Group Project Report

PaintKid System

MUSTAFA Batin Efe (23501154), BABETZKI Luka (23500859), [L] ESCOBAR Valeria (23501170)

1. System Criteria

Group members:

- Mustafa Batin Efe (23501154)
- Luka Babetzki (23500859)
- [L] Valeria Escobar - Martinez (2350117)

IDE used: Eclipse

Approach implemented: Basic client-server approach

2. System Walkthrough

2.1. Username and Chat:

We've integrated a username input field to allow users to personalise their interactions. When a user sends a chat message, the client encapsulates the message along with the username and sends it to the server over the established TCP connection. This not only facilitates meaningful communication but also helps identify the source of messages in the collaborative environment. If a user doesn't provide a username, the application defaults to labelling them as "Guest" in the chat.

2.2. Grid Size and Canvas:

Users can dynamically change the grid size through a dropdown menu. The selected size triggers a confirmation dialog, warning that changing the grid size will clear the canvas. Upon confirmation, the client updates its local canvas data and sends the new size to the server over the TCP connection. This ensures synchronised grid size changes across all connected clients, maintaining a consistent drawing environment.

2.3. Brush Mode and Pen Sizes:

Our application supports different painting modes, such as Brush and Pixel modes. Users can switch between these modes using toggle buttons. Additionally, there are options for varying pen sizes, providing flexibility in drawing strokes. Each mode and pen size selection triggers an update in the client's local state and is broadcasted to all connected clients for a consistent drawing experience.

2.4. Eraser Tool and Clear Button:

The application features an eraser tool that allows users to selectively remove drawn content. The eraser tool functions similarly to the brush, changing the colour of pixels to match the background. Moreover, users can clear the entire canvas with a "Clear" button, which prompts a confirmation dialog before initiating the clearing process. Both the eraser tool and clear button operations are transmitted to the server, ensuring synchronised updates for all connected clients.

2.5. TCP Connections:

We establish TCP connections using the Java **Socket** class. After obtaining the server's address and port information through a UDP broadcast message, we create new sockets with the received details. These TCP connections enable reliable, bidirectional communication between the client and the server. We use input and output streams (**DataInputStream** and **DataOutputStream**) to send and receive data. The communication includes sending chat messages, updating the canvas, and handling user actions like changing the grid size. We've employed a dedicated thread to continuously receive data from the server, ensuring that the client remains responsive to incoming messages and updates.

2.6. UDP Connections:

We use UDP for broadcasting server information to discover the server's address and port. We create UDP sockets (**DatagramSocket**) on specified ports (e.g., 12345) for sending and

receiving broadcast messages. The client broadcasts a message containing its username, and the server responds with its IP address and port number. We then create TCP sockets using the received server information to establish more stable and reliable connections for further communication. The combination of UDP for initial discovery and TCP for ongoing communication provides an effective and efficient approach for a collaborative drawing application.

3. Additional Features

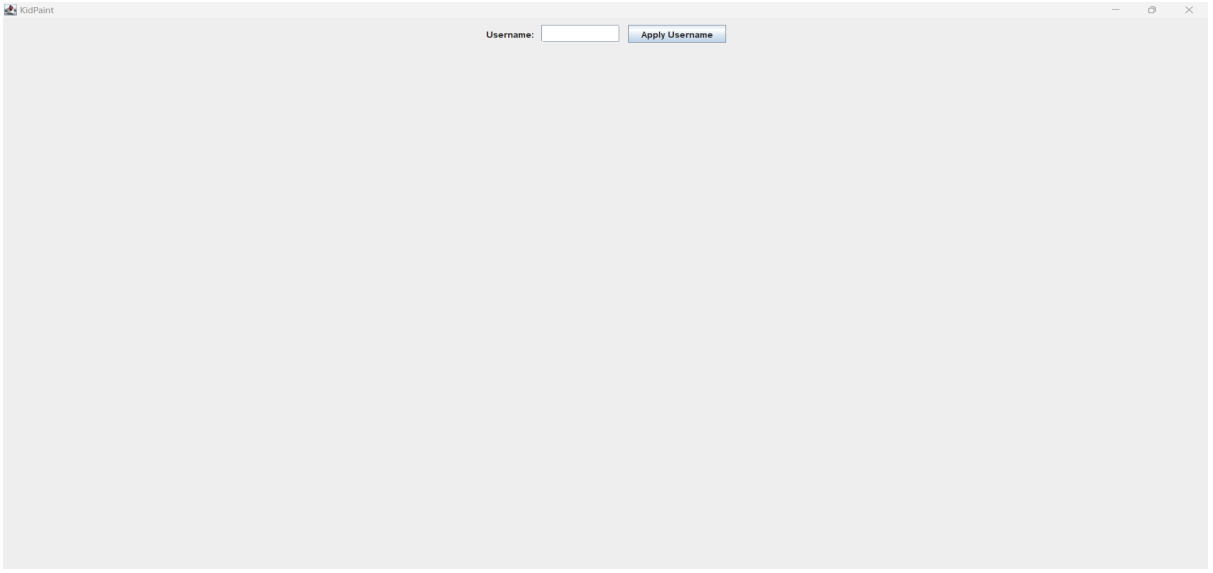
Besides developing the base system, we added some additional features to add more options and functionality to the user and thereby enhance the system as a whole.

3.1. Add Username:

Allows the user to assign themselves a username

This additional feature will ask the user to input a username, which is displayed across all clients interfaces.

Username assigning window:



We implemented the eraser tool feature as follows:

- The username is obtained from a text field (**usernameField**).
- When the user presses Enter in the text field (**msgField**), the **onTextInputted** method is invoked.
- The username and message are sent to the server using the **out** DataOutputStream.

How to use this feature:

Step 1 - Opening the Project in Eclipse: Begin by launching Eclipse and loading our "KidPaint_Client" and “Server” project. Run the “server” to establish a connection to the server, and then run the KidPaint java file as a java application by right clicking it on the left of the workspace, then Run As > Java application. If there are multiple users, they do not need to run the “server”, but instead run the “KidPaint” java file while being connected to the same subnet.

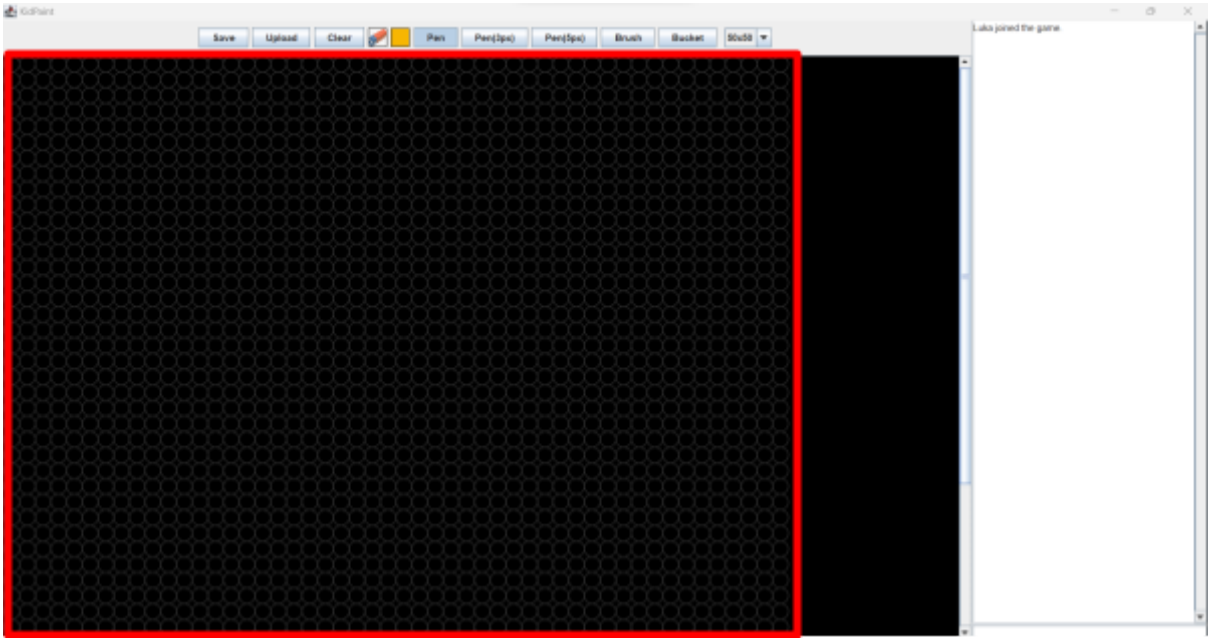
Step 2 - Enter Username: Simply enter your given username and click “Apply Username”.

3.2. Grid Size:

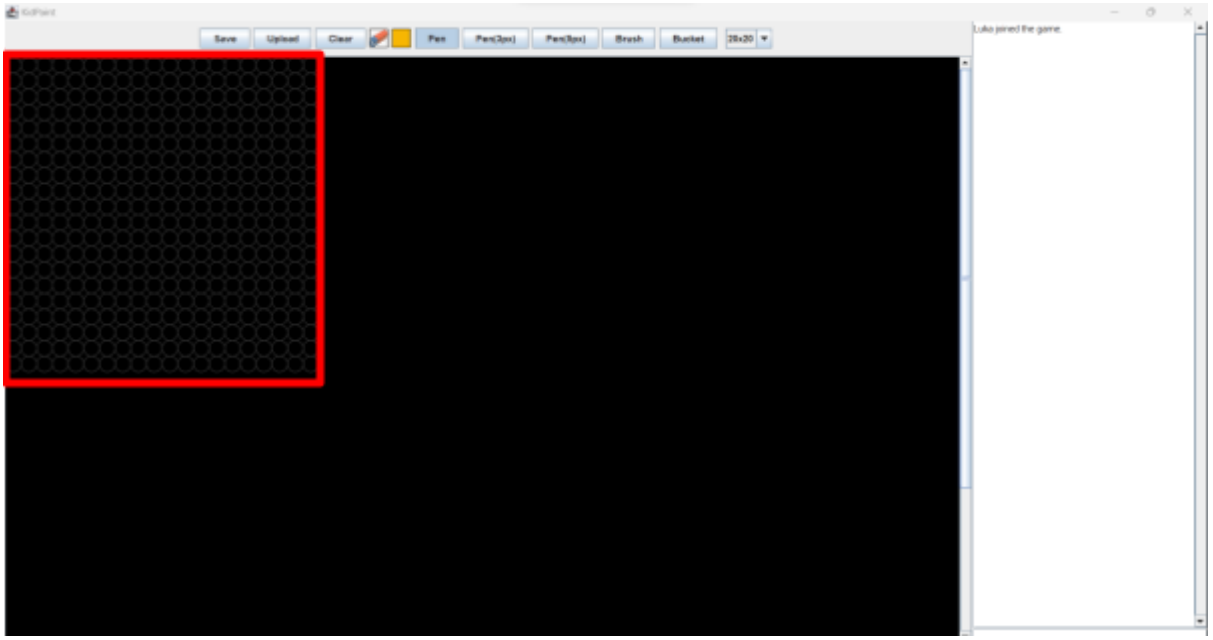
Changing the Grid Size

This feature enables users to alter the default canvas size of 50x50 to any equal x to y value sized grid. This enhancement provides flexibility for users who may prefer working on smaller, more concise canvases. Below are screenshots that demonstrate this change in window size:

50x50 grid size:



20x20 grid size:



We implemented this feature of resizing the grid canvas through the following sequence:

- The grid size is represented by a combo box (**gridSizeComboBox**) with options generated using the **generateGridSizeOptions** method.
- An **ItemListener** is attached to the combo box to detect changes in selection.
- When the user selects a new grid size, a confirmation dialog is shown, and if confirmed, the grid size is updated.
- The updated grid size information is sent to the server using the **out** DataOutputStream.

How to use this feature:

Step 1 - Opening the Project in Eclipse: Begin by launching Eclipse and loading our "KidPaint_Client" and “Server” project. Run the “server” to establish a connection to the server, and then run the KidPaint java file as a java application by right clicking it on the left of the workspace, then Run As > Java application. If there are multiple users, they do not need to run the “server”, but instead run the “KidPaint” java file while being connected to the same subnet.

Step 2 - Enter Username: Simply enter your given username and click “Apply Username”.

Step 3 - Locating the Grid Changing Menu: At the top of the window, there is a horizontal bar menu with some buttons. Locate the button with a drop down menu on the far right of the bar called “50x50” by default.

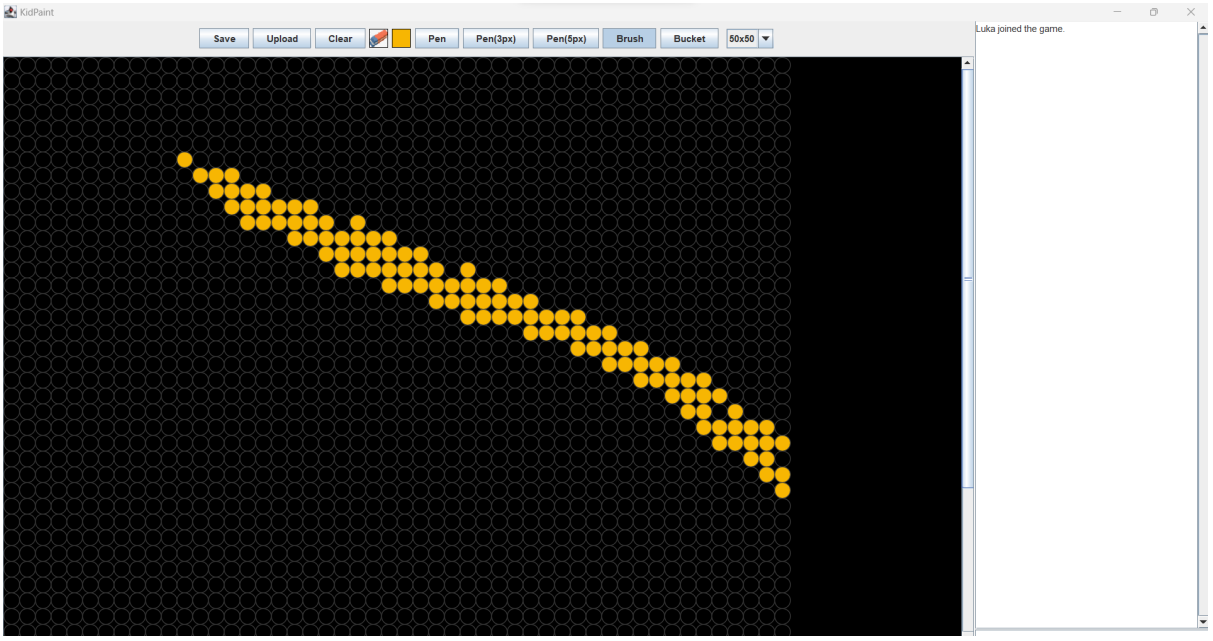
Step 4 - Changing the Size of the Grid: To change the size of the grid, simply click on the down arrow of the “50x50” button and select the size you wish to change the grid size to.

Note: The resized grid will NOT retain the existing artwork.

3.3. Brush Mode:

Changing to Brush Mode

Changing to brush mode provides users with the ability to dynamically adjust the flow of the paint brush used for drawing in the application. Below is a screenshot of this feature in action:



We implemented the brush size feature as follows:

- Brush mode is implemented with a toggle button **(tglPen2).**
- When the button is clicked, it sets the **paintMode** to **PaintMode.Brush.**
- Similar logic is implemented for other paint modes (**PaintMode.Pixel and PaintMode.Area).**

How to use this feature:

Step 1 - Opening the Project in Eclipse: Begin by launching Eclipse and loading our "KidPaint_Client" and “Server” project. Run the “server” to establish a connection to the server, and then run the KidPaint java file as a java application by right clicking it on the left of the workspace, then Run As > Java application. If there are multiple users, they do not need to run the “server”, but instead run the “KidPaint” java file while being connected to the same subnet.

Step 2 - Enter Username: Simply enter your given username and click “Apply Username”.

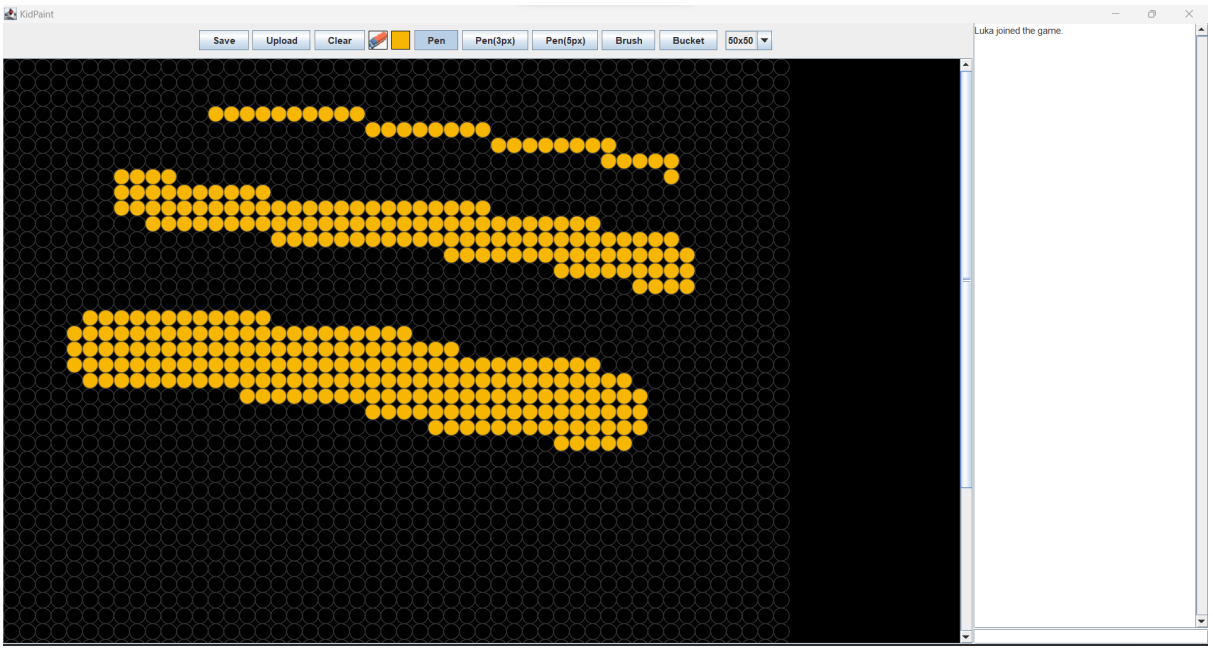
Step 3 - Locating the Brush Button: At the top of the window, there is a horizontal bar menu with some buttons. Locate the button called “Brush”.

Step 4 - Changing to Brush Mode: To change to the Brush tool by clicking the “Brush” button and draw with a new unique stroke.

3.4. Pen Size:

Changing the Pen Size between 1, 3, and 5 mode

Changing the Brush Size (1, 3, and 5 mode) provides users with the ability to dynamically adjust the width and height of the paint brush used for drawing in the application. Below is a screenshot of these features in action:



We implemented the pen size feature as follows:

- Different pen sizes are represented by toggle buttons (**tglPen**, **tglPen2**, **tglPen3**, **tglPen4**).
- Each button is associated with an **ActionListener** to handle the selection and set the **paintMode** accordingly.
- The size of the pen is likely determined by the selected button.

How to use this feature:

Step 1 - Opening the Project in Eclipse: Begin by launching Eclipse and loading our "KidPaint_Client" and “Server” project. Run the “server” to establish a connection to the server, and then run the KidPaint java file as a java application by right clicking it on the left of the workspace, then Run As > Java application. If there are multiple users, they do not need to run the “server”, but instead run the “KidPaint” java file while being connected to the same subnet.

Step 2 - Enter Username: Simply enter your given username and click “Apply Username”.

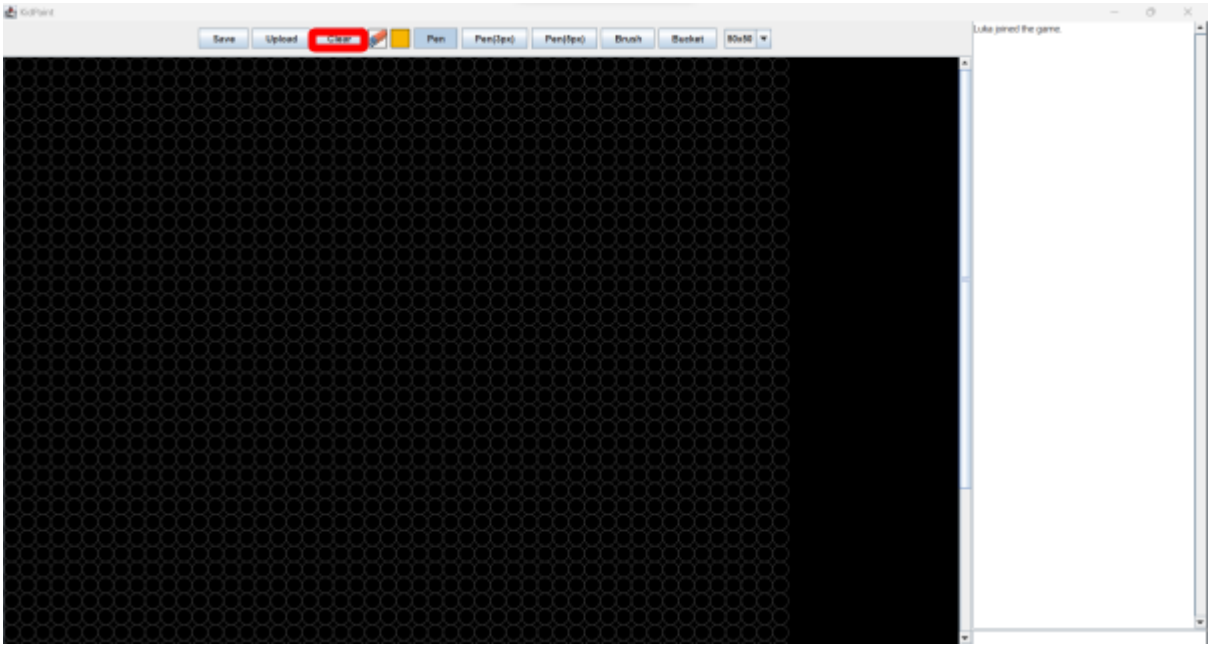
Step 3 - Locating the Pen Size Buttons: At the top of the window, there is a horizontal bar menu with some buttons. Locate the buttons called Pen, Pen(3px), and Pen(5px).

Step 4 - Adjusting Pen Size: To change the size of the pen, simply select the size you wish to draw with, click and drag the mouse and select new pen sizes when needed.

3.5. Clear Button:

Completely clearing paint from the Window

The "Clear" button allows users to reset the canvas entirely, providing a clean slate for new drawings.



We implemented the clear button feature as follows:

- The clear button triggers the **clearPaint** method.
- It iterates through the entire grid, sets each pixel to black, and sends the pixel data to the server using the **out** DataOutputStream.

How to use this feature:

Step 1 - Opening the Project in Eclipse: Begin by launching Eclipse and loading our "KidPaint_Client" and “Server” project. Run the “server” to establish a connection to the server, and then run the KidPaint java file as a java application by right clicking it on the left of the workspace, then Run As > Java application. If there are multiple users, they do not need to run the “server”, but instead run the “KidPaint” java file while being connected to the same subnet.

Step 2 - Enter Username: Simply enter your given username and click “Apply Username”.

Step 3 - Locating the Clear Button: At the top of the window, there is a horizontal bar menu with some buttons. Locate the button called “Clear”.

Step 4 - Clearing the Canvas: Within the tool panel, find the "Clear" button and click it to reset the canvas entirely.

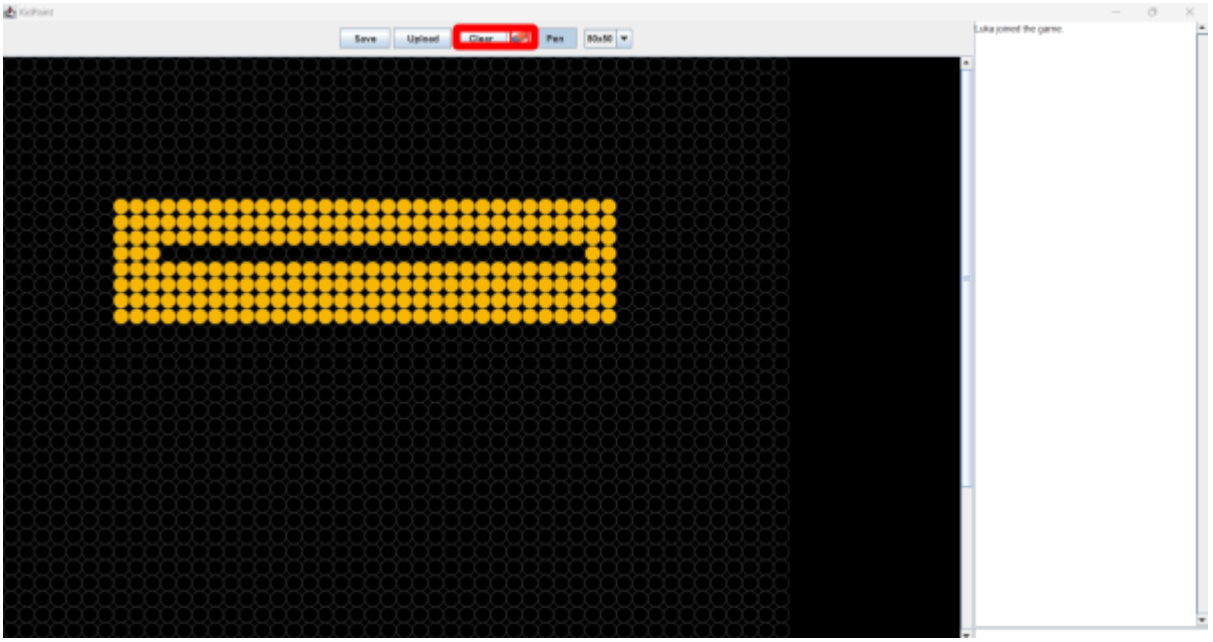
Note: Users should be careful, as this action is permanent and cannot be reversed.

3.6. Eraser Tool:

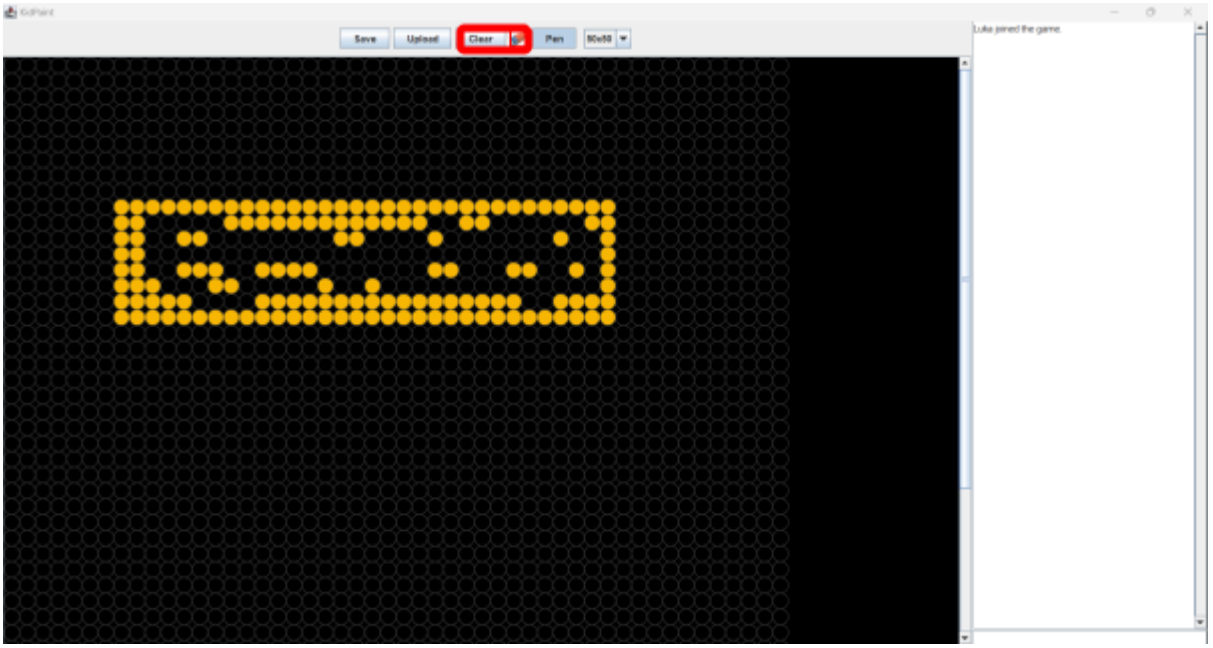
Selectively clearing paint from the Window

Similarly to the ‘Clear’ tool, the Eraser tool removes paint from the canvas, but instead acts as a brush that removes paint when the user clicks with their mouse.

Drawing before eraser:



Drawing after eraser:



We implemented the eraser tool feature as follows:

- The eraser tool might be represented by a button (**eraserButton**).
- The logic for the eraser tool is not explicitly provided, but it's likely that it sets the **paintMode** to an eraser mode and modifies pixel data accordingly.

How to use this feature:

Step 1 - Opening the Project in Eclipse: Begin by launching Eclipse and loading our "KidPaint_Client" and “Server” project. Run the “server” to establish a connection to the server, and then run the KidPaint java file as a java application by right clicking it on the left of the workspace, then Run As > Java application. If there are multiple users, they do not need to run the “server”, but instead run the “KidPaint” java file while being connected to the same subnet.

Step 2 - Enter Username: Simply enter your given username and click “Apply Username”.

Step 3 - Locating the Eraser Tool: At the top of the window, there is a horizontal bar menu with some buttons. Locate the button with the graphic of an eraser, situated to the right of the “clear” button mentioned previously.

Step 3 - Using the Eraser Tool: Once selected, the user may now draw on the canvas to erase any paint painted onto the grid canvas.

4. Contribution Table

	Mustafa Batin Efe (23501154)	**Luka Babetzki (23500859)**	**[L] Valeria Escobar (23501170)**
Contribution (content)	**Additional features implementation** - Adjustable window size - Clear option **Basic Approach Server**	**Additional features implementation:** - Erasing option of 3 pen sizes **Group Project Report**	**Additional features implementation:** - Drawing option of 3 pen sizes **Group Video**
Contribution (%)	50%	25%	25%
Signature: