Algorithm 1 Brush

Introduction to Computer Graphics, Fall 2021

Due September 15 at 10:00pm EST

Instructions: You may discuss this assignment with other students, but you must abide by the rules stated in the collaboration policy (no notes from any discussions, your handin must be completely written up by you and contain only your own work, 15 minute delay between discussions and writing down any notes of your own) and write the logins of the student(s) you collaborated with next to each problem. Hand in the assignment on Gradescope as a PDF (please make sure the document is anonymous) no later than 6:00pm EST on the due date. *Late hand-ins are not accepted under any circumstances.*

1 Blending

You will blend the color of the brush with the color on the canvas using the mask mentioned in the Brush project 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 transparency or blend ratio of the virtual paint your brush is putting down on the canvas. Hint: consider when $\alpha = 0$ and 1.

[2 points]

F=

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 Figure 3 represents the core of your drawing loop, fill in the blanks shown in the C++ code. (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, etc.).

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 = 512" and "height = 256", answer the following questions for **1 point each**.

1. What is the pixel index of a pixel at row = 43 and col = 242 ? (Where the first pixel is at row 0, column 0, and the canvas is an array of RGBA structs that each represent one pixel)
2. What is the row and col of the pixel at pixel index 12345?
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)
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)

```
/** 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*/ ___________________;

int rowEnd = /*1 pt*/ _____________________;

int colStart = /*1 pt*/ ___________________;

int colEnd = /*1 pt*/ _____________________;

int rowCounter;
int colCounter;

for (rowCounter = rowStart; rowCounter < rowEnd; rowCounter++)
{
   for (colCounter = colStart; colCounter < colEnd; colCounter++)
   {
      //...do stuff to the image at (rowCounter, colCounter)...
   }
}
```

Figure 1: The code for question 2.