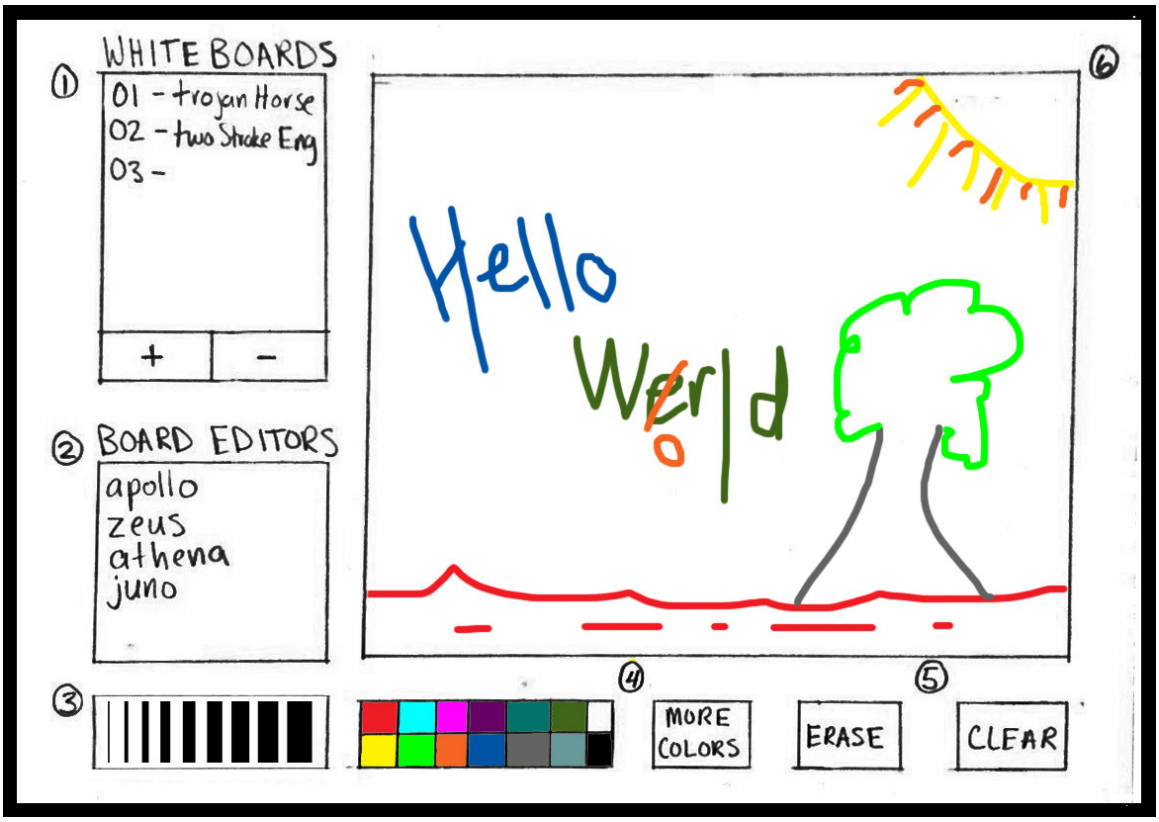
Design Milestone

Andre Aboulian, Cathleen Gendron, & Jon Beaulieu

6.005 Software Construction - Fall 2013 - Project 2: "Collaborative Whiteboard"

I. User Functionality Overview



I. Whiteboard Selector

The board selector in the left pane includes a list of all current whiteboards and appears the same for all users. Each line represents an individual board, which are numbered sequentially and named by the user. Upon clicking the "+" button, the user will be prompted to name the new board. When the board has been created on the server, it will be appended to the list for each user. Selecting a board in the list will download the board from the server, overwrite the local copy if one exists, and display the board in the canvas window.

II. Board Editors

Displays a list of users, including the viewer, who are currently modifying the selected board. This list will be updated as users enter and exit the board.

III. Thickness Selector

This tool allows the user to select a brush/eraser thickness for drawing on the whiteboard.

IV. Color Selector

The main color palette displays a grid of colors from which the user can choose to paint with. The color currently in use will be highlighted. Clicking the "more colors" button will open Swing's built-in color chooser, which will offer a larger selection of colors.

V. Erasing Tools

The erase button will allow the user to toggle between erasing and painting. "Erasing" will be defined as drawing with a white selection. Erasing will happen in the same order as drawing, so whichever request reaches the server first will erase all that has been drawn under it. Toggling back to painting will restore the user's previous color choice.

VI. Whiteboard Window

Displays the currently selected whiteboard, including all of its drawn strokes and erasures. The whiteboard be real-time interactive to allow users to collaborate simultaneously. Edits will be made in the order that modifications reach the server. In other words, a stroke logged on the server at a specific instant will be drawn over any strokes drawn before that instant.

II. Server-Client Communication

I. Protocol

I.1 Grammar

The following grammar will facilitate the text-based communication between the clients and the server. The server will send a `StoC_MSG` to the client, which will be able to send a `CtoS_MSG` back to the server.

```
StoC_MSG :== (STROKE | BRD_INFO | BRD_DEL | USER_INIT | BRD_USERS) N

CtoS_MSG :== (STROKE | SEL | BRD_REQ | BRD_DEL | BRD_ALL | USER_REQ) N

STROKE :== "stroke" S BOARD_ID S THICK S COORDS S COLOR
COORDS :== X1 S Y1 S X2 S Y2
X1, Y1, X2, Y2 :== INT
COLOR :== [0-255] S [0-255] S [0-255]
THICK :== [1-10]

SEL :== "select" S BOARD_ID

BRD_REQ :== "board_req" S NAME
BRD_ALL :== "board_all"
BRD_INFO :== "board" S BOARD_ID S NAME
BRD_DEL :== "del" S BOARD_ID
```

```
BRD_USERS :== "board_users" S BOARD_ID (S USER_NAME)+

USER_REQ :== "user_req" S USER_NAME
USER_INIT :== "you_are" S USER_NAME

NAME :== [^N]
USER_NAME :== [A-Za-z]([A-Za-z0-9]?)+
BOARD_ID :== INT

INT :== [0-9]+
N :== "\r?\n"
S :== " "
```

I.2 Usage

Adding and Removing Users When a new client attempts to...

Adding and Removing Boards TEXT

Selecting Boards TEXT

Drawing Strokes TEXT

II. Interaction

II.1 New Connections

The Whiteboard server will maintain a background thread that listens for new connections. Upon accepting a new socket, a new thread prompts the client for a username, and then instantiates a new User, which is then added to the ArrayList of all Users.

II.2 Request Streams

GUI: The GUI maintains two threads, one which listens for requests from the associated User, and another which contains a BlockingQueue containing requests to be sent to the User. Possible requests include creating a new Whiteboard, deleting a Whiteboard, selecting a different Whiteboard, or sending a Stroke.

User: Each User object holds the client-side socket. The User maintains two threads, one which listens for requests from either its MasterBoard or its GUI, and another which contains a BlockingQueue containing requests to be sent to the GUI. The request() method parses GUI requests and sends the appropriately formatted text protocol message to either the MasterBoard (when a Stroke is made) or the WhiteboardServer (for other User/Whiteboard requests).

MasterBoard: The MasterBoard maintains two threads, one which listens for requests from any of its associated Users, and another which contains a BlockingQueue containing requests to be sent to all of its associated Users. For example, the listening thread will accept a request to create a new Stroke to be added to its strokes ArrayList, and then send that updated ArrayList to all its Users.

WhiteboardServer: The WhiteboardServer is responsible for instantiating new Users with each new connection, and creating new MasterBoard objects in response to requests from Users. It maintains an InputStream to receive requests from the Users, but does not need an OutputStream.

III. Thread Safety

I. Processes

I.1 Adding New Users

I.2 Adding and Removing Boards

I.3 Drawing Strokes

I.4 Selecting Boards

II. Averted Race Conditions

II.1 New Board and New User

II.2 Concurrent Strokes

II.3 Serve-Client Races

II.4 Atomic ID Generation