#include <stdio.h>

#include <conio.h>

#include <graphics.h>

#include <dos.h>

void ballmotion(int,int,int,int,int,int,int,int);

void main(){

char exitChar;

//Margin for the arena

int left, top, right, bottom, margin;

int midx;

//Declaration for the ball

int h,k,radius;

//Initiation of the graphics

int gd = DETECT, gm;

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

setbkcolor(3);//Set The background color;

//Set the constraints for the arena;

margin = 100;

right = getmaxx() - margin;

left = margin;

top = margin;

bottom = getmaxy() - margin;

//middle line

midx = ((right-left)/2)+left;

//Fill color for the ball;

setcolor(6);

setfillstyle(SOLID\_FILL, 6);

circle(h,k,radius);

floodfill(h+1, k+1, 6);

startGame:

ballmotion(left, top, right, bottom, h, k, radius, midx);

gotoxy(50,3);

printf("\nGame Over");

while(1){

exitChar = getch();

if(int(exitChar) == 113){

break;

}

if(int(exitChar) == 114){

goto startGame;

}

}

closegraph();

}

void ballmotion(int left, int top, int right, int bottom, int h, int k, int radius, int midx){

//Left Bar

int llft = left;

int ltp = top + (bottom-top)/2-20;

int lrgt = llft + 5;

int lbtm = top + (bottom-top)/2+20;

//Right Bar

int rrgt = right;

int rlft = right-5;

int rtp = top + (bottom-top)/2-20;

int rbtm = top + (bottom-top)/2+20;

//Character => kbhit();

char ch;

//new position of the ball

int newh, newk, dx, dy;

//Initial position of the ball;

h = (right-left)/2 + left;

k = (bottom-top)/2 + top;

radius = 5;

//Score

int leftScore = 0;

int rightScore = 0;

//Generating a random number among 1 and 2;

dx = 1;

dy = 1;

while(1){

//New Position Setting;

newh = h + dx;

newk = k + dy;

h = newh;

k = newk;

delay(10);

cleardevice();

//Outline of the arena;

line(left, top, right, top);//Top line

line(left, top, left, bottom);//Vertical left line

line(right, top, right, bottom);//Vertical right line

line(left, bottom, right, bottom);//Horizontal Bottom line

line(midx, top, midx, bottom);//Partition line

//Bat left part;

//set line for the left bat and draw left bat;

setcolor(4);

setfillstyle(SOLID\_FILL, 4);

line(lrgt, top, lrgt, bottom);

bar(llft, ltp, lrgt, lbtm);

//set line for the right bat and draw right bat;

setcolor(1);

setfillstyle(SOLID\_FILL, 1);

line(rlft, top, rlft, bottom);

bar(rlft, rtp, rrgt, rbtm);

//Drawing of the new position of the ball

setcolor(6);

setfillstyle(SOLID\_FILL, 6);

circle(h,k,radius);

floodfill(h+1, k+1, 6);

//Conditions for the reflection

if((k+radius) > bottom || (k-radius) < top){

//condition for the upper and lower bound of the area

dy = -dy;

sound(400);

delay(25);

nosound();

}

//codition for the collision ith the right bat;

if((h+radius) >= rlft){

if(k >= rtp && k <= rbtm){

sound(1000);

delay(25);

nosound();

dx = -dx;

}

}

//Condition for the collission with the left bat;

if((h-radius) <= lrgt){

if(k >= ltp && k <= lbtm){

sound(1000);

delay(25);

nosound();

dx = -dx;

}

}

//Score panel

if((h-radius) < left){

sound(400);

delay(25);

nosound();

rightScore+=1;

dx = -dx;

}

if((h+radius) > right){

sound(400);

delay(25);

nosound();

leftScore+=1;

dx = -dx;

}

if(leftScore == 3){

printf("RED TEAM WIN");

break;

}

if(rightScore == 3){

printf("BLUE TEAM WIN");

break;

}

//Exit from loop

//keyboard input for moving the bat;

if(kbhit()){

ch = getch();

if(int(ch) == 113)

break;

if(int(ch) == 119){//setting for key w

if(ltp > top){

ltp-=5;

lbtm-=5;

}

}

if(int(ch) == 115){

if(lbtm < bottom){

ltp+=5;

lbtm+=5;

}

}

if(int(ch) == 111){

if(rtp > top){

rtp-=5;

rbtm-=5;

}

}

if(int(ch) == 108){

if(rbtm < bottom){

rtp+=5;

rbtm+=5;

}

}

}

}

}