|  |  |
| --- | --- |
| **Name:** | Yash Ravindra Kerkar |
| **Roll No:** |  |
| **Class/Sem:** | SE/III |
| **Experiment No.:** | 2 |
| **Title:** | Bresenham’s Line Drawing Algorithm |
| **Date of Performance:** |  |
| **Date of Submission:** |  |
| **Marks:** |  |
| **Sign of Faculty:** |  |

**Experiment No. 2**

|  |  |
| --- | --- |
| **Aim** | Write a program to implement Bresenham Line Drawing Algorithm in C. |
| **Objective** | To implement Bresenham line drawing algorithm for drawing a line segment between two points A (x1, y1) and B (x2, y2) |
| **Theory** | **Bresenham Algorithm**  This algorithm is used for scan converting a line. It was developed by Bresenham. It is an efficient method because it involves only integer addition, subtractions, and multiplication operations. These operations can be performed very rapidly so lines can be generated quickly.  In this method, next pixel selected is that one who has the least distance from true line.    **Basic Concept:**  Move across the x axis in unit intervals and at each step choose between two different y coordinates.  For example, from position (2, 3) we have to choose between (3, 3) and (3, 4). We would like the point that is closer to the original line.  So we have to take decision to choose next point. So next pixels are  selected based on the value of decision parameter p. The equations are given in below algorithm |

|  |  |
| --- | --- |
| **Algorithm** | **Step1:** Start Algorithm  **Step2:** Declare variable x1,x2,y1,y2,d,i1,i2,dx,dy  **Step3:** Enter value of x1,y1,x2,y2  Where x1,y1are coordinates of starting point And x2,y2 are coordinates of Ending point  **Step4:** Calculate dx = x2-x1  Calculate dy = y2-y1 Calculate i1=2\*dy Calculate i2=2\*(dy-dx) Calculate d=i1-dx  **Step5:** Consider (x, y) as starting point and xend as maximum possible value of x.  If dx < 0  Then x = x2 y = y2 xend=x1  If dx > 0  Then x = x1 y = y1  xend=x2  **Step6:** Generate point at (x,y)coordinates.  **Step7:** Check if whole line is generated.  If x > = xend Stop.  **Step8:** Calculate co-ordinates of the next pixel If d < 0  Then d = d + i1 If d ≥ 0  Then d = d + i2 Increment y = y + 1  **Step9:** Increment x = x + 1  **Step10:** Draw a point of latest (x, y) coordinates  **Step11:** Go to step 7  **Step12:** End  **Code:**  #include<graphics.h>  #include<stdio.h>  #include<conio.h>  int main()  {  int x,y,x1,y1,x2,y2,p,dx,dy;  int gd=DETECT,gm=0;  initgraph(&gd,&gm, "");  printf("\n Enter x1 cordinate: ");  scanf("%d",&x1);  printf("\n Enter y1 cordinate: ");  scanf("%d",&y1);  printf("\n Enter x2 cordinate: ");  scanf("%d",&x2);  printf("\n Enter y2 cordinate: ");  scanf("%d",&y2);    x=x1;  y=y1;  dx=x2-x1;  dy=y2-y1;    putpixel (x,y, RED);  p = (2 \* dy-dx);    while(x <= x2)  {  if(p<0)  {  x = x+1;  p = p + 2\*dy;  }  else  {  x = x + 1;  y = y + 1;  p = p + (2 \* dy) - (2 \* dx);    }  putpixel (x,y, RED);    }  getch();  closegraph();  } |
| **Output** |  |

**Conclusion:**  In this practical, we learned how to draw a line on the screen using the Bresenham's Line Drawing algorithm. The program takes two sets of coordinates (x1, y1) and (x2, y2) as input and calculates the intermediate points to draw a line between them.