Pimpri Chinchwad Education Trust's
Pimpri Chinchwad College of Engineering, Pune

Department of Computer Engineering, 2018

COMPUTER GRAPHICS
Mini Project

Animation for walking man with balloons

Faculty: Ms. Anagha Chaudhari

By-
Bhargavi JahagirdarSECOA136
Vaibhav Joshi   SECOA141

AIM :- Write C++ program to simulate walking man animation on ground with balloons in his hand

OBJECTIVE :- To generate walking man animation using line drawing and circle drawing functions in compute.

THEORY :-
Circle using Bresenham's algorithm :-
We cannot display a continuous arc on the raster display. Instead, we have to choose the nearest pixel position to complete the arc. From the following illustration, you can see that we have put the pixel at (X, Y) location and now need to decide where to put the next pixel − at N (X+1, Y) or at S (X+1, Y-1).
This can be decided by the decision parameter d.

If d <= 0, then N(X+1, Y) is to be chosen as next pixel.

If d > 0, then S(X+1, Y-1) is to be chosen as the next pixel.
Algorithm

Step 1 − Get the coordinates of the center of the circle and radius, and store them in x, y, and R respectively. Set P=0 and Q=R.

Step 2 − Set decision parameter D = 3 – 2R.

Step 3 − Repeat through step-8 while P ≤ Q.

Step 4 − Call Draw Circle (X, Y, P, Q).

Step 5 − Increment the value of P.

Step 6 − If D < 0 then D = D + 4P + 6.

Step 7 − Else Set R = R - 1, D = D + 4(P-Q) + 10.

Step 8 − Call Draw Circle (X, Y, P, Q).

DDA line drawing algorithm :-
In computer graphics, a digital differential analyzer (DDA) is hardware or software used for interpolation of variables over an interval between start and end point. DDA is used for rasterization of lines, triangles and polygons. In any 2 Dimensional plane if we connect two points (x0, y0) and (x1, y1), we get a line segment. But in the case of computer graphics we can not directly join any two

coordinate points, for that we should calculate intermediate point's coordinate and put a pixel for each intermediate point, of the desired color with help of functions like putpixel(x, y, K) in C, where (x,y) is our co-ordinate and K denotes some color.

Algorithm

Step 1 – Get the input of two end points (X0,Y0) and (X1,Y1).

Step 2 – Calculate the difference between two end points.

```
dx = X1 - X0
dy = Y1 - Y0
```

Step 3 – Based on the calculated difference in step-2, you need to identify the number of steps to put pixel.

If dx > dy, then you need more steps in x coordinate; otherwise in y coordinate.

```
if (absolute(dx) > absolute(dy))
Steps = absolute(dx);
else
Steps = absolute(dy);
```

Step 4 – Calculate the increment in x coordinate and y coordinate.

```
Xincrement = dx / (float) steps;
Yincrement = dy / (float) steps;
```

Step 5 – Put the pixel by successfully incrementing x and y coordinates accordingly and complete the drawing of the line.

```
for(int v=0; v < Steps; v++)
{
x = x + Xincrement;
y = y + Yincrement;
putpixel(Round(x), Round(y));
}
```

Step 6: stop.

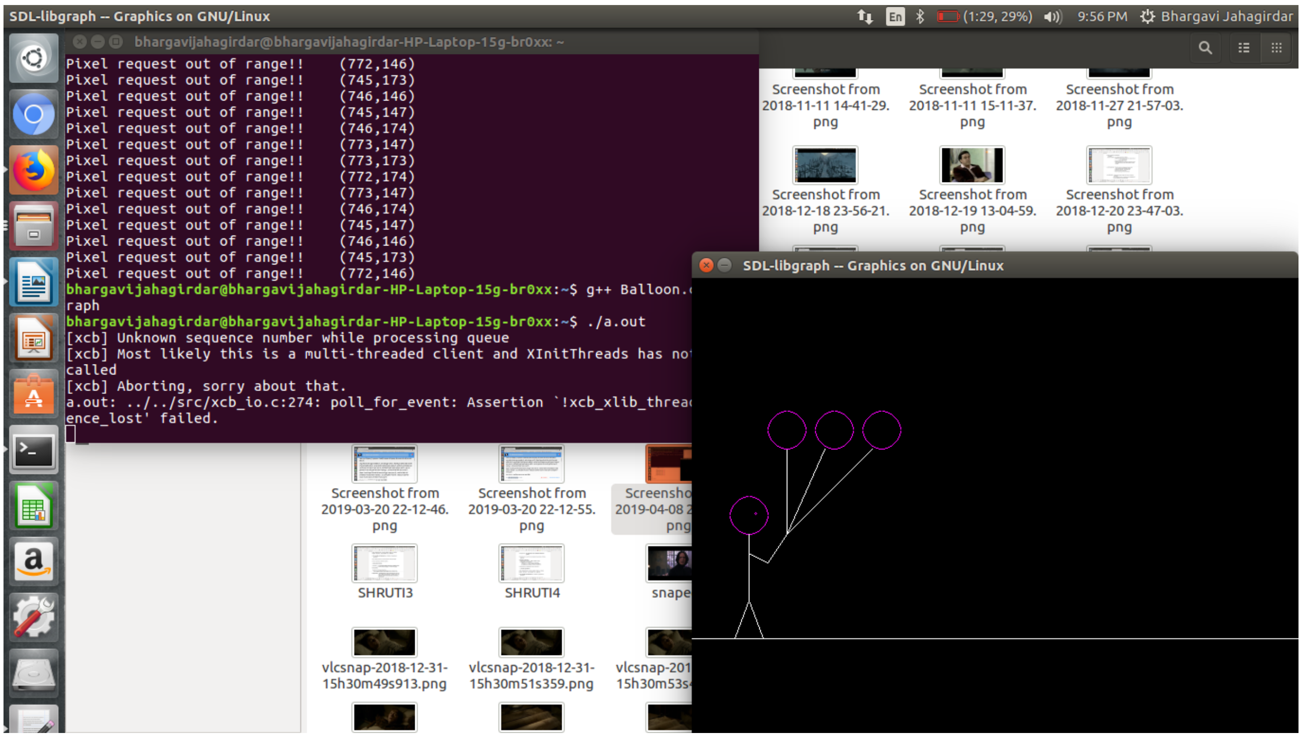
Code:

```
#include<iostream>
#include<graphics.h>
#include<math.h>
using namespace std;
class pixel
{
   public:
   void put_pixel(int x,int y,int z)
   {
     putpixel(x,y,5);
   }
};

class Circle
{
public:
             void cir(int xc,int yc,int r)
    {
       int d=3-(2*r);
       int x=0;
       int y=r;
       putpixel(xc+x,yc+y,5);
       while(x<=y)
       {
          if(d<0)
          {
             x=x+1;
             y=y;
             d=d+(4*x)+6;
          }
          else
          {
             x=x+1;
             y=y-1;
             d=d+(4*(x-y))+10;
          }
          putpixel(xc+x,yc+y,5);
          putpixel(xc+y,yc+x,5);
          putpixel(xc+x,yc-y,5);
          putpixel(xc-y,yc+x,5);
          putpixel(xc-x,yc-y,5);
          putpixel(xc-y,yc-x,5);
          putpixel(xc-x,yc+y,5);
          putpixel(xc+y,yc-x,5);
       }
}
void dda(int x1,int y1,int x2,int y2)
{
int length;
int i=1;
```

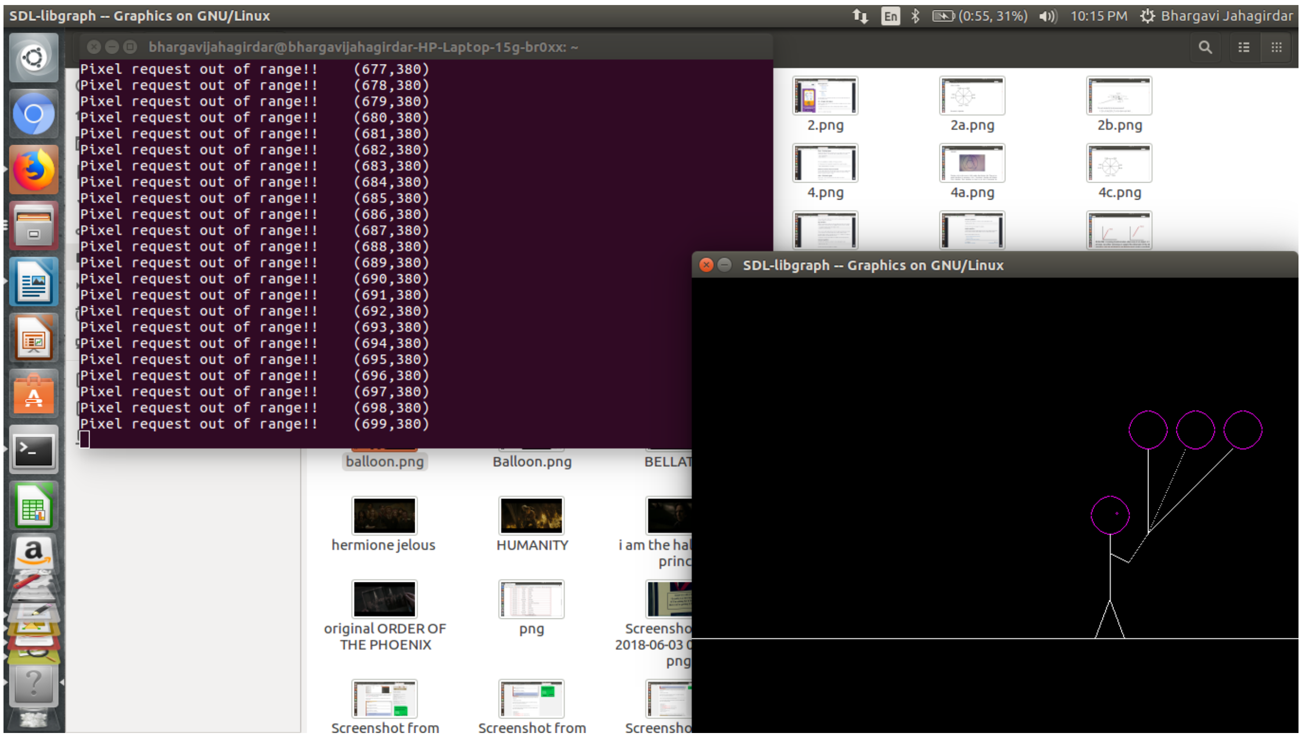
```
double dx,dy,x,y;

        dx=x2-x1;
        dy=y2-y1;

        if(dx>=dy)
                length=dx;
        else
                length=dy;

        dx=dx/length;
        dy=dy/length;

        if(dx>=0)
                x=x1+0.5*1;
        else
                x=x1+0.5*(-1);
        if(dy>=0)
                y=y1+0.5*1;
        else
                y=y1+0.5*(-1);

        while(i<=length)
        {
                putpixel(x,y,WHITE);
                x=x+dx;
                y=y+dy;
                i++;
        }
}
};

int main()
{
Circle obj;
 int gd=DETECT,gm;
 initgraph(&gd,&gm,NULL);

 for(int i=0;i<600;i++)
 {
        //Head
        obj.cir(20+i,250,20);
        obj.dda(20+i,270,20+i,340);

        //Eye
        obj.cir(27+i,248,1);

        //leg
        obj.dda(20+i,340,5+i,380);
        obj.dda(20+i,340,35+i,380);

        //hand
```

```
            obj.dda(20+i,290,40+i,300);
            obj.dda(40+i,300,60+i,270);

            //1st Balloon
            line(60+i,270,60+i,180);
            obj.cir(60+i,160,20);

            //2nd Balloon
            obj.dda(60+i,270,100+i,180);
            obj.cir(110+i,160,20);

            //3rd Balloon
            obj.dda(60+i,270,150+i,180);
            obj.cir(160+i,160,20);

            //Road
            obj.dda(0,380,700,380);

            delay(40);
            cleardevice();
 }
 closegraph();
}
```

Output:


SDL-libgraph -- Graphics on GNU/Linux
(1:29, 29%)
9:56 PM
Bhargavi Jahagirdar
bhargavijahagirdar@bhargavijahagirdar-HP-Laptop-15g-br0xx: ~
SDL-libgraph -- Graphics on GNU/Linux

```
Pixel request out of range!!    (772,146)
Pixel request out of range!!    (745,173)
Pixel request out of range!!    (746,146)
Pixel request out of range!!    (745,147)
Pixel request out of range!!    (746,174)
Pixel request out of range!!    (773,147)
Pixel request out of range!!    (773,173)
Pixel request out of range!!    (772,174)
Pixel request out of range!!    (773,147)
Pixel request out of range!!    (746,174)
Pixel request out of range!!    (745,147)
Pixel request out of range!!    (746,146)
Pixel request out of range!!    (745,173)
Pixel request out of range!!    (772,146)
bhargavijahagirdar@bhargavijahagirdar-HP-Laptop-15g-br0xx:~$ g++ Balloon.c
raph
bhargavijahagirdar@bhargavijahagirdar-HP-Laptop-15g-br0xx:~$ ./a.out
[xcb] Unknown sequence number while processing queue
[xcb] Most likely this is a multi-threaded client and XInitThreads has no
called
[xcb] Aborting, sorry about that.
a.out: ../../src/xcb_io.c:274: poll_for_event: Assertion `!xcb_xlib_threa
ence_lost' failed.
```


SDL-libgraph -- Graphics on GNU/Linux
(0:55, 31%)
10:15 PM
Bhargavi Jahagirdar
bhargavijahagirdar@bhargavijahagirdar-HP-Laptop-15g-br0xx: ~
SDL-libgraph -- Graphics on GNU/Linux

```
Pixel request out of range!!    (677,380)
Pixel request out of range!!    (678,380)
Pixel request out of range!!    (679,380)
Pixel request out of range!!    (680,380)
Pixel request out of range!!    (681,380)
Pixel request out of range!!    (682,380)
Pixel request out of range!!    (683,380)
Pixel request out of range!!    (684,380)
Pixel request out of range!!    (685,380)
Pixel request out of range!!    (686,380)
Pixel request out of range!!    (687,380)
Pixel request out of range!!    (688,380)
Pixel request out of range!!    (689,380)
Pixel request out of range!!    (690,380)
Pixel request out of range!!    (691,380)
Pixel request out of range!!    (692,380)
Pixel request out of range!!    (693,380)
Pixel request out of range!!    (694,380)
Pixel request out of range!!    (695,380)
Pixel request out of range!!    (696,380)
Pixel request out of range!!    (697,380)
Pixel request out of range!!    (698,380)
Pixel request out of range!!    (699,380)
```