How to Create Firework Images

National Chiao Tung University
Chun-Jen Tsai
12/23/2017

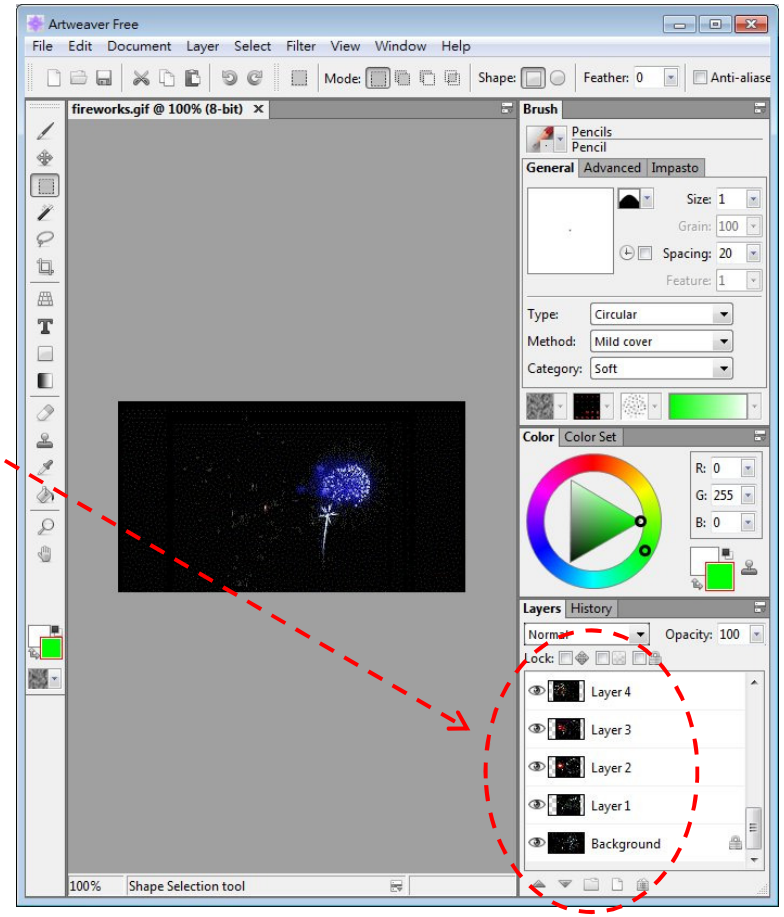
How to Create Fireworks for Lab 10

- You can use the same techniques of creating moon animations in the sample circuit to create fireworks

- Steps
  - Download a firework animation GIF image using web search
  - Use a free image editing tool to scale the GIF animation file to proper size and save each picture in the file as a PPM file
  - Convert the sequence of PPM files to a memory file for Verilog
  - Modify your circuit such that if the (x, y) position specified by the VGA sync controller is located in the firework area, show a pixel from one of the firework images

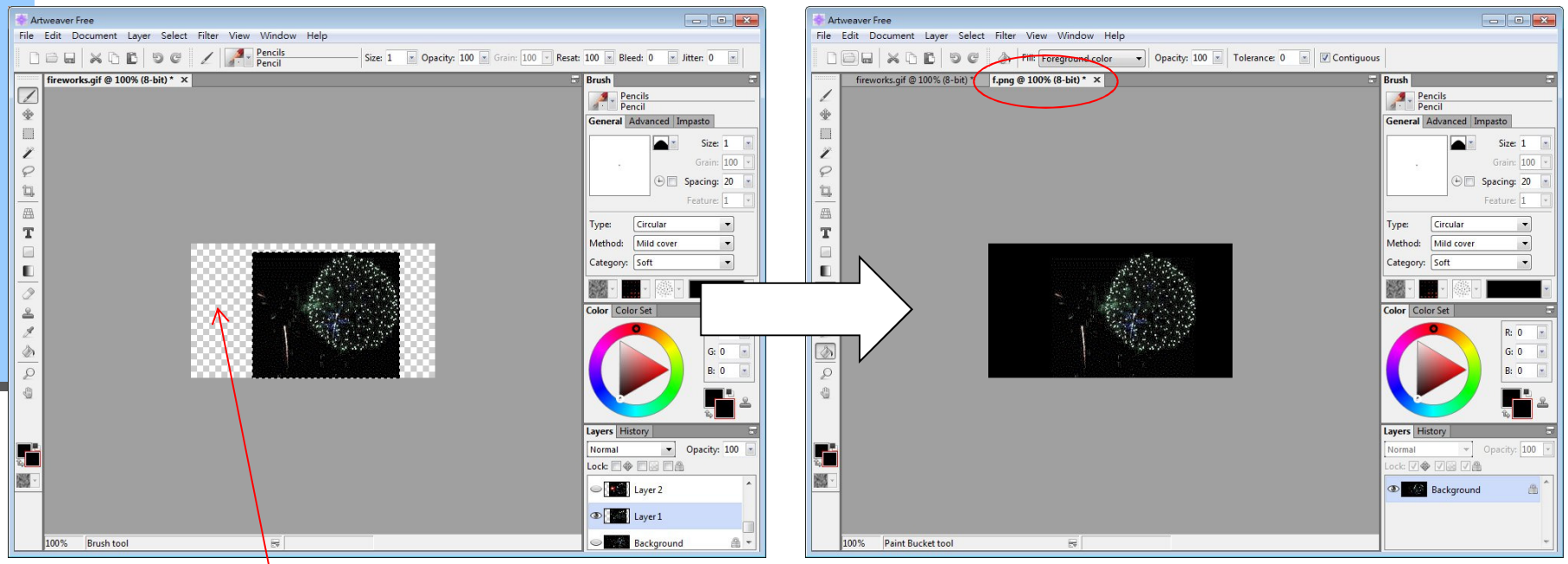
A Suggestion to Image Editing Tool

- One free image editing tool you can use is Artweaver
  - Each frame in the GIF animation file is treated as an image layer
  - You can scale all the layers to a smaller size at once
  - Then, you should select one layer at a time and save it as a 24-bit PNG image



Save Each Frame as PNG

- The transparent area of the GIF frame will become white pixels in the PNG file → open the PNG file using Artweaver and fill the white area with black pixels.



Converting PNG to PPM

- Now, you should have a sequence of 24-bit color PNG images that display the animated firework
- Use another free image tool, say Xnview, to convert the PNG images to PPM images (because Artweaver does not support the PPM image format)
- We can now convert the sequence of PPM images to a Verilog memory file using a simple C program

Converting *.PPM to *.MEM

- The following C code can convert a single 24-bit PPM image to a Verilog MEM file
  - You can concatenate one MEM files after another to create a large MEM file with all the animation images

```
#include <stdio.h>

unsigned char buf[320*240*3];

int main(int argc, char **argv)
{
    int width, height, idx;
    FILE *fp = fopen(argv[1], "rb");
    if (fp == NULL) return 1;

    fgets(buf, sizeof(buf), fp); fgets(buf, sizeof(buf), fp);
    sscanf(buf, "%d %d", &width, &height);
    fgets(buf, sizeof(buf), fp);
    fread(buf, width, height*3, fp); fclose(fp);

    for (idx = 0; idx < width*height; idx++)
        printf("%1x%1x%1x\n", buf[3*idx+0]>>4, buf[3*idx+1]>>4, buf[3*idx+2]>>4);

    return 0;
}
```

Assume that your animation image size is smaller than 320×240!

Final Comments

- You have about 98KB left in the on-chip memory of the FPGA for your animation images
- Your Verilog MEM files must be padded to the same size of the SRAM declared in your circuit
  - The MEM file is a ASCII text file
  - Each line the MEM file stands for the initial value of one memory cell in the SRAM
- You can declare a second SRAM block to store your animation images, or you can merge the animation images into the MEM file in the Lab 10 sample code