EC-9

AN ACTION-PACKED DODGE AND CHASE EXPERIENCE

BE PREPARED TO BE THRILLED

COMPUTER SCIENCE PROJECT BY

RAGHAV THAKAR

XI-B; ROLL NUMBER - 16

THE MOTHER'S INTERNATIONAL SCHOOL

INTRODUCTION:

EC-9 is a self designed action-packed game wherein the objective is to collect 10 O's before while simultaneously avoiding the ever-present threat by C and E.

C and E are two enemies which move using two completely different methods. E uses the random() function to decide its very next step by moving to an adjacent point, and does that throughout the course of the game while C, uses the random() function to set a target anywhere on the screen, reaches the target, and repeats the process.

One wins the game upon successfully collecting 10 O's. Upon hitting either C or E, or moving out of the playfield, the player loses.

The kbhit() function is a function that I used here, which I didn't know before. It is what allows 9 to keep moving in a set direction with only a single key press, instead of having to individually press keys for every movement.

The game does not involve the use of graphics.h.

CODE:

```
#include <iostream.h>
#include <math.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <iomanip.h>
#include <dos.h>
int x=10, y=10, goalcounter=0;
int goalx=55, goaly=12;
int enemyx=24, enemyy=12;
int compx=25, compy=13, compxgoal, compygoal;
void movetogoal();
void compmove()
{
     int x1, x2, x3, y1, y2;
     randomize();
     x1=random(14);
     randomize();
     x2=random(4);
     randomize();
     x3=random(3);
     compxgoal=x1*x2*x3;
     if(compxgoal<6)
          compxgoal=6;
     randomize();
     y1=random(5);
     randomize();
     y2=random(5);
     compygoal=y1*y2;
     if(compygoal<6)
          compygoal=6;
     movetogoal();
}
void movetogoal()
{
     gotoxy(compx, compy);
     cout<<" ";
     if(compxgoal>compx)
          compx++;
     else if(compx==compxgoal && compygoal!=compy)
          compx=compxgoal;
     else
          compx--;
     if(compygoal>compy)
          compy++;
     else if(compx!=compxgoal && compygoal==compy)
          compy=compygoal;
     else
          compy--;
```

```
     gotoxy(compx, compy);
     cout<<"C";
     if(compx==compxgoal && compy==compygoal)
          compmove();
}
int enemycheck()
{
     if(x==enemyx && y==enemyy)
          return(1);
     else if(x==compx && y==compy)
          return(1);
     else
          return(0);
}
void enemymove()
{
     int enemydir;
     gotoxy(enemyx, enemyy);
     cout<<"E";
     randomize();
     enemydir=random(200);
     delay(20);
     if(enemydir>100 && enemydir<=150)
     {
          gotoxy(enemyx, enemyy);
          cout<<" ";
          gotoxy(enemyx, enemyy+1);
          cout<<" ";
          gotoxy(enemyx, enemyy-1);
          cout<<" ";
          gotoxy(enemyx+1, enemyy+1);
          cout<<" ";
          gotoxy(enemyx+1, enemyy-1);
          cout<<" ";
          gotoxy(enemyx+2, enemyy-1);
          cout<<" ";
          gotoxy(enemyx+2, enemyy);
          cout<<" ";
          gotoxy(enemyx+2, enemyy+1);
          cout<<" ";
          gotoxy(enemyx + 1, enemyy);
          cout<<"E";
          enemyx=enemyx+1;
          if(enemyx>54)
               enemyx=54;
          else if(enemyx<6)
               enemyx=6;
     }
     else if(enemydir>150)
     {
          gotoxy(enemyx, enemyy);
          cout<<" ";
          gotoxy(enemyx, enemyy+1);
          cout<<" ";
          gotoxy(enemyx, enemyy-1);
```

```
     cout<<" ";
     gotoxy(enemyx-1, enemyy+1);
     cout<<" ";
     gotoxy(enemyx-1, enemyy-1);
     cout<<" ";
     gotoxy(enemyx-2, enemyy-1);
     cout<<" ";
     gotoxy(enemyx-2, enemyy);
     cout<<" ";
     gotoxy(enemyx-2, enemyy+1);
     cout<<" ";
     gotoxy(enemyx - 1, enemyy);
     cout<<"E";
     enemyx=enemyx-1;
     if(enemyx>54)
          enemyx=54;
     else if(enemyx<6)
          enemyx=6;
}
if(enemydir<=100 && enemydir>50)
{
     gotoxy(enemyx, enemyy);
     cout<<" ";
     gotoxy(enemyx, enemyy+2);
     cout<<" ";
     gotoxy(enemyx+1, enemyy);
     cout<<" ";
     gotoxy(enemyx+1, enemyy+1);
     cout<<" ";
     gotoxy(enemyx+1, enemyy+2);
     cout<<" ";
     gotoxy(enemyx-1, enemyy-1);
     cout<<" ";
     gotoxy(enemyx-1, enemyy);
     cout<<" ";
     gotoxy(enemyx-1, enemyy+1);
     cout<<" ";
     gotoxy(enemyx, enemyy + 1);
     cout<<"E";
     enemyy=enemyy + 1;
     if(enemyy>23)
          enemyy=23;
     else if(enemyy<6)
          enemyy=6;
}
else if(enemydir<=50)
{
     gotoxy(enemyx, enemyy);
     cout<<" ";
     gotoxy(enemyx, enemyy+1);
     cout<<" ";
     gotoxy(enemyx+1, enemyy+1);
     cout<<" ";
     gotoxy(enemyx+1, enemyy-1);
     cout<<" ";
```

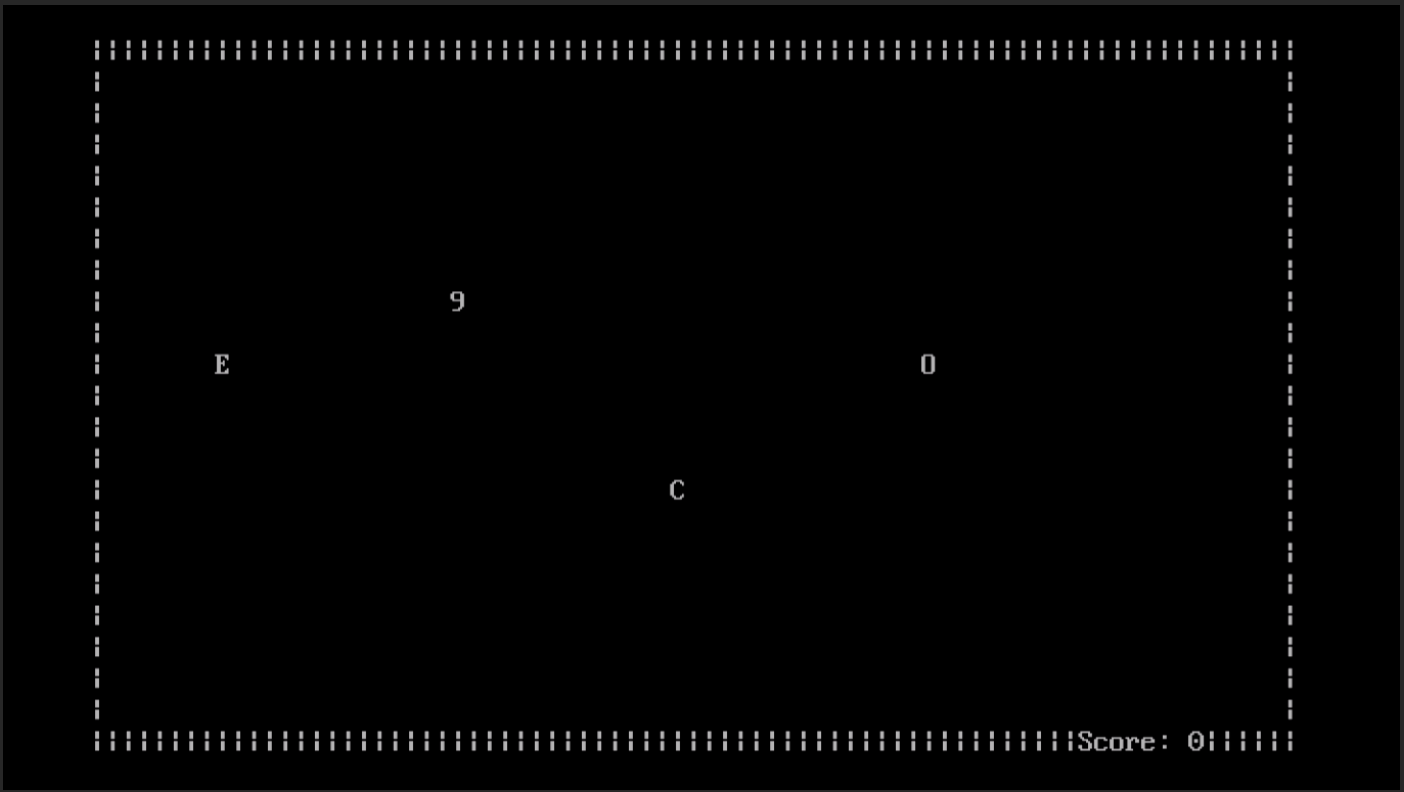
```
          gotoxy(enemyx+1, enemyy-1);
          cout<<" ";
          gotoxy(enemyx-1, enemyy);
          cout<<" ";
          gotoxy(enemyx-1, enemyy+1);
          cout<<" ";
          gotoxy(enemyx-1, enemyy-1);
          cout<<" ";
          gotoxy(enemyx, enemyy-1);
          cout<<"E";
          enemyy=enemyy-1;
          if(enemyy>23)
               enemyy=23;
          else if(enemyy<6)
               enemyy=6;
     }
}
int checkloss()
{
     if(x<2 || x>77)
          return(1);
     if(y<3 || y>23)
          return(1);
     else
          return(0);
}
void loss()
{
     clrscr();
     cout<<"YOU LOSE!!!!!!!!"<<endl;
     delay(500);
     cout<<"Press any key to exit";
     getch();
     exit(0);
}
void defineplayfield()
{
     for(int m=2; m<25; m++)
     {
          gotoxy(2, m);
          cout<<"|";
          gotoxy(78, m);
          cout<<"|";
     }
     for(int n=2; n<78; n++)
     {
          gotoxy(n, 2);
          cout<<"|";
          gotoxy(n, 24);
          cout<<"|";
     }
}
void victory()
{
     clrscr();
```

```
     cout<<"YOU HAVE WON!!!!!!!!"<<endl<<"Press any key to exit";
     getch();
     exit(0);
}
void printcounter()
{
     gotoxy(65, 24);
     cout<<"Score: "<<goalcounter;
}
void printgoal()
{
     gotoxy(goalx, goaly);
     cout<<"O";
}
void setgoal()
{
     randomize();
     goalx=random(64);
     if(goalx<6)
          goalx=6;
     goaly=random(18);
     if (goaly<6)
          goaly=6;
     gotoxy(goalx, goaly);
     cout<<"O";
     goalcounter++;
     if(goalcounter==10)
          victory();
}
void right()
{
     while(!kbhit())
     {
          gotoxy(x, y);
          cout<<" ";
          x=x+1;
          gotoxy(x, y);
          cout<<"9";
          delay(20);
          printgoal();
          defineplayfield();
          checkloss();
          printcounter();
          randomize();
          enemymove();
          enemycheck();
          compmove();
          if(checkloss()==1 || enemycheck()==1)
               loss();
          if(x==goalx && y==goaly)
               setgoal();
     }
}
void left()
{
```

```
       while(!kbhit())
       {
           gotoxy(x, y);
           cout<<" ";
           x=x-1;
           gotoxy(x, y);
           cout<<"9";
           delay(20);
           printgoal();
           defineplayfield();
           checkloss();
           randomize();
           printcounter();
           enemymove();
           enemycheck();
           compmove();
           if(checkloss()==1 || enemycheck()==1)
                loss();
           if(x==goalx && y==goaly)
                setgoal();
       }
}
void down()
{
     while(!kbhit())
       {
           gotoxy(x, y);
           cout<<" ";
           y=y+1;
           gotoxy(x, y);
           cout<<"9";
           delay(30);
           printgoal();
           defineplayfield();
           checkloss();
           randomize();
           printcounter();
           enemymove();
           enemycheck();
           compmove();
           if(checkloss()==1 || enemycheck()==1)
                loss();
           if(x==goalx && y==goaly)
                setgoal();
       }
}
void up()
{
     while(!kbhit())
     {
           gotoxy(x, y);
           cout<<" ";
           y=y-1;
           gotoxy(x, y);
           cout<<"9";
```

```
          delay(30);
          printgoal();
          defineplayfield();
          checkloss();
          randomize();
          printcounter();
          enemymove();
          enemycheck();
          compmove();
          if(checkloss()==1 || enemycheck()==1)
               loss();
          if(x==goalx && y==goaly)
               setgoal();
     }
}
void main()
{
     clrscr();
     char direction;
     defineplayfield();
     for(int i=0;; i++)
     {
          //gotoxy(1,1);
          printgoal();
          defineplayfield();
          direction=(getch());
          clrscr();
          if(direction == 'l')
               right();
          if(direction == 'k')
               down();
          if(direction == 'j')
               left();
          if(direction == 'i')
               up();
          if(direction == 'e')
               exit(0);
     }
}
```

OUTPUTS:



CONTROLLING *9*, TRYING TO COLLECT *O* WHILE AVOIDING *E* AND *C*

LOSS SCREEN

```
YOU HAVE WON!!!!!!!!
Press any key to exit_
```

VICTORY SCREEN

Thank you