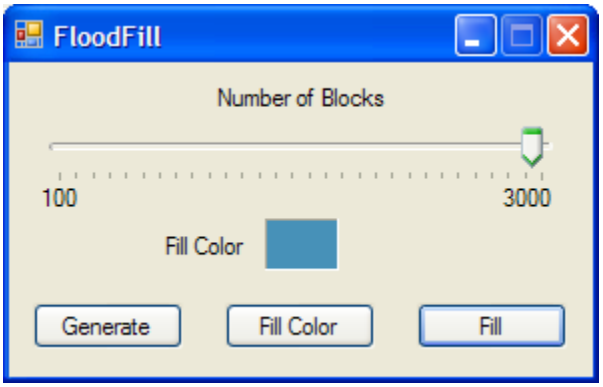
School of Applied Sciences And Technology

Department of IST

Program: CNT

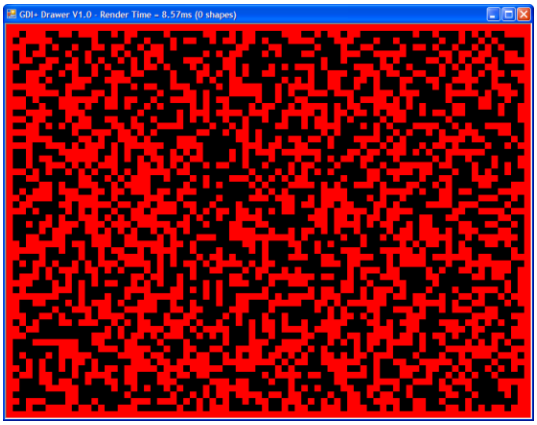
CMPE1666 – ICA10- FloodFill Algorithm

In this ICA, you will apply **recursion** to performing a flood fill on a graphical object. You will create a Windows application consisting of a single form, as shown below. The form should be fixed in size. Add the controls shown below and a ColorDialog.



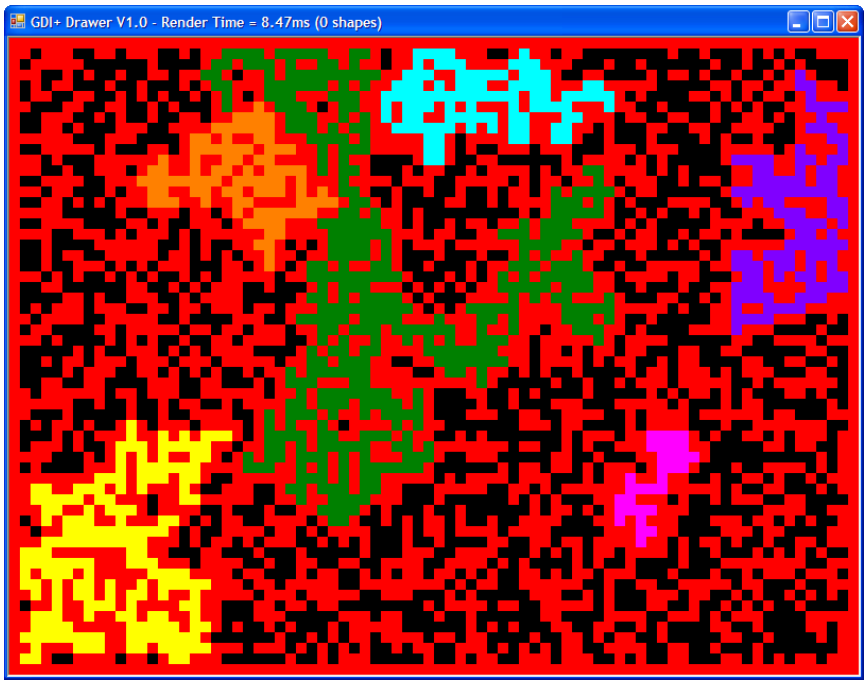
You will use the GDIDrawer to produce the graphical output. Set the Scale of the GDIDrawer window to 10, such that each pixel drawn will be a square that is 10x10 pixels in size.

Create a 2D array of Color values, with the dimensions 80x60. Each location of the array corresponds to a scaled pixel location in the drawing window. Locations that are red will be the boundaries within which the floodfill will take place. When the user presses the **Generate** button, clear the contents of the array by placing the color Black in each location. In the Color array, draw a boundary around the outer edges of the screen using red. Next, determine the number of blocks desired from the trackbar, then place pixels into random locations of the Color array using the color red. Display the contents of the array to the drawing window using SetBBScaledPixel, as shown below.



Initially, when the program starts, a default fill color will be used and displayed in the label placed above the **Fill Color** button. The user may press the **Fill Color** button to display the **ColorDialog**. If the user selects a new fill color, the color selected will be displayed in the label above the button.

Clicking the **Fill** button will start a timer the program will use to detect left mouse button click in the drawing window. When the mouse is clicked, the program will use the **recursive** method **FloodFill( )** to fill any connected black areas with the current fill color. If the mouse is clicked on a red square, nothing will happen. The example shown below is after I pressed Fill a few times using different fill colors.



FloodFill( ) Method

The FloodFill method will be used to **recursively** fill black areas of the drawing window with the desired fill color. FloodFill will receive as parameters the X and Y location of the location being checked, the target color to be filled (black), and the replacement color to be used to replace the target (fill color selected by user). The method will perform the following steps:

- If the color of the location being checked is not equal to the target color, return.
- If the color of the location being checked is the same as the replacement color, return.
- If both of the above tests fail, replace the color in that location of the array with the

replacement color, and draw that scaled pixel to the drawing window.
- Recursively call FloodFill four times; first to the point immediately left, then to the point immediately to the right, then above and below the current point.
- After all of that is done, return.

Your instructor will need to examine your code to ensure that recursion was used in its solution.

Rubric- Max Marks: 30

This application will require visual inspection of functionality and code.
Mark loss is at your instructor's discretion but will be applied consistently across all students.

Item	Marks	Penalties
UI Design (10)		
• UI is as directed.	2	
• Clicking on "Fill Color" button cause ColorDialog to popup.	2	
• Color chosen by user appears above Fill Color button	2	
• Professional layout	2	
• No. of blocks filled according to the Range value	2	
Code Design and Implementation (20)		Recursion not used for flood-fill: -10
• CDrawer object created	2	
• Boundaries filled with read color	3	
• Required number of blocks of red color placed at random locations	3	
• Areas filled, as required, with appropriate color when user clicks on black area of the CDrawer object. The application uses using recursive flood-fill algorithm	10	Using MouseLeftClick instead of polling: -3
• Timer disabled after executing floodfill until fill button is clicked again	2	
Documentation • Programmer Block • Well commented code • Appropriate Variable Names • Proper spacing between blocks of code • Control names are consistent and appropriate.		Missing components of documentation: -1 to -6