Thiago Ciriaco

Professor Mazidi

2340.0W1 Computer Architecture

25 April 2021

Bitmap Project Report

My Program: **MIPS Drawing Studio**

Instructions to Run:

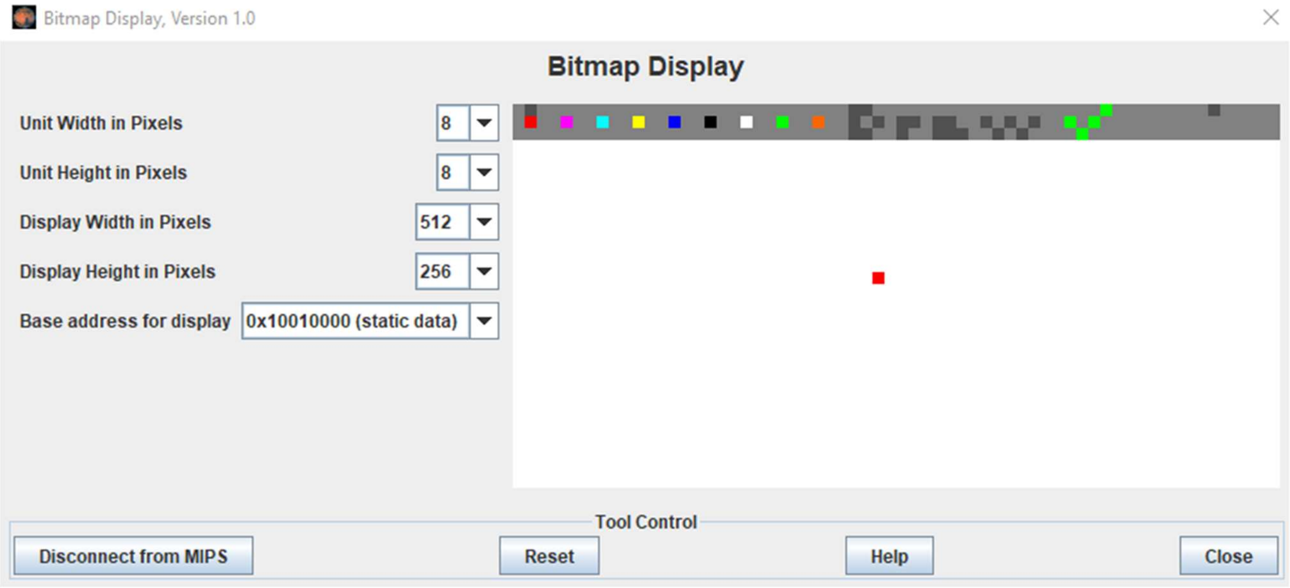
- Do **NOT** use the ordinary MARS_4_5.jar file. It is bugged.
- "Extract all" from the zip folder and open the **MARS_4_5_FIXED.jar** file
- File -> Open -> tac200002-0w1-BitmapProject.asm
- File -> Open -> bitmapMacros.asm
- Tools -> Keyboard and Display MMIO Simulator -> Connect to MIPS
- Tools -> Bitmap Display -> Connect to MIPS
    - Set "Unit Width in Pixels" to 8
    - Set "Unit Height in Pixels" to 8
    - Leave display size alone
- Run -> Assemble; Run -> Go
- The following drawing controls are also listed in program output. Type in the keyboard simulator to interact.

Controls

- **WASD** – Move Cursor (up, left, down, right respectively)
- **1-9** – Select color from the top menu based on its position
- **Q** – Decrease brush size (down to 1x1) (must be in draw mode)

- **E** – Increase brush size (up to 3x3) (must be in draw mode)
- **C** – Clear screen and fill with selected color
- **R** – Toggle between Draw Mode and Cursor Mode (must use brush size 1x1)
  - **Draw Mode**: moving will leave behind a trail based on brush size and selected color. Indicated by a blinking cursor.
  - **Cursor Mode**: moving will not overwrite any drawn pixels and brush size/color cannot be changed. Indicated by a solid cursor.
- **Space** – Completes the drawing. Then follow the prompts in MARS output (I/O) to apply a filter to your drawing, if desired

MIPS Drawing Studio initial startup screen:



Unit Width in Pixels: 8

Unit Height in Pixels: 8

Display Width in Pixels: 512

Display Height in Pixels: 256

Base address for display: 0x10010000 (static data)

Program Functionality Overview:

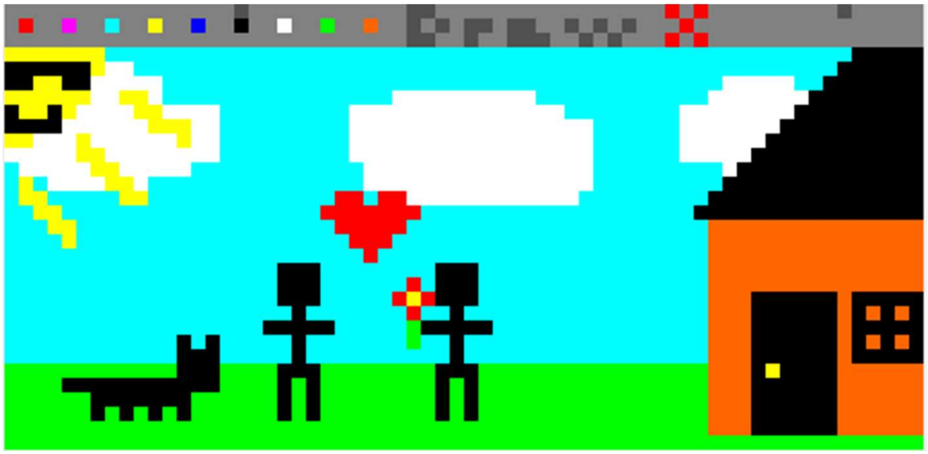
MIPS Drawing Studio is a fun program that allows users to blossom with their own creativity by drawing pixel-art in the bitmap display. There are a handful of implemented tools and features that make this significantly easier for the user. For one, the program provides a **blinking cursor** feature when the user is drawing. This makes it much easier to see where the cursor is, all while maintaining visibility of the pixel below the cursor. The program also has a **cursor mode** vs **draw mode** feature. The former allows users to relocate their pencil without drawing at the same time, which makes it more convenient to draw multiple objects. The latter is accompanied by a **brush size** feature, which makes drawing less tedious by making it easy to fill in large spaces. Of course, the user can also choose between **9 colors when drawing**, and they can clearly see which color they have selected in the **helpful UI** on the top of the screen.

Once the user if satisfied with their drawing, they are also given the choice of adding 1 of **4 drawing filters**. The filters can grayscale, invert, cool down, or warm up the finished drawing. This adds another fun, unique aspect to the experience. Once all is done, the program gives users **instructions on how to screenshot** their work on Windows or Mac. The user can then **begin a new drawing** with a clean slate or exit the program.

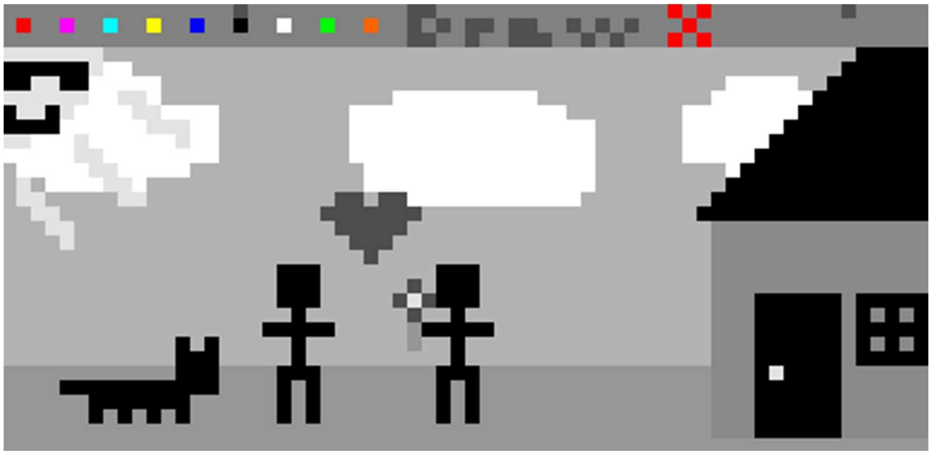
Sample Runs:

- These sample runs show off the before and after of the 4 different filters on some completed drawings by Thiago Ciriaco.

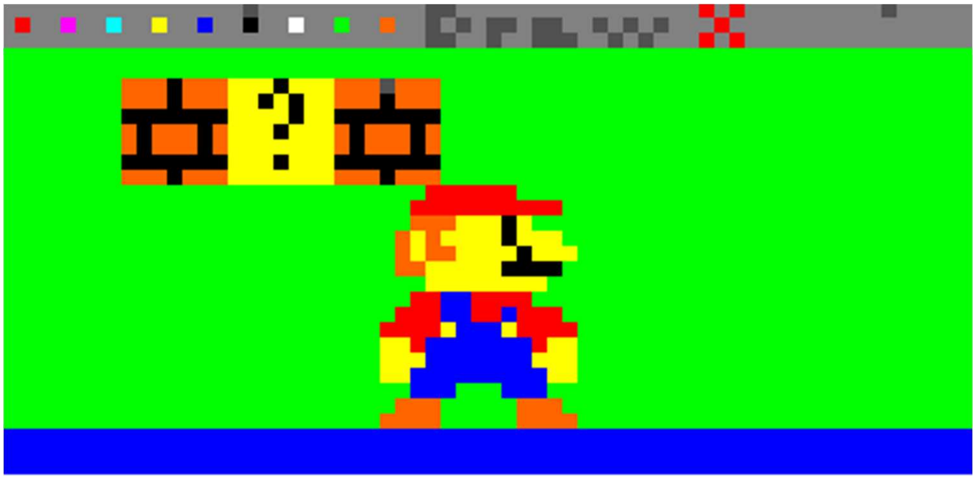
Original



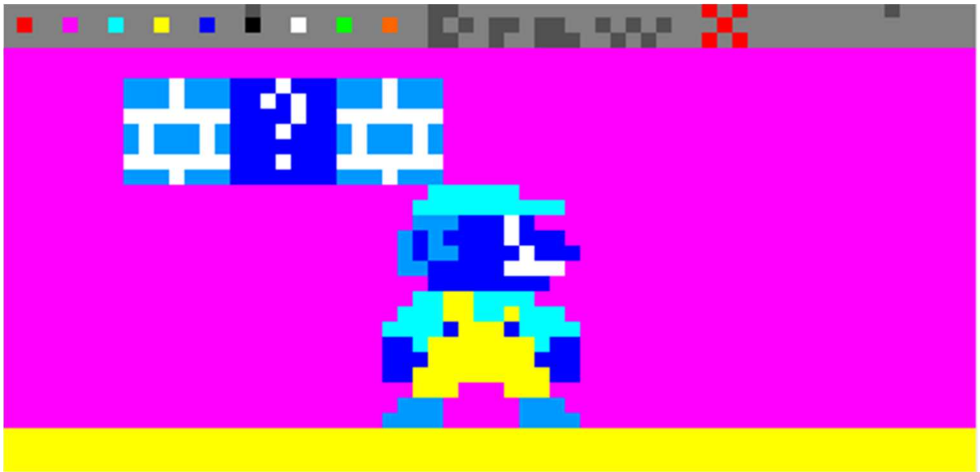
Grayscale Filter



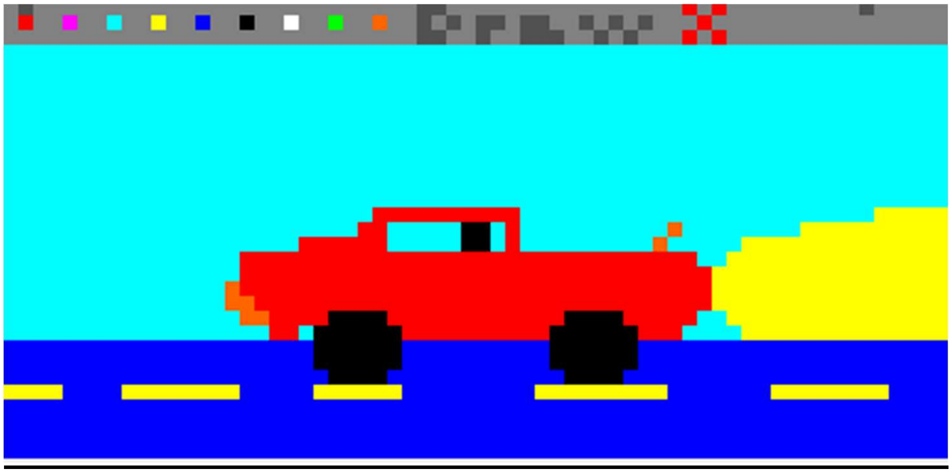
Original



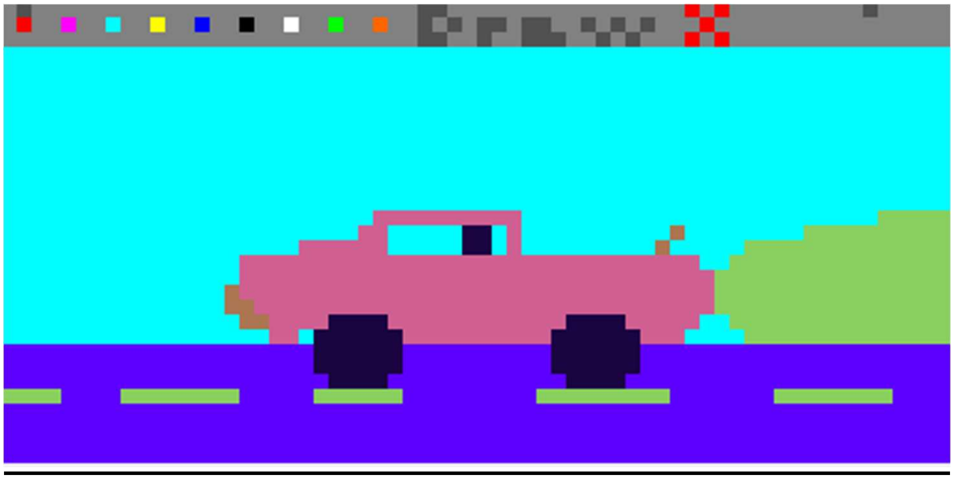
Inverted Filter



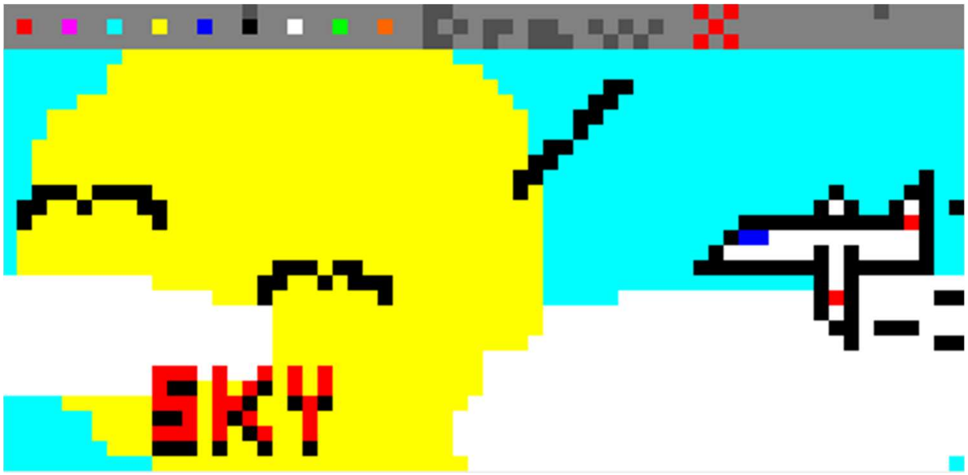
Original



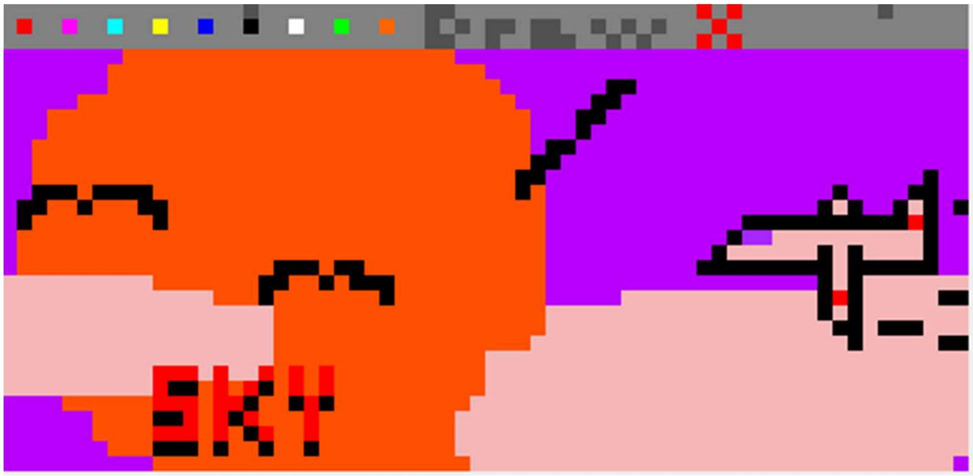
Cooler Colors Filter



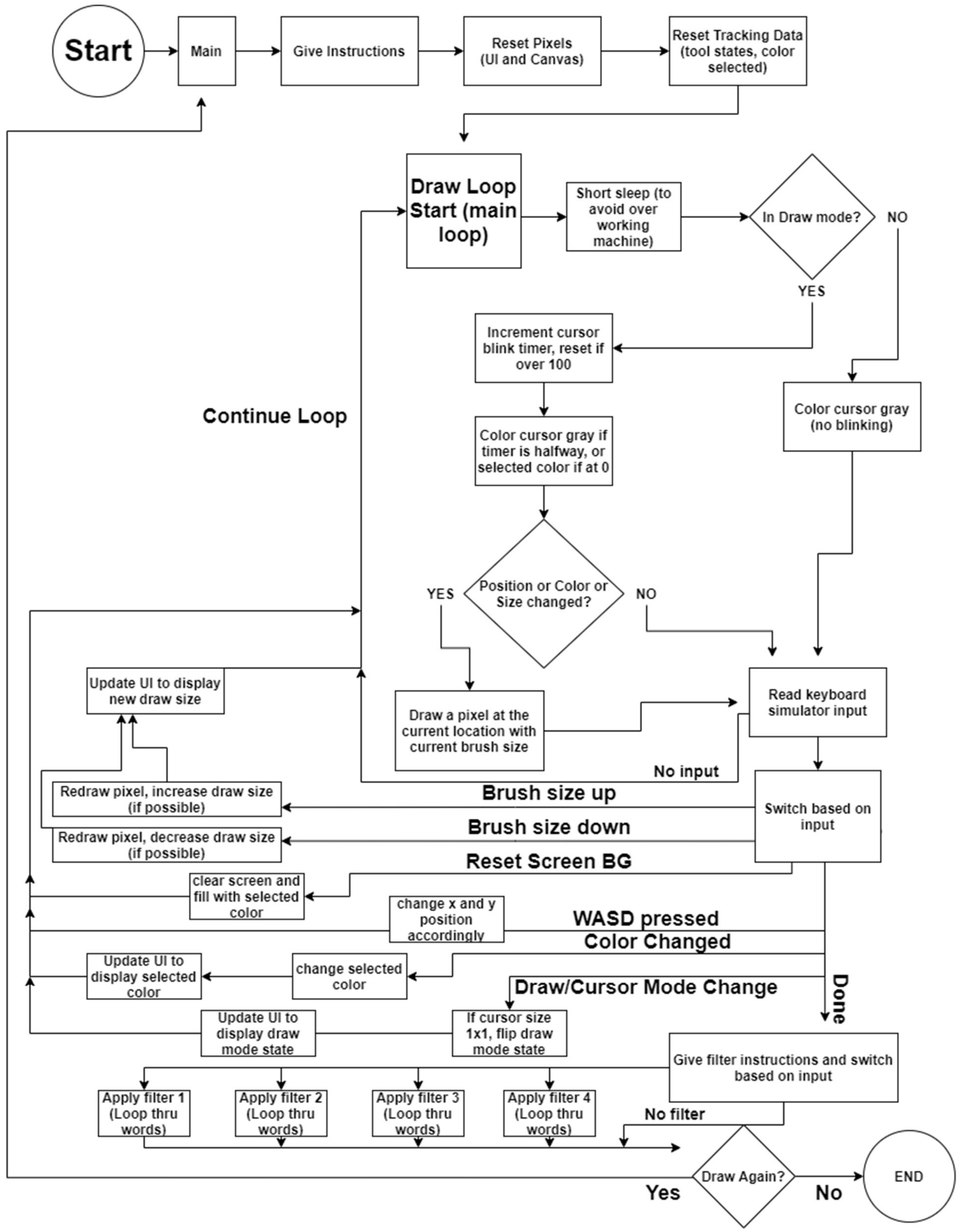
Original



Warmer Colors Filter



Flowchart of Main Program Logic:


Start
Main
Give Instructions
Reset Pixels (UI and Canvas)
Reset Tracking Data (tool states, color selected)
Draw Loop Start (main loop)
Short sleep (to avoid over working machine)
In Draw mode?
NO
YES
Increment cursor blink timer, reset if over 100
Color cursor gray (no blinking)
Continue Loop
Color cursor gray if timer is halfway, or selected color if at 0
Position or Color or Size changed?
YES
NO
Update UI to display new draw size
Draw a pixel at the current location with current brush size
Read keyboard simulator input
No input
Redraw pixel, increase draw size (if possible)
Brush size up
Switch based on input
Brush size down
Redraw pixel, decrease draw size (if possible)
Reset Screen BG
clear screen and fill with selected color
change x and y position accordingly
WASD pressed
Color Changed
Update UI to display selected color
change selected color
Draw/Cursor Mode Change
Done
Update UI to display draw mode state
If cursor size 1x1, flip draw mode state
Give filter instructions and switch based on input
Apply filter 1 (Loop thru words)
Apply filter 2 (Loop thru words)
Apply filter 3 (Loop thru words)
Apply filter 4 (Loop thru words)
No filter
Draw Again?
Yes
No
END