HelpWanted.js

CSC309 Individual Project Alpha Release

Neeco Fabian

About the Library

HelpWanted.js is a JavaScript library to conveniently create website tours and instruction tutorials. Developers can create instructions which the library wraps into a series of modals. In Guided Mode, a sequence of annotations emphasize specific parts of the website. Modals can be pinned to desired page elements and appear one by one as users step through the tour. Further, the standout feature of *HelpWanted.js* is Discovery mode, which rethinks what Help buttons do. Instead of leading to FAQ's or a dictionary, Discovery Mode darkens the web page, except for elements with a pinned annotation. Users can then visually explore the page by hovering over an element, which reveals the developer's annotation. If a user is unsure what an element does, instead of traversing FAQ's and definitions, the Help button can turn on Discovery Mode so the user can simply point to the element and read the developer's descriptive annotation. Developers can choose between the modes, tweak background darkening, and other styles (colour, border radius). *HelpWanted.js* is applicable to websites where users require additional direction to maximize the services. For example, online text editors such as Overleaf.com and Google Docs can use Guided mode to point out new features for returning users. There can also be a separate tour for new users to learn about the basic functionality. Additionally, browser games such as Agar.io and RuneScape can use the library for the initial game tutorial. Annotations can sequentially point to UI elements, which are often scattered across the screen and usually make modals tricky to position. Thirdly, Discovery mode is perfect for adding annotations to websites like Bell.ca and Scotiabank.com, which contain confusing terms during sensitive processes like billing. These could save users from making errors by making information visual (through pointing) and available on-demand. The goal of *HelpWanted.js* is to let developers shift their time towards how they present information, rather than the implementation of the tours and tutorials.

Figure 1: Proposal mock-up of Guided Mode. Note: only one pin is shown to the user at once.

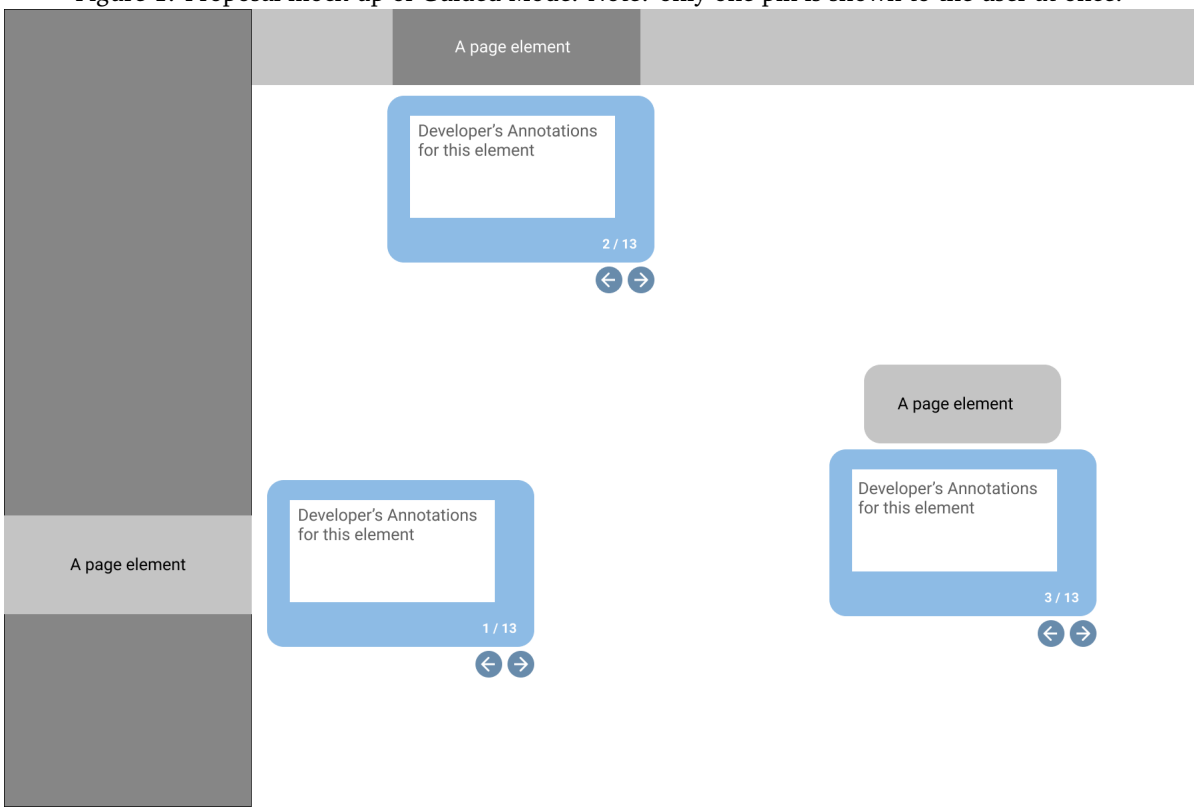
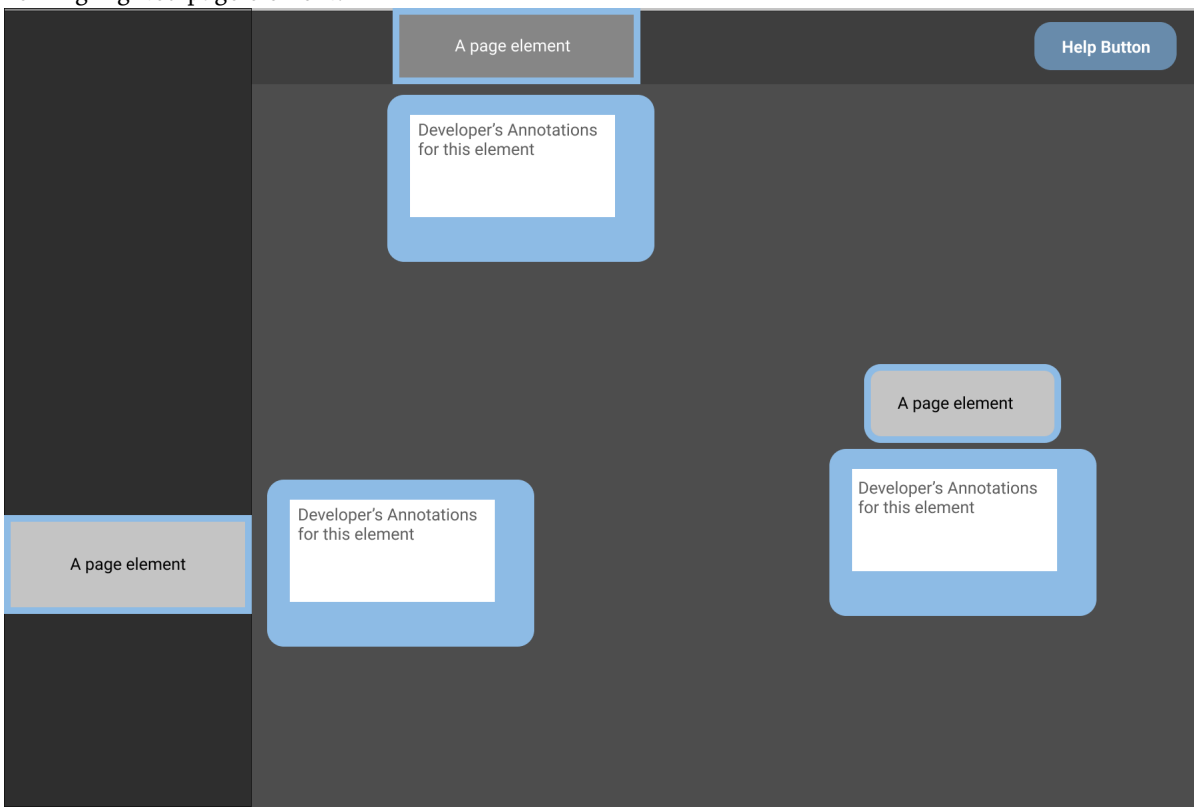


Figure 2: Proposal mock-up of Discovery Mode. Note: annotations are only shown when a user hovers over their highlighted page element.



Alpha Release Notes and Features

After doing more research on modal libraries, I found that Centered Mode (which I proposed in the first phase as a single, centered modal with instructions to swipe through) and other slide show-type libraries are frequently implemented. Though I could have continued with it, I wanted to spend time on features that provide more unique functionality to make my library stand out. That is why for my alpha and final release, I will not be implementing Centered mode. Guided and Discovery mode on the other hand, share a common theme of allowing developers to highlight specific sections of their site, which gives a clearer focus for *HelpWanted.js*. In the alpha, both features are functional; I will go into more details in the API. Developers can construct `Sequence` objects and create pins that are attached to the top, bottom, left, or right side of an HTML element. As long as the content is an HTML object, including nested elements, it can be placed inside the pin as the `content` attribute. Also, any HTML element can be used as the element to pin to. When Guided Mode starts, users can now traverse through the sequence using arrow buttons. The "<" arrow button is disabled in the first pin and the ">" arrow button is replaced with a close button in the last pin. I also added key listeners so users can press the left and right arrow keys to navigate through the sequence. Developers can call the static functions `next()` and `back()` to manually traverse the list using their own buttons and listeners. And in Discovery Mode, the dark background is added if desired, while elements with pins are uncovered. Hovering over a highlighted element now adds an outline around it and reveals its corresponding pin. When finished, users can click the close button or press the `ESC` key. Similarly, developers can remap this functionality to their custom buttons using the `stop()` function. The styles for the library are stored in `helpwanted.css`, while the functionality is in `helpwanted.js`. Moreover, something to note is that a single sequence can be switched between Guided Mode and Discovery Mode, which eliminates the need for duplicate code and objects. Methods exist to facilitate the toggle. Developers can make as many distinct `Sequence` objects but only one sequence can be running at a time.

Deployment

The deployed site contains descriptions of `HelpWanted.js`'s general functionality and its API, which includes all functions and descriptions. There is also a mock navigation bar to facilitate with the demo. To try out Guided Mode, click the "Start Demo Tour" button which takes the user on a brief tour of the page and the library's main features. "See Annotations" turns on Discovery Mode and uses the same sequence as the the demo tour. In this case, "See Annotations" behaves as a "Help" button implemented with `helpwanted.js`. All features specified in the previous section should function. The deployed site can be found here:

http://helpwantedjs.herokuapp.com

Object Structures, DOM, and API Functions

The most prevalent DOM-related objects in `HelpWanted.js` are Pins. A Pin includes `content`, an HTML element that contains the information the developer wants to display in the annotation, `element`, the HTML element to pin onto, and `discoveryMode`, which represents if Discovery Mode is enabled. Additionally, a pin contains its `position` in relation to the element it is being pinned to, and `pinElem`, the wrapper HTML element that is eventually inserted into the DOM. Inside of `Pin._showPin()`, `discoveryMode` determines what gets inserted in the DOM because a Pin shown on Guided Mode (when `!discoveryMode`) contains the "<" and ">" arrow buttons and a paragraph element that represents the user's progress in the sequence. Thus, after `pinElem` is created, `content` is first inserted into `pinElem`, followed by the mode-appropriate elements. Only then is `pinElem` placed in the HTML body. Positioning of `pinElem` is then done with styling. Once it's time to hide the Pin, the Pin object's reference to `pinElem` allows for the easy removal of the pin from the DOM. Looking beyond, pins are stored in a list attribute in Sequence object it belongs to, though this is not important to the DOM. Even further, Sequences are stored in global list variable `sequences`.

This is an Pin object that has just been displayed:

```
{
    content: div.content,
    discoveryMode: false,
    element: button#navBtn1.navbarButton,
    pinElem: div.pinElem.visible,
    position: "bottom"
}
```

Another DOM-related object is the global variable `overlayContainer`. It is inserted into the DOM in `Sequence.start()` if Discovery Mode is on and the developer sets `darkenBackground` to true. `overlayContainer` includes if it should be `visible`, the actual `overlay` div, and the `closeBtn` to exit Discovery Mode. Once the `closeBtn` is pressed and `Sequence.stop()` is called, then `overlay` and `closeBtn` are removed from the DOM and `visible` is set to false. Here's an example of `overlayContainer` when the overlay is in the DOM.

```
{
    visible: true,
    overlay: div#overlay,
    closeBtn: button#overlayCloseBtn
}
```

I expect overlayContainer to gain more attributes as I add more customization. Therefore, when pins are displayed and hidden, and background is obscured, the DOM is manipulated. The following is a subset of the API Functions that developers have access to (the full list is in the deployed page):

- `next()`
  - Displays the next element in the sequence currently running. Stops the sequence if the current pin is the last pin in the sequence. Called in Guided Mode only.
- `back()`
  - Displays the previous element in the sequence currently running. Has no effect if the current pin is the first pin in the sequence. Called in Guided Mode only.
- `Sequence(discoveryMode=false, darkenBackground=true)`
  - Creates a new `Sequence` object. Sets which mode is called when `Sequence.prototype.start()` is called. Sets the boolean that darkens the background during Discovery Mode.
- `Sequence.prototype.addPin(content, element, position="bottom)`
  - Adds a pin containing the HTML element `content` on the `position` side of HTML element `element`. `position` is one of `"top"`, `"bottom"` (default), `"left"` or, `"right"`.
- `Sequence.prototype.removePin(pinNum)`
  - Removes the pin at index `pinNum` if it exists.
- `Sequence.prototype.start()`
  - Starts the sequence in the specified mode (Guided or Discovery) if no other sequence is running.
- `Sequence.prototype.stop()`
  - Stops the sequence if it is running.
- `Sequence.prototype.discoveryModeOn()`
  - Enables Discovery Mode when the sequence is started.
- `Sequence.prototype.discoveryModeOff()`
  - Enables Guided Mode when the sequence is started.

What's Next?

The main features of *HelpWanted.js* are functional. For the final release, my goal is to add complexity and customization. In Sequence, I plan on adding an optional parameter to `addPin()` to indicate the index to insert the new `Pin` into the list of pins.

With Pins, developers will have the option to add pin titles to be displayed beside the back and next buttons. Further, developers should be able to change some of the style attributes of a pin such as its border radius, button colours, and background colour. Currently, I am unsure of how to implement this in a way that gives many options, but is not verbose. Otherwise, it would be too cumbersome for the developer, and may be easier to style without the library.

For background overlay, I will include options to set the overlay colour and its transparency. Also, in this phase, I had problems implementing a blurred background due to browser support, so I will try to implement that for the final release.