

10LINES

Draw together to kickstart your creativity!

TEAM ART-ATTACK

Selina Her - Design + Branding

Nyasha Smith - User Research + Development

Ben-han Sung - Design + Development

Sarah Wymer - Filming + Development

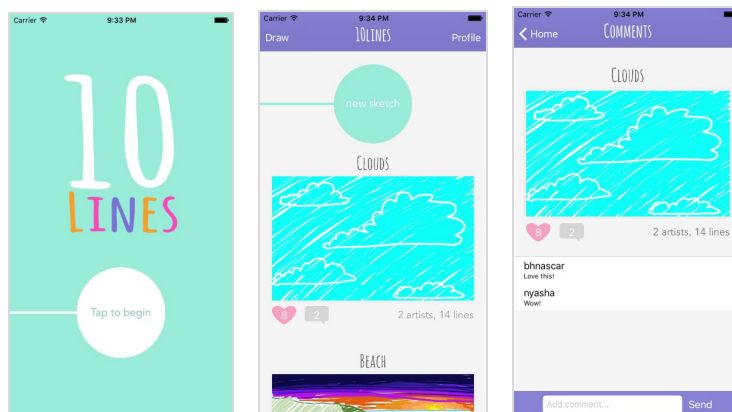
PROBLEM + SOLUTION OVERVIEW

Everyone knows about writer's block, but artists have their struggles too. After interviewing art students and professionals in the local area, we realized that artists often lack inspiration when business and academic obligations pressure them to deliver. Our mission is to channel the creative energy that arises from collaboration in order to help artists defeat creative paralysis. 10lines is casual, collaborative drawing game where participants take turns adding to a sketch one line at a time. With only 10 lines and 3 undos allowed per participant, players are implicitly encouraged to abandon the idea of perfectionism and find wild ways to transcend the limitations. The constraints fuel creativity by challenging each participant to adapt to the art direction of the other players and create a cohesive work of art together, line by line.

TASKS + FINAL INTERFACE SCENARIOS

The following is a list of tasks that a 10lines enables users to accomplish, along with a rationale for each tasks and screenshots of the task flow:

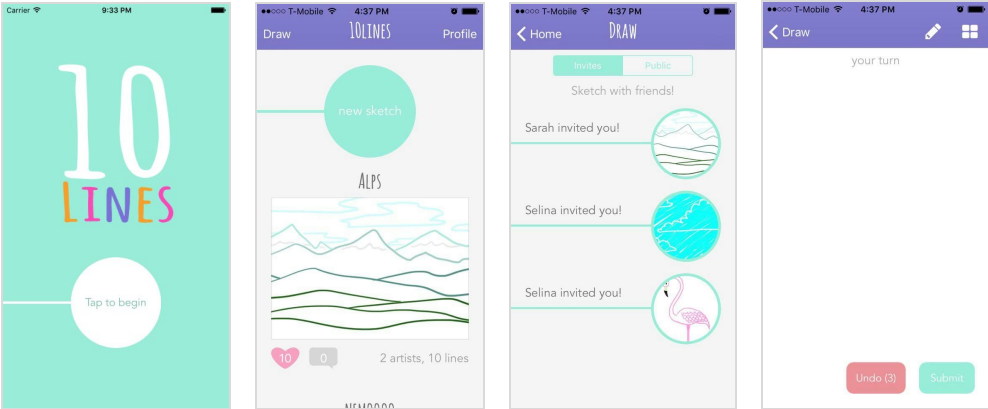
Simple - Browse your friends' drawings and public drawings for inspiration.



In keeping with 10line's mission to inspire artists and get their creative gears turning, we wanted a way for users to view sketches submitted by other people. Thus, we created an art feed, which will be the first view the user sees when they enter the app. By making the feed front and center, users will get maximum exposure to other people's ideas.

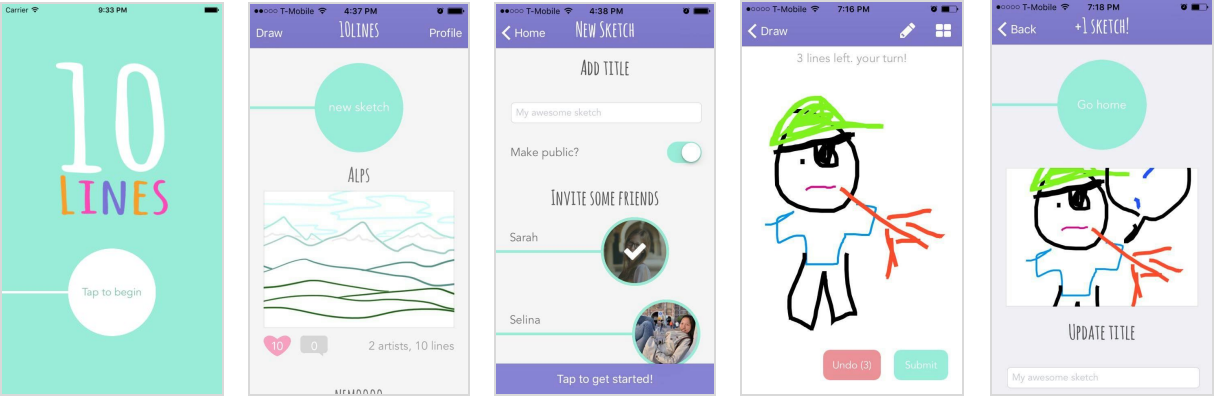
Users can scroll through the feed by swiping up the screen, which will keep loading images as it scrolls. Users can also tap on a particular image to add comments and show appreciation to the artists.

Medium - Draw a picture with your friends.



This is the core functionality of the app. Once a user has looked through some existing sketches, he/she might be encouraged to start their own sketch. Sketching with friends is yet another way to find inspiration. Some users are more comfortable interacting with people they know at first, so creating a task flow that allows users to invite friends was a rational step. Users can join a sketch through invitations from friends or invite friends when creating their own sketch.

Complex - Draw a sketch with the public - meet people you might now know!

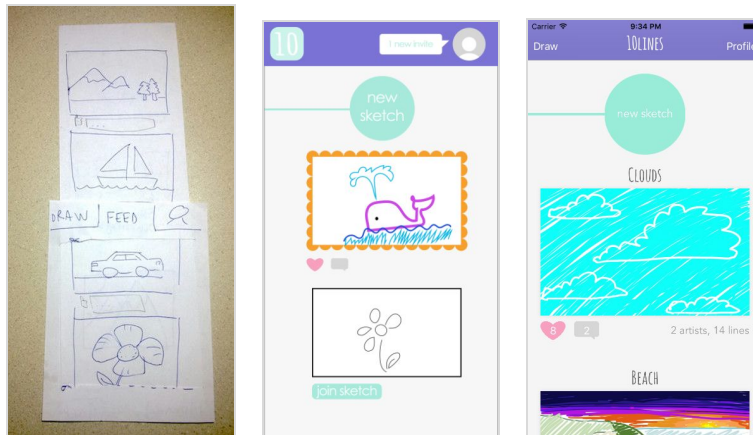


Once users are comfortable with using the interface, they may feel adventurous and want to draw with the public at large. In essence, the big difference between what is open to the public and what is with friends only is the use of toggles. The default setting is to show the “friends” only sketches. Through toggles, users can also make sketches they create open for the public to join, and also join existing public sketches. After a sketch is complete, users can add the other people they were drawing with as friends.

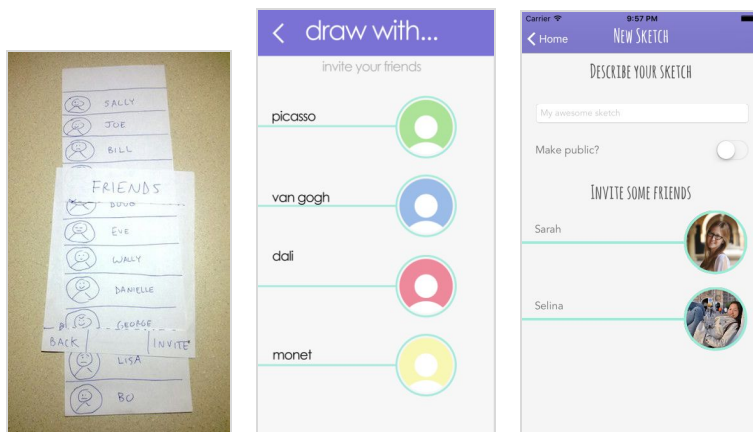
DESIGN EVOLUTION

Here’s how our 3 tasks evolved from sketches to the final design. More details and justifications on the changes from medium-fi to hi-fi can be found in the next section.

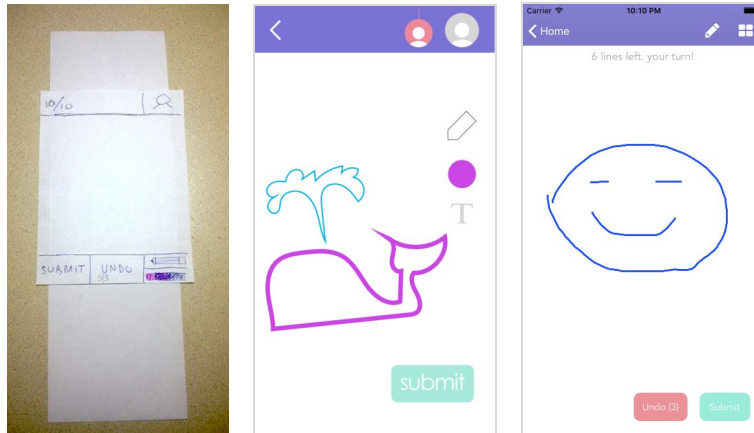
Simple - Browse your friends’ drawings and public drawings for inspiration.



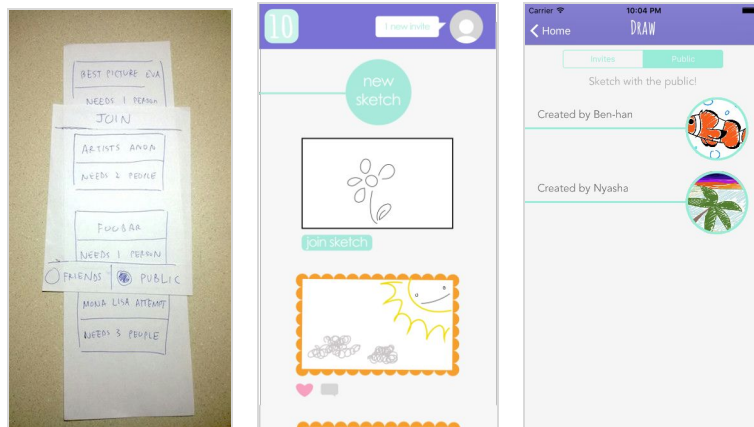
Medium - Draw a picture with your friends.



And the drawing screen’s evolution:



Complex - Draw a sketch with the public - meet people you might now know!



MAJOR USABILITY PROBLEMS ADDRESSED

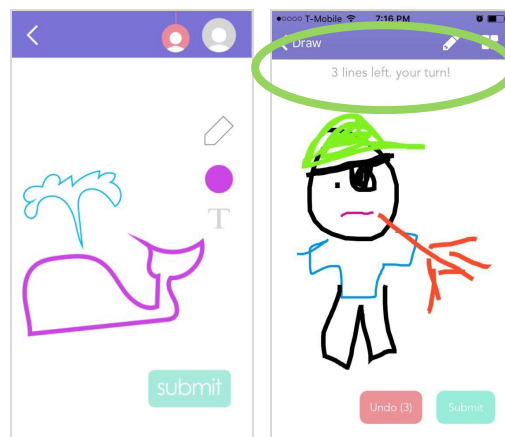
Design Changes' Reasoning

Our medium-fi prototype was evaluated heuristically by our fellow classmates in studio. We used the heuristic violations that were found to make iterative improvements to our application design. Below are the heuristic violations that were found for severity level of three and four and the steps we took to fix them.

The medium-fi screens with the heuristic problems are presented on the *left*. The high-fi screens with changes are present on the *right*, with changes marked in **green**.

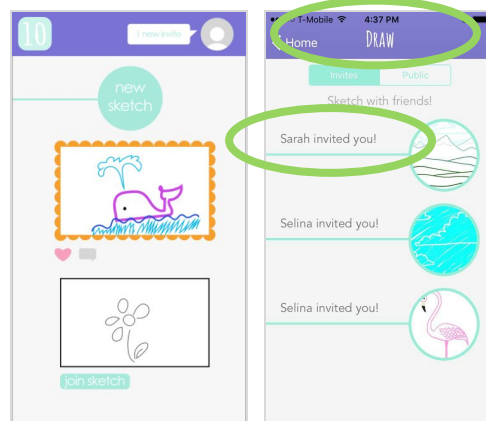
1. [H2-2 Match Sys & World] [Severity 4] When you submit, it looks like you can no longer work on the picture. I say this because usually after your turn, it tells you it's someone else's turn and lets you know how many lines are left. However, when you submit, this doesn't pop up. This not only seems inconvenient for the user, but can be especially inconvenient for the other collaborators (especially if they wanted to keep going). I would expect it to allow me to submit my current progress but still go through the normal turns screens to continue the drawing.

Solution: We created a label below the navigation bar showing the number of lines left. This label updates automatically after a user completes drawing a line. For example after one line is drawn the user will see the "9/10 lines left" right below the drawing screen header.



2. [H2-3 User Control & Freedom] [Severity 4] Clicking the invite button at the top right sends the user directly to a sketch. What happens if there is more than one invite? How do you pick?

Solution: The invite button now takes user to a new screen containing all of their invites listed and each invite displayed for example says "John has invited you!". Tapping this invite will take you to the selected sketch.



3. [H2-5 Error Prevention] [Severity 4] If a user clicks the back button in the top left corner of the drawing page after edits have been made to the drawing, the user is taken back to the main menu screen. If the edited drawing is reopened then the edits are no longer present. It seems likely that a user may accidentally click back and lose their unsubmitted progress or abuse this mechanism as a way to undo lines.

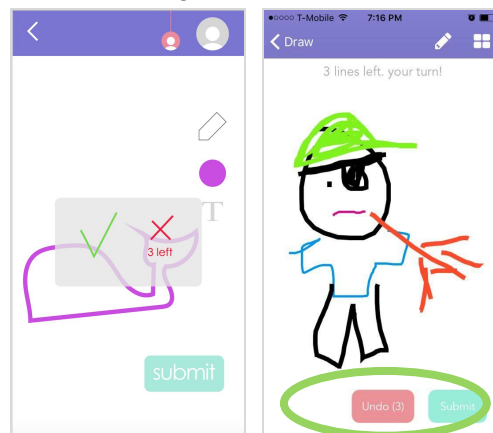
Solution: We are planning to create a pop-up tab to ask the user to confirm they are navigating away from drawing board and their changes will not be saved. This is not currently implemented in the current prototype due to time constraints.

4. [H2-6 Recognition Rather Than Recall] [Severity 4] When you made your own sketch, it constantly told you how many lines you had left. However, when you join a sketch, it doesn't tell you any of this.

Solution: We have fixed this so that both users joining and creating sketches will see the 10 line count down system (see Number 1).

5. [H2-1 Visibility of System Status] [Severity 3] After a user draws a new line, the check and X marks in the resulting popup are ambiguous. It is unclear what they do.

Solution: Instead of requiring users to approve every line, lines are automatically approved. This interface has reduces friction in the drawing process. We also added a clearly labeled undo button on the bottom of the drawing board.



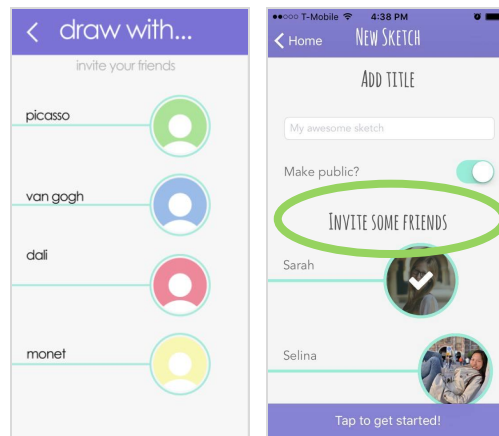
6. [H2-4 Consistency & Standards] [Severity 3] When accepting an invite, it's unclear who is which circle in the top right. It seems more like it's the friend, since they have a speech bubble with the invite. However, when you make a new game, you are very clearly the white circle and your friend is the red one. Which is right?

Solution: Both are right. The original intention is that the currently-drawing participant's icon will be the big white circle and other players will be the smaller circles. We believe this will

be more obvious is a functional prototype. However, this is not implemented in the high-fi prototype due to time constraints.

7. [H2-4 Consistency & Standards] [Severity 3] How are names organized when you invite them? I would like to be able to organize by order of people I've recently played with or played with most. This would turn into a major problem if a user has a large amount of friends (on the order of hundreds).

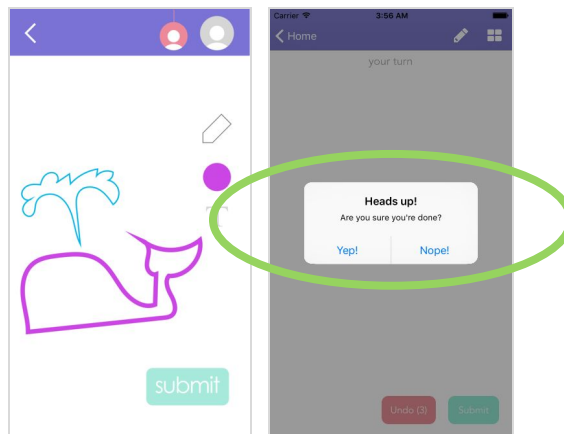
Solution: We decided to sort the friends list by who you most recently drew with. Friends you haven't drawn with in a while will be at the bottom of the list.



Friends are automatically sorted by "draw most often with" to "draw least with"

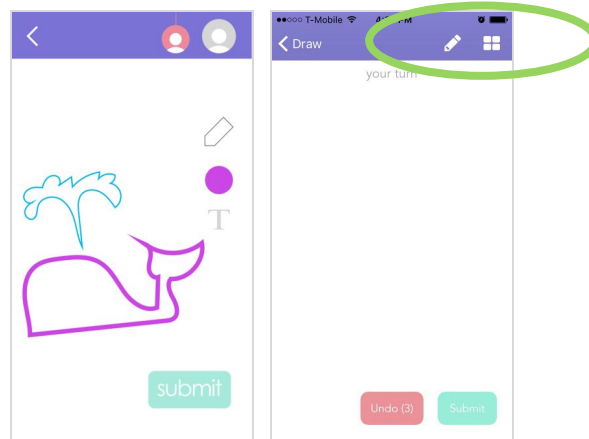
8. [H2-5 Error Prevention] [Severity 3] It is unclear what would happen if a user was drawing a line and accidentally drew their line over the submit button. Given the size and placement of this button this seems like a distinct possibility and could prove frustrating for a user if not handled gracefully.

Solution: To prevent accidental submissions, a warning popup is shown when the user hits submit. The user must then confirm that they are done in order to proceed.

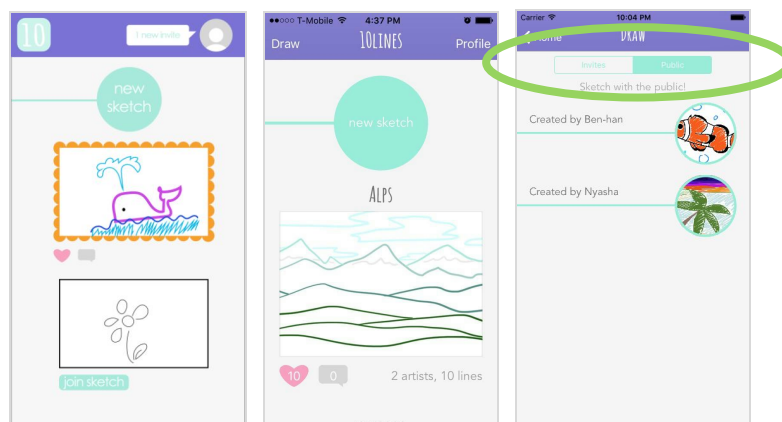


9. [H2-8 Aesthetic and Minimalist Design] [Severity 3] With the drawing tools placed on the right side of the drawing space and the submit button taking up a large amount of space in the bottom, the drawable area feels a bit squished into the upper left. This can feel claustrophobic especially on a mobile device with a small screen.

Solution: We moved the drawing tools from the drawing space to the navigation bar.



10. [Other self critique] One thing that we noticed while critiquing our own prototype was that the thumbnails themselves could be made much larger and be easier for users to see given that a phone is already not very wide. We originally also had the idea of differentiating between finished and unfinished sketches using gold frames to denote finished ones and black outlines to denote unfinished ones. However, having them in the same newsfeed was a bit confusing for users. We thus decided to split them up by having all the finished ones appear in the home screen/newsfeed upon opening the app and having the unfinished ones accessible via the “Draw” button in the upper left. The unfinished sketches in the Draw tab are filtered by a toggle between sketches between a user and his/her friends, and public sketches.



PROTOTYPE IMPLEMENTATION

Tools

We implemented all 3 tasks flows above in a high-fidelity prototype, which is a real iOS app made with Xcode 7. Development involved a combination of using Interface Builder to create the application interface and programming in Swift 2 to wire up items in the interface to real actions. Selina was the lead UI designer and created many assets like button backgrounds, logos, and so forth using Adobe Illustrator CC.

While Interface Builder is an extremely nice tool that abstracts away the difficulties of procedural GUI programming, interface builder (IB) files were not designed for collaborative work with Github. When multiple people worked on the application interface individually and then attempted to sync their work, IB files would conflict all the time and require time-consuming manual resolution. This was particularly frustrating because manual conflict resolution was difficult enough to block some of us from working for extended periods of time.

Swift is an very modern programming language, and in that sense it was extremely nice to work with. However, it still had a high learning curve, especially for some team members who haven't yet taken programming courses at the CS107 level. Achieving just the level of interaction present in the medium-fi prototype took a lot of effort and time on everyone's parts.

Hard-coded data and Wizard of Oz techniques

There is no backend distributed with our prototype so all data that would be served up by a backend (sketch comments, sketch pictures, user friends, and so forth) are hardcoded.¹ Correspondingly, the actual restriction of drawing one line at a time is not implemented because it is difficult to simulate turns without a real backend to serve lines from other participants. For usability testing, turns can be wizard of oz'ed by passing a single phone around.

An app that keeps track of per-user friends and per-user sketches will also obviously require user accounts, which in turn requires some kind of signup/login UI. Typically, a signup/login screen is shown the first time an app is started and then never shown again after a user has logged in. This prototype represents the usage of our app in the scenario where a user is already logged in, i.e. a signup/login UI is not shown and therefore not implemented.

Future work, uncertainties

¹ Note: For the project fair we had a second version of this prototype with a real (but simple) backend built with Rails. You can find the source for the second version and also for the backend at <https://github.com/bhnascar/10-Lines> under the "realtime" branch. Because this version is much more complicated to set up and run (requires installing Rails, having a laptop to act as a server, and at least two real iPhones to talk to each other), we are not using it as our official submission for this assignment.

Obvious next steps would be to implement the back end that is missing from the prototype, so that users can save sketches they've drawn, actually draw sketches with other users (1 line at a time, of course), and browse through new sketches in real-time as they are finished.

There are some big questions here such as the best technology stack (Rails vs. node.js), how to synchronize drawings in realtime (WebSockets vs. long-polling vs. short-polling), where to host the web server (Heroku vs. Linode), and more. To implement the back end well, we will have to give a lot more than a week's worth of consideration to all the aforementioned questions, and put in a lot of programming hours.

SUMMARY

In summary, this quarter has been a whirlwind of learning and implementing the design process in a very rewarding way that we experienced during the final project presentations. Although we started off with a rocky start and explored very different paths, we ultimately evolved 10lines from an idea to an interactive prototype that is for the most part intuitive, fun, and impactful. The feedback from our users, our studio, and our visitors/judges last night was invaluable and we sought to incorporate the best of that into our prototypes. We've come a long way and are so honored to have been awarded 2nd in the Best Overall Project category (in addition to feeling flattered by people asking to be added to our beta list). A special thanks to you, Stephany, for all your guidance and support!!