Project Title

Nicholas Carignan

Pics here

- Purveyor of knowledge and wisdom
- Brings his passion for pizza and puppies to every party

Ben

Pics here

- Whiz at debugging
- Code Hero, fights the Code Bad- guys

Problem domain

We were planning a scavenger hunt for Halloween but wanted to make it themed for spookiness. It would be fun to play a giant game of clue scavenger hunt where we had to run around a few blocks collecting clues at pubs and parks.

We wanted an app that would facilitate the placement of clues in a radius around the start point at the most visible public places that appear in a “things near me” google search.

Implementation

We decided to build a mobile app to play clue on. We implemented google location api, yelp api, and the seattle.gov parks api to find notable locations and map them out. We focused on an easy and quick to use ui/ux to quickly start and play the game. And we host our games from a socket.io server to keep the connections running among all the phones.

Go to the project

Cluescavenger.com

Do a walk through, have fun, change things, make new users, etc.

Slides go here

Put pics here of your github project board (s).

Pics of cool code blocks

Pics of cool graphics you made in photoshop

etc

Some Challenges

Sharmarke and I fought over which api to use, He beat me in a game of ping pong which we agreed to in our team agreement, so we went with yelp api.

On day 3, we had split off from pair programming and both worked on app.js. We had a huuuuge merge conflict. We fixed it and added a better checkout process to our team agreement, we also separated app.js into more specific files.

We could not find out how to render our buttons and send clicks out easily. We read this cool article www.rendercoolthings.com and went with this workaround

Wins

We tested our app and can hit it with 10 million users.

We found out how to store state of the game and the game you are currently in in localstorage so you could leave and come back as long as at least one person is still playing