

Reddit Place Re-make

Group 9

Alex Rueb

Dillon Tice

Jakob Frank

Project can be found at <https://github.com/AlexRueb/PlaceFrontEnd> with instructions on how to get it running.

Objective

We set out on this project to make a close replica of Reddits 'Place' event, which turned out extremely well.

Summary

This website enables users to change the color of a single pixel on the canvas every 10 seconds. The canvas and its corresponding backend was designed in such a way that whoever is in control of the server can change the size as they wish before they make the website live. For our purposes, we made it a 100x100 grid. We accomplished this with only PHP, Javascript, HTML, and CSS. The API consists of 2 endpoints; one setter to update a given pixel and one getter to retrieve the most recent pixels that have been updated. The backend is a basic MySQL table with 100,000 entries, each having a unique ID and the canvas is represented in that order. We use front-end techniques to wrap a one-dimensional pixel list after 100 entries instead of rendering a 100x100 grid and storing x/y coordinates in the database. We did use Bootstrap for some small animations when loading the pages.

Similarities with /r/place and Background Information

The original implementation of Reddits /r/place was done in a single row of "pixels" as well. However, due to their use of a larger board (1000x1000) and heavier stress on their system, Reddit had to use redis and a bitfield of 1,000,000 4-bit integers instead of the more traditional database Cassandra. However, they kept a backup and additional metadata in Cassandra in case of redis failure or the need for inspecting tile placement information, respectively. With redis, it took 100ms to read the entire canvas in contrast to the 30 *seconds* it took with Cassandra.

Race conditions were not handled in Reddits /r/place. Users could potentially place multiple tiles before their cooldown starting if they clicked fast enough (nearly simultaneous, only achieved by a bot). This accounted for ~0.09% of the tiles placed in the event. For our

implementation, we did not see a need for specifically addressing race conditions as we did not think anyone would come to class with a click-bot.

Differences from /r/place and Background Information

We were only able to do basic stress-testing of our website with three users at a time. Thankfully, everything worked out okay as no one in-class during presentations voiced any issues with placing pixels. Reddit, however, planned to support an average of 333 updates per second (100,000 tiles every five minutes). However, Reddits /r/place had 80,000 users at its peak whereas we likely had 25 - 30 at most.

Caching was another implementation detail that we did not have to consider seriously. For Reddit, if many users were to refresh all at once, they would all request the board at the same time. This specifically can't be helped, but the server stress can be minimized using caching. Reddit used the Content Delivery Network "Fastly" for their caching needs.

For our frontend, we decided not to specifically implement mobile support but we did see that some users were able to use their phones to somewhat effectively place pixels. Also, Reddit had zoom and pan features which our website lacked in its initial release.

Dillon Tice

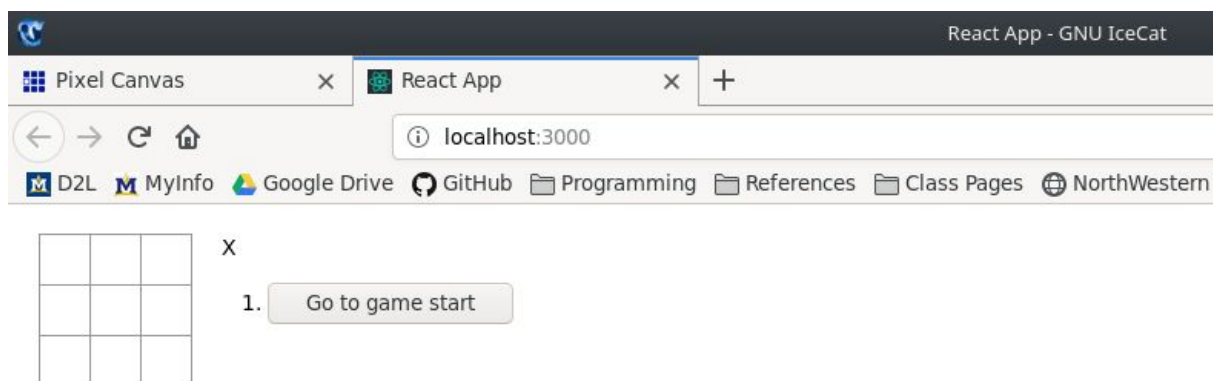
Almost everything I did with this project was in the backend. I created 2 API endpoints for the server: one POST method for setting a pixel and one GET for getting the most recent pixels as of a given timestamp. These were realized in the form of a pixel.php file which the index page made AJAX requests to. The database was as simple as creating one table for each of the possible pixels and storing their color and a timestamp of the last update. Overall, it was basically just an extension of the LAMP project, but it ended up working pretty well.

If I were to redo this project, I would definitely take a Reactive approach, since our current iteration does not work well with phones (it works, but the styling is weird). I would use a document storage solution instead of MySQL, and I would probably choose Golang as my backend language.

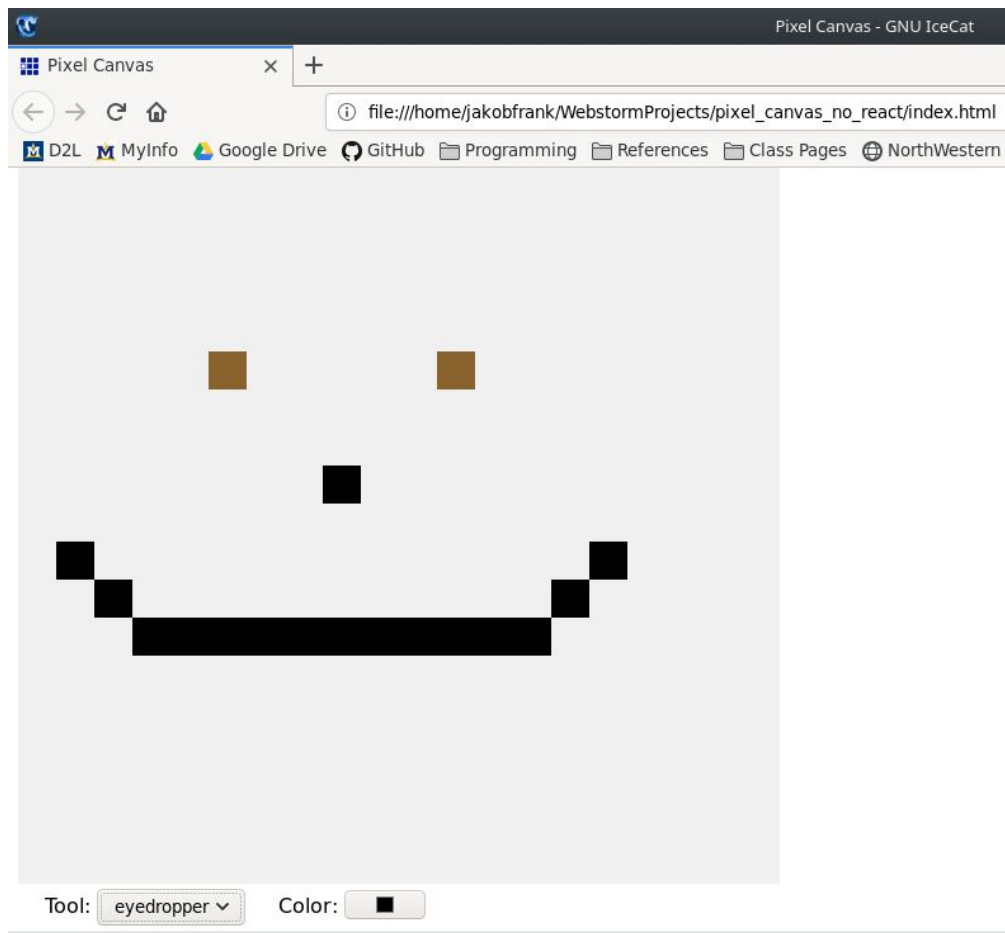
Jakob Frank

During our initial plans, we attempted to use React as it seemed to yield nice looking applications. However, we had issues and tried out pure JavaScript to create our pixel canvas.

This is our prototype in React which we never got working. Our goal was to expand the boards rows and columns to more accurately reflect /r/place but we ran into a lot of issues. It started out as a working tic-tac-toe board that was made in a React tutorial:



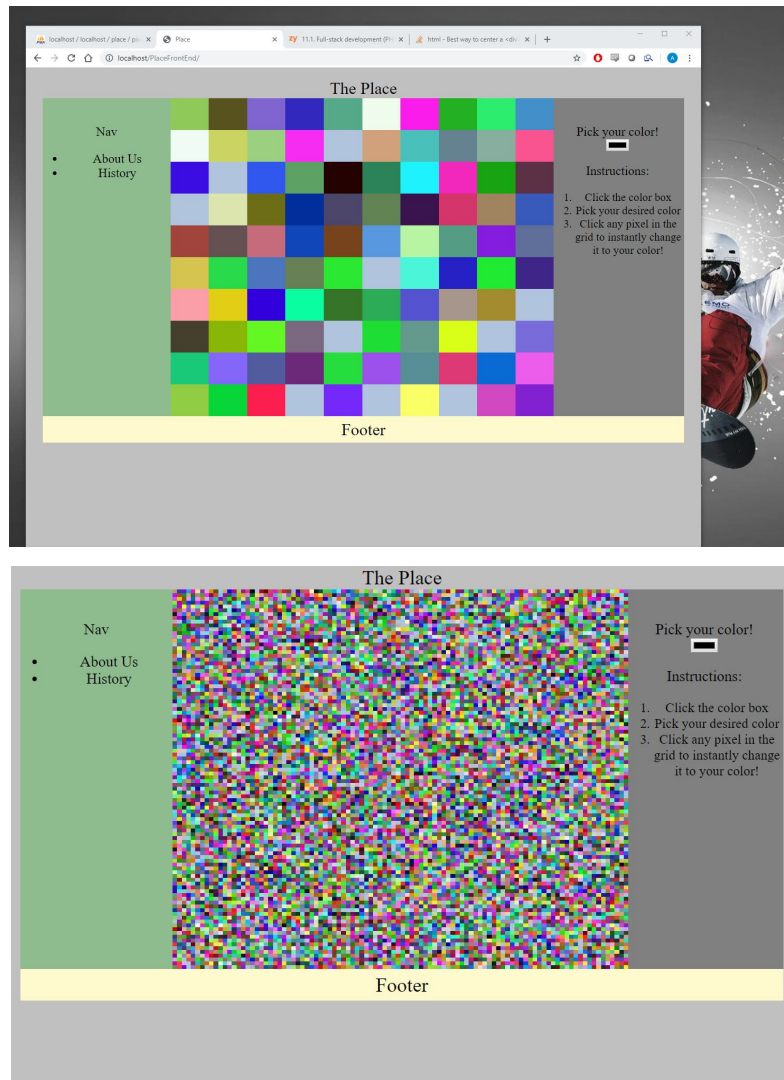
Here is my initial implementation of our pixel canvas in JavaScript:



We had multiple people working at once on this project which resulted in some code conflicts and problems with our database implementation so we started over from scratch using PHP. The PHP implementation is what we used for our current, third iteration of the project. Bootstrap enabled us to make a fairly nice looking website (much better than above). Dillon had the most experience with PHP so Alex and I consulted with him a lot about his implementation of the database and MySQL.

Alex Rueb

During this assignment, I got the initial Github setup and put all the files onto it, setup the initial wireframe of the website and got the first draft of the styling done. This was heavily influenced by the guides on ZyBooks and helped me get the hang of CSS a lot more. Below you can see the initial board when we had the less colors and our first ideas of what we were gonna do.



Later, I implemented the links to the 'About us' and 'History' pages, and styled them as well. We used bootstrap to make the animations and to make the page look the way we wanted, and it was actually crazy how easy it was to make work.

Conclusion

We learned quite a bit in our attempt to recreate /r/place. In our first iteration of the project, we tried to use React to develop our website. However, none of us had experience with it and had issues making something effective. We decided to fallback on what we are good at: basic, but highly effective HTML, CSS, JavaScript, PHP and a dash of Bootstrap. The project was smooth sailing after the switch and our biggest issue (which we didn't find a suitable alternative for when we presented) was our placement timer. The timer was done in JavaScript which enabled users to simply refresh the page if they wanted to place a pixel before their 10 second cooldown was over.

References

<https://cassandra.apache.org/>

<https://redis.io/commands/bitfield>

<https://redditblog.com/2017/04/13/how-we-built-rplace/>