click – selectable via a variable set in index.html (not in the DB). Triggering on hovering will be problem with tagged-text in the info pane – address this conflict by not allowing the info pane to be populated while hovering in the info pane. Currently it will be updated while hovering in the narration pane.). **Map**
* **zooming speed / stutter** – zooming isn't very smooth compared to this CSS example: <https://kumu.io/WorldGreenBuildingCouncil/cop-buildings-day#map-KrS5oW3h/elem-AUovVmXd>  
  Done.
* **Tooltip Position** – add field(s) in the DB, and new code, to specify the position of the tooltip: top, bottom, left, right, and an offset value. DB field to be created based on how the tooltip positions is set in the code. **Done**
* **Hover Interrupts Panning**  - would be helpful to disable hovering while click ‘n dragging. - Done
* **Page Title –** gets set to the clicked-object, but with a pre-pended “IDiagram” instead of the map title as set in map initiator .html file.Done
* **Capture Pan & Zoom in URL** – (new feature) in order to set-up/capture zoom & pan URLs, it would be nice to be able set the URL without having to click on an object with a vvv-group. A **shift-click** will fill the URL with just the current zoom/pan coordinates. Will not capture any object ID’s. If the URL already has zoom/pan parameters defined, they will be updated with the current values. Done
* **Tooltip Position Errors** - when quickly moving between objects, eg between the legend objects, the tooltip will sometimes appear on top of, or above, the trigger-object. When this happens, the info pane for that object fails to load. And I’ve occasionally gotten a stuck tooltip. DONE - fixed by removing a tooltip animation - and another bug.
* **Location hash in URL**. The location value after the hash tag will scroll the navigation or info-pane to the first element that contains an id equal to the location value. For example, given this URL: <http://www.example.com/tigermc/#part2/?on=basemap&on=FinancialSupplyChain&on=Corporations&panx=0.00&pany=39.97&zoom=1.00&wpane=open&epane=open>  
  and this element’s matching id within the text of the info-pane: <span id="part2">Lorem Ipsum</span>   
  The browser will scroll down to that location in the info-pane.  
    
  **Commanding objects wherever they are**

- the tiger code currently does not allow nested objects that are off to be turned on, eg: Done  
<http://45.55.23.31:3000/tigermc/vvvoooMap.html#/?on=basemap&on=thing1>  
does not show thing1 as it should. Open, doesn't work either, eg:  
<http://45.55.23.31:3000/tigermc/vvvoooMap.html#/?on=basemap&open=thing1>  
The off command does work:  
<http://45.55.23.31:3000/tigermc/vvvoooMap.html#/?on=basemap&open=thingA&open=thingA&off=thing1&off=thing3>  
(notice in the URL above (created by first clicking on thingA, the manually adding the off commands) there are 2 copies of “open=thingA” - the hover bug below generates this repeated URL info.)

### Pane Open/Close

– a pane toggle open/close (but not a slide open/close) resets the zoom to show the full map. This should instead be reset to the zoom setting in the URL. Done.

– **Children not getting hidden properly** if they’re open, and they have children, and then their parent is hidden. The behaviors is a erratic - works, then not after a hover on/off. Hiding things down the generations not fully working. **Done**

### Individually command’able UNOs while maintaining stacking order Done

- From our google doc chat 12/14 - Larry to modify code to allow on/off open/close commands to control UNOs regardless where they are or the state of their parents – while also maintaining the relative stacking order of the shown objects - as specified in the original SVG file.

– This will fix the **”Add Thing 1, 2, 3”** link in the story pane that no longer working as it used to.

### Hovering over an object then off sometimes leaves the tooltip on, and hangs everything.

### Extra “&=” Fixed - getting left in the URL after interactions.

### rrr & css stuff - Done

–  **Parse <text> tags** as well as <g> tags for rrr and css commands

–  **Add code to handle rrrS / rrrH** – Marshall added spec for rrr and css Show and Hide command - see **URL, CSS, and JSON Commands from the SVG file** below. Larry to review to see if it’s viable.

– **Trigger rrr’s and css’s on hover as well as click**. The hover on/off can work differently than the click on/off as it can just go back to the previous state when hovered-off (which won’t work reliably for click on/off).

### Info-pane - not getting filled when hovering on pane text. Fixed

### Pan bug

It’s quite useful to be able to set the startup URL in the index.html file, and the designer should be responsible for setting the initial pan / zoom / pane values. In my experimenting with putting a full URL in vvvoooMap.html, that URL seems to override the one created by the code. So should we leave things as it, or comment out the code that automatically adds the startup pan/zoom values ?

In vvv ooo test, there seems to be a conflict between the user-assigned start-up URL and the default/code-assigned URL (?) - that causes the pan to shift. Reload the site, then hover on the story-pane title. **Fixed**

**Flashing** **Fixed**- when hovering across the pane links you can see the artwork flashing on/off in the wrong position.

### - Simplify URL coordinates Fixed

### Mouse Jitter

On both test sites (vvv and rrr) moving the mouse across a ooo or ccc object causes jitter between open/close states. This is worse in Safari, less prominent in Chrome and Firefox. **Fixed**.

### Pane-text hover bug

Click on “Open ThingA” in the story pane

Then close ThingA by clicking on it in the map pane

Now a hover on “Open ThingA” becomes persistent i.e. acts like a click instead of hover - *for just the first hover on/off* - if you hover on/off the text a second time, it works properly (A opens/closes on a hover/unhover). **Fixed**

### Restore shorter delay

I don’t think we need the longer delay on hovering to prevent jitter - restore to 200ms. **Fixed**

### Design bug in vvvoooMap?

I noticed that thingB in the vvvoooMap is not always getting closed when hovering/clicking over the ccc. I thought it was a bug but the rrr class map version seems to be working fine. See: <http://45.55.23.31:3000/tigerlm/vvvoooMap.html#/?on=basemap&panx=0.698&pany=0.532&zoom=1.000&wpane=open&epane=open>

ThingB’s ccc group contained artworked that was grouped together in illustrator - taking this artwork out of the group, and putting it directly under ccc, fixed the close issue AND the residual ‘Mouse Jitter’ I was seeing (but only seeing it on B’s ccc).

You can see the difference between “ooo vvv test 0.4 copy.svg” and “ooo vvv test 0.3 copy.svg” - ungrouping ThingB’s ccc fixed issue. ***However***, it should be perfectly fine to have grouped artwork within a ccc or ooo, so there is probably something in the code that needs to be fixed. **Fixed**

### Master + Override Databases

On tigermc we’re testing both SEMap and the Thing’y test files. I’ve loaded the SEMap info as the master DB, and the Thing info into the override DB. But the Thing test files are non-interactive indicating the they’re not getting connected to the override DB. Fixed.

### Ignore UNO’s in sss’s

Objects with named IDs within sss’s should be treated like plain artwork - and ignored re interactivity. **Fixed**

### Hide UNOs in closed vvv’s

In SEMap.html, UNOs in a vvv are currently being shown when the parent UNO closed. Oddly, this is not a problem in vvvoooMap.html. **Fixed**

### Mouse events for hidden children

Now that this code has changed, the hidden children under vvv’s are getting mouse events, and showing off their tooltips, which needs to be fixed. I think I fixed this.

### InstitutionalInvestors UNO - not getting click events

This one UNO is not picking up mouse-click events. Nothing weird in the SVG that I can see. Fixed

### InstitutionalInvestors UNO - not getting click events

This one UNO is not picking up mouse-click events. Nothing weird in the SVG that I can see. Fixed

### ondoubleclick filed not getting executed properly Fixed

See:

<http://45.55.23.31:3000/SEmap2/SEMap.html#/?on=basemap&on=NGOLayers&zoom=1.000&panx=0.500&pany=0.721&wpane=open&epane=open&open=FriendsProvidentMeeting>

double-clicking on any of the organization logos take you to Esmee Fairbain - instead of what in the ondoubleclickfield.

### css stuff - Working

Add code to handle cssS / cssH. Here is a sample snippet of css code: .topRow{opacity:0.0 !important;}

This needed the “:” and the !important and the “;” . Wrong way: .bottomRow{opacity=0.0}

### Child State

Children are now getting closed when their parent is closed - instead of remembering their state as they used to. See [spec here.](#_pg6d4wdcz9lh)

When a parent is closed, run the rrrH and cssH logic for the children. When parent is opened, run the children’s rrS and cssS logic. **rrr working, css working. Fixed?**

### ooo’s not being hidden on hover

ooo’s not getting hidden when hovering - like they used to. ccc’s get hidden on hovering, and clicking works as expected.

### Intercepting mouse events - or not

Objects that are non-interactive should be transparent to mouse interactions (UNO's under them should be accessible).

Because ooo’s and ccc’s are the only interactive objects in the map-pane, should we **use pointer-events to limit mouse-listening to ooo's / ccc's ?**

Non-interactive objects interfere with mouse events: eg “Friends Provident …” is on top, and can be opened, but the others “Leverage Points” etc. can’t be moused-on as they’re inaccessible beneath the big “Friends Provident …” dimming rectangle. And actually, the rest of the map is hidden from mouse events as well - when NGOLayers is on.

### FIx handling of Opacity Groups & Stacking Order

**Use the CSS ‘visibility’ property to show/hide things - thus avoiding conflict with opacities set in the SVG ?**

CSS is used to set all UNO's and vvv's opacity to 1.0. Then to hide things, the opacity for artwork *within* the UNO / vvv is set to 0.0 (when off/closed) or 1.0 (when on/open).

UNOs, ooo’s, vvv’s, ccc’s should not include an opacity value as part of their <g> tag. In illustrator, those objects should be created as Illustrator Layers, not as groups: this prevents the setting of opacity across that group (in Illustrator an opacity can be set of a group, but not for a layer.

### ?+++ Command: use the current pan, zoom, pane-states - Fixed

Currently, executing a +++ command does not use the current pan, zoom, and pane-states when updating the URL.

If there are no pan/zoom/pane values in the +++ command, then the new URL should use the *current* pan/zoom/pane values. Note that we don’t want to use the values from the current URL, but the values from the actual current pan/zoom/pane state. Those values will need to be grabbed - as we do on a shift-click - and then used to form the new URL.

### ‘Direct’ pane-links not getting added to URL/history - Fixed

+++ commands from pane-links work as expected and update the URL (except when it breaks - see the next item below). But ‘direct’ URL commands, eg:

<a href="#/?on=basemap&open=Finance" class="tagged"> <span style="color:black">Basic Structure plus Finance</span> </a>

Are not updating (overwriting) the URL.

### Set URL/history on zoom controls: ‘+’, ‘-’, or ‘Reset’ fixed

The URL & browser history should be updated after hitting the zoom in/out/reset controls, i.e. do an automatic shift-click operation after hitting the zoom controls.

### +++&closed Bug - User (MC) Error

The +++ command has different syntax for open and closed:

?+++&open= & required

?+++closed= & prohibited

Also, a command can be entered in the browser address bar works. e.g.:

?+++closed=Finance (without the &)

but the exact same command from a pane-link does not work, eg in SEmap:

#####<a href="#/?+++closed=Finance" class="tagged"> <span style="color:black">Close Finance</span> </a>

because it creates this URL:

[http://45.55.23.31:3000/SEmap2/SEMap.html#/+++closed=Finance?panx=0.500&pany=0.550&zoom=1.000&wpane=open&epane=open](http://45.55.23.31:3000/SEmap2/SEMap-flat.html#/+++closed=Finance?panx=0.500&pany=0.550&zoom=1.000&wpane=open&epane=open)

which shows nothing. Adding the ‘&’ doesn’t help (then it does nothing).

### A hover on a pane-link can set the URL FIxed

A hover should never update the URL, but sometimes this happens, eg in

<http://45.55.23.31:3000/SEmap2/SEMap.html>

for this link:

####<a href="#/?on=basemap" class="tagged"> <span style="color:black">Basic Structure</span> </a>

NOTE: if the previous command was a +++ command eg the **All Rows & Columns** link, then hovering on **Basic** **Structure** works ok (does not set the URL). But if the previous command was a not a +++, eg **Basic Structure plus Finance** then a hover on **Basic Structure** does mistakenly set the URL.

### Fast-clicking pane-links garbles the URL test it now. FIxed

Occasionally - seems to be caused by clicking quickly on different pane-links - the URL will get garbled - eg:

#/?basemapNGOLayers=

… &Government=& ...

#/?basemap=&

#/+++=+++?panx=0.500 Got this one out

Example URLs from when this happens:

<http://45.55.23.31:3000/SEmap2/SEMap-flat.html#/?basemapNGOLayers=&zoom=1.086&panx=0.532&pany=0.589&wpane=open&epane=closed&Government=&open=Research&open=StateFinance&open=Information&open=Pressure&open=Banking>

<http://45.55.23.31:3000/SEmap2/SEMap-flat.html#/?basemap=&zoom=1.000&panx=0.500&pany=0.665&wpane=open&epane=open&open=Banking>

### open no longer overrides a parental close test it now - Fixed

An open command for a UNO no longer works (shows the opened UNO) if the parent is closed. This used to work.

### manually adding a URL in address bar containing a +++ may be problematic Fixed

...because if children who’s open commands are purged from url may not have them purged because the +++ command with only the change is still in address bar when a hidden command attemps to purge the open command.

### Bug in clicking on pane-links? fixed

In playing with SEMap2, the links from the story-pane work great - until they don’t. After loading the page, clicking things on (from the story-pane) duly add the new items to the URL. But after some number of clicks, it stops updating the URL, losing the proper browser history.

### After closing a pane then hovering on link, pane re-opens… Can’t duplicate

...and the pane open commands shows up in url.

### Pane state/width in the URL Done

The pane state/width can be set via the URL commands:

* The ‘pane=closed’ and ‘pane=0’ commands are equivalent and will close the pane.
* A ‘pane=open’ command, will open the pane to the last opened width.
* A specific pane width, e.g. ‘wpane=300’, will open or re-size the pane to that width.

### URL update on on pane-resizing

The URL should update when a pane is just resized - without opening or closing the pane **- Fixed**.

The url is not getting updated when a pane is toggled open **- Fixed.**

### Pan/zoom-centering for different window sizes - FIXED !!!

The pan x/y values currently captured in the URL don’t work correctly if the size of the browser window changes. The zoom value does work correctly as it is relative the window size. What we need to capture in the pan x/y coordinates is the absolute coordinates of the ***center*** of the view. This absolute center position (i.e. a point on the map centered in the browser window) combined with the zoom value, should give a more consistent view of the map despite changes to the browser window size.

How this should work:

Pan/zoom values (in the URL) should always be relative to the current window size, e.g. the URL:

**zoom=1.00&panx=0&pany=0**

***should produce the same results as hitting the ‘Reset’ zoom button:*** displaying the full width/height of the map centered in the map pane - *for any window size.* Similarly, if you pan/zoom in so that some UNO is centered and fills the map pane, and you save that URL, that URL should show the UNO centered/filling the map pane at any window size.

Currently, when the browser window is resized, the pan/zoom values are calculated relative to the original window size. (Which brought up the issue below ‘Zoom Reset’.)

### Pane slide vs. toggle fix. – Done

This turned out to be a simple fix for the making sure the map-pane size is correct:

In idiagram-ui.js change the toggler lengths to -1:

togglerLength\_closed : -1, // "100%" OR -1 = full height

togglerLength\_open : -1, // WIDTH of toggler on north/south edges - HEIGHT on east/west edges

This will prevent sliding, and only allow toggling

Get rid of the slider button image by commenting it out in mapWindow.css

/\* background-image: url("Toggle-Button.jpg"); \*/

### Set up node.js hosting / development site

To get ready for the launch is move the platform to a scalable node.js host. These pages review some of the options:

<https://seroter.wordpress.com/2013/07/29/where-the-heck-do-i-host-my-node-js-app/>

<https://github.com/nodejs/node-v0.x-archive/wiki/Node-Hosting>

OpenShift looks like it might be promising:

<https://www.openshift.com/web-hosting/node-js.html>

And it looks like you can align the hosted application with domain name of your choice:

<https://blog.openshift.com/domain-names-and-ssl-in-the-openshift-web-console/>

Here is a page on server options:

http://lab-01.com/wiki/doku.php?id=admin-install-ubuntu

### Add password for test site access -

Works - need to implement on the actual site.

We need a pswd protected space for the beta test site.

### Pane-text and file commands

Change the command names to **info=, map=, story=**  Add the story file command **– Working.**

Set the story= in the startup URL, and maintain it in the URL as we go - **Done**

Slide links are not getting reset/updated when the story= command reloads the story-pane. **Done**

Add a CSS style for text loaded by the infotext= command. - **Done**

### Finishing Presentation Code

Add Shift-comma/period key-command for first/last link. **– Done**

Clicking on a class:slide link should set SlideSelected to that link - and highlight that link **– Done**

The links should not wrap-around top-bottom, bottom top, but stop at the end/beginning. **– Done**

URL Updating – as you move through the links the URL does not update, and even a shift-click doesn’t capture the current view.**– working**

### Default/Initial URL – Fixed

The default URL from the startup.html page is not loading on startup. This leaves the map blank on startup - or on loading a new story-pane with story=.

### Per-UNO all commands: openall= and closeall=

With the new super torture test file:

http://45.55.23.31:3000/demo/things.html

Try this test - go to the pane links:

**Per UNO All Commands:**

Open all A

Close all A

* Click on Open all A - A opens all correctly.
* Click on Close all A - oops - the connections between objects do not get hidden. - those connection lines are UNOs in the base of the B and C objects - they are not in a vvv-group.

Somehow the open/closeall functions are missing things in the base-UNO - the openall/closeall isn’t properly showing/hiding the base-contents of the UNOs.

Add the option-click key-command - to trigger an openall/closeall.

**Hovering-off**: From the pane links:  **– Works**

### Copy URL to clipboard on shift-click

Because of security reasons, this only works if user clicks on something like a form that has the text in it. I could copy it to a textbox.

This is a nice-to-have, if it’s quick to implement.

In addition to setting the current URL, it would be nice to set that URL in the clipboard. Should work in most browsers - but not Safari.

FYI - there are 2 shift-click handlers in the current code.

See: <https://developer.mozilla.org/en-US/docs/Web/API/Document/execCommand>

And: <http://codepen.io/netsi1964/full/QbLLGW/>

I tried this line of code in the shift-click handler - didn’t work:

document.execCommand("copy", 0, "I copied this text!");

### Per-UNO all commands: openall= and closeall=

Fix closeall functionality.

The closeall=unoA command shouldn’t turn unoA off.

### SVG processing should ignore <symbol> tags – Fixed

I’ve just run into a bug with symbols in Illustrator not displaying properly. Illustrator saves symbols within a <symbol> tag at the beginning of the SVG, and the problem is that some of those symbols also get a <g> tag such as: <g id="TellusMaterLogo\_3\_">. The SVG processor then thinks this a UNO, doesn’t find it in the database, doesn’t display it.

So, we should be fine if the processor just ignores the contents of <symbol> tags when looking for UNOs etc.

### Move rrrS/rrrH to Database – Works

Rather than specifying the rrr commands in the illustrator file, it will be easier to design/maintain them if they are defined in the database.

Four new fields:  **onURL offURL openURL closeURL**

Here is the new schema that will be ready soon:

*id label synonyms kind level value shortDescription longDescription webLinks references connections notes onhover onclick ondoubleclick onDimming tooltip infopane css target ttPosition ttTopOffset ttLeftOffset onURL offURL openURL closeURL*

contain the URL commands that will be executed when that UNO’s is turned on/off or opened/closed.

Comment-out the code that parses the rrr commands from the SVG.

### Issues with css classes – Testing - working

Two issues with adding css classes.

1. We should allow classes to be added to ooo, vvv, ccc, sss groups. So that we could have things like:

<g id="sss\_class:background">

Would allow CSS control of things in the sss group. Currently that doesn’t work.

We should also allow assigning CSS classes to any <g> object - even if they aren’t UNOs o special groups, i.e. we should correctly parse things like:

<g id="\_class:foreground\_2\_">

<g id="\_class:background\_class:connection\_3\_">

2. There’s a bug when trying to add multiple classes to a group - it doesn’t seem to correctly parse the underscore between the two class designations. **- Fixed**

### Designer Bug

See: <http://45.55.23.31:3000/demo/things.html>

I was testing the rrr-in-the-DB, and the Show All Objects UNO/button started throwing this error:

jquery.js:557 Uncaught RangeError: Maximum call stack size exceeded

Open/closing Stop Intro triggers the same error. All other things (UNOs & rrr commands) seem to work OK. Same error in Chrome and Safari.

### Zoom/browse key overlap - Fixed

(on my mac anyhow) The browse back/forward keys cmd+[ and cmd+] also trigger a zoom out/in action when the keys are released.

### e / w pane Toggle not Slide fixed

Rather than sliding: eg myLayout.slideToggle("east");

The key-command code should *toggle* the panes open/closed.

### Move rrrS/rrrH and cssS/cssH to Database

closeURL seems to have just stopped working… **- Working Again - yeah**

onURL, offURL, and openURL working ok.

See: <http://45.55.23.31:3000/demo/things.html>

In the URL Controls: **Show All Object** and **Open All O, A, X** - both use the rrr commands - they show but don’t hide.

See: <http://45.55.23.31:3000/demo/things.html>

In the map - **CSS Controls** all use the css commands. Click **Open All O, A, X** in **URL Controls**, then try the CSS Controls; nothing happening. **This now works again like it was before.**

### Goto Command – Working - Testing/ed - Done!

**Current Syntax**

There are two goto commands:

**goto=** pans to center on the specified UNO

**gotoz=** pans to center on the specified UNO, ***and*** zooms to encompass the objects + 20% extra space.

The goto commands will take as arguments a UNO id with an optional prefix that specifies which part of the UNO to pan/zoom to: e.g. for the UNO ThingA, the possible goto (or gotoz) commands are:

**goto=ThingA** - pans/zooms to center-on/encompass the full contents of the UNO

**goto=oooThingA** - pans/zooms to center-on/encompass the ooo-group of the UNO

**goto=cccThingA** - pans/zooms to center-on/encompass the ccc-group of the UNO

**goto=vvvThingA** - pans/zooms to center-on/encompass the vvv-group of the UNO

Goto commands can be sent via a URL command, and the *result* will be reflected in the new URL, but the goto= command will never appear in URL in the address bar.

**Potential Alternate Syntax**

Rather than the goto=xxxThing, I’m thining that something like this makes more sense (unless I’m missing something):

**gotoCz=thingA** - will pan/zoom to center on the full extent of the closed UNO (eg. encompassing and centering on everything in the base-uno) + X%.

**gotoOz=thingA** will pan/zoom to center on the full extent of the open vvv group + X%.

**gotoC=thingA** - will pan to center on the full extent of the closed UNO (eg. encompassing and centering on everything in the base-uno).

**gotoO=thingA** will pan to center on the full extent of the open vvv group.

### Opening child uno’s need +++’s to retain state

After making changes to the code to fix a bug that caused the url to lose the open command when certain links were hovered over, this started happening: When clicking on thingA and thing1, then clicking on the pan link, the ***state is lost***:

### Startup URL mismatch

**Working** - much better now with the new code that uses callbacks.

On start-up, the URL in the address bar matches the **var defaultURL**, but the map is not really positioned at that URL - if you hit shift click (immediately after opening), the URL changes to something slightly different.

The problem is caused by the startup URL changing the e/wpanes from their default values (currently set to 250). In that case, the start-up pan/zoom is set ***then*** the panes are adjusted, throwing off the xpan value by a bit. It’s a minor issue, but we should re-order the timing so that we adjust the panes first, and then do the panning/zooming.

On initial map loading it’s also not doing a proper normalizeMapPane() before executing the startup URL. e.g. the startup URL for pztest.html should center the map and zoom to fit the window (i.e. it’s the same URL you should get when doing a reset) - but that doesn’t happen. ON start I believe the order of things should be:

* normalizeMapPane
* Get the startup URL
* Adjust the panes (if they’re changed by the URL)
* Call normalizeMapPane because you just adjusted the panes.
* Execute the rest of the startup URL

**New pan/zoom with callback code:**

Unfortunately has made the problem worse. I created these test files for *testing-* and have been assuming you run all new code against demo/things.html and pztest.html. This testing was done with <http://45.55.23.31:3000/demo/pztest.html>  
That map is 1400 x 1000 points, so   
 panx=700.0&pany=500.0&zoom=1.0  
should center/fit the map - this is what a reset produces, and what’s in the startup URL.

The original problem persists, only worse:

• on startup the map now goes through number of gyrations - and still doesn’t set the startup URL correctly.

In addition, there are a number of new problems/bugs:

• Changing the window size can cause a zoom - that shouldn’t happen

• Changing the pane size can cause a zoom

• Gyrations - where it zooms in/out - often on window resizing • There is now a quite noticeable delay on any pan/zoom operation e.g. when hitting the pane-links to trigger a goto=

• It’s possible to confuse the zoom setup so that zoom no longer works correctly

### Multiple Pane-Link Classification – Fixed

I’ve just discovered that you can’t apply more that 1 class to a pane link. Only the first class applied will work. See: <http://45.55.23.31:3000/SAMap/SAMap.html>

The slide links work, but those links also have class=”hover” - but hovering doesn’t work (because “hover” is a second-class class).

Correct syntax: Instead of class="slide" class="hover" do this:

class="slide hover"

### Disable the Multi-function Space Bar – Fixed

The space bar both stops/resumes paying and - of course - pages-down the story pane. Would be nice to disable the page-down, see:

<http://stackoverflow.com/questions/18522864/disable-scroll-down-when-spacebar-is-pressed-on-firefox>

### Handle local links from ondoubleclick – Works

Just set the target to ‘\_self’ !

It would be nice to use the double-click field of the DB to store local URL commands, such as goto links, e.g.:

#/?+++&gotoz=ListedCompanies

This link does work on this page (double-click on Listed Companies):

<http://45.55.23.31:3000/FPMap/SDGMap2.html>

But only after reloading the whole page. Can we intercept the ondoubleclick links and execute local map links as URL commands?

### Collapsable Text – works

It’s working using the jquery accordion funciton.

Add the ability to expand/collapse headers in the panes. See: <http://www.w3schools.com/bootstrap/bootstrap_ref_js_collapse.asp>

### Import HTML files to the panes - fix linking – Working

**To do:**  While the code works, we just need to accept .html file extensions: eg in the story= and info= commands. And maybe bypass the markdown converter? Although it might not be necessary as full HTML is permitted in markdown.

In /SAMap folder, SAMapStory.html & SAMapStory.txt are the same thing. We just want the .html one to load.

-------

Importing HTML works now - so long as you fool the code by giving the .html a .txt extension. See:

[http://45.55.23.31:3000/SAMap/SAMap.html#/?on=Basics&on=SIPlatform&on=SheetInfo&panx=1224.0&pany=792.0&zoom=1.000&wpane=280&epane=0&story=accord2.txt](http://45.55.23.31:3000/SAMap/SAMap.html#/?on=Basics&on=SIPlatform&on=SheetInfo&panx=1224.0&pany=792.0&zoom=1.000&wpane=280&epane=0&story=accord.txt)

That uses the jquery accordion text: http://jqueryui.com/accordion/

The slide links work fine. However the other #/ Links don’t work - as I imagine things are confused by the links being in a different HTML page/div ?

So there’s the easy bit - having the story= and info= commands recognize .html extensions (and probably bypass the markdown converter).

### Display:none

Due to the way you have open/close and temporary open/close, this is an easy fix. See /demo/mc-mapWindow.css - where we split the vvv close into 2 parts:

.vvv.close > \*:not([uno-id]) {

display: none;

}

.vvv.temporary-close > \*:not([uno-id]) {

opacity: 0.0;

}

And then always turn things off with display: none:

g[id].off > \*:not([uno-id]), g[id].open.off > \*:not([uno-id]), g[id].close.off > \*:not([uno-id]), g[id].temporary-open.off > \*:not([uno-id]), g[id].temporary-close.off > \*:not([uno-id]), g[id].parentHide > \*:not([uno-id]) {

/\* Use display: none instead of opacity: 0.0 to make pan/zoom faster \*/

display: none;

pointer-events: none;

}

-------

I did more testing - this time the right way - and found that indeed display:none is the better/faster way to hide things because it pans/zoom as if the hidden things are not there vs. the way we now hide things where pan/zoom is always pan/zooming the hidden things - with the accompanying performance hit.

This is a really good thing, as it make the difference between complicated maps always pan/zooming slowing vs. only being slow when lots of things are on/open.

So, we should change the hiding method(s) to use **display: none** instead of **opacity: 0.**

The ooo and ccc groups will still need to be hidden using opacity: 0 - as we still need to catch mouse events for them while they’re hidden. But all other groups – especially for turning off UNOs – could use display:none

### Finishing the Animation Code

**RR/FF Delay: Logical Groupings Work**

A different approach, on ‘r’: - use logical groupings.

For the documentation:

Animations consist of a list of media ‘elements’ - currently the elements can be URLs or audio files. This list of elements can be grouped/nested to any number of levels. The first-level grouping of elements is called a ‘segment’. The second-level grouping of elements (the first level within a segment) are the logical-groups that will be skipped over by the forward (f-key) and reverse (r-key) commands (these 2nd level groups will typically contain a URL and an audio file). Audio/url subgroups should have a synchronous type, i.e.,

{

"type": "synchronous",

"postdelay": 1,

"elements": [{

"type": "url",

"postdelay": 1,

"content": "?on=thing1"

}, {

"type": "audio",

"content": "http://45.55.23.31:3000/present/tts2.mp3"

} ]

}

**Looping Segment**s - **Working** add a way to do it - extend ‘loop:true’ to segments. Need a method of telling a *segment* to go back the beginning when it reaches the end.

Looping *audio* will stop when the segment it’s in finishes. Looping *segments* will keep looping until another segment is played, or a stop command is issued.

Tried this in SAMap.html (Play Looping Steps) - both the segment constructs below play, BUT they both only loop the last element:

"segment\_7": {

"id": "loopingSteps",

"elements": [

{

"type": "segment",

"loop":true,

"content": "step1to3"

} ]

}

OR -----------------------------------------------------

"segment\_7": {

"id": "loopingSteps",

"loop": true,

"elements": [

{

"type" : "url",

"content" : "?+++&open=SIPT&open=Synthesis&panx=1080&pany=520&zoom=2.4"

},

{

"type" : "audio",

"content" : "audio/Marker 04.mp3"

},

{

"type" : "url",

"content" : "?+++&open=SIPT&open=Mapping&panx=1224&pany=520&zoom=2.4"

},

{

"type" : "audio",

"content" : "audio/Marker 05.mp3"

}

]

}

**Pause/Resume Delay-only Segments** - **Working** the for animations with just delays (see Play Delayed Steps in SAMap), the spacebar no work.

**RR/FF for Delayed Segments** – make it work for audio-less elements. Done(?) and Done.

**Nesting Segments** - do I have the syntax wrong or? **– Fixed**

• **Looping – Works!**

Add the ability to play looping sounds (and looping animations; done by nesting segment within itself).

I just tested looping audio with the r keyboard shortcut. I found again that I needed to insert a postdelay in the url type to get this to work correctly:

"type": "synchronous",

"elements": [{

"type": "url",

"postdelay": 1, // Usually don’t want this value to be longer than the audio

"content": "?on=basemap&wpane=250&epane=250&panx=0.727&pany=0.317&zoom=2.119"}, {

"type": "audio",

"loop": false,

"content": "http://45.55.23.31:3000/present/tts2.mp3"}]

• **Skipping – Works - with the right JSON grouping**

Add forward and backward key-commands.

• **Change how delays work – Works!**

In the current code: “delay value is how long to wait ***before*** playing stuff.”

What it should be is: “delay value is how long to wait ***after*** playing stuff; before going on the the next element.”

The delay function will usually be used to provide a delay when you don’t have an audio file to time things. So you want an element to first execute the ‘content :’ (e.g. go to a URL), then wait the delay, then go to the next element.

• **Nesting** **– Works!**

Add nesting segments. Nesting a segment by inserting a URL into another segment doesn’t produce the right behaviour: the URL is executed and the then the player moves on to the next segment without waiting for that new segment to finish. You would have to put in an appropriate delay, and it’s not easy to know what that delay should be. I think we need a new element type:

"type" : "segment"

“content” : “thisSegmentName”

The code would essentially insert the elements from “thisSegmentName” into the parent segment - and they would play as per usual.

Looping segments would be created by embedding a segment in itself.

**• Command Syntax Change – Works!**

Rather than the commands:

animate=thisSegmentName, play

animate=thisSegmentName, stop

Change to this syntax:

play=thisSegmentName - plays the segment from the start

stop - stops the playing segment

For now, I don’t see the need for a pause or resume URL command - that is handled by the user hitting the spacebar - and otherwise when/how would you ever know to send a pause/resume URL command?

**• Fail gracefully for missing audio files – Works!**

If an audio file is missing, the player should post a console log warning, but otherwise continue on playing without interruption.

This will be useful from a design standpoint when you want to block-out and test the animation/presentation before the audio files are complete.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Animate commands in the URL? – Currently, animate commands get left in the URL indefinitely. We could remove them from the URL when paused/finished, but maybe it’s easier to never add them / remove them immediately , so they never appear in the address bar. **– Working**

### Gotoz a bit broken - add fix to idiagram-svg.js

In ~line 2605 of idiagram-svg.js the variable sv is incorrect:

var sv = $(".svg-pan-zoom\_viewport")[0].getBBox();

The function returns the BBox of only the current visible SVG artwork. What we really want is for sv to always equal the SVG viewbox specified in the .svg file.

We don’t need the variable sv, and instead use s.viewBox.width and s.viewBox.width. See the corrected code in /js/mc-idiagram-svg.js

### UNO Naming Harshness - Fixed

Currently, if a designer creates a layer in Illustrator (thinking that she’s going to create an UNO - or just by mistake), but then fails to enter that name in the DB (or has typo in the DB name), the map fails to load - and gives no indication of what failed, other than the error that shows up in the debugger.

It would be less harsh - and much easier to fix - if the code could throw an alert for any named-group that doesn’t have DB entry (showing the missing name in the alert). As a bonus, it would be nice if we could then just ignore those groups (treat them like sss groups) and continue loading the map - rather than crashing.

### Finishing the Animation Code - Fixed

The id can still be used - but it not needed.lay command

one more thing...:)

**Segment name and id** - why do segments have both names and id’s? - seems redundant/confusing… Wouldn’t it be simpler to get rid of the id? Or does each segment have to start with “segment”?

Why both?

"segment\_2": {

"id": "aloneloop",

### Pane-Scroll fix – Fixed

Problem: for example, if a long info pane is scrolled to the bottom, and a new long content is loaded to the info-pane, the new content will also be scrolled to the bottom - you won’t see the new title etc. See:

<http://45.55.23.31:3000/SDGMap/SDGMap2.html#/?on=Map&panx=2265.0&pany=1417.0&zoom=1.000&wpane=270&epane=220&open=Finance&open=SDGs&story=SDGMapStory.txt>

then hover between SDG 13, 14, 15 (at the bottom of map) - the info pane is scrolled down at all, it will stays scrolled down as you switch SDGs - rather than scrolling to the top when the info pan data changes.

Whenever the info or story panes are reloaded with new content, we should reset the pane <div> scroll value to the top, e.g.: infopane.scrollTop=0

See: <http://www.w3schools.com/jsref/prop_element_scrolltop.asp>

### g-clicks – Working

Distinguishing map keyboard commands from input to forms in the pane. See:

<http://45.55.23.31:3000/demo/Radial.html#/?on=Map&on=IssueX&epane=300&wpane=260&panx=1071.2&pany=1000.0&zoom=1.440&story=RadialStory.md&info=issueXinput.html&open=A>

If the user is focused on a text box input form, then any text keyboard shortcuts that are entered into this form will not be used as a shortcut.

For an example of this problem.

It's triggering a gotoz=oooUNOid or gotoz=cccUNOid, which is not really what we want.

What it should do - so that it works consistently - is always trigger these two commands(in this order):

open=UNOid&gotoz=UNOid

### On/Open URL command bug – Fixed

The database URL commands: onURL offURL openURL closeURL

Work well when a UNO is *individually* triggered. However, when an UNO is turned on or opened by their *parent* UNO, the DB openURL command does not get executed.

See this test: <http://45.55.23.31:3000/demo/URLtest.html>

Click on URLtest - which open to show its one child: Open123.

Click on Open123 - which just executes a openURL command to open B1, B2, B3

Click on URLtest to close it - the DB closeURL command is getting executed because B1, B2, B3 close.

Click on URLtest to open it. Open123 is still open, but B1, B2, B3 do not get opened as they should.

### Long zoom value – Fixed

When g-clicking on an UNO the zoom value in the URL is not getting truncated, e.g the URL get things like: zoom=4.761914508437452

### Option/Alt-Click should show up in the URL - Fixed

When option clicking on an UNO, an openall=thisUNO or closeall=thisUNO need to be added to the URL - which they are now - but now a regular-click should remove them from the URL.

### Save as PDF – Done

THE scenario:

• hit the ‘p’ key

• the svg/css file of the current map state is opened in a new window/tab

• the user can print this svg/css window to a PDF

-------------------------

**Testing 2nd Version**

* Works on Mac Chrome and Opera, but not in Firefox (rasterized the image) or Safari (blank print output)
* Runs into pop-up blocking - but that’s easily gone around
* Can we remove the pan/zoom controls: - + - Reset
* The ‘printed-to-PDF’ files open perfectly in Acrobat and Illustrator
* It saves the current view *including the pan/zoom settings* - so that the resulting PDF reflects the pan/zoomed view (within the limits of the aspect ratios (of the browser-window vs. the paper-size) and paper size (chosen in the print dialog)) . This is a good thing - what we originally envisioned as ‘saving the current view to PDF’.

**Testing the 1st version:**

* After hitting ‘p’, we should use a the OS **standard save-file dialog** (rather than a special CSS formated window) for the download. Currently, the file of unknown name is downloaded to an unknown location.
* The hybrid svg/css file does display correctly in browsers, but nothing else can open it e.g. it crashes Illustrator. I checked to see if there were simple errors in the file with <https://validator.w3.org> - but it seems to be a file-wide issue with the CSS and class assignments in the SVG groups.
* It is possible to create a PDF by opening the SVG in Chrome and printing to a PDF file.

Because the SVG is not directly useable (eg it can’t be opened in Illustrator), and the opening/saving in Chrome is cumbersome, we’re not quite there yet.

My guess is that browsers are the only thing that is going to open the file and apply the CSS correctly. If that is the case, the solution may be to ‘manually’ apply the CSS to SVG as it’s saved i.e. save only the visible elements to the SVG file, and don’t save any CSS definitions, or extra info in the <g> tags - e.g. none of this stuff:

class="ooo close parentHide off" mytype="ooo" uno-id="thing4" data-original-title="" title=""

-------------------------

DISCUSSION

**Trying out these possible solutions:**

Trying out <http://wkhtmltopdf.org/downloads.html>

<http://www.cloudformatter.com/css2pdf#>

<https://www.onlineconverter.com/svg-to-pdf>

Tried <http://cairosvg.org> for the PDF conversion Didn't work.

Batik: <http://xmlgraphics.apache.org/batik/> - java, won’t work.

Rsvg - <http://superuser.com/questions/381125/how-do-i-convert-an-svg-to-a-pdf-on-linux>

This doesn’t work: <http://www.hiqpdf.com/demo/ConvertHtmlToPdf.aspx>

Nor does this: <http://www.evopdf.com/demo/HTML_to_PDF/HTML5_Features/SVG_to_PDF.aspx>

There’s this: <https://github.com/CBiX/svgToPdf.js/>

I tried a couple of other things, and they only save the full/raw svg file - not what’s in the map-pane. This may be much trickier than anticipated.

JSFiddle for cloudformatter: <http://jsfiddle.net/zvx6eb7e/>

Here is a discussion of saving svg: <http://stackoverflow.com/questions/23582101/generating-viewing-and-saving-svg-client-side-in-browser> (LM)

### Database ID Format

I keep running into the need to create numbered UNOs, eg

1. First Thing
   1. First Sub-Thing
   2. Second Sub-Thing
2. Second Thing
   1. First Sub-Thing
   2. Second Sub-Thing

And it would be *really really* nice to just be able make the UNO IDs:

1

1.1

1.2

2

2.1

2.2

I think you ran into problems with both having a leading number and periods in the UNO id, but I’m wondering if there’s any way around this??

It’s easy enough to put some leading alpa-character before the number (if the leading number is a problem), but there’s no good substitute for ‘.’ (well, dashes would be ok - but they’re forbidden too).

I’m currently starting at a spreadsheet I need to render with 220 rows - each with x.x.x.x number + a long text title - so you can see the problem with trying to use only camelCase.

See this test file:

<http://45.55.23.31:3000/Arcadis/numberTest.html>

The UNO ID (in the SVG and in the DB) are = the map lables e.g. 1, 1.1, 1.2 etc. Note the lack of interactivity...

**Work-around:**

From the test map we can see that the work-around is to add a leading alpha character, and use dashes instead of periods. We can live with that. And with luck we will not run into the issue that Larry flagged: “there is some code in the library (jquery or bootstrap's tooltip, I can't recall) that thinks the dashes are minus signs and tries to do the math.”

### User Registration

LM to spend a bit of time investigating how best to implement, and will report back with recommendations re. Which approach/library to use and if LM will implement or if we should hire another programmer to do it.

**Requirements:**

* The registration system should integrate with Mongo DB - user info should be accessible through Mongo.
* Should handle user email lists - and basic community communication stuff.
* Different user classes will have access to certain otherwise restricted page (maps) on the site.
* We’ll also want to add a commenting/survey capability - so ideally we’re looking for a system that does both user registration and some kind of simple commenting system - using Mongo DB.

**A couple Relevant Links:**

<http://jasonwatmore.com/post/2015/12/09/MEAN-Stack-User-Registration-and-Login-Example.aspx>

<http://code.tutsplus.com/tutorials/authenticating-nodejs-applications-with-passport--cms-21619>

<https://hackhands.com/mongodb-crud-mvc-way-with-passport-authentication/>

<http://mherman.org/blog/2013/11/11/user-authentication-with-passport-dot-js/#.VxauqmMeWbA>

Jsonlint.com

my-html-file-name.json

{

"svgFile": "mysvgfile.svg",

"folder": "frmap",

"storyFile": "SDGMapStory.txt",

"infoFile": "nfo.txt",

"masterDb": "mycollection\_master",

"overrideDb": "mycollection\_override",

"zoomSensitivity": 5.5,

"segmentsFile":"presentation.json",

"defaultUrl": "on=basemap&panx=2265&pany=1417&zoom=1.0&wpane=250&epane=220&open=Finance"

}

### Add tween for .classes – Working

LM to add .classes to tween functions

### JSON Preference file name – Working

Should match the startup.html file name

### Compact URL Format - too tricky to implement

The URL tween commands got me thinking about the URL, and how we might make it more compact and readable.

My thought is that we could group similar commands, e.g. this URL:

#/?on=unoA&on=unoB&on=unoC&open=unoD&open=unoE&open=unoF

Could be something like this:#/?on=unoA+unoB+unoC&open=unoD+unoE+unoF

This shortened URL format could be issued as a command (making the URLs easier to write), as well as written to the address bar (making the URLs easier to read).

I would assume that parsing URL commands this way would be relatively easy - just a bit more parsing code. Writing the URLs for the address bar might involve more coding; unless you already maintain an array with all the on/off open/close UNOs in the URL - that would just need a bit of sorting.

Let me know what you think.

### Bugs in Webpack Version

* + **Misalignment** - **Fixed**  
    These misalignment problems seem to be specific to the arcDemo.html (I’m going to try removing the accordion stuff to see if that’s the culprit). This test with the new code seems to be aligned OK (but the panes don’t drag):  
    <http://45.55.23.31:3000/demo/pztestnew.html>
  + It’s not the head and body tags - but something else with the accordion code that’s causing the misalignment - specifically the .css files for accordion.
  + **Pane misalignment** - the info pane content is shifted to the right on loading; closing/opening the panes fixes that shift, but causes the draggable divider to show up in the wrong place - shifted about 30px into the map-pane. OR sometimes the divider in the right place but the story/info pane is 30px too wide.
  + **Map misalignment -** related to the problem above (I’m guessing), the map is also misaligned in the map pane by about 45 px up and left - which is about the same amount as the pane misalignment.
  + Reset/+/– buttons not repositioning on window-sizing or pane open/close - fixed in the latest bundle

### Draggable Panes – Fixed

**Pane-dividers not draggable** -How about now? Oooo - the draggy panes drag

### Custom Function Error - Resolved

For http://45.55.23.31:3000/Arcadis/arcDemo2.html

I tried running this simple custom function in Arcadis/customstuff.js

function rotateStep(time,relative,amount) {

var q = idiagramSvg.getURLParameterList("+++&rotate=Step0,"+time+","+relative+","+amount);

idiagramSvg. processCommandsInURL(q);

}

Via the URL command: ?+++&run=rotateStep,4,f,180

But it throws this error:

Uncaught ReferenceError: getURLParameterList is not defined - customstuff.js:4

**NOTE:** these commands (like processCommandsInURL()) may cause problems under certain circumstances. That is because processCommandsInURL() is already in the middle of processing commands in the URL when it is run again in this custom function, rotateStep().

### Browsefify’ing – implemented - Working

The Browserify bundling spec is in the file:  **/js/mod/index.js**

Maybe switch to **Webpack** in the future - if we need it.

-----------------------

We should probably modularize the .js files a bit - pulling logical chunks out of idiagram.svg and giving them their own files. But let’s not go overboard with too many files.

**File Naming**

We need to remove the name ‘idiagram’ from the file names. The separate JS files are kept the the folder js/mod/ - but then concatenated into modActivator.js as the main JS include file.

* **Visualizer.jsx** - adobe extendScript that creates .ai files from .csv input
* **Devisualizer.jsx** - adobe extendScript that creates .csv files from .ai input
* **modActivator.js** - main concatenated code - replaces idiagram.svg
* **activator -** the bulk of the map loading/interactivity code.
* **presentor.js** - currently present.js - the ‘animation’ code
* **ui.js** - currently idiagram-ui.js - the UI code
* **dynomator.js** - the dynamic SVG functions

Not sure what the new sub-files - or their names - should be…

### Qtip2 - working

LM implementing Qtip2 and updating the spreadsheet/DB schema and docs.

With updated DB schema to include ‘ttStyle’ field

* + Remember to optimize the qtip library so you don’t haul in the whole kitchen sink. See <http://qtip2.com/download#section-builder>
  + <https://bitbucket.org/slidetrip/idiagram/issues/3/remember-to-optimize-qtip>
  + Proposed example for content of ttStyle is using an actual json structure, which can be inserted into the options parameter of each qtip that is created. At some point this and all fields in the database should be parsed for potential security problems:
  + Where/how to format/name the tooltip styles? Looks like it can be done in javascript, see these examples:  
    <http://codepen.io/gtb104/pen/Crfim> - in this example the ‘ttStyle’ name is set in the JS as ‘foo’  
    <http://codepen.io/storskegg/pen/gsfHj>
  + Qtip2 documentation:
    - <http://qtip1.com/docs/tutorials>
    - This whole page. The positioning is in this part of the page: <http://qtip2.com/options#position>
    - In the code that I have now, I am not using the offset at all. I noticed that there is an offset parameter in the style option example: <http://qtip2.com/options#style>
  + So I believe we need to: ready for testing  
    To get this to work,
    - I had to reload the database with the ttStyle with the name of the style object,
    - which is found in /public/demo/js/customstuff.js
    - Again, the style option docs are here: <http://qtip2.com/options#style>
    - At some point remove ttPosition, ttTopOffset, and ttLeftOffset from the DB schema.
  + A short 9-minute video that talks about the qtip stuff, and how to tweak it while debugging in chrome: <https://youtu.be/UaF_oJD4AC8>
  + Example of tooltip stuff working, though it seems you have to use the d3.js library: https://codepen.io/recursiev/pen/zpJxs

### IE 11 Compatibility - Fixed

Updated to the new version of Jquery - which seems to play nice with IE 11.

Need to fix invalid state error on: this.aud.currentTime = 0;

### Help & Preferences screen – MC Implementing

Add a ‘?’ button that floats with the zoom controls in the map-pane. Clicking on the help button, or hitting the ‘h’ key, will pop-up the help.html screen. The help pop-up could overlay the map-pane - like Kumu does (see the map= command, in the spec below, re using the map=command to overlay files on the map pane).

**Or** - your little temp pop-up got me thinking - that it’s probably even better for the help pop-up to be separate window. That way the user could move it off to the side (where it’s still visible, not overlapping the map) and then go back to the map and try out the commands. Probably easier to implement that the map-pane overlap thing as well.

Preference - can go in the Help window - or maybe in their own window?

### Help Page / Help Key - Done

We need a new entry in the JSON file for the help page, e.g.:

"helpLink": "/docs/help.txt"

And hitting the ‘h’ key should trigger a modal help screen:

### Override (and Non-Override) Show/Hide – Working!

**Override & Non-override** - we want to be able to specify whether or not to override the parent’s. We’d like the option of having it work as it does now (the parent has control) OR being able to override the parent.

Maybe extend the on/off command to classes, e.g.:

on=.odd

And then, once we get that working, write override versions of the tween functions, e.g. **fadeo**, in addition to **fade**.

I tested this URL to fade classes on, and it works(!):

?+++&fade=.even,0,f,0.0&open=.even&fade=.even,3,f,1.0

But for kicks, perhaps we could write a custom function - e.g. **fadeon -** that would handle this more elegantly,

Alternately, we could leave the codes as is for the non-override case, and use the custom function for overriding - in case that helps. The custom function could also be a good way to test different approaches.

When specifying a CSS action - either from the DB (openCSS / closeCSS) or using a tween command - e.g. fade=.odd,2,f,1 - we would like to be able to **optionally** *override the parent’s CSS settings.* See: <http://stackoverflow.com/questions/946645/how-to-ignore-parent-css-style>

<http://stackoverflow.com/questions/11178673/how-to-override-important>

Opacity is not an inherited property. See:

<http://www.w3.org/TR/SVG/masking.html#OpacityProperty>

***However:*** “Opacity is not inherited, but because the parent has opacity that applies to everything within it. You cannot make a child element less transparent than the parent, without some trickery.”

See the test map: <http://45.55.23.31:3000/demo/things2.html>

For example, we’d like to make all the class .odd things visible even if the parents of those .odd things are off (visibility:hidden) or closed (opacity:0.0).

All the Things on the page are either class .odd or .even. If you option-click on A to open it all up, you can run a command like:

?+++&fade=.odd,2,f,.2

And the .odd Things will fade.

Or see the database-specified **CSS Controls** – in the map – which turn .odd or .even off/on.

What we’d like to happen is, *when A is closed,* so no B’s or C’s a visible, to run the command:

?+++&fade=.odd,2,f,1.0

And have all the odd things become visible.

LM to try adding an additional rule to the classes added from the SVG, e.g. .odd, so that they will override their parent’s visibility setting.

Note that to turn things off we use: **visibility:hidden**, and for close: **opacity:0.0**  
So the fade example above might not work as it may need to override BOTH the visibility and the opacity e.g. it would have to send: **{ visibility:visible; opacity:1.0 }**

### Tween around center - Works great

The tween commands such as **scale** and **rotate** should do so around the object’s center point, rather than the registration point (the upper-left corner). This should be easily accomplished with the CSS transform origin:

transformOrigin:"50% 50%"

I’ve also purchased a shockingly green Club Greensocks license, and the full complement of GSAP code is on the dev site: /public/GreenSock

### Toggle Switch - Works!

See: <http://45.55.23.31:3000/Arcadis/arcDemo2.html>

And ArcStory.html - line 114 - 128

Make the Connections toggle switch in the story pane, accordion **Controls,** *do something* - like toggle the .connection class fading by sending the URL commands that are just above the switch.

Here is the link I used to turn on the connections before clicking the toggle button:

<http://45.55.23.31:3000/Arcadis/arcDemo2.html#/?on=Map&on=.link&panx=3500.0&pany=2500.0&zoom=1.000&wpane=300&epane=0&story=ArcStory.html>

Here are the links. All I did here was give them an id:

<a **id="connectionOff"** href="#/?+++&fade=.link,2,f,0" > <span style="color:black; font-size:12px">Connections Off</span> </a> <br>

<a **id="connectionOn**" href="#/?+++&fade=.link,2,f,1.0" > <span style="color:black; font-size:12px">Connections On</span> </a> <br>

Here is the javascript that I used to click those links:

<script>

$("#myonoffswitch").click(function(e){

var link ;

if (e.currentTarget.checked == true){

link = $("#connectionOff")[0]; //button is toggled off, so turn off connections

} else {

link = $("#connectionOn")[0];

}

var event = new MouseEvent('click', {

'view': window,

'bubbles': false,

'cancelable': true

});

link.dispatchEvent(event);

});

//# sourceURL=ArcStory.js **← This allows me to see this source in the debugger.**

</script>

### Keyboard Focus Issue – Code commented out for now.

In future, we should attempt to distinguish when you’re in a *text box*, vs. clicking on other active elements in the panes, e.g. links, accordions, buttons, etc., which cause the keyboard focus to shift away from the map-pane.

When we ‘fixed’ the code to enable keyboard input to the forms (in the info-pane), we also introduced a ‘bug’ where if the focus is not in the map-pane the map controls (pan, zoom, etc.), and slide controls, don’t work.

e.g. in the Arcadis map, click on a story-pane accordion, then try to pan/zoom with the keyboard.

As we don’t have any form-inputting at the moments - it would ok for now just to comment-out the type-in-form feature.

### Home & End keys - Done

For completeness, we should have the Home / End keys do the same as shift-comma / shift-period – go to the first / last slide:

Key code action

end 35 go to last slide

home 36 go to first slide

### importSpread UI - Done

Some of the UI bits on the import page have disappeared: the upload filename, the “Cool …”

I am trying to maintain consistency in naming conventions, so OverrideDB is now overrideDB. Also I changed <http://malsup.github.com/jquery.form.js> to <https://malsup.github.com/jquery.form.js>

### Printing Window/file Name - Done

Might it be at all possible to say, use the regular map window title, for the printing window title, as opposed to cryptic monikers like: blob\_http%3A\_\_45.55.23.pdf ?

### Pan & Zoom Tween – working

Add global pan/zoom duration variable to the JSON file - Note: values with decimal numbers MUST have a leading 0 if you want 0.4 (.4 will cause the json file to not load because it is invalid).

And

So that we can do slower pan/zooms - a URL tween command:

/?+++&**panzoom** = duration, xpos, ypos, zoom

### Rotation Bug – Working

The problem is that artwork set to display:none (when turned off) don’t exist to the browser - or have real coordinates - thus they don’t get rotated properly when they’re turned off.

The solution: for things to be tweened, temporarily set display:none to opacity:0.

-----------------

Perhaps the obvious fix to try would be to replace

transformOrigin:"50% 50%"

With a ‘manual’ rotation point something like:

var svgRect = this.uno.getBBox();

var centerX = svgRect.x + svgRect.height/2;

var centerY = svgRect.y - svgRect.height/2; // subtraction as Y is neg. from UL corner

TweenLite.to(svgElement, 1, {rotation:angle, svgOrigin:"centerX centerY"});

See “**Challenge: transform SVG elements around any point in the SVG canvas**” in

<http://greensock.com/svg-tips>

---------------------

It seems that things don’t take well to be turned around while closed. To produce the bug, load:

<http://45.55.23.31:3000/Arcadis/arcwayv1.html>

Go to **Controls** - and rotate the map to something other than **default**. Then click on **Open MtO, PtW, DtR** to see where things have spun off too. Different combinations of closing-rotating-opening UNOs will produce different results. I’ve even seen the *scale,* as well as the position, go astray.

### Tween Compound & Multiple Classes/IDs – Working

Firefox (on the Mac) bug - it seems to be balking at the single-quotes: - Fixed

------------------

I would be nice to tween multiple classes with a single command, see: <http://greensock.com/forums/topic/12263-same-tween-on-multiple-elements-in-timelinelite/>

So we could issue commands ***with*** a comma between the classes, in order to tween multiple things at once. Such as:

&scale=’.step3, .step4’,1,f,1.5,1.5

That would scale *both* the classes .step3. and .step4.

AND

So we could issue commands ***without*** a space between the classes, in order to tween more specific CSS selections.

As far as I can tell from reading this: <https://css-tricks.com/multiple-class-id-selectors/>

We should be able to specify tweens on concatenated CSS classes and/or IDs.

For example (I just ran into this need), in the Arcadis map, I’d like to increase the size of the objects in the third ‘level’. The map is already separated into levels with the class names: .step1, .step2, .step3, .step4. So, to increase the level 3 size by 50%, the command would be: ?+++&scale=’.step3>.ooo’,1,t,.5,.5&scale=’.step3>.vvv>.ccc’,1,t,.5,.5&scale=’.step3>.arclabel’,1,t,.5,.5

Or, to put it more succinctly - putting the compound AND multiple CSS in one command:

?+++&scale=’.step3>.ooo, @Steps>.vvv>.bccc, .step3>.arclabel’,1,f,1.3,1.3

* **Do We Need a gazillion Fade commands in url? – Working**

They do rather pile up. But sometimes you’ll really want the *one* command that makes that map look the way you want (with the link you can send to people). But it doesn’t seem easy to intelligently remove the redundant ones. I think you’d have to look for repeat ‘fade=unoid’ strings and remove if there’s more than one (this would also apply to all the tween commands).

So do you have an idea of how to do it? Yes, just re-use my code that already does it.

### Animations Not Playing - use ‘segmentFile’

I just update the Systems Agency map to use the new setup:

<http://45.55.23.31:3000/sa/samap.html>

But the animations are not playing - see **Play Steps** at the bottom of the story-pane.

In present.js, the Segments[] seem to be undefined.

### Remove Duplicate URL Commands - Implemented

Now that you’ve done the magic to prevent multiple tween commands from piling up in the URL, it would be good purge the other duplicate commands that can accumulate. The idea would be – before posting the URL to the address bar – to check each bit of text between &’s, and delete any duplicates.

It’s easy to get things like this in the URL (after hitting some of the toggle switches a few times):

arcwayv1.html#/?on=Map&on=.enabler&on=.enabler&on=.enabler&on=.enabler&on=.enabler&on=.enabler&panx=3200.0 …

### Accordion Reset Issue – Fixed

The function, processCommandsInURL()

under case "story": was loading that story file every time (but not info files).

--------

In some browsers – Firefox, IE – the story-pane accordion gets reset – the**Introduction** accordion gets opened and the Perspectives & Controls lose their settings – every time the user clicks on a story-pane element e.g. a slide link or toggle switch, etc.

So we either need to A. prevent the story-pane html from resetting when a URL command is sent to the page (better solution), or B. capture the currently active accordion and all the control settings, and make sure it stays set to that state when the page resets. See:

https://api.jqueryui.com/accordion/

You’ll see that I’ve added the getter and setter functions to AcrStory.html, but I don’t know when/how to call them to enforce the current accordion state.

If we need to store the active state we could use The sessionStorage Object:

<http://www.w3schools.com/html/html5_webstorage.asp>

### importSpread UI - CamelCase on the DB names!

The importSpread UI is working correctly in /Arcadis, but when I copy that same importSpread.html file to other directories (/demo/importSpread, /sa/importSpread, /se/importSpread), and just change the name of the DB folder, the UI on the importSpread page doesn’t work right.

### Custom Tweens Examples - Done

Move the standard tween code –  e.g. move, rotate, fade, etc. –  into customstuff.js - to provide some starting points for custom modifications.

### Stop including all the stop commands in URL in an animated presentation. working

### Bug: .ccc class as a child combinator in tween commands - not a bug

**Answer:** This is because there is no UNO id that has a direct child with a ccc class. I prepend vvv to the uno id of a vvv class, and also a ccc to the uno id. So in this case, you could select the element with '@cccA' or '@vvvA > .ccc' or '@A .ccc' (but then in this latter case you would select all .ccc descendants of #A)

------

The .ccc class does not seem to work in tween commands.

Using <http://45.55.23.31:3000/demo/radial.html>

These commands work - using .vvv or .ooo as a child combinator, or .ccc on it’s own:

?+++&scale='@A>.vvv',1,f,2,2

?+++&scale='@A>.ooo',1,f,2,2

?+++&scale='.ccc',1,f,2,2

But specifying .ccc as a child combinator, does not work:

Not this ?+++&scale='.ccc',1.5,f,1.4,1.4, but this:

?+++&scale='@A>.ccc',1,f,2,2

### “On” command is disappearing from the URL address box – Working

Remove the closeall= command from the address bar - as we do the close= commands.

Here’s what I’m seeing. Note that there are 2 scenarios with open/close-all: the URL command, and option-clicking an UNO. Testing with the Arcadis map, and the ArcadisWay UNO (the **Ways of Working** at the top):

**URL Commands**

If the URL contains &openall=ArcadisWay or &open=ArcadisWay, and you send ?+++&closeall=ArcadisWay, the closeall does not appear, *however* the openall or the open remains in the address bar - which is wrong.

**Option-Click**

If the URL contains &openall=ArcadisWay and you option-click on **Ways of Working,** the openall is removed correctly, *however* the closeall remains in the address bar - which is wrong.

If the URL contains &open=ArcadisWay, and you option-click on **Ways of Working**, the closeall remains in the address bar and open remains in the address bar - both wrong.

-------------

For example, this snippet in the arcadis: &off=.enabler&on=.enabler&fade='.enabler',1,f,1.0

This is now fixed. Instead of deleting just the extra command/value pairs, it was deleting all duplicate commands with no regard to what the values were. So now it will only delete extra command/value pairs.

### Info-pane Loading Issue – Works most of the time

If the URL get filled up with extraneous stuff, then it can go wrong: it initially puts up the right info-pane content, but then replaces it with something else. This behavior is different from using the key - which seem quite reliable.

I'm not sure if it's an important issue to solve right now - it usually works ok - as long as gunk hasn't accumulated in the URL.

-------------------

This works when you add the hover class to the anchor tag. If the object is already open when clicked on, it does not open it again. If it is not, then the info pane is still not opened because it is only triggered by hovering, which you have to do in order to click on it.

You are probably thinking that if we can show the info pane with keys, we should be able to do it with clicking. I know there is code that will open the pane for last uno in the url, which may cause some complications if I mess with the code.

This code looks for any open commands, if not there, looks for on commands. If you also want openall commands then let me know. I had to copy some code that is doing this for clicking on hover-classed anchors. If there are multiple open commands, this may or may not open the info panel for the last open. Let me know if there's a problem with that.

----------------

In <http://45.55.23.31:3000/Arcadis/arcwayv2.html>, the slide-links in the perspectives accordion, when you move through the links using the keys, the info-pane loads the correct information. But if you click on the links in the story-pane, the info-pane does not get loaded.

### Keyboard focus for forms with textboxes – Works

If a shortcut key is pressed while the user is typing in a textbox, don’t execute the shortcut command.

### Arcadis Interaction - Working

**Interactivity Options –** for the modifications below, it would be really nice to have variables that control the interactivity behavior. These could be set in the map’s json file. For instance:

hoverOpenClose = true; // works as it does now

hoverOpenClose = false; // disables the open/close

clickAction = “openClose”; // works as it does now

clickAction = “gotoz” // runs the gotoz. The url is still updated the same as before. If the uno is in a close state, it will not open it before zooming into it. Pressing c-click will produce the original click action.

clickAction = “somethingCool” // for future use... easy button to solve world hunger - excellent idea

**Hovering** – disable open/close on hovering. Hover should show tooltip / info-pane, but that’s all.

**Clicking** – disable open/close on clicking. Instead, clicking should on unoX should run a gotoz=unoX.

**Gotoz zooming** – as we did few days ago, delete the *second* occurrence of /0.83333 - so that the gotoz function zooms to the exact width or height of the UNO.

**Note:** if hovering is disabled, and clicking is normal, then your results may vary, since I assume when clicking that you have hovered on a group, so I do not re-open it. Let me know if there are problems.

### Set up idiagram.com droplet – Up and running

We should set up a for-profit production droplet (instead of the non-profit systemsagency.org) where I can put the Arcadis map - and leave the development site for development.

LM to clone the current SA production site to a new place on the SA droplet - to serve as a home for idiagram.com - including free https from Google. Then let me know the DNS coordinates so I can redirect from my ISP - if they’re different from systemsagency.org.

### Password for Arcadis production folder – user registration implemented

Password - or preferably with the user login - to prevent access to production site folders.

I created a new folder on systemsagency.org. It is called **/var/www/production/authorized**

All sub-folders in **authorized** will require a log-in. Currently, it is easy to gain access to these folders because all one has to do is register and they automatically get access to these folders. At some point it will be nice to have an admin page that gives permission to users, allowing them to only access certain folders.

To change a folder from being public to non-public, you simply move the folder from the ***public*** folder to the ***authorized*** folder. If you have a copy of the same folder in both ***public*** and ***authorized***, the server will always serve the folder that is under ***public***, because it always looks there first to see if the subfolder is in there.

From the user’s standpoint, they see no special folder in the URL. They don’t see ***public*** or ***authorized*** in the address.

Right now the **importSpread.html** only works from the **Arcadis** and **demo** folders, because of the https requirement -- it can’t pull in js files from remote http servers. They have to be http**s** servers. So I need to go through the other folders that contain importSpread.html files and have them point to local js files.

### Info-pane filling on click - FIxed

When clicking on an UNO, the info-pane is not getting filled with that UNO’s long description. This happens when clickAction = “gotoz” in the json file.

See <https://idiagram.com/arcadis/arcwayv1.html> - hover on and UNO, then click on that UNO. The info-pane shouldn’t change, but it does.

### Additional hover options - Working

Change the hover options (from true/false) to add all options for hovering. *hoverOpenClose* is defined in the json file (with the same name as the map’s html file):

**Interactivity Options –** there are several options for the actions that are triggered by hovering or clicking on an UNO:

"hoverOpenClose":"both"; // hovering will open a closed UNO, and close an open one

"hoverOpenClose":"open" // hovering will open a closed UNO i.e. if the UNO is closed, hovering on it will open it, and then hovering off it (without clicking) will close it. If the UNO is open, hovering on it will do NOT close it.

"hoverOpenClose":"close" // hovering will close an open UNO

"hoverOpenClose":"none" // hovering will not open or close an UNO - but will show tooltip / info-pane content if it exists.

### After login folder needs to change from order - Working

Changed to “home.”

### Commands without parameters - Working

I accidentally sent an incomplete command:

….&zoom

with no ‘=’ or value. It resulted in a zoom=Nan, which broke interactivity. So we should probably handle incomplete commands more gracefully.

### Hover/Tooltip timing bug – Working

Rapidly moving from one UNO to another confuses the tooltip syncing / logic - causing the tooltips to flash on/off, until the user again moves slowly off/on and UNO.

Seen in acf.html (more apparent there as the UNOs are so close together) and in arcwayv1.html (where the space between UNOs makes it less apparent, but moving across the closely spaced UNOs will cause the problem).

Fixed by code tweaks. "showDelay":0, and "hideDelay":0 added to the JSON file.

### Info Pane Filling Bug – Fixed

See: <https://gsap.systemsagency.org/acf/acf.html>

* On loading the map the URL has no open=unoX commands, and you can hover on any of the boxes and the info-pane gets filled.
* If you click on one of the small boxes, the URL is appended with e.g. &open=CBT6
* Now, if you hover on a *closed* UNO, the info-pane does not get filled (this is the bug) - the info pane retains the information for CBT6

**LM Fix**

I found some fairly recent code. When you call an onURL command, which is executed with a hover, such as, "+++&open=L7", it calls this code after populating the info-pace with the correct info. This has the effect of updating the URL, and when the URL is updated, the code automatically populates the info-pane with the last item in the URL. Here's the code:

$.address.parameter("rotate", ""); //remove all these commands from url

$.address.parameter("fade", ""); //remove all these commands from url

$.address.parameter("scale", ""); //remove all these commands from url

$.address.parameter("move", ""); //remove all these commands

$.address.parameter("stop", ""); //remove all these commands from url

$.address.parameter("stopall", ""); //remove all these commands

$.address.parameter("masteron", ""); //remove all these commands

//removeCommandFromAddressDotValueString( $.address.value(),"masteron"); //remove all these commands

$.address.parameter("masteroff", ""); //remove all these commands

$.address.parameter("closeall", ""); //remove all these commands

$.address.update();

I could comment out the $.address.update() and see if it will update itself later to remove all those commands. That $.address.update() is ugly -- circular code.

It worked when I commented out $.address.update() . And the extra commands do go away when you click on something or manually update the URL.

### Hovering / Tooltip Issues – good enough for now

Tooltips not being fully in-the-moment, recognizing where they are. Especially in slow browsers like Safari it’s not guaranteed that, hovering from across closely-spaced UNOs, a tooltip will appear. The user then has to give the mouse an additional nudge to make the tooltip appear.

**LM - code changes:**

The class, SmartTooltip, currently found in idiagram-svg.js, contains the logic for handling explicit calls to the qtip api. All I did in this class was to comment out lines of code that included .show() or .hide(). If we know that we are going to let the hovering event automatically show or hide, then at some point I can eliminate this class altogether.

It would be nice if the hovering was fast, accurate, and definitive about where the mouse is (why *are* the tooltips so slow in Safari?). However, it’s not clear how much more time we should spend trying to fix something that works pretty well in most/fast browsers.

How well does it work on IE 11? On IE there are some cases where quickly hovering on an element does not open the tooltip, and one must move the mouse a bit before it will.

### Tweening: Custom Function Modifications - working

**URL - done**

The custom function commands pile up in the URL - they should be removed like the standard tween functions.

To remove the custom function from the URL, in the function use:

$.address.parameter("myFunctionName", ""); //remove all these commands from url

**Remove ‘run=’ requirement – working**

Do we really need the ‘run=’ for the custom functions? (and then the different syntax - the addition of the ( ) around the parameters). It would be nice if functions in customstuff.js could just be found while parsing the URL, so we could use the same syntax in the URL for standard and custom functions, e.g.

?+++&scale='@vvvADDP',2,1,0,1

And

?+++&scaleLeft='@vvvADDP',2,1,0,1

Now that I looked at the code, it seems to me that we could just eliminate the case: “run” and replace the default: with doing a hail-Mary to the custom functions, assuming that function will be in there. And if nothing is found there, then throw the alert:

var customFunction = window[keyName]; //custom function, such as move or fade

if (typeof customFunction === "function") {

var selectorAndResultArray = getSelectorAndResultArray(q[i][keyName]);

customFunction.call(this, selectorAndResultArray);

}

else

alert("Bad command in URL: " + keyName);

break;

### Tweening: Schemaless Spreadsheet Import to schemaless DB - DONE

Change the DB import from Excel so that it takes the keys from the first row, and the values from each subsequent row (if that row has a valid ID value) to form the key:value pairs in the DB.

Possibly helpful links:

<http://stackoverflow.com/questions/35799329/dynamic-html-form-generation-in-mean-stack>

<https://docs.mongodb.com/manual/reference/program/mongoimport/>

<http://stackoverflow.com/questions/tagged/schemaless>

And update code for DB form editing. - Check