Melissa Powel mpowel@andrew.cmu.edu Due: Feb 26, 2017

Project Milestone 3: Input, Controls, Info, & Progress

Web App Development 95881 A3

Summary of Changes

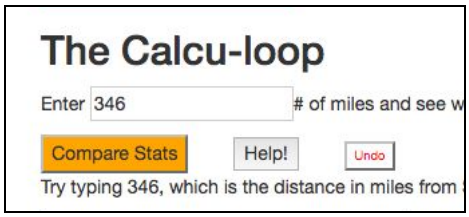
I split up my content into completely different files making the content much more ordered both for the user and for anyone trying to review my code. I have 4 html files and one JavaScript file. I also added in bootstrap js elements, which has made the overall layout of the site much more pleasant.

Superficially, I added a video of the Hyperloop team to the quiz, a photo of a high fidelity rendering of the CMU hyperloop pod to the home page, and a futuristic image of hyperloop transportation. I also adjusted the location of the footer and the margins on the side to avoid having the text come right up to the edge of the page.

Input Design Patterns

Undo

I've implemented an undo button similar to the example on the calculoop.html I also have a back button in the hyperloop_quiz.html, which is described below in the Jump to Section design pattern below.

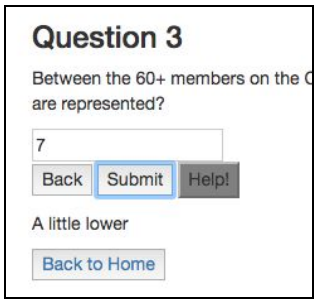


Related Context

This happens to be the same code I used for the recommendations section discussed below.

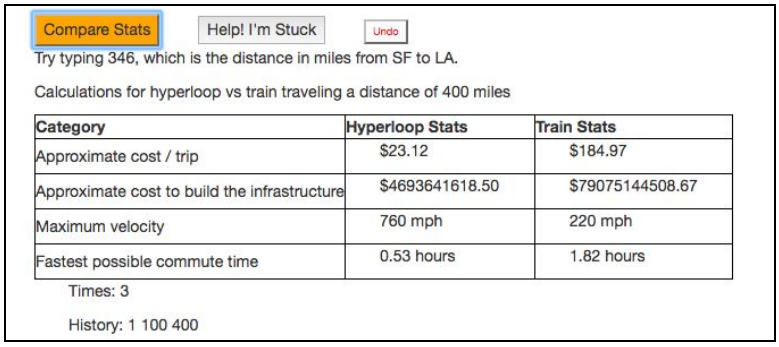
Pictured Right

1. If the user guesses a number over 5, they get the hint text: A little lower.
2. If the user guesses a number under 5, they get the hint text: Try a little higher!
3. If the user guesses 5, they get it right!



Notification & History

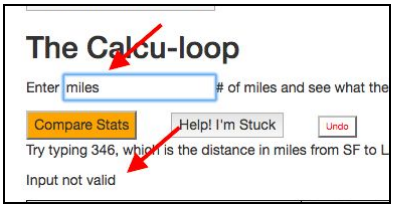
Implemented per the example to incrementally track the previous inputs and add them to the variable "count" and to incrementally count with a variable I named "diamond" (yes, diamond). These can be found at the bottom of the table on the calculoop.html page.



Input Design Patterns

Inline Validation

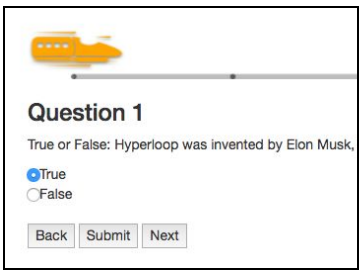
NaN validation is set up for the hyperloop calculator on the first page as soon as the user starts entering a value into the text field. For the initial validation, it's only checking for if isNaN is true, but it ignores an empty text field as the user may not be ready to enter a value. If the user leaves the input field blank and tries to click "Compare Stats" a new error message appears that follows the

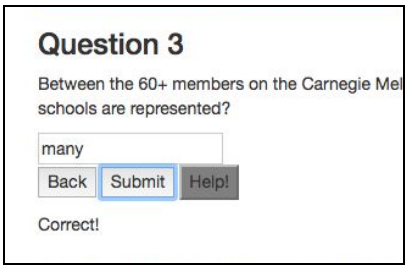


Input Hint design pattern as it instructs the user what might be wrong with their entry. See Input Hint for more details.

Default Values / Auto-Complete

All true false questions have true pre-selected. This is purposefully somewhat misleading as the correct answer to both of the true/false questions in the quiz is false.



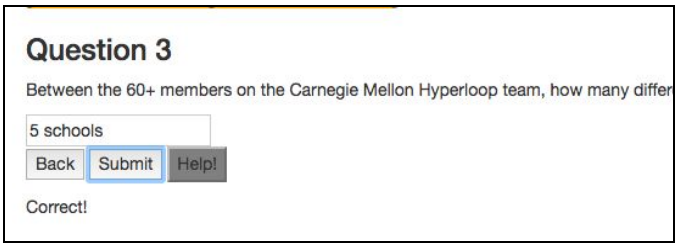


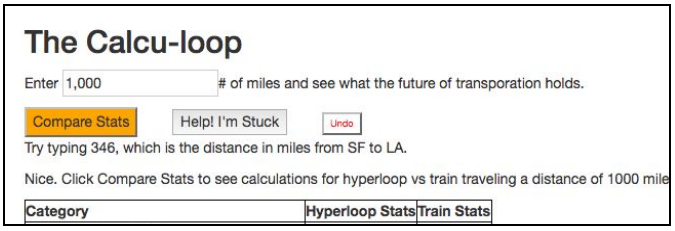
Natural Language Inputs

I fixed this from my last submission. Previously, I had 5 schools as natural language, but that's more forgiving format. I've adjusted this to allow "a lot" as an answer.

Forgiving Formats

Commas in numbers are allowed by removing them from user entries before plugging the value into the next function. Eg. both 1,000 and 1000 would be valid entries. You can see how the tool processes the information in the "Action Context" design pattern. It's receiving a string "1,000" as an input, yet the function is able to interpret the input as the integer "1000" by removing the comma.

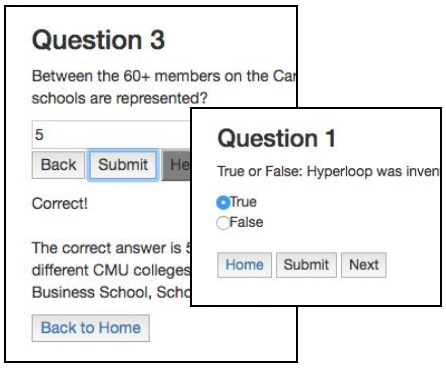




The key difference between forgiving formats and natural language is that forgiving formats usually has to do with symbols and other ways of presenting the same data, eg. phone numbers, social security number, address, etc. Natural language inputs accounts for variations in spoken language for example a greeting might be "hi", "hey", or "how are ya?". Fields that accept natural language inputs would need to account for a wide variety of ways people might say the same thing.

Melissa Powel mpowel@andrew.cmu.edu Due: Feb 26, 2017

Controls Design Patterns

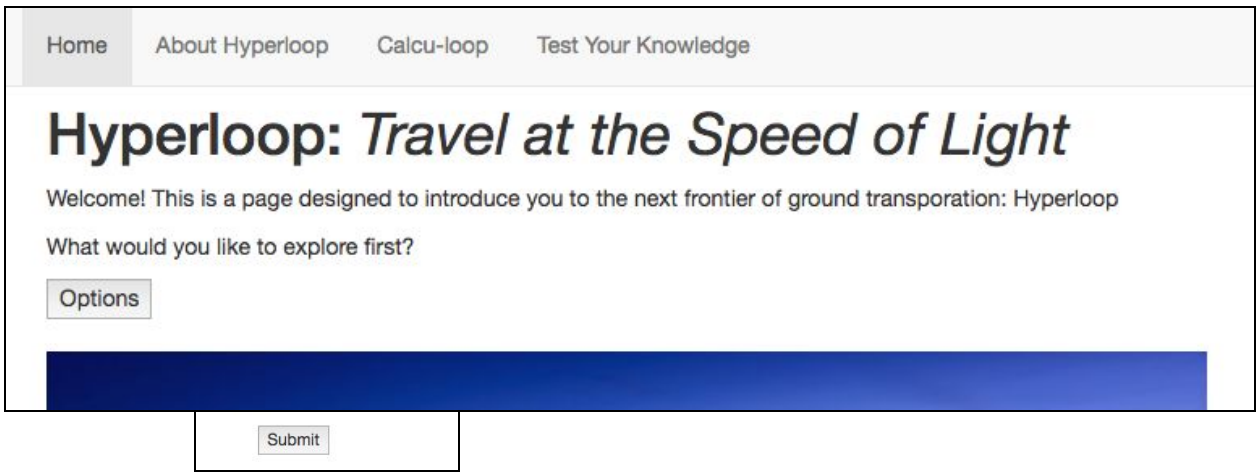


Jump to section

Even though there are only 3 questions in the quiz, it's still frustrating not to be able to jump to a specific question, so I added a back button to all three questions and still have the "Back to Home" button on the final question. I also made the "Next" button available immediately rather than showing it only after the first "Submit". I figure there's no need to force the quiz. They won't be graded on the quiz, after all.

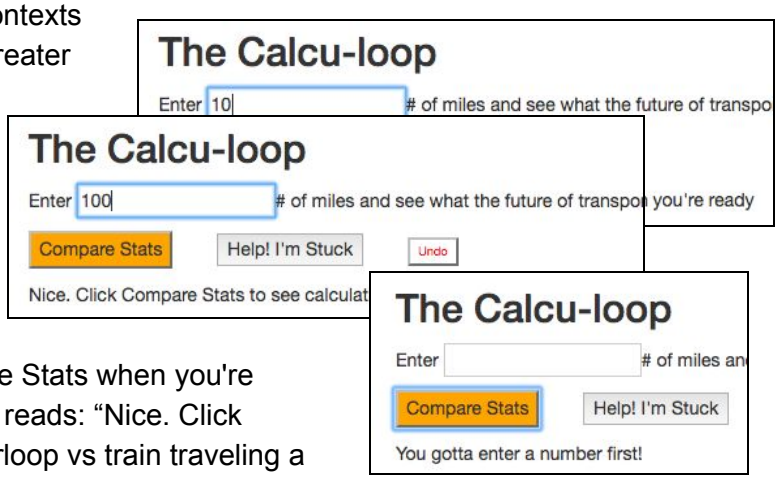
Overflow menus

I decided to leave the overflow menu even though the top nav bar provides sufficient information to navigate through the various sections of the site. I figure redundancy in this case adds clarity.



Action Context

My inline validation includes some action contexts that encourages the user to enter a value greater than 100. The best example of Action Context is when a site gives you immediate feedback about the strength of your password. In a similar sense, these prompts tell the user whether they are going too high or too low. If the entry is less than 100, the message reads: "Try an even bigger number! Then click Compare Stats when you're ready". If it's greater than or equal to 100, it reads: "Nice. Click Compare Stats to see calculations for hyperloop vs train traveling a distance of " + x + " miles".



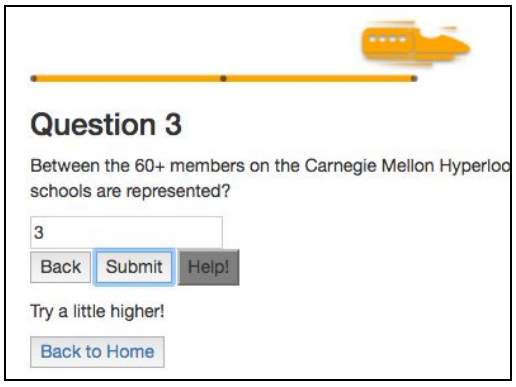
Input hints

Since the Inline Validation doesn't check for empty text fields, I added another validation that provides the input hint "You gotta enter a number first!" when the Compare Stats button is clicked, but the text field is empty.

Information Design Patterns

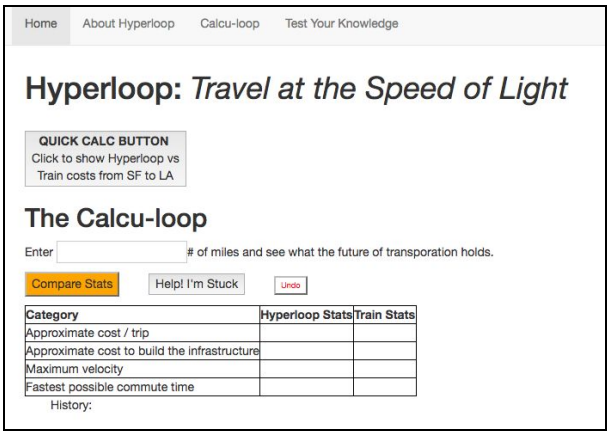
Hidden Content

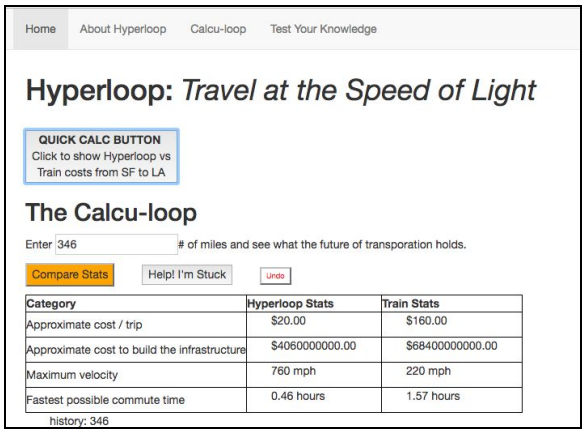
All answers are hidden throughout the quiz and are displayed depending on the users input. The Back to Home button on the third question is also hidden. This content is displayed only when the user submits an answer (although they can navigate back using the top nav bar). Finally, the content of the overflow menu is also hidden.



Featured Content

I moved the featured content button to the calculoop.html page. It's a nice way to jump into the content. I also added a nice visual surprise (not pictured below).





Coach

Previous implementation: Go on, enter a number! Then click Compare Stats. Or if you need more help, click the Help! button.

Adjustment: I deleted the previous version of the coach and replaced it with a mouseover input hint that is triggered when the user hovers over the input field. I felt the previous coach text was too wordy and not very helpful. That said, this is more

of an input hint than a coach as it's not explaining the two buttons. Another note about this screen shot is the Help! button launches a dialog box that provides the user with more guidance. See the "Wizard" section below for more details on that design pattern.

Pop Over

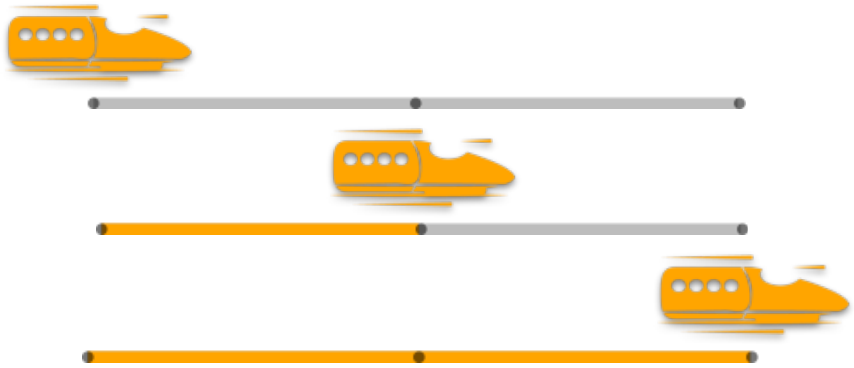
Removed. I converted the pop over to a wizard to allow the user to enter their input directly from the pop up. See the "Wizard" section below for more details on that design pattern.

Progress Design Patterns

No changes were made to any of the following design patterns since my last submission. I was satisfied with the look, feel, and functionality of these already. The only change I made to the quiz was to add a jump to section design pattern.

Completeness Meter

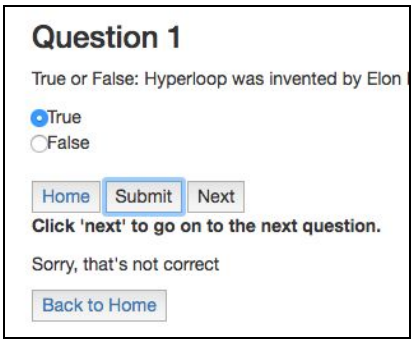
No changes since my last submission.
Pictured Below: I decided to use icons for each of the 3 questions in the interactive quiz rather than use numbers. It's a bit easier to see at a glance.



Implementation: Each completeness meter is a static image that is shown and hidden depending on the user's progress.
Future Improvements: The images could expand according to the size of the browser.

Next Steps

No changes since my last submission.
Pictured Right: The next steps text only appears on question one as the naive user should already be familiar with the UI by question 2 and should recognize what to do next.



Implementation: The next step is triggered upon the first submit and is written specifically for this event. It is then hidden with .style.display = “none” after the Next button is clicked.

Recommendations

No changes since my last submission.

Pictured Right:

4. If the user guesses a number over 5, they get the hint text: A little lower.
5. If the user guesses a number under 5, they get the hint text: Try a little higher!
6. If the user guesses 5, they get it right!

Wizard

No changes since my last submission.

1. Wizard dialog box that appears when users click the “Help!” button.
2. Per the class example, the value is then taken from the prompt and used as input for the input field.
3. The example here shows and initial number that was incorrect, followed by a number that will end up being too high. She’ll get there...