Bouncy ball

#include<stdio.h>

#include<conio.h>

#include<dos.h>

#include<graphics.h>

int main()

{

int gd = DETECT,gm;

int x, y = 0, j, t = 400, c = 1;

initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

setcolor(RED);

setfillstyle(SOLID\_FILL, RED);

for (x = 40; x < 602; x++)

{

cleardevice();

circle(x, y, 30);

floodfill(x, y, RED);

delay(50);

if (y >= 400)

{

c = 0;

t -= 20;

}

if (y <= (400 - t))

c = 1;

y = y + (c ? 15 : -15);

}

getch();

closegraph();

return 0;

Dda line

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<dos.h>

#include<graphics.h>

void ddaLine(int x1,int y1,int x2,int y2)

{

int dx, dy, length, i;

float x, y, xinc, yinc;

dx = x2 - x1;

dy = y2 - y1;

if (abs(dx) > abs(dy))

length = abs(dx);

else

length = abs(dy);

xinc = dx / (float)length;

yinc = dy / (float)length;

x = x1;

y = y1;

putpixel(x, y, 15);

for (i = 0; i < length; i++)

{

putpixel(x, y,15);

x = x + xinc;

y = y + yinc;

delay(10);

}

}

int main()

{

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

setcolor(WHITE);

ddaLine(150,180,250,180);

ddaLine(250,180,250,300);

ddaLine(250,300,150,300);

ddaLine(150,300,150,180);

ddaLine(250,180,420,180);

ddaLine(420,180,420,300);

ddaLine(420,300,250,300);

ddaLine(250,300,250,180);

ddaLine(180,250,220,250);

ddaLine(220,250,220,300);

ddaLine(220,300,180,300);

ddaLine(180,300,180,250);

ddaLine(200,100,150,180);

ddaLine(200,100,250,180);

ddaLine(200,100,370,100);

ddaLine(370,100,420,180);

setfillstyle(SOLID\_FILL, CYAN);

floodfill(152, 182, WHITE);

floodfill(252, 182, WHITE);

setfillstyle(SLASH\_FILL, RED);

floodfill(182, 252, WHITE);

setfillstyle(HATCH\_FILL, RED);

floodfill(200, 105, WHITE);

floodfill(210, 105, WHITE);

getch();

closegraph();

return 0;

}

Flood fill

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<dos.h>

void floodFill(int x,int y,int fillcol,int oldcol)

{

if(getpixel(x,y) == oldcol)

{

putpixel(x,y,fillcol);

floodFill(x+1,y,fillcol,oldcol);

floodFill(x-1,y,fillcol,oldcol);

floodFill(x,y+1,fillcol,oldcol);

floodFill(x,y-1,fillcol,oldcol);

}

}

int main()

{

int gm,gd=DETECT;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

rectangle(50,50,300,300);

floodFill(51,51,14,0);

getch();

closegraph();

return 0;

}

Transformation

#include<stdio.h>

#include<conio.h>

#include<dos.h>

#include<math.h>

#include<graphics.h>

int main()

{

int gd=DETECT,gm,tx,ty,sx,sy,x1,y1,x2,y2;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

printf("BASIC TRANSFORMATIONS : \n");

setcolor(WHITE);

rectangle(50,50,100,100);

delay(1500);

cleardevice();

// Translation

tx=150,ty=50;

outtextxy(200,70,"Translation");

setcolor(RED);

rectangle(50+tx,50+ty,100+tx,100+ty);

delay(1500);

cleardevice();

// Rotation

double angle = (30\*3.14)/180;

setcolor(WHITE);

outtextxy(50,20,"Rotation");

setcolor(YELLOW);

x1 = (int)(50+((50-50)\*cos(angle)-(100-50)\*sin(angle)));

y1 = (int)(50+((50-50)\*sin(angle)+(100-50)\*cos(angle)));

x2 = (int)(100+((100-100)\*cos(angle)-(100-50)\*sin(angle)));

y2 = (int)(100+((100-100)\*sin(angle)+(100-50)\*cos(angle)));

line(50,50,x1,y1);

line(x1,y1,x2,y2);

line(x2,y2,100,50);

line(100,50,50,50);

delay(1500);

cleardevice();

// Scaling

sx=3,sy=4;

setcolor(WHITE);

outtextxy(180,170,"Scaling");

setcolor(GREEN);

rectangle(50\*sx,50\*sy,100\*sx,100\*sy);

delay(1500);

getch();

closegraph();

return 0;

}