Algorithm 1 Brush

Introduction to Computer Graphics, Fall 2012

Due Sunday, September 9 at noon

Instructions: Complete this assignment by yourself without any help from anyone or anything except a current CS123 TA, the lecture notes, official textbook, and the professor. Hand in the assignment yourself at the handin bin on the second floor of the CIT no later than noon on the due date. If you are sick or unable to go to the CIT, scan or type your answers and submit them in PDF format before the deadline by e-mail attachment to cs123tas@cs.brown.edu. Late hand-ins are not accepted under any circumstances. This assignment is worth 10% of your final grade for Brush.

1 Blending

You will blend the color of the brush with the color on the canvas using the mask mentioned in the handout. Although the image on your canvas will be colored, for this exercise assume that your image is grayscale and has only one channel, called intensity, which ranges in floating point from 0 (totally black) to 1 (totally white). What is the value of the final intensity F on the canvas, given the original color intensity of the canvas $C \in [0,1]$, the value of the mask at that point $M \in [0,1]$, the current brush intensity (B), and the current "alpha" value $\alpha \in [0,1]$? Think of the α as the flow rate of the virtual paint your brush is putting down on the canvas. Hint: consider when α=0 and 1.

[2 points]

F$= M \cdot \alpha \cdot B + (1 - M \cdot \alpha) \cdot C$**... or equivalent expression**

2 Mouse Interaction

Given a click point (x, y), canvas dimensions (w, h) where w = width and h = height, and a mask radius R, you will need to figure out what area to iterate over in your drawing loop. If the following represents the core of your drawing loop, fill in the blanks shown in the C++ code below: (Remember that the mask will always have an odd width and height; a radius of 1 is a mask of width 3, a radius of 2 is a mask of width 5, et cetera).

```
/** given: w, h, R, x, y. You can use MIN(j,k), MAX(j,k), or if
    statements in the blank space if it makes your job easier. **/

    int rowstart = /*1 pt*/ MAX(0, mouseY-R);

    int rowend = /*1 pt*/ MIN(height, mouseY+R+1);

    int colstart = /*1 pt*/ MAX(0, mouseX-R);

    int colend = /*1 pt*/ MIN(width, mouseX+R+1);

    int rowcounter;
    int colcounter;

for (rowcounter=rowstart; rowcounter<rowend; rowcounter++)
{
    for (colcounter=colstart; colcounter<colend; colcounter++)
    {
        //...do stuff to the image at (rowcounter, colcounter)...
    }
}
```

3 Image Data

On a modern microprocessor, the cache allows for especially efficient access to contiguous memory locations; that is, it is faster to access memory in sequential order than to jump around a lot. As stated in the assignment handout, the data for brush is stored in row-major order. For an image canvas (the normal canvas2d explained in the assignment handout, not the monochromatic canvas from the first problem) with dimensions "width = 256" and "height = 256", answer the following questions for **1 point each**.

1. What is the pixel index of a pixel at row = 12 and col = 201 ? (Where the first pixel is at row 0, column 0)
   12*256+201=3273

2. What is the row and col of the pixel at pixel index 12345?
   Row=Floor12345÷256=48
   Col=12345 mod 256=57

3. How many bytes separate the beginning of one pixel from the beginning of the next horizontally adjacent pixel in memory? (That is, two pixels that are to the left or right of each other on the screen)
   One byte (eight bits) for each of the red, green, blue, and alpha channels, which range from 0 to 255

4. How many bytes separate the beginning of one pixel from the beginning of the next vertically adjacent pixel in memory? (That is, two pixels that are above or below each other on the screen)
   256*4 = 1024. An entire row of 256 pixels at 32 bytes each.