

**CASE STUDY
SNAKE VS EAGLE**

Submitted by
121810313024(Kamisetty Venkata Harshith)
121810313053 (Yamini Muvvala)

Under the Supervision of
Mr.S.Amarnadh
Asst.Professor
**Department of CSE,
GITAM INSTITUTE OF TECHNOLOGY, GITAM**



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING
GITAM INSTITUTE OF TECHNOLOGY
GITAM
(DEEMED TO BE UNIVERISTY)
VISA KHAPATNAM
(2019-20)**

DECLARATION

We **Kamisetty Venkata Harshith (Reg. No.121810313024)** and **Yamini Muvvala (Reg. No.121810313053)** hereby declare that the case study report entitled “Title: **Snake Vs Eagle** ” is an original and authentic work done in the Department of CSE, GITAM Institute of Technology, GITAM Deemed to be University, Rushikonda, Visakhapatnam submitted in partial fulfilment of the requirement for the award of the degree of Bachelor of CSE.

Kamisetty Venkata Harshith
(Reg. No.121810313024)

Yamini Muvvala
(Reg. No.121810313053)

Signature

INTRODUCTION

Page No:-1

Snake vs Eagle game is a fight between the snake and the eagle where the snake fights for life and eagle for fun with the snake.

Rules and conditions:

1. Press any key to start the game.
2. 4 arrows are used to control the movement of the snake. When we use "up" arrow the snake moves up, "down" arrow snake moves down, "left" arrow snake moves left, "right" arrow snake moves right.
3. If we use up arrow or left arrow we cannot use the down arrow or right arrow immediate next to it respectively.
4. Press "p" to pause the game.
5. Press any key to resume except "p".
6. Press "esc" to exit at any instant of the game.
7. When the game finishes, press 'y' to continue the game.
8. Press 'n' to close the game.

Game point theory:

1. Snake will have 3 lives and dies automatically after those are lost.
2. This game is constrained in the walls. If the snake moves through the rightside wall it enters from the leftside wall and viceversa. If the snake moves through the upside wall it enters from the downside wall and viceversa.
3. The food (eggs) will be generated randomly at different locations. The snake has to go and eat the eggs.
4. If the snake is successful in eating the eggs, the snake length will increase by one unit and score will increase by 10 points.
5. There will be bonus eggs which generate randomly and stay for a limited period of time.
6. If snake eats bonus egg, the snake life will be increased by one and score will increase by 100 points.
7. Eagles will generate from different locations. If the snake touches the eagle nose, the snake length will be reduced by 2 units and will lose one life.
8. When eagle touches the egg, egg disappears with a beep sound.
9. When all the 3 lives are gone the game exits automatically or when the snake touches any other part of the snake the game exits automatically.
10. When game gets completed it displays the score & ask the user whether to play again or not.

➤ **rectangle():**

It is used to draw a rectangle. Coordinates of left top and right bottom corner are required to draw the rectangle. left specifies the X-coordinate of top left corner, top specifies the Y-coordinate of top left corner, right specifies the X-coordinate of right bottom corner, bottom specifies the Y-coordinate of right bottom corner.

Syntax:

```
rectangle(int left ,int top,int right, int bottom);
```

➤ **circle():**

Header file `graphic.h` contain the `circle` function. Which is used to draw the circle. This circle function consists of 3 parameters center coordinates of the circle(x,y) and the radius of the circle is to be given.

Syntax:

```
circle(x,y,radius of circle);
```

Here x,y are the centre coordinates of the circle.

➤ **ellipse():**

Header file `graphic.h` contain the `ellipse` function. Which is used to draw the ellipse. This ellipse function contains 6 parameters.

Syntax:

```
void ellipse(int x, int y, int stangle, int endangle,  
int xradius, int yradius);
```

This is used to draw an Ellipse at (x,y) are coordinates of center of the ellipse, stangle is the starting angle, end angle is the ending angle, and fifth and sixth parameters specifies the X and Y radius of the ellipse. To draw a complete ellipse stangles and end angle should be 0 and 360 respectively.

➤ **Line():**

Header file `graphic.h` contain the line function which is used to draw the line from a point (x1,y1) to a point (x2,y2) (i.e., (x1,y1) & (x2,y2) are the end points of line) . line function contains 4 parameters.

Syntax: `void line(int x1, int y1, int x2, int y2);`

Here (x1,y1) & (x2,y2) are the end points of the line.

➤ **fillellipse():**

Header file `graphic.h` contains `fillellipse` function which is used to fill the ellipse. `fillellipse` contains 4 parameters.

Syntax:

```
fillellipse(int x,int y,int x_radius,int y_radius);
```

➤ **setfillstyle():**

It is a function which sets the current fill pattern and fill color.

Syntax:

```
setfillstyle(int pattern , int color);
```

➤ **floodfill():**

This function is used to fill an enclosed area. Current fill pattern and fill color is used to fill the area.

Syntax:

```
floodfill(int x ,int y ,int border_color);
```

➤ **setcolor():**

It is a function which is used to set the current drawing color to the new color.

Syntax:

```
setcolor(int color);
```

➤ **getcolor():**

This function is used to return the current working color.it returns in the form of integer.

Syntax: `int var=getcolor();`

➤ **settextstyle():**

This is a function which is used to change the way in which text appears. Using it we can modify the size of text, change direction of text and change the font of text.

Syntax:

```
settextstyle(int font , int direction , int font_size);
```

➤ **outtextxy():**

This is a function which displays the text or string at a specified point (x, y) on the screen.

Syntax:

```
outtextxy(int x ,int y, char *string);
```

➤ **cleardevice():**

This is a function which clears the screen in graphics mode and sets the current position to (0,0). Clearing the screen consists of filling the screen with current background color.

Syntax:

```
cleardevice();
```

➤ **delay():**

This is a function which is used to delay the code execution.

Syntax:

```
delay(int time);
```

➤ **itoa():**

This function converts int data type to string data type.

Syntax:

```
itoa(int value ,char *string ,int base);
```

➤ **kbhit():** This is used to determine if a key has been pressed or not. If a key has been pressed then it returns a non zero value otherwise returns zero.**Syntax:**

```
kbhit();
```

➤ **exit():**

This function is used to exit from the program.its requires header file <stdlib>.

Syntax:

```
exit(0);
```

➤ **Beep:**

Beep is predefined function which is used to produce a sound. The beep function consists of 2 parameters.

Syntax:

```
Beep(int fre,int duration);
```

The frequency of the sound, in hertz. This parameter must be in the range 37 through 32,767.

Duration must be in range of milliseconds.

➤ **Setbkcolor():**

This function used to change the background color of the screen.function takes only one argument it would be either the name of color defined in **graphics.h** header file or number associated with those colors.

Syntax: setbkcolor(int color);

➤ **Class:**

Class is a user defined data type based on which objects are created. Properties or characteristics of similar objects and methods are grouped together to form a Class. Thus, a class is a collection of objects. Characteristics of an object are represented in a class as data members. The actions that can be performed by objects are referred as Methods. No memory is allocated when a class is created.

It allows the data (and functions) to be hidden, if necessary, from external use. Generally a class specification has two parts:

- i. Class declaration
- ii. Class function definitions

Class declaration describes the type and scope of its members. The class function definitions describe how the class functions are implemented. The key word class specifies, that what follows is an abstract data of type class_name. The body of a class is enclosed within braces and terminated by a semicolon. The class body contains the declaration of variables and functions. These functions and variables are collectively called class members. They are usually grouped under two sections, namely, private and public to denote which of the members are private and which of them are public. The keywords private and public are known as visibility labels. Not that these keywords are followed by a colon. If both the labels are missing, then, by default, all the members are private.

Syntax:

```
class classname
{
<visibility mode>:
...
};
```


➤ **Object:**

It is a real world entity. Objects are the basic run-time entities in an object-oriented program. They may represent a specific person, a specific place, a specific bank account or any specific item that the program has to handle. Object is the basic unit of object-oriented programming and is identified by its unique name.

Syntax:

```
classname<object_name>;
```

(For creating object for class <classname>)

```
object_name.var    or    object_name.fun()
```

(For accessing variable <var> or For calling member function fun() using non pointer object)

Source code

```

#include<iostream>
#include<conio.h>
#include<graphics.h>
#include<stdlib.h>
#include<dos.h>
#include<math.h>
#include<time.h>
using namespace std;
void snake_int();
void snake_move();
int check_egg();
void fruit();
int m = 6, life=3;
int scr=0, exi=0;
char pre=0;
char p[20], lf[20];
void select();
void instructions()
{
    setcolor(BLACK);
    rectangle(getmaxx()-270, getmaxy()-
490, getmaxx(), getmaxy());

    setcolor(10);
    settextstyle(2, 0, 8);
    outtextxy(getmaxx()-250, getmaxy()-400, " CONTROLS ");
    settextstyle(3, 0, 3);
    outtextxy(getmaxx()-250, getmaxy()-350, "p = PAUSE");
    outtextxy(getmaxx()-250, getmaxy()-300, "Esc = EXIT");
    outtextxy(getmaxx()-250, getmaxy()-250, "ARROWS");
    outtextxy(getmaxx()-252, getmaxy()-220, "      -MOVEMENT");
}
void print();
void sound_g()
{
    Beep(600, 50);
    delay(2);
}

void quit()
{
    exe:
    char eti[10];

```

```

char choice;
    cleardevice();
    setbkcolor(BLACK);
    cleardevice();
    setbkcolor(BLACK);
setcolor(RED);
settextstyle(8,0,4);
ltoa(scr,eti,10);
    rectangle(10,10,getmaxx(),getmaxy());
    outtextxy(50,250,"SCORE:");
    outtextxy(210,250,eti);
    outtextxy(100,300,"....GAME OVER...");
    outtextxy(getmaxx()-550,360,"DO U WANT PLAY AGAIN !!");
    outtextxy(getmaxx()-550,400,"PRESS (y/Y) TO PLAY
AGAIN");
    outtextxy(getmaxx()-550,450,"PRESS (N/n) TO EXIT THE
GAME");
choice=getch();

    if(choice=='Y' || choice=='y')
    {
        cleardevice();
        print();
    }
    else if(choice=='N' || choice=='n')
    exit(0);
    else
    {
        cleardevice();
settextstyle(8,0,4);
        outtextxy(getmaxx()-550,360,"INVALID CHOICE!!");
        delay(100);
        goto exe;
    }

}

void score()
{
if(m<0 || life<=0)
{
quit();
}
setcolor(BLACK);

```

```

rectangle(getmaxx()-270,0,getmaxx(),getmaxy()-500);
    ltoa(scr,p,10);
    settextstyle(8,0,2);
setcolor(4);
    outtextxy(getmaxx()-100,70,"SCORE: ");
    outtextxy(getmaxx()-35,70,p);
    ltoa(life,lf,10);
    outtextxy(getmaxx()-100,120,"LIFE:");
    outtextxy(getmaxx()-40,120,lf);
    setcolor(BLACK);
}
int snk_body[50][3];

int snk_wid = 10;
int snk_posx=150;
int snk_posy=150;
int snk_dir=1;
int egen=0,bgen=0;
int ex,ey,bx,by,hor=0,ver=0;
class eagle
{
public:
    int x;
    int y;
    eagle()
    {
        x=100;
        y=100;
    }
    void move()
    {
        setcolor(4);
        y=y+5;
        if(y>=getmaxy())
        {
            x=rand()%(getmaxx()-350);
            y=rand()%(200);
        }
        setcolor(RED);
        line(x,y,x-5,y-5);
        line(x,y,x+5,y-5);
        ellipse(x,y-22,0,360,9,15);
        ellipse(x-5,y-7,0,360,2,2);
        fillellipse(x-5,y-7,2,2);
    }
}

```

```

    ellipse(x+5,y-7,0,360,2,2);
    fillellipse(x+5,y-7,2,2);
    line(x-2,y-37,x-5,y-47);
    line(x+2,y-37,x+5,y-47);
    ellipse(x,y-47,0,180,7,4);
    line(x-15,y-22,x-10,y-18); //hands
    line(x+15,y-22,x+10,y-18); //hands
    setcolor(BLACK);
    ellipse(x+13,y-22,0,180,15,3);
    ellipse(x-13,y-22,0,180,15,3);
    setfillstyle(1,0);
    fillellipse(x-13,y-22,15,3);
    fillellipse(x+13,y-22,15,3);
    setfillstyle(1,4);
    fillellipse(x,y-22,9,15);
setcolor(RED);
    check();
}

void check()
{
    int j=0;

    int      r=sqrt(((x-snk_body[j][1])*(x-snk_body[j][1]))+((y-
snk_body[j][2])*(y-snk_body[j][2])));
    int p=sqrt(((x-snk_body[j][1])*(x-snk_body[j][1]))+((y-2-
snk_body[j][2])*(y-2-snk_body[j][2])));
    if(r<=10||p<=10)
    {
        sound_g();
        m=m-2;
        delay(100);
        life=life-1;
    }

}

};

void check_boundaries()
{int clr;

    if(snk_body[0][1]<0)//when snake goes left
    {
        snk_body[0][1]=(getmaxx()-300)-20;
    }
}

```

```

if(snk_body[0][1]+20>getmaxx()-300)//when snake goes right
{
    snk_body[0][1]=0;
}
if(snk_body[0][2]>getmaxy())//when snake goes down
{
    snk_body[0][2]=0;
}
if(snk_body[0][2]<0)//when snake goes up
{
    snk_body[0][2]=(getmaxy())-20;
}
for(int i=1;i<m;i++)
{
    if(snk_body[0][1] == snk_body[i][1] && snk_body[0][2]
== snk_body[i][2])
    {
        exi=exi+1;
        if(exi>2)
        {
            quit();
        }
    }
}

if(snk_dir == 'p')
{snk_dir='r';
settextstyle(11,0,9);char c;int qw=0;clr = getcolor();
while(!kbhit())
{
    setcolor(10); settextstyle(8,0,5);
    rectangle(50,150,getmaxx()-400,getmaxy()-200);
    outtextxy(100,300,"PAUSED-- PRESS ANY KEY TO CONTINUE");
}
setcolor(clr);

}
}
void disappear(eagle b[4])
{for(int i=0;i<4;i++)
{
int dis=sqrt((ex-b[i].x)*(ex-b[i].x)+(ey-b[i].y)*(ey-
b[i].y));
int dis2=sqrt((bx-b[i].x)*(bx-b[i].x)+(by-b[i].y)*(by-
b[i].y));

```

```

if(dis<=8)
{
    egen=0;
    Beep(600,50);
}
if(dis2<=11)
{bgen=0;
Beep(600,50);
}

}
}
void bonus()
{setfillstyle(1,1);
cout<<bgen;
    setcolor(BLUE);
    if(bgen==0)
    {
        bx=rand()%(getmaxx()-300-50)+50;
        by=rand()%(getmaxy()-50)+50;
    }
    while(ex!=bx)
    {
        bx=rand()%(getmaxx()-300-50)+50;
    }
    ellipse(bx,by,0,360,10,10);
    fillellipse(bx,by,10,10);
    bgen=1;
}
int check_bonus()
{
    int bo=sqrt(((bx-snk_body[0][1])*(bx-
snk_body[0][1]))+((by-snk_body[0][2])*(by-
snk_body[0][2])));
    if(bo<=21)
    {
        bgen=0;
        scr=scr+100;
        if(life<3)
            life=life+1;

        return 1;
    }
    else

```

```

        return 0;
    }
    void fruit()
    {
        if(egen==0)
        {
            ex=rand()%(getmaxx()-300-50)+50;
            ey=rand()%(getmaxy()-50)+50;
        }
        setcolor(BLACK);
        ellipse(ex,ey,0,360,5,5);
        setfillstyle(1,10);
        fillellipse(ex,ey,5,5);
        egen=1;
    }
    int check_egg()
    {int    n=sqrt(((ex-snk_body[0][1])*(ex-snk_body[0][1]))+((ey-
snk_body[0][2])*(ey-snk_body[0][2])));

        if(n<=16)
        {
            egen=0;
            m++;
            scr=scr+10;
            return 1;
        }
        else
        return 0;
    }

    void snake_init()
    {
        int i=0;
        snk_body[0][1]=snk_posx;
        snk_body[0][2]=snk_posy;
        setcolor(BLACK);
        rectangle(0,0,getmaxx()-300,getmaxy());
        for(i=1;i<=m;i++)
        {
            snk_body[i][1]=snk_body[i-1][1]+snk_wid*2;
            snk_body[i][2]=snk_body[i-1][2];
        }
    }

```



```

    }
}

```

```

void snake_move()
{
    int i=0;
    setcolor(RED);
    for(i=m;i>=1;i--)
    {
        snk_body[i][1]=snk_body[i-1][1];
        snk_body[i][2]=snk_body[i-1][2];
    }

    switch(snk_dir)
    {
        case 75:
            if(pre!=77)
            {
                hor=-20;
                ver=0;
                pre=75;
            }
            break;
        case 77:
            if(pre!=75)
            {
                hor=20;pre=77;
                ver=0;
            }
            break;
        case 80:
            if(pre!=72)
            {
                hor=0;
                ver=14;pre=80;
            }
            break;
        case 72:
            if(pre!=80)
            {
                hor=0;
                ver=-14;pre=72;
            }
    }
}

```

```

        break;
    case 1:
        hor=20;
        ver=0;
        break;
    }
    snk_body[0][1]=snk_body[0][1]+hor;
    snk_body[0][2]=snk_body[0][2]+ver;
    if(check_egg()==1)
    {
        snk_body[m][1]=snk_body[m-1][1]-hor;
        snk_body[m][2]=snk_body[m-1][2]-ver;
    }

    setcolor(BLACK);
    rectangle(0,0,getmaxx()-300,getmaxy());

setcolor(BLACK);
    for(i=0;i<=m;i++)
    {

        ellipse(snk_body[i][1],snk_body[i][2],0,360,snk_wid,7)
;
    }

}
void select()
{
    m = 6;
    life=3;
    scr=0;
    int exi=0;
    pre=0;
    int key_b =0,b=51;
    int ter=0;
    snake_init();
    eagle a1[4];
    int ch;
    while (snk_dir != 27)
{setbkcolor(WHITE);
    ter++;
    while (kbhit())

    {

```

```

        snk_dir=getch();
    }

    score();
    delay(100);
    cleardevice();
    snake_move();
instructions();
    check_boundaries();
    fruit();
    disappear(a1);
    a1[0].move();
    a1[1].move();
    a1[2].move();
    a1[3].move();
    if(ter%200==0)
    {
        b=0;
    }
    if(b<=50)
    {
        bonus();
        b++;

        if(check_bonus()==1)
            b=51;
    }
}

}

void print()
{setbkcolor(BLACK);
    char ch;
    dev:
    cleardevice();
    settextstyle(3,0,2);
    outtextxy(300,100,"..WELCOME TO SNAKE GAME..");
    setcolor(10);
    settextstyle(4,0,8);
    outtextxy(90,200,"SNAKE");
    setcolor(14);
    outtextxy(500,300,"VS");
    setcolor(RED);
    outtextxy(650,400,"EAGLE");
}

```

```
settextstyle(3,0,2);
setcolor(13);
    outtextxy(750,600,"PRESS ENTER TO START THE GAME..");
ch=getch();
if(ch==13)
{
    cleardevice();
    select();
}

else
{cleardevice();
    setcolor(RED);
    outtextxy(getmaxx()-200,getmaxy()-200,"INVALID OPTION");
    delay(100);
    goto dev;
}
}

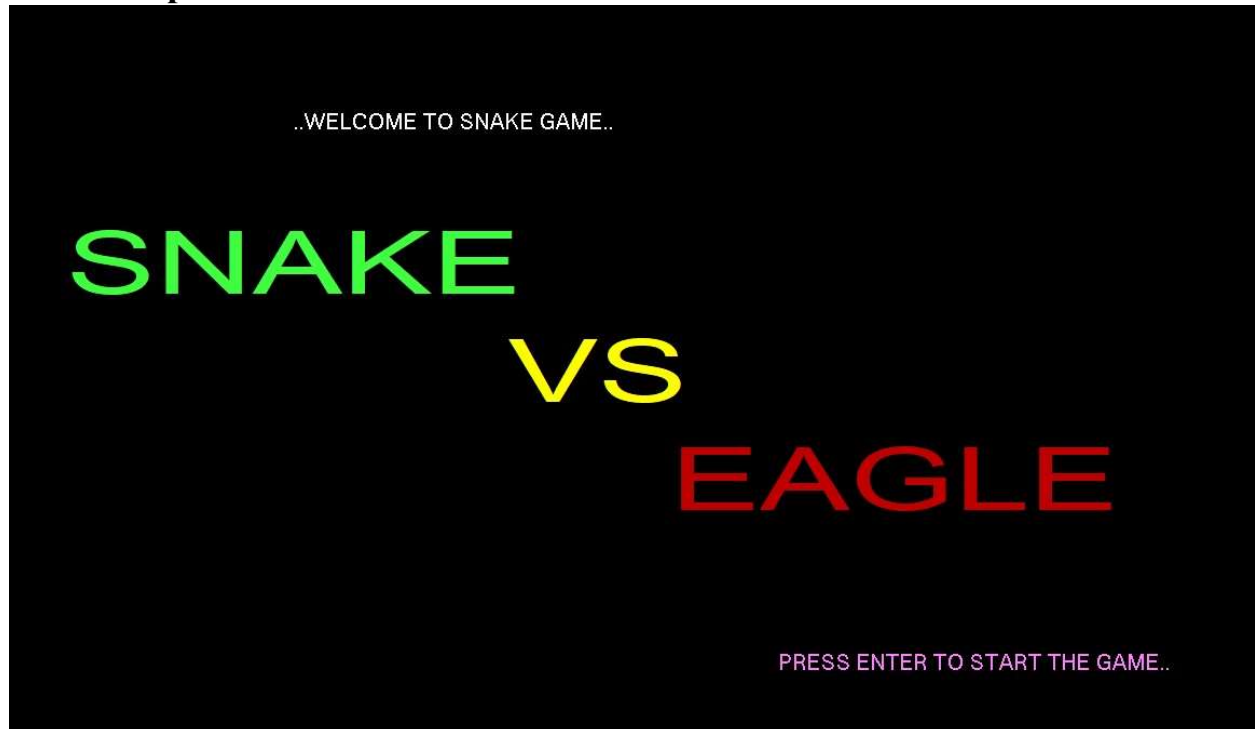
int main()
{
    int gd=DETECT,gm;
    initgraph(&gd,&gm,"C:\\TURBOC3\\bgi");
    print();

    return 0;

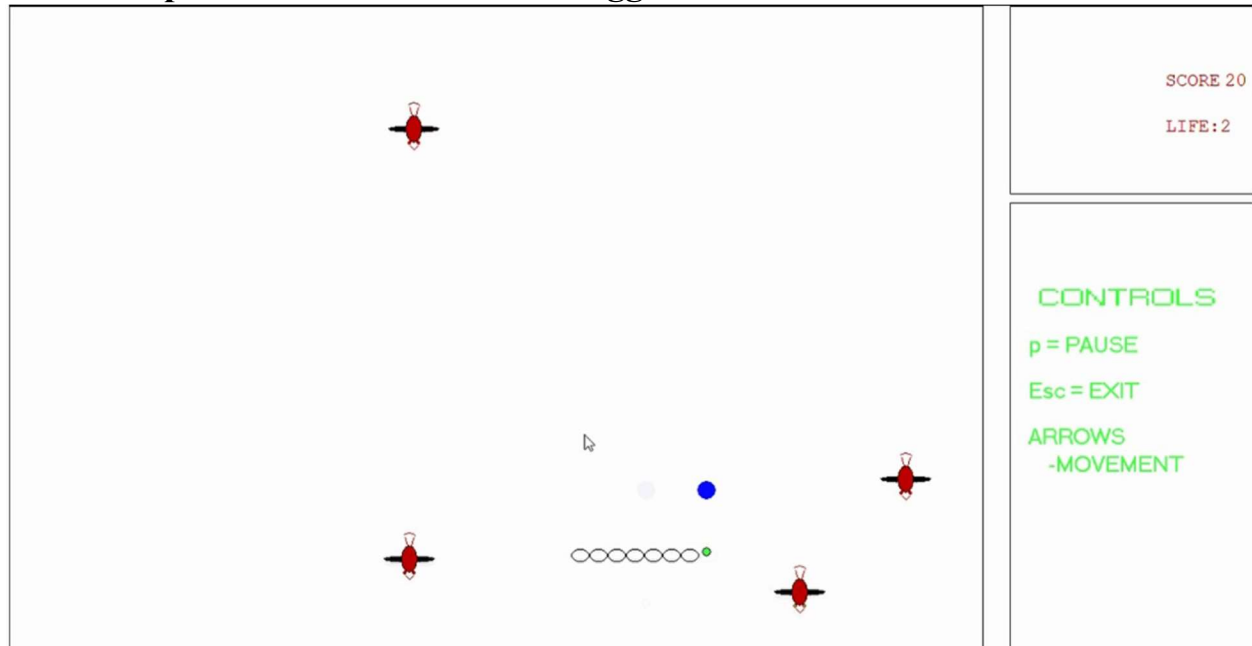
}
```

OUTPUT SCREENSHOTS

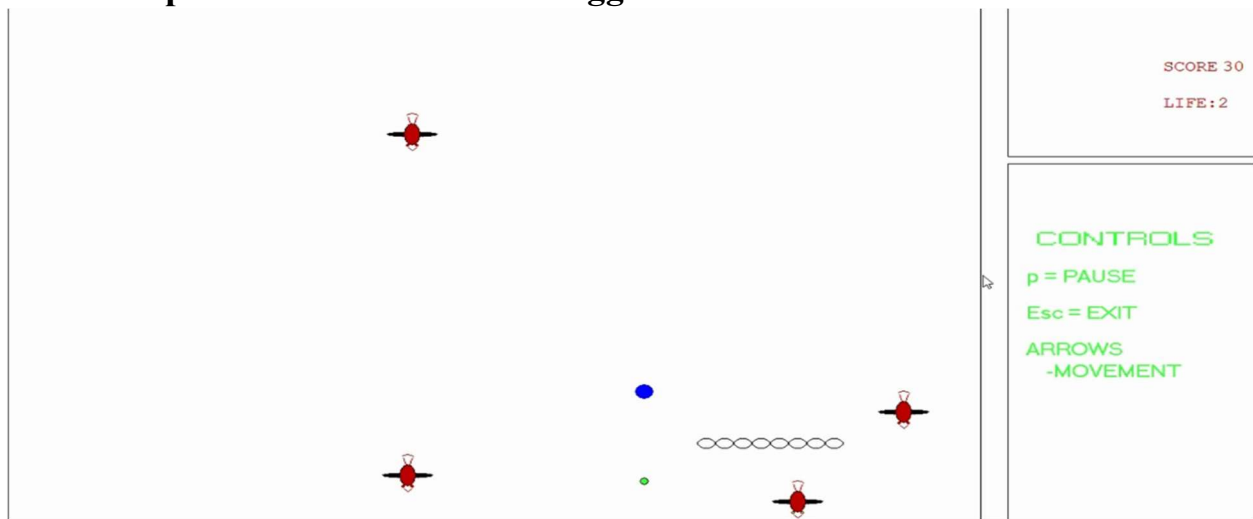
Output 1:



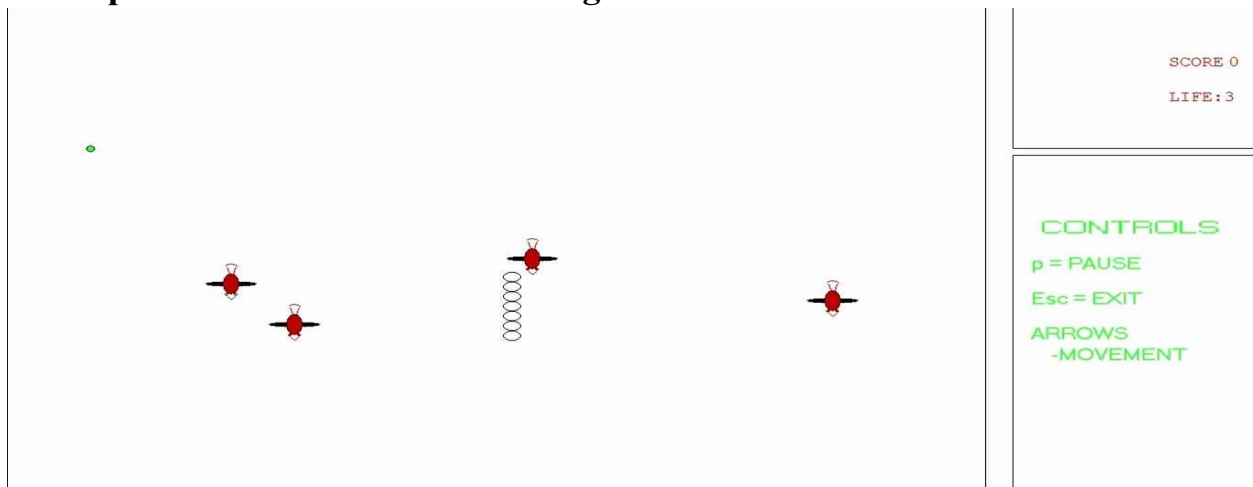
Output 2: Before snake eat the egg.



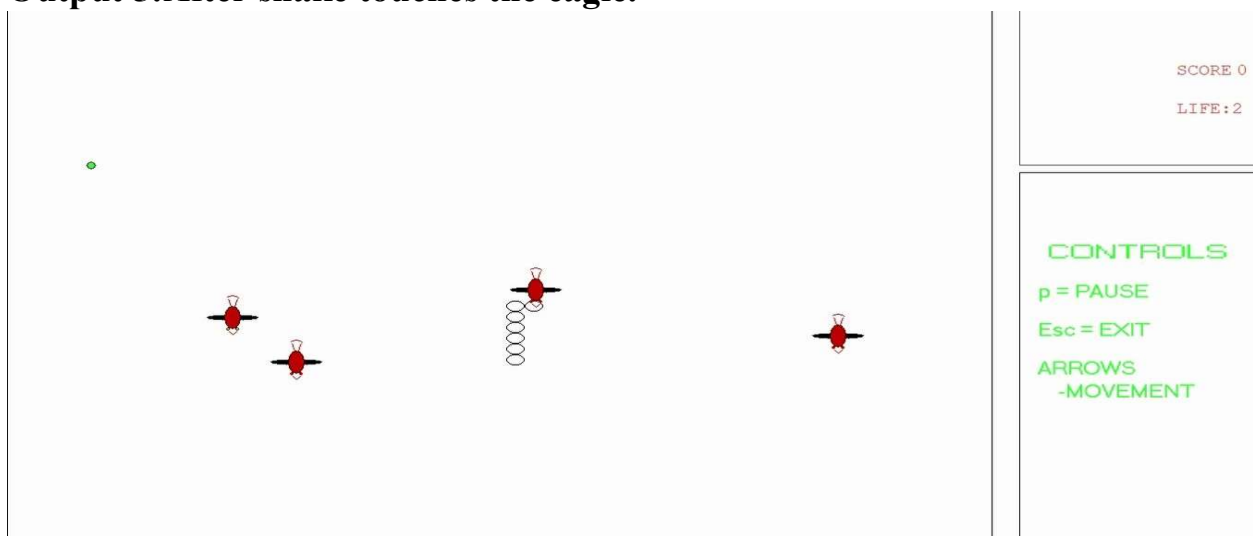
Output 3:After snake ate the egg:



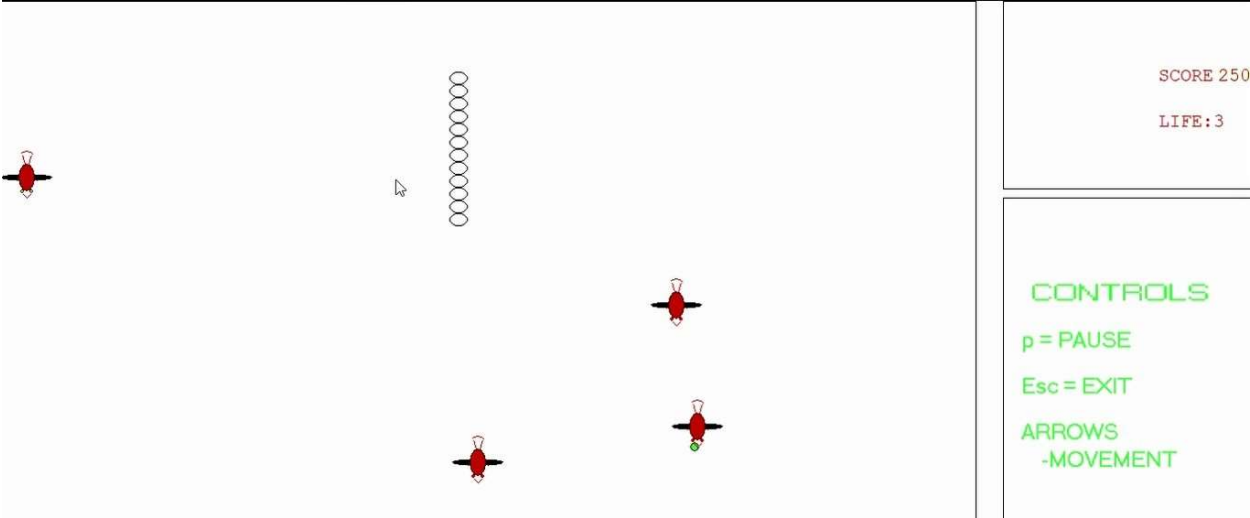
Output 4:Before snake touches eagle:



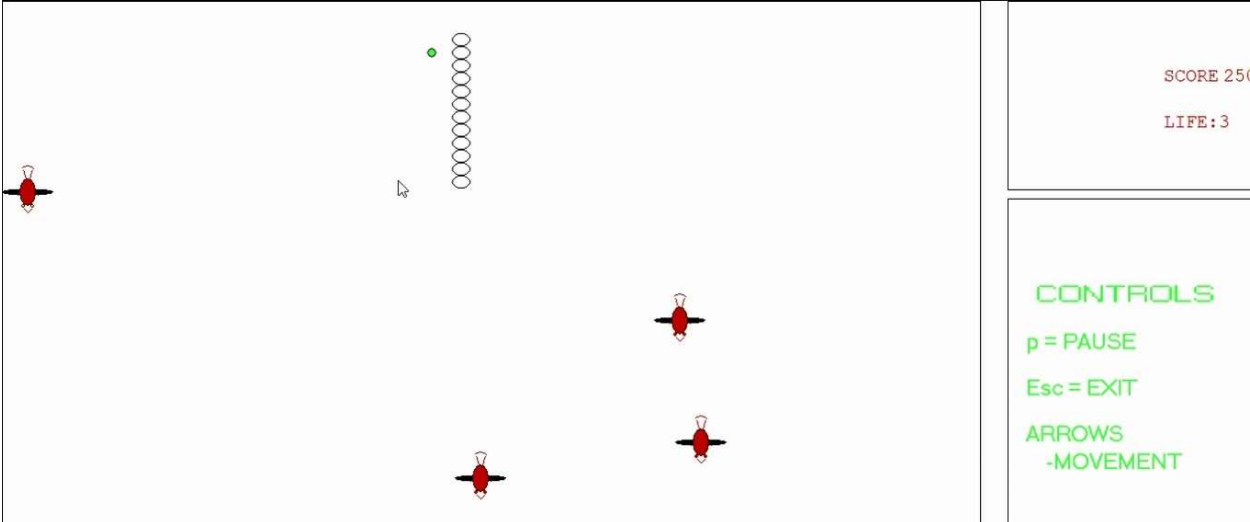
Output 5:After snake touches the eagle.



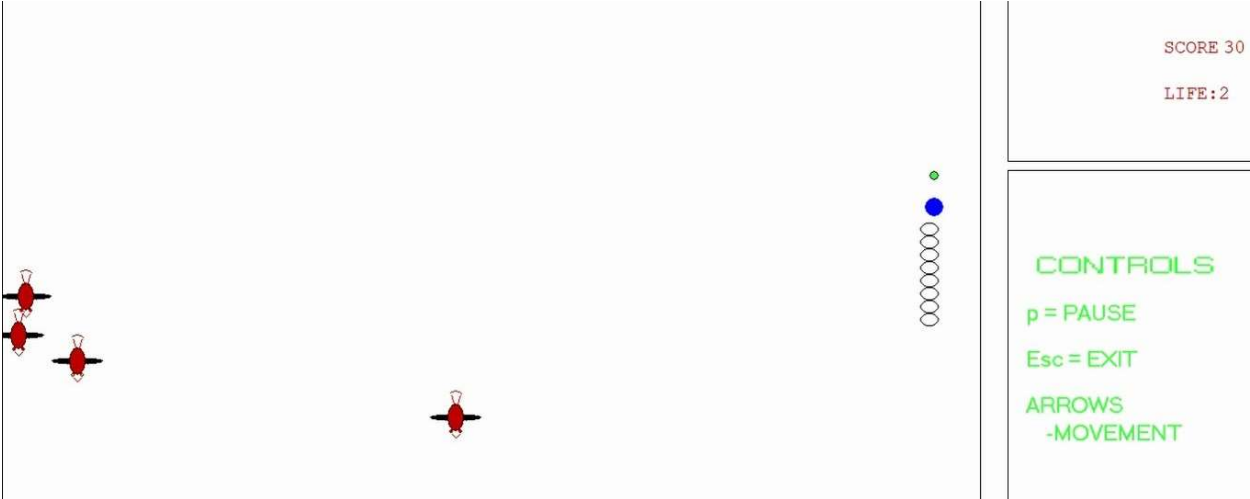
Output 6-when eagle about to touch the egg.



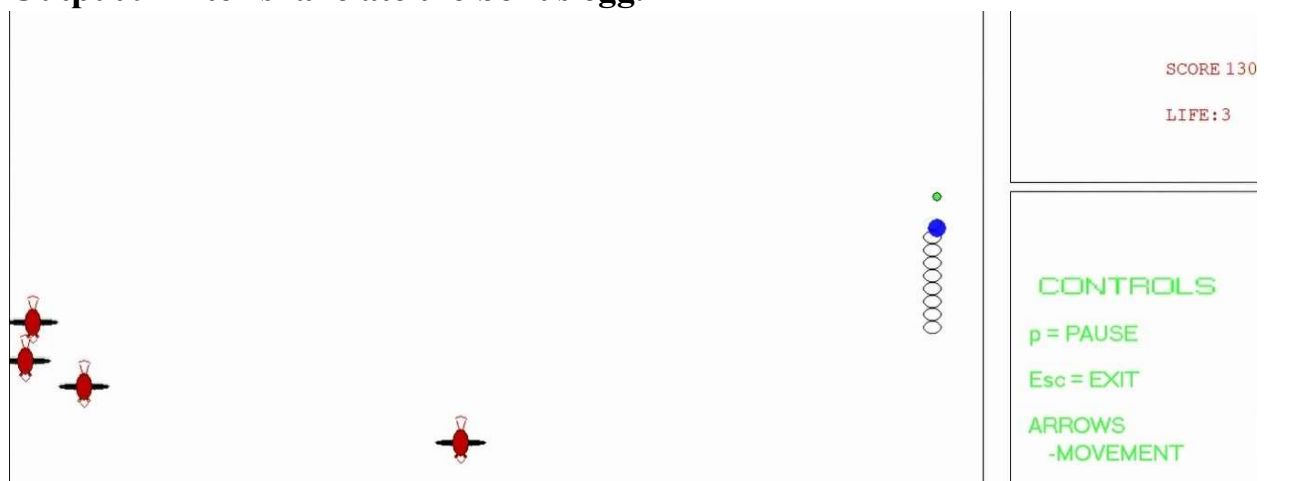
Output 7-when eagle ate the egg,egg will disappear.



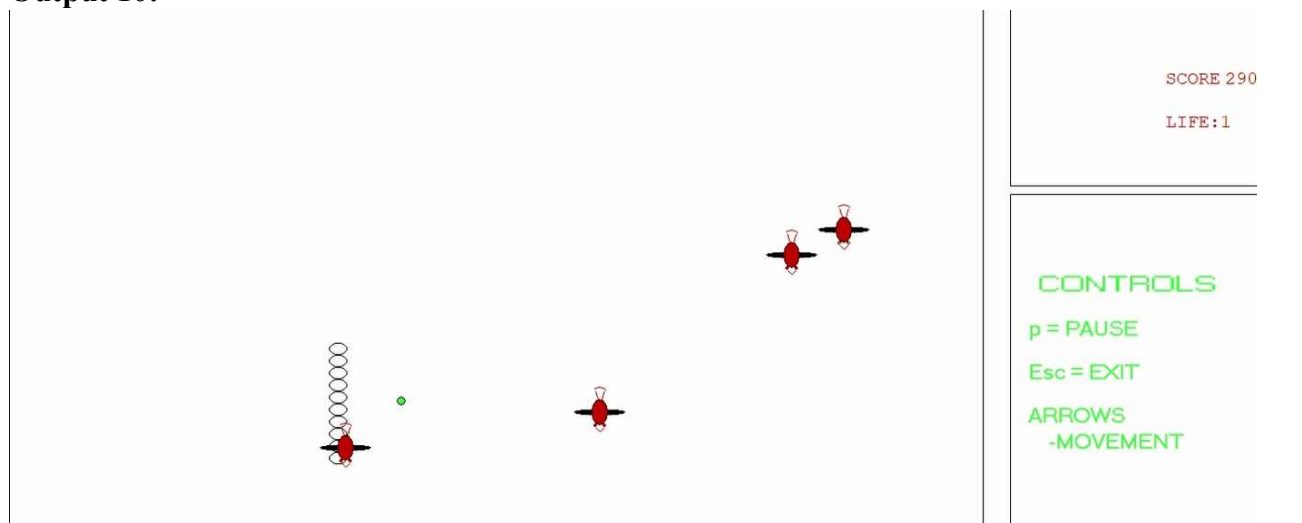
Output 8-Before snake eats the bonus egg.



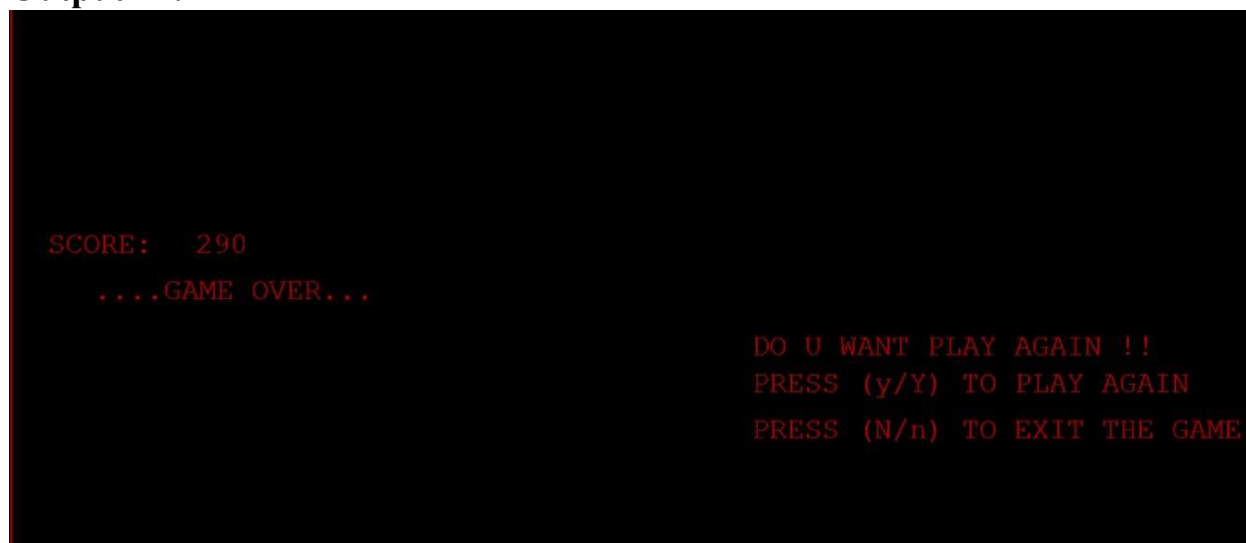
Output 9-After snake ate the bonus egg.



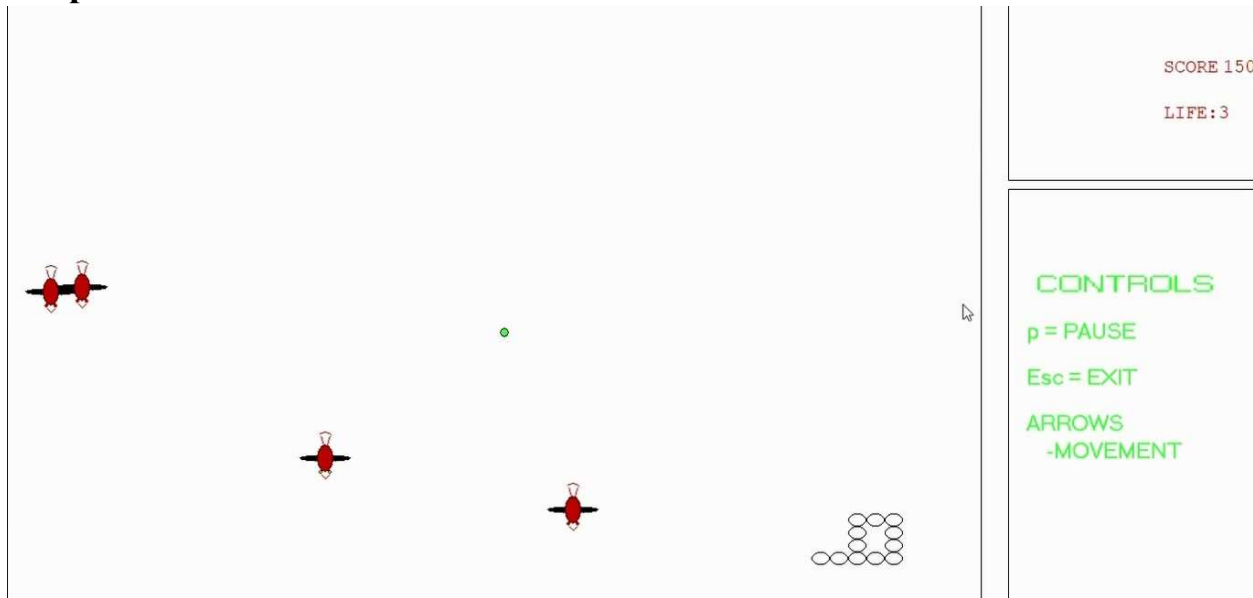
Output 10:



Output 11:



Output 12-when snake touches it self:-



Output 13-Game over:-

