



Artbot

Final Report



Vincent Nicandro
UI/UX Designer & Project Manager



Casey Wong
Lead Developer & Product Manager



Zach Goodale
Full Stack Developer



Lexi Stein
Front-End Developer

Your antagonistic, artful companion.

Problem & Solution Overview

We set out to tackle the problem artists at all levels face during the creation process – the block. The block is when artists become too attached to their work but are unable to see a path forward in their work that satisfies them. Artbot attempts to free artists from this mode of thinking using a gamified approach coupled with long-term check-ins. Ultimately, our goal with Artbot is to empower artists at all levels by boosting their creative adaptability.



Task & Final Interface Scenarios

Simple Task: Look at previously drawn disasterpieces. (Figure 1)



Figure 1 - Task flow for simple task.

We chose this task as it allows for self-reflection on the user's part; moreover, it's a way for users to remind themselves of potential points of inspiration that may inform new pieces. Users get to the Gallery by clicking the *Gallery* icon on the bottom navigation bar, then clicking on the disasterpiece they wish to look at.

Moderate Task: Create a disasterpiece with Artbot's intervention. (Figure 2)

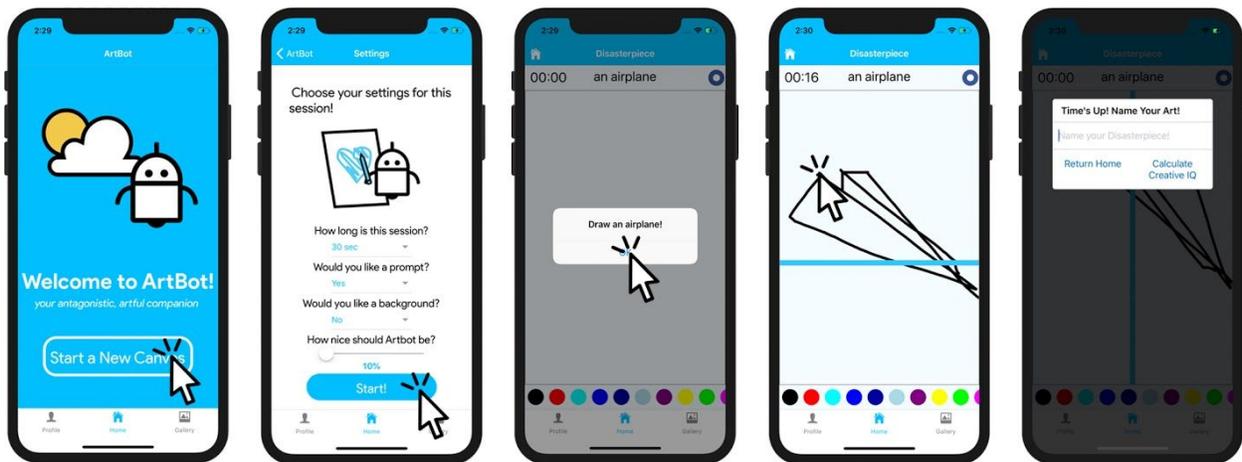


Figure 2 - Task flow for moderate task.

We chose this task as it is the very heart of Artbot's personality, and the disruption Artbot makes is what exercises the user's creative adaptability. Users initiate a session with Artbot by first clicking *Start a New Canvas*. Then, they modify the settings they'd like for the session, then press *Start*. The user will receive a pop-up with a prompt (if they decide to do so) and then will spend their chosen

session time drawing while Artbot disrupts their artwork. Once the session is complete, a pop-up screen appears asking users to name their piece.

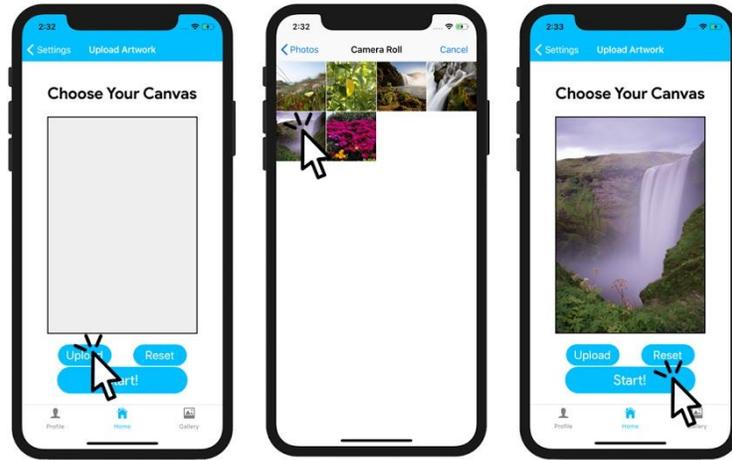


Figure 3 - Alternate task flow for moderate task.

There is a slightly alternate task flow if users decide to upload an artwork (Figure 3). After pressing *Start*, users will be taken to the *Choose Your Canvas* screen. The user then clicks *Upload*, at which point they can either take a photo or be taken to their camera roll to choose an image. After they've done so, they press *Start*, and the task flow continues as expected with the canvas portion.

Complex Task: Complete long-term check-ins with Artbot. (Figure 4)



Figure 4 - Task flow for complex task.

We chose this task because it gives users an opportunity to reflect on how their creative adaptability has grown through working with Artbot. For this task, there are two main parts for maintaining long-term reflection with Artbot - checking in and viewing old artwork. With the check-in, users are able to engage in a conversation with Artbot, reflecting on how satisfied, creative, proud, and

adaptable they felt during their session. From this, Artbot will calculate the user's creative IQ for the session and assign it to the disasterpiece they just created.

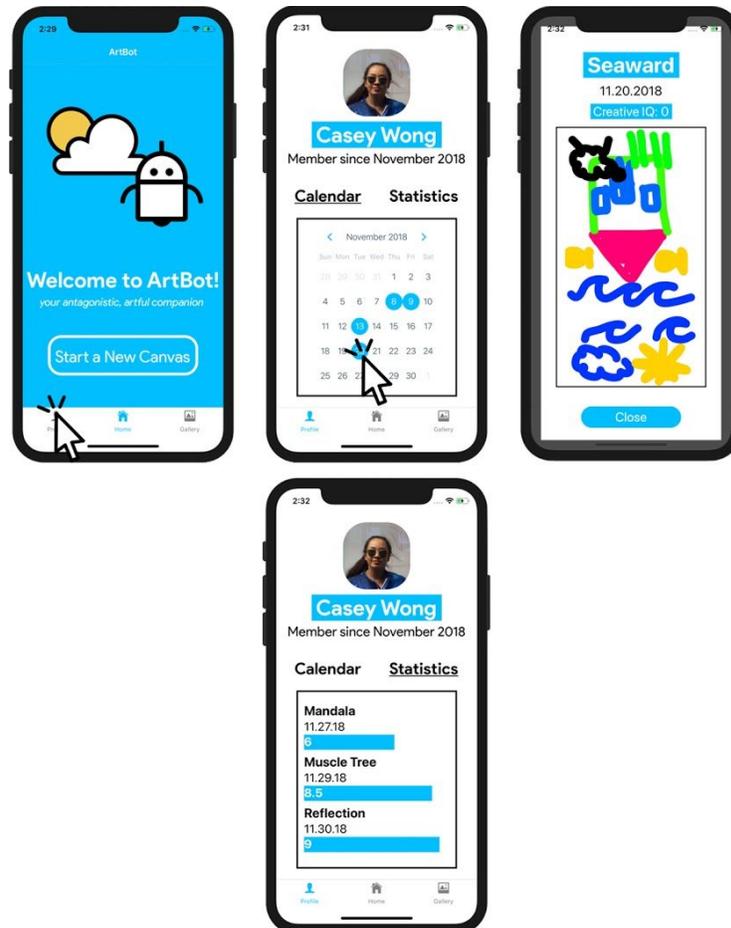


Figure 5 - Task flow for long-term reflection.

The second part of the long-term reflection with Artbot is revisiting old disasterpieces and witness their creative IQ growth over time (Figure 5). From the profile screen, the user can interact with a calendar and statistics view. From the calendar, they can navigate to previous dates in which they created a disasterpiece and then subsequently view that piece. From the statistics view, users are able to see how their creative IQ has grown through a bar graph. Again, the user can click on the bar graph information and view their disasterpiece.

Design Evolution



Figure 6 - Low fidelity prototype screens.

For our Lo-Fi prototype (Figure 6), we strove to keep a relatively straightforward design to make the application flow as intuitive as possible. Users would be welcomed by a home screen where they had 3 options: start a new project, go to their gallery, or view their progress. In this section, I will walk the reader through each of those tasks and present the design evolutions from our original lo-fi to the medium-fi and, finally, to the hi-fi.

On selection of *Start a New Project*, the user was taken to a new project screen where they could either upload a canvas background or opt for a blank canvas. In our lo-fi prototype testing, we found users preferred to start with a blank canvas, making us opt to move the option for a canvas background to the home screen to minimize the flow to begin on a blank screen. This meant the new project screen was bypassed completely and the user was taken to the session settings screen. If *Upload Artwork* was selected on the med-fi, the user would be prompted to choose a canvas background before continuing to the settings screen. After further thought on this placement, we finally relocated this option to the settings screen in our hi-fi prototype to keep our home screen more streamlined. This meant that the upload background screen would only be encountered if a user opted for a canvas background in the settings screen.



Figure 7 - Evolution of settings screen across three prototypes.

Moving on to the settings screen, where the user could modify how the ArtBot canvas session would proceed, we opted to change the inclusion and phrasing of many settings over time. Of note, the settings title, “Artbot Intrusion” tended to cause confusion among participants, which we eventually changed to “How nice would you like artbot to be”. It was a point of difficulty to come up with phrasing that was intuitive as the concept of artbot had to be understood for this setting to be understood. In a release of this project, we would definitely opt for an onboarding approach to familiarize the user with each of these settings and their functions. The actual canvas has stayed fairly constant throughout the design iterations; our final placement of canvas tools really was based upon how our API was formatted. The biggest design evolution came in the task flow after completion of the disasterpiece. Initially, we had the user complete a survey in order to calculate creative IQ. After deciding this was too formulaic, we opted to exchange this survey for a more interactive chatbot! On the med-fi and hi-fi, the user has an option to either talk to the chatbot immediately after completing their disasterpiece or later if they would prefer.



Figure 8 - Medium fidelity prototype screens.

Throughout our design iterations, our gallery has stayed more or less the same. We opted for a simple gallery display where users can see their past disasterpieces and click on them for more details. In our final design iteration, we decided to put a link to chat with our chatbot in order to calculate creative IQ. This allowed us to further streamline our home screen and remove the option to do so there.

Finally, for viewing progress over time, we stuck with a calendar and chart implementation to do so. Design-wise, we initially had an initial screen where the user could choose which feature to see; however, we modified this to have one visible with a tab available to switch over to the other view.

Circling back to our home screen, we changed what and how functionalities were accessible on each iteration. Initially, each of our three tasks discussed above was accessible as buttons on the home screen. With the med-fi, we added a bottom tab navigator to go to the gallery or to view progress, with some of the features that were initially all embedded in the disasterpiece flow relegated to the home screen as buttons. For our final iteration, we reduced our home screen down to one simple button to start a disasterpiece. We then moved accessing the chatbot to the gallery and choosing a canvas background to the session settings screen.

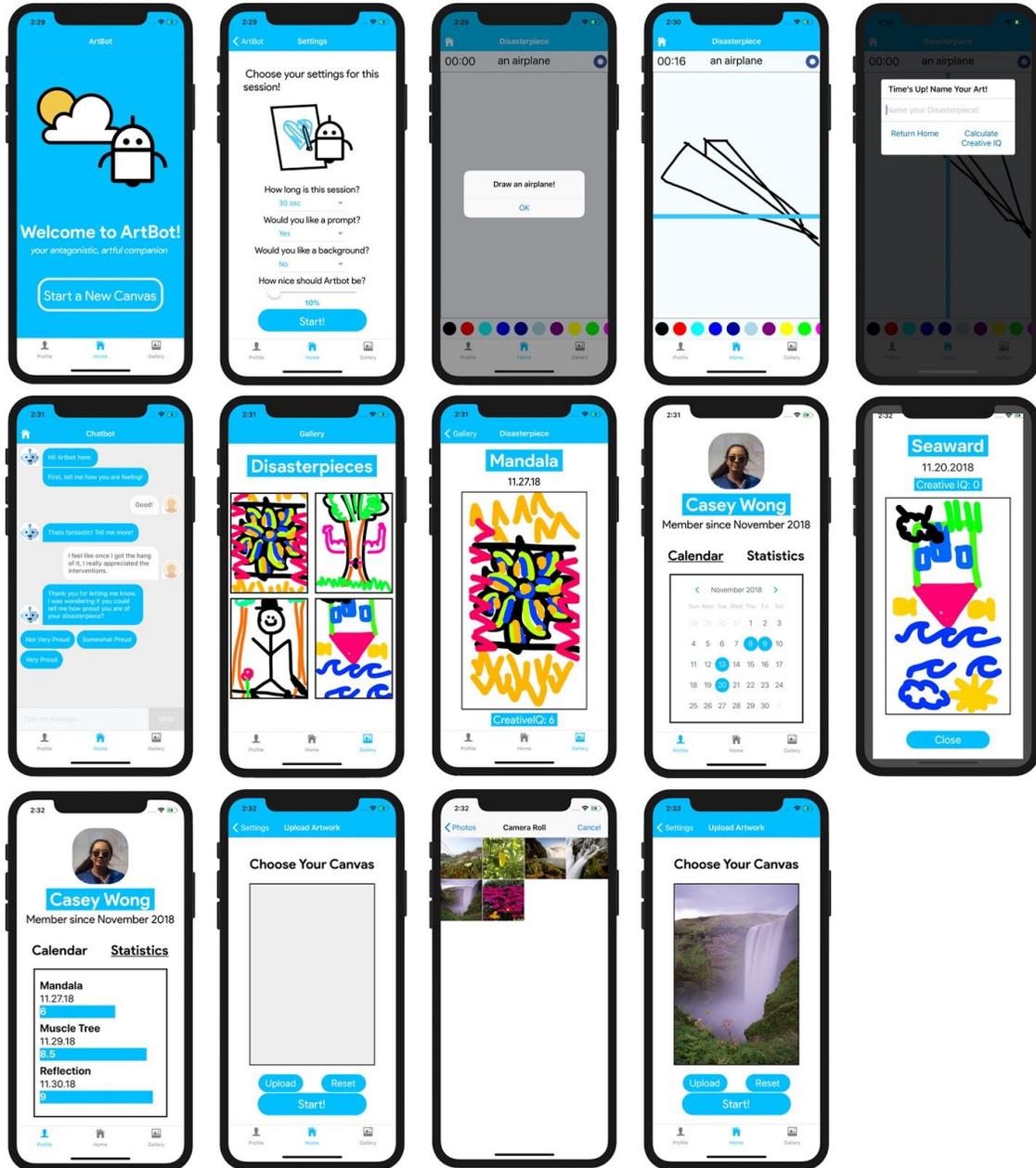


Figure 9 - High fidelity prototype screens.

Major Usability Problems Addressed

From our thorough review of our Heuristic Evaluation, we wanted to point out the major usability problems testers observed, as well as our response:

1. H7 - Flexibility and efficiency of use

Tester comment: On the actual canvas the bar on the left for brush size is nice but might get in the way and make starting strokes on the left side of the phone difficult.

We agree, and our solution to this was a toggle function which allowed users to click to change pen size. We acknowledge that this isn't a perfect solution, and we intended to have a pinching motion for this section (similar to Snapchat's pen functionalities), but the package we were using for our canvas was not amenable in this regard. Regardless, this keeps the pen size within the bounds of the screen.



Figure 10 - Evolution of stroke width tool after heuristic evaluation.

2. H7 - Flexibility and efficiency of use

Tester comment: The navbar disappears when you click on an option on the home screen, which doesn't exactly solve the issue mentioned in the presentation of reducing steps to reach different app screens.

This was intentional as it initiates a new task flow that doesn't require (and is, in fact, complicated) with the presence of the profile and the gallery. However, we also saw value in giving the user total control. We chose to predict a generous number of accidental clicks and the nav bar is now almost always accessible.

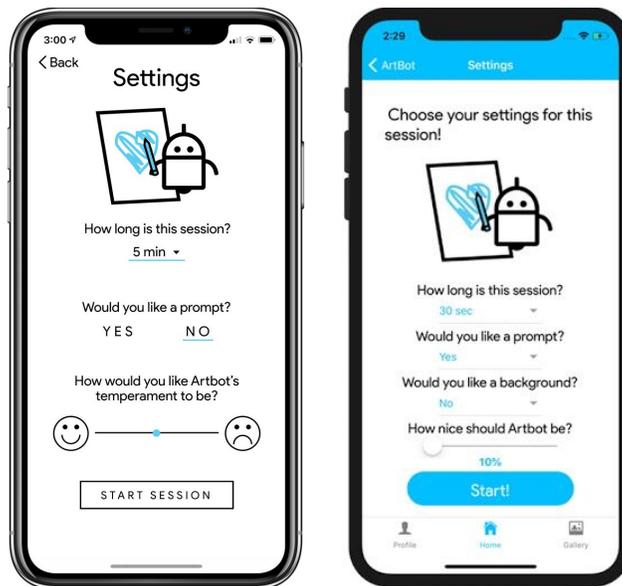


Figure 11 - Evolution of navbar access after heuristic evaluation.

3. H7 - Flexibility and efficiency of use

Tester comment: Both pages of personal screen provide links to the user's artwork - 2 ways to access the same information.

We'd argue that both don't show the same information; one shows specific points in a session while the other shows relational data. We'll clarify this in the way we present our data further, but we like the redundancies this provides.



Figure 12 - Evolution of statistics page after heuristic evaluation.

4. H3 - User control and freedom

Tester comment: When the user exits from a creation session the only two options are to delete the artwork or continue drawing.

This goes against the ethos of our app; the point of the app is for on-the-spot creativity and adaptability. Adding such a feature to save and exit means that users can stop and give themselves time to think about their artwork, which isn't really what we want. We actually no longer have the screen that stops you from going home because we didn't have time to add it to our prototype, but we still purposefully don't give the user the option to save an image before the time runs out.

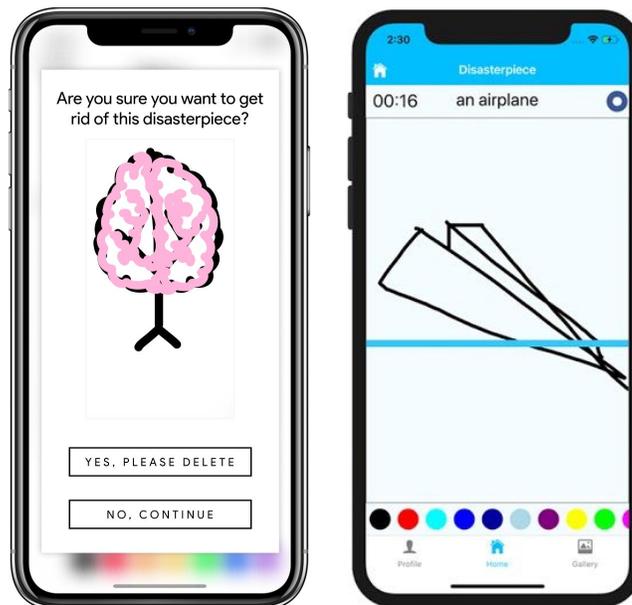


Figure 13 - Consistency of prevention from saving before time runs out after heuristic evaluation.

5. H5 - Error prevention

Tester comment: The chatbot interface is quite nice and friendly, but I hesitate to how many users would actually want to use that functionality.

This was actually something we already considered in the lo-fi prototype, but we found it was too clinical. However, given the limitations of our hi-fi prototype, we ultimately implemented a hybrid of this option, with a chatbot that offered both hard-coded quantitative responses as well as a qualitative response that allowed users to further develop their answers.



Figure 14 - Evolution of chatbot after heuristic evaluation.

There were other usability problems (such as removing/saving images, introducing Artbot as a character, etc.) that we wanted to implement but due to limitations of our working knowledge or timeframe couldn't do so; were we to continue, however, we'd definitely implement these fixes.

In addition to these main problems, we decided to implement many fixes, including:

- Clarifying status indication for check-ins for pieces, implementing that tracking in the gallery pages in place of the home screen.
- Have the confirm artwork screen by an inset window rather than a new screen.
- Make the prompt a pop-up to maximize real estate.
- Clarifying Creative IQ as a quantitative measure of creative adaptability over a period of time using the app (calculating a score based on the responses chosen).
- Change behavior indicators of Artbot from happy/sad to nice/mean.
- Select camera roll.
- Delineating the drawing canvas.

Prototype Implementation

For Artbot, we used React Native and Xcode. React Native provided us with many useful tools such as buttons, navigation bars, and component structures. We were also able to find React-Native compatible packages to execute our more complex features such as the calendar, the chatbot, and the canvas. On a stylistic level, React-Native provided style and flex boxes to allow us to design a clean and consistent user interface.

However, there were some aspects of React-Native that impeded our development of Artbot. While our functionality relied heavily on outside packages, the outside packages were not amenable to modification or restyling. In addition, there were some features not provided by react-native, such as the statistics graph on the profile screen. The most hindering aspect in developing this app was the fragility of React Native and Xcode. We frequently had to remove our node_modules folder and reinstall in order to get our code to work. We also had to manually link many of the packages to our project and install indirectly related packages in order to use install features. Even if we were able to link and install these packages on one computer, the same approach wouldn't work on another's computer.

In light of these limitations, we had to use some Wizard of Oz techniques and hard code features. First, we hard-coded all the of the chatbot responses, such as "Oh really? Tell me more!" and "I'm sorry to hear that. Could you explain more in detail?" in order to give the illusion that Artbot was processing and understanding your responses. Second, we also used Wizard of Oz techniques in calculating creative IQ. Within chatbot, users were given three options analogous to "bad", "ok", and "good" given arbitrary values 1, 5, and 10 respectively. With these values, we averaged the values to generate a creative IQ for our users. Finally, our biggest Wizard of Oz technique was Artbot's canvas intervention. Eventually, we want Artbot to be able to analyze what the user has drawn and intervene based on that analysis. However, for our hi-fi prototype, we created a data set of interventions (ie. drawing lines, erasing, and flipping the canvas) from which we randomly selected to disrupt the user's canvas. Alongside our Wizard of Oz techniques, we also hard-coded several Artbot features. For one, we were unable to set up a storage system for our disasterpieces. Thus, the gallery display were all hard-coded from a predefined array of our disasterpiece images. Similarly, for the calendar and stats graphs, the artwork available for a chosen day or graph data point was hard coded.

There are many features missing from our final prototype. Given time, we would implement the following:

- Artbot's intervention during the drawing process utilizing AI
- Artbot's chatbot taking in qualitative and quantitative measures and outputting an actual CreativeIQ score, rathering than averaging arbitrary values
- Dynamic storage for Disasterpieces including the image, name, date, and creative IQ score.
- A way to edit and change the user profile
- Using a line graph instead of a bar graph on the statistics page
- A tutorial that would navigate users through their first session with Artbot
- An achievements section detailing milestones reached with CreativeIQ scores/streaks/etc.

- A more refined way to toggle pen thickness during the session (currently, the user has to click at the toggle to change sizes)
- Error prevention and the ability to delete old disasterpieces or data.

Summary

Many artists struggle with pressures of perfection when creating art. This mindset often renders artists unable to move forward with their creative process. It is our hope that with Artbot, we can create a gamified, low stakes application that will help free artists from this mode of thinking. Through Artbot's antagonist intervention of the user's work, Artbot will push users to think in new and creative ways, and overtime, see their creative adaptability grow. Through this course, we have modified our design to push the user's perception of creating art and observing their own self-growth. Furthermore, we've learned how to conduct need-finding, productively brainstorm ideas, generate prototypes and conduct testing to create a working product. From this project, we hope to apply these skills in both programming and design thinking beyond Artbot.