BHCI Capstone Project | Spring 2018

Authoring Tools for Easy Creation of Adaptive Tutoring

Chris Feng
Grace Guo
Siting Jin
Carolyn Zhou

Table of Contents

Project Overview

Executive Summary

Our team worked with the developers of the Cognitive Tutor Authoring Tools (CTAT) to conduct user-centered research on the software's barriers to adoption, in order to create a solution that makes both understanding the concept of cognitive tutors and learning to create them more accessible and approachable.

As a result, we created a newly designed website for CTAT that features a tutor gallery and redesigned tutorials for key features of CTAT. In this report, we present our final design solution with a high level overview of the design and the process that led to this solution. Our final deliverables for our clients include a coded website, a working minimum viable product (MVP) version of our tutorial, and documentation for maintaining the site and creating tutorials.

Clients

Vincent Aleven
Jonathan Sewall

Advisors

Robert Kraut
Steven Dang

Team

Grace Guo
Design Researcher

Carolyn Zhou
Design Lead

Siting Jin
Technical Lead

Chris Feng
Design Researcher

Client Needs

CTAT is a tutor authoring tool that already has a mature code base with extensive research behind it. There has been a number of tutors built with the tool and in use now. However, CTAT's user base is still fairly small and the tool has yet to be widely adopted among customers. Our clients want us to find the cause of the current low usage and improve the product to increase adoption of this tool in classrooms. At the same time, our client wants us to keep the research relatively open to explore on other possible problems or solutions.

Hunt Statement

Our goal is to increase the adoption of CTAT among instructors and instructional designers in higher education settings.

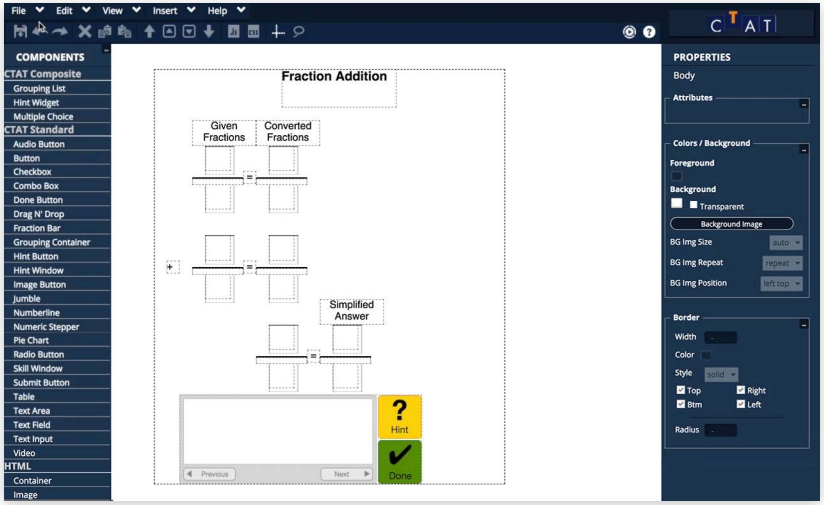
Acceptance Criteria

The project team will conduct user research with a wide range of potential users and submit research findings in a comprehensive report. Based on user research findings, subsequent prototypes will focus on increasing adoption of CTAT amongst one of these user groups. The project team will do its best to ensure a high quality final deliverable that significantly addresses client needs. While we cannot guarantee that the final design will be fully implemented in code, we will provide all prototypes and design documentations needed such that such an implementation can be executed past the completion of this course.

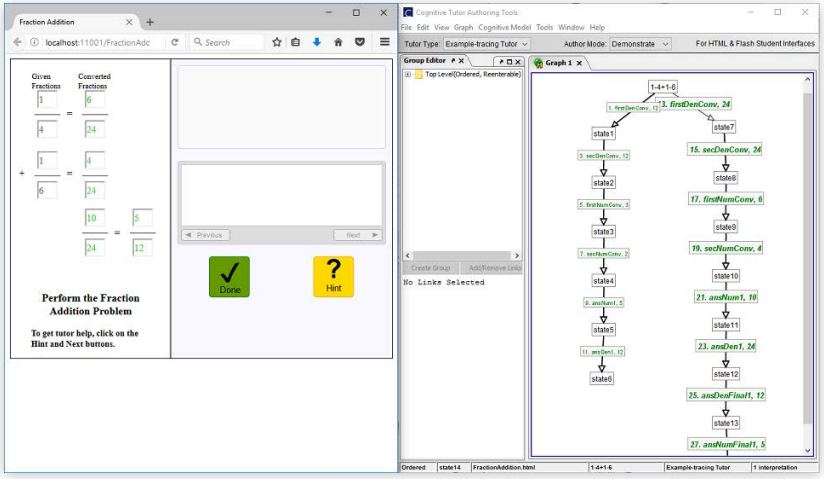
What is CTAT?

The Cognitive Tutor Authoring Tools (CTAT) is a tool suite for creating online cognitive tutors, developed by our clients, Vincent Aleven and Jonathan Sewall from CMU's Human Computer Interaction Institute. CTAT supports contextual help through customized hints and tracks students' progress to provide feedback as they work through a problem. Behavior graphs draw on cognitive models for intelligent behavior, and allow instructors to guide students to arrive at different solution paths.

CTAT can be integrated with LMS's such as Canvas, Blackboard, Open Learning Initiatives (OLI) and TutorShop. CTAT also supports logging data related to student-tutor interactions that can be used by instructors and researchers to analyze learning trajectories.



CTAT HTML Editor



CTAT Behavior Graph (Example-Tracing tutor)

Final Solution

User-Facing

- CTAT website
- Tutor gallery
- Tutorials

The user-facing deliverable is a redesigned website for CTAT, which aims to make the software more approachable and appealing to new users. The final iteration of our redesign is hosted at https://tinyurl.com/ybk787jr. The site hosts several features we worked on as well, including a tutor gallery and tutorials. We will provide further explanation on each deliverable.

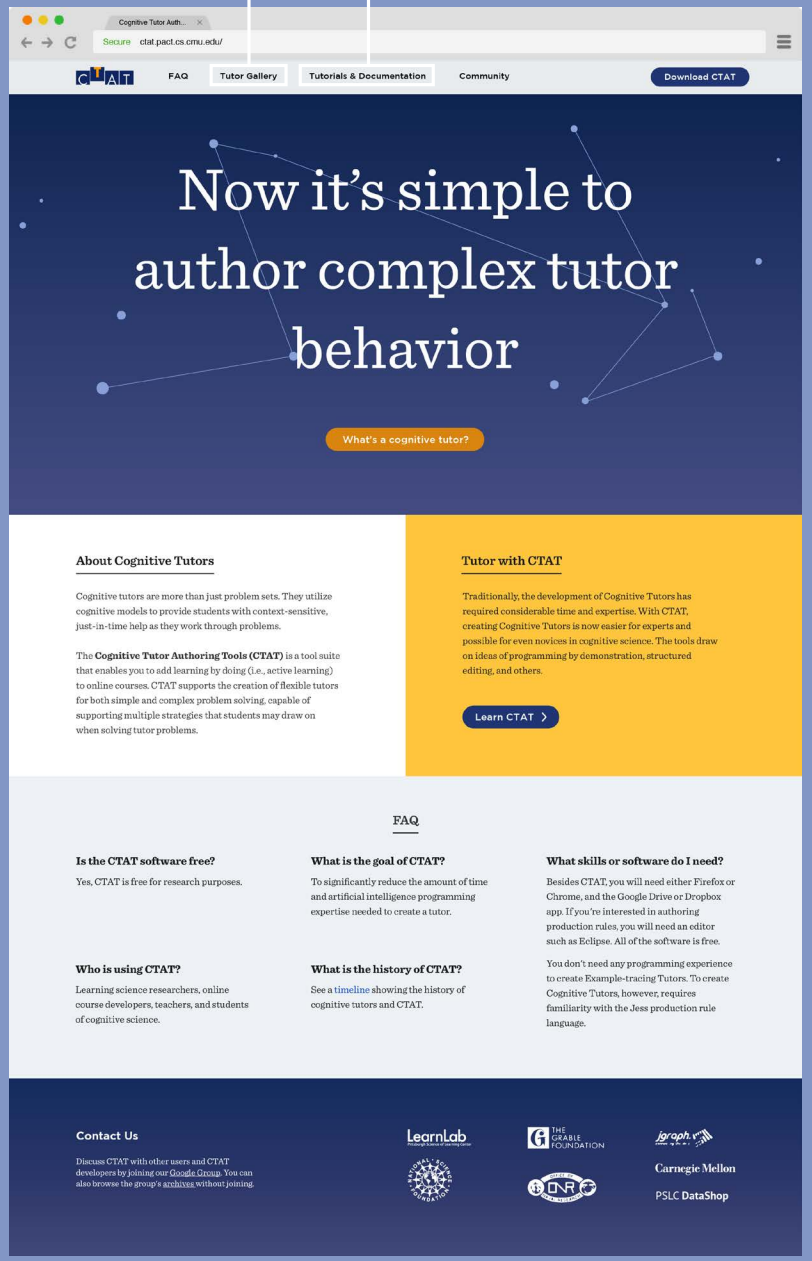
Client-Facing

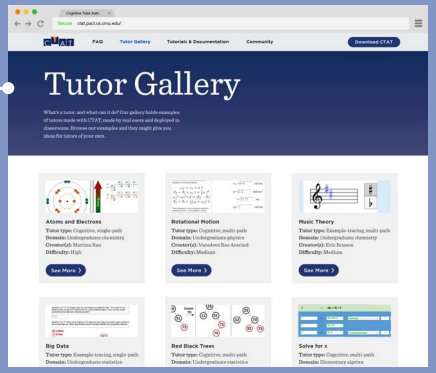
- Tutorial design documentation

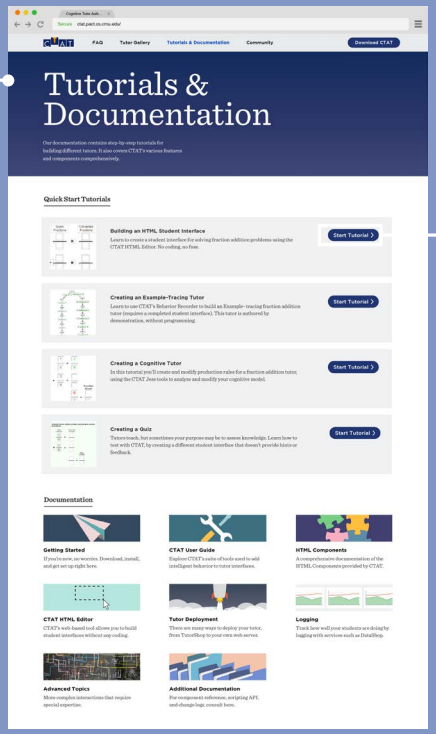
The client-facing deliverable is a detailed documentation of the guidelines and specifications for creating our designed tutorials. The purpose of this documentation is to enable our client to create more tutorials in the same fashion, and to maintain our current tutorials.

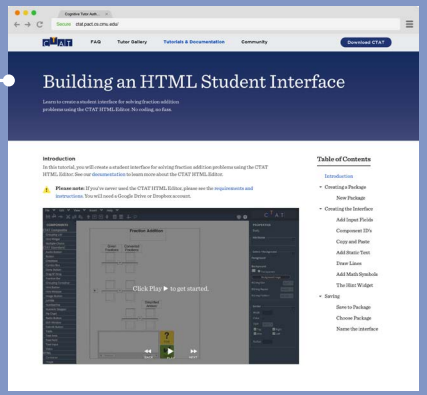
Final Solution

Site Map







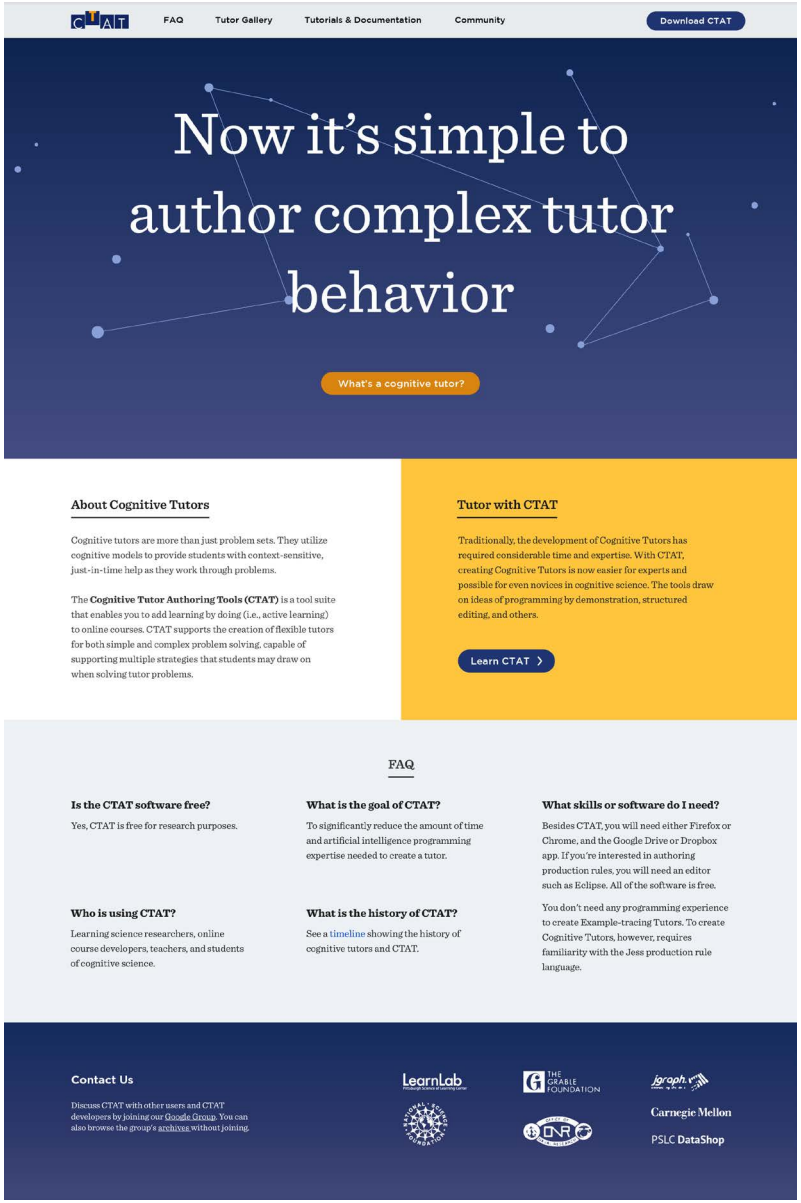


CTAT Website

The website aims to simplify conceptual explanations and use approachable language. The site is intended to serve as the main touchpoint by which users would learn about CTAT, so it should make the subject of cognitive tutors feel approachable and easy to understand.

The site's visual design aims to be modern, clean and soothing to increase the software's appeal. The cleanness also aids the simplification effort, removes visual clutter and calls attention to important aspects of the site. For example, the buttons on the home page are prominent calls to action.

The flow of the website is designed to make the tutor gallery and tutorials easily discoverable. Through testing, we believe that these are features are critical to converting new users.

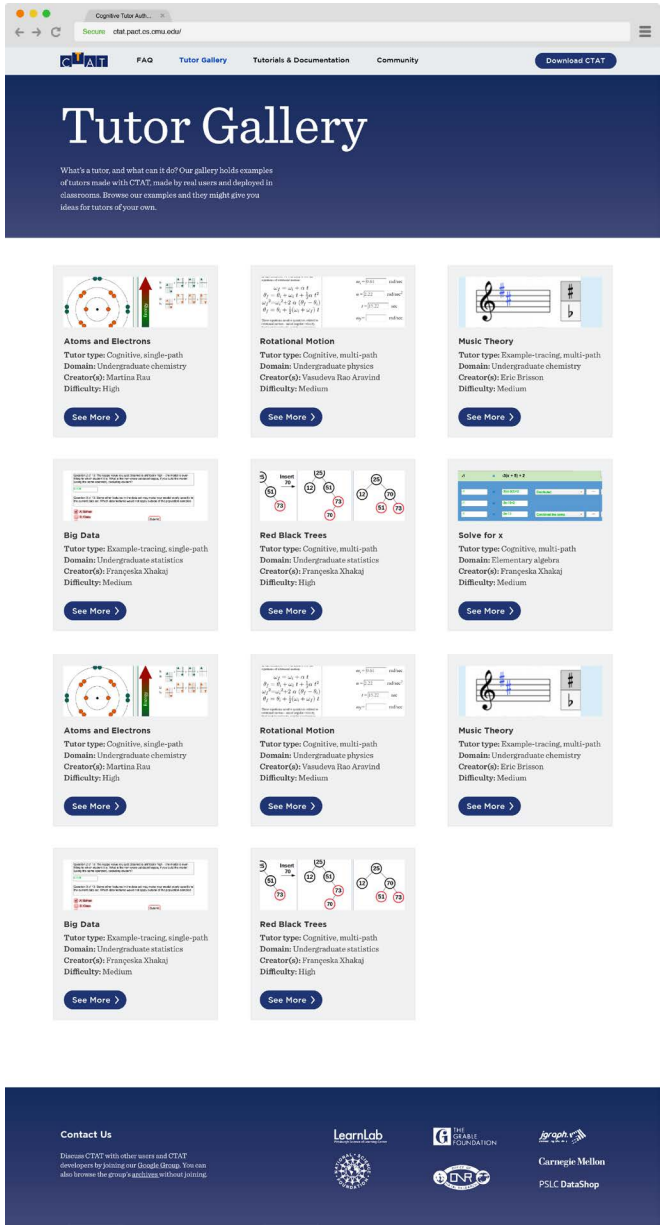


Tutor Gallery

The idea behind the gallery is to allow new users to see examples of real tutors made with CTAT. Clicking “see more” would bring visitors to a live or demo version of that tutor, though we did not prototype or build this experience.

The design of this page and its content have not been fine-tuned. Rather, we only asked for general impressions of the page shown, and users stated that **they thought it would be helpful to see concrete examples of how they could use tutors.**

Interestingly, users who had previously completed our tutorials expressed the desire to see examples of the tutor back end (the behavior graph) as well, in addition to the student-facing tutor interface.

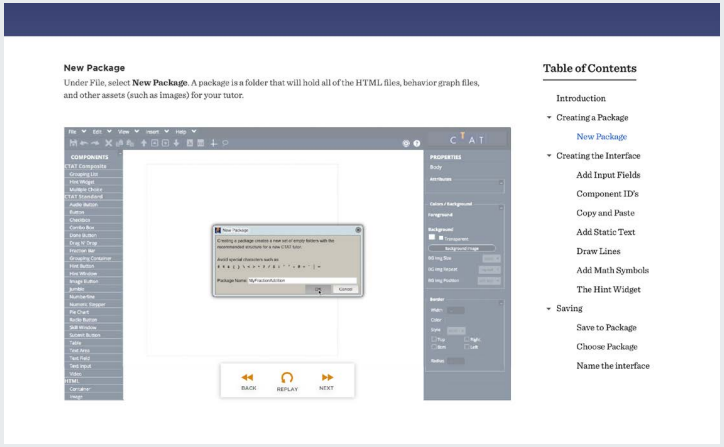


Tutorials

From previous research, we had identified tutorials as a possible point of intervention to pique interest in CTAT and thus increase its adoption. We found that sufficient understanding of intelligent tutor behvaior often left users impressed and curious about what else CTAT could do, and tutorials had the opportunity to give users that understanding.

We designed a tutorial that was approachable and easy to navigate, more so than CTAT's current text-heavy and cumbersome web pages. By using less text, users appeared more willing to dive into the content. Including navigation through the icons and the Table of Contents also made the tutorial feel more parseable.

The tutorials use both text and animation to walk users step-by-step through some process. They feature minimal text with a call to action and a conceptual explanation or rationale for what that particular step is accomplishing. The animation helps to show exactly what the text is describing. We iterated through several rounds of testing to determine an appropriate way to write the text, break up the steps, and use animation. The design guidelines can be found in our tutorial documentation (included in this report).



Two states of the same tutorial page

Research

Research Methods

We asked users to perform **think alouds** to test our tutorials for a fraction addition tutor and the website where the tutorials live. The task was to follow the tutorials however they wanted and to build the fraction addition tutor from beginning to the end. The participants navigated through the web pages to the tutorial and had the chance to get through the high-level introduction to CTAT and cognitive tutor while browsing the web pages.

In addition to usability testing, we also conducted **heuristic evaluations** among ourselves and sought **informal critique** on our designs.

Test Participants

Our users were **5 CMU students** who have never used CTAT or other tutor authoring tools. One of them has been a teaching assistant (TA) for three different of math courses. Another has been a TA for an artificial intelligence course. The rest have not been in teaching positions before.

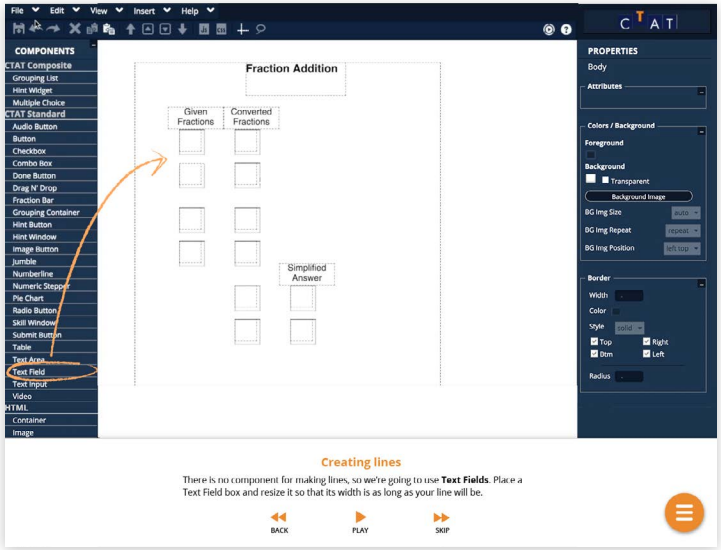
Key Findings

1. **The tutorials are easier to follow and improve learning and retention.**

   The combination of step by step short gifs and text instructions make it easier for users to follow the tutorials. The skip, replay, and back options give users more flexibility to complete the tutorials at their own pace. The important parts of the text instructions were bolded to draw users' attention and facilitate the "learn by doing" experience.

   The gifs allow users to locate where components are so that they don't have to make an extra effort to look for certain components on the editor screen. The step-by-step format of the tutorials make it easier for users to understand what is happening by providing more granular instructions. The gifs helped clarify what users were going to accomplish for each step as stated in the text instructions so users were able to follow the directions.

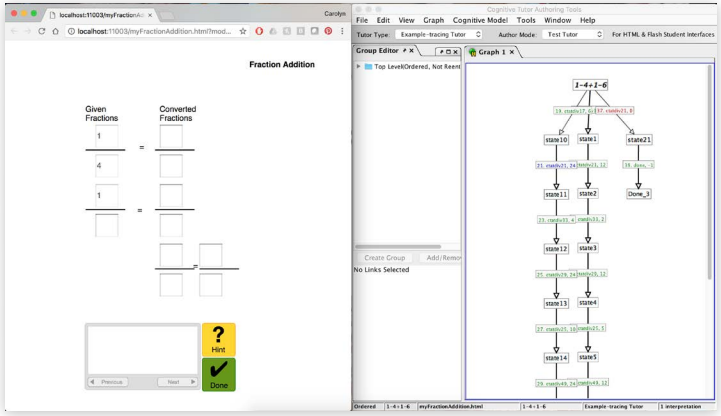
   When asked to plan out their own tutors after finishing the tutorial, users recalled specific CTAT components by name that they would use for particular pieces of their imagined tutor interfaces.



Layout from a previous tutorial iteration that we tested.

"The step by step videos were helpful. The animation made it so that you can see what is happening."

"Once I got to the [Behavior Graph tutorial], I started to see how the student interface and tutor behavior fit together."



One test subject's finished interface and behavior graph.

2. **Users are able to gain a better understanding of what CTAT does.**

After completing the tutorials, all of the participants were able to understand the philosophy of CTAT and envision how CTAT might be used in real world education. Most of the participants were new to CTAT and were not familiar with CTAT concepts such as behavior graphs and component ID's before watching the tutorials. As they walked through the tutorials, they were able to understand how the HTML interface and the behavior graph fit together. The tutorials also help users understand CTAT's potential, which made them curious about other features such as matching multiple inputs without explicitly creating all the possible paths.

Additionally, our participants were able to envision how CTAT might be used in their classes. Most of the users believed that CTAT would be very helpful for creating tutors that had procedural solutions such as statistics problems.

Future Considerations

Current Challenges

At the end of this project, there remains a number of challenges that limit the scope and effectiveness of our proposed solution.

Most significantly, within our limited time frame **we were not able to fully test the effectiveness of our tutorial redesign on CTAT adoption**. While our user testing has shown that the site redesign and the new tutorials made it easier for novice users to understand CTAT's features and capabilities, there remains the possibility that this may not translate into increased adoption of CTAT over time.

Additionally, we could only design, test and implement a **limited number of tutorials** within the time frame of this project. There are a number of additional advanced topics such as dynamic interfaces and grouping that users are frequently interested in and may thus want to learn about in a tutorial.

Next Steps

Working off the challenges discussed, some possible next steps for this project include:

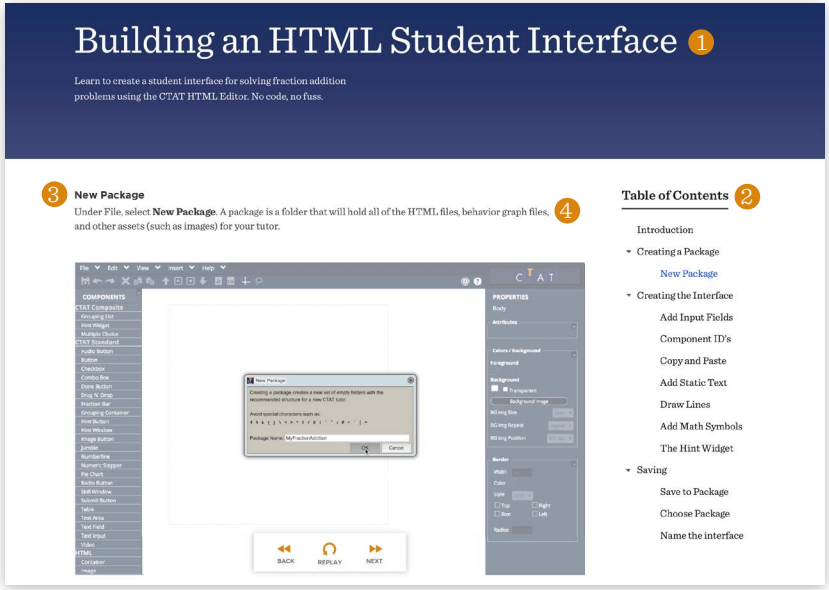
1. Deploying and evaluating the long-term effect of the new website and its features on CTAT adoption. A good indication of tutorial effectiveness could be a correlated upward trend between tutorial views and CTAT users and tutorials created. Successful tutorials should give novice users the confidence and competence to create tutors for their own use.

2. In the long-term, it would also be useful for a greater range of tutorials to be created. Users who are considering CTAT for its superior flexibility and capabilities may want to learn how to use a wider range of CTAT's features, including topics that fall under advanced features. Tutorials that can make these advanced topics more learnable will definitely appeal to these users and help them create effective tutors for a diverse range of domains.

Tutorial Documentation

Table of Contents

Page layout for HTML Editor tutorial



Design

Layout

- Text should be placed above the gif so that it is more prominent.

  > Text is easier to skim than video, but animation may help make a step easier to follow if there is confusion.

- Keep text, navigation elements and gif frame in a consistent place.

Table of Contents

- Show which step the user is currently on with a highlight.
- Make the TOC accessible to the user at any point. In this design it is always accessible from the right hand side.

Style Guide

The style guide (next page) applies to the design of the website as a whole. The web pages where the tutorials live follow these guidelines as well.

Main Page Title — Sentinal Book 104pt

1 Small Page Title — Sentinel Book 60pt

2 Section Heading — Sentinel Semibold 21pt, underlined

3 Paragraph or Card Title — Gotham Bold 16pt

Button Text — Gotham Medium 16pt

FAQ Question — Sentinel Bold 18pt

4 Body text — Sentinel Semibold 16pt/26pt

Footer body — Sentinel Book 14pt/18pt

Blue Gradient Top: #132B5E Bottom: #404978	Blue (buttons) #203573	Orange (buttons) #D9840F
Light Gray (nav bar) #E8EBED	Light Gray (background) #EDF1F5	Light Orange (background) #FFC53D
Light Gray (cards) #F0F1F2	Link Color #0056AD	Font Color #333333

"New Package

Under File, select **New Package**. A package is a folder that will hold all of the HTML files, behavior graph files, and other assets (such as images) for your tutor."

"Component ID's

Set a unique **component ID** for each component. Best practice is to give them mnemonic names to help you identify them, like "textInput1." The component ID's will come into play in the Behavior Graph, and it is important that they are distinctive."

"Add Input Fields

Add a **Text Input** component to your interface. Text inputs are fields in which students will be able to enter their answers."

"Add Static Text

Add a title in your interface using a **Text Field** component. The text you enter with text fields will be static. You can use these to place titles, labels, or problem statements."

Defining Steps

Granularity

- Each step should show a single change in interface, or users start to feel overwhelmed and unable to follow.

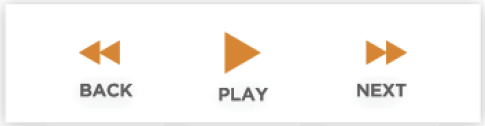
  When testing with a video tutorial, users would pause after about one or two changes to imitate the changes before returning to the tutorial.

Repetitive Steps

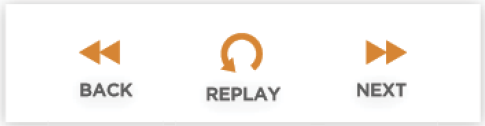
- If the component or feature has been demonstrated before, remind the user what it is at least once.
- Users prefer when repetitive steps are fastforwarded, as long as there is sufficient explanation that the tutorial will be fastforwarding or skipping ahead.

Table of Contents

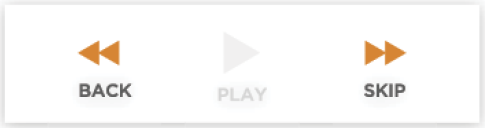
Before gif plays



After gif plays (page stays on current step)



While gif is playing (user can still skip or go back)



Tutorial Navigation

- Allow users to navigate to previous and next steps
- Allow users to replay a step
- Allow users to jump to any step at any time, using a table of contents that lists all the steps
- In the table of contents, group the steps into sections which should be collapsible (with an indicator such as an arrow icon) for cases in which the TOC gets long

Copy writing

Title

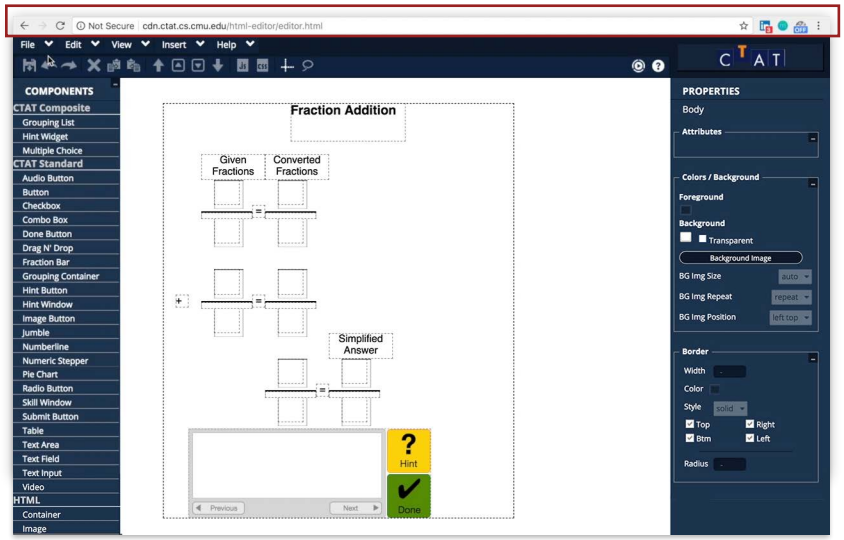
- Title should name the relevant element or component being used
- Avoid making the title a full sentence with an action call, because users will be tempted to skip reading the text underneath which may include useful information
- Titles should be descriptive enough to stand on their own, as they are used in the Table of Contents. You may need to test for the best title that makes sense to users.

Description

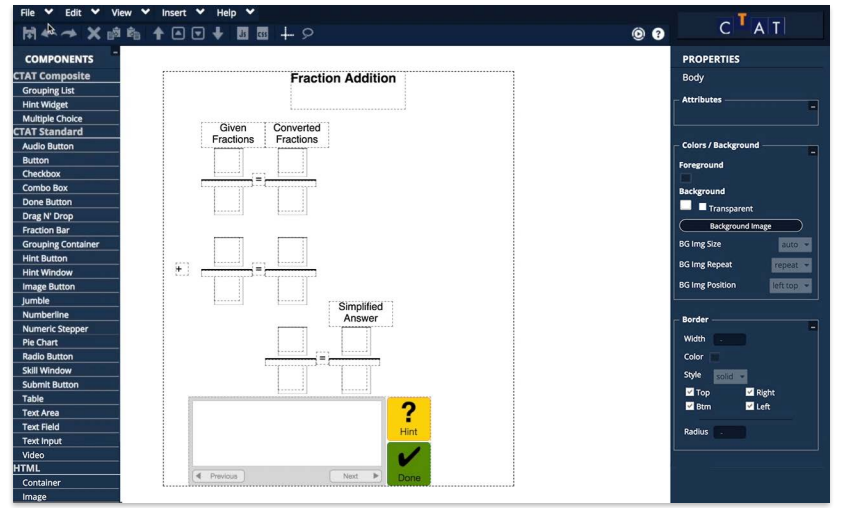
- Have a clear call to action for each step.
- Explain the component being used in the step.
- **Bold** the most important element or component being addressed for easier skimming
- Provide rationale/a conceptual explanation: What is accomplished by completing this step?

    For example, when asking users to provide a unique ID for components, explain that they need to be unique for the Behavior Graph and this could also be a mnemonic device

With header ✗



Without header ✓



Animation (gifs)

Recording Videos

Technology: QuickTime Player (Mac)

- Use any screen recording software to record your steps as you complete the usual process of creating a tutor
- Any parts involving reading, typing, or locating something, i.e. entering Hint Messages or finding something on a menu, may require longer-than-normal pauses.
- If possible, crop the recording so that your browser headings, docks, menu bars, etc. are not visible. If this is not possible with you recording software, you may do this step in editing (next page).

Editing Videos

Technology: iMovie (Mac), Adobe Premiere

- Split the videos into separate steps using any video editing software. These steps should correspond to the discretized steps from "Defining Tutorial Steps."

  Alternatively, you may record each step as a separate video.

- For each video, adjust the speed if necessary.

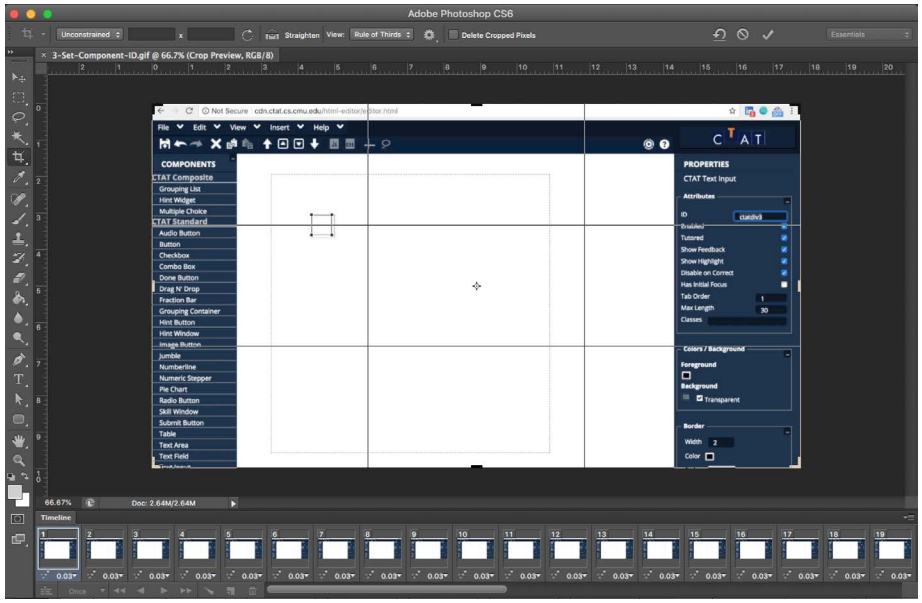
  Steps that involve manual work that don't require explanation, such as copy and pasting multiple components or demonstrating solution paths, may be sped up.

  Steps that involve reading, typing, or locating something, i.e. entering Hint Messages or finding something on a menu, may require pauses or slowing down.

- If your recording contains headers or other visuals that are not part of CTAT's interface, crop them off. (It's also possible to crop in Photosop; see next page.)

- Save and export videos as mp4/mov.

Photoshop crop tool selected, with the Timeline panel open.



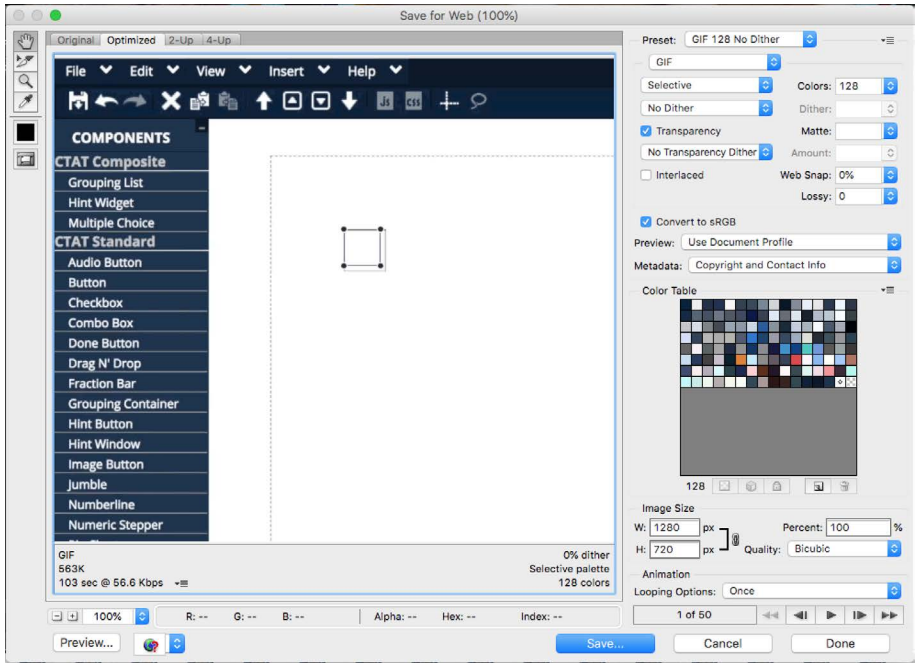
Creating Gifs

Technology: Adobe Photoshop

For how to make gifs, visit https://helpx.adobe.com/photoshop/how-to/make-animated-gif.html

- There should be no more than 300 frames for each gif
- Each frame should appear for a duration of 0.02/0.03 seconds
- For frames with an important visual (e.g. demonstrating where to click on a menu), you might adjust the duration for 0.5-1 seconds
- Loop setting should be set to Once. This way, in the tutorial the step will end on the last frame of the gif.

Export window



Gif Specifications

- Loop should be set to Once.
- For prototyping in InVision, each file must be less than 10MB. Otherwise, there are no constraints to size.
- Color settings should be 128 no dither.

Appendix

Methods for Initial Research

Literature Reviews

Think Alouds with Novices

Competitive Analysis

Expert Interviews

Literature Reviews

We started our research with a literature review of the history of CTAT and intelligent tutoring systems. From here, we were able to gain an understanding of the unique value that CTAT brings to the ecosystem of education technologies.

1. **Unlike many other software, CTAT provides students with greater support and guidance in both situation of accurate and inaccurate problem solving.** Some of these features include the ability to provide error feedback messages and interfaces that adapt to state of problem solving.
2. **Additionally, CTAT is able to support dynamic solutions to a problem.** CTAT tutors allow the specification of variable and interdependent steps, partial ordering and optional repetition of steps, and input substitution. Taken together, CTAT allows the creation of multiple correct solution paths for a single problem, thus managing the often diverse methods students may take to solve a single problem.
3. **To supplement the "inner loop" described that captures multiple solution paths, CTAT has an "outer loop" that makes pedagogical decisions.** The outer loop tracks student knowledge growth through a learner model and selects the appropriate problems that can best help a particular student master a certain knowledge component.

Competitive Analysis

We looked at 4 other education technologies: ASSISTments, Schoolhouse, Qualtrics and Oppia. These platforms target a wide range of users, from teachers to businesses and the general public. With the except of Qualtrix, they tend to serve math, technology and other STEM fields.

1. **All of the above products support question creation through fixed templates.** Users can select from a variety of predetermined question types, such as multiple-choice, short answers, checklists and others. These templates greatly improve the ease of authoring problem sets. Conversely, the amount of customization users can apply to a each template is limited. There is almost no way for question types and interactions to be designed from scratch in any of these products.

2. **Feedback is only provided after question submission, if at all.** ASSISTments and Oppia provide feedback to students at the end of the answer, often to let them know whether their response was right or wrong. Qualtrix, being a primarily survey software, does not tend to provide feedback, although it is possible to do so. Schoolhouse does not provide digital feedback, as problem sets are meant to be printed out, so feedback is provided by instructors in the traditional manner of graded worksheets.

3. **To supplement the "inner loop" described that captures multiple solution paths, CTAT has an "outer loop" that makes pedagogical decisions.** The outer loop tracks student knowledge growth through a learner model and selects the appropriate problems that can best help a particular student master a certain knowledge component.

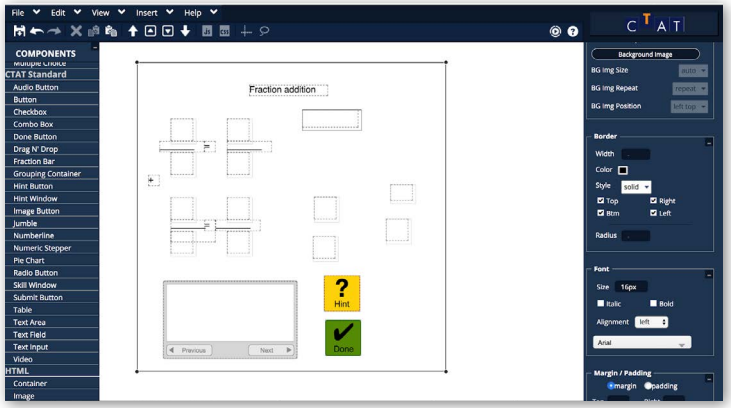
4. **Some of these products offer unique features that supplement the user experience.** Qualtrix provides real-time online collaboration, allowing multiple individuals to work on a single question set. Oppia, on the other hand, has a user community that can provide feedback for question sets and resolve technical difficulties. In terms of the interface, Schoolhouse resembles Microsoft Word, which potentially makes the program easier to intuit for novice users.

Think Alouds with Novices

We conducted 2 think alouds with CTAT novices. They were given a picture of a final tutor interface and asked to use the CTAT HTML5 editor to recreate the interface as closely as possible. One of the participants was provided with the CTAT tutorial while the other was asked to freely explore the editor. We were primarily interested in how intuitive the current interface is, and placed less significance on stylistic differences.

1. **Participants liked the drag-and-drop feature of the HTML5 editor.** Both found that adding components to the canvas was fairly straightforward. The final interfaces created also appeared very similar to the picture provided. Additionally, the participant who was given the tutorial was able to follow all of the steps without any major confusion or obstructions.

2. **The organization of the components toolbar needed some improvement.** Participants found it too cluttered, with the components organized unintuitively. There were also some options that participants did not immediately understand the function of. Some examples include radio button, jumble and fraction bar.



Finished HTML interface of one user tester.

3. **Having more geometric primitives in the toolbar could be useful.** Given that the final output is an interactive graphical interface, participants felt that there should be the options of creating diagrams more complicated than circles and lines. Currently, the creation of the fraction line felt like a 'hack' and was a struggle without the tutorial.

4. **Participants were unclear about where to seek help.** There was no obvious place to get additional information about specific tasks or components from within the HTML5 editor. Additionally, when the user made an error, they expected the editor to detect and highlight the error or to provide hints and suggestions.

Key Insights

1. Users misperceive CTAT's capabilities.

To novice users, it is not immediately obvious what CTAT is best suited for, and CTAT may seem more frustrating when it is not used for the right task. As Lorelei Walch of the CMU Eberly Center explains, CTAT is most suitable for STEM subjects where answers are clearly declared, and "tricky for humanity courses because a lot of answers are acceptable [and there are] no clearly defined solutions.

Additionally, Judy Brooks, director of the CMU Eberly Center, shared situations where faculty think they need to try new technology such as CTAT and use "technology for technology's sake". Often times, CTAT is not what they are looking for, and users who try to apply CTAT to problems it is not suitable for may come to see the tool as being less useful or intuitive. As such, it seems that new users need to be better informed about CTAT's capabilities and when CTAT is appropriate to meet their needs.

2. CTAT Incurs a high cognitive load.

Better integration with existing learning management systems is necessary to reduce the cognitive load of students and instructors. As Judy Brooks, director of the CMU Eberly Center describes, "...we don't want students to be focused on juggling tolls, as opposed to completing an assignment." When various tools become accessible through a single learning management system, this lowers the cognitive load of students and helps them pay more attention to class content rather than keeping track of the various tools they need to use.

Time commitment is the greatest concern for instructors when choosing a tool to use. Most instructors have multiple competing priorities of which teaching is only one of them. As such, when creating material for classes, instructors take into consideration the time they need to put in, including the time taken to learn a new tool. Professor Iliano at CMU mentioned that despite being interested creating CTAT activities for his online Computer Science course, he considers it to be a side project because it was too much of a time commitment.

Time considerations also mean that instructors want CTAT created content to effectively serve large numbers of students. Instructors are more willing to put in the time to learn a new tool if more students benefit from it. Professor Martina Rau at University of Wisconsin-Madison is a CTAT power user who has created a range of Chemistry tutors but does not use CTAT in her own classes. This, she explains, is because her classes rarely have more than 10 students, making it not worth her time to create CTAT tutors for such a small class.

Given the above concerns, there is therefore a need to redesign CTAT and its accompanying documentation such that novice users can both learn and use CTAT in a short amount of time.

3. There is insufficient support for novice users.

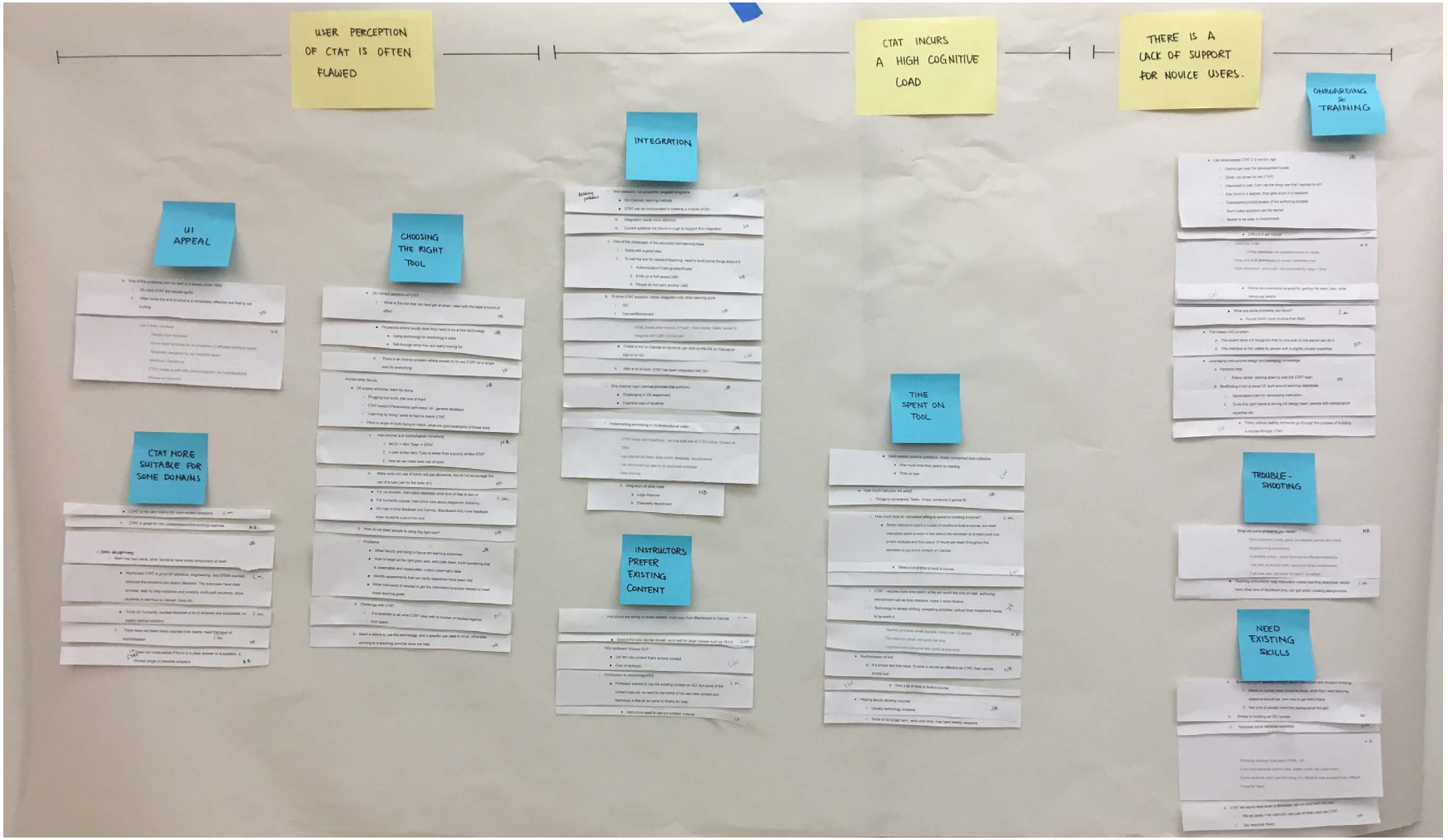
Novice users are often unaware of how they can troubleshoot and resolve problems, and users inevitably run into difficulties when using a tool as sophisticated as CTAT. The lack of comprehensive online documentation and Q&A platforms creates a substantial barrier for users trying to recover from errors. Additionally, despite an active Google group existing, the forum is not well advertised and many novice users are simply not aware of it. Professor Aravind of Clarion University shared that when first learning to use CTAT, he drove down to CMU to sit with our client to resolve his issues. His story is indicative of how difficult it might be for other novices who may not live so close to Pittsburgh.

The current onboarding and training process is also not comprehensive. Although CTAT is an extremely powerful tool, its lack of tutorials leaves much of its potential undiscovered by novice users. Lorelei Walch, an instructional designer at the CMU Eberly Center, found the online tutorials and documentation useful only for getting a basic idea. She shared that seeing someone go through the process of building a course through CTAT was most effective in getting her to understand how the separate parts of CTAT (editor, interface and behavior graph) came together.

Finally, CTAT novices may feel discouraged believing that they need prior skills to utilize CTAT fully. In particular, multiple interviewees mentioned that an understanding of HTML, CSS, and javascript was necessary to create more complex interfaces and interactions. Professor Aravind, who previously had basic skills in HTML, learned javascript later to better use the tool. Professor Rau also mentioned that the more elaborate style templates she used were not supported by drag and drop interfaces, thus her PhD students had to learn to code before they were able to assist her with her CTAT work. Additionally, she found that some of the documentation for CTAT seemed targeted at people with programming experience, making it less accessible to individuals without these prior skills.

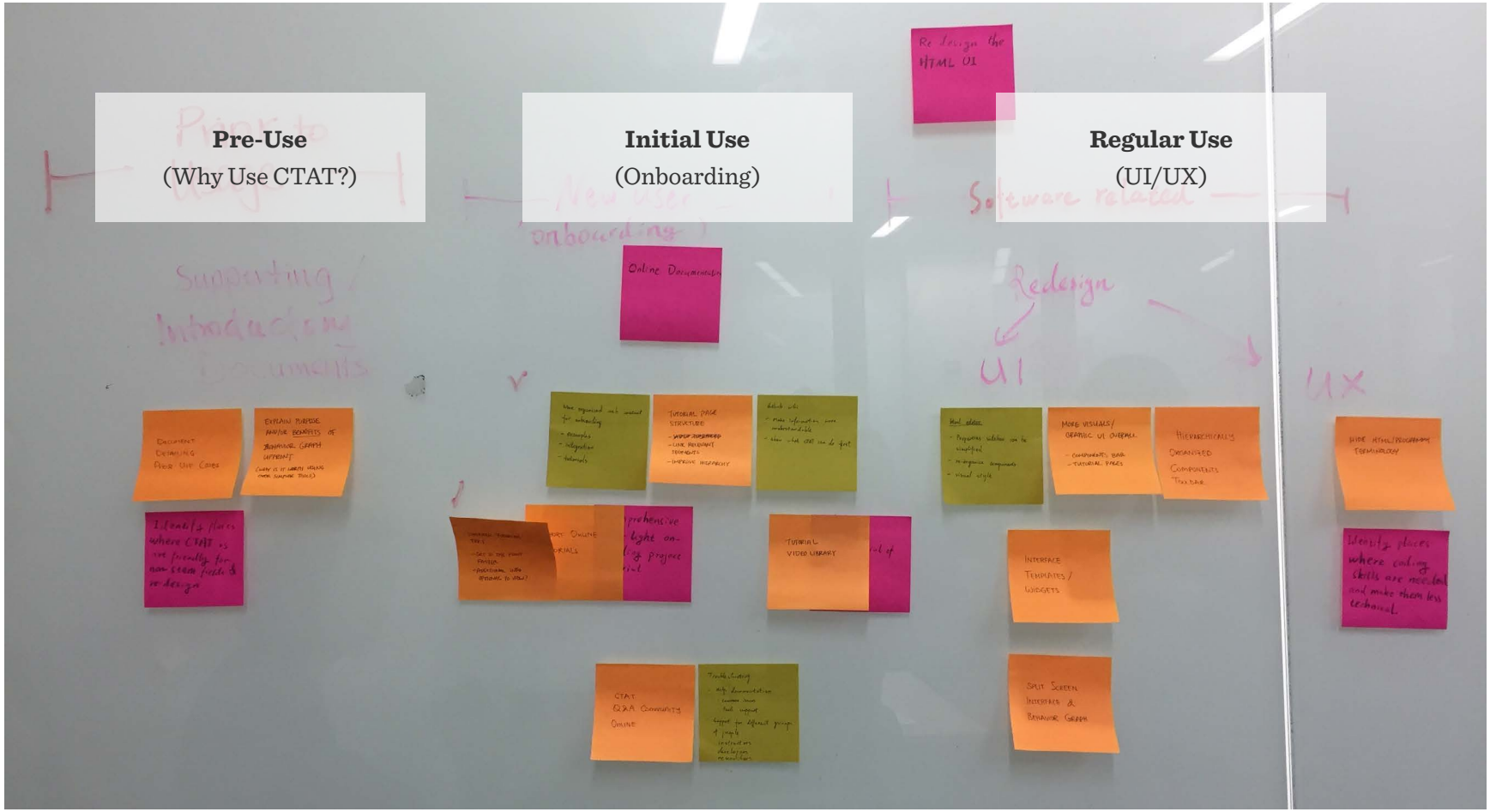
Although CTAT is an extremely powerful tool, much of its potential remains undiscovered by novice users.

Affinity Diagram



Lo-Fi Design Goals

We summarized the key insights from our previous research that explain barriers to adoption of CTAT. We then categorized these ideas into three phases of use, each of which addressed one of our research findings and corresponding design goals.



Pre-Use

During our research, we found that some people misunderstood the capabilities of CTAT. Since it is such a complex tool, the creation of simple problem sets would take less time with other, simpler tools. However, CTAT is more valuable in showing multi-path solutions, as well as providing customized, step-by-step feedback. Therefore, our first design goals was to **help users understand the significant added value of CTAT** by providing them with a wide range of supporting materials, which might include case studies of previously made tutors, example questions and use cases, etc.

Initial Use

Many novice users have a hard time getting started with CTAT because the current Github wiki tutorials are lengthy and text-heavy. As a result, walking through the steps for creating a single question may take a long time for new users. Therefore, we wanted to **design tutorials to be more approachable and easier to follow**. Some design ideas included designing a more organized, comprehensive online documentation and replacing the current text tutorials with video- or image-based tutorials.

Regular Use

When using CTAT to create tutors, users sometimes find it difficult to do what they want. The editor interface uses terms that are difficult for users to understand and find, particularly in the components and properties side bars. New users may be aware of the capabilities of CTAT, but find that the interface is not flexible enough and may require code to edit sufficiently. Therefore, we aimed to **make the user interface and user experience more intuitive** based on our feedback from interviews and initial user testings.

Lo-Fi Testing Methodology

Because each goal constituted a huge area to explore, we decided to focus on **1) tutorials for onboarding** and **2) the user interface for creating tutors.**

Our user testing was a think-aloud process, divided into two sections.

In the first section we tested the modality of tutorials. We showed one of our participants the video tutorial and the other participant the screenshots tutorial that take them through the steps for building a multiple choice question, including both the student interface and the behavioral graph.

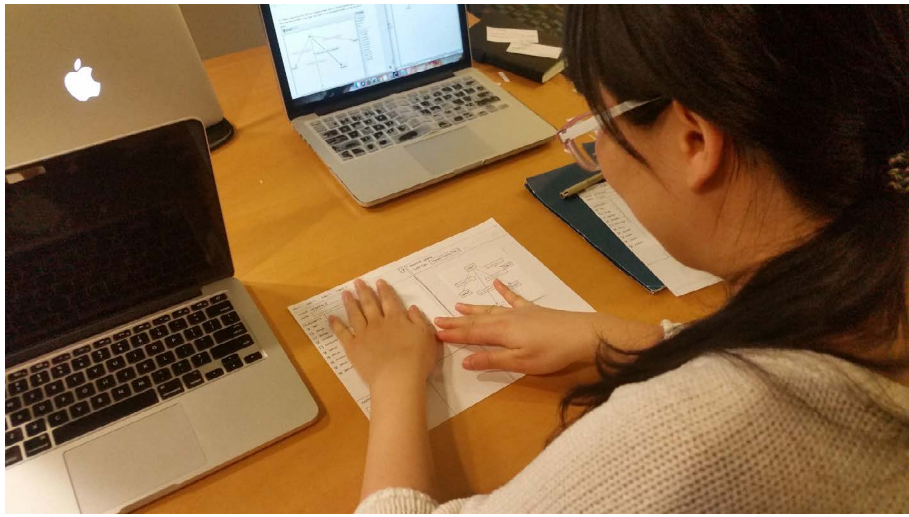
In the second section, we proceeded to test our interface redesign. We had two paper prototypes with slight differences in layout. Each participant was shown one prototype each. Participants were asked to reproduce the multiple choice question using the paper UI prototype. Our prototypes were modular, and interacting with them will cause additional interface sections (e.g. menus, pop-up windows) to be layered on top. The HTML interfaces and behavior graphs that the participants "created" were drawn directly onto the paper.

Image: Participant interacting with paper prototypes while following one of our tutorials.

User Profiles

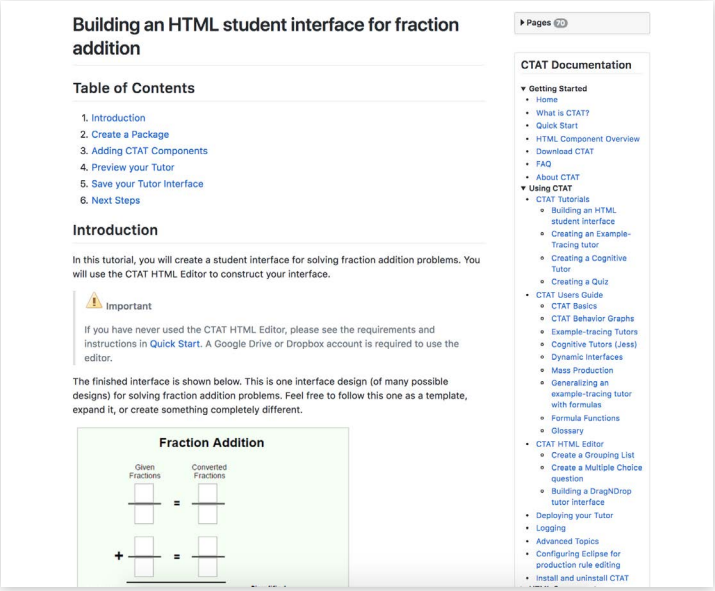
Participant 1 is an Educational Psychology PhD student. She started to use CTAT this semester and learned the concepts underlying the behavioral graph and cognitive tutor in the Personalized Online Learning course. She had some difficulties using CTAT initially but figured out the problems herself.

Participant 2 is a graduate student at the University of Pittsburgh who recently created some CTAT tutors. She started to use CTAT at the beginning of the fall semester and focused on the back end side of the system. She knows some programming languages such as Python and Java but is not familiar with HTML or Javascript.



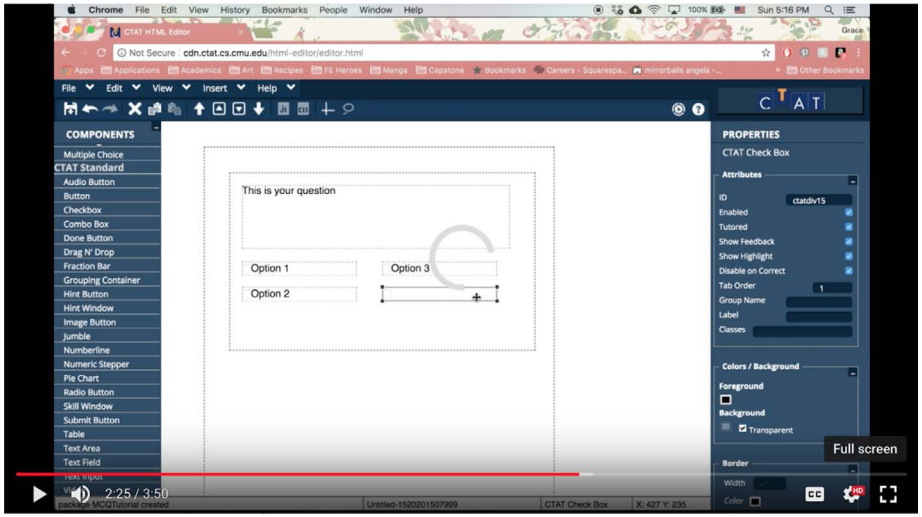
Tutor Modality Prototype

In order to support novice users starting out on CTAT, we considered making prototypes more approachable and easier to use. Our initial user-testing revealed that the existing tutorial pages on the CTAT wiki are too text-heavy, containing large sections of explanations that distracted from the steps needed to complete a task. As one of our participants said, “The instructions had a lot of text so you might miss helpful tips or illustrations.”



CTAT’s HTML interface tutorial today.

To address these issues, we prototyped new tutorials in two modalities. The new tutorials adapt instructions for creating a multiple-choice question from the existing CTAT wiki. One of the tutorials presents the information as a step-by-step video with a voiceover, while the other uses screenshots with light text. Both modalities aimed to be more approachable by reducing the amount of text. During our user testing sessions, participants were provided with either the video or the screenshots tutorial, and asked to create a multiple choice question using the current CTAT software.



Modality: Video

The video tutorial uses screen capture and voice over to take viewers through the step-by-step process of creating a multiple choice interface and behavior graph.

Feedback

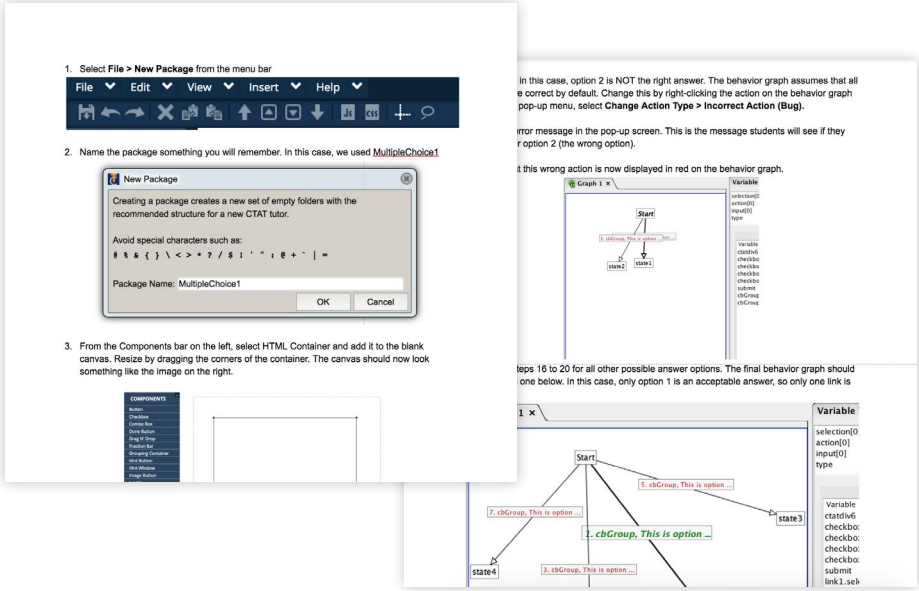
Participants who used the video tutorial found it more efficient than the original tutorial. The screen capture made it easy to locate components being used. However, the video format provided less explanation for why certain steps were necessary. Furthermore, participants followed along on their own editor and pause the video every 30 seconds or so in order to complete a step before moving on.

Modality: Screenshots

The screenshot tutorial (see appendix) uses light text and images for each step. It is more similar to the existing tutorial format on the CTAT wiki.

Feedback

Unlike the video tutorial, participants who used this tutorial liked that the text format was able to provide short explanations for some features of CTAT. On the other hand, information from a screenshot is limited, making it harder to locate certain things. This was exacerbated when we used technical terms that users didn't recognize, like "state" or "action."



User Interface Prototype

The original user interface has a number of usability problems. Based on feedback and data collected throughout our research process, we identified specific problems with the original UI, including the HTML components bar, the properties panel, and the lasso tool, among others.

1. The Components Bar

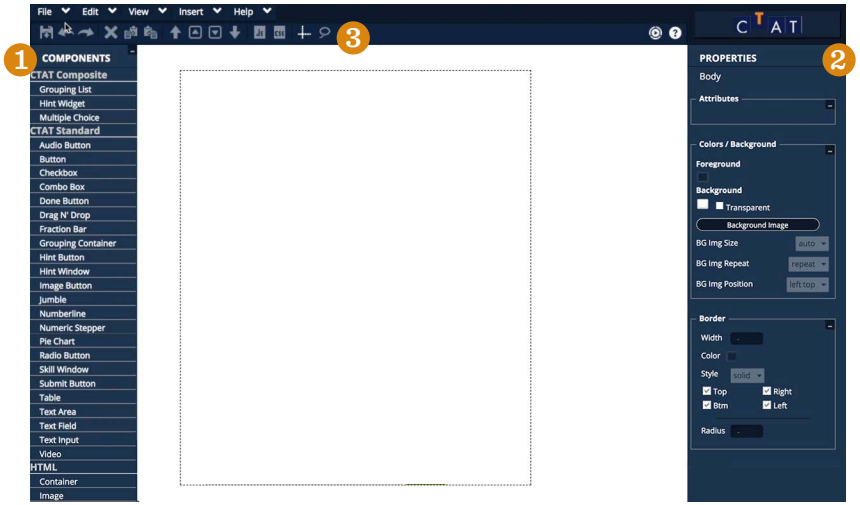
In the original interface, the items in the HTML components bar are not organized on a way that is easy for users to locate any specific one. Some components with similar properties are placed apart, while others have very similar names (i.e. Text area, Text field, Text input) without clearly distinct functionalities. Taken together, users usually have trouble locating the components they are looking for, and do not intuitively understand the features and usage of each component.

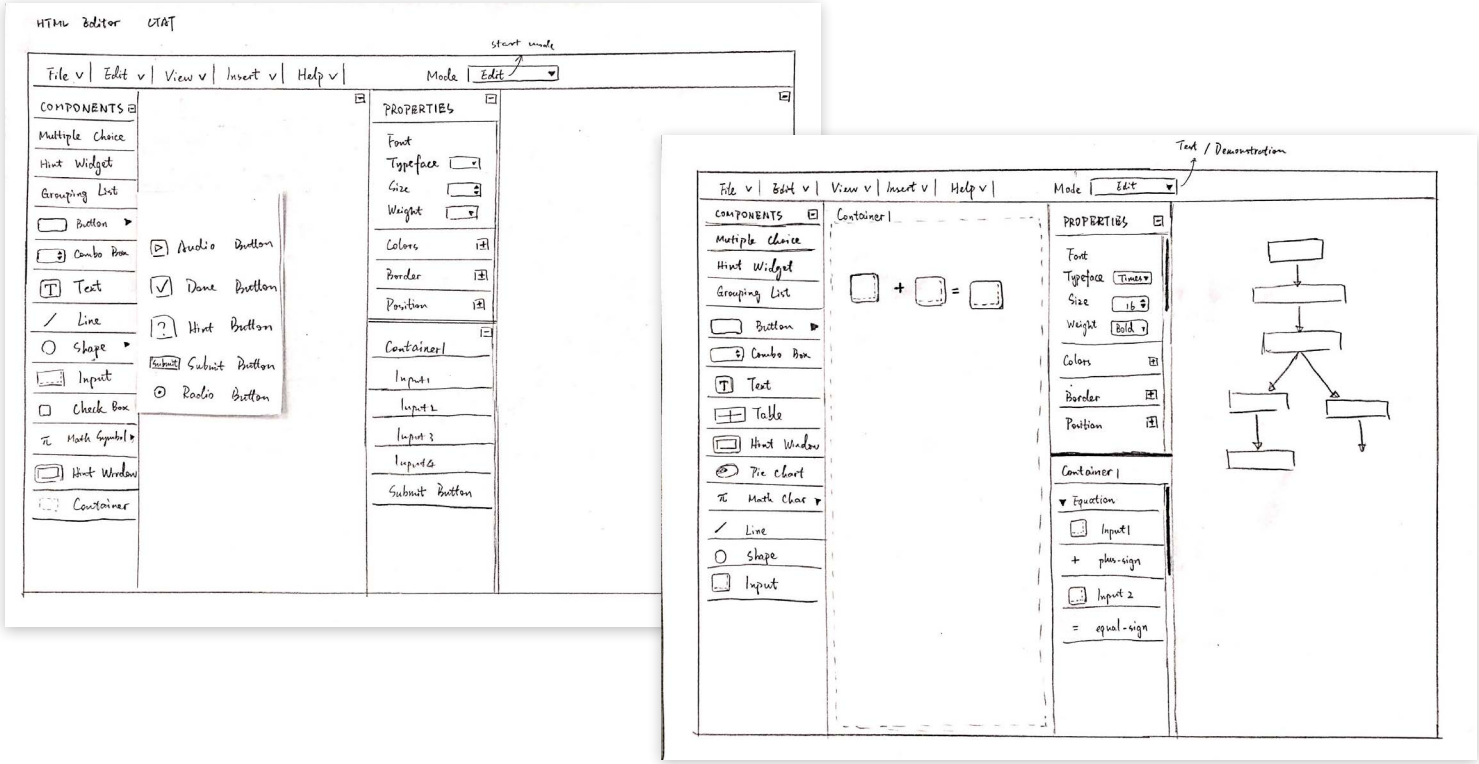
2. The Properties Panel

The properties panel is where properties like ID, Tab Order, Label, Group Name and so on are displayed. The feedback overwhelmingly focused on the naming of these properties. While providing maximum flexibility to expert users of CTAT, this fine-grain control of properties can be overwhelming and confusing to novice users. Additionally, although users may find this high level of flexibility useful for creating more complex tutors, these tasks become tedious for simpler tutors. This contradicts the ideology of "making simple things simple."

3. The Lasso Tool

Although minor compared to the previous problems, the lasso tool exemplifies frustrating usability flaws. Users expect to be able to select multiple components at once by clicking and dragging, but only with the lasso tool enabled. However, the tool stays undiscovered by most users because they do not expect this interaction.



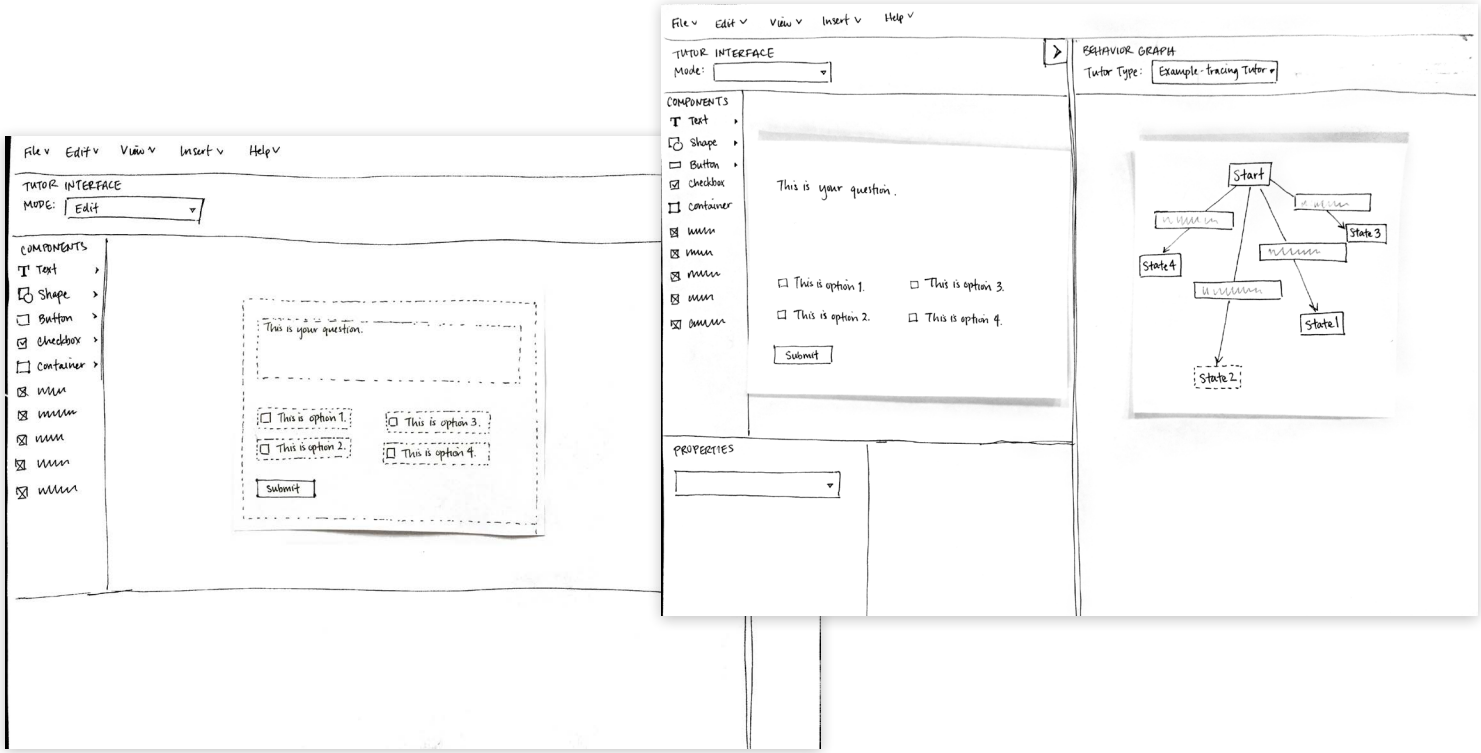


UI Version A

In prototype A, icons in the components bar serve as visual cues to help users locate items more easily, and to hint at the item's function. In addition, components with similar functions are grouped into submenus. The interface is split to show the HTML editor and the behavior graph at the same time. Users can switch between building the interface and the behavior graph by selecting the desired "state" from a dropdown menu.

Feedback

Overall, feedback was positive. Users found the reorganized HTML components bar, with submenus and icons, was helpful for locating components. The split screen also received positive feedback, as users did not have to switch between windows, between workspaces. However, the lack of an ability to mass edit was not addressed and will need to be furthered improved in future prototypes.



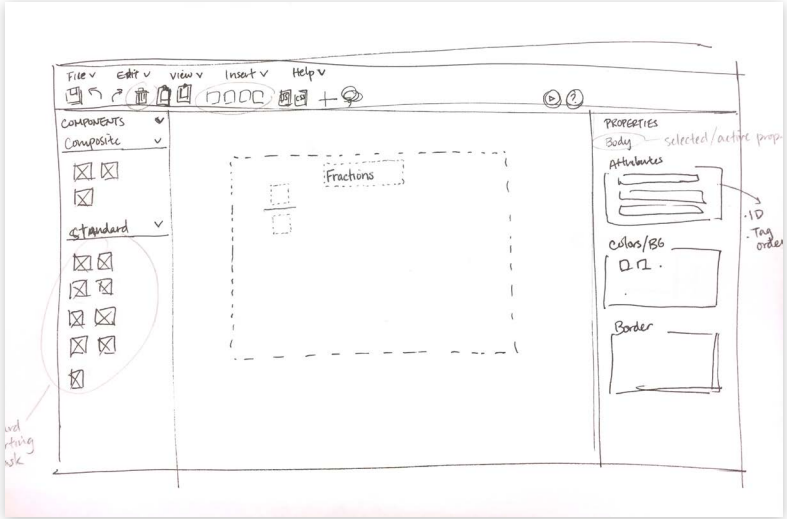
UI Version B

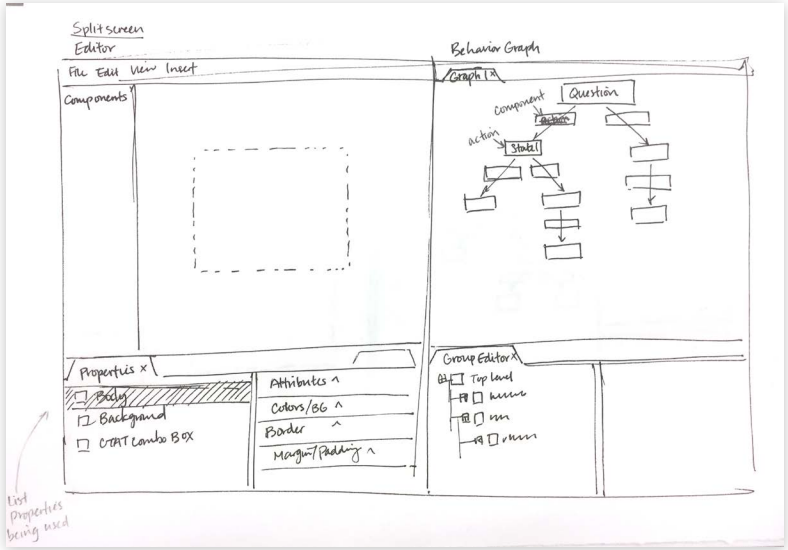
Version B features panels that users can expand or collapse depending on what interface they are working on, allowing users to focus on the task at hand in a more intuitive workflow. The "Mode" dropdown has combined options from the original HTML editor and behavior graph to control which panels are open as well. This prototype also features a grouped, icon-based components bar.

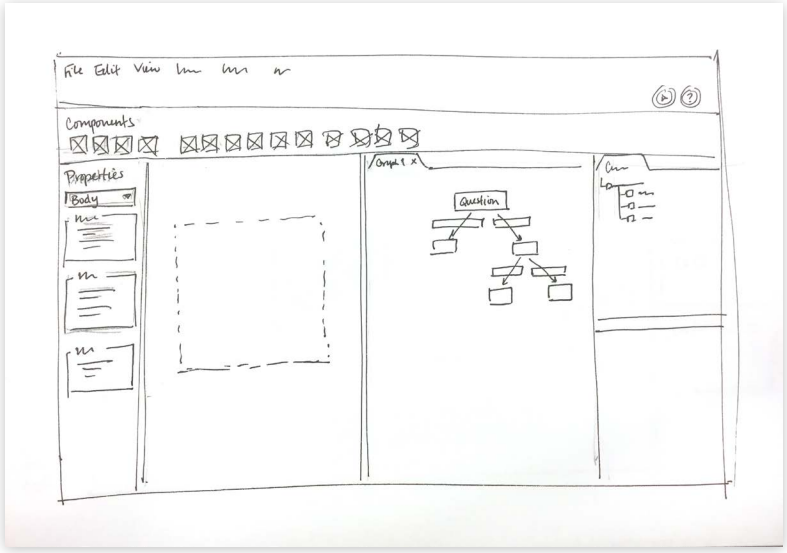
Feedback

Users found the ability to see both the tutor editor and behavior graph useful. When one user first expanded the empty behavior graph, she said the blankness led her to not know what it was for and she liked that she could ignore it. However, the copy-and-paste actions from the original interface were repetitive and tedious, and were not addressed in this prototype.

UI Prototype Iterations



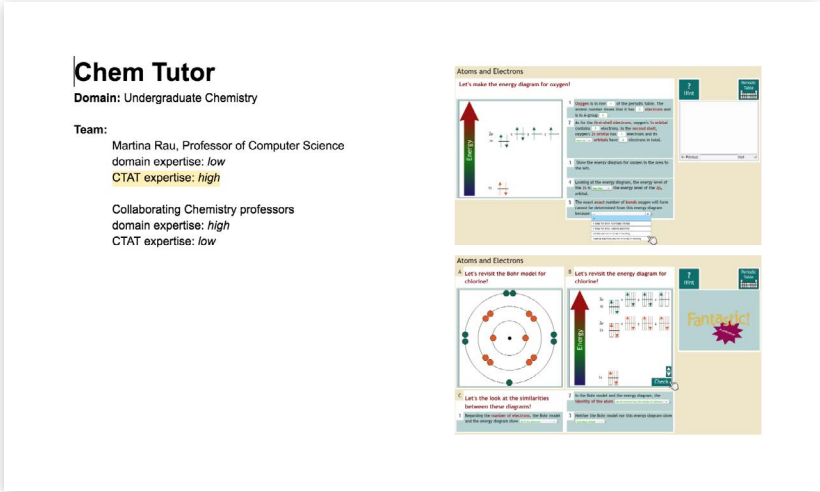


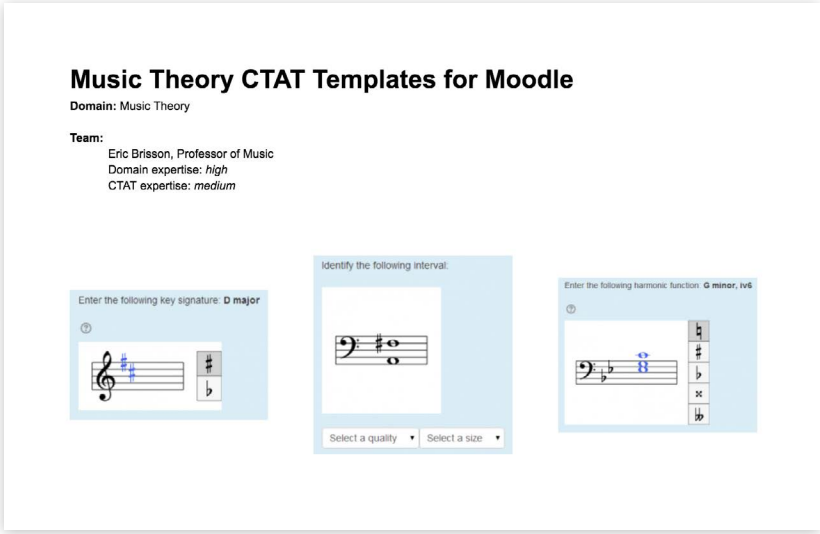


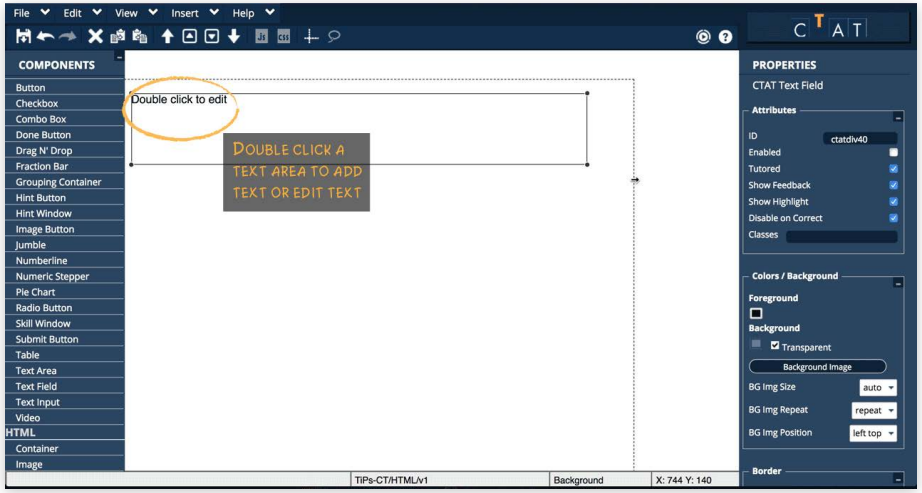
Case Studies Prototype (Untested)

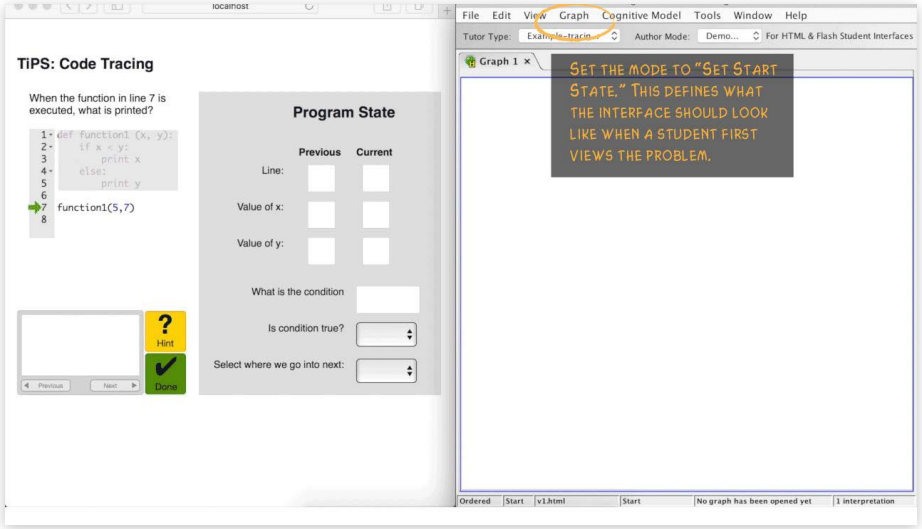
One of the earlier prototypes we created was a document of CTAT use cases. This was meant to address our finding that many users misunderstand the optimal use cases of CTAT. In order to help potential users understand how CTAT may be applied to their work, we considered creating a document that provides information about cognitive tutors created with CTAT. This document could be used as a resource for instructors and instructional designers to determine whether CTAT is suitable for problems sets that they want to create.

Additionally, for each tutor, we included information about the creators, their domain knowledge and their CTAT expertise. This was meant to combat misconceptions about the amount of experience needed to create effective cognitive tutors. In general, potential users believe they need more CTAT expertise than they actually do. Having this information available aimed to encourage new users to start using CTAT as well.









Mid-fi Prototype Design

In the mid-fi phase, we chose to work on the **initial use stage (tutorials)** because we felt that this experience would be the most beneficial to the goal of increasing adoption. We wanted to answer questions such as:

- Can users easily follow the tutorial?
- Can users remember what they learned, and can they apply it to make their own problems?
- What kinds of problems do users think CTAT is good at making after watching the tutorials?

The prototype we tested is an iteration of the video and screenshots tutorials created during the lo-fi phase. The problem we demonstrated was a code tracing problem in the computer science domain, instead of the MCQ problem used in the lo-fi phase.

The tutorial breaks the task down into distinct steps. Each step includes an overlay of descriptive text on the CTAT interface, followed by a short animated gif demonstrating the step. This design works to combine the benefits of the video and screenshots tutorials. As we learnt from testing, participants felt that text provided better explanations, while video made it easier to locate features on the screen. We hoped to see the text segments help users understand and remember what they learnt, and the gif segments make the tutorial easy to follow as per the original video format.

Mid-fi Testing Methodology

Participants were provided with the tutorial and a finished CTAT interface. The tutorial demonstrated how to use CTAT to create a code tracing problem (domain: computer science). Participants were shown the final tutor, then asked to recreate that tutor using the tutorial we provided. Understanding the details of the code tracing problem was not necessary for this task.

Participants were asked to use the tutorial and perform a **think aloud** following **one of the three tasks** below:

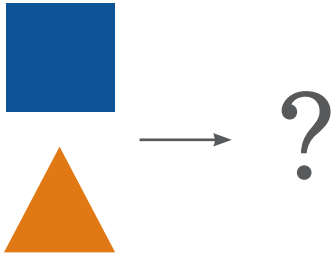
Test Participants

Our users were **6 CMU students** who have never used CTAT or other tutor authoring tools. One of them was a teaching assistant (TA) for a design course, while another was a TA for a computer science course. The rest have not been in teaching positions before.

Two users were assigned to each of the tasks mentioned previously.

1) Watch the whole tutorial and then recreate the problem

2) Recreate the problem following tutorial step-by-step



3) Recreate the problem using the tutorial in any way

Mid-fi Findings

1. **The tutorial was digestible, engaging, and educational overall.**

   From our user testing, we received feedback supporting our hypotheses that the combination of text and gifs helped participants complete the tutorial and understand how to use CTAT's features.

   Specifically, participants felt that the visuals in the tutorial effectively highlighted which interface components they should interact with and how. Furthermore, the gifs clearly showed what the start and end states of the interface should look like for each step, which helped our testers determine whether they performed an action correctly. Participants also found the amount of text in the overlays to be just right, not too much but enough to convey the requisite information about the CTAT feature in question.

   Compared to our earlier video tutorials, the steps had greater granularity which made the gifs easier to follow, because of their shorter length. Overall, our testers felt that they got a good understanding of the purpose of CTAT by completing the tutorial. As one of our participants said, "[The tutorial] made me want to know more because it only uses a few things to make something decent, so I want to know what else I can do..."

2. **Tutorial navigation: Users should be able to see an overview of contents and navigate between steps.**

   One key improvement for our next prototype would be the navigation. During our testing sessions, many users tried to go backward, skip forward, or jump to a specific step in the tutorial. Specifically, for users who expressed the need to jump to specific steps that they didn't quite understand or remember, an efficient navigation feature would be essential. From user data, we concluded that an ideal navigation apparatus should allow users to:

   1) Jump to any specific step at any point in the tutorial

   2) Refer back to the sample final result at any point during the tutorial.

   3) Skip a step if they want to.

   4) Repeat a step if they don't understand the first time

   Another piece of feedback regarding navigation was the consistency of interactions. For some steps, such as the welcome screen, the interaction to proceed to the next step is to click on the text overlay box, but for other steps, the user must click on the interface component itself. Having a consistent navigation interaction would make the tutorial easier to follow.

3. **Tutorial content: The tutorial needs additional segments to give users more clarity and understanding.**

   This feedback pertained to the actual content of the tutorial. Users mentioned that the overall goal of the tutorial is not clear, which hindered their understanding of the concept of a cognitive tutor. Our next iteration should clearly state the end goal at the beginning to ensure that the user understands what they are making before they jump in.

   We also received feedback about the granularity of the steps. Users felt that some steps contained too much information while others contained too little. One example of the former is the "set start state" step in the behavior graph tutorial. This single step covers clicking on the menu, selecting "set start state," naming the state, and clicking "ok." Some examples of the latter situation include the "scroll down" step in the HTML interface builder tutorial. It is so simple that it can be easily integrated into other steps.

   Lastly, many users felt anxious about making mistakes when following the tutorial, due to not knowing how to recover. This predicament is especially prevalent in the behavior graph tutorial. To solve this problem, the tutorial might address some common errors that the users are likely to make.

4. **Tutorial copy: Our copy should more effective in making each step as clear as possible.**

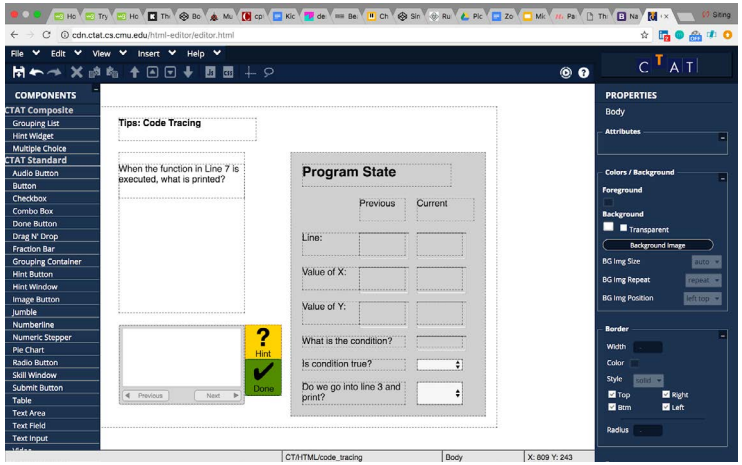
   One major problem we saw was that the overlay copy was not very effective. Some of the text was too wordy, so that users skipped reading or found it confusing. We realized that we were mixing two different types of information in the text overlay: the definition of terms, and action calls. In our future iterations, we plan to create a hierarchy in the overlay text so that users can choose to go through the steps very quickly, but still catch important information or read more about specific terms or interface components if they want to.

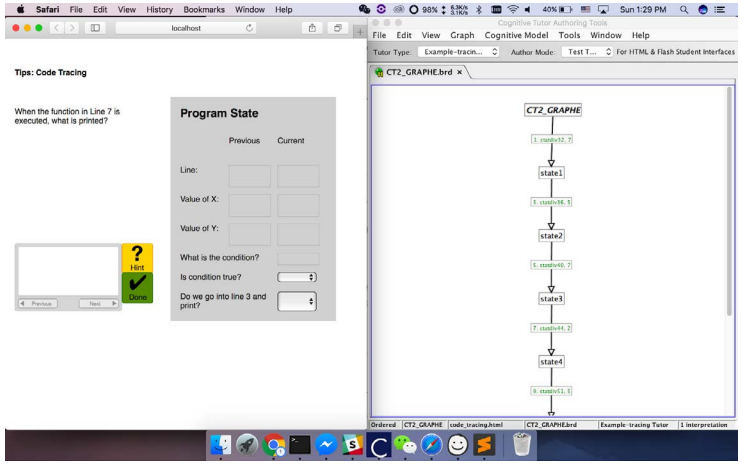
   Another problem was a lack of clarity. For example, we jumped over some steps showing the use of duplicate items that we already covered. One step called for many text boxes to be made, but we skipped ahead to show them all done. This jump was jarring as we didn't tell users why there was a jump, so they didn't anticipate it or easily follow what happened. We realized we should address this in our instruction boxes. Finally, incorporating more feedforward would provide some reassurance to users. The tutorial can ease the user's stress by letting them know what should happen or what to expect after they perform an action.

User Testing Results

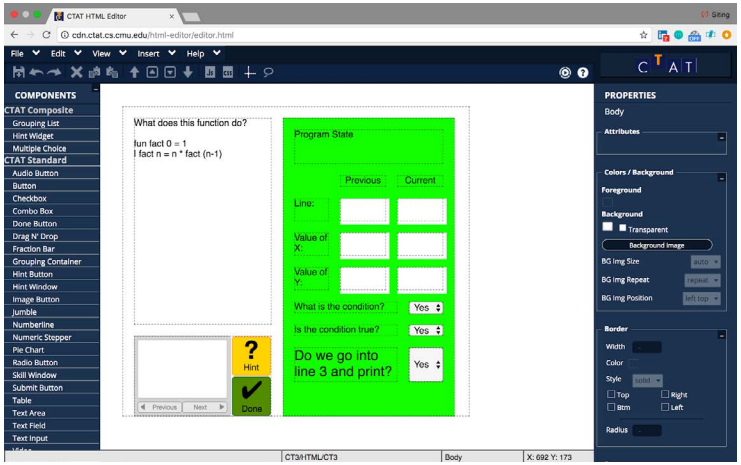
Here are 2 examples of tutors created by our user testing participants following our tutorial. Some experimented with the CTAT interface and created diverse outcomes.

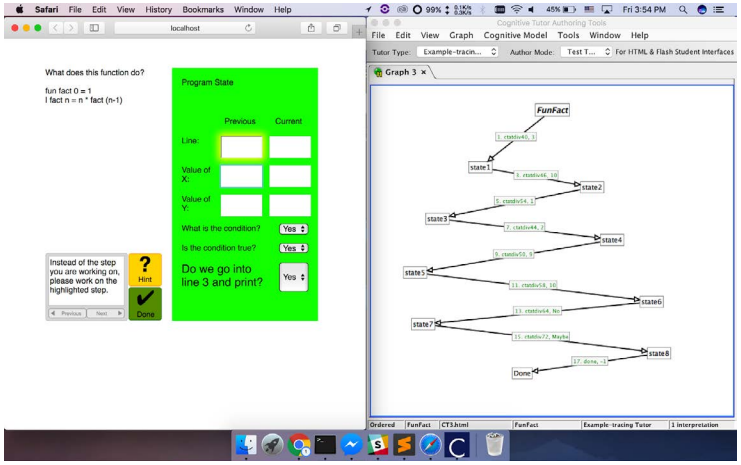
Result #1





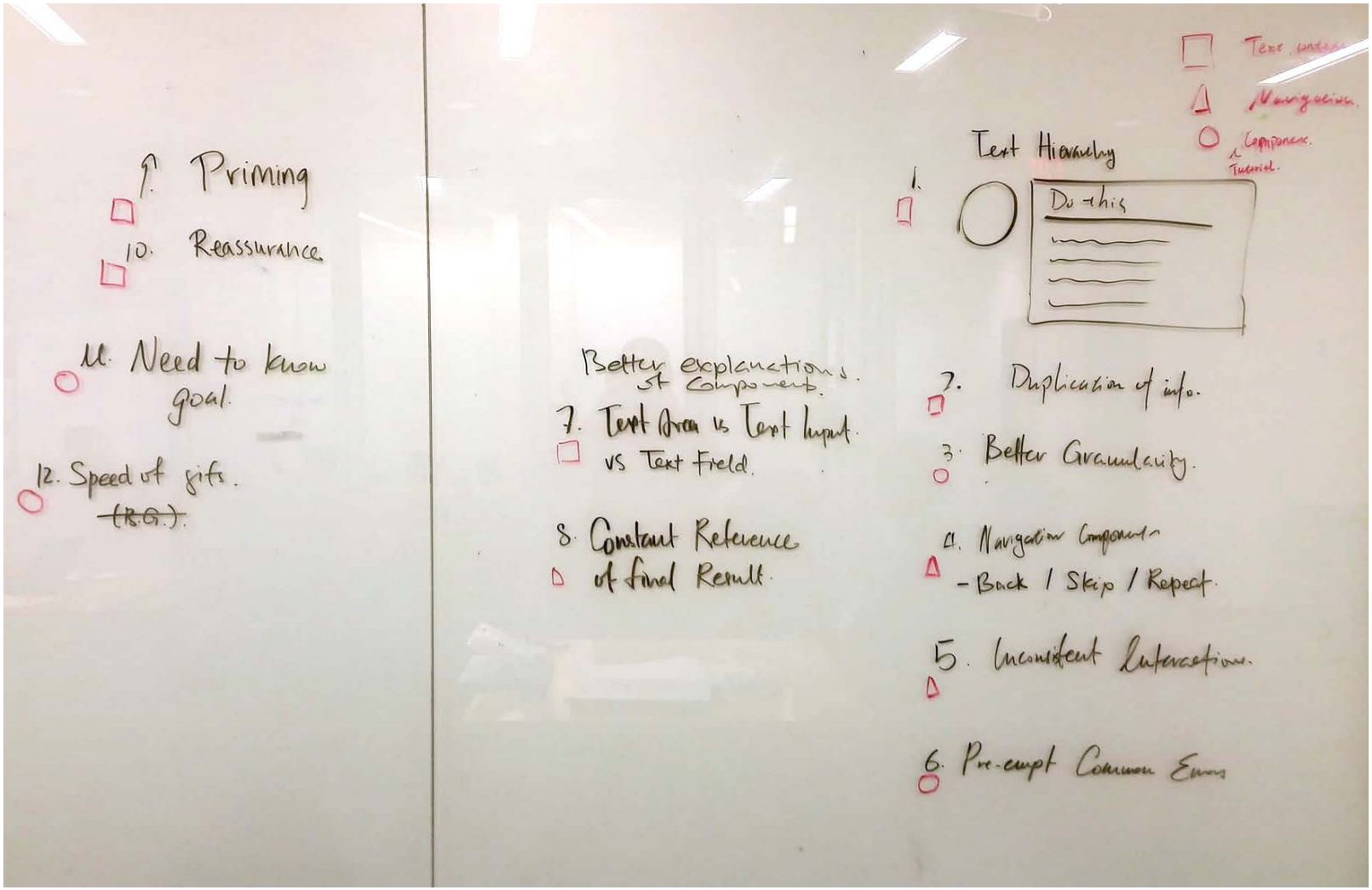
Result #2





Categorizing Feedback

Copy Navigation Components



Hi-fi Prototypes

Iteration 1

HTML tutorial:
https://invis.io/U5GIKANGMRT

Behavior Graph tutorial:
https://invis.io/T3GIPWSQFYR

Iteration 2

HTML tutorial:
https://invis.io/HMIKEM5X72E

Behavior Graph tutorial:
https://invis.io/6XIKEMBFYEH

Iteration 3

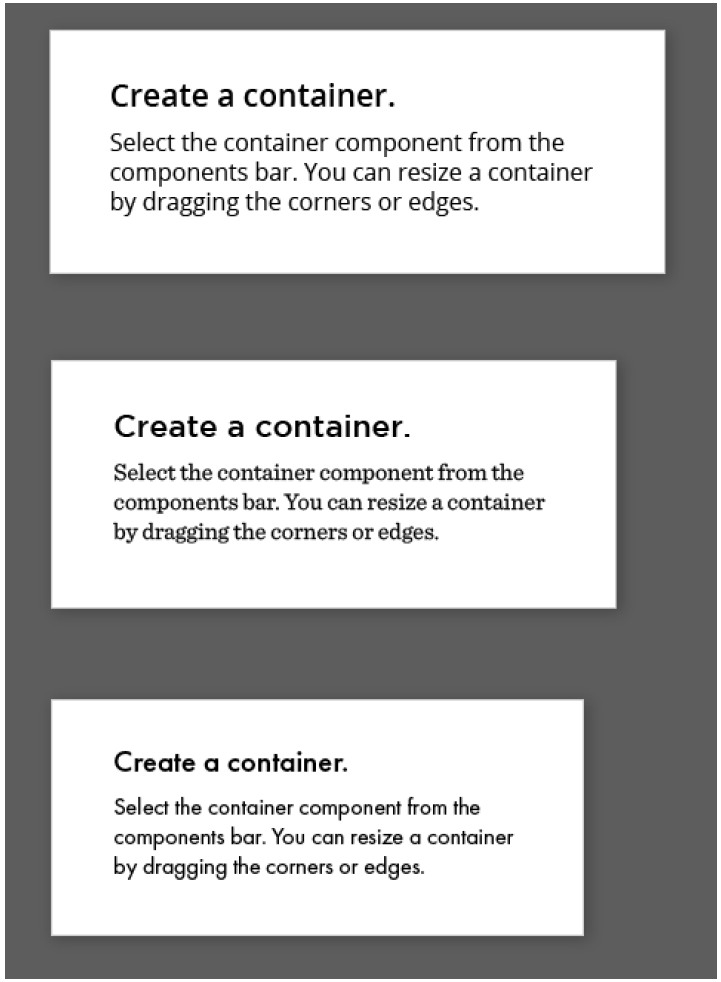
Web prototype (with tutorials):
https://invis.io/7KIKEE0XVCF

Final Iteration

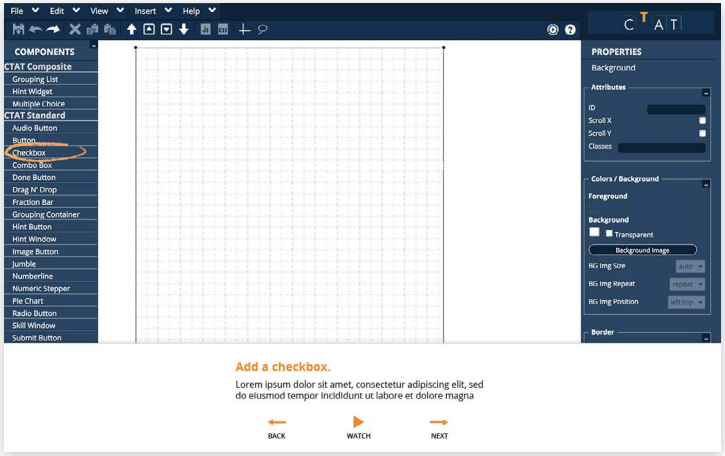
https://tinyurl.com/ybk787jr

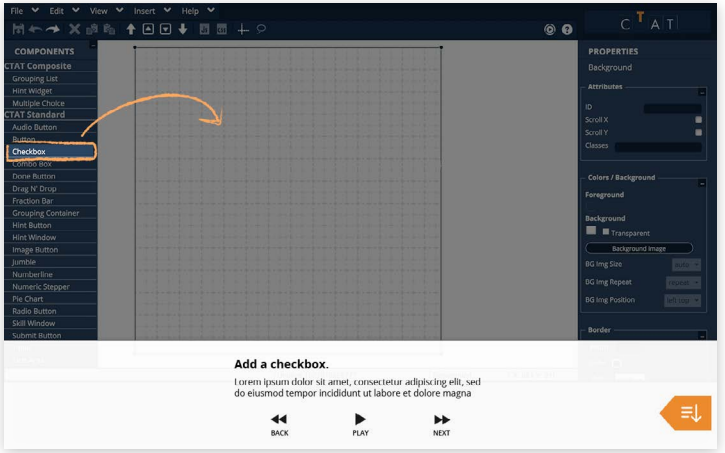
Iterations: Icons

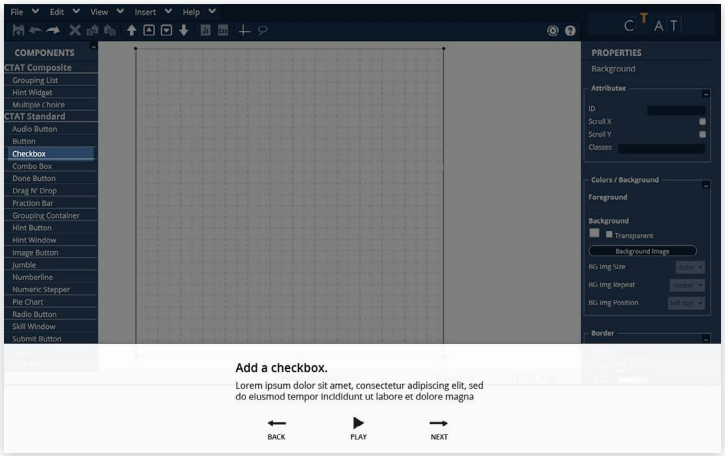
Iterations: Typography



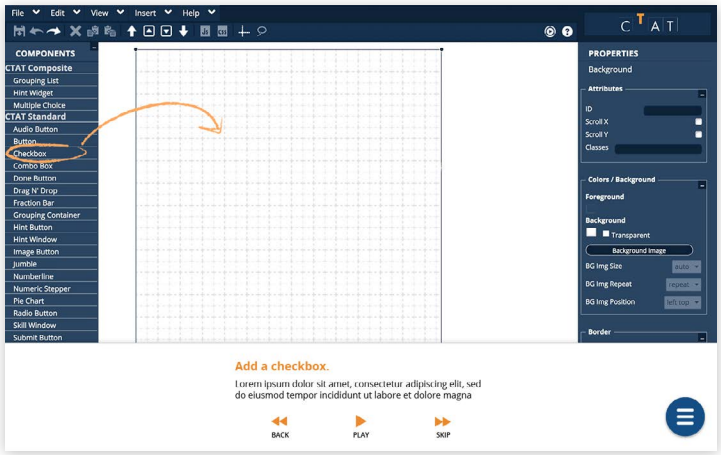
Iterations: Highlighting Components

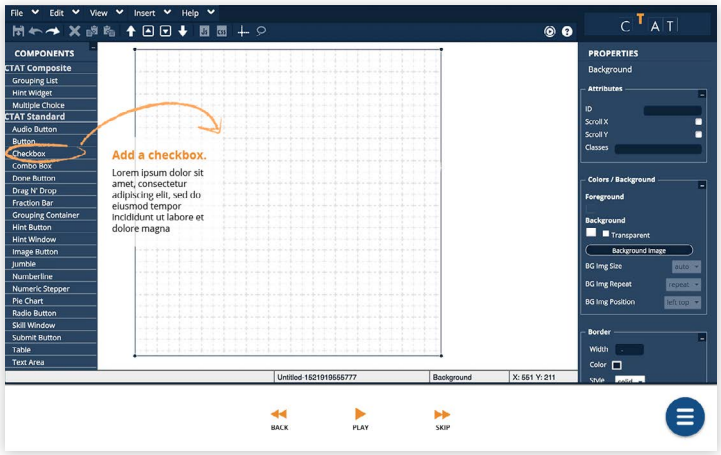


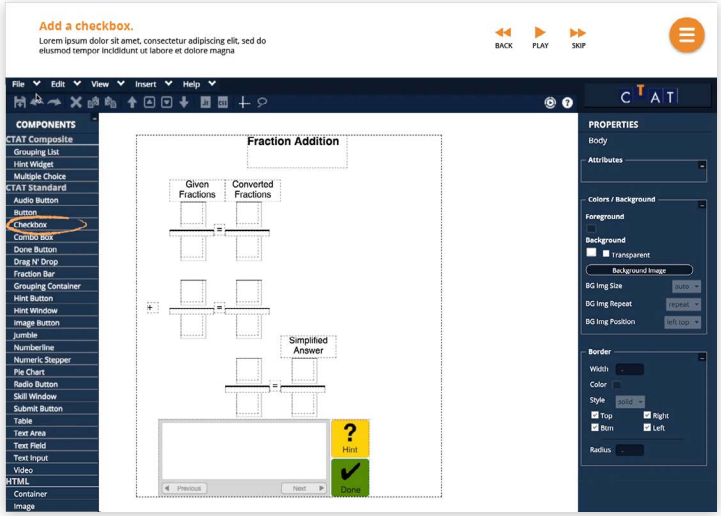


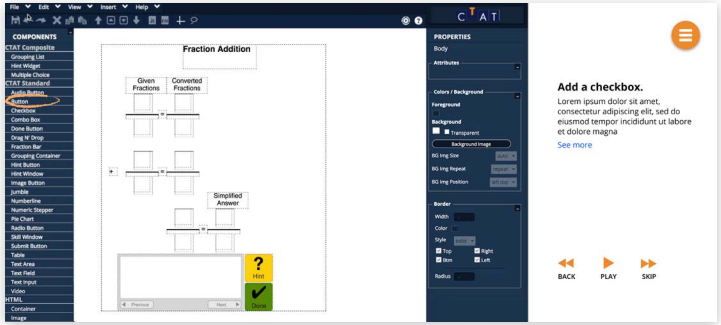


Iterations: Layout & Text Placement









Iterations: Web Pages