1. Introduction  
   **What is this program about?**This program is about implementing the algorithm for filling an object. There are several algorithms written in this program to fulfil the task given. Those algorithms are recursive flood fill, recursive boundary fill, eight-way region fill, region fill stack, scanline region fill, scanline region fill stack.  
   **In what language is the program implemented?**This program is implemented in Python using Python 3.x
2. Basic Theory  
   **Explain what region fill is about.**  
   Region fill is a term for algorithm in image processing, it deals simply with filling the region of an object/image. Region fill can be divided into two, Boundary fill and Flood fill  
   **Explain flood fill and boundary fill.**
3. Flood Fill

Flood fill is one of the fill region types, which will be filled using the pixel to pixel technique that have same color with seed pixel

1. Boundary Fill

Boundary fill is one of the fill region types, which will be filled using the pixel to pixel technique that have boundary or limit. The boundary or limit for this fill is the boundary color

**Explain how to perform a recursive pixel-to-pixel flood fill. Write down the pseudocode.**

* First of all we need to determine the color or the target color that we want to fill
* Then, we need to set the seed pixel
* If the color of the pixel is different with seed pixel then do nothing
* If the color of the pixel is same with seed pixel then we fill the pixel with target color
* Do the recursive for right, left, top, bottom

(x+1, y)

(x-1, y)

(x, y+1)

(x, y-1)

**Explain how to perform a recursive pixel-to-pixel boundary fill. Write down the pseudocode.**

* First of all we need to determine the color or the target color that we want to fill and the boundary color (border)
* Then, we need to set the seed pixel
* If the color of the target color and border are same with seed pixel then do nothing
* If the color of target color and border are different with seed pixel then we fill the pixel with target color
* Do the recursive for right, left, top, bottom

(x+1, y)

(x-1, y)

(x, y+1)

(x, y-1)

**What are the disadvantages of recursive pixel-to-pixel fill?**

The disadvantages for recursive pixel to pixel fill are if we want to fill the big space or big object will not work perfectly because there is a recursive limit. It also very slow because everytime we call the recursive function it will use get pixel method over and over again

**Explain how to perform a pixel-to-pixel flood fill using a stack. Write down the pseudocode.**

* First of all we need to determine the color or the target color that we want to fill
* Then, we set the empty stack or initiate the empty stack
* Set or insert the seed pixel to the stack
* Then we will do the looping until the stack is empty
* While looping we do:

1. Remove or Pop the seed pixel from the stack
2. If the color of the pixel is different with seed pixel then do nothing
3. If the color of the pixel is same with seed pixel then we fill the pixel with target color

Add left pixel (x-1, y) to the stack

1. Add right pixel (x+1, y) to the stack
2. Add top pixel (x, y+1) to the stack
3. Add bottom pixel (x, y-1) to the stack

**Explain how to perform a recursive scanline flood fill. Write down the pseudocode.**

**Explain how to perform a scanline flood fill using a stack. Write down the pseudocode.**

1. Implementation

**Explain the main interface of the program.**

**Explain every feature in the program and how to use them.**

1. Design

**Explain the main data structures (if any) used in the program.**

**Explain the main/global variables used in the program.**

**Explain how the bonuses (if done) are implemented.**

1. Evaluation

**Evaluate each case for region fill. Try the following test cases:**

* + Flood fill on a convex region (all 4 methods).
  + Flood fill on a non-convex region (all 4 methods).
  + Boundary fill on a convex region (all 4 methods).
  + Boundary on a non-convex region (all 4 methods).
  + Region fill on an empty screen (all 4 methods).

**Also perform a test case for all the bonuses you implemented.**

**Include screenshots of each test case.**

**Explain whether each case is successful.**

1. Work log

|  |  |  |
| --- | --- | --- |
| 04-March-2021 | Creating GUI, Line and Circle | Wirahmat, Andre |
| 05-March-2021 | Updating GUI, implement flood fill recursive, bound fill | Wirahmat, Andre, Aziza |
| 06-March-2021 | Implement 8 Way fill, flood fill stack, Debugging | Andre, Wirahmat, Aziza |
| 07-March 2021 | Debugging | Andre, Wirahmat, Aziza |
|  |  |  |

1. Conclusion and remarks.

**Does the program work as expected?**

**If some parts of the program do not work as expected, explain why.**

**What are your comments about this assignment?**

Due to lack of practice, example of programming and documentation of the library on the internet this assignment will be hard to implement despite the Difficulty being fairly low