In this sprint I accomplished a basic, minimum viable product version of the app to test out some features and assess my design.

Currently, the entire app’s functionality is being handled through the React front-end. I lifted the state of the called numbers and win conditions to the main app.js component file and sent them as props to the respective children. It got messy to look at, but I was already planning on implementing access and modification of the data through an Express Backend/RESTful API, so that will be the next modification made at the start of the next “sprint”.

I must also add multiple-card functionality, though this will be easier after the back end is set up.

Currently, the app looks like this:

|  |  |
| --- | --- |
| A screenshot of a game  Description automatically generated  Default view, with a randomly generated card for demonstration purposes only (hence the duplicate 57 in the G column). The final product will only allow camera-scanned or manually entered cards. | A screenshot of a game  Description automatically generated  Adding win conditions |
| Multiple Win Conditions Confirmed | Numbers 11, 4, 33, and 50 called |
| After calling number 30 | After calling number 30 |
| After calling number 30 |  |

In future sprints I will continue to add functionalities including:

* Shifting current game-state management to backend storage and API access
* multiple card management
* manual card addition
* camera/ML recognition card addition
* more robust win conditions (two lines on the same card, 2 lines on different cards, etc.)

I also still need to migrate the web-app to an android app and improve its styling/readability.

Overall, I’m rather pleased with development so far. I’ve created a large portion of the final functionalities already, practiced and advanced my react development skills, and have achieved both validation that my design is appropriate and verification that my implementation is correct!