

INSTITUTE OF ENGINEERING
ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT
KALANKI, KATHMANDU
(AFFILIATED TO TRIBHUVAN UNIVERSITY)



ADVANCED COLLEGE
OF ENGINEERING & MANAGEMENT

PROJECT ON:
COMPUTER GRAPHICS
(SNAKE GAME)

SUBMITTED BY :

NAME : PRASUNA SAPKOTA

ROLL NO : ACE077BCT058

DATE : AUGUST 13, 2023

SUBMITTED TO:

DEPARTMENT OF COMPUTER
AND ELECTRONICS

SOURCE CODE

```
#include<conio.h>
```

```

#include<graphics.h>
#include<dos.h>
#include<stdlib.h>
#include<stdio.h>
#include<time.h>
#include<string.h>
class Snake
{
int p1,p2,v1,v2,v3,e1,e2,prev,now,n,colr,dsp,cnt,dly,m;
int stp, egGen;
int xr, yr;
void caught();
public:
    long scr;
int strtX,strtY,endX,endY;
int pos[100][2];
void show();
void init();
void egg();
void transpose();
void gnrtCond();
void gnrtUnCond();
void check();
void checkEgg();
void move();
void chngDir();
void sndEt();
void sndCgt();
int test();
void score();
Snake();
Snake(Snake*);
~Snake();
};
Snake::Snake()
{
}

```

```

Snake::~~Snake()
{
}
void Snake::checkEgg()
{
if((e1 == p1) && (e2 == p2))
{ sndEt();
  egg();
  dly--;
  score();
  n++;
}
}
void Snake::sndEt()
{ nosound();
  sound(2500);
  delay(2);
  nosound();
}
void Snake::sndCgt()
{ nosound();
  for(int x=1000;x>0;x--)
  { sound(x);
    delay(1);
  }
  nosound();
}
void Snake::score()
{ char *p;
  ltoa(scr,p,10);
  settextstyle(8,0,1);
  setcolor(0);
  outtextxy(585,40,p);
  if(egGen != 1){
    scr = scr + dly / 10;
  }
  ltoa(scr,p,10);
}

```

```

    setcolor(10);
    outtextxy(585,40,p);
}
void Snake::gnrtCond()
{ if(n < 367)
{ if(now == 8 && (prev != 8 && prev != 2))
{ pos[0][0] = p1;
  pos[0][1] = p2 - dsp;
  prev = now;
}
if(now == 4 && (prev != 4 && prev != 1))
{ pos[0][0] = p1 + dsp;
  pos[0][1] = p2;
  prev = now;
}
if(now == 2 && (prev != 8 && prev != 2))
{ pos[0][0] = p1;
  pos[0][1] = p2 + dsp;
  prev = now;
}
if(now == 1 && (prev != 1 && prev != 4))
{pos[0][0] = p1 - dsp;
 pos[0][1] = p2;
 prev = now;
}
}
}
void Snake::gnrtUnCond()
{
  if( prev == 8 )
  { pos[0][0] = p1;
    pos[0][1] = p2 - dsp;
  }
  if( prev == 4 )
  {pos[0][0] = p1 + dsp;
   pos[0][1] = p2;
  }
}

```

```

if( prev == 2 )
{
    pos[0][0] = p1;
    pos[0][1] = p2 + dsp;
}
if( prev == 1 )
{
    pos[0][0] = p1 - dsp;
    pos[0][1] = p2;
}
p1 = pos[0][0];
p2 = pos[0][1];
}
void Snake::check()
{
    if(p1 > endX)
    {
        p1 = strtX;
    }
    else if(p1 < strtX)
    {
        p1 = endX;
    }
    if(p2 > endY)
    {
        p2 = strtY;
    }
    else if(p2 < strtY)
    {
        p2 = endY;
    }
    pos[0][0] = p1;
    pos[0][1] = p2;
    for(int i = 1; i < n; i++)
    {
        if(p1 == pos[i][0] && p2 == pos[i][1])
        {
            caught();
            break;
        }
    }
}
void Snake::show()
{
    int x = getcolor();
    if(egGen != 1)
    {
        setcolor(getbkcolor());
        setfillstyle(1, getbkcolor());
    }
}

```

```

    fillellipse(v1,v2,yr,yr);
}
else
    egGen = 0;
if(egGen == 2)
    egGen--;
setcolor(colr);
setfillstyle(1,9);
if(now == 8 || now == 2)
    fillellipse(pos[0][0],pos[0][1],xr,yr);
else if(now == 4 || now == 1)
    fillellipse(pos[0][0],pos[0][1],yr,xr);
setcolor(x);
}
void Snake::transpose()
{ int i,j,x,y;
  p1 = pos[0][0];
  p2 = pos[0][1];
  if(!egGen){
    v1 = pos[n-1][0];
    v2 = pos[n-1][1];
  }
  else
    egGen = 0;
  for(i = n-1;i >= 1;i--)
    {pos[i][0] = pos[i-1][0];
     pos[i][1] = pos[i-1][1];
    }
}
void Snake::move()
{ int st = 0;
  do{
    if(!kbhit())
    { checkEgg();
      if(!st)
        show();
      else

```

```

    st = 0;
    delay(dly/4);
    transpose();
    delay(dly/4);
    gnrtUnCond();
    delay(dly/4);
    check();
    delay(dly/4);
    }
    else if(stp){
    chngDir();
    gnrtCond();
    check();
    show();
    st = 1;
    }
    } while(stp);
}
void Snake::init()
{time_t tm;
srand(time(&tm));
dsp = 20;
n = 5;
prev = 4;
for(int i = 4;i >= 0;i--)
{ pos[i][0] = 201 + (n - i - 1) * dsp;
  pos[i][1] = 301;
}
strtX = 21;
strtY = 21;
endX = 481;
endY = 361;
colr = 14;
now = prev;
dsp = 20;
stp = 111;
cnt = -1;

```

```

scr = 0;
dly = 150;
xr = 3;
yr = 9;
egg();
egGen = 1;
score();
int x = getcolor();
setlinestyle(0,1,3);
setcolor(15);
rectangle(strtX-15,strtY-15,endX+15,endY+15);
rectangle(endX+25,strtY-15,getmaxx()-15,endY+15);
rectangle(strtX-15,endY+25,getmaxx()-15,getmaxy()-5);
line(endX+25,strtY+75,getmaxx()-15,strtY+75);
line(endX+25,strtY+200,getmaxx()-15,strtY+200);
line(endX+25,strtY+275,getmaxx()-15,strtY+275);
setlinestyle(0,1,1);
settextstyle(8,0,1);
setcolor(11);
outtextxy(514,40,"SCORE");
setcolor(14);
settextstyle(11,0,5);
outtextxy(524,110," CONTROLS ");
outtextxy(522,135,"p = PAUSE");
outtextxy(522,155,"g = RESUME");
outtextxy(522,175,"e = EXIT");
outtextxy(513,195,"ARROWS");
outtextxy(512,205," -MOVEMENT");
setcolor(14);
settextstyle(4,0,9);
outtextxy(getmaxx()-500,getmaxy()-110,"SNAKE");
settextstyle(8,0,1);
setcolor(x);
}
void Snake::caught()
{
stp = 0;

```



```

sndCgt();
for(int i=0;i<=7;i++)
{ if(i%2)
  { setcolor(10);
    outtextxy(512,250,"GAME OVER");
    delay(900);
  }
  else
  {setcolor(0);
    outtextxy(512,250,"GAME OVER");
    delay(500);
  }
}
sleep(1);
}
void Snake::chngDir()
{ int clr;
  fillsettingstype *p;
  char x = getch();
  if(x == 72)
    now = 8;
  else if(x == 77)
    now = 4;
  else if(x == 80)
    now = 2;
  else if(x == 75)
    now = 1;
  else if(x == 'e')
    caught();
  else if(x == 'p')
  { //int y = getcolor();
    int twinkl = 1;
    setttextstyle(11,0,9);
    while(1)
    {if(kbhit())
      { int c = getch();
        if(c == 'g')

```

```

{ clr = getcolor();
  setcolor(0);
  rectangle(endX+40,endY-10,getmaxx()-35,getmaxy()-160);
  outtextxy(endX+60,endY-29,"PAUSE");
  break;
}
}
else
{ if(twnkl%2)
{ clr = getcolor();
  setcolor(10);
  rectangle(endX+40,endY-10,getmaxx()-35,getmaxy()-160);
  outtextxy(endX+60,endY-29,"PAUSE");
  setcolor(clr);
  delay(1000);
}
else
{
  clr = getcolor();
  setcolor(0);
  rectangle(endX+40,endY-10,getmaxx()-35,getmaxy()-160);
  outtextxy(endX+60,endY-29,"PAUSE");
  delay(1000);
}
}
twnkl++;
}
  settextstyle(8,0,1);
}
}
Snake::Snake(Snake *p)
{
  *p=NULL;
}
void Snake::egg()
{ do
  { e1 = (rand() % 100) * dsp + strtX;

```

```

    e2 = (rand() % 100) * dsp + strtY;
    } while(test());
    int x = getcolor();
    setcolor(7);
    setfillstyle(1,random(15)+1);
    fillellipse(e1,e2,xr+2,xr+2);
    setcolor(x);
    egGen = 2;
}
int Snake::test()
{
    for(int i=0;i<n;i++)
    { if(e1 == pos[i][0] && e2 == pos[i][1])
        break;
      if(v1 == e1 && v2 == e2)
        break;
      if((e1 >= endX+1) || (e2 >= endY+1))
        break;
    }
    if(i != n)
        return 1;
    else
        return 0;
}
void main()
{
    Snake snk;
    int gd=DETECT,gm,i,j,k,x,y;
    clrscr();
    initgraph(&gd,&gm,"C:\\Turboc3\\bgi");
    snk.init();
    snk.move();
    closegraph();
    restorecrtmode(); }

```

OUTPUT:

