

CJ

ColorPicker.js Element

+ Contains 12 colors from below color-wheel & black/white with a total of 14 different colors which the user can select.

+ When a color is clicked on the picker, it is highlighted in some way to indicate it being selected.

+ Contain a property/state "selected_color" representing the selected color. Starting at 0 for red going clock-wise on the color wheel.

Ex: 0 for red, 1 for red-orange, 2 for orange, 3 for yellow-orange, etc... all the way to 11 for red-violet. 12 for white, and 13 for black.

<https://www.usability.gov/how-to-and-tools/methods/color-basics.html>

Hemang

Chat.js Element

+ Display chat messages (use states & map?)

+ Display textbox whose text can be submitted by pressing ENTER

+ Use effect to receive socketio emits from server:

- "chat_update"
- contains username & new chat message

+ Have a property/state "username" containing string with logged-in username.

+ When ENTER is pressed while textbox is selected, emit a socketio "chat_submit" containing username & message.

Note: When ENTER is pressed, the only thing that should be done is to emit a socketio "chat_submit". The messages should not be updated. This will be done when a "chat_update" emit is received.

+ Have a property/state "enabled" that when True, allows the user to select the textbox and submit messages, and when False, disables the textbox and does not allow message submission.

Naqeeb

Canvas.js Element

+ Have a property/state "data" containing pixel color for each pixel on board. Also have variables for canvas dimensions in pixels.

+ Draw pixels on a HTML canvas based on the "data" property/state.

+ When board is first loaded, emit a socketio "canvas_request" which will request canvas state from the server (wait for app.py to respond with "canvas_state" emit. show a loading circle while waiting).

+ (After sending "canvas_request") Receive socketio emit "canvas_state" which contains board settings, and the color of every pixel. Update/set "data" state w/ this information.

+ Receive socketio emits "canvas_update" which contains information on a pixel and its new color. It also contains time.

+ When user clicks on canvas (when enabled) emit a socketio message: "canvas_set" which contains information on a pixel and its new color. It also contains current time.

+ Have a property/state "enabled" that when True, allows user to click on board with a color selected to modify the board. When False, prevents any modification of the canvas and will not emit any changes/actions.

