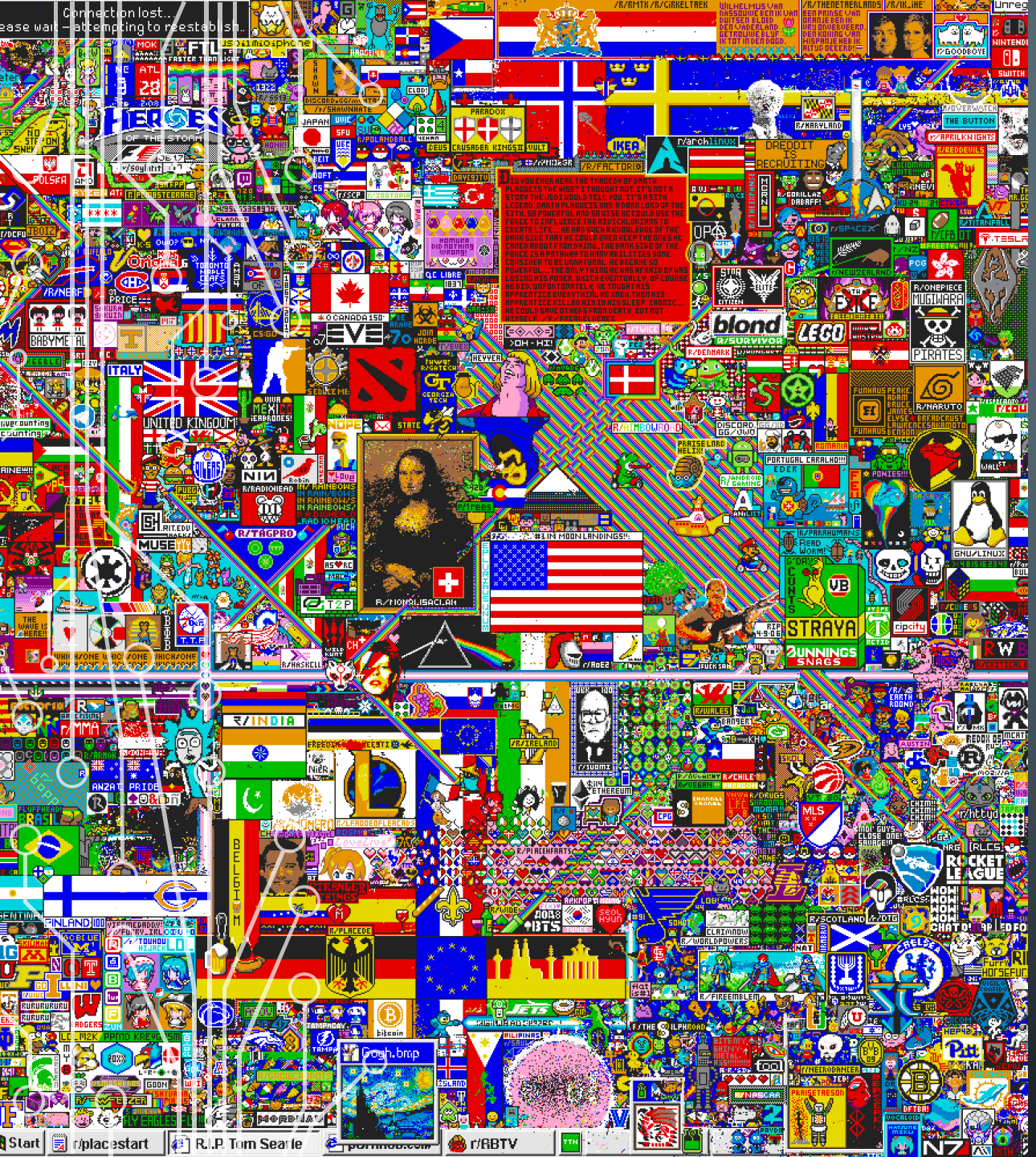
CANVAS

TEAM BIG BALLERS

NILAY SHAH, JAMES WANG, LUCA MATSUMOTO, LARI BOYMOUSHAKIAN



INSPIRATION

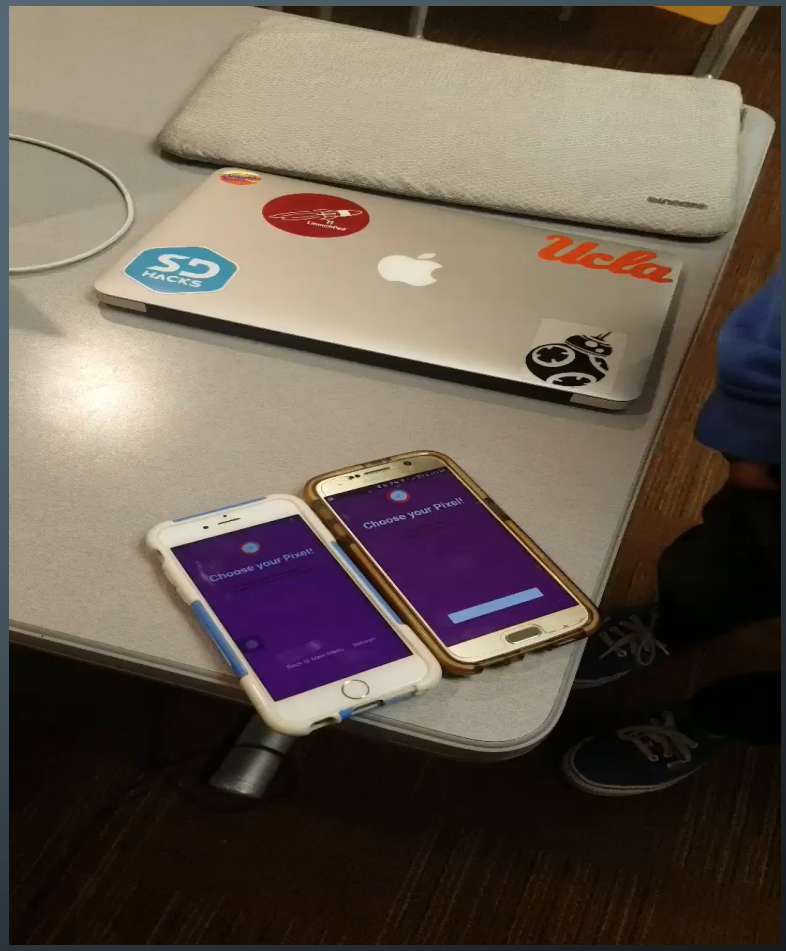
- Reddit's r/place event that took place in April 2017 which exploded into an international phenomenon garnering over 220,000 subscribers
- Became an open forum for people to display their creativity

FUNCTIONALITY

- Same concept as r/place, except instead of a big canvas for anyone on the internet, tie multiple canvases to different geographic locations
- Phone grabs a canvas associated with a location if it is in range geographically
- Canvases can be changed over time, and affected by users or groups of users with enough persistence
- Canvases persist in the database and can be loaded w/o modification to look at
- Timer countdown between modifications to prevent spamming changes

FUNCTIONALITY/DEMO

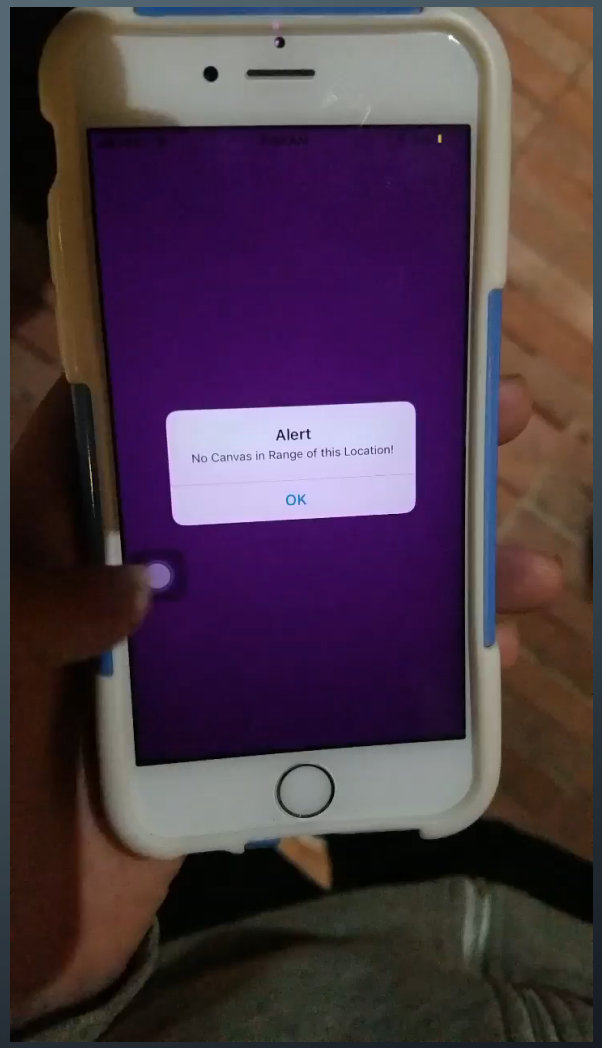
- Same location, same canvas, different users
  - Currently 5 locations across UCLA
- Canvas display identical within refresh of around 3 seconds
- Users can select a box by pressing on it and overwrite the existing color with a new one picked via a color picker screen



Two devices operating in same location. Location was preset for testing purposes.

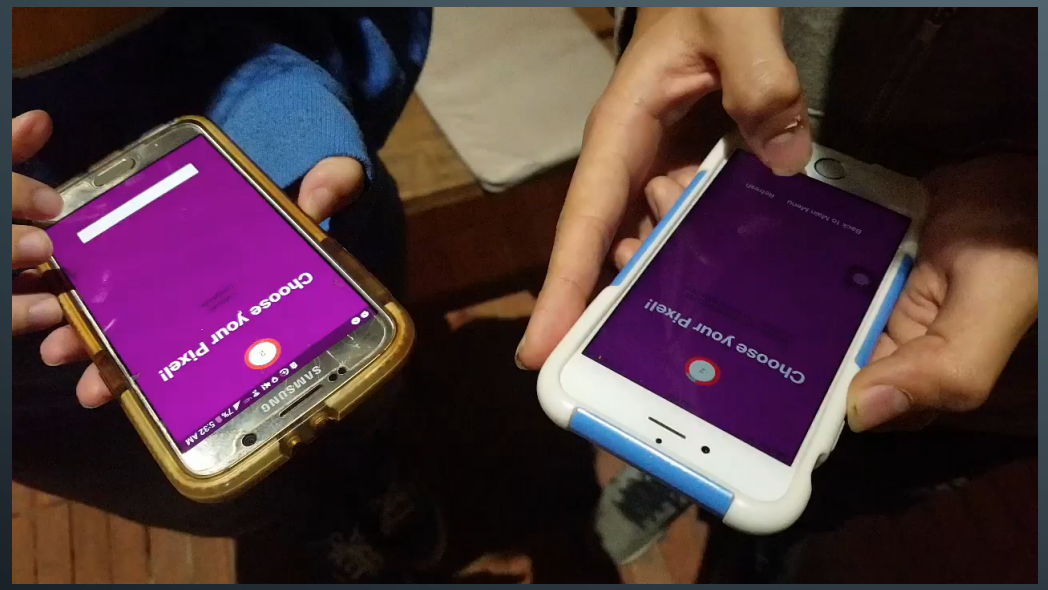
FUNCTIONALITY/DEMO

- Initial screen tells user to check for canvas.
  - If no canvas found, triggers simple alert and returns to the initial screen
  - Otherwise loads canvas associated with nearby location
- Returns to initial screen when on a grid and walking out of range

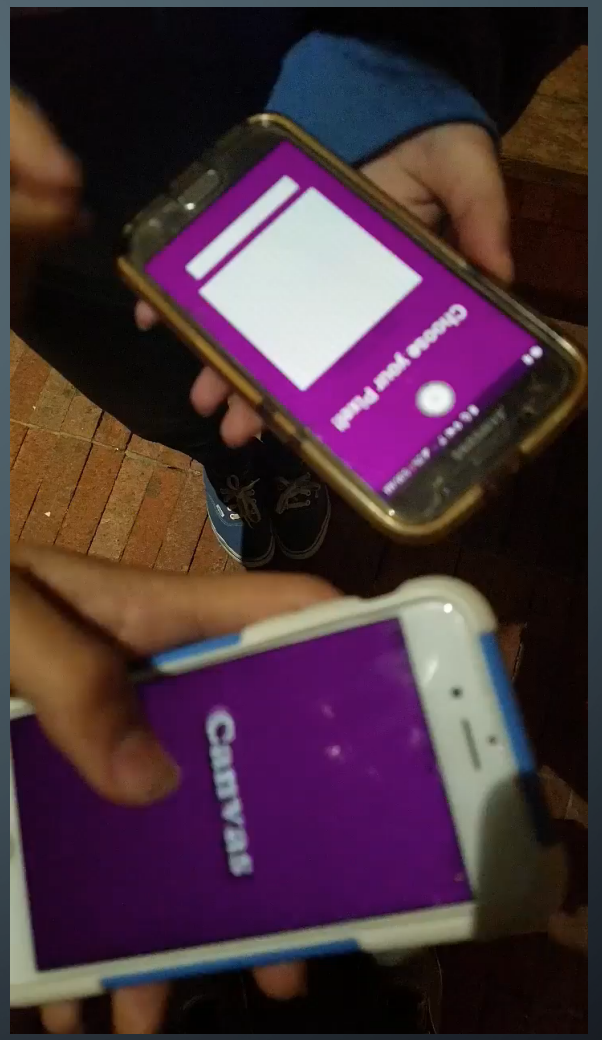


Simple out of range test

FUNCTIONALITY/DEMO



Two users working together on constructing blue tower on canvas associated with Sproul Hall



Initial blank canvas testing near Sproul Hall location

IMPLEMENTATION

- Front-End
  - Developed with React Native (ReactJS)
  - Two function API calls (DB lookup and DB update) to interact with back end
- Back-End
  - Developed with Ruby on Rails
    - Implemented our own GET/POST routing methods for our frontend-backend integration
  - Database built with PostgreSQL
    - Used to manage grids tied to geographical locations around UCLA

WIRELESS TECHNOLOGIES EMPLOYED

- IEEE 802.11 WiFi
  - A wireless network allows users to connect to our server and gain access to various community canvases. Required for user to repeatedly refresh canvas and check range conditions
- Global Positioning System (GPS)
  - React Native's Geolocation API enables our application to determine the user's location and match them with the appropriate canvas

CHALLENGES

- Manipulating states of components in React to properly render and modify the displays
  - Determining index of the pixel selected by user to appropriately modify its color parameter in the database
  - Updating the color of the pixels on the grid to reflect database entry representation
- New to technology:
  - First time working with either React or Ruby on Rails

RESULTS

- App currently works as intended by design spec
  - Canvas refresh could be improved but not bad
  - Database interaction, front end linked to back end, all complete
  - Front end iOS compatible, small hiccups on Android with non-canvas button colors

IMPROVEMENTS

- Make canvases appear more atomic for users by improving refresh rate
- Improve precision of geolocation (difficult to use latitude and longitude to define small locations for canvases)
- Display comparison of current user's location vs current canvas' location

WHAT'S NEXT

- User selection of Geographic Locations to expand our library of grids
- Allow users to save snapshots of the canvas at particular instances in time
- Stitch together various canvases into one large mural
- More direct peer-to-peer interaction: allow for multiple users to share pixel modifications and try to push them simultaneously

MEMBER CONTRIBUTIONS

- Nilay
  - Geolocation algorithm, React component to database grid index algorithm
- Luca
  - Front end (design + implementation + component color update), linking front end to back end, project setup and maintenance
- Lari
  - Front end (component color update), debugging, pair programming partner alongside James and Luca
- James
  - Project design, back end (API, database interaction + management), linking back end to front end, front end (async handling, debugging)