**Textfyre UI Components**

**Functional Requirements**

**Textfyre, Inc.**

**Last Updated: Saturday, December 15, 2007**

Contents

[Map 3](#_Toc185381804)

[Description 3](#_Toc185381805)

[Functional Requirements 3](#_Toc185381806)

[Beginning State 3](#_Toc185381807)

[Layout 3](#_Toc185381808)

[Mouse Interactions 3](#_Toc185381809)

[Hint Cube (Click and Flip) 4](#_Toc185381810)

[Description 4](#_Toc185381811)

[Functional Requirements 4](#_Toc185381812)

[Beginning State 4](#_Toc185381813)

[Mouse Interactions 4](#_Toc185381814)

[Conversation Window 4](#_Toc185381815)

[Description 4](#_Toc185381816)

[Functional Requirements 5](#_Toc185381817)

[Compass Rose 5](#_Toc185381818)

[Description 5](#_Toc185381819)

[Functional Requirements 5](#_Toc185381820)

[Progress Card Catchall and Progress Cards 5](#_Toc185381821)

[Description 5](#_Toc185381822)

[Functional Requirements 5](#_Toc185381823)

# Tabs

## Settings

The settings tab will be a small tab beneath the document window that when clicked, raises up to show settings information that the user can change. Settings include whether the hint cube is displayed or not, whether the progress card holder is displayed or not, whether progress card animations are displayed or not, whether the map is available or not, and whether the compass rose is available or not. Clicking on the tab again hides the tab.

## Save/Load

The Save/Load tab will be a small tab beneath the document window that when clicked raises up to show the save/load screen. This will be a list of saved games on the left hand side. When a game is selected, the location within that saved game is displayed along with a summary of goals accomplished (to the right of the list). It will also show the progress card catchall with the achieved progress cards displayed. The user can click on the progress cards here the same as when they are playing the game to review them. The user can click a Load button to load a selected save game or they can click Save, which will automatically set all of the information including a default title. The user can change the title at any time.

# Map

## Description

The map user interface component will be a sizable, see-through wire map of the current game state. Locations that have been visited will be displayed and will have details about that location. The user will be able to open and close the map at anytime and anytime the map changes, its icon will flash on the screen. The user will be able to resize the map and change its opacity. The user will also be able to hover the mouse over a location, which will be magnified with detailed information.

## Functional Requirements

### Beginning State

The map will start as an icon in the lower middle portion of the screen, underneath the document window.

### Layout

The map will be a wire diagram with connecting lines between each location. Each location will be a rectangle with rounded corners. The map colors will be determined on a game by game basis. Each location within the map must have two states; one for visited (in which case it is visible), and one for not-visited (which renders it invisible, but leaving the connectors visible).

The smaller, hover-only version of the map will be dots and lines, although if the player hovers over a dot, we may want to expand it to show the details.

### Mouse Interactions

1. One or Two Clicks will cause the map to expand to 80% of the screen (centered) with a 50% opacity OR to the previous expanded setting. A secondary click will cause the map to contract back to the icon. There should be an animation of the map expanding and contracting.
2. Dragging the Icon will cause the map to expand from the point of the icon outward. When the user releases the mouse, the map will retain its size and position in relation to the icon. A click on the map will contract the map.
3. Hovering over the icon for at least one second will cause the map to appear (no animation) quickly in a very small version with the current location highlighted. When the mouse moves away from the icon, the small map will quickly disappear.
4. When the map is expanded, hovering over a location will cause that location to expand and display detailed text and an image. The expanded location will animate to its larger size and surrounding locations and connectors will be overlaid. An alternative might be to have the surrounding locations and connectors “make way” for the expanded location, but this is currently not required.

# Hint Cube (Click and Flip)

## Description

The hint cube will be a 3-dimensional cube on the left side of the user interface even with the bottom left corner. It should take up about half of the height of the document window. At any given time it will be display “Do you want a hint?” and in small type “(click me)”. If the user clicks the cube, a question will appear. The user can keep clicking until they get a question they want answered. Then they can either click a button that comes with the question or they can “flip” the cube any direction. Flipping is the same as quickly dragging the cube any direction. When flipped or when the button is clicked, the cube will rotate and on the newly displayed “face”, the answer to the hint will be shown. After thirty seconds the cube will auto-flip and the “Do you want a hint?” face will be shown again. The cube will also have a face for settings that can be displayed that allows the user to hide already viewed hints or leave them visible.

## Functional Requirements

### Beginning State

The cube will display “Do you want a hint?”, a “Get Hints” button. The cube will be slightly offset so that the user can see the settings face.

### Mouse Interactions

1. Clicking the “Do you want a hint?” face will flip the cube backward (top toward you, bottom away) and display a hint relative to the current location or game state. After three seconds, the cube will automatically flip forwards back to the original state. While a hint question is displayed, the partially visible right-face will show “Answer”.
2. Clicking the “Get Hints” button will do the same as #1.
3. Dragging the cube downward will do the same as #1 only it will allow the user to control the animation.
4. Clicking the “Settings” face will flip the cube to the left and display the following settings:
   1. Display previously viewed hints [ ] (toggle on or off with an “X”)
   2. Display amusing commands [ ] (toggle on or off with an “X”)
5. While a hint question is displayed, the user can click the face again which will flip it backward again, showing a new question or looping back to the first question.
6. Clicking “Answer” will flip the cube to the left and display the hint answer. The partially visible right face will display “Back”.
7. Click “Back” will flip the cube to the right, displaying the question again.
8. The user can also drag the “Settings”, “Answer”, or “Back” face to the left to flip it back.

# Conversation Window

## Description

The conversation window will be a list of dialogue presented as coming from the player or player-character. The list should be scrollable. If the user clicks a phrase, the phrase should “fly” to the document window where a translation of the command is sent to the virtual machine. An example might be “I suppose if I get a better look at leather goods, I might find something interesting.” is translated to “examine leather”. The tricky part is that I want the animation of the phrase to the document window.

## Functional Requirements

1. Display a list box with wrap-around capability and vertical scrolling.
2. When a phrase is clicked, it is animated as flying to the document window.
3. A translation is sent to the VM.

# Compass Rose

## Description

A 3-dimensional compass rose that allows the user to click on the 8 major directions, but also has mechanisms for “in, out, up, and down”. The design of the compass is open and might be a sphere with paddles or tabs. The physical design should be done in partnership with our graphic illustrator, Erika Swanson. Each clickable element should also have hover states as well as “lighted” states to show the player which directions are currently “open”.

## Functional Requirements

1. Program must be able to tell component which directions (North, Northeast, East, Southeast, South, Southwest, West, Northwest, Up, Down, In, Out) are available so that these can be highlighted in the control and enabled while disabling all others.
2. When user clicks control, the appropriate command is sent to the document window.

# Progress Card Catchall and Progress Cards

## Description

On the left-hand side of the screen and above the hint cube will be a round-edged square with small rectangular “pockets”. Each pocket is the size of a small trading card. When the user reaches a point in the game where they’ve solved a puzzle or moved to a new scene, the program will signify this. When this happens, a trading card image should flip into the center of the screen (with bells and whoosh sounds). While it’s in the middle of the screen, the user can click it to look at both sides. There should be a small bar with the word “close” at the bottom of the card. If the close bar is clicked, the card spins and shrinks to the appropriate pocket on the left-hand side. The user can click any card on the left-hand side and the card should spin and grow back to the center of the screen to be reviewed.

## Functional Requirements

1. Display a rectangle with slots for images.
2. Animate an image out of “nowhere” onto middle of screen.
3. Animate and shrink image from middle of screen to a predetermined pocket.
4. Animate and grow image from pocket to middle of screen