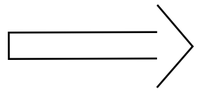
sign in:

- store “name” and “nickname” fields in variables
- run addUser() with name and nickName as args
- ***loop*** over userList, check ***if*** nickName is taken. if taken, confirm prompt to ensure correct nickName. if not, add new User object to userList. in both cases, break loop.

data generated/needed:

- needed: name and nickName from inputs
- generated: new User object with name and nickName filled in, update userList with new User



SIGNED IN

data on page:

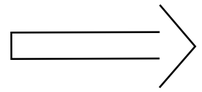
- show top 10: loop over user list highestScore property, return topTen array with 10 highest scores in order (highest to lowest).
- current user rank: loop over user list highestScore property, order highest to lowest, return number value for index position
- last score: loop over current user scores array, return most recent addition

user actions:

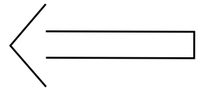
- click username: navigate to user profile
- start game: initiate Simon function. **loop** over 4 possible class names and pick 4 at random. increase the speed and increase amount of picks by 1 while user picks match simon picks (**do/while?**). when user pick fails, add the score to user object and display with game reset button.
- reset: initiate simon function.
- control pad: click action on a control button checks if current data-color matched the color class for the corresponding simon square in the simon array (positoins must match). save simon picks and user picks each round and compare the two arrays. every 3 intervals, increase point var by 2.

data generated/needed:

- needed: userList, currentUser name nickName highestScore lowestScore recentScore allScores
- generated: simon picks array, user picks array, new score



TO USER PROFILE



BACK TO GAME

data on page:

- header: user object name/nickName
- performance stats: show lowest score, highest score, most recent, top 10 (for user).
- current user rank: loop over user list highestScore property, order highest to lowest, return number value for index position

user actions: back button, navigate to game page

data generated/needed:

- needed: user name/nickName, lowest score, highest score, most recent, top 10 (for user), rank