Department of computer science Engineering

Minor Project

on

“Mobile Phone Snake Game”

Basic steps and brief explanation about snake game:

Steps to be followed for snake game:

Step1: Learning some basic syntax for developing snake form or shape or structure.

We need to learn basic syntax like:

1. #include<iostream>

2.#include <stdlib.h> // used for rand()

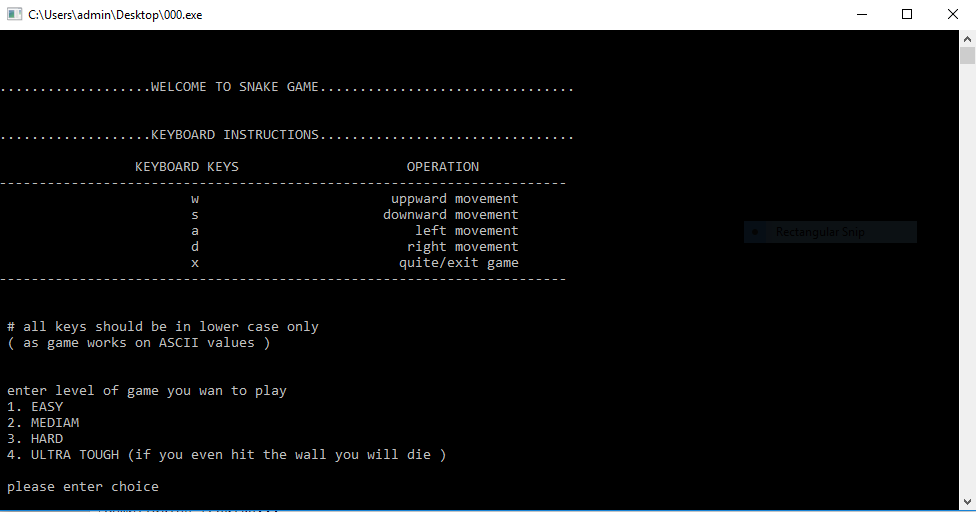
3.#include<conio.h> // used for getch() and kbhit()

4.#include<windows.h> // used for sleep()

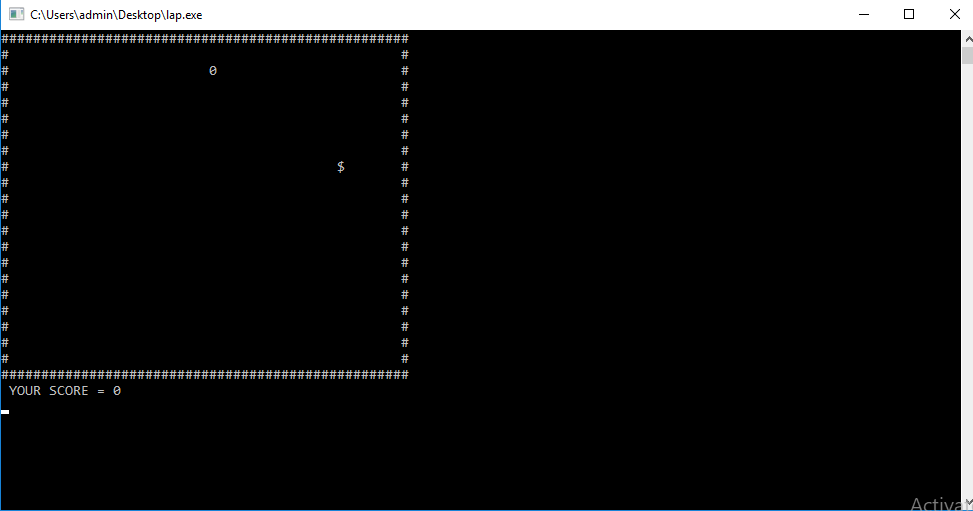
5.#include<iomanip>

Step2: Welcome page of the game:

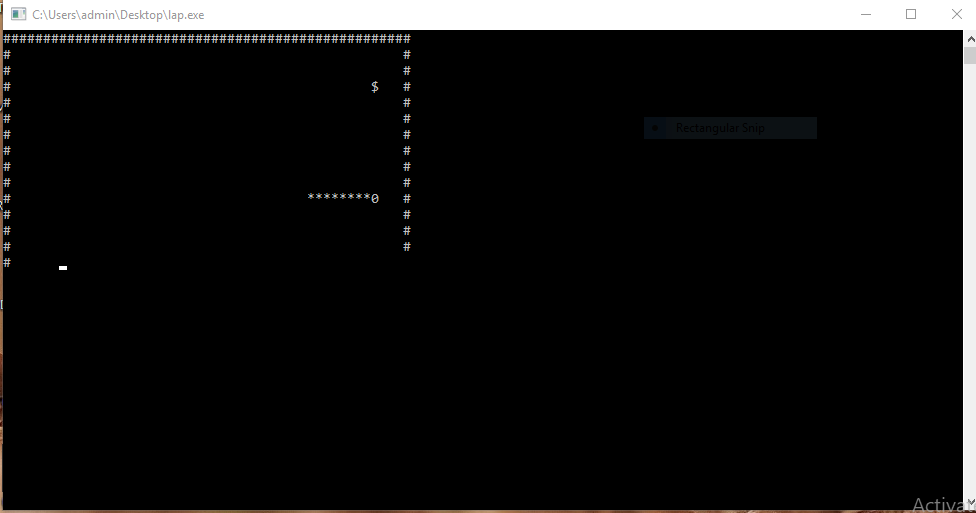
Welcome the user by showing the game title



Step3:Display format



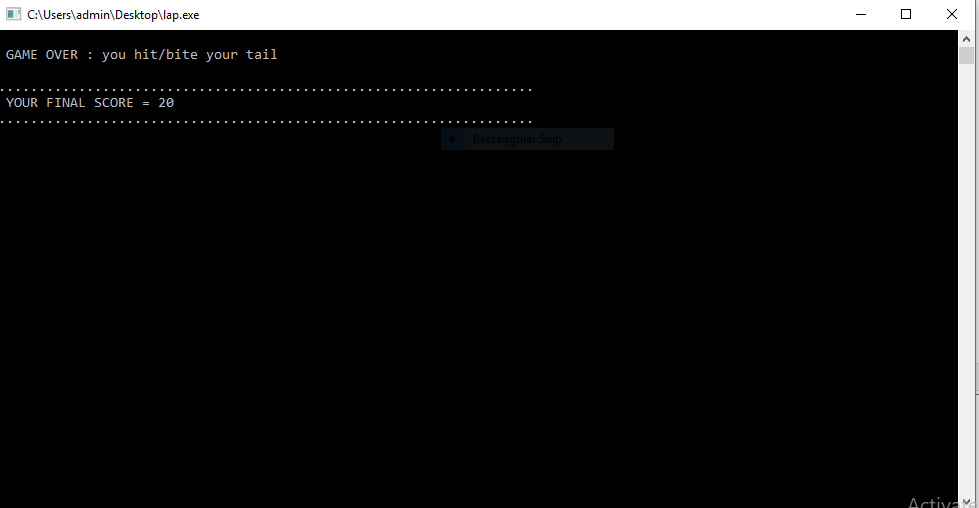
Step4: tail of the snake must be increased:



Step5:Game over condition or format:

When snake touch the body each other then it must be displayed that GAME OVER.

And displays the score.



Step6:changing directions:

The direction of snake can be changed using khbit() function when you press the character accordingly it will change the direction of the snake.

Step7:Algorithm for increasing size of snake and Increasing score:

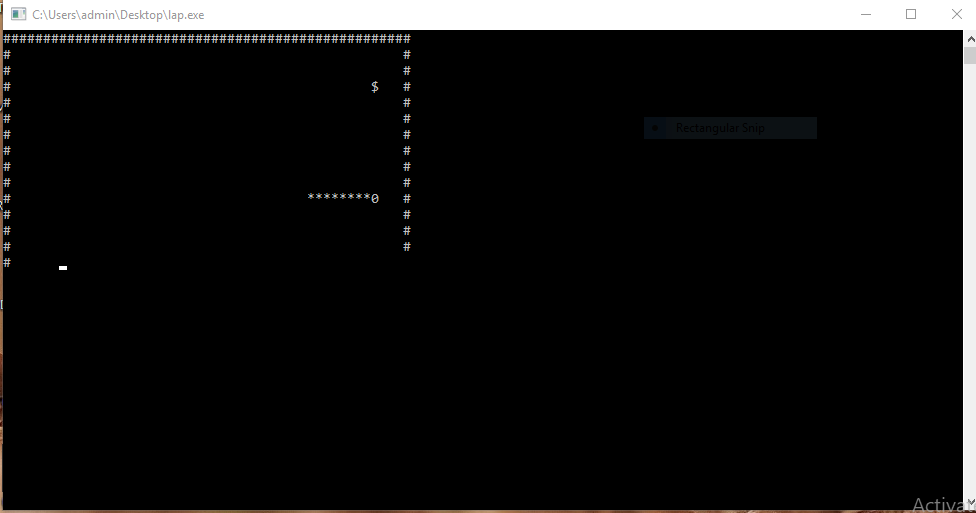
Increasing size of snake:

When snake moves forward its size increases and when the coordinate of snake match with food coordinate then the size of snake is increases.

Increasing score:

When snake feeds the food the score gets increased.

From below image it can be clear that user score and size of snake is increased.



Behind this game there is code to run this game.

Code for snake game:(c language)

using namespace std ;

#include <stdlib.h> // used for rand()

#include<conio.h> // used for getch() and kbhit()

#include<windows.h> // used for sleep()

#include<iomanip>

bool gameover ; // can also use flag variable

const int height=20 ;

const int width=50 ;

int x,y,fruitx,fruity,score ;

int tailx[100],taily[100] ;

int tail,tough=0 ;

enum direction{ // can also use glabal const int variables

STOP=0, LEFT,RIGHT,UP,DOWN // of name STOP,LEFT,RIGHT,UP,DOWN having

}; // values 0,1,2,3,4 respectivally

direction dir ;

void setup() // inicialise the inicial values,run only once

{

gameover=false ;

dir=UP ; // default movemebnt of snake

x=width/2 ; // x and y are coordinats of default snake head

y=height/2 ; // position

fruitx=rand()%width ; // coordinates of default randam fruit position

fruity=rand()%height ;

score=0 ;

tail=0 ;

}

void draw()

{

system("cls") ; //used to clear the previous screen , so that

int i,j,k ; //each time when draw() runs the position of

for(i=0;i<width+1;i++) //the box remains the same .

printf("#"); // prints upper boundary

printf("\n");

for(i=0;i<height;i++)

{

for(j=0;j<width;j++)

{

if(j==0||j==width-1) // prints side boundary

printf("#" );

if(i==y&&j==x) // snake head

printf("@" );

else if(i==fruity&&j==fruitx) // fruit

printf("$" );

else

{

bool print=false ; // can also use int flag variable

for(k=0;k<tail;k++)

{

if(tailx[k]==j&&taily[k]==i) // printing of tail of snake

{

printf("\*" );

print=true ;

}

}

if(!print)

printf(" " );

}

}

printf("\n");

}

for(i=0;i<width+1;i++) // lower boundary

printf("#") ;

printf("\n");

printf("YOUR SCORE =%d\n");

}

void input()

{

char ch ;

if(!kbhit()) // keyboard hit f(), returns 1 when we hit the

{ // keyboard else returns 0

switch(ch)

{

case 'w' : dir=UP ;

break ;

case 'a' : dir=LEFT ;

break ;

case 's' : dir=DOWN ;

break ;

case 'd' : dir=RIGHT ;

break ;

case 'x' : gameover=true ;

break ;

}

}

else // else part for default movement of

{ // the snake in the same direction till

switch(getch()) // user don't inputs any movement

{

case 'w' : dir=UP ;

ch='w' ;

break ;

case 'a' : dir=LEFT ;

ch='a' ;

break ;

case 's' : dir=DOWN ;

ch='s' ;

break ;

case 'd' : dir=RIGHT ;

ch='d' ;

break ;

case 'x' : gameover=true ;

break ;

}

}

}

void logic()

{

int prevx,prevy,prev2x,prev2y,i ; //line 120-136 for the tail

prevx=tailx[0] ;

prevy=taily[0] ;

tailx[0]=x ;

taily[0]=y ;

for(i=1;i<tail;i++) // SWAPPING of values of array

{

prev2x=tailx[i] ;

prev2y=taily[i] ;

tailx[i]=prevx ;

taily[i]=prevy ;

prevx=prev2x ;

prevy=prev2y ;

}

switch(dir)

{

case UP : y-- ;

break ;

case DOWN : y++ ;

break ;

case LEFT : x-- ;

break ;

case RIGHT : x++ ;

break ;

default : break ;

}

for(i=0;i<tail;i++) // if snake hits its own tail it will

{ // die

if(tailx[i]==x&&taily[i]==y)

{

gameover=true ;

system("cls") ;

printf("\n GAME OVER : you hit/bite your tail" );

}

}

if(tough==1)

{

if(x<0||x>width||y<0||y>height) // if the snake hit the wall it

{ // will die

gameover=true ;

system("cls") ;

printf("\n GAME OVER : you hit the wall ") ;

}

}

else

{

if(x<0) // if snake hit the wall it will continue

x=width-1 ; // from same position on the opposite

else if(x>width-2) // wall

x=0 ;

if(y<0)

y=height-1 ;

else if(y>height)

y=0 ;

}

if(x==fruitx&&y==fruity) // if snake ate the fruit score

{ // should increase and new fruit

score+=100 ; // position should be atomatically

fruitx=rand()%width ; // genereted on the screen

fruity=rand()%height ;

tail++ ; // length of tail increases

}

}

int main()

{

int n,choice ;

printf("\n\n\n...................WELCOME TO SNAKE GAME................................");

printf("\n\n\n...................KEYBOARD INSTRUCTIONS................................" );

printf("\n\n\n\t\t\tKEYBOARD KEYS\t\t\tOPERATION");

printf("\n-----------------------------------------------------------------------" );

printf("\n\n\n\t\t\tw\t\t\tuppward movement");

printf("\n\n\n\t\t\ts\t\t\tdownward movement");

printf("\n\n\n\t\t\ta\t\t\tleft movement");

printf("\n\n\n\t\t\td\t\t\tright movement");

printf("\n\n\n\t\t\tx\t\t\tquite/exit game");

printf("\n-----------------------------------------------------------------------" );

printf("\n\n\n # all keys should be in lower case only\n ( as game works on ASCII values )") ;

printf("\n\n\n enter level of game you wan to play ");

printf("\n 1. EASY \n 2. MEDIAM \n 3. HARD ");

printf("\n 4. ULTRA TOUGH (if you even hit the wall you will die )") ;

printf("\n\n please enter choice " );

scanf("&d") ;

if(choice==1)

n=100 ;

else if(choice==2)

n=50 ;

else if(choice==3)

n=10 ;

else

{

tough=1 ;

n=70 ;

}

system("cls") ;

setup() ;

while(!gameover) // can also use while(flag) we only need to run the

{ // loop infinitr no of times

draw() ;

input() ;

logic() ;

Sleep(n) ; // f() to delay the output screen each time it

} // get executed for n milli-secondes

printf("\n\n...................................................................");

printf("\n YOUR FINAL SCORE =%d") ;

printf("\n...................................................................");

getch() ;

getch() ;

return 0;

}

THANK YOU