CSCI 3280 Introduction to Multimedia Systems

Spring 2017, Assignment 1 - Inverse ASCII Art

Due Date : Feb. 16, 2017 (11:59pm)

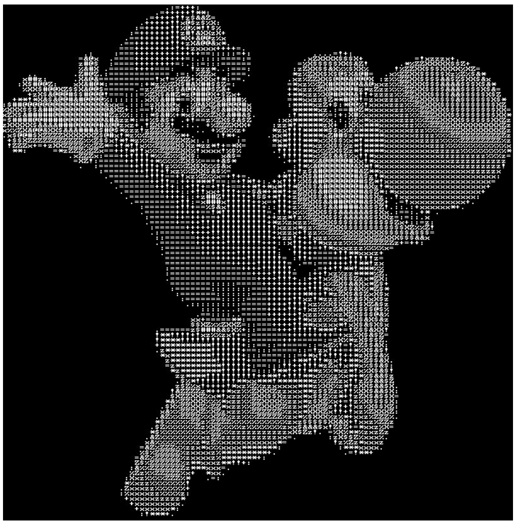
Submission via elearn.cuhk.edu.hk

Late submission penalty : 10-point deduction per day

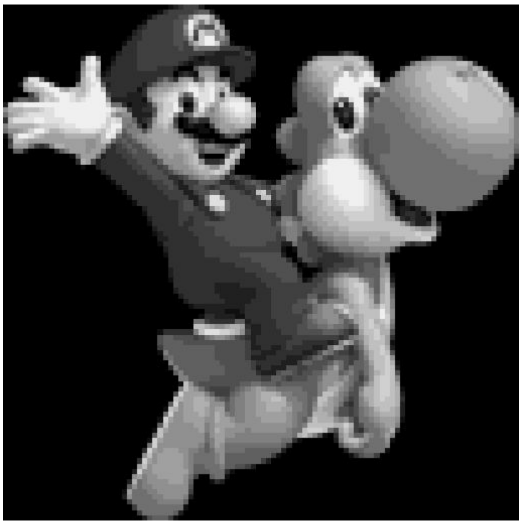
PLAGIARISM Penalty : Whole Course Failed

Introduction

Tone-based ASCII Art is an interesting art form which uses limited ASCII character set as their basic image elements and it was a popular way to print large graphics posters from dot matrix printers back in the old days but it still remains a popular art form. In this assignment, you are required to complete two small programs, the first one (`inverse.cpp`) converts a tone-based ASCII Art (16 tones) into gray-scale bitmap in .BMP format and the second one (`ascii.cpp`) converts regular RGB bitmap into 16-tone ASCII art.



into



General Requirements

1. Program must be coded in ANSI C/C++, no additional libraries allowed.
2. The compiled programs must run in **Windows** command prompt as a console program and accepts input with the following syntax.

```
C:\> inverse <art.txt> <art.bmp>
```

   `inverse` is your program executable
   `<art.txt>` is the full path name to the given ASCII art file
   `<art.bmp>` is the full path name for the output bitmap

3. A simple `.bmp` file library is included in the package (`bmp.h` and `bmp.cpp`).
4. You may assume input ASCII art or bitmap has size not bigger than 100 x 100.

5. You are required to **submit source code only**. We will use Visual Studio 2015 C++ compiler and have your program compiled via visual studio command prompt with the following command line.
   Please make sure your source code gets compiled well with it, 'failed to compile' receives 10-point deduction for each failed source.

```
C:\> cl.exe inverse.cpp bmp.cpp
```

6. ASCII art and bitmap test files are included for testing your programs.

Part 1 - Inverse ASCII Art (`inverse.cpp`, 60 points)

Complete the program source file `inverse.cpp`. Your program should process an ASCII art text file and output a gray-scale (R, G and B channels have the same value) bitmap in `.bmp` format. A basic 8-level ASCII character set is defined in `inverse.cpp`. and remember R,G and B of bitmaps can take values from 0-255 (`unsigned char` data type).

ASCII art file format:

```
<width> <height>
< row 0 ASCII characters >
< row 1 ASCII characters >
...
...
...
...
< row height-1 ASCII characters >
```

Example:

```
10 8
*#*#*#*#*#
*#*#*#*#*#
*#*#*#*#*#
*#*#*#*#*#
*#*#*#*#*#
*#*#*#*#*#
*#*#*#*#*#
*#*#*#*#*#
```

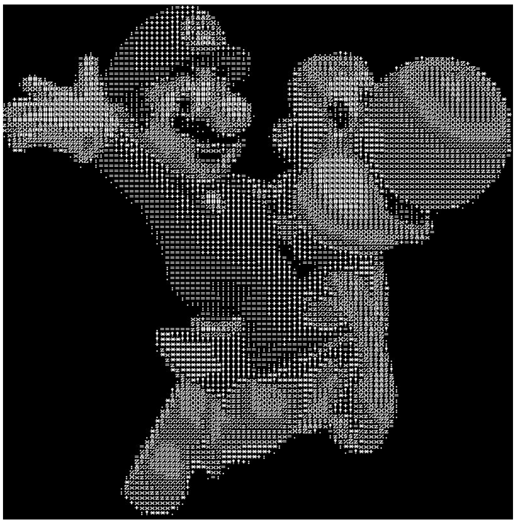
Part 2 - ASCII Art Generation (`ascii.cpp`, 40 points)

Complete the program source file `ascii.cpp`. Your program should process a given RGB bitmap file in `.bmp` format and produce an ASCII art file with the file format given in Part 1. You have to convert RGB into gray-scale by using the following formula:

Gray = 0.299 * R + 0.587 * G + 0.114 * B;

Gray-scale value should then be quantized into 16 levels properly. The same 16-level ASCII character set used in part 1 should be used to produce your result.

into



Bonus Part (10 points)

You are encouraged to implement the following enhancement plus some features that you find interesting and put this bonus-part program into its own standalone source file named **`bonus.cpp`**

- Convert a given RGB bitmap into a **Colored ASCII Art** HTML page for web browser (Try to use CSS style to minimize the size of the HTML !!) with your own ASCII character set. Be Creative !

Submission (Deadline : Feb. 16, 2017 11:59pm)

We expect the following files zipped into a file named by your CWEM (e.g. s1234567890.zip) and have it uploaded to the course's elearn page.

- **`README.TXT`** (Tell us what to pay attention to, especially about the bonus part)
- **`inverse.cpp`** (Part 1 source code)
- **`ascii.cpp`** (Part 2 source code)
- **`bonus.cpp`** (OPTIONAL, bonus part source code)