Traverse

High-Fidelity Prototype README

Katherine Wang | Michelle Buyan | Samuel Tong | Saniya Mahate

Introduction

Unlike traditional art discovery platforms or museum apps that emphasize static catalogs or linear browsing, Traverse offers an interactive and exploratory way to experience art. Instead of scrolling through lists or strolling through galleries, users navigate a living graph of interconnected artworks, reflections, and ideas, encouraging open-ended discovery driven by curiosity rather than search.

Traverse is designed for casual creatives and art enthusiasts who want to find connection, meaning, and community in their art experiences without the pressure of performance or exposure.

The high-fi prototype supports three main tasks:

1. **Explore the Artwork Graph**
    - Navigate the graph
    - Enter an artwork page
2. **Share Reflections About an Artwork**
    - Read reflections
    - Choose a region on the image
    - Write a public or private reflection
    - Submit it and see it appear at the top of reflections

3. **Upload an Artwork**
    - Upload an image
    - Add a title/artist name
    - Tag up to three related works
    - Submit the new artwork and see an updated graph

How to Set Up the Prototype for Testing

Traverse is accessible on both iOS devices or Androids. We recommend using an iOS device, specifically an iPhone 16 Pro for an optimized experience.

1. Install Expo Go on your mobile device
2. Scan the QR code on the left.
3. On first open, wait for Traverse to load on your device.
4. Explore, connect, and create in Traverse!

Limitations

- The hi-fi prototype assumes and skips past the log-in authentication, which would store user information (i.e. algorithm, reflections, profile) in real life. Additionally, there is currently no user profile and the user profile button is deactivated, since all users of the hi-fi prototype are set to a default guest.
- Traverse has currently manually organized the artwork graph with its connections for the prototype, but an actual AI algorithm would analyze art similarities according to different metrics. In real life, this algorithm would result in the graph on the app to actively change and expand via further use of the app, and from machine learning.
- There are 24 artworks on the graph, due to the limitations of us manually forming connections between artworks.
- All users of the app are currently guest users who are automatically taken to the same starting graph screen and artwork point. This would vary in real life based on user and algorithm data.

- Zooming in and out on the graph was not incorporated for the prototype, since the graph was relatively small and zoom was very complicated to implement with the artworks being pressable buttons with further components and user tasks stacked within each artwork. The user would be able to more easily zoom in/out, and pan the graph in a fully-formed app.
- The graph metric options button currently is limited to show manually formed graphs for the prototype, whereas the actual app would utilise the AI algorithm to re-form and display new graphs.
- When viewing an artwork's reflections in the medium-fi prototype, the topmost visible reflection is assumed to be highlighted in order to show its corresponding region. In practice, this code implementation is nontrivial within React Native; in the high-fi prototype, users instead press on a reflection to highlight it.
- The AI segmentation model powering reflection regions is hosted on a dedicated server on Replicate; for cost efficiency, the server remains shut down by default until manually enabled. While the server is shut down, no new region selection requests can be made; existing regions can still be viewed.

Accessibility Considerations

- There are currently no limitations to what a user can upload to the graph, but to ensure safety and usability measures, Traverse manually moderates content to prevent inappropriate content, copyrighted content, duplicate content, reported content, and other types of content that would violate ethical considerations.
- While it is not in scope for the high-fi prototype, a production version of the app would utilize alt text and screen reader-friendly displays to be accessible for users with visual impairments or disabilities

Wizard of Oz Techniques

- Graph similarity is determined manually rather than through ML algorithms because it would require real similarity models.
- Uploaded content is local to the prototype's device machine (submission is not linked to the database). The upload is hard-coded to be displayed +

connected to the same part of the graph (so users of the prototype can only actively add or replace the same artwork). However, to prototype users, it seems like they are actively changing and adding content live to the graph.

Hard-Coded Techniques

- User profiles are hard-coded into reflections, with the guest user set as the default user with a fixed image (profile information cannot be changed or edited)
- A pre-populated database of artworks and reflections is utilized for the graph
- Graph connections are hard-coded instead of formed via a ML algorithm