User Manual

Group 3

June 8, 2022

Contents

Revision History

Revision	Date	Author(s)	Description
0.0.1	10.04.22	pl942	Draft in Word
0.1.0	06.06.22	dm1306	Transcribe into LaTeX
0.2.0	07.06.22	dm1306	Add more content
0.2.1	07.06.22	pl942	Add to sections
0.2.1	07.06.22	dm1306	Revise

1 Introduction

Welcome to the SuperPres User Manual. The User Manual contains all essential information for the user to make full use of the application. This manual includes a description of the system capabilities and procedures for application use.// This document is designed to be read by any user of SuperPres considering most users will have access to the features documented.

2 Description and Operation

2.1 Supported Formats

The application supports the following media file formats:

- Video
  - MPEG-2
  - WebM
- Audio
  - Opus
  - MP3
- Images
  - JPEG
  - PNG
  - BMP
  - GIF

Text is entered visually through our User Interface, which shall be considered later in this Section.

2.2 Navigating the UI

When the application is opened, a start screen containing two butttons is displayed. The "Open a File" button will open a file-picker dialogue to allow you to choose your desired file; files with the "zip" extension will not be displayed by default - to display these, click on the file-format drop-down menu in the file-picker (which should currently be displaying "spres" or "SuperPres") and select "zip" from the list.
The "Start New" button shall open an empty document, and display an empty Card.

2.2.1 Navigating a Stack

When a Stack or Document is loaded, the Cards in the Stack shall be listed, displaying their title, in the bar at the bottom of the window; to jump to a Card, simply select it from the list. It is also possible to navigate forwards and backwards by pressing the Right and Left arrow keys respectively; the "back" behaviour navigates to the previously displayed card, as in a Web Browser, whilst "forwards" navigates to the next Card in the sequence.

2.2.2 Editing a Stack

To add another Card, simply click on the large "Plus" button at the end of the Card list, located in the bottom-right of the window. To add an element to a card, click on the icon of the desired tool from the tool-box in the top-left of the Window, locate your cursor at the desired element location, and then click to add the element to the Card.
To Edit an element, click on the Edit tool icon (A large cross-hair icon) in the tool-boxm then click on the element you desire to edit; you will then notice the element's properties are displayed in the properties box towards the bottom left of the Window - editing these properties shall edit the element.

2.2.3 Saving Your Changes

To save your changes, click on the "File" menu at the top-left of the Window, and then click on the "Save" or "Save As" item (depending on if you desire to save the file to a new location). If "Save" is selected and the Stack is new, a file-picker dialogue shall be displayed for you to select the desired save location as would be the case for selecting "Save As". If the Stack is not new, pressing "Save" will save your changes to the original location. The file format is simply a Zip archive containing all your Stack assets.

2.2.4 Console

To load the Python Console, select the "View" menu item from the menu-bar at the top of the Window, and then click on the "Show Console" option - a new window containing a Python Console shall then be displayed.

2.2.5 Editing Scripts

To edit an element's scripts, select the element with the "Edit" tool, and an "Edit Script" button shall then appear at the bottom of the properties box; clicking this button shall cause an "Editor" window to be displayed containing the element's current script if any is present or an empty box if not, a "Save" button, a drop-down box containing the supported languages (set to "Python" by default), and the ID of the element the script is bound to.
To write your script in a language other than Python, click on the langauge drop-down menu and select your language, write your script, and then click on the "Save" button.

3 Error Handling

If an error is detected, the user shall be notified. If the error is severe or unrecoverable, the console shall pop-up containing the error message. If the error is simple and recoverable, it shall be displayed for between 5 and 10 seconds in the message box and the very bottom of the Window.

4 Scripting Engine

4.1 Scopes

The scope for variables and functions follows the structure of the Document; a function call or variable access from an Element on a Card shall first search that Element's scope for the item, and if the item is not found the scope of the parent Element shall be searched (in this case, the Card); if the item is still not found, the search shall continue upwards until either a SuperPres builtin function is called or the item is not found. This allows the user to place generic handlers on a Card, and have them be triggered from elements.

4.2 Triggers

The Scripting Engine currently supports automatically triggering the following functions:

4.2.1 onClick

Triggered by a click on the element. Takes the following parameters:

- Mouse Button: JavaFX MouseButton Enum Value.
- X: (Double) X location of click.
- Y: (Double) Y location of click.
- isDown: (Boolean) Is the mouse currently down?

4.2.2 onMouseMoved

Triggered when mouse moves inside of the element.

- Mouse Button: JavaFX MouseButton Enum Value.
- X: (Double) X location of click.
- Y: (Double) Y location of click.
- isDown: (Boolean) Is the mouse currently down?

4.2.3 onMouseEnter

Triggered when mouse enters the element.

- X: (Double) X location of entry.
- Y: (Double) Y location of entry.

4.2.4 onMouseExit

Triggered when mouse exits the element.

- X: (Double) X location of exit.
- Y: (Double) Y location of exit.

4.2.5 onDrag

Triggered when the element is dragged.

- X: (Double) new X location.
- Y: (Double) new Y location.

4.2.6 onLoad

Triggered when the element is loaded.

- Mouse Button: JavaFX MouseButton Enum Value.
- X: (Double) X location of click.
- Y: (Double) Y location of click.
- isDown: (Boolean) Is the mouse currently down?

4.2.7 onKeyPress

Triggered when a key is pressed and the element, or an element in the scope-path, has focus.

- (String) Key Name: Name of the key that is pressed.
- isControldown: (Boolean) Is the Control key down?
- isAltDown: (Boolean) Is the Alt key down?
- isMetaDown: (Boolean) Is the Meta (Win/Option) key down?
- isKeyDown: (Boolean) Is the named key down?

4.3 Access

The scripting engine has access to all public methods of the Engine and Document, through the variables “engine” and “document”. The current page is also exposed directly to the engine through the “currentPage” variable, and the current element is bound to, in Python, “this”, and in other languages, such as JavaScript, to “me” to avoid collisions. The best documentation for the Elements and Engine methods is found in our JavaDoc; our examples may also be used.