/\*

THE FADED SHOOTING STARS

\*/

#include<graphics.h>

#include<windows.h>

#include<conio.h>

#include<process.h>

DWORD width,height;

inline void initiateWindow()

{

width = GetSystemMetrics(SM\_CXSCREEN);

height = GetSystemMetrics(SM\_CYSCREEN);

initwindow(width,height,"SCREEN SAVER");

}

void graphics()

{

int turn=1;

//get the cursor positions

POINT cursorPosition;

GetCursorPos(&cursorPosition);

int mx = cursorPosition.x;

int my = cursorPosition.y;

//initialise the x and y with center

int x=getmaxx()/2,y=getmaxy()/2;

while(!kbhit())

{

//if there's a mouse movement

//then exit like a screen saver

GetCursorPos(&cursorPosition);

if(mx!=cursorPosition.x&&my!=cursorPosition.y)

exit(0);

//generate random coordinates for x and y to fill the whole screen with black color

int px = rand()%width+1;

int py = rand()%height+1;

setcolor(BLACK);

rectangle(px,py,20,20);

//change the color according to the random turn generated.

switch(turn)

{

case 1:

setcolor(GREEN);

circle(x--,y--,4);

break;

case 2:

setcolor(WHITE);

circle(x++,y++,4);

break;

case 3:

setcolor(CYAN);

circle(x++,y++,4);

break;

case 4:

setcolor(RED);

circle(x++,y--,4);

break;

case 5:

setcolor(YELLOW);

circle(x--,y++,5);

break;

case 6:

setcolor(MAGENTA);

circle(x++,y--,5);

break;

case 7:

setcolor(LIGHTGRAY);

circle(x--,y++,4);

break;

case 8:

setcolor(DARKGRAY);

circle(x++,y--,4);

break;

case 9:

setcolor(BROWN);

circle(x--,y--,4);

break;

}

//if the pixel is out of the screen if(y==height||x==width||x==0||y==0) {

y=rand()%height+1;

x=rand()%width+1;

turn = rand()%8+1;

}

}

}

int main(){

initiateWindow();

graphics();

return 0;

closegraph();

}