vike/place

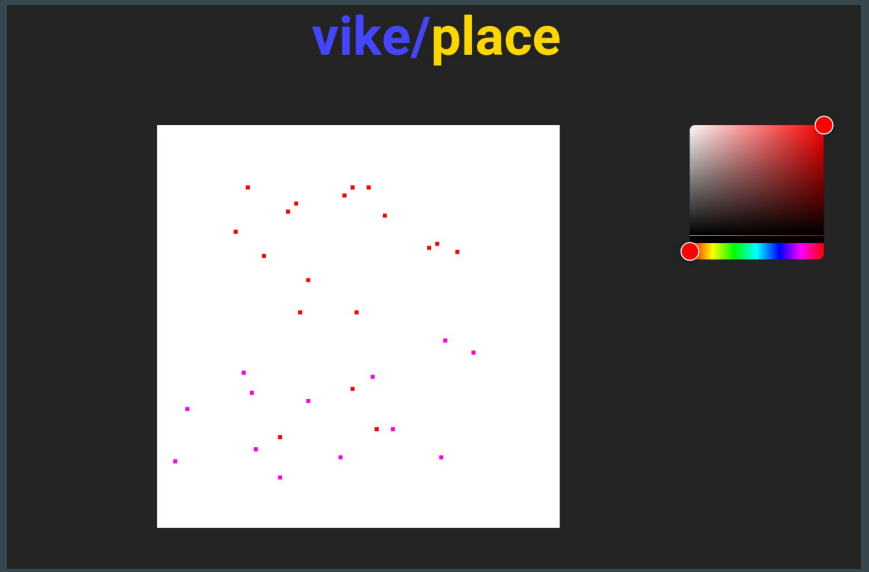
...

Inspiration

- We wanted our app to be BIG
- We were inspired by the largest real-time app we could think of, and we wanted to build our own version of it!
- r/place had 10.5 million users, and was only up for 3.5 days
- Users could interact in real time, which was a major factor in user enjoyment

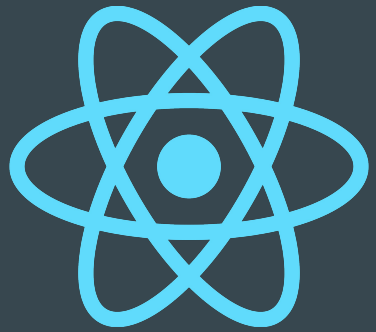
Our App – https://vike.place

- vike/place is a massively online realtime collaborative pixel art platform
- Users can engage in co-op or PVP, and claim parts of the map for their own
- No scoring is needed, it's a purely creative/territorial game
- No client-side state, tolerant of flaky networks



Tech: Frontend

- We used React to build our site, and bundled it to statically-hosted files using Vite
- Few libraries were used: Zod, p5.js, and react-colorful
- Thin-client approach, state is stored on the server to minimize complexity



Tech: Backend

- We used Rust with the Axum webserver on the Tokio runtime
- Tokio is meant for high-throughput low-latency applications with good vertical scaling
- Pixels are stored in-memory in a flat array, client messages mutate in-place
- Only the diff is sent to clients to minimize network traffic
- Users are rate-limited, so botting is expected (and almost encouraged)

Scaling: Current and future

- The frontend is deployed as static files, so it scales horizontally to the moon
- The backend can scale vertically very well. We will likely hit the Linux connection limit (on websockets) before the backend has any scaling issues
- With a caching service (think Redis), the backend could store no state and scale horizontally

Reflection & Future Work

What we did well:

- Used simple technology
- Kept the scope manageable

  ... which meant we could create an awesome prototype

What we would change for next time:

- p5.js wasn't the right tool for us; should've used HTML Canvas directly
- more time for integration

Thank you!

Questions!