Design Doc

DataType Design

- Representation of a whiteboard: Colour Bitmap
- How to erase: Draw white line instead of a colour line

Classes:

- `Canvas`: Already provided, represents a whiteboard.
    - Includes a JPanel that the user can draw on
    - Includes listeners that lets the user draw
        - Listeners will also send updates to `ClientGUI`'s queue of changes
    - Includes methods to be called by `ClientGUI` to change the properties of the "brush" that draws the lines onscreen
- `Toolbar`:
    - This is a panel on the side with the appropriate JComponents to allow the option to:
        - change colour of the brush
        - switch between paint and erase
        - change stroke size of the brush
        - and other yet to be decided functionality.
    - Has listeners that calls `ClientGUI's` methods in order to change the properties of the drawing tool
- `SessionBar`: This is a panel at the top which displays the current whiteboard and options to:
    - create a new whiteboard
    - pick a different whiteboard from the server
    - change the name of the whiteboard currently in use.
- `ClientGUI`: Each client runs this class as a Java application. Creates a new window that can connect to the `WhiteboardServer`.
    - One of `ClientGUI`'s purpose is to act as a wrapper for `Canvas`'s methods so that `Toolbar` can be independent of `Canvas`. `Toolbar`'s listeners call `ClientGUI`'s methods which in turn calls `Canvas`'s methods
    - A `ClientGUI` contains:
        - a `Canvas` object which shows the whiteboard the client is currently working on.
        - a `Toolbar` object
        - a `SessionBar` object
        - queue of changes to be sent to server every 15 millisecs
    - Whenever the whiteboard is switched, the `ClientGUI` will be given a new serialized `Image` (bitmap) by the `WhiteboardServer`, representing the current

state of the whiteboard
        - `ClientGUI` will clear the old canvas and display the `Image`
- WhiteboardServer:
    - Handles incoming connections and creates `Client` objects for them.
- WhiteboardManager:
    - Manages `Whiteboard` instances.
- Client
    - Represents a client, running on its own thread.
    - Receives and processes messages from the client and passes them off as appropriate to anything that is interested.
    - Equality on clients is transitive, reflexive, symmetric and consistent.
- Whiteboard
    - Represents a whiteboard, and contains its image data and set of active clients.
- ClientException
    - This is an exception. It can be thrown.
    - When it is thrown in a message handler, an error response is returned to the client.

Protocol

- Space separated string
- Server to client:
    - `DRAW [colour] [start1] [end1] [start2] [end2] [start3] [end3]…`
    - `JOIN [whiteboard]`
    - `HELLO [whiteboards]`
    - `WHITEBOARD [base64-encoded whiteboard bitmap] [names of people]`
    - `GOODBYE`
- Client to server:
    - `PAINT [colour] [start1] [end1] [start2] [end2] [start3] [end3]…`
    - `JOIN [whiteboard]`
    - `CREATE [name]`
    - `HELLO [username]`
    - `QUIT`

Concurrency Strategy

- Server Threading:
    - One thread per client
    - Locking should be used as appropriate.

    - LOCK ALL THE THINGS.
- Threading (client-side)
    - Event dispatch thread that updates the GUI
    - Separate thread that has a timer and sends updates to server every 15 millisecs
    - Lock on queue that contains batches of updates to send to server
- Wow. Much lock. So synchronize. Very deadlock. Too threads. Many interleaves. Wow.

Testing Strategy

- Test all public methods with minimal use of client-server / threading.
- Test GUI layout and event actions by inspection -> will document testing in appropriate classes
    - General test areas:
        - check that GUI looks right
        - try to do reasonable user things like resizing, clicking really fast
        - make sure GUI is responsive at all times (even when updates are being sent to server)
        - check that all buttons and gui elements behave as expected

Additional Features

- Change colours
- Change brush size [done]
- Fun Minigames
- Be able to draw different shapes (ellipses, rectangles, etc.) with different sizes
- Set mouse cursor to current tool state
- Add text
- Show all users connected to current whiteboard

Tasks

- gracefully handle resizing

Tasks
* warmup: be able to draw white line, toggle between white and black
* client code
    - send updates periodically
    - receive updates

* set up server
    - accept multiple clients
    - take updates
    - broadcast updates