**Pixel Art Editor**

Course Code & Section: CSE 115.2

Project Group 6

Author Information

Nafis Ul Islam Nafis, Hridoy Kabir, Salman Farsy, Md. Shaiful Islam Palash

**Abstract**

This paper presents a simple pixel art editor implemented in C using the SDL2 library. The editor allows users to draw and edit images at a pixel level within a 32x32 grid. Basic functionalities include drawing, erasing, and selecting colors via keyboard shortcuts. The project serves as a foundation for further improvements, such as undo/redo capabilities, saving/loading functionality, and a more user-friendly graphical interface.

**Keywords**- Pixel Art Editor, SDL2, Graphics Programming, C Language, Input Handling, Rendering, Grid System.

**I. Introduction**

Pixel art is a form of digital art where images are edited at the pixel level, widely used in retro gaming and low-resolution design. This project implements a simple pixel art editor using SDL2, a powerful multimedia library that facilitates graphics rendering and input handling in C.

The objective of this project is to provide a lightweight and efficient tool for creating pixel-based drawings, offering essential features while maintaining an intuitive interface.

**II. System Design and Implementation**

**2.1 Graphics and Rendering**

The application utilizes SDL2 to render a 32x32 grid of pixels, each represented by a 20x20 square. The rendering loop continuously updates the screen to reflect user interactions.

**2.2 Input Handling**

User interaction is facilitated through both mouse and keyboard inputs:

Mouse Input:

Left-click: Draw using the selected color.

Right-click: Erase (turn pixel white).

Keyboard Input for Color Selection:

R → Red

G → Green

B → Blue

W → White

K → Black

**2.3 Grid System**

Each pixel in the canvas is mapped to a 20x20 square, allowing precise pixel-based drawing and erasing. The program efficiently manages grid rendering and updates to ensure smooth operation.

**2.4 Rendering Loop**

The SDL2 loop continuously listens for user inputs and updates the grid accordingly. This ensures real-time feedback for a seamless drawing experience.

**2.5 Cleanup and Resource Management**

To prevent memory leaks, SDL2 resources are properly released upon program termination.

**III. Results and Discussion**

The pixel art editor successfully allows users to create simple pixel-based drawings with minimal latency. The implementation of a keyboard-driven color selection system enhances usability. However, certain limitations exist, such as the lack of undo/redo functionality and an advanced color selection palette.

**IV. Future Enhancements**

Several improvements can be incorporated to enhance the usability and functionality of the editor:

Undo/Redo functionality: To allow users to revert mistakes and improve editing efficiency.

Saving and Loading Drawings: Enabling users to store and retrieve pixel art creations.

Expanded Color Palette: Providing more color options for creative flexibility.

Graphical User Interface (GUI) Enhancements: Improving the visual appeal and accessibility of the editor.

**V. Conclusion**

This project provides a functional base for a pixel art editor using SDL2. By implementing core features such as drawing, erasing, and keyboard-based color selection, it offers a solid starting point for future expansions. With further improvements, this editor can become a more robust and versatile tool for digital artists and game developers.

**VI. References**

[1] SDL2 Documentation. Available: https://wiki.libsdl.org/

[2] Stroustrup, B., "The C++ Programming Language," 4th Edition, Addison-Wesley, 2013.

[3] Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F., "Computer Graphics: Principles and Practice," 3rd Edition, Addison-Wesley, 2013.

[4] C Programming Language. Brian W. Kernighan and Dennis M. Ritchie.

[5] Pixel Art Tutorials. Available: https://www.pixelart.com/

**VII. Acknowledgment**

This project was developed as a learning exercise in C and SDL2 graphics programming. Special thanks to the open-source community for providing extensive documentation and resources on SDL2 development.