

PIXEL ART EDITOR

A SIMPLE AND INTERACTIVE TOOL WRITTEN IN C THAT ALLOWS USERS TO DRAW AND EDIT IMAGES PIXEL BY PIXEL, WITH TOOLS FOR DRAWING, ERASING, AND FILLING COLORS.

CSE115.02 Project Update

Group-06

Team Members:

Jamil Kabir Hridoy (1931032642)

Nafis UI Islam Nafis (2513053642)

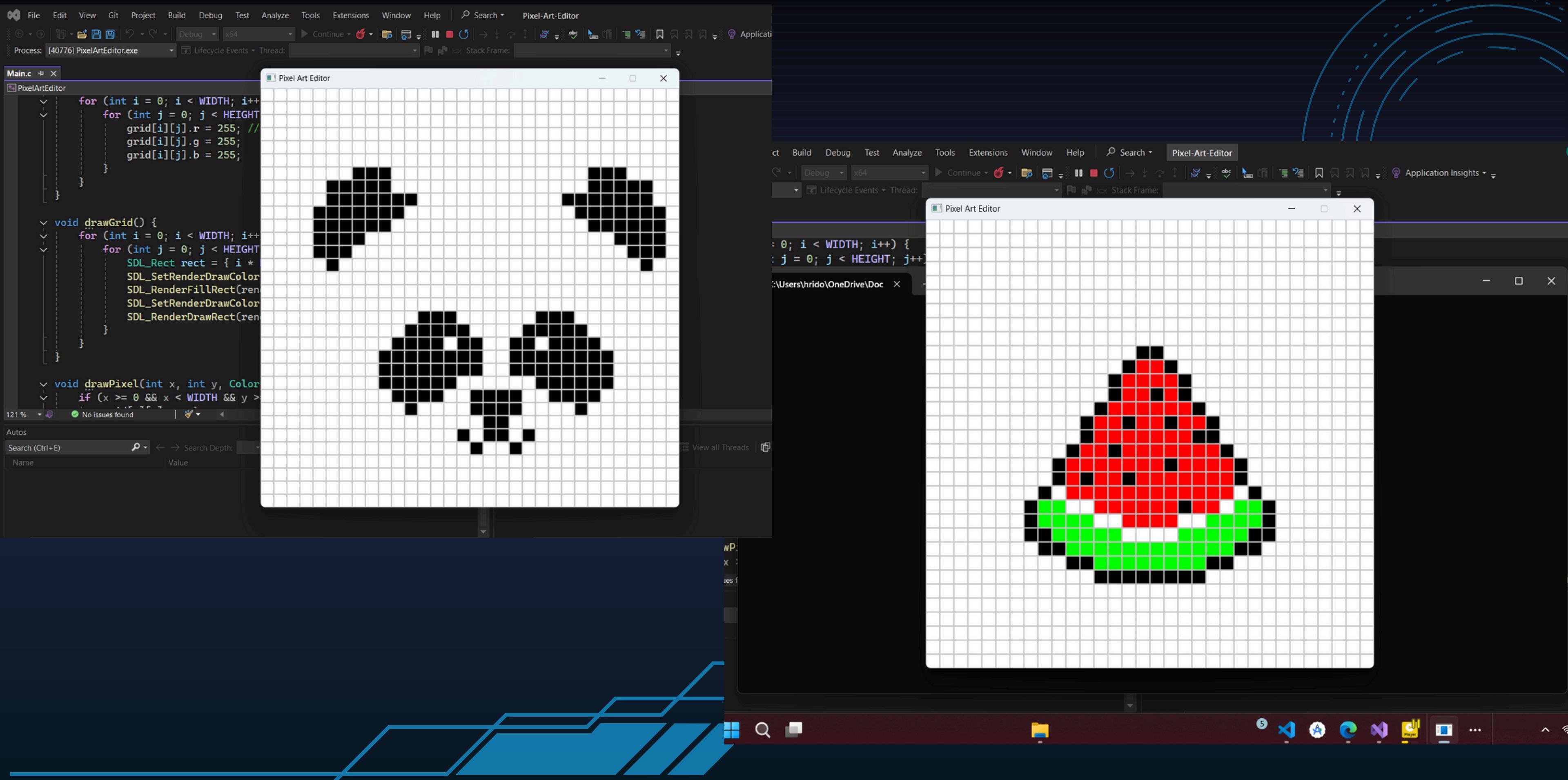
Md. Shaiful Islam (2511965642)

Salman Farsy (2513446642)

SIGNIFICANCE

- ✓ This project demonstrates core graphics programming concepts in C while providing a functional tool for artists and game developers. It's lightweight, easy to use, and a great starting point for anyone interested in building their own digital art tools.
 - ✓ This pixel art editor lies in its educational, practical, and creative value. It provides hands-on experience with graphics programming in C using SDL2, making it a great learning tool for developers.
 - ✓ Its efficiency and simplicity make it accessible to a wide range of users, running smoothly even on low-end systems.
 - ✓ By embracing pixel art's nostalgic charm, this editor keeps the art form alive while offering a user-friendly tool for digital creativity.
- 

SAMPLE ARTS



Features Overview



**Keyboard
shortcuts:**

- R → Red
- G → Green
- B → Blue
- W → White
- K → Black



**32x32 pixel grid with
customizable
colors.**



**Mouse controls:
Left-click to draw,
right-click to erase.**

Real-time rendering with SDL2



TOOLS & METHOD

1 Setup & Configuration – Installed **Visual Studio 2022**, configure **SDL2**, and link necessary libraries.

2 Application Design – Created a grid-based canvas, defined modules for input handling, rendering, and file management. The **SDL2** loop continuously listens for user inputs and updates the grid accordingly. This ensures real-time feedback for a seamless drawing experience.

3 Core Implementation:

- Graphics & UI – Initialize **SDL2**, render grid, added a color palette.
- Drawing Tools – Implement pencil, eraser, and fill bucket.
- User Input – Handle mouse clicks, movements, and keyboard shortcuts.
- File Handling (Future) – Support saving/loading images.
-

4 Testing & Debugging – Run tests, debug with Visual Studio tools, and optimized performance.

5 Finalization & Enhancements – Refining UI, add layers, animations, and export options for future upgrades. 

FUTURE WORK

Possible Enhancements

- Undo/Redo: Track pixel history using a stack.
- Save/Load: Export pixel art to a bitmap (.bmp) file.
- Layers & Frames: Support animation-like effects.
- More Colors & UI Elements: Use buttons for better interaction.
- Brush Size Options for different stroke thicknesses.



CONCLUSION

The pixel art editor is a simple yet effective tool for creating and editing pixel-based graphics. Built with C, it demonstrates essential graphics programming while serving as a practical tool for artists and game developers.

With features like grid-based drawing and color selection, it provides a solid foundation for future enhancements like layers and exporting. This project makes pixel art creation accessible, efficient, and enjoyable.



A pixelated landscape featuring a central white rectangular area containing the words "Thank You". The background is a blue sky with various colorful, abstract shapes at the top and bottom, resembling clouds or foliage. The terrain below is composed of green, blue, and purple pixels.

Thank
You