**24 Game**

Summary/ Rules

* Present a graphic with 4 numbers on it
* *Counter starts (90 seconds)*
* First person to get the right equation gets it
* Keyboard has all the operators, parentheses & numbers
* 10 cards in a game

Players:

* Can play by oneself
* Or in competition mode
* Competition mode:
  + Start Game
    - Public
    - Private
      * get a code
  + Join Game
    - Public
      * If no one else, can “start playing”
    - Private
      * Give a code

Models:

* Game
  + Counter 0 through 10
  + Players
* Card
  + Belongs to Game
  + Numbers 1 - 4
  + Winner, as user
  + *Difficulty*

Players

* + Username
  + Points

Data flow:

* Show Home Page
* Press to start Game
  + Open a Game on backend
  + Append player to Game, with name and points
  + Return an id
* Press to join Game with ID
  + Append player to Game on backend
* Show Game Page
  + Continually *fetch* players of this game, including their points
* Press to Start Game
  + Open a new Card on backend
* Show Card page - Play
  + Continually *fetch* for winner of this game, set as Winner
  + If 24, post to Card there is a winner, with current User
* Show Card page - Winner
  + Congratulate Winner
  + Set Points to backend (if winner is you, get 10 points) to GAME
* Press Next Card
  + Tick up Card Counter
* Game Component –

TO DO:

* Create sessions / authentication of some kind; each session has to be associated with one user, so that this user can be recognized as a new vs. duplicate player in any room they go into, and retain identity after refreshes
* Within Lobby.js –
  + Once you select to Join a game:
    - validate if existing game exists
    - then create a new player associated with that Game (does this count as a game update?? Maybe not. Should also listen for player updates for that game specifically)
  + Start a new game:
    - Create new game
    - Create new player associated with that game (same as above, will this be an update?)
  + Game play…
  + Enable leaving / signing out
  + How to get the numbers? Just start with 1234 for now

(Machine learning?)

1. LOBBY
   1. Loads all games – private and open
2. Player enters **LOBBY**
   1. Join game
      1. (private only) – check if that game exists
      2. Prompted to enter playerName
      3. Create new Player for that game; validate uniqueness of name
      4. Resulting name & id for player are stored in sessionStorage
   2. Start game
      1. Capture playerName
      2. Create game on backend & broadcast
      3. Create new Player on backend
      4. Add id & name to sessionStorage
   3. **GAME**
      1. Load players, count, active card
      2. Player clicks “ready to play”
         1. Send update to backend
            1. Update points, update counter
         2. Back end sends out refreshed players & counter
            1. setState
         3. Backend also checks if this leads to “all players ready”
            1. If so, create new Card, and broadcast

setState

* + - * 1. If this is first card, also tick up the counter, and blast out

SetState

* + 1. Load **Card** from state
  1. **CARD**
     1. If someone reaches 24, then
        1. Send to backend as an update to the Game
        2. Backend calculates points, returns new card counter + updated players, who are adjusted to be “not ready”
        3. Front end captures in state
           1. Ready? buttons made visible
        4. See above 2.c.ii