

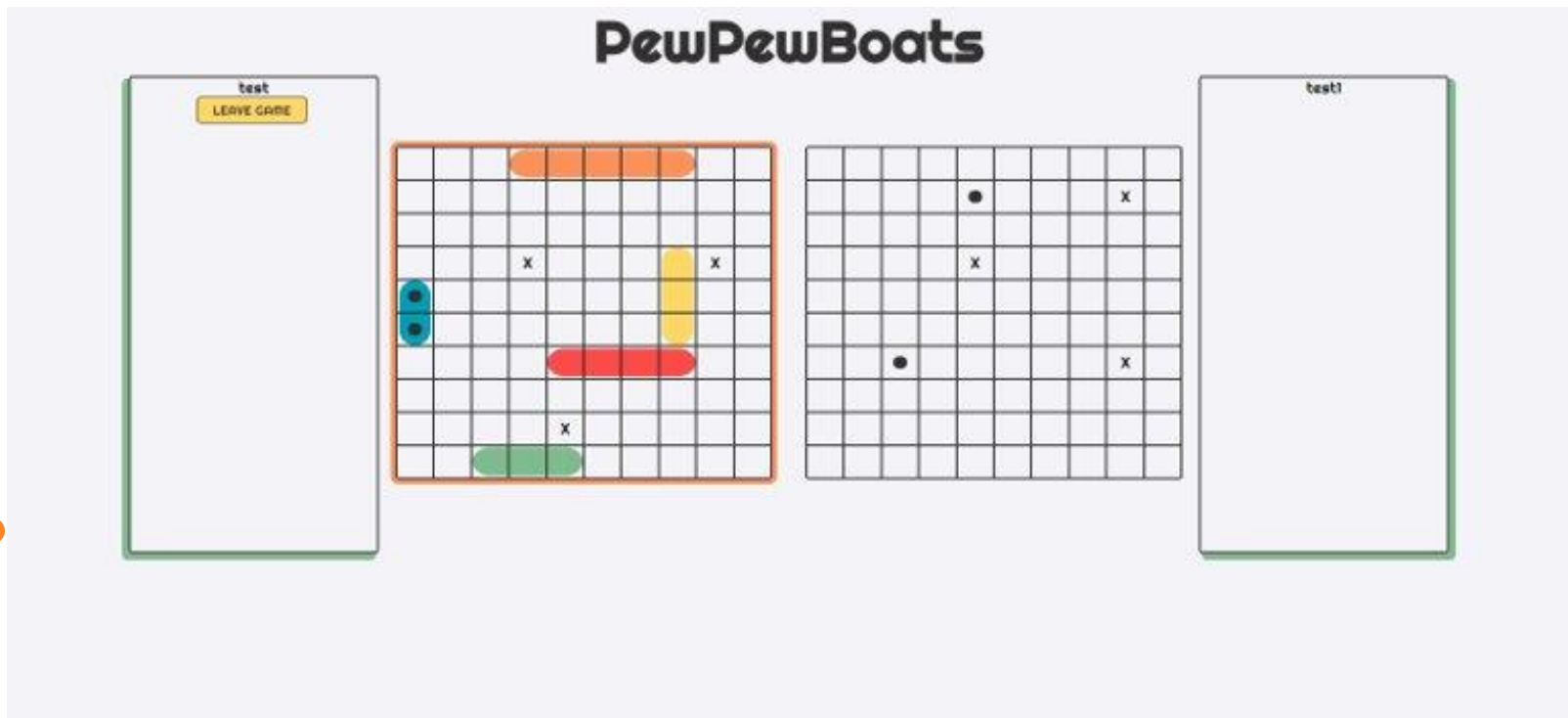


Senior Project

Curtis Flaming

Overview

- My own rendition of the boardgame, BattleShip
- Web app built with JavaScript (React) on the front end.
- Using serverless services from Amazon Web Services for backend.



Objectives


- To familiarize myself with serverless technologies
- Gain experience building a 'full stack' application.
- Build a user interface with reasonable level of complexity

Requirements

- User Authentication/Authorization with user sign-in/out
- Persistent game state
- Responsive UI
- Working game logic




Technologies Used

- AWS Amplify
 - Front End Language
 - React - JavaScript (HTML, CSS)
 - GraphQL
 - DynamoDB
 - AWS Cognito
- 



How it works

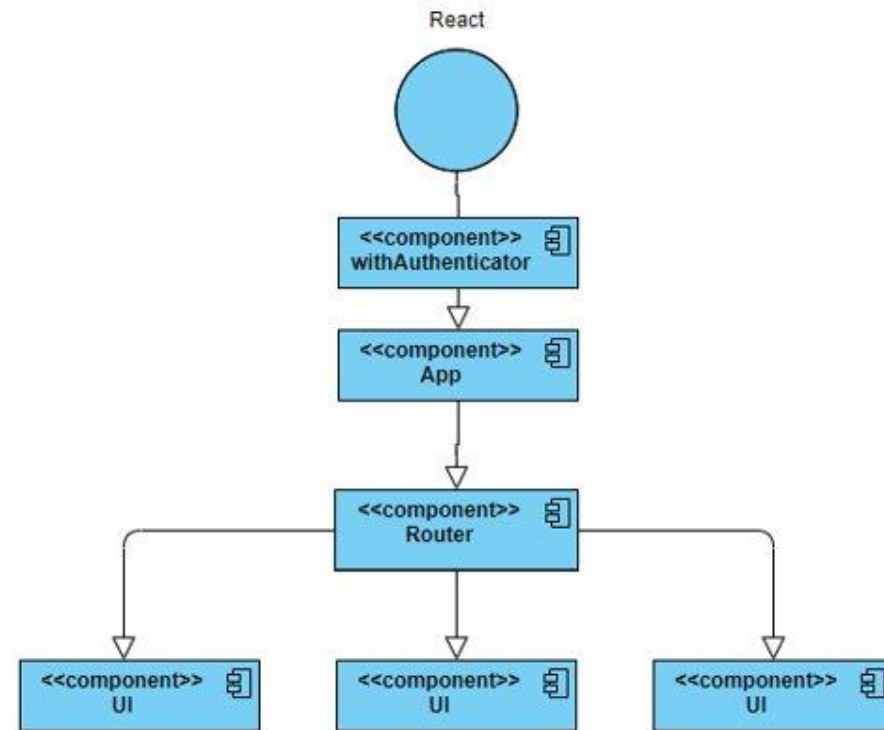
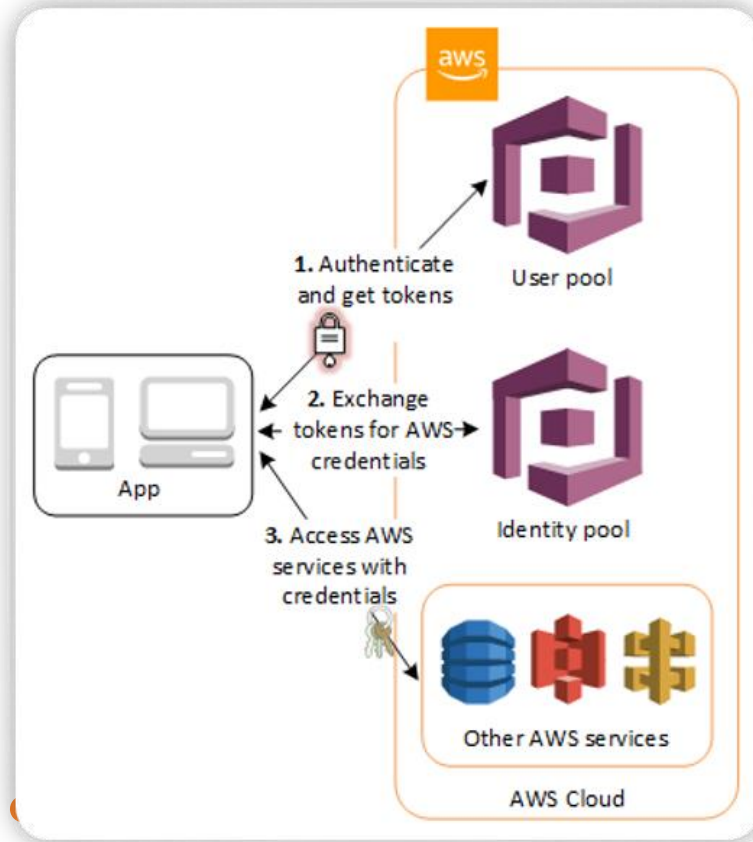
- Amplify
 - Set of tools which provide front end code for schema definition.
 - Provides code to perform queries/mutations/subscriptions.
 - Works with AWS AppSync to configure a GraphQL serverless endpoint on AWS.
 - Provides front end libraries to connect with Cognito
- 

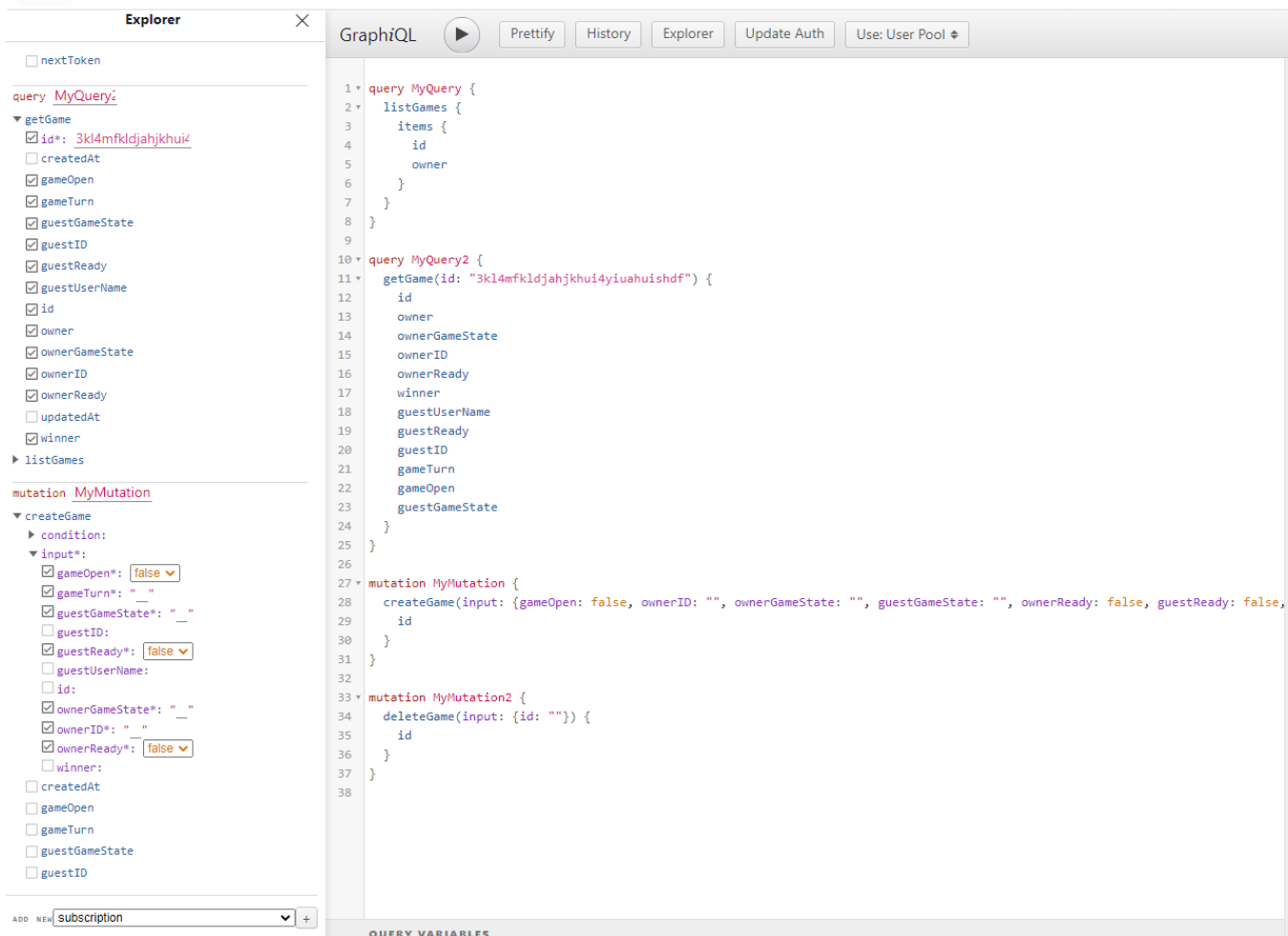
GraphQL

```
type Game @model @auth(rules: [{ allow: owner, operations: [create] }])  
{  
  id: ID!  
  gameOpen: Boolean!  
  ownerID: String!  
  guestID: String  
  guestUserName: String  
  ownerGameState: String!  
  guestGameState: String!  
  ownerReady: Boolean!  
  guestReady: Boolean!  
  gameTurn: String!  
  winner: String  
}  
  
type Subscription {  
  subToGame(id: ID!): Game @aws_subscribe(mutations: ["updateGame"])  
  subToGameDelete(id: ID!): Game @aws_subscribe(mutations: ["deleteGame"])  
}
```

- Define the schema
- Amplify will create a DynamoDB table on every @model directive
- Subscriptions are used to create real time user interaction with JavaScript observables

Cognito





Testing

- Front End Testing
 - Hand testing myself and with friends
- API Testing
 - Amplify Mock API




The image features a minimalist abstract design on a white background. A large, solid orange circle is the central focus, containing the word "DEMO" in white, uppercase, sans-serif font. Surrounding this central circle are several other geometric elements: a smaller orange circle in the top-left corner, a red line forming a right-angled corner in the top-right, another red line forming a right-angled corner in the bottom-left, and a small orange circle with three dashed orange lines radiating from it in the bottom-center. The overall aesthetic is clean and modern, using a limited color palette of orange, red, and white.

DEMO



What I learned

- Higher level JavaScript and React techniques
 - Familiarity with some of AWS many services
 - Experience with client/server relationship and real-time interaction on a single source of data
- 

Original Plans and Difficulties

- Original/future Plans
 - More rich boat-themed web game, not just a 1:1 rendition of BattleShip
 - Mobile design included
- Difficulties
 - COVID
 - Figuring out all of the HTML drag and drop API features
 - Learning how to use GraphQL
 - Syncing clients without overwriting client states
 - Antivirus software denying web socket connection



Questions?