Electrostat Master

By Hitarth Choubisa(130070027) and Shorya Consul(13D070027)

- Objective:
  The objective of the project is to enable its user to easily visualise the variation of electric field for a given charge configuration (by mapping the electric field line pattern). The charge configuration constitutes an arrangement of point charges (position) as well as the magnitudes of the respective charges.

- Input:
  Firstly, the program will ask for the number of charges to be placed. The program has been designed such that it will quit upon receiving an input of greater than 5 for the number of charges to be placed.

  Otherwise, a screen (canvas) of 500×500 pixels will be initialised. The program will then ask for the magnitude of the charge to be placed (via the console) and subsequently, wait for the user to specify the location of the charge by clicking on the canvas.

  We have specified some sample inputs below (The locations of the charges have not been specified in these sample inputs.)

  1) Number of charges : 1
     Charge: -10

  2) Number of charges : 2
     Charge: 20, -10

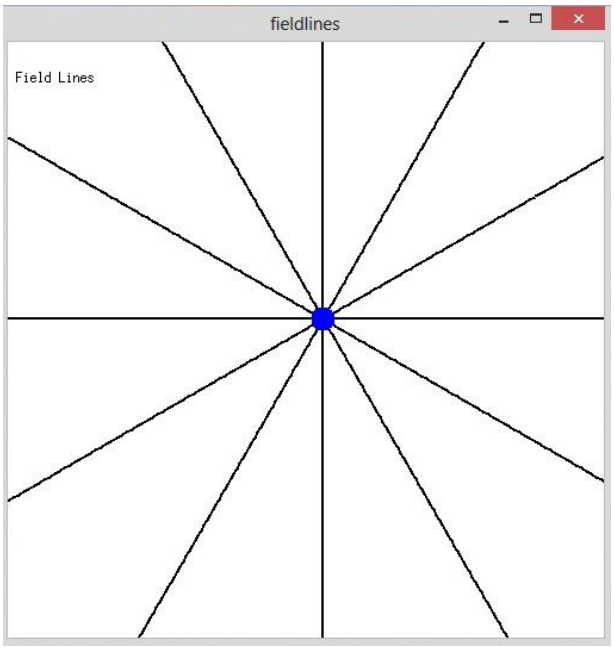
  3) Number of charges : 3
     Charge: 5, 1.564, 15

  4) Number of charges : 4
     Charge: -10, 20, -12, 5

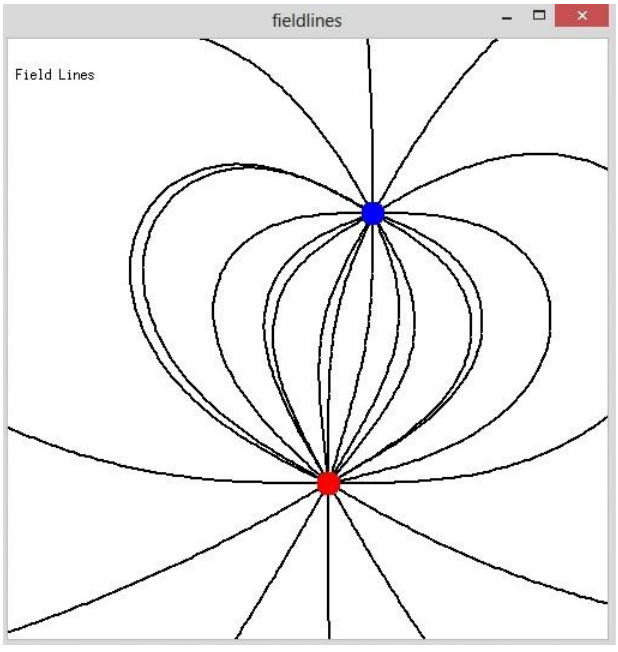
  5) Number of charges : 5
     Charge: 22, -25, -10, 5, 11

- Output :

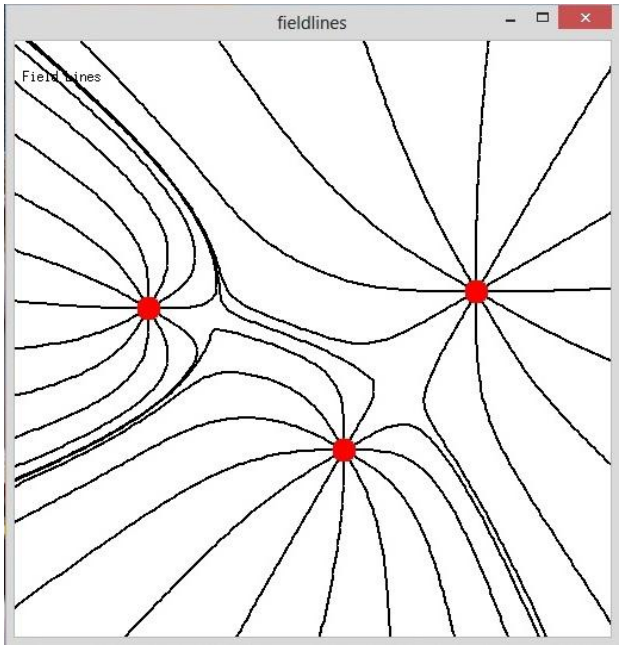
1)



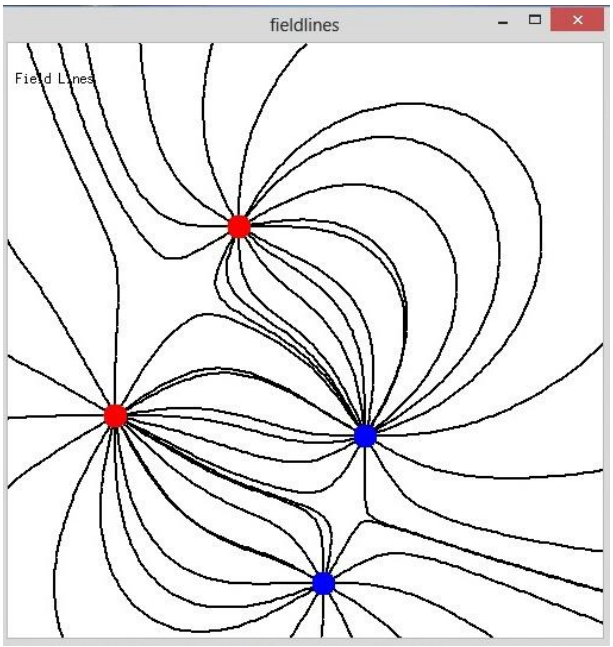
2)



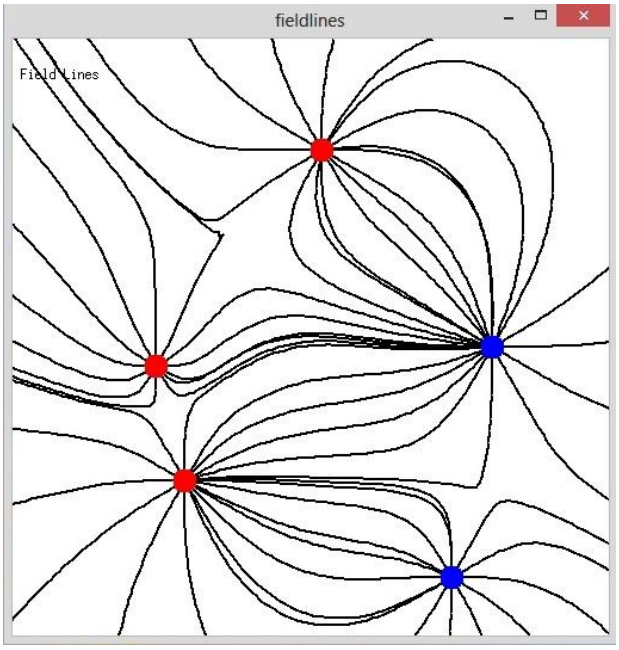
3)



4)



5)



The images shown above display the output on the canvas for the respective inputs given to the program. The positive charges are depicted by red circles while the negative charges are depicted by blue circles. This enables the user to distinguish between the two types of charges. On successful completion of the program, the message “FIELD LINES MAPPED SUCCESSFULLY” is also printed on the console.

- Algorithm:

1) First of all, a very small length straight line of length 5 pixels is drawn from one of the charge in the horizontal direction. The logic behind this is that electric field near a point charge is mostly influenced by charge only.

2) After then there is a bifurcation :
   i. **If the charge is positive:** At the end point of the initializer line, the unit vector in the direction of net electric field is calculated and the curve is advanced by 5 pixels along that direction.
   ii. **If charge is negative:** The only difference is that in this case, the direction vector is multiplied by (-1) so that the curve be extended in the correct direction. The logic for this is that for a negative charge the electric field lines terminate at it.
3) The process is continued till the line is completed by either reaching the boundary of the canvas or by being terminated at one of the negative charges.
4) The process is repeated for a line initialized at 30° to the earlier line.

5) The entire process is then repeated for the next point charge.

- Major Classes/ Functions:

1) Class V2: This is the vector class and is the core of the whole program. It has two components – the x component and the y component. It has a constructor giving its default value to be (0, 0). It also has a constructor which takes two arguments – the x-component and y-component – to create an object of the V2 class with the required x and y components.

2) Class ptcharge: This class is responsible for storing all the information about a specific charge. It stores its charge and its corresponding coordinates.

3) Field function: This function calculates the net electric field at any point and returns the net field, which is then used to calculate the direction of the net electric field at that point.

4) Check function: This function checks at every point whether the boundaries are crossed or whether there is a charge nearby (within a radius of 5 pixels). If any of the above are observed it returns true, else it returns false.

5) Mod function: This function is used to calculate the magnitude of the given vector.

- How to run the code:
  1) First of all execute the program using s++ compiler and make an executable.
  2) Run the executable.
  3) A message will appear asking you to give an input for the number of charges. Give an appropriate input.
  4) The canvas window will open. A new message will appear asking you give the charge on the particle. Give the desired charge and then, click on the window at the position where you want your charge to be placed. Continue till you have given all the requisite information.
  5) The electric field lines will be drawn on the canvas. Positive charges will be shown in red colour and negative charges in blue.
  6) After the field lines are drawn completely, the canvas will remain on the screen for 10 seconds before closing. If the program runs successfully, a message “FIELD LINESS MAPPED SUCCESFULLY” will be printed on the console.

- Speciality of the code:

The program fulfils all its basic objectives. Moreover, the resolution of the electric field pattern can be improved by making only small changes to the code itself – The line would have to be advanced by a smaller number of pixels in the main program. However, this would come at the expense of the speed of the program.

Moreover, the code handles the problem of lines terminating at negative charges and originating at positive charges by a simple conditional statement, which reverses the direction of the line to be printed if the charge is negative.

The vector class V2 has also been coded for ease of use as all the basic operations (+,-,* and modulus) have been included in the program.