

**UCLA Academic Technologies Services**

# New Run-Time Software Functional Specifications

3rd Draft

## Contents

<b>Revision History .....</b>	<b>1</b>
<b>Software Overview .....</b>	<b>1</b>
Set-Up .....	1
Anticipated File Structure .....	1
File Types.....	2
Definitions .....	4
<b>Navigation .....</b>	<b>5</b>
Keyboard Hot Keys .....	6
Fixed Hot Keys.....	7
Do we need the following shortcuts?:.....	7
Unassigned Keys (User-programmable?) .....	7
<b>Main Toolbar Summary .....</b>	<b>8</b>
File.....	8
Settings.....	15
Global settings .....	15
Advanced Settings.....	16
Lighting.....	16
Content Creator .....	17
Display.....	17
Binary restrictions .....	18
Narratives.....	19
Resources .....	19
Search.....	20
<b>Narratives.....</b>	<b>21</b>
Narratives List (Player functions only) .....	21
List of Narratives Bar (Player and edit functions) .....	22
Narrative Player (Player functions only) .....	23
User-Interaction during Narrative playback .....	24
Narrative Player/Editor .....	25
Creating a Narrative .....	27
User Interaction during Narrative Creation and Editing .....	27
Narrative Settings Dialog Boxes .....	28
Narrative Settings .....	28
Node Settings.....	28
Transition Settings .....	29
Overlay Editor .....	29
The Spline Editor .....	31
Points of Interest (aka Jump Points/Browse).....	32
<b>Branding option .....</b>	<b>32</b>
<b>Embedded Resources.....</b>	<b>33</b>
Displaying Embedded Resources .....	34

User Interaction with the Embedded Resources .....	34
Displaying Embedded Resources (adding and editing enabled).....	34
User Interaction with the Embedded Resources (adding and editing enabled).....	35
Embedded Resource Settings dialog box.....	36
<b>Screenshots .....</b>	<b>37</b>
<b>Movies.....</b>	<b>38</b>
<b>Creator functionality to accommodate on load .....</b>	<b>39</b>
Degree of freedom triggers.....	39
Creator group-level animations .....	39
Switches .....	39
<b>Graphic: Narrative Sequence.....</b>	<b>41</b>
<b>Graphic: Embedded Resource Sequence .....</b>	<b>42</b>
<b>Phase 1.5 and beyond ... ..</b>	<b>43</b>

## Revision History

Draft 1: Meghana Reddy, 6/1/2010

Draft 2: Working from draft 1, Eduardo Poyart and Lisa M. Snyder, distributed for 9/1/2010 meeting on 8/30/2010

9/1/2010 meeting held in the ATS Visualization Portal to discuss Draft 2. Attendees: Eduardo Poyart, Lisa M. Snyder, Scott Friedman, Jennifer Dillon, Meghana Reddy, Elaine Sullivan, Dave Sartoris, Zack Rynew, Bruce McCrimmon, and Itay Zaharovits.

Draft 3: Initiated on 9/2/2010, incorporating input from attendees at above-mentioned meeting.

## Software Overview

This *run-time software interface* is meant to provide “the mechanism to explore, annotate, craft narratives, and build arguments within a three-dimensional space” [NEHStartUp.doc, pg 5]. As such, it should contain the following functionality:

### Set-Up – Platform, installation, and filetype

- a. Must operate on Windows and Mac OS for users
  - i. For use in ATS Visualization Portal, must also operate on Linux
- b. Deliverable via simple download from ATS site to be determined, with an executable installation file.
- c. Model files (both raw and binary), narrative files, narrative content, embedded resource files, embedded resource content, and settings profiles are currently conceptualized as completely independent from the run-time software.
- d. Application window is capable of adjusting to field of view
  - i. E.g. if window changed from square to wide rectangle, rather than stretching out the current view, it adjusts the field of view to show more of the model horizontally
- e. Capable of importing multiple file types, *both raw and binary*, in any combination. Currently accepted file types are listed in the ‘File->Import’ section below.

## Anticipated File Structure

The current specifications assume that the software and model files are structured as follows:

- a. The New Run Time Software is contained in its own folder in the C:\Programs folder (on Mac, in the Applications folder)
- b. When working from raw files, the file structure is irrelevant (the loader will pull external references and textures from their locations as dictated by the raw files)
  - i. Content creators can import and reference Narratives and their content from any location and it is up to them to manage their own file structure.
  - ii. Exporting BIN and RBIN files will create a zip file that ‘packages’ all of the referenced files in the structure described below (ZBIN for reference).

- c. In situations where the end-user is working with content files packaged in the ZBIN file it can be located anywhere in the user's file system.
  - i. The software will unzip the ZBIN in memory for any given flight session
- d. An end-user may unzip the files, in which case, a model-specific folder is created
  - i. All BIN, Narrative, and Embed files for the model should be directly below the main folder level
  - ii. A folder for Resources is also directly below the main folder level
    - (a) The Resource folder holds all images, audio files, video files, and pdfs for both the Narratives and Embed.
  - iii. All paths within the BIN are assumed to be relative

## File Types

Conceptually, these are the files that the software will create and read:

1. **Raw model files:** the files from the various modeling packages supported that can be opened in their source programs (i.e., .flt files from Creator, .3ds files from 3DStudioMax, .kml files from SketchUp)
2. **Workspace files:** These files are the mechanism through which files are managed and aggregated in the software. Workspace files reference the other file types.
3. **Binary file:** BIN, RBIN, or ZBIN are all variations of the aggregated output from our software.
  - a. The binary files will include references to required narratives, and all settings and restrictions in place at the time of the file's creation. (To be determined: do binary files need to reference embedded resource files?)
  - b. The RBIN is essentially the same as the BIN, except that the content creator has imposed "use restrictions" on the file.
  - c. The ZBIN is the zipped version of the files.
  - d. Actual extension for the BIN, RBIN, and ZBIN files to be determined later.
  - e. The extensions-to-be-names-later will be associated with our new run-time software so that users can double-click on a file and it launches our software.
4. **Narratives:** All data required for the narratives, somehow aggregated.
  - a. Includes both the text-based components and links to the associated content.
  - b. **Paths** are subsets of Narratives that use the same file structure.
  - c. It is possible for calls to narratives to be included as part of a binary file. The content creator can set a restriction that limits end-user interaction to narratives; if none of the pre-defined narratives are available, the flight session should be aborted. (The files, therefore, should include flags that cross-reference one another.)
  - d. Narratives can be exported and shared as a stand-alone entity that can be imported into an ongoing flight session.
    - i. The Narrative file will include a model ID that must match the model ID of the BIN, RBIN, or ZBIN file being loaded.
    - ii. By selecting multiple Narratives, a group of Narratives and their content can be exported into a single package.

- e. Storing the imagery from narratives in a folder outside the binary allows end users to re-use the content for their own narratives (i.e., the folder becomes a database of content about the modeled environment). We just have to be sure to only circulate content we own or is in the public domain.
  - f. In Phase 2, we might address copyright more rigorously... with Phase I, we're leaving it to the content creators. In Phase 2, perhaps we can develop the imagery folder to have different access permissions based on ownership (i.e. a Public folder for open content and an inaccessible Private folder for copyrighted content).
5. **Embedded resources:** All data required for the embedded resources, somehow aggregated.
- a. Includes the text-based components and links to associated content. The software will load all embedded resource files that are in the model file structure.
  - b. The embedded resource needs to include a reference to the appropriate BIN/RBIN/ZBIN.
  - c. Embedded resources can be exported and shared as a stand-alone entity that can be imported into an ongoing flight session.
    - i. The Embed file will include a model ID that must match the model ID of the BIN, RBIN, or ZBIN file being loaded.
    - ii. By selecting multiple Embeds, a group Embeds and their content can be exported into a single package.
  - d. As with narratives, storing the imagery in a folder outside the binary allows end users to re-use the content for their own work (i.e., the folder becomes a database of content about the modeled environment). The problem we're not addressing is copyright ... we just have to be sure to only circulate content we own or is in the public domain.
6. **Settings/restrictions profile:** All text-based flags and fields associated with user setting preferences and restrictions to be imposed on end users.
- a. It is possible for these settings/restrictions to be included as part of a binary file, or as a stand-alone entity that can be imported into an ongoing flight session.

In Phase 1, the export function for narrative and embedded resource files creates links to a single 'content' directory, but does not include the content itself in any aggregated single file. This is an ongoing question ... should we include all content in the files or only include links? If narratives or embedded resources are exported, is the associated external content included in the file? Included in the appropriate file structure in a zip file? Or just left as paths to the content location at the time of the export? If aggregated, file management is simplified, but the files may become huge; if not, end-user has access to content for their own narrative/embed files, but has to manage the file structure for things to work properly. Is there a way to include both options and let the content creator decide? Meghana's comment: We definitely shouldn't expect the end-user to understand the local file paths! Use the Google Earth model—The Embedded Resources and Narratives files are essentially .kml, as they contain text content and links to all other content types. A zipped package containing 2 things—the resource file and associated Content folder—would be a .kmz. So the content creator can export just the file or a zipped package. One detail is that a plain resource file would have to omit features that have "broken links" to content that hasn't been imported.

Phase 2 of the software should include a repository solution for the different run-time software file types (models, narratives, and embeds)

## Definitions

The following are definitions for the terms used throughout the functional and technical specifications:

1. **Raw model files:** the files from the various modeling packages supported that can be opened in their source programs (i.e., .flt files from Creator, .3ds files from 3DStudioMax, .kml files from SketchUp)
2. **Binary file (aka BIN):** the aggregated output from our software that merges all geometry, textures, switches, settings, etc. The conceptual variants of the BIN are described below (WBIN, RBIN, ZBIN).
3. **Workspace files (aka WBIN):** This may or may not be a variant ... the distinguishing feature is that this BIN includes links to multiple files. Mostly this is a semantic distinction ...
4. **Restricted binary (aka RBIN):** the aggregated output from our software that merges all geometry, textures, switches, settings, possible calls to narratives, etc. The only different from a BIN is that the RBIN includes end-user restrictions defined by the content creator
5. **Zippered binary (aka ZBIN):** The BIN and all its assorted files, packaged for distribution such that if an end-user unzips the file, it would be arranged in a nice orderly file structure. The end user can also work directly with the ZBIN in a transparent way (i.e., leaving it zipped) ... this is the most desirable use of the ZBIN.
6. **Content creator:** one who develops or controls the raw model files and is using our software to interact in real-time with their environment and package it for use by others (the end-users)
7. **Modeled environment:** the extent of the virtual 3D space developed by the content creator, either as raw files or in binary format
8. **End users:** the secondary users of the modeled environments, conceptualized as scholars, educators, students, or museum visitors.
9. **Narrative:** A linear argument constructed within the modeled environment that can be 'played' by subsequent end users
10. **Narrative bar:** A horizontal band across the top of the viewport that houses all controls related to creating, editing, or playing narratives (aka List of Narratives)
11. **Narrative Player/Editor:** The functional element that allows Narrative playback and editing ... virtually the same layout with minor differences depending on whether or not editing is enabled: the player has all player icons plus an 'edit' icon; the editor has all the player icons plus all of the edit icons.
12. **Narrative node:** A defined key frame within the modeled environment upon which can be added image and textual content as part of a narrative (think beads on a necklace – the necklace is the narrative, the individual bead is the node). Narrative Nodes are accessed through the Narrative Node Bar)
13. **Overlay:** The PowerPoint-like arrangement of text and images that will be displayed on the viewport. Used for both narrative node creation and branding. (For Narratives, accessed through the Narrative Node bar, for branding, through the Settings menu.)
14. **Overlay editor:** The function that allows the addition of text and imagery as an overlay on the narrative node or branding bar.

15. **Embedded resource:** Any piece of reference material that is spatially identified with a coordinate in the modeled environment
16. **Embedded resource bar:** A horizontal band across the bottom of the viewport that houses all controls related to creating, editing, or launching embedded resources
17. **Branding bar:** An overlay of text or imagery (logo?) that the content creator defines and may require during all end user flight sessions
18. **Spline:** A system-generated line/arc created by interpolating between the xyz coordinates of two adjacent nodes of a narrative
19. **Spline Edit Nodes:** Control points on the spline that can be manipulated to finesse the auto-generated line/arc
20. **Movie:** Video clip exported by the software of the currently loaded modeled environment.
21. **Screenshot:** A static image exported by the software of the currently loaded modeled environment.
22. **Path:** A set of xyz coordinates connected by splines. A path does not include any orientation settings, but still uses the same file structure as Narratives.

## Navigation – Navigation options once a model is loaded

First-person flight simulator, with control in all directions, designed primarily for a three-button mouse. The software should also be compatible with a (1) two-button mouse, (2) scroll mouse, and (3) WASD keyboard commands. (Maybe in Phase 2, a joystick and the hockey-puck thing.)

1. **General navigation comments**
  - a. Left click accelerates in concert with velocity setting
  - b. Right click decelerates/reverses in concert with velocity setting
  - c. Middle click stops all forward motion and resets speed to 0.00
  - d. Spacebar freezes the scene and resets speed to 0.00
  - e. The 'C' key toggles collision detection on and off
2. **Fly**
  - a. User has control over speed, xyz and hpr (like a flight simulator)
  - b. Uses uSim as basis for flight controls
  - c. The 'F' key toggles between flight and drive mode
  - d. First check user restrictions (free navigation may not be allowed)
  - e. If depressed while in drive mode, 'F' changes to flight mode
  - f. If depressed while playing a narrative and there are no restrictions in place, 'F' pauses the narrative and lets the user go into free flight mode; resuming play on the narrative will return the user to either (1) their prior location along the narrative's spline or (2) at the beginning of the node they were on when they paused the narrative.
3. **Drive**
  - a. User has control over speed, xy and hpr
  - b. Uses uSim as basis for drive controls with the addition of WASD keyboard commands
  - c. This requires detection of solid objects below the user, such that she stops "falling" when solid ground is reached
  - d. The 'F' key toggles between flight and drive mode
  - e. First check user restrictions (free navigation may not be allowed)



- f. If depressed while in flight mode, 'F' changes to drive mode (movement tethered to a fixed height above the ground)
  - g. If depressed while playing a narrative and there are no restrictions in place, 'F' pauses the narrative and lets the user go into drive mode; resuming play on the narrative will return the user to either (1) their prior location along the narrative's spline or (2) at the beginning of the node they were on when they paused the narrative.
  - h. When in drive mode, WASD keyboard commands are also enabled.
4. **Attached to a Narrative**
- a. When 'on a narrative' the user is following a linear path through the environment
  - b. Primarily controlled by the Narrative settings (transitions, timing, keypress required for next Node, etc.)
  - c. User can take control of hpr by taking control of the mouse and clicking it in the viewport.
  - d. User can also take control of speed by depressing the left & right mouse buttons.
  - e. If the user stops interacting with the software, a time out setting will return the camera back to the defined narrative settings. (Resume Narrative Control function)
  - f. Pausing the Narrative freezes all movement
    - i. If free flight is restricted, user can only choose another narrative or resume on current narrative
    - ii. If free flight is not restricted, pausing a narrative freezes all movement. Upon user-issued navigation commands, free exploration is initiated (fly or drive). (See Narrative Player, 9a)
  - g. A time out setting for the Pause will resume Narrative control. (This is specifically a feature for the museum kiosk ... if a user takes control for a bit and then walks away, the system needs to reset itself.)
  - h. See Narrative section below for detailed information
5. **Attached to a Path**
- a. A user can open the Path Manger and attach to an object attached to a path
  - b. The Narratives take the place of the uSim-style attach to paths functionality
6. **Eduardo's new experiments with interaction**
- a. Includes WASD keyboard controls (like all first-person shooters) (W= forward, S=back, A=strafe left, D=strafe right) plus mouse controls
  - b. Possible option to set a 'target' in the environment and rotate around that object.
  - c. USim navigation controls are default until Eduardo convinces us that his new way is better.

## Keyboard Hot Keys

Ongoing question – how many of the hot keys should be fixed versus completely customizable. Eduardo is lobbying for completely customizable; Lisa wants some fixed keys, and the remaining customizable if there are any left. Either way, a look-up table is required. Meghana suggests that the commonly used buttons be fixed, but all others can be set via a Hot Keys menu – the user has a list of available keys, like below, and can choose from a drop-down of functionalities.

The following are the shortcut hot keys that are commonly used in the existing pieces of software. In some fashion, these should be accommodated. Where there are multiple options, we have to figure out what to do. These commands should be regarded as quick access to functions for experiences users. In

making the software more intuitive for general users, a better option is to include these commands within on-screen menus and widgets which can appear automatically as needed (e.g. hover over map to retrieve the map zoom and clip plane settings).

### Fixed Hot Keys

The following are the current hot keys that are to be fixed in the new run-time software:

- W = NEW keyboard navigation (the WASD motion)
- A = NEW navigation command (the WASD motion)
- S = NEW navigation command (the WASD motion)
- D = NEW navigation command (the WASD motion)
- B = bird's eye (vrNav)
- C = toggle collision on/off (vrNav)
- F = Toggles between fly and drive mode (uSim has both F and D as hotkeys)
- L = lighting (vrNav) Will this happen?
- M = map view toggle (vrNav)
- ] = next switch index (N in vrNav)
- [ = previous switch index (P in vrNav)
- = equals next switch ID
- - equals previous switch ID
- R = Start/stop record (used in both uSim and vrNav)
- T = Toggles between textured output, scribed, and wireframe (neither existing pieces of software has this, but I can see benefits)
- Alt and mouse = strafe (uSim)
- Spacebar = freeze forward motion, reset motion speed to 0.0 (uSim)

### Do we need the following shortcuts?:

- Screenshot (F2 in uSim)
- Write coordinates (F4 in uSim; is this still needed? Eduardo thinks raw coordinates should be forbidden)
- Exit/Quit (Esc in uSim; Q in vrNav)
- Pick (Ctrl + left mouse in uSim)
- Advance time switch (I in vrNav)
- Previous time switch (J in vrNav)
- Map zoom in and out (0 & 9 in vrNav)
- Map clip plane up and down (Page Up and Page Down in vrNav)

### Unassigned Keys (User-programmable?)

- E =
- G =
- H =
- I =
- J =
- K =

- N =
- P =
- O =
- Q =
- U =
- V =
- X =
- Y =
- Z =
- F1 through F12 =
- Esc =
- 1 through 0 =
- Combinations with Alt
- Combinations with Ctrl
- Combinations with Shift

## Main Toolbar Summary

The top toolbar is a static list of functions that cannot be removed/hidden (except in Full Screen navigation mode). It encompasses all of the ways in which the interface can interact with the models and *modify* what the user views.

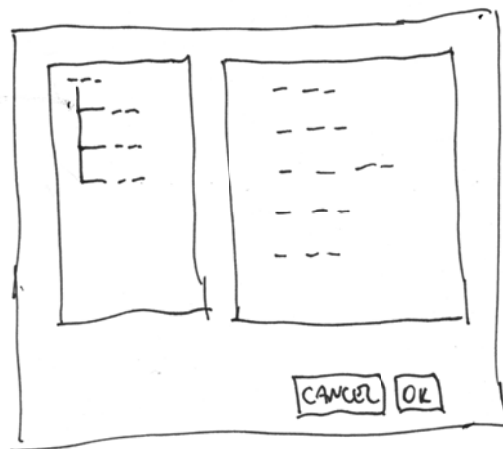
The main toolbar includes options for File, Settings, Narratives, Resources, and Search. Each of these has its own menu options as detailed below.

### File

The file options are those that involve modifications and/or actions on a file level.

1. **Open** – Opens a BIN, RBIN or ZBIN file.
  - a. The software loads the BIN/RBIN/ZBIN file
  - b. The software loads all associated Narratives, Embeds, and raw model files referenced in the (RZ)BIN file
  - c. Interactions with the various BIN files must always include checks against the restriction settings. (Some actions may be blocked so the displays will need to adjust accordingly.)
2. **Import (need new word?)** – The import function adds new files to the session (e.g., raw model files, settings, Narratives, Embedded resource files) with the intent of displaying the content and building a binary file.

• OPEN BIN FILE



Importing files will only add references to the binary file. The content will not be automatically aggregated – later on, the user can export a new BIN to aggregate the referenced BIN and FLT files become embedded through another operation described in Section ##.

- a. The software loads model files only in the following accepted formats.
- b. Raw model files saved as FLT (Creator), SKP (Sketchup), KML (Google Earth), OBJ, 3DS (3D StudioMax), DAE
  - i. As part of the loading process, the software should scan for and delete duplicate textures, and re-assigned texture pointers as necessary to ensure the visual fidelity of the model
  - ii. As part of the loading process, the software should scan for and correct mal-formed textures (i.e., it should re-size textures to a powers-of-two size ... is this still necessary?)
  - iii. The software should be able to read textures in .rgb, .rgba, .int, .inta, .jpg, .png, and ... formats (are there others that should be on this list?)
- c. BIN or RBIN files
  - i. The software loads the BIN/RBIN/ZBIN file and adds a reference to the file in the currently open BIN file
  - ii. The software loads any associated Narratives and Embeds referenced in the (RZ)BIN file
  - iii. The software WILL NOT load any BIN files referenced in the file being imported. (This is to avoid potential recursion issues ... if a content creator wants to load multiple BIN files, they must take care to aggregate as necessary to work within these parameters.)
    - (1) An error message should be thrown that says 'file x didn't load ...' or similar
- d. Narrative files
  - i. The model ID stored in the narrative file must match the model that is currently loaded
  - ii. The appropriate model file must be loaded before its matching narrative files can be opened (might a content creator want to load a Narrative to edit it without bothering with the model?)
  - iii. In Phase 1, the software just reads the imported file and its links from its current location. In Phase 2, perhaps we'll deal with copying the file and its associated content to the local/active model directory (iTunes asks if you want to copy or not copy to keep your folders organized)
- e. Embedded resource files
  - i. The model ID stored in the embed file must match the model that is currently loaded
  - ii. The appropriate model file must be loaded before its matching embed files can be opened
  - iii. In Phase 1, the software just reads the imported file and its links from its current location. In Phase 2, perhaps we'll deal with copying the file and its associated content to the local/active model directory (iTunes asks if you want to copy or not copy to keep your folders organized)
- f. Settings profile files
  - i. If the user has specified starting position coordinates (x,y,z and h,p,r coordinates), the software should position the model appropriately upon load
  - ii. If no starting position coordinates are specified, the software should position the user so that the loaded model(s) are in completely in view.

- iii. If an RBIN file is loaded and a settings profile is imported, restrictions will be checked and only unrestricted settings are changed.
- 3. **Save** – Saves the file as a BIN file using the same name (only possible if a BIN file is loaded).
  - a. If an RBIN file is loaded, the Save function is disabled.
  - b. **Are BIN and RBIN files auto-saved?**
- 4. **Save As/Export (need new word?)** – From the 'File' level, the user can export the following four file types: BIN, RBIN, ZBIN, any combination of narrative and embedded resources files, and a settings/restriction file.
  - a. The BIN file is a binary file created from the content currently loaded.
    - i. All textures associated with the model(s) are compressed and included in this binary file. This creates the BIN file (a single, proprietary binary file).
    - ii. Any images and text associated with the branding overlay are saved within the BIN
    - iii. All paths and objects attached to paths are saved within the BIN
    - iv. Current settings are saved within the BIN
    - v. If Narratives and Embeds are loaded, a dialog box asks the user which files to include in the BIN zip.
      - (1) List of Narrative and Embeds allows user to require the file or not to load (defaults to require). In this case, the RBIN includes pointers to the narrative files selected in the dialog box and subsequent loads of that file must include those narratives.
      - (2) In Phase 2, we may include an option to aggregate all content into one file. We're not doing it now because this will likely create very large files, so the software should flash a confirmation of expected file size with an option to only creates links to the required Narrative and Embed files. (Also in Phase 2, maybe scan for duplicate uses of the same resources across Narratives and Embeds.)
  - b. The Restricted BIN (RBIN) includes user restrictions and which cannot be subsequently altered. The intent of this is to create an aggregate package in a binary file format that is meant for distribution to end-users who should *not* have access to the underlying raw model file.
    - i. If an RBIN file is loaded, the Export/Save As function may be disabled. (Phase 2 – include a separate encryption dll so that it's really hard to hack the RBIN.)
    - ii. If an export is allowed, flags on subsequent exports will be set per the restrictions settings in place at the time of the original save. (Should display how the settings are set so the content creator has to say OK or Cancel to change the settings.)
    - iii. These settings and restrictions are stored in the RBIN file itself.
    - iv. Any images and text associated with the branding overlay are saved within the RBIN and are forced on any subsequent exports
  - c. A ZBIN or RRBIN is the zip file is created of the BIN, the possible restrictions, all Narratives, Embeds, and their contents.
  - d. Raw narrative files and embedded resource files are saved during their creation
    - i. Exporting narrative(s) and embedded resource(s) aggregates the content called in the file(s) and creates a zip file (or a new directory?). Essentially a copy operation.
    - ii. Exporting narratives and embedded resource files requires an opportunity for the content creator to lock the file.

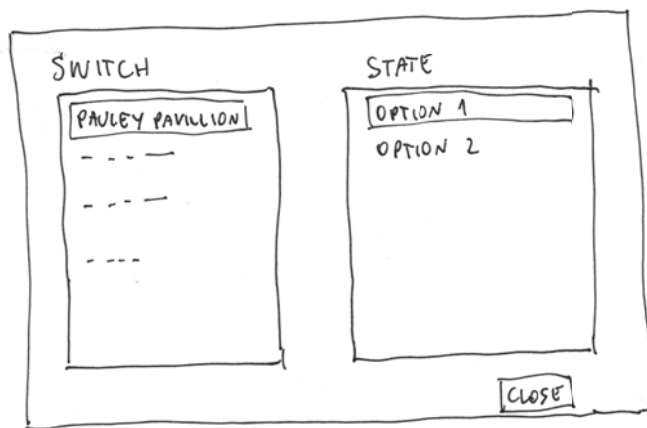
- iii. Multiple narratives and embeds can be exported at the same time and the software will include only a single copy of any content used by the files.
- e. Exporting the settings/restrictions/branding file creates a text-based file that includes all settings, restrictions, and branding elements in place at the time of export

• FILE / SAVE AS



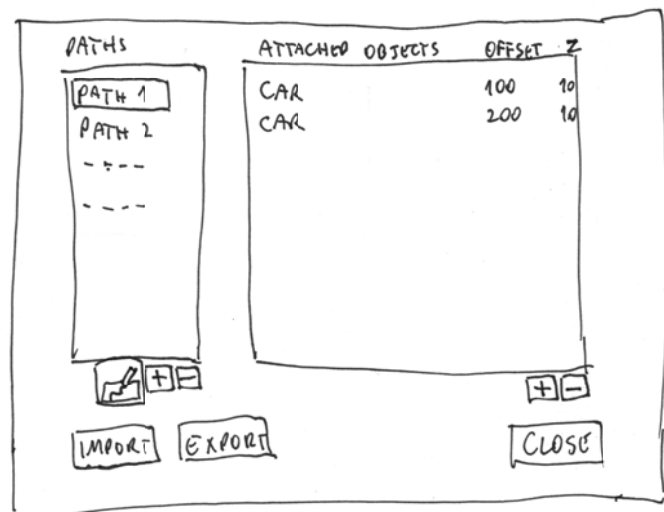
5. **Switch Manager** – A mechanism to display and choose from the various switches/options in the currently loaded model.
  - a. Switches can also be manipulated in the flight session by fixed hot keys or the time slider
  - b. Switches must be programmed in Creator's Switch Nodes; no other switching mechanism is accommodated at this time.
  - c. Provides a list of the available switches (pulling from the Switch Attribute ID)
  - d. Selecting a switch name displays a list of available options for that switch (pulled from the individual Switch Mask Names)
  - e. Selecting in individual Switch Mask Name updates the scene being rendered in the viewport.
  - f. More details on the switch mechanism are provided below.

SWITCH MANAGER



6. **Paths Manager** – A mechanism to manage the Paths included within the modeled environment, specifically, this manager is necessary for attaching objects to a path (currently used in uSim for cars, boats, and other forms of transit; in Phase 2 may be used for ‘people’ and other behaviors).
- Paths are similar to Narratives, share the same file structure and many of the same functions, but have their own file extensions.
  - Paths are managed by the Paths Manager window.
    - Whenever a path is highlighted in the Paths Manager window, the corresponding path appears on the viewport.
  - When paths are included in a BIN or RBIN file, they are saved within the binary file itself (not as a separate entity like Narratives or Embeds).
    - Objects attached to paths are also saved within the BIN or RBIN file.
  - Path files can be created, edited, imported, exported, deleted, have objects attached to them, and, as a general user, you can attach to objects during a flight session.
  - Create** -- The path creation procedure is the same as the Narrative Node-by-Node and Quick Path Record features. (In Phase 2, perhaps we shift to a strategy where the user “aims” to the position of the beginning and end of the path with a crosshair in the center of the screen. During this pointing, there is a “live preview” of the position of the path on the ground, to facilitate the operation.)
    - Once created, if the content creator wants to reuse the path with another model, it should be exported so that it can be imported into subsequent flight sessions.
    - If changes in speed are desired, the path creator should insert nodes at the appropriate point so that speed settings can be added and modified.
  - Edit** – The path is edited with the same spline editor as the Narrative.
  - Import** – adds a path to the scene
    - User clicks to import, chooses file from source, clicks DONE
    - Only allowable file format is that of the Path
    - User types Path Name (a short meaningful phrase that describes the path route for display in the Path Manager)
    - Sets coordinates to loop or not.
    - Imported file now included in list of paths
  - Export** – Saves all selected paths in a single path file
    - Saves all coordinates, splines, and speed settings

PATHS MANAGER



- ii. Saves the settings for objects attached to the paths, but only links to the objects attached to the path.
  - i. **Delete** – removes a path and all its associated objects from the scene
    - i. User clicks the path name and hits delete
    - ii. Error message says 'ARE YOU SURE?'
  - j. **Attach Object** – Lets the content creator attach objects to the Path
    - i. User selects path name
    - ii. Hits 'Attach Object'
    - iii. Browses for file to attach (allowable file formats TBD)
    - iv. Defines offset (distance from beginning of path to start object on load) and other variables
    - v. The speed is controlled through the
  - k. **Attach to Path** – Allows user to attach to a path
    - i. Uses the speed settings established in path
    - ii. Gimbal is automatically on
    - iii. Place to adjust offset from path (to assure perfect placement in the train or whatever).
7. **Links Manager** – A mechanism to manage the load of multiple and various file types (this is the equivalent of the load list in the uSim resource file).
- a. Provides a list of the files associated with the currently loaded model.
    - i. If no model is loaded, the list is blank
  - b. For large existing UST projects, this will require the initial construction of a BIN file to reference its component parts
  - c. Dialog box shows the file names of all loaded files: raw model files (.flt, kml, .3ds), BIN/RBIN files, Narrative files, Embed files
  - d. Check boxes or radio buttons indicate whether or not the contents of the file are hidden/shown or required to be present on load of a subsequent RBIN. (Phase 2 may include option to aggregate selected content into a subsequent BIN export. This would address problems we may encounter stemming from misplaced files or broken links.)
    - i. Newly imported files are, by default, set at 'show' and 'not required'
    - ii. If a files is required, it is, by default, also shown
    - iii. Hidden files are not included in subsequent exports
    - iv. For Phase 1, all file types are set to be links only (i.e., NOT aggregated)
  - e. Icons at the bottom of the dialog box allow files to be added and deleted from the list
    - i. Adding (+) launches the import function
    - ii. Deleting (-) removes the contents of that file from the active session.

LINKS MANAGER

	HIDE	REQUIRED
--- .FLT	<input type="checkbox"/>	<input type="checkbox"/>
--- .BIN	<input type="checkbox"/>	<input type="checkbox"/>
--- .NAR	<input type="checkbox"/>	<input type="checkbox"/>
--- .EMB	<input type="checkbox"/>	<input type="checkbox"/>

☐ ☐



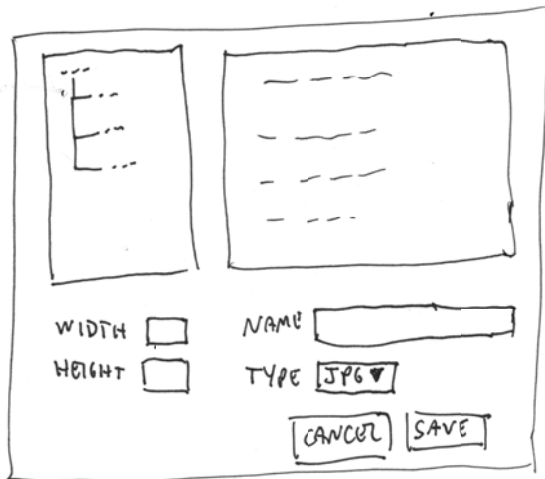
## 8. Explore Contents Folder

- a. Provides a mechanism for the content creator to look at the images and other files in the contents archive.

## 9. Screenshots – functions for creating static images. See the separate section below for a complete discussion of the screenshot function.

- a. The ability to create screenshots may be disabled or restricted in RBIN
- b. User hits single keystroke or chooses 'screenshot' from content menu
- c. The screenshot captures all visuals (the modeled environment plus all text/imagery on the viewport at the time the screenshot is requested)
- d. Maybe add in a box to temporarily change the LOD scale? (The software would display the current LOD scale setting, if changed, it would be restored after the screenshot.)

SCREENSHOT

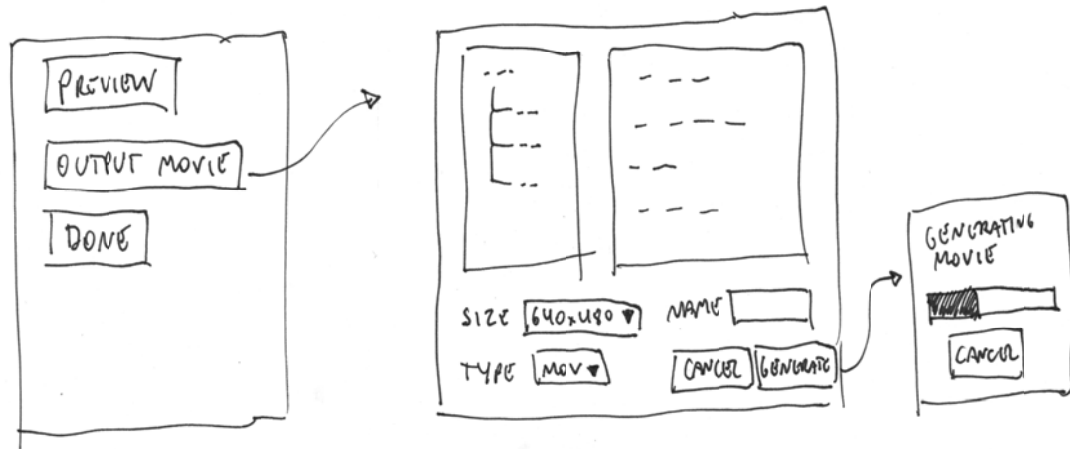


## 10. Movies – functions for creating movies. See the separate section below for a complete discussion of the movie-making function

- a. The ability to create movies may be disabled or restricted in RBIN
- b. The presumption is that the user has loaded the model from which they want to make the video.
- c. User chooses 'movie' from content menu
- d. The software allows the user to generate a movie through the Movie Dialog Box ... can choose to export at fixed settings (e.g., a low-resolution video for the web) or to set advanced settings (large resolution for other purposed.)
  - i. Maybe add in a box to temporarily change the LOD scale? (The software would display the current LOD scale setting, if changed, it would be restored after the movie.)

- e. The user should be able to import and make a movie from Narrative files.

MOVIE



11. **Close** – the software closes the currently loaded model(s) and remains ready to import another. Is this necessary? Eduardo thinks not that the user should just say to open another file (ala notepad) and then the system will close the current model and load the next one.
12. **Exit** – the software closes the currently loaded model(s), shuts down all active processes, and terminates.

## Settings

Default settings (FOV, FOVV, clip, eye height, etc.) are used when a model is loaded from raw files. These can be manually adjusted by the content creator or updated by importing a settings/restrictions file created in a past flight session. Settings/restrictions in place at the time a binary file is exported are saved within that binary file. The general user can only adjust these settings if not restricted by the content creator when the binary file is exported. When an RBIN is loaded, the locked settings will be grayed out so the user has a visual cue that they do not have permission to change these settings.

## Global settings

These are settings that have impact on the user's experience within the modeled environment.

1. Start position (xyz and hpr)
2. Screen size/ratio (set ratio 4x3, 16x9, full screen toggle on/off, possibility of restricting screen dimension by content creator?, typing in pixel dimensions)
3. Full screen
4. FOV (Field of View)
5. Eye height
6. Collision detect (toggle on/off)
7. Rendering style: Wireframe, Scribed, Textured (radio buttons with Textured as default)
8. Map (toggle on/off) and settings related to the map view (Clip, Size of Map, etc.)
  - a. The user appears as a dot/arrow
  - b. There should be keys to zoom in/out of this overhead view
  - c. Possibly also shows icons for location of multimedia content

- d. Possibly also shows dots for location of nodes (while in a Narrative only)
- 9. Near clip
- 10. Far clip
- 11. LOD scale (Level of Detail – presumes the model includes LODs)
- 12. Toggle for narrative loop (will cycle through list of loaded narratives with time-out to allow for possible user interaction)

#### **Advanced Settings** (accessed from arrow at bottom of Global Settings)

- 1. FOVV (Field of View Vertical – better is there is only one setting and it's always relative)
- 2. Aerial perspective (fog on/off, possible slider based on factor of bounding region for near and far fog, color?)
- 3. Ground color
- 4. Sky color
- 5. Bird's eye height/angle? (this exists in vrNav ... do we care?)
- 6. Astronomical Objects? (this exists in vrNav ... do we care?)
- 7. Branding overlay (toggle on/off)
- 8. Time slider (on/off; shows as an icon on screen view; node name has ID for time step)
- 9. Settle factor (?)
- 10. Acceleration factor (?)
- 11. Maximum speed
- 12. Time out for Pause button
- 13. Time out to Resume Camera Orientation during Narrative playback
- 14. Model ID string (user must input for a double check with narrative and embed files)

#### **Lighting**

This tab includes all settings for how the software deals with options for sun position, etc. Lighting (Environment scale (brightness), time of day, date (day, month, year); advanced lighting controls would allow user to toggle on/off; and offset model to proper lat, long, and rotation; and fine tune sun position previously set by TOD and date)

- 1. Ambient Lighting: All surfaces are illuminated from all directions
- 2. Fixed sun position: the surfaces are illuminated with ...
- 3. No Lighting: Turns off interface lighting; only illumination comes from "lights" in model
- 4. Time of Day: Simulates time-of-day using N/S/E/W directions and angle of sunlight inside model dome
  - a. Dawn
  - b. Morning
  - c. Noon
  - d. Afternoon
  - e. Dusk
- 5. Custom Lighting: Enables a GUI with (1) Light Intensity and (2) Origin Coordinates

## Content Creator

The function to create a single screen element that will overlay on the viewport when the binary file is distributed.

1. Branding
  - a. Launches the overlay editor
  - b. Branding elements saved in the BIN/RBIN files (will require a separate overlay save from that used for narratives)
  - c. The branding elements are included in the export of a settings/restrictions/branding file, text elements and any static imagery.
  - d. Can be turned on/off through the settings
  - e. Can be forced on in an RBIN through the restrictions
2. Categories for embedded resources
  - a. Includes a place to type the Titles for the categories to be used as filters for the Embedded Resources
    - i. 'Uncategorized' is the only thing on the list at the onset
    - ii. Add button creates a new category above that and provides a place to add the Title
    - iii. Delete button deletes a category
      - (1) If any Resources are assigned to that category, warning pops up that says 'Resources in this category will be re-assigned as Uncategorized. OK or Cancel'
    - iv. Up and down arrows allow creator to re-order the categories (the way they appear in this list is the way that they'll be sorted in the Embedded Resources bar)
  - b. The logic of the embedded resources section of the code operates on the category numbers, but displays the labels
3. Default browser
  - a. Is there a compelling reason not to use the system default browser?

## Display

This area sets the default fonts, colors, and sizes for the fonts used in the Narratives and Embedded Resources. In Phase 1, we're allowing the user to pick from a short list of fonts that will be embedded in the software (maybe ... this may change to a short list of common system or web-friendly fonts). For simplicities sake, we're allowing the user to set four font styles.

1. Default Font: User picks from short list (default is ?)
2. X-Large: Ratio of view port to use for x-large font (default is .08)
3. Large: Ratio of view port to use for large font (default is .0667)
4. Medium: Ratio of view port to use for medium font (default is .0533)
5. Small: Ratio of view port to use for small font (default is .04)
6. X-Small: Ratio of view port to use for x-small font (default is .0267)
7. Heading 1: Settings for font (from short list; default to default as set above), size (default is large), style (regular, italic, bold, bold italic; default to bold), and color (from short palette; default to white).
8. Heading 2: Settings for font (from short list; default to default as set above), size (default to medium), style (default to regular), and color (default to white).

9. Text: Settings for font (from short list; default to default as set above), size (default to small), style (default to regular), and color (default to white).
10. Caption: Settings for font (from short list; default to default as set above), size (default to x-small), style (default to italic), and color (default to white).
11. Leading for all fonts are calculated automatically and can't be changed in Phase 1.
12. Transparency on/off
13. Transparency color: User picks the default color to be used for the transparent box that lives behind all text (default is black)
14. Transparency opacity: User picks the screen percentage to control the level of transparency used for the text boxes (default is 50%)
15. Line width:
16. Line color
17. Arrowhead on beginning
18. Arrowhead on end
19. Shape color:
20. Shape border width:
21. Drop shadow on/off
22. Drop shadow color (default is black or white depending on the text color being used)

### Binary restrictions

A menu that sets the restriction flags that will be in play when a restricted binary file is written. Allows the content creator to limit what subsequent users can do with the binary file. The default is for the software to be completely open. When an RBIN is loaded, the Binary Restrictions tab is inaccessible to the general user (maybe not there at all?). Determine which of the following restrictions must be its own toggle and which can be lumped into a single category.

### Viewport Restrictions

1. Window size – user can specify the pixel size (for performance, for control of screenshots).
2. Aspect ratio – user can specify the aspect ratio so that the narratives always work properly.
3. Restrict changes to general settings (FOV, LOD scale, etc.)

### Branding Restriction

4. Force branding overlay ON

### Navigation Restrictions

5. Restrict user's interaction to the loaded Narratives. While restricted, the user:
  - a. Can pick and choose amongst available Narratives
  - b. Can hit PLAY/PAUSE and jump to Nodes on the available Narratives
  - c. Can "gimble" and accelerate in either direction of a Narrative's path
6. Force collision ON (If this is desirous, the content creator should build an invisible bounding box in their environment so the user can't fly into never-never land ... maybe in phase 2 include a way to handle it in our software or include Scott's wormholes idea.)

### Output Restrictions

7. Restrict ability to create/edit/delete Narratives
  - a. Individual Narratives can also be locked at the Narrative level (which means it can't be edited even if the user has those privileges)
8. Restrict ability to create Embedded Resource files
  - a. This action also restricts the user's ability to edit Embedded Resource files
  - b. Individual Embedded Resource files can also be locked at the Embedded Resource level (which means it can't be edited even if the user has those privileges)
9. Restrict export/save as (users cannot create any subsequent BIN or RBIN files from the currently loaded file)
10. Video output – content creator can restrict/allow video output
11. Video output resolution – content creator can specify the maximum pixel size for video output
12. Image output – content creator can restrict/allow image output
13. Image output resolution – content creator can specify the maximum pixel size for image output

## Narratives

This is all functions related to narratives that may be associated with the model. Narratives are linear arguments that follow a path through the world and are augmented at specific points along the way using PowerPoint-like slides. If the narrative bar is hidden when the menu option is chosen, only the Show and Create commands are active (unless creating is restricted, in which case it would only be Show). (All others require a narrative be selected to initiate their respective function.) See the separate section below for a complete discussion of narrative functionality.

1. **Show narratives** – this is the equivalent of the maximize arrow on the narrative bar.
2. **Play narrative** – this is the equivalent of double-clicking on a narrative from the narrative bar. It launches the narrative player.
3. **Create a narrative** – Launches the function to create a new narrative. Is the equivalent of clicking the + icon in the list of available narratives (if not restricted).
  - a. At the onset, the user is asked to name a new file.
  - b. The narrative saves throughout its creation.
  - c. Narratives can also be exported so that they are packaged for distribution.
4. **Edit narrative** – editing an existing linear argument. This is the equivalent of clicking the Edit button on the Narratives bar (if not restricted). Launches the narrative editor.
5. **Delete narrative** – removes a narrative from the world (unlinks from the current raw or BIN file and deletes the Narrative file from the folder structure. This action may be restricted in RBIN).
  - a. An error message should display (Are you sure?)
  - b. The images associated with the Narrative will be left in the folder structure in case they are used by other Narrative or Embed files.

## Resources

This is all functions related to the resources that can be embedded spatially within the world. See the separate section below for complete discussion of the embedded resource functionality.

1. **Show embeds** – this is the equivalent of the maximize arrow on the embed bar

2. **Create embedded resources** – creating a file for a single resource, embedded spatially within the world. Is the equivalent of clicking the + icon in the list of resources (if not restricted).
  - a. At the onset, the user is asked to make a file name for the new embedded resource file.
  - b. The embedded resource file saves throughout its creation.
  - c. Embedded resource files can also be exported so that they are packaged for distribution.
  - d. Requires navigating through the world and picking the xyz hpr points.
3. **Phase 2 Resource manager ... a mechanism so that general user can't mess up the contents folder created by our hard-at-work content creators ... also the opportunity to have public/private resource folders so that things can be given permissions for access.**

## Search

Search allows the user to type in a keyword that looks through the text elements of the embedded resources and text elements of the Narratives and displays results in a vertical bar along the left side of the viewport.

1. The Search feature is accessed through a small text field in the top-right corner (the standard location for search bars on websites and web browsers)
  - a. The user types in a word or phrase
  - b. Hitting return or go button (whatever is typical) launches the search through the text elements of the Embed and Narrative files
  - c. Results are displayed in a vertical bar along the left side of the viewport
    - i. Background for the bar is semi-transparent?
  - d. Single click on the items to see its text description
  - e. Double click to go to that position in the modeled environment
  - f. If the selected item is a Narrative Node, the Narrative Player displays that Narrative with the Node in question highlighted and the Player in Pause mode. (The user can click play to watch the Narrative from that point. (Is this necessary?))
  - g. The display results bar must be closed by the user.
2. Example: User types 'Brazil' ... a vertical sidebar shows all individual Narrative Nodes with text that includes the word Brazil, and the embedded resources with text that includes the word Brazil.
  - a. Clicking on the narrative launches that narrative with the user at that node ... takes the camera there and updates the narrative bar
  - b. Clicking on the embedded resource takes the user to the spatial position for that resource, updates the embedded bar, and launches that resource in a browser window.
3. We briefly considered adding the ability for the user to 'Browse' though the world as a way to navigate the modeled environment. This idea has been abandoned because we are accommodating a browse-like functionality within the Narrative mechanism. A content creator would create a Points of Interest narrative and use the individual nodes as jump points).

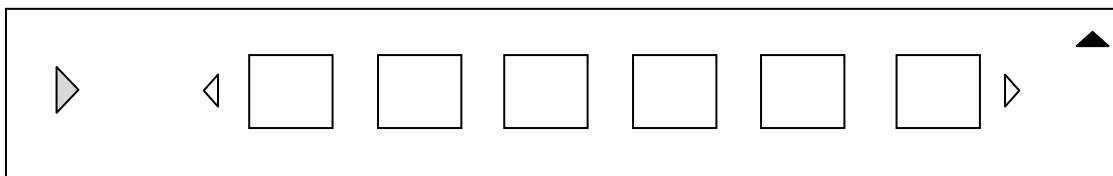
## Narratives – Playing, creating, and editing narratives within the world

The narratives are envisioned as a mechanism to play, create, edit, and play arguments specific to the world written by a single user or user group. These can be stored, shared, and edited. They are currently envisioned as a file independent of the modeled environment.

1. Individual elements that make up a narrative are called the narrative nodes. Any material such as images or text overlaid on the viewport as part of a narrative node are called node content.
2. The following content can be added to a narrative: images, multiple text elements, graphics (shapes and lines), and audio clips ... maybe more in Phase 2
3. The default save for the narratives is to include each narrative, its nodes, and links to all associated node content will be saved in a single external file. This means that the associated node could be scattered across a user's computer.
4. Exporting a narrative packages up the narrative file and its associated content into an easy-to-manage package for sharing
5. Narratives can be locked by the creator. This means that it can only be used for playback by subsequent end users. (This requires the software to check the locked flag to decide whether or not to gray out the edit button.)
6. In Phase 2, maybe we'll include an option to choose between storing all content in the narrative file or only the file references (this will allow a smaller narrative file in the case that large video or audio files have been used). (Write the code originally to save only the file reference for all external content; if time allows, write the option to include all associated node content.)
7. The ability to create or edit narratives may be restricted by the content creator; therefore that function must be able to be disabled in the software.
8. The visuals related to the Narratives are currently spec'd to occupy the top sixth of the viewport.
9. The background is a programmable color (default to black) rendered at a programmable transparency setting (default to 70%)

### Narratives List (Player functions only)

The Narrative List is the bar that is the access point from which the user shops among the various narratives available for playback.



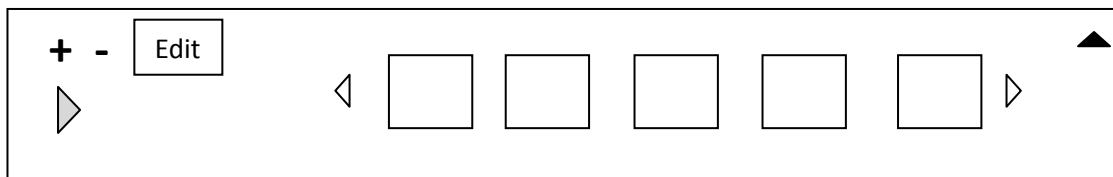
1. This bar sits along the top edge of the viewport and is slightly transparent.
2. An Up-triangle allows the user to hide/show the Narratives List
3. A Play icon is on the left side of the bar.
4. The bar contains thumbnails of the first frames of all available narratives



- a. Single click selects the narrative
  - i. The user ‘jumps’ to the first frame of the narrative and shows the text description overlaid on the viewport
  - ii. Selecting a different narrative will jump the user to that first frame
  - iii. If the user shifts to free flight at this point, she remains at this location.
  - iv. With a narrative selected, pressing the play icon will show the Narrative Player bar and start playing the Narrative.
- b. Double click on the narrative opens the Narrative Node bar
  - i. From the Narrative Node bar, the user can hit play or look at the thumbnails of the Narrative to get a sense of the content.
5. If there are multiple narratives available, scroll arrows appear at the sides so that the user can scroll the list of narratives to the left and right
6. While the list of narratives is visible, a mouse click in the main viewport window puts the user into free flight in the environment (unless the viewer is restricted to one of the available narratives).
7. Hover over on a narrative shows a text description for that narrative

### List of Narratives Bar (Player and edit functions)

If Narrative creation and editing is enabled, the Narrative bar includes additional functions: a plus icon (+) to initiate the creation of a new narrative, a (-) to delete a Narrative from the list, a Quick Record button to record a path as the basis for a Narrative, and an Edit button. If creation and editing is restricted by the content creator, the icons for these functions do not appear on the display.

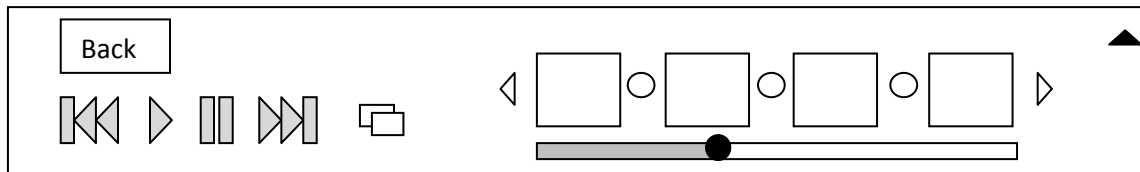


1. +: Adds new narrative
  - a. Shows a “New File” dialog box, which allows the user to type a name for the new narrative.
    - i. The file will save in the current model directory if not otherwise specified.
    - ii. The user is taken to the Narrative Nodes bar to begin building their Narrative.
2. -: Delete narrative
  - a. Grayed out if no narrative selected
  - b. A narrative must be selected
  - c. Clicking - removes the selected narrative from the list for the current flight session
3. Edit narrative
  - a. Grayed out if no narrative selected

- i. Also grayed out if selected narrative is locked
- b. A narrative must be selected, the user can hit the Edit key (if not restricted)
- c. The user hits the Edit key and the Narrative List bar smoothly zooms into the Narrative Player/Editor bar which comes from inside the chosen Narrative.

## Narrative Player (Player functions only)

Playing a narrative essentially places the user on a predefined route with pre-determined speed, direction, and media content (i.e. to be used by educators or in museum installations).



1. Playback is launched when the user selects a Narrative from the Narratives List and presses play.
  - a. Double clicking a Narrative from the Narratives List opens the Narrative player, but does not launch playback.
2. The List of Narratives is replaced with a filmstrip-like series menu appears across the top featuring screenshots of the nodes and its title.
3. Icons on Narrative Player (play-related only)
  - a. Up-triangle: hide/show bar
  - b. Back button: return to the List of Narratives
  - c. Rewind: goes to the beginning of the Narrative. (If there is an audio or video clip being played on the current node, it is aborted.)
  - d. Play: Start playing from the current position
  - e. Pause: Stop playing
  - f. An icon allows user to select full screen play (if not restricted)
  - g. Go to end: goes to the end of the Narrative. (If there is an audio or video clip being played on the current node, it is aborted.)
  - h. Elapsed time slider shows the relative location of the user/Node within the linear narrative.
4. As playback commences, the Nodes and transitions are highlighted to indicate position along the Narrative
  - a. Nodes can be selected and used as navigation tools; the transitions are disabled
5. Along the bottom of the bar is a time slider that charts the user's progress through the narrative.
  - a. At the beginning, the slider is all the way to the left ... as the narrative plays, the slider moves to the right.
  - b. This requires that the software calculates and constantly updates the total duration of each narrative according to the time settings of that narrative.

- i. Mouse over on the left side of the slider shows time elapsed
  - ii. Mouse over on the right side of the slider shows time remaining
- c. If a user takes control of the speed of the narrative, the time slider continues to mark their relative position along the path, but there's no mouse over for time elapsed or remaining.
- d. If the user returns control of the narrative, the time elapsed/remaining reappears and is based purely on the calculations of the narrative timing as if the user had never taken control.
- 6. The Narratives Node bar minimizes x seconds after playing commences so that the bar will not conflict with overlay images
  - a. Moving the mouse over the Narrative Node bar space brings it back – OR – we have a small icon that the user needs to click to bring it back.
  - b. If the bar is open and there is conflict, should the software shift the overlay content down to accommodate the bar?

### User-Interaction during Narrative playback

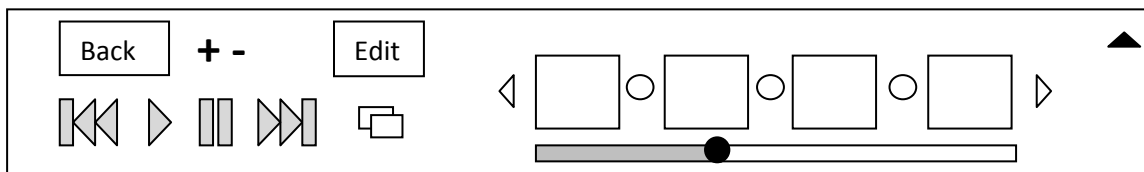
- 1. The narrative can play from start to finish without interaction from the user.
- 2. The user, however, can pause playback.
  - a. If the user is not restricted to narratives, pausing will disengage them from the narrative and allow them to move through the environment in free flight
    - i. Pressing play at any point will reattach them to the narrative where they left off.
    - ii. This may require a smooth move to get them back to the xyz hpr of the narrative node they left
    - iii. The narrative resumes at the beginning of the node that was interrupted
  - b. If the user is restricted to narratives, pausing simply interrupts the narrative.
    - i. After x seconds, the software returns to the List of Narratives bar (for use primarily in a kiosk mode after the museum visitor has walked away).
- 3. The first time a Narrative is played, playback always begins at the beginning node of the narrative.
  - a. Narratives must be tracked as they are played so that users can resume play (which means that subsequent playback picks up where the user left off).
    - i. If Narrative playback starts in the middle and the user wants to start again from the beginning, they can double click on the first node to reset to the beginning.
- 4. Hover over on a node shows the text description
- 5. Single click on a node: selects that node
  - a. If the narrative is playing, nothing happens
  - b. If the narrative is not playing, the user can hit play to go to that node location and play from that location
- 6. Double click on a node:
  - a. If the narrative is playing, the camera goes to that node location and continues playing from that node
  - b. if the narrative is not playing, the camera goes to that node location and the user has to hit play to begin playing from that node
- 7. At the conclusion of a Narrative, playback simply stops at the end Node.

- a. Time out allows loop to next narrative if no interaction is detected in x seconds following the conclusion of N1 (this is to facilitate use in a museum kiosk and not likely to be used by general users).
- 8. During playback, the user can also control the speed and heading along the narrative path.
  - a. Restricted by a maximum speed setting
  - b. **Ongoing question: if a user takes control of the mouse during a narrative, what happens when they pass a narrative node? They could be looking anywhere ... do we force them to look at the node? (Jennie and Eduardo think we should force them, Meghana does not.) Ignore the node? Call their attention to it in some fashion?**

## Narrative Player/Editor

Creating narratives is the process through which an authorized user builds a linear narrative within the modeled environment. Editing a narrative is making modifications to a pre-existing narrative, presuming it has not been locked by its creator. Creating and editing are both done through the Narrative Player/Editor.

1. This function “Facilitates the... construction of lesson plans, arguments, and narratives by either the content contributor (as subject expert) or a student user (a learning exercise)”
2. Feature can be disabled by content creator
3. If allowed, the user would click on the + icon to begin building a narrative. A Narrative Settings dialog box prompts the user for file name, title, text description, set default fonts (?), ...
4. The Narrative Nodes Bar replaces the List of Narratives. This bar includes commands for creating nodes, adding settings, editing splines, etc.



5. In addition to the playback icons, the following are displayed when editing is enabled:
  - a. +: Create new node
    - i. The node captures the current camera position/orientation
    - ii. User should decide where they want the Node to be placed in the linear narrative
      - (1) If no node is selected, new nodes are added to the end of the Narrative
      - (2) If a node is selected, the new node is created AFTER that node (A – A1 – B); the spline between A and B is recalculated (A-A1, A1-B)
      - (3) If a transition is selected when the + sign is clicked, the new node is created BETWEEN the nodes on either side of that transition (A – A1 – B); the spline between A and B is recalculated (A-A1, A1-B)

- iii. If the user is playing the narrative, pauses playback, and chooses to add a Node, the logic describes above is followed.
      - (1) If playback is paused on a node, the new node is created AFTER that node (A – A1 – B); the spline between A and B is recalculated (A-A1, A1-B)
      - (2) If playback is paused at any point along the spline in a transition when the + sign is clicked, the new node is created BETWEEN the nodes on either side of that transition (A – A1 – B); the spline between A and B is recalculated based on the position of the new node (A-A1, A1-B)
    - iv. The Node Overlay Editor appears at the viewport as in Figure 1.
  - b. - : Delete node (would not show if narrative is locked; if not locked, would be grayed out during playback)
    - i. A node must be selected for the delete to work
    - ii. The spline between the nodes remaining on either side is recalculated (A – B B – C is recalculated as A – C)
    - iii. The transition AFTER the deleted node is also deleted (and the user may want to modify the remaining transition; if there's some weird difference, we may want to pop-up a message to remind the user to check the transition settings.)
    - iv. If the user has paused playback on a node and hits delete, that Node is deleted and the spline between the remaining nodes is recalculated
    - v. A transition, by itself, cannot be deleted.
  - c. The Free Flight Recorder records a path and adds a node at the end of that path
    - i. This button is enabled only when there are no nodes defined in the Narrative or the user is positioned on a selected Node.
    - ii. A dialog box tells the user to find their starting position and press 'r' to start and stop recording.
    - iii. The user positions themselves, presses 'r' to start the recording session, flies around, presses 'r' to stop recording.
      - (1) If there are no defined nodes in the Narrative, at the conclusion of the Free Flight record two nodes are populated: the beginning and end points of the path.
      - (2) If the user is positioned on a selected Node, the Free Flight Recorder records a path and adds a node at the end of that path. (A –**A1** where there is no B)
        - (a) If there is a node after the newly created node, a new spline is calculated between them. (A-**A1**-B)
- 6. Edit – grayed out unless a Node or Transition is selected.
  - a. If a Node is selected, hitting Edit will launch the Overlay Editor
  - b. If a Transition is selected, hitting Edit will launch the Spline Editor
- 7. Camera Record – Eduardo wants to add this so that a user can record a path, then play back and record the desired camera orientation. – A la USIM. Can be scheduled for Phase 1.5.

## Creating a Narrative

The narrative function allows either the content creator or end-user to record precise motion sequences. There are two ways to create a narrative within the modeled environment: Node-by-Node and from a Free Flight path. In either process, the user defines the key visuals that she wants to stress in the narrative, and uses the Overlay Editor to augment these key visuals with text or imagery. During playback, the software reads the Narrative file to recreate the content creator's desired movements through the modeled environment. This movement can be edited in world with the Spline Editor. Settings can be imposed on either the node or the transition between nodes.

There are two ways to build a narrative: Node-by-Node and from a free flight path.

1. In **Node-by-Node** creation, the content creator would move through the modeled environment, adding nodes at each key point within the model.
  - a. If the content creator wants to display sequential images from a single xyz location or include pans within the narrative, they will have to make multiple nodes at that point. (Phase 2 might involve some actions in the Overlay Editor.)
  - b. Transitions between nodes are set at defaults that can be changed
  - c. Ideal operation would have the content creator identifying all nodes in their argument/tour, adjusting the node and transition settings, then adding any desired text elements or supplementary content for display on the viewport.
  - d. Additional nodes can be added as desired and the software will recalculate the splines before and after the new node as necessary to integrate it into the Narrative
2. Using the **Free Flight Recorder**, the content creator would tell the software to record their movement through the modeled environment and use the resulting file as the basis for a Narrative.
  - a. The free flight path would include two nodes: one at the beginning and one at the end.
  - b. The content creator would 'play' the Narrative and mark
  - c. The Free Flight Recorder is able to be used to create a string of Nodes. In this variant, the content creator adds a Node, then records a Free Flight path to her next desired stopping point, and adds a Node, and so on.

## User Interaction during Narrative Creation and Editing

The user interactions are the same

1. Hover over on a node shows the text description
2. Single click on a node: selects that node
  - a. If the narrative is playing, nothing happens
  - b. If the narrative is not playing, the user can hit play to go to that node location and play from that location
3. Double click on a node:
  - a. If the narrative is playing, the camera goes to that node location and continues playing from that node

- b. if the narrative is not playing, the camera goes to that node location and the user has to hit play to begin playing from that node
4. Right click on a node: provides access to the Node Overlay Editor if the Narrative is not locked
  - a. The Node Settings (**Error! Reference source not found.**) always show with the Node Overlay Editor
5. Double click on the Transition: Shows the Transition Settings if the node is not locked (**Error! Reference source not found.**)
6. Drag and dropping Nodes allows the user to restructure their linear Narrative
  - a. Splines are recalculated during this process
  - b. A confirmation box is displayed if an edited spline is about to be destroyed (this will require a flag so that the software recognizes splines that have been edited and/or created with the Free Flight Recorder).
7. Obviously, if the ability to create or edit narratives has been disabled, this would also disable the ability to edit nodes; therefore we have to make sure that there's no access for unauthorized end-users in the software.

## Narrative Settings Dialog Boxes

### Narrative Settings

If not restricted, the user would click on the + icon on the List of Narratives to begin building a narrative. A Narrative Settings dialog box prompts the user for file name, title, text description, set default fonts (?), default speed, acceleration, what else ?...

The settings can also be accessed by right clicking on the Narrative on the List of Narratives bar.

TO BE DEFINED IN THE WORKING DRAFT; NEEDS GRAPHIC

### Node Settings

The Node Settings box includes the following: UPDATE GRAPHIC

- a. Title – a text box to identify the content (this displays below the node in the Narrative Node bar)
  - i. Is a searchable field
- b. Short description – a text box to explain the content (this display shows on mouse hover over when the user is looking at nodes in the Narrative Node bar)
  - i. Is a searchable field
- c. Fade In – number of seconds required to fade in the Overlay (default is 1 second)
- d. Fade Out – number of seconds required to fade out the Overlay (default is 1 second)

The diagram shows a rectangular dialog box with a thin border. Inside, the following elements are arranged from top to bottom:
 

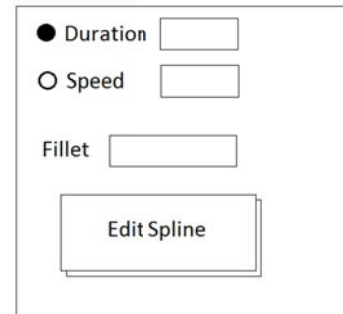
- The word "Fade" is followed by "In" with a checked checkbox and "Out" with an unchecked checkbox.
- The text "Go On" is followed by a selected radio button labeled "Duration" and an empty text input box, and below that, an unselected radio button labeled "Keypress".
- A single-line text input box labeled "Title".
- A multi-line text input box labeled "Description".

- e. Toggle between setting the Pause Duration, hold for Keypress, or Continuous Motion.
  - i. Pause Duration equals the number of seconds to linger on the node,
  - ii. Hold for Keypress means that a keypress by the user is required before advancing to the next node
  - iii. Continuous Motion means that the Pause Duration is set at 0, and the user can specify the number of sections that the Overlay is visible in the Narrative. (Half of the time specified applied to the front of the node and half to the back.)
- f. Check box to use the resource as an Embedded Resource.
  - i. The Node Orientation is used
  - ii. User must provide Radius
  - iii. User might need to provide Angle

### Transition Settings

The Transition Settings box includes the following:

1. Toggle between Duration and Speed.
  - a. Duration is the number of seconds the software will take to move between the Nodes on either side of the transition
  - b. Speed is the number of units of length to be traveled in one second. This will vary depending on the unit of measure used in the model (e.g., to travel at a walking rate of three miles an hour, the user would enter 4.4 units per second for a model built in feet, but 1.34 for a model built in meters).
2. Edit spline launches the Spline Editor.



The image shows a 'Transition Settings' dialog box. It contains three radio buttons: 'Duration' (selected), 'Speed', and 'Fillet'. Each radio button is followed by a text input field. Below these is a button labeled 'Edit Spline'.

### Overlay Editor

The Overlay Editor is a 2D editor used to create content that will display on the viewport and that describes and/or augments in some fashion a specific xyz hpr keyframe within the modeled environment. Typically, the content will point to or reference something in the 3D environment. In phase 2, the ability to add an action to a node may be added (e.g., the ability to trigger a switch automatically. How this might happen has yet to be determined.)

1. Primary functions of the Overlay Editor include the ability to create basic shapes (circles and squares), lines, free-form drawing (?), and text blocks, and the ability to import 2D image and audio files. (Phase 2 may include the ability to import video files.)
2. The Overlay Editor can also be used in conjunction with the directional settings for embedded resources with the potential to automatically pop-up during a user's exploration of the world.
3. Dotted lines divide the viewport into three zones – the safe zone is the center two-thirds of the horizontal space (the top and bottom sixth of the viewport may conflict with the Embedded Resources and Narrative bars).
4. We could get really wild with this, but for Phase 1 we will only accommodate:
  - a. Line ... choose line tool ... left click to set one corner and drag to second point.



- i. The line defaults to color, width, and arrowhead default settings
  - ii. The line shape, properties, and position can be edited
- b. Shapes ... (Circle/Square/Rounded Corner Rectangle as three options stacked together (ala PhotoShop) or shown separately) ... user chooses one ... left click to set one corner and drag to second point.
  - i. The shape defaults to color and width defaults
  - ii. The shape's shape, properties, and position can be edited
  - iii. There is no fill (thinking that it will be used mostly to highlight objects in the keyframe)
- c. Free-form drawing ... left click to set first line and hold down to scribble
  - i. The line defaults to color and width defaults set for line.
  - ii. The line as a unit can be edited (width, color, and position)
- d. Text box ... user chooses text tool for desired style ... left click to set one corner of text box
  - i. Click cursor inside box to begin typing
  - ii. The textbox holds to the width drawn and one line of text high
  - iii. User types text
  - iv. Text box is automatically filled with transparent fill (how big a border?)
  - v. Type is default for chosen style, but style and color can be modified (right click to show change options ... maybe a box to change that style in default?)
  - vi. Text box can be dragged and its shape manipulated
  - vii. Hyphenation is off
  - viii. Left click to pick the box (or shape) to resize or move; left click and drag to highlight a text; right click to show change.
  - ix. First do this in 2D, then translate to 3D space so that the type element can spin for Phase 1.5
- e. Importing image and audio files ... choose icon from tool palette to import file ... browse for and choose image or audio file. UPDATE GRAPHIC TO REFLECT THESE CHANGES
  - i. Images
    - (1) Image drops into the overlay editor
    - (2) Image can be scaled.
    - (3) Image can be moved and repositioned as desired
    - (4) Image will have a subtle shadow/border TBD
  - ii. Audio
    - (1) Settings are auto-play or requires a keypress to play
    - (2) Audio will play until its conclusion
    - (3) If a second piece of audio content is encountered, the first clip is stopped and the second launched
    - (4) Ongoing question: How to handle audio if the user starts jumping around and going fast? Meghana suggests adding "radius in which audio will play" setting.**
- 5. Check box to allow overlay to be used as an embedded resource as long as the Narrative is loaded
- 6. The Overlay Editor includes buttons for CANCEL (revert to last version?) and DONE (save and return to Narrative Editor)

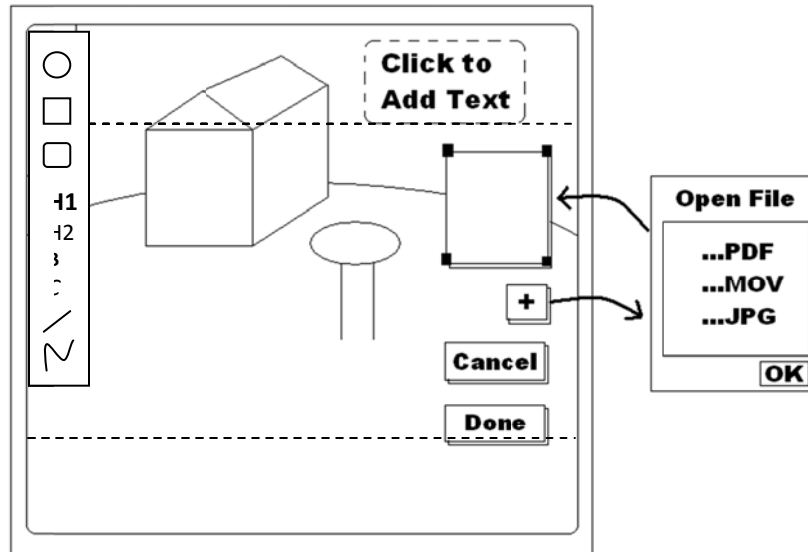


Figure 1: Embedded Editor

## The Spline Editor

This function generates a spline path from line segments determined by the position (xyz hpr) of the origin and destination nodes

1. This spline is calculated and saved as soon as the origin and destination node positions are known.
2. the **calculated spline** then appears within the model as a colored line
3. If changes are desired, content creator can click to edit the calculated spline.
  - a. In this function, the spline can be grabbed and modified, and it is possible to add in intermediate points.
  - b. Depending on how the viewing frustum works while on the spline, we may need to require the content creator adds in extra nodes to really control the camera?
4. This **modified spline** is saved in place of the calculated spline.
5. Resulting narrative file will be able to reproduce desired motion on playback (using calculated or modified spline, the timer and/or duration of movement settings, etc.)
6. When nodes are deleted, the splines before and after the deleted nodes gets dropped, and the software recalculates the spline between the remaining nodes.
  - a. Every delete operation will have an undo option ... the number of undo levels is currently conceived as infinite.
7. In playback, when you click on a narrative node and get sucked to that location, the software calculates the spline between the current location and the destination node and executes that move.
  - a. This may require a slight arc in z to avoid being sucked through a lot of buildings. The issue is making this arc shallow enough that it only just skims the rooftops (it shouldn't get to helicopter view).

## Points of Interest (aka Jump Points/Browse)

Rather than a separate mechanism for creating and displaying jump points, the content creator will be encouraged to build a narrative where the individual nodes illustrate the key elements of the model. Each point of interest (POI) should include a descriptive title and a short description in the node settings box that can provide the user with enough information to make a decision about choosing that POI.

1. Controls of the POI narrative are left to the content creator:
  - a. The order of the nodes is dependent on how the narrative has been built (so the content creator can make them alphabetical, spatial, in order of importance ...)
  - b. Transition settings could be set so that the POIs could be played like a narrative
  - c. Transition settings could also be set as Duration=0 (this, coupled with the duration linger on the Node itself, would result in a slide show)
  - d. The user may also click on the individual narrative nodes to jump directly to that node.

## Branding option

The Branding function allows the content creator to build a single Overlay that will always stay on top of the viewport. The intent is to ensure that their intellectual property rights are acknowledged and to assuage any concerns that they have about sharing their academic content.

In Phase 1, all code will be open source, which might not address all content creators' IP concerns. (Hacking the branding and user restrictions would require some computer expertise; for example, the code could be recompiled without those modules.) Language in the license could reinforce the IP rights of the content creators. This would be likely that Phase 1.5 or 2 of the code will split out the module that deals with Branding and user restrictions and add encryption.

1. Branding is set through the Content Creator settings
  - a. Launches the Overlay Editor
    - i. The content creator is not allowed to place Branding elements in the top sixth of the screen and the bottom sixth of the screen (this must be visually indicated in the Overlay Editor somehow)
    - ii. The remaining screen space is divided into three horizontal zones
      - (1) Elements with their centers in the bottom zone will anchor to the edge of the bottom bar so that they will auto-adjust down when the Embedded Resources bar collapses.
      - (2) Elements with their centers in the center zone will be immobile
      - (3) Elements with their centers in the top zone will anchor to the edge of the top bar so that they will auto-adjust up when the Narrative bars collapse.
  - b. Branding Overlay can include text, shapes, lines, and images (not audio)
2. The display of Branding Overlay is controlled in the Global and Restrictions settings
  - a. Can be turned on/off through the settings
  - b. Can be forced on in an RBIN through the restrictions
3. Branding elements are saved in the BIN/RBIN files (will require a separate overlay save from that used for narratives)

4. The branding elements are included in the export of a settings/restrictions/branding file, text elements and any static imagery.

## Embedded Resources

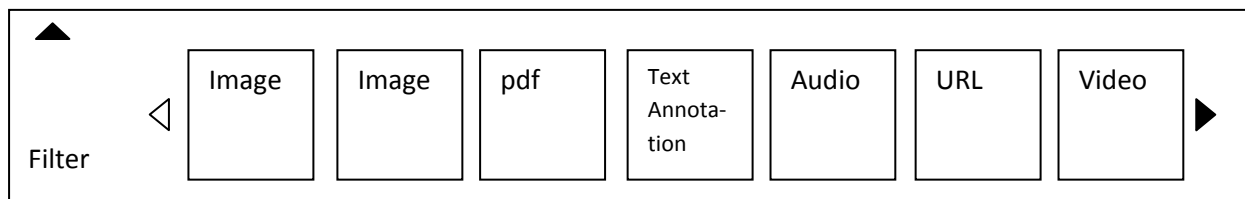
The point of the embedded resources is to enhance the user's free exploration of the modeled environment by providing additional information at spatially defined positions. Involves both the ability to open and view them while in the modeled environment and, if allowed by the content creator, the ability to add spatial resources.

1. Embedded Resources are displayed as 'movie strip' of thumbnails, which updates based on the creator-defined proximity to the content.
2. Things that can be embedded:
  - a. Images (any file type that is typically supported by Firefox and Internet Explorer should be acceptable) .jpg, .gif, .png, .tif, .bmp, .tga, etc.)
  - b. PDFs
  - c. Links to URLs (preferably to persistent URLs or a directory to resolve DOIs to URLs) ... could be to a simple website, an online streaming video, etc.)
  - d. Audio files (.mp3)
  - e. Video files (.avi, .wmv, .mov, .mg4)
  - f. Word documents (would be problematic for Linux and Mac users ... so is this desirable?)
  - g. Text defined by the content creator ... this is critical for annotating models ... the content creator has to have the opportunity to write some amount of text about a spatially defined area . They could write notes to one another, it could be a bibliographic entry, it could be a short discussion about the model, whatever ...
  - h. Nodes built with the Overlay Editor if marked to use as an Embedded Resource.
    - i. Is it desirable to include an option for these to automatically pop up if the user hits the activation area within the model?
3. We should encourage use of URLs or simple text annotations for embedded resources simply because of problems with file size.
4. The Resources are assigned Categories that have been previously defined by the content creator. Each Resource that uses the model must adhere to the same categories for the filters to work properly (when interacting with the model, the user has the option of showing all resources, or any combination of the resources).
5. At the export or save as, the user is given the option to lock the file; the software never lets you save a locked file over a previously existing unlocked file. (This is to prevent files from being permalocked.)
6. The visuals related to the Embedded Resources are currently spec'd to occupy the bottom sixth of the viewport.
7. The background is a programmable color (default to black) rendered at a programmable transparency setting (default to 70%)

## Displaying Embedded Resources

The embedded content bar is located horizontally along the bottom of the window.<sup>1</sup> It can be hidden, but the icon to unhide the bar will be present any time there are resources embedded near the user's position in the model.

1. Icons on the embed bar include an arrow to minimize/maximize the bar, a filter icon, thumbnails for the resources, and scroll arrows to move left and right in the list of resources if there are more than can be displayed on a single screen.
  - a. Minimize/maximize: hides or displays the Embedded Resource bar
  - b. Filter: allows the user to set the filters for the resource display
    - i. Categories: User selects all or any combination of the seven categories allowed by the software
    - ii. Files: User selects all or any combination of the various Embedded Resource files loaded in the model
    - iii. Toggle for Spatial Display or All Resources.
2. The list of Resource to be displayed is based on the user's current location within the modeled environment and the position and target of the Resource (e.g., if the user is within a modeler-specified distance of a resource's position and target, its thumbnail appears in the menu. Similarly, once the user leaves that resource's cone of influence, the thumbnail disappears.)
3. The order that Resources are displayed is determined by the following rule:
  - a. (Local (Category (Title))) (Global (Category (Title)))
  - b. In English, Global Resources should be at the end of the display and Locals at the front. Within those two groups, they should be sorted by Category and then Title
4. What's shown for each resource is the thumbnail and the title below it



## User Interaction with the Embedded Resources

1. Single click on the resource selects the resource
  - a. Shows its text description
2. Double click launches the resource in a separate browser window (the user is responsible for their own damn windows management and the ugliness of their user interface)
  - a. Phase 2 could include built-in viewers for each file type so that the visuals are more unified.

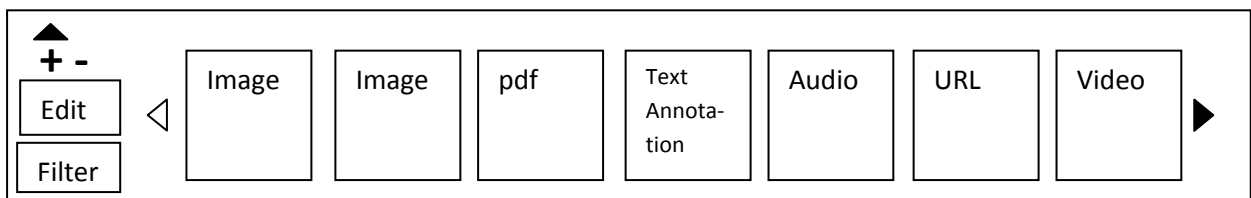
## Displaying Embedded Resources (adding and editing enabled)

Adding embeds involves adding a suite of spatially organized resources to the modeled environment. Editing involves modifying an existing file of embedded objects by taking a pre-existing embed file and

<sup>1</sup> World Wide Telescope

adding additional resources to it, deleting resources, or editing the settings for any of the embedded objects. The users can either create a new embed file or add new embedded objects to an existing file. (Adding new embeds to an existing file would be used to build one large file.)

1. The content creator may have restricted the ability to add and edit Embed files
2. Individual Embed files may have been locked by their creators
3. The display is the same as above, except for the addition of the editing icons.
  - a. +: Create a new Resource and provide the opportunity to create a new Embedded Resource file
  - b. -: Delete Resource removes a Resource from its Embedded Resource file
    - i. Grayed out unless an editable Resource is selected (individual Embed file could be locked)
    - ii. Requires confirmation (Are you sure?)
  - c. Edit – Allows the user to edit the settings of an Embedded Resource
    - i. Grayed out unless an editable Resource is selected (individual Embed file could be locked)
    - ii. Launches the Embedded Resource Settings dialog box (discussed below)
    - iii. If a Transition is selected, hitting Edit will launch the Spline Editor



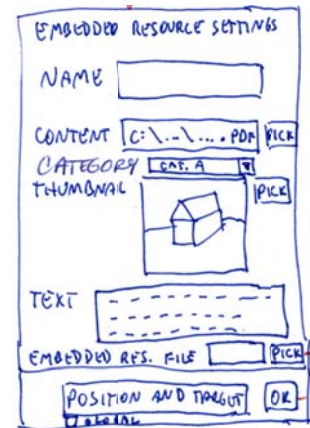
### User Interaction with the Embedded Resources (adding and editing enabled)

1. Single click on the resource selects the resource
  - a. Shows its text description
  - b. If the individual Embed file is not locked, hitting '-' to delete will prompt a confirmation and then delete
  - c. If the individual Embed file is not locked, hitting Edit will launch the Embedded Resource Settings dialog box
2. Double click launches the resource in a separate browser window (the user is responsible for their own damn windows management and the ugliness of their user interface)
  - a. Phase 2 could include built-in viewers for each file type so that the visuals are more unified.
3. If no resources are selected, only the '+' icon is active
  - a. Clicking '+' will launch the 'Pick a File' to define where the new Resource is to be saved
    - i. User picks file or establishes new file location and name
    - ii. Subsequent '+' commands will default to the last used Resource File name, but can be changed by the user

## Embedded Resource Settings dialog box

1. The embedded resource settings box allows the user to add or edit the information related to an Embedded Resource.
2. The user press (+) on the embed banner to add a resource and picks the file in which to save the embed information.
  - a. The dialog box shows the list of editable embed files currently loaded
  - b. Also allows the creation of a new file.
3. The ER Settings dialog box opens. The user must provide the following information:
  - a. Resource Name/Title – the name that will appear under the icon in the resource bar
  - b. Content – the path to the file on the content creator’s computer system OR the URL of website to be used. For files, there is a “pick” button, which will invoke a File/Open dialog and allow the appropriate types (.jpg, .png, .mov, .mp3, .pdf...). For URLs, the content creators can copy-and-paste in the text box.
  - c. Category association – A drop down list of the category titles as set by the content creator and stored in the BIN.
  - d. Thumbnail – generated by the software for image files at the point of selection (and any other file type where the thumbnail can be automatically generated) ; user must browse for their thumbnails if they exist for all other files types
    - i. The thumbnails will be displayed in a rectangular portrait orientation at approximately 128x100 pixels (the size for a 1600x1200 screen)
    - ii. Thumbnails loaded by the content creators should ideally be close to that, but any size image will be accepted and scaled by the software.
    - iii. In the absence of a thumbnail provided by the user, generic thumbnail icons will be used by file type. (Jennie to design)
  - e. Text – The Text field allows the user to associate text of arbitrary length with the resource. Length of text field to be determined.
    - i. In phase 2, multiple text fields might be desirable. (Add resource source as separate field?)
  - f. Auto-display check box – not active if there is content defined other than text
    - i. If checked, the text will auto-display the same as the hover over, but with a bigger font
  - g. Global resource check box – means that the resource should always be displayed in the Embedded Resources bar
  - h. Position and Target – a button at the bottom of the ER Settings dialog box is access point to spatial position and target.
    - i. The content creator has to move within the modeled environment so that the position, range, and target of her Embedded Resource are in view
    - ii. Cursor says ‘click on resource center’
      - (1) The first left click marks the center position of the resource
      - (2) A visual bounding sphere appears with a default radius of 15 units

- (3) Manipulating the center wheel of the mouse enlarges and shrinks the bounding sphere
- iii. Cursor says 'click on resource target'
  - (1) A second left click marks the target of the resource
- iv. User clicks on DONE on Position/Target section of the dialog box
- v. (Maybe have two options: one to set Position and one to set Target)
- 4. The software stores all information about the resource that has been stored in temporary memory: ID#, name, file name/location, URL, coordinates (x, y, z, h, p, r), bounding radius, viewing angle, text field (for caption, annotation, etc.), category, thumbnails, etc.



## Screenshots

This section describes the function for creating static images from within the modeled environment. Accessed from the File menu.

1. The ability to create screenshots May be disabled or restricted in RBIN
2. Screenshots may also be restricted to a max resolution by the content creator
3. If branding is on, the branding elements will be visible in the screenshot
  - a. If branding is not restricted, the user can toggle off the branding elements
  - b. If branding is restricted, all screenshots will include the branding elements
4. If working from 'raw' files, no restrictions are in place.
5. User hits single keystroke or chooses 'screenshot' from content menu
6. Screenshot menu opens with the following: place to browse for folder, place to type file name, place to set a new pixel size for output resolution.
7. Browse for folder defines where the screenshot is to be saved; the path is saved in the flight session for subsequent screenshots (maybe saved in a user setting file?). The path remains in memory until the end of the flight session or until it is replaced by another path name.
8. Two input boxes, one for width and one for height, allow the user to set a new pixel size for the image output.
  - a. The aspect ratio is always the same as the viewport, and the ratio between the two boxes is always frozen. (Should we allow aspect ratios other than the viewport? I think not because then the composition is iffy.)
  - b. User input to update either pixel dimension updates the other ... the user can change these until they are bored to tears.
  - c. Possible temp LOD setting? (would restore after the screenshot is taken)
9. At the bottom are save and cancel buttons
10. Static image files can be saved in the following formats: .jpg, .tif, .png, .tga (are multiple options necessary?)
  - a. The user specifies the file type by choosing from a drop box



- b. The chosen file type is saved in temporary memory and pre-loaded on subsequent screenshots
- 11. If a file with the same path and file name already exists, the software returns an error and allows the user to specify a new file name

## Movies

This section describes the functions for creating movies. Accessed from the File menu.

1. The ability to create movies may be disabled or restricted in RBIN
2. Screenshots may also be restricted to a max resolution by the content creator
3. If branding is on, the branding elements will be visible in the screenshot
4. The presumption is that the user has loaded the model from which they want to make the video.
5. User chooses 'movie' from content menu
6. The software allows two variations for saving video clips:
  - a. The first is to Quick Record (accessed both through the Narrative bar and the Movie dialog box; uses the same mechanism as the Narrative Quick Path Record and the Create Path function).
  - b. The second option is to generate the clip Off-line from a Narrative file (this option presumes that the output will require extensive processing for an amped LOD scale and/or high resolution, but it could just be that the user wants some custom settings or aspect ratio for a low-resolution clip)
    - i. **NOTE: The uSim-style session file has been replaced with the Narrative file. (The user can edit the splines, speeds, and settings via the Narrative, so there is no need for the session file as it currently exists.)**
7. Quick Record – A 'Start Record' button appears, the user finds her desired start location, presses the button (or hits a keystroke?), flies around, stops the recording.
  - a. Output Dialog Box gives options to Preview, Output Movie, or Done.
    - i. If Preview, the software replays the action from temporary memory and returns to the Output Dialog Box
    - ii. If Output Movie, a new dialog box asks for the output file name, for the user to pick from two output sizes (640x480 or 320x240), and for the user to pick from two file types (.mov, .wmv). The software assumes 30 frames per second, generates the frames, and builds the movie file.
8. Off-Line – off-line movie output involves importing a narrative, possibly previewing the output, and setting the output options necessary to save the movie file.
  - a. The software takes as much time as it needs to output the graphic frames and build the clip.
  - b. When the user chooses the Off-line option from the movie menu, the Off-line Movie dialog box is launched.
    - i. The user browses for the Narrative file to use as the basis for the clip
    - ii. Defines the location of the file to be saved
    - iii. Frame rate (default is 30) – PAL vs NTSC?
    - iv. File type from a drop down menu (.avi, .mov, .wmv?)
    - v. Pixel dimensions of output (see image size controls as described for saving static images)
    - vi. Codex to use (?) Eduardo says no, and that we just use H264.

- vii. Possible temp LOD setting? (would restore after the screenshot is taken)
- viii. At the bottom are options for OK and Cancel.
- c. Next to where the Narrative file has been identified, a Preview button is enabled so that the user can preview the session before generating the clip.
- d. Any content in overlays of the narrative nodes must also be included in the movie
  - i. **How to handle audio or video from Narratives in movie output?**

## Creator functionality to accommodate on load

The following must also be accommodated:

### Degree of freedom triggers

I.e. Ferris Wheel goes 'round, doors open on proximity triggers; this is a loading issue

### Creator group-level animations

I.e. rapid cycling through geometry for special effects ... blinking traffic lights, water; this is a loading issue

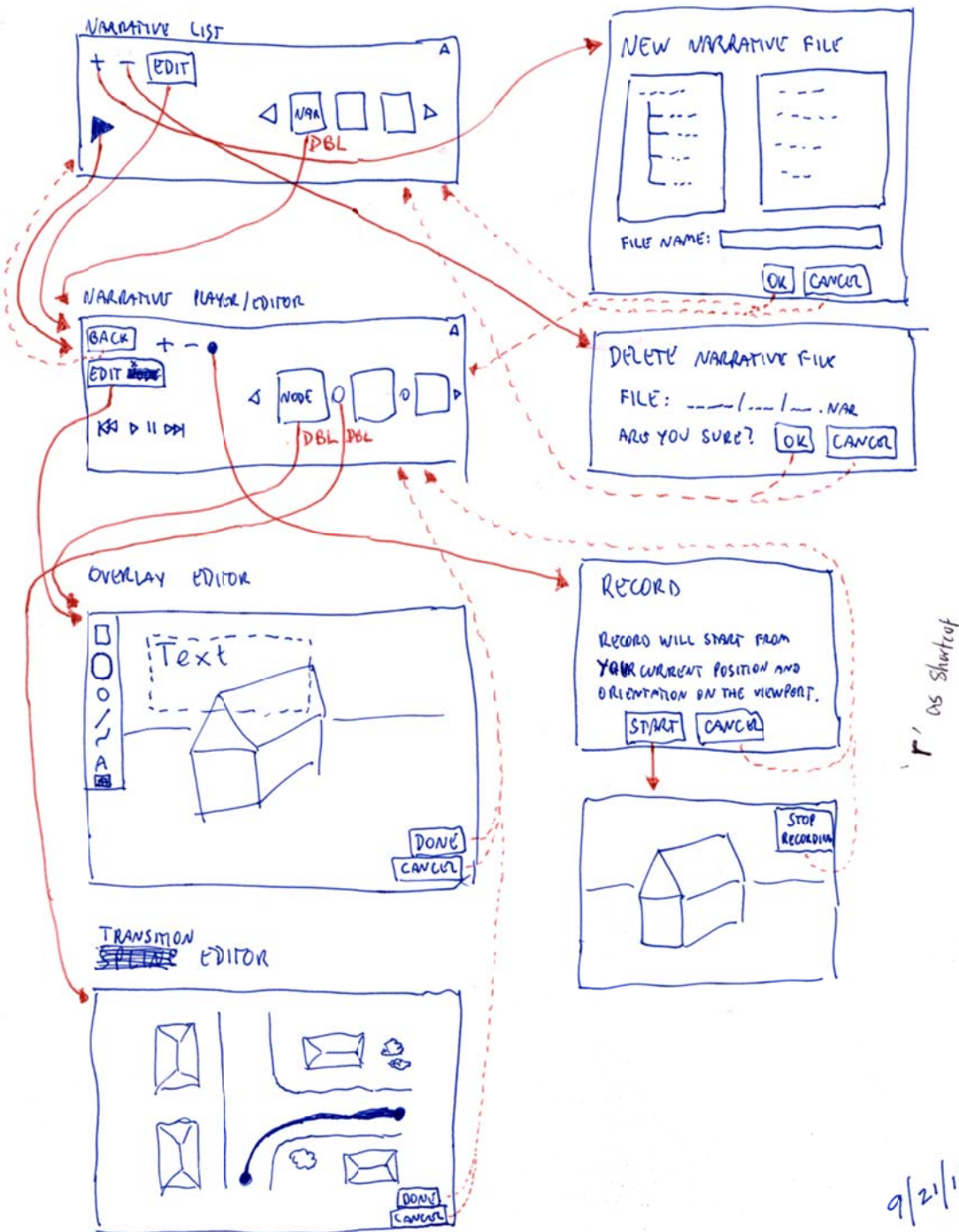
### Switches

The software accommodates two types of switches, both of which are programmed in the Creator model files. No other type of switch is accommodated at this time; in Phase 2, spreadsheet type switches may be added.

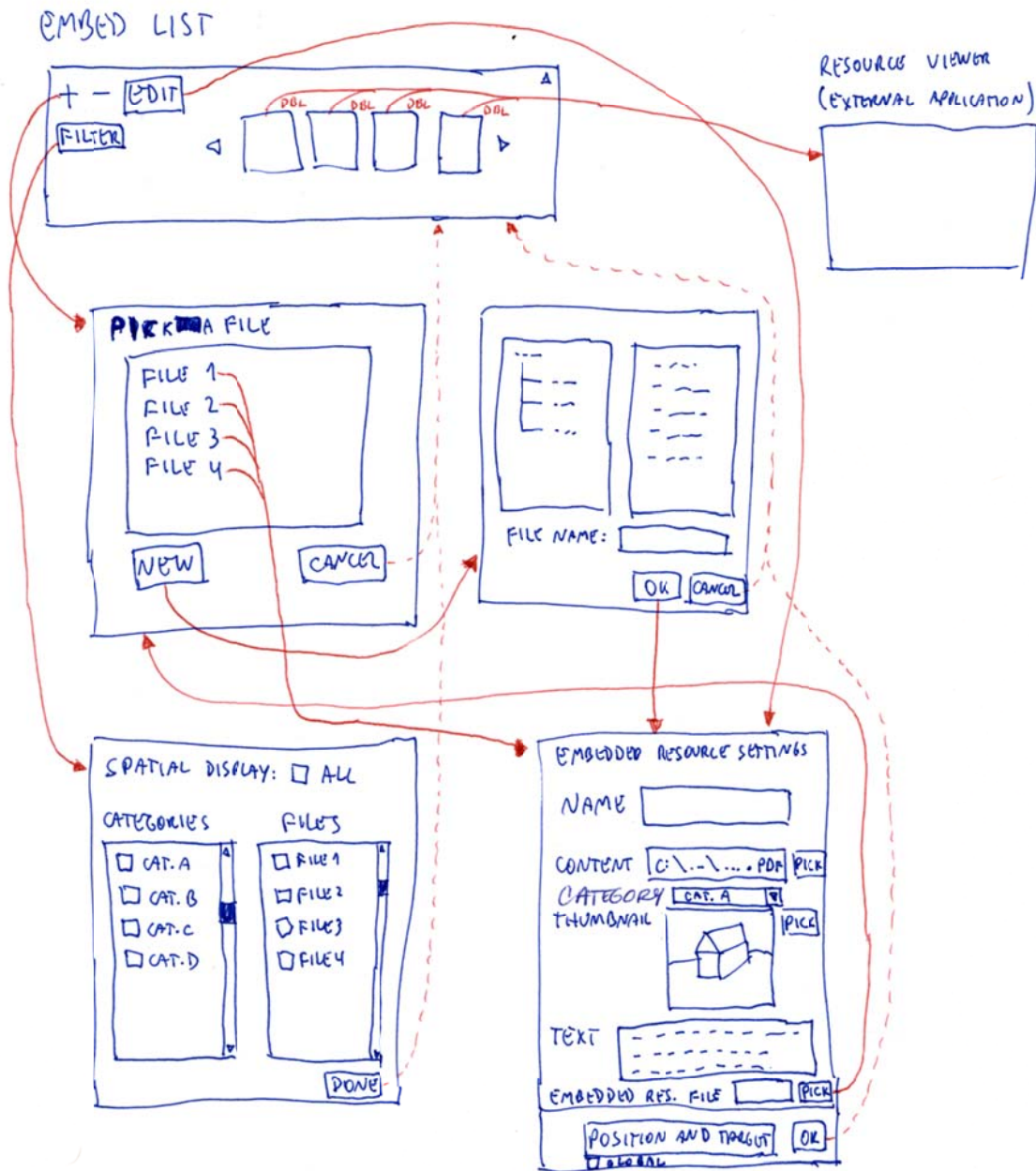
1. Controls for all switches must be programmed within the Creator switch mechanism.
  - a. For existing uSim models, this means the Group Node comments must be converted to Switch Attribute IDs and Index Names to function properly.
  - b. For vrNav models that include timed switches (e.g. Karnak), this means the various switches in the Creator files must updated to function properly.
2. On loading, the software must read the Creator switch attributes and isolate the node ID, and the mask index number and name for each mask within the switch.
  - a. The Switch Attribute Node ID is the name that will be displayed in the Switch Manager.
  - b. The individual Mask Names will be the names that are displayed in the Options section of the Switch Manager.
3. For a model using a basic switch (i.e., switches that allow the user to toggle through a number of modeled options pre-determined by the content creator), the Switch Attribute Node ID should be an alpha-numeric string that is meaningful to describe the general switch content (e.g. Pauley Pavilion)
  - a. The individual Switch Index Names should be alpha-numeric strings that describe the contents of the current mask that will be displayed (or not) within the model (e.g. Existing Conditions)
4. For a model using date-trigger switches, the Switch Attribute Node ID must be begin with the string "TIME: " followed by the *model name* (e.g. "TIME: Karnak")
  - a. All switches loaded in a single flight session that include the string "TIME: " in the Switch Attribute Node ID will be included in the time slider.

- b. The time slider will allow the user to move through the range of dates associated with the model as identified by the dates written in the individual Switch Index Names.
- c. The lowest date in the individual Switch Index Names will be the date shown on the left of the time slider
- d. The highest date in the individual Switch Index Names will be the date shown on the right side of the time slider
- e. Graphically, small triangles along the slider will indicate the various dates; mouse over will show the year that triangle represents

## Graphic: Narrative Sequence



## Graphic: Embedded Resource Sequence



## Phase 1.5 and beyond ...

Water! Definitely needed for the Exposition!

Real-time shadow casting

Physics-based explosions, fires, wind, etc.

First-person shooter

Animated objects that react to users' actions

People and crowds

Path/Narrative creation – the God's Eye View way:

- a. The mechanism for creating paths is as follows:
  - i. The user positions himself in such a way that he sees the intended starting point of the path in front of him.
  - ii. In the Paths Management Window, the user clicks on "+" to add a new path.
  - iii. An on-screen informational text will tell the user to point to the beginning of the path with the crosshair that appears in the center of the screen. A live preview will mark the projection of the crosshair onto the ground, so that the user sees the exact point in the world. The user fine tunes his aim and clicks a button or presses a shortcut key.
  - iv. An on-screen informational text will tell the user to move to the end point of the path and point the crosshair to the intended location. A live preview will mark the projection of the crosshair onto the ground, so that the user sees the exact point in the world. The user fine tunes his aim and clicks a button or presses a shortcut key.
  - v. This operation is analogous to "add a node" in narratives, so the same shortcut key should be used.
  - vi. The path creation is done. The user immediately sees the resulting path on the viewport, and he can edit the path to fine tune it if desired.