

Illumia Labs *Scenario Builder* Discount Evaluation

AYRTON CHILIBECK*, the University of Alberta, Canada

ERIC KIM*, the University of Alberta, Canada

YU LIU*, the University of Alberta, Canada

VEDANT TALATI*, the University of Alberta, Canada

MARCUS WILSON*, the University of Alberta, Canada

ACM Reference Format:

Ayrton Chilibeck, Eric Kim, Yu Liu, Vedant Talati, and Marcus Wilson. 2024. Illumia Labs *Scenario Builder* Discount Evaluation. In *Proceedings of (CMPUT 302 '24: Illumia Labs)*. University of Alberta, Edmonton, AB, CA, Article, 6 pages. <https://doi.org/N/A>

1 INTRODUCTION

We assess Illumia Lab's *Scenario Builder* to identify areas for enhancement and offer a short-term development roadmap. We address issues in UI, system functionality, and program documentation, drawing upon Human-Computer Interaction (HCI), Gestalt, and CRAP design principles. Our solutions align with established HCI research, color theory, and user experience insights.

2 SYSTEM FLAWS

Throughout our investigation and utilization of the system, we identified issues in the UI, program functionality, and documentation. We highlight the key findings in the subsequent sections.

2.1 UI

Our examination of the UI primarily addresses cosmetic issues, yet the current software state hinders effective user interaction. The layout fails to efficiently display scene information, and scene transitioning demands excessive user effort. Moreover, clear indications of appropriate user actions are lacking throughout the scene-building process.

2.1.1 Color Scheme. The existing color scheme (Purple (#07012F), Blue (#0191FD), and Red (#FC5C00)) is visually straining. Research indicates that prolonged exposure to red and purple hues can be taxing for users [10]. Some users may prefer alternative color schemes, and implementing a light and dark mode could enhance interface versatility. Additionally, the use of red to signify the program's desired state among multiple choices may confuse users, given red's association with 'negation' or 'emergency' decisions.

2.1.2 Tab Display. The current tab display in the scene builder inadequately communicates the program's status to users. Tabs for each scene lack context regarding their purpose or contained information. Although the preview pane partially addresses this issue, the scene-graph representation lacks clarity regarding scene relationships.

*All authors contributed equally to this research, and are listed in alphabetical order for simplicity

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2024 Copyright held by the owner/author(s).

Manuscript submitted to ACM

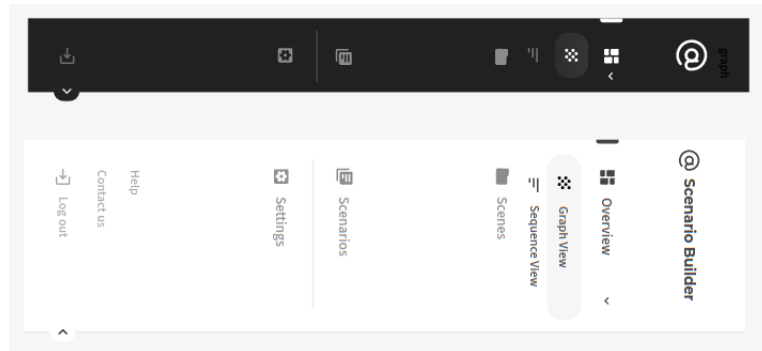


Fig. 1. A prototype sidebar navigation system.

2.1.3 Preview Pane. The preview pane suffers from poor alignment and lacks dynamic screen resizing, diminishing the utility of presented data, especially for mobile and resizable web pages.

2.1.4 Ease of Use. Constructing a scene currently requires a minimum of 9 clicks. While the '3-click rule' has been debunked [7], ensuring ease of information access remains crucial in design. Recent research on 'Interaction Elasticity' underscores the importance of minimizing unnecessary interactions [9]. Presently, the scene builder imposes excessive interaction burdens on users through these clicks.

2.2 Functionality

The website currently lacks implementation of several usability-enhancing features, impacting user experience. We outline the affected systems below.

2.2.1 Saving. At present, the system does not support scene saving or the ability to resume work on previously saved scenes. This limitation hampers users' ability to create meticulously crafted scenes.

2.2.2 Avatar. The inclusion of an avatar in scene development appears superfluous, as its necessity is not reflected in the builder's stated requirements or business logic. Observing the intention to include the avatar, we assume that this feature is not yet implemented.

2.3 Documentation

Overall, the builder suffers from inadequate documentation. Many terms and software interactions remain unexplained, decreasing user understanding of the program's functionality.

3 REMEDIATIONS

We propose a number of possible solutions to the problems mentioned above that will improve the usability of the application and touch on additional, smaller problems that we neglect to mention in this report, but are nonetheless important.

3.1 UI

We suggest a sidebar-based UI such as pictured in Figure 1. This UI paradigm will provide the user with more information on any given scene in the program and allow the user to recognize the full state of the program at a glance without wading through screen after screen of data.

When combined with an information page for the selected screen (or the graph view of connections between scenes), we can give the user a more granular view of the data. It is important to note that this type of modification is an overhaul of the frontend design for the website, so the transition will not be easy but the application of this type of UI is well loved in modern UI design [8] [6] [3]. Adding the flexibility for the user to navigate to any scene in a single click will then reduce useless interaction with the program, improving the user experience.

This type of UI will also make error handling more visible to the user. Instead of using the single exclamation point, we can highlight all the fields invalid data in the associated page and provide a clear message to the user with respect to the nature of their violations all on one screen. We can also highlight the issues in all scenes by highlighting nodes on the sidebar as required. This way, the user does not need to chase every bug individually and can rather see where every bug is without having to perform trial and error modifications.

3.1.1 The Scenarios Folder. You will notice in the mockup that we have added a scenarios folder. This serves to give the user a space to switch between different scenarios they are developing, and possibly to store scene elements they use frequently.

3.1.2 Log Out. The sidebar also contains a Log-Out button, it is assumed that eventually the avatar will be associated to the user in order to create some kind of account for the user to log in and out with. This would be an easy way to implement this feature.

3.1.3 The Scenario Page. The accompanying scenario page shown in Figure 2 will give the user the options to input all necessary data and efficiently view all of their options. This will provide a visual interface for the user to modify their scene without added interaction.

Additionally, there is the option to outline the elements in red or some other colour to indicate a non-conforming piece of data.

We could also implement a dropdown in the sidebar in each scene that gives a short summary of the data entered and can also indicate errors in the data entry. This will give users an even more general view of their scene.

Note: We are unsure of the difference between the emotion and reaction elements. The two descriptions seem to be linked, and as such we wonder if the program could eliminate one or the other.

3.2 Functionality

Introducing a sidebar navigation paradigm can streamline the user experience by simplifying actions previously executed through the tab layout. This includes eliminating arrows above tabs and the delete button, which can now be integrated into the sidebar element or its dropdown menu. Additionally, this paradigm resolves various alignment issues, consolidating data control into a single point for easier manipulation and viewing.

3.2.1 Saving. We propose organizing all user-stored scenarios within a dedicated 'Scenarios' folder. Account status and data could be displayed on a ribbon across the screen's top, potentially featuring links to other tools licensed by

Scene Builder

My Scenario

- Overview
- Scenes
 - My Scene
 - Other Scene
- Scenarios
- Settings
- Help
- Contact us
- Log out

User Scene Editable

Prompt:*
Type here

Transitions:*

Response	From	To
	Select	Select
	Select	Select
	Select	Select
	Select	Select

Hint:
Type here

Emotion:*
Select

Reaction:*
Select

Fig. 2. A mockup of a scene page for editing individual scenes in the program

Illumia Labs. This approach not only facilitates scenario storage and template usage, but also promotes engagement with Illumia Labs' entire ecosystem.

3.2.2 Avatar. We recommend integrating the avatar creation process into the account creation phase and removing avatar functionality from the scenario builder. This eliminates confusion arising from the opening scene's request for an avatar while allowing users to proceed without one. Additionally, abstracting the avatar creator from the builder decouples account logic from the builder's business logic, leading to improved system-level encapsulation.

3.3 Documentation

Developing succinct and useful documentation will help the user experience immensely. Currently, it is unclear what the scenario builder does based on the interface. We suggest adding a 'help' icon to present a brief description of what a given element does and pointing the user to a full documentation page if the logic is too complex to describe in a tooltip.

4 SUGGESTED ROADMAP

Due to the number of distinct faults we suggest changing and their complexity, we suggest proceeding with the following roadmap in mind:

- **4-6 weeks** UI Revamp
- **4-6 weeks** Documentation site

- **1 week** Integrate the documentation as tooltips and links on the UI
- **2-4 days** Choose a different colour scheme
- **4-6 weeks** Add a graph representation of the scenes and their interactions

For a grand total of 20 weeks of work. We will detail the phases below.

We should note that much of this work can be done collaboratively and in parallel, thus the work cycle should not take the full 20 weeks we estimate.

4.1 UI Revamp

We propose changing the UI to reflect our suggestions. This will require making a new landing page and then implementing the business logic and making the appropriate changes to the representation and functionality we described in section 3.

Ideally, this is accompanied by changes in the backend to better reflect the data acquired from the user in their interactions with the program. We will not comment more on the backend due to our lack of exposure.

4.2 Documentation

We propose mounting a separate website for documentation using some easily configurable and customizable system such as Docusaurus [2], GitBook [4] or Docsify [1]. This will need to contain the following sections:

- **Quickstart:** A section to get the user acquainted with the software quickly, exposing them to the most important and relevant features of the program.
- **Introduction:** A section to describe the purpose of the software (possibly pointing to examples of existing systems or your own examples).
- **Documentation:** A section to fully document all features of the site, including individual actions the user can perform and their expected effect. This section may also contain information as to data input requirements and data use and privacy.

4.3 Integrate Documentation

This consists of linking the documentation created in the previous step and linking it to the correct element associated with the action. This also embodies the creation of tooltips and help buttons as required on every screen. This will allow the user to quickly and efficiently find the answer to their question and increase the usability of the site in general.

4.4 New Colour Scheme

Considering the taste of the current colour scheme (relatively dark and cool colours), we suggest implementing a colour scheme of complementary colours.

4.5 Graph Representation

The graph representation of scenes will serve as a replacement for the preview panel. Instead of providing a text based representation of the relationships between scenes, consider providing a visual one such as displayed in figure 3. A popular example comes from Obsidian's knowledge graph [5], which contains back-links and grouping to visually demonstrate the relationship between different nodes in the graph.

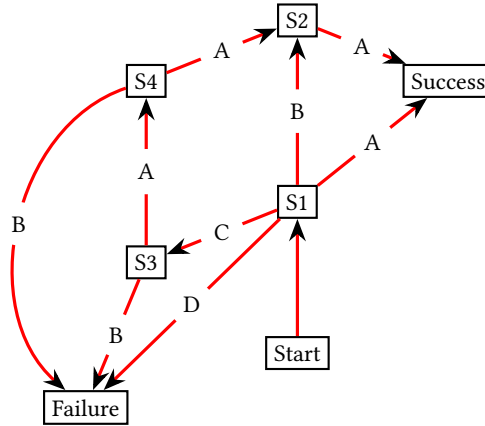


Fig. 3. A depiction of the relationship between scenes using a graph representation. Adapted from [11]

An automatically generated graph could easily be produced from the data in a scene, but rendering the graph would take some engineering, hence the predicted time allotment.

In addition to allowing the user control over the way their scenes interact and providing more detailed information on the scenario structure, this also adds an option for a new type of node creation that could be done visually in the graph. This, though an interesting idea, we leave cost-benefit analysis to the reader.

5 CONCLUSION

Throughout this paper, we have highlighted various visual, functional, and documentation deficiencies within the Illumia Labs *Scenario Builder*. Our assessment reveals a user interface that is cumbersome to navigate, characterized by unnecessary interactions, which diminishes program usability. Issues such as color scheme, layout, alignment, and contrast, as well as several Gestalt principles outlined in Section 2, contribute to this usability challenge.

In response, we propose a comprehensive five-step plan to rectify these identified issues, offering potential solutions for each problem. Additionally, we delve into specific UI problems in the appendix and advocate for a complete overhaul of the UI. Our analysis draws from both scholarly literature and prevalent community practices, providing valuable insights to enhance the attractiveness and user-friendliness of the Illumia Labs *Scenario Builder*.

REFERENCES

- [1] [n. d.]. *Docsifyjs/Docsify: A Magical Documentation Site Generator*. <https://github.com/docsifyjs/docsify/>
- [2] Meta [n. d.]. *Facebook/DocuSaurus*. Meta. <https://github.com/facebook/docuSaurus>
- [3] [n. d.]. *Flutter - Build Apps for Any Screen*. [//flutter.dev/](https://flutter.dev/)
- [4] [n. d.]. *GitBook - Knowledge Management for Technical Teams*. <https://www.gitbook.com/>
- [5] [n. d.]. *Obsidian - Sharpen Your Thinking*. <https://obsidian.md/>
- [6] Supabase [n. d.]. *Supabase / The Open Source Firebase Alternative*. Supabase. <https://supabase.com/>
- [7] [n. d.]. *Three-Click Rule*. https://en.wikipedia.org/w/index.php?title=Three-click_rule&oldid=1173851653
- [8] Notion [n. d.]. *Your Connected Workspace for Wiki, Docs & Projects*. Notion. <https://www.notion.so>
- [9] World Leaders in Research-Based User Experience. [n. d.]. *Interaction Elasticity*. Nielsen Norman Group. <https://www.nngroup.com/articles/interaction-elasticity/>
- [10] Mark Kirkland Jones. [n. d.]. *Human-Computer Interaction: A Design Guide*. Educational Technology.
- [11] Torbjørn T. [n. d.]. *Answer to "Draw a Graph in Latex with Tikz"*. TeX - LaTeX Stack Exchange. <https://tex.stackexchange.com/a/270546>