Project Log for Palette App

CM1010 Introduction to Programming ll Final Project

Prepared by ***Roya Nikseresht***

MY GOALS, PROGRESS, CHALLENGES, AND REFLECTIONS.

Topic:	ScissorsTool Development
Date:	January 13, 2025 to January 19, 2025

Progress Made:
- **ScissorsTool Created:**   - **Selection:** I managed to get the selection working. Users can click and drag to define a rectangular area on the canvas, and as they drag, the selection area updates.   - **Cutting:** Once the area is selected, users can cut it, which saves the selected region and clears it from the canvas (by setting the selected area to white).   - **Pasting:** After cutting, users can paste the selected image back onto the canvas at the current mouse position.   - **UI:** The tool now has working buttons for **Cut**, **Paste**, and **Reset Selection**. The Paste button only works if an image is selected, and Reset clears the selection. - **UI and Interaction:**   - The buttons for Cut, Paste, and Reset are all implemented. The interface changes depending on whether there's an image to paste, making it more intuitive for the user. - **Mouse Events:**   - I've set up mouse event handlers to control when and how the user interacts with the canvas. The selection works when the mouse is dragged, and pasting works by clicking with the mouse in the right spot.

Challenges and Solutions:

- **Canvas State Management:**
  Challenge: Managing the canvas state when cutting and pasting was tricky. I needed to ensure that only the selected area was erased and not the whole canvas.
  Solution: I used `loadPixels()` and `updatePixels()` to save and restore the canvas state. This allowed me to make pixel changes (like cutting the selection) while keeping everything else intact.

- **UI Feedback for Paste Button:**
  Challenge: The Paste button needed to be disabled when no image was selected, but making sure this worked smoothly was a bit tricky.
  Solution: I added a condition to check whether there was an image to paste before enabling the button. If there's no selected image, the Paste button remains disabled to prevent errors.

Next Steps/Goals for Next Week:

- Improve the UI by giving better feedback, such as changing the button color when it's disabled.

- Implement the **ImageTool** to allow users to upload and manipulate images on the canvas.

- Add functionality for resizing images using sliders.

- Optimize rendering for smoother resizing and movement.

Reflection:

What went well?

- The core features are working! Cutting, pasting, and resetting are all functional, and the canvas updates as expected.
- The UI components were easy to integrate and react well to the tool's state.

What could have been improved?

- The reset function works, but I think it could be made more efficient, especially when restoring canvas states or clearing selections.
- The mouse event handling could be more refined to make interactions smoother.

What I learned:

- Managing pixel-level changes on a canvas is all about attention to detail. A small mistake can mess things up, so it’s important to be mindful of the canvas state at all times.
- UI feedback really matters when working with interactive tools. Making sure users know what’s possible (or not possible) at each step goes a long way in making the tool more user-friendly.

Topic:	ImageTool Development
Date:	January 20, 2025 to January 26, 2025

Progress Made:

- **ImageTool Implemented:**
  - Users can now upload images via a file input button. Once an image is selected, it's displayed on the canvas.
  - Added sliders for resizing the image in real-time. These sliders adjust the image width and height dynamically as the user changes them.
  - The tool interacts smoothly with other elements on the canvas, without interfering with the existing tools.
- **UI Enhancements:**
  - Real-time feedback is provided on the sliders, so users see the image size change immediately as they adjust the sliders.
  - The tool is well-integrated into the project, and everything works without disrupting other functions.

Challenges and Solutions:

- **Image Loading Delay:**
  Challenge: When an image was uploaded, it wasn't showing up right away.
  Solution: I used `loadImage(file.data, callback)` to make sure the image is fully loaded before displaying it on the canvas.

- **Sliders Not Updating Dynamically:**
  Challenge: The sliders didn't update properly when a new image was uploaded.
  Solution: I made sure that the sliders get updated based on the image dimensions when it's loaded and used `redraw()` to refresh the canvas.

Next Steps/Goals for Next Week:

- Optimize the tool to make resizing and moving images even smoother.

- Start working on the **HeartTool and TextTool**, allowing users to draw hearts and sparkles and texts on the canvas.

- Add a size slider for them so users can adjust their size dynamically.
- Include a shape selector that lets users switch between hearts and sparkles.
- Ensure smooth rendering when multiple shapes are drawn on the canvas.

Reflection:

Everything is moving along as planned. The **ImageTool** is functional, and the next steps will focus on refining and adding new features.

Topic:	Heart and Sparkle and Text Tool Development
Date:	January 27, 2025 to February 2, 2025

Progress Made:

- **HeartTool and TextTool Enhanced:**
  - The tool now allows users to draw both hearts and sparkles on the canvas with adjustable sizes.
  - A size slider for adjusting the heart and sparkle sizes, along with a shape selector, makes the tool much more versatile.
- **TextTool Integration:**
  - Implemented the **Text Tool**, allowing users to **add custom text elements** to the canvas.
  - Added a **size slider** to let users **adjust text size** easily.
    Users can type text into an input box, and clicking on the canvas **places the text** at the cursor's position.
  - Improved **UI for text entry**, making it more **user-friendly** and **responsive**.
- **UI Improvements:**
  - The HeartTool's UI is now more user-friendly, with live updates to the size and shape of the drawn elements based on slider adjustments.
  - Cleaned up the tool's UI to be more intuitive and interactive.

Challenges and Solutions:

- **Shape Switching Between Heart and Sparkle:**
  Challenge: Initially, it was hard to switch between drawing a heart and a sparkle smoothly.
  Solution: Added a shape selector to the UI, making it easy for users to pick between hearts and sparkles without needing to switch tools.

- **Smooth Rendering of Multiple Shapes:**
  Challenge: Drawing many hearts or sparkles at once sometimes caused performance hiccups.
  Solution: Optimized the rendering process by limiting the number of objects drawn at once and reducing unnecessary updates to the canvas.

Next Steps/Goals for Next Week:

- Implement the **StarBrushTool** to allow users to draw colorful stars on the canvas with random sizes, colors, and opacities.
- Develop the **DottedBrushTool** to let users draw colorful, jittery dots for a more dynamic and playful brush effect.
- Both tools will include various randomization features for a more organic drawing experience.
- Optimize rendering for both tools to ensure smooth performance, even when drawing large numbers of stars or dots.
- Improve UI by adding an icon for each new tool, making it easy for users to switch between tools.

Reflection:

The **HeartTool and Sparkle Tool** are now **fully functional and optimized**, and the **Text Tool addition** brings a **new level of customization** for users. These features make the drawing experience **more dynamic, interactive, and user-friendly**. The upcoming **StarBrushTool and DottedBrushTool** will **further enhance creativity**, adding **more fun and versatility** to the app.

Topic:	StartBrushTool, DottedBrushTool, ColourPalette and EraserTool Development
Date:	February 3, 2025 to February 9, 2025

Progress Made:

- **StarBrushTool:**
  - Successfully implemented a **StarBrushTool** that allows users to draw stars with random sizes, colors, and opacities.
  - Each time the mouse is pressed, a star is drawn with a unique size and color selected from a predefined palette.
- **DottedBrushTool:**
  - Developed the **DottedBrushTool**, enabling users to draw colorful, jittery dots with random sizes, colors, and opacities.
  - The jitter effect adds a dynamic quality to the dots, making the drawing experience more playful.
- **ColourPalette Tool:**
  - Implemented a **ColourPalette** for selecting drawing colors and adjusting pen sizes.
  - The tool includes a color picker and a slider for pen size, allowing users to easily change their drawing settings.
- **EraserTool:**
  - Created the **EraserTool** that lets users erase parts of the canvas with adjustable eraser size.
  - The tool is responsive and smooth, allowing precise erasing with a customizable stroke size.
- **UI and Interaction:**
  - All tools (StarBrushTool, DottedBrushTool, ColourPalette, EraserTool) are integrated into the UI, with clear feedback for each action.
  - The user interface is now more dynamic and responsive, making the tools easily accessible and customizable.

Challenges and Solutions:

- **Performance with Multiple Stars/Dots:**
  Challenge: Drawing large numbers of stars or dots led to minor lag.
  Solution: Optimized the drawing process by reducing redundant operations and ensuring smoother rendering.

- **ColourPalette Tool Interaction:**
  Challenge: Initially, the color and size changes from the **ColourPalette** weren't updating instantly during drawing.
  Solution: Added event listeners to ensure immediate updates when the user changes color or pen size, providing real-time feedback.

- **EraserTool Responsiveness:**
  Challenge: The **EraserTool** was a bit slow when used with larger sizes.
  Solution: Optimized the eraser drawing process by reducing unnecessary redraws and improving the stroke rendering.

Next Steps/Goals for Next Week:

- Work on the current tools to refine their functionality.

- Debug any issues and improve performance for smoother drawing.

- Test all tools to ensure they are interacting correctly and fixing any bugs that arise.

Reflection:

The project is progressing well and is on track. The new tools are functioning as intended, and I'm on track to focus on debugging and refining them in the coming week and probably submit my project.

Topic:	**Final Project Log**
Date:	Ending: February 12, 2025

Project Overview:

This project is a feature-rich **drawing application** that merges creativity with interactive game elements to enhance user engagement. It includes a comprehensive suite of **drawing tools**, such as traditional tools (**Freehand, Line, and Shape tools**) and unique features like the **Spray Can and Editable Shapes and Mirror Tool and Art Generator Tool and Random Art Generator Tool for automated creativity.**

Additionally, game-like challenges such as the **Shape Memory Game and Circle Draw Game** add an element of interactive play, making the application both fun and educational.

The latest development phase introduced new tools, including the **Scissors Tool, StarBrushTool, DottedBrushTool, Sparkle and Heart Effects, Text Tool, Image Tool ColourPalette, and EraserTool**, providing more customization and flexibility in the drawing experience. The app is designed to be **user-friendly, visually appealing, and accessible across devices**, supporting both **light and dark modes** for an enhanced user experience.

The app is designed to be **user-friendly, visually appealing, and accessible across devices**, supporting both **light and dark modes**.

Final Features:

Traditional Drawing Tools

✔ **Freehand Tool** – Draw freely with smooth strokes.

✔ **Line Tool** – Create straight lines with precision.

✔ **Shape Tools** – Draw various shapes like rectangles, circles, and polygons.

Interactive & Special Effect Tools

✔ **Spray Can Tool** – A spray effect for dynamic, scattered strokes.

✔ **Sparkle and Heart Effects** – Adds glowing sparkle elements to drawings.

✔ **Editable Shapes Tool** – Allows resizing and transforming drawn shapes.

✔ **Art Generator Tool** – Uses **procedural generation** to create unique artwork automatically.

✔ **Random Art Generator Tool** – Generates randomized abstract art based on **user-selected themes**.

Newly Added Tools

✔ **StarBrushTool** – A brush that creates star-shaped strokes with adjustable size and color.
✔ **DottedBrushTool** – A brush that generates evenly spaced dotted patterns.
✔ **ColourPalette** – Allows users to customize **color selection**, including **gradient options**, and adjust **brush/eraser size** dynamically.
✔ **EraserTool** – An eraser with **adjustable size**, allowing users to erase sections of their drawing efficiently.
✔ **Text Tool** – Allows users to **add, edit, resize, and move text** on the canvas. Users can adjust the text size with a slider and customize the text content.

Game-Like Features

✔ **Shape Memory Game** – Challenges users to remember and redraw displayed shapes.
✔ **Circle Draw Game** – A fun game where users try to draw a perfect circle.

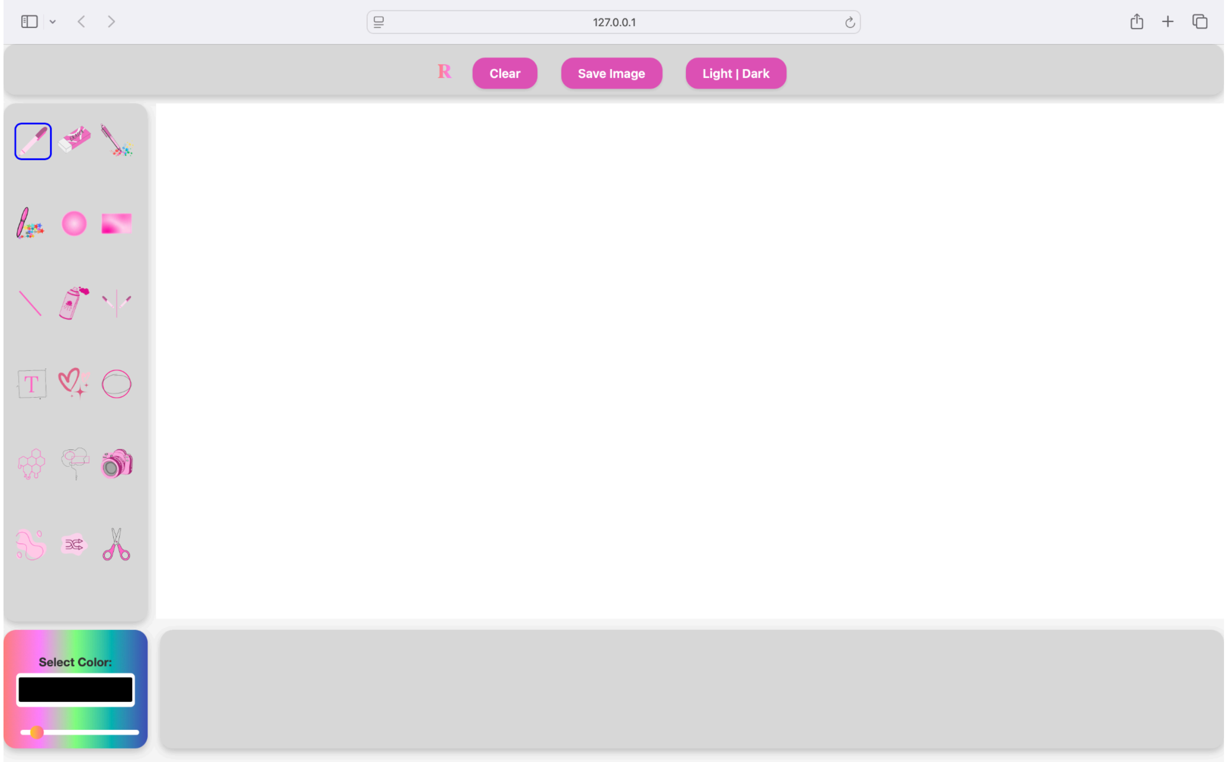
Testing and Debugging:

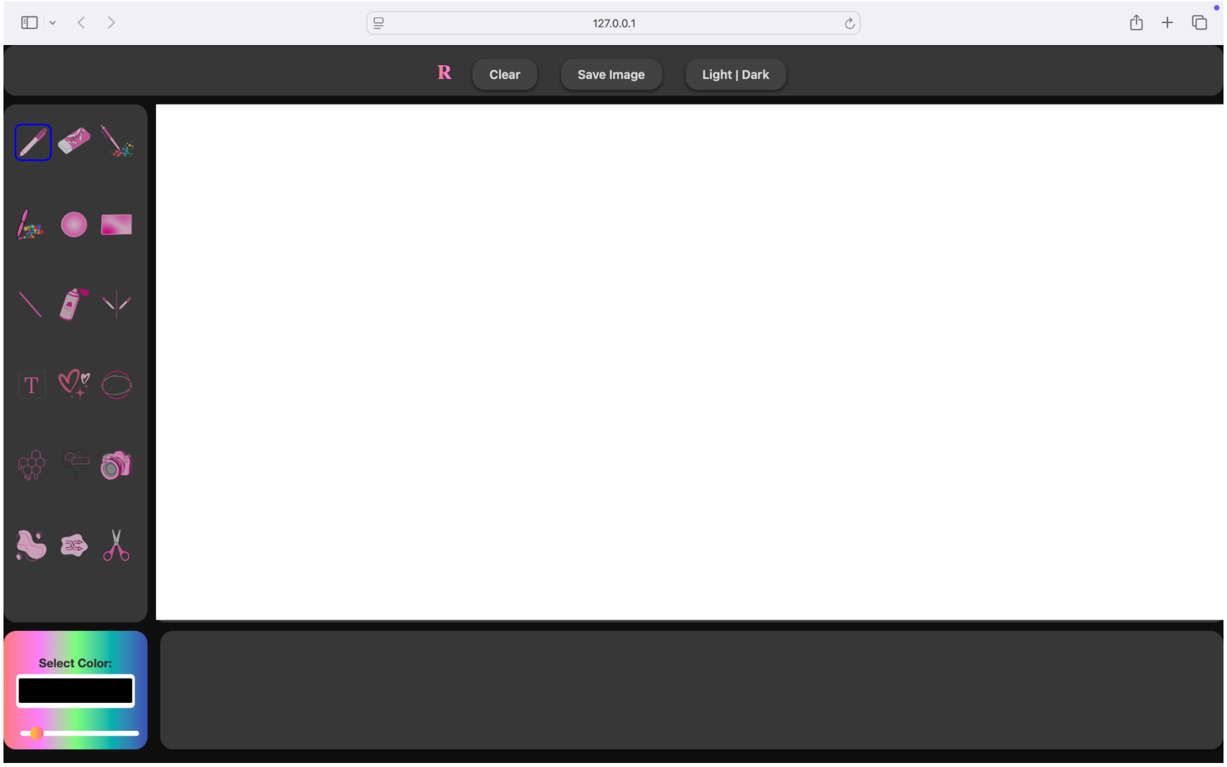
- **Testing:**
  - Conducted **extensive tests** on each tool, ensuring they function correctly in both light and dark modes.
  - Tested **canvas interactions**, making sure tools like the **Eraser, Text Tool, and Art Generators** worked properly.

- **Debugging:**
    - Fixed **tool inconsistencies** across themes.
    - Solved **EraserTool sizing issues**.
    - Improved **text positioning and resizing mechanics**.

- **Performance Optimization:**
    - Optimized **canvas rendering** to handle **procedural art generation smoothly**.

Final Look - Light & Dark Mode





Reflections

This project was an **exciting blend of UI design, tool development, and interactive features**. The addition of the **Art Generator Tools** and **Text Tool** added a new dimension to user creativity, making the application **more versatile**.

Key Takeaways

✔ Improved skills in **CSS variables & JavaScript event handling** for theme switching.

✔ Learned **procedural art generation techniques** for creative drawing tools.

✔ Gained experience in **UI/UX improvements**, particularly in **tool responsiveness and interaction**.