DESIGN FINAL

ANNIEJ
MARQUEZ
OLGASH

DESIGN OVERVIEW

Start up the WhiteboardServer and one or more Artists. The Artists are login screens that ask the user to connect to an IP. After connecting, the user has the option of making a new whiteboard with a name and background color, or joining an existing board. Both of these open a Canvas (with respective parameters). From there, the user experiences synchronized drawing across all users connected to that particular Whiteboard.

FEATURES

- multiple users can draw concurrently on one whiteboard
- multiple whiteboards can be opened in parallel
- can switch from a whiteboard to another whiteboard without reconnecting
- artsy meter indicates level of artsiness of current whiteboard
- can export images drawn to png file
- can post images drawn to the user's Facebook wall

CHANGE LOG [CHANGES SINCE MILESTONE]

- improved automated testing documentation for client/server interaction
- added UI tests (only test constructors)
- added switching boards without disconnecting
- added ability to export images
- added Facebook posting support
- fixed concurrency bugs
- added duplicate whiteboard name handling

DESIGN FINAL

ANNIEJ
MARQUEZ
OLGASH

DATATYPE DESIGN

- Server-side:
  - WhiteboardServer
    * has a list of Whiteboards
    * sends and receives messages to/from clients, creating and modifying Whiteboards
    * these messages include: selecting a whiteboard, making a new white-board, drawing, changing background color, users entering/exiting
  - Whiteboard (ADT)
    * has a unique name with no spaces
    * holds history of all actions done to it
    * calculates artsy meter

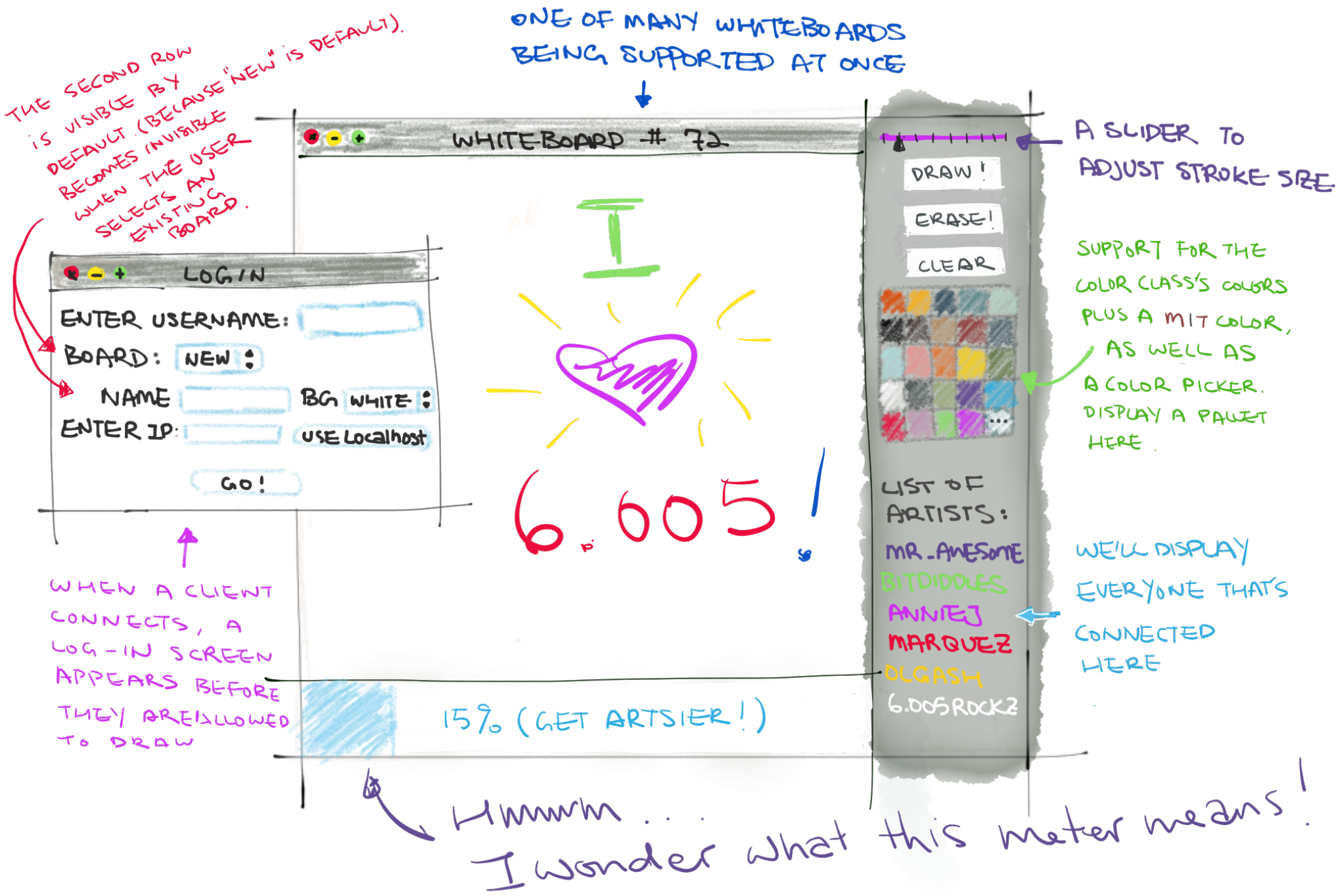
- Client-side GUIs:
  - Artist (login screen):
    * must enter a valid IP in order to connect to server
    * user can then select whiteboard from list, or create a new one (this opens a Canvas)
      - to select an existing one, user only needs to enter username
      - to create one, must enter username, board name (if this isn't unique, we will append a (1) and then (2) and so on to the end), as well as a bg color
      - the dropdown of existing whiteboards updates with new additions

DESIGN FINAL

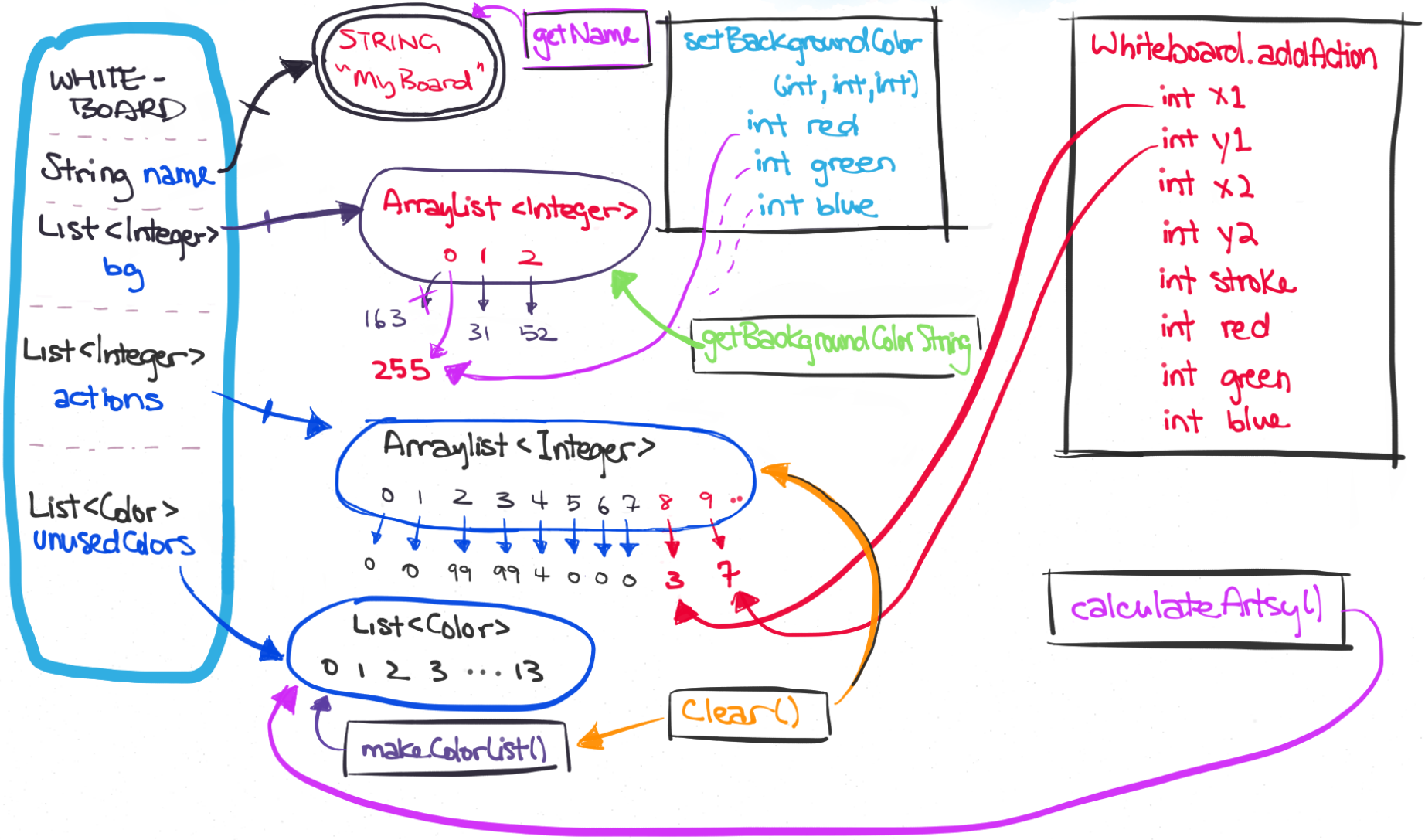
ANNIEJ
MARQUEZ
OLGASH

DATATYPE DESIGN [CONTINUED]

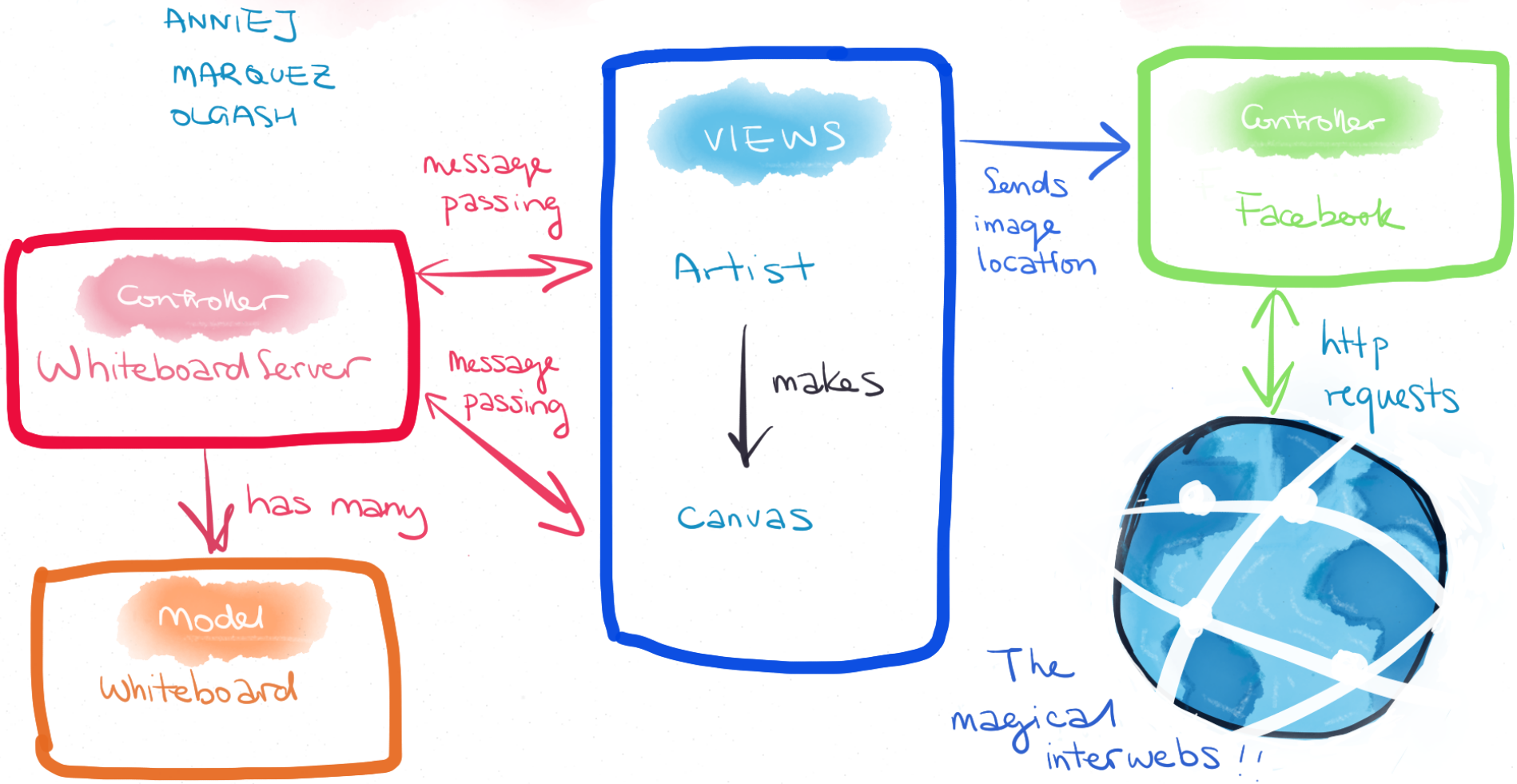
- Canvas
  * initially sends a create message to server and gets back a history of all actions and users
  * sends and receives new draw actions and users connects/disconnects to/from server, displays them accordingly
  * side panel provides tools that allows client-controlled adjustment
    - a slider that changes the stroke size
    - a pallet that holds all of Color's colors (with custom options). These colors can be dragged onto the Canvas to change the background color
    - a list that shows the usernames of collaborators
    - a button to erase and a button to draw
    - a button to clear everything from the board
    - a button to draw doge
    - a button to export the drawing
    - a button to post drawing to the user's Facebook
  * artsy meter at bottom of screen (whiteboard specific)
  * rest of the window is the drawable whiteboard
  * on close, notify server of closing
- Facebook
  * uses Scribe and Restfb libraries
  * prompts user to visit a url to get user's verification token
  * from there, finds the access token (OAuth)
  * posts images to user's facebook


THE SECOND ROW IS VISIBLE BY DEFAULT (BECAUSE "NEW" IS DEFAULT). BECOMES INVISIBLE WHEN THE USER SELECTS AN EXISTING BOARD.
ONE OF MANY WHITEBOARDS BEING SUPPORTED AT ONCE
WHITEBOARD # 72
LOGIN
ENTER USERNAME:
BOARD: NEW
NAME
BG WHITE
ENTER IP:
USE Localhost
GO!
WHEN A CLIENT CONNECTS, A LOG-IN SCREEN APPEARS BEFORE THEY ARE ALLOWED TO DRAW
6.005!
DRAW!
ERASE!
CLEAR
A SLIDER TO ADJUST STROKE SIZE
SUPPORT FOR THE COLOR CLASS'S COLORS PLUS A MIT COLOR, AS WELL AS A COLOR PICKER. DISPLAY A PALLET HERE
LIST OF ARTISTS:
MR_AWESOME
BITDIDDLES
ANNIEJ
MARQUEZ
OLGASH
6.005ROCK2
WE'LL DISPLAY EVERYONE THAT'S CONNECTED HERE
15% (GET ARTSIER!)
Hmmm... I wonder what this meter means!

ADT SNAPSHOT DIAGRAM



SYSTEM DIAGRAM



DESIGN FINAL

ANNIEJ
MARQUEZ
OLGASH

METHODS

- Client (GUIs)
  - Artist
    * toggleNewWhiteboard(boolean visible) - controls whether new whiteboard inputs are visible. If the user does not want to make a new whiteboard, hide the enter whiteboard name and choose background color options.
    * addListeners() - to split the GUI into methods. This adds listeners to the buttons/text fields/combo boxes
    * containsSpace(String input) - returns true if a String contains a space. This is to provide appropriate error popups for when usernames and whiteboardnames contain spaces.
    * handleInput(String input)- responds to server's requests
    * makeNewWhiteboard(String userName, String whiteboardName)- attempts to make new whiteboard
    * makeColors()- creates a color map
    * addWhiteboard(final String boardName)- adds a new whiteboard name to the UI
    * setupWhiteboards()- sets up whiteboard UI things, adds the new whiteboard option
    * startConnection()- spins up threads that listen for server input and return output

DESIGN FINAL

ANNIEJ
MARQUEZ
OLGASH

METHODS [CONTINUED]

- Client (GUIs) [Continued]
  - Canvas
    * setupButtons() - separate the GUI into shorter methods; compartmentalization
    * addListeners() - similar to Artist;
    * setArtsy(int) - determines artsiness of the whiteboard using a hand-picked set of mysterious and magical criteria
    * addRemoveUsers(String, boolean) - adds/removes a username from the JTable of users currently working on the whiteboard
    * setupWhiteboard() - communications with server; Send the server the message to create/select a whiteboard, if it's a new whiteboard, nothing is returned; if it's an existing one, set the background color and users list and draw the actions.
    * parseActions(String, boolean withArtsy) - takes in a string representing actions and draws them. The boolean withArtsy lets the method know if a Artsiness value is expected with the string of actions
    * fillBackground() - this was changed from the staff code of fillWithWhite(), because we would like to initialize a whiteboard with client-chosen background color
    * doge() - creates a welcoming image on the default board
    * handleRequest(String) - this method responds to the server's inputs by parsing the string.
    * exportImage()- exports the image drawn to a PNG file, saved via a file chooser

DESIGN FINAL

ANNIEJ
MARQUEZ
OLGASH

METHODS [CONTINUED]

    * saveFileChooser()- runs the file chooser
    * paintComponents(Graphics), makeDrawingBuffer(), drawLineSegment(), addDrawingController() - these methods are provided by staff

- Server
  - Whiteboard
    * makeColorList() - a helper method for determining artsiness
    * getName() - returns a string representing whiteboard name
    * addAction(8 ints) - adds two pairs of coordinates, stroke size, and rgb values into the list of actions. Also updates the artsiness calculations
    * getBackgroundColorString() - returns the background color's RGB values in a string
    * setBackgroundColor(int red, int green, int blue) - changes the background color. This is synchronized for concurrency.
    * createStringOfActions() - returns a string representing actions performed on the whiteboard. Each action is represented by "x1 y1 x2 y2 [stroke] R G B".
    * calculateArtsy() - determines the artsiness of the board
    * clear() - clears the actions on the board
  - WhiteboardServer
    * createWhiteboard(String, int, int, int) - make a new whiteboard and add it to the list of existing whiteboards
    * selectWhiteboard(String, String, int) - chooses a whiteboard for according to the Client based on the board name, their username, and their client ID

DESIGN FINAL

ANNIEJ
MARQUEZ
OLGASH

METHODS [CONTINUED]

* createListOfActions(String) - chooses a whiteboard and calls their creatEListOfActions method. For descriptions, see above
* listWhiteboards() - returns a string of all whiteboard names separated by spaces
* listUsers() - returns all a string of all usernames separated by spaces
* changeBackgroundColor(board name, RGB ints) - changes a whiteboard's background colors based on RGB values
* draw(boardName, x1, y1, x2, y2, stroke, RGB) - returns a string of the draw action
* clear(boardName) - clears everything from the board
* putOnAllQueuesBut(int clientID, String boardName, String message) - puts the message on all the queues of clients, except the specified client
* fixDuplicate(String name)- fixes duplicate names by appending a (1) at the end, and then, if there is already a (1), change it to (2), and so on.
* serve() - runs server, listens for and handles the connections.
* handleRequest(String input, int clientID)- reacts to client requests appropriately
* handleInput(Socket, int), handleOutput(Socket, int)- spin up threads that listen to client input and return output

– Facebook

* publishImage(String name)- publishes an image to the user's facebook wall

DESIGN FINAL

ANNIEJ
MARQUEZ
OLGASH

PROTOCOL

- Server - > Client
  - a list of whiteboard names (“LIST” WB_NAME WB_NAME...)
  - new whiteboard (“NEWBOARD” WB_NAME)
  - commands to catch new users up to speed when they join an existing whiteboard (background color, artsy meter, users, history of actions) (BGCOLOR_R BCOLOR_G BGCOLOR_B ARTSY_METER “USERS” USER_NAME USER_NAME... “ACTIONS” X1 Y1 X2 Y2 STROKE COLOR_R COLOR_G COLOR_B X1 Y1 X2 Y2 STROKE COLOR_R COLOR_G COLOR_B...)
  - new draw actions to everyone connected to a particular whiteboard (“DRAW” ARTSY_METER X1 Y1 X2 Y2 STROKE COLOR_R COLOR_G COLOR_B)
  - new client joins (“NEWUSER” USER_NAME) to all but new client
  - new name of whiteboard (“NEWNAME” NAME) to original client, to ensure no duplicates
  - change background color (“BG” COLOR_R COLOR_G COLOR_B)
  - clear everything from board (“CLEAR”)
  - a client leaves (“BYEUSER” USER_NAME)
  - an Artist leaves (“BYEARTIST”)

DESIGN FINAL

ANNIEJ
MARQUEZ
OLGASH

PROTOCOL [CONTINUED]

- Client- > Server
  - initial connect message to request whiteboard names (“HELLO”)
  - select whiteboard and get back whiteboard state (“SELECT” WB_NAME USER_NAME)
  - make new whiteboard with color, name (“NEW” WB_NAME COLOR_R COLOR_G COLOR_B USER_NAME)
  - new draw actions (“DRAW” WB_NAME X1 Y1 X2 Y2 STROKE COLOR_R COLOR_G COLOR_B)
  - change whiteboard bg color (“BG” WB_NAME COLOR_R COLOR_G COLOR_B)
  - clear everything from board (“CLEAR” WB_NAME)
  - disconnect message (“BYE” WB_NAME USER_NAME)

ERROR HANDLING

- The GUI utilizes JOptionPane.showMessageDialog to provide pop-ups with specific error messages.
- Whiteboard name is taken: “That whiteboard name is taken. Please choose a different one!”;
- Whiteboard name contains spaces/is empty: “Whiteboard name cannot be empty and cannot contain spaces.”
- Chosen username contains spaces/is empty: “Username cannot be empty and cannot contain spaces.”

DESIGN FINAL

ANNIEJ
MARQUEZ
OLGASH

CONCURRENCY & THREAD SAFETY

- No objects are shared between any classes, and no mutable objects shared between instances
- All messages sent are sent as strings
- No sockets shared, only IP addresses
- ID numbers for clients will use AtomicInteger as incrementing counter
- Whiteboards are treated independently and do not share information
- Draw actions and add/remove user actions are synchronized on all levels, so no information can be lost/overriden
- Drawing only happens when a message is received from the server, no local drawing, so all users on a whiteboard see the same thing
- Clients and server have blocking queues that processes messages
- Only call UI repaint in Swingutilities.invokeLater()
- To ensure whiteboard names are unique, they are sent to the server first and in a synchronized block, the server ensure the name is unique and creates a white-board and then sends the client the name back

DESIGN FINAL

ANNIEJ
MARQUEZ
OLGASH

TESTING STRATEGY

- ADT tests (test public methods of Whiteboard)
  - make sure on initialization, the whiteboard has an artsy meter of zero, and no actions
  - test the name getter
  - test the background color setter and getter
  - test adding new actions and getting them in string format
  - test clearing all actions
  - test the artsy meter increasing when new colors are in actions, but not when custom or repeated colors are added
  - test that the artsy meter returns to 0 when the board is cleared
- Automated testing for client/server interactions
  - test that a "HELLO" message merits a response with the list of whiteboards ("LIST" WB_NAME..)
  - test that "DRAW", "CLEAR", and "BG" messages get sent to all connected clients (but only those associated with the whiteboard)
  - test that "BYE" messages send "BYEUSER" messages to all but the client that sent the message, and that that client's connection gets terminated by the server
  - test that "SELECT" messages return a response with all of the board's history (empty or not) to the sender, and "NEWUSER" messages to everyone else
  - test that "NEW" messages return a response of "NEWNAME" to original client, and "NEWBOARD" to all Artist clients (also ensure duplicate names do not happen)
  - "BYEARTIST" should cause that client's connection gets terminated by the server

DESIGN FINAL

ANNIEJ
MARQUEZ
OLGASH

TESTING STRATEGY [CONTINUED]

- Client/server interactions, concurrency, and UI (manual testing)
  - first make sure we can connect via IP address (and localhost) and that populates the dropdown of whiteboard names.
  - make sure creating a new whiteboard updates the dropdowns of other Artists
  - make sure we can both create and select existing whiteboard with different names and bg colors.
  - make sure the Canvas UI works as planned (artsy meter increases with more colors, all colors work, erasing works, doge button works, clear works).
  - test multiple users sharing one whiteboard (ensure both see the same thing).
  - test multiple whiteboard support (ensure different whiteboards don't send actions to each other).
  - make sure behavior is as expected when one user draws and another user draws/erases common pixels (in equilibrium, the same image must be on both).
  - check to see that whiteboard state is saved during disconnect/reconnect.
  - test switching whiteboards (Artist dialogue should appear, filled out)
  - test saving an image
  - test posting to Facebook