# Project 1

# <Gluttonous Snake>

# <Version>

Name: Xiaojun He

Class: CSC-17A 48096

Date: 10/28/2015

Contents

# 1. Introduction(From wiki):

The Snake design dates back to the arcade game Blockade,[3][4] developed and published by Gremlin in 1976.[5] It was cloned as Bigfoot Bonkers the same year. In 1977, Atari released two Blockade-inspired titles: the arcade game Dominos and Atari 2600 game Surround.[6] Surround was one of the nine Atari 2600 (VCS) launch titles in the United States and was also sold by Sears under the name Chase. That same year, a similar game was launched for the Bally Astrocade as Checkmate.[7]

The first known personal computer version, titled Worm, was programmed in 1978 by Peter Trefonas of the US on the TRS-80,[3] and published by CLOAD magazine in the same year. This was followed shortly afterwards with versions from the same author for the Commodore PET and Apple II. A microcomputer clone of the Hustle arcade game, itself a clone of Blockade, was written by Peter Trefonas in 1979 and published by CLOAD.[8] An authorized version of Hustle was published by Milton Bradley for the TI-99/4A in 1980.[9] In 1982's Snake for the BBC Micro, by Dave Bresnen, the snake is controlled using the left and right arrow keys relative to the direction it is heading in. The snake increases in speed as it gets longer, and there's only one life; one mistake means starting from the beginning.

Nibbler (1982) is a single-player arcade game where the snake fits tightly into a maze, and the gameplay is faster than most snake designs. Another single-player version is part of the 1982 Tron arcade game, themed with light cycles. It created new interest in the snake concept, and many subsequent games borrowed the light cycle theme.

Starting in 1991, Nibbles was included with MS-DOS for a period of time as a QBasic sample program. In 1992 Rattler Race was released as part of the second Microsoft Entertainment Pack. It adds enemy snakes to the familiar apple-eating gameplay.

Slither.io (2016) is a massively multiplayer version of Snake.

## 2. Summary

| | |
|---|---|
| Total Line of Code | 252 |
| Variable | 15 |
| Function | 4 |

This game contains most concepts that we have learn in the class. I use the pointer to get the x and y coordinate. I use the getch() to get the information from the key board. This function do not need me to pass enter to work. I use a lot loop to build the game work.

## 3. Problem during coding

The most important problem is I cannot use just the switch statement to get the player pass. Because in this game, it do not want the player to pass enter key to control the snake. So I google it, then the friend from the web tall me I should use the conio.h library. But this library is not work in NetBeans. Then I go into NetBeans website to update my NetBeans frist.

4. System Libraries

<isotream>

<ctime>

<cstdlib>

<stdlib.h>

<conio.h>

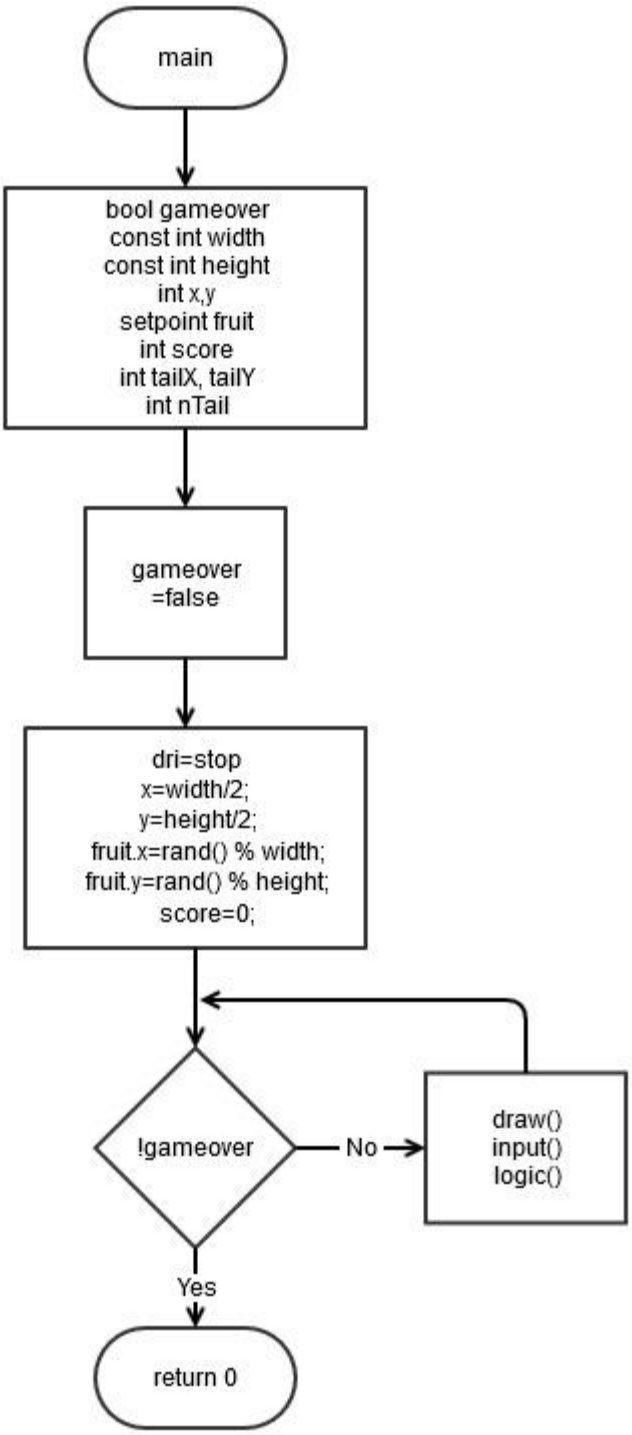
5. Variables List

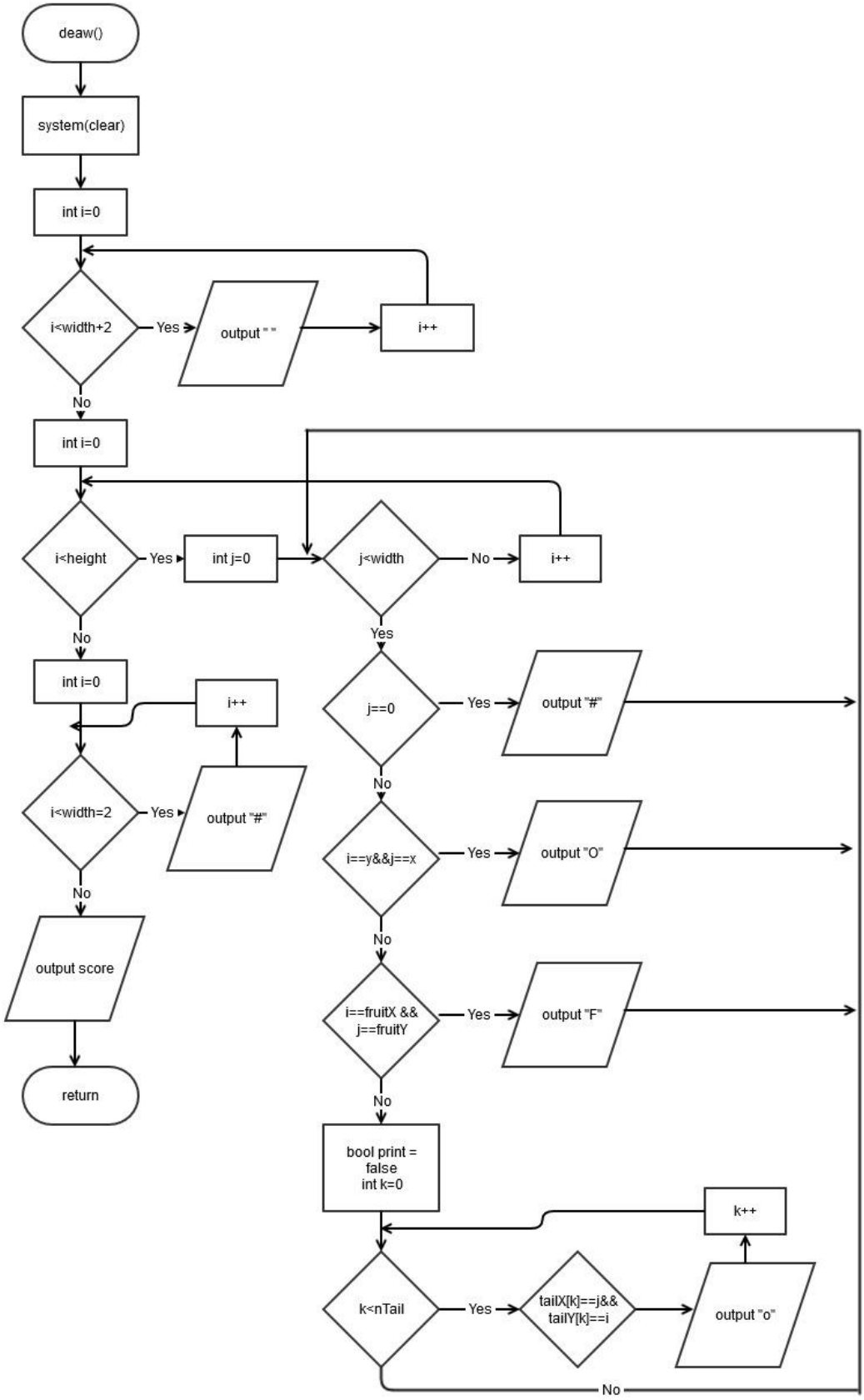
| Type | Name | Description | Line |
|---|---|---|---|
| int | X | Point x | 20 |
| | y | Point y | 20 |
| bool | gameover | Game over | 46 |
| Const int | width | width | 47 |
| | height | high | 48 |
| int | x | number | 49 |
| | y | number | 49 |
| | Fruit x | Point fruit | 50 |
| | Fruit y | Point fruit | 50 |
| | score | Score | 51 |
| | tailX | Point tail | 52 |
| | tailY | Point tail | 52 |
| | nTail | Tail long | 53 |

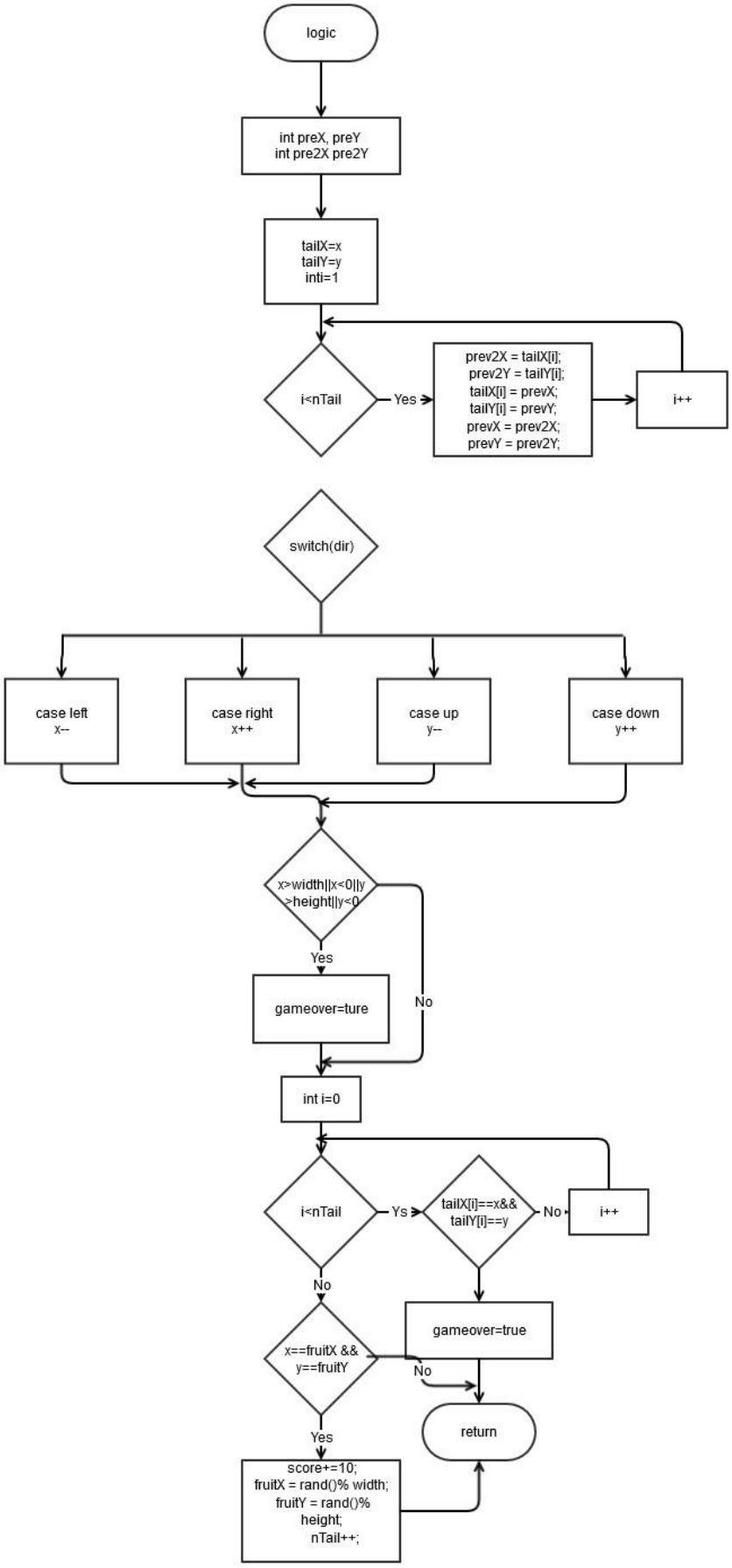
## 6. Function List

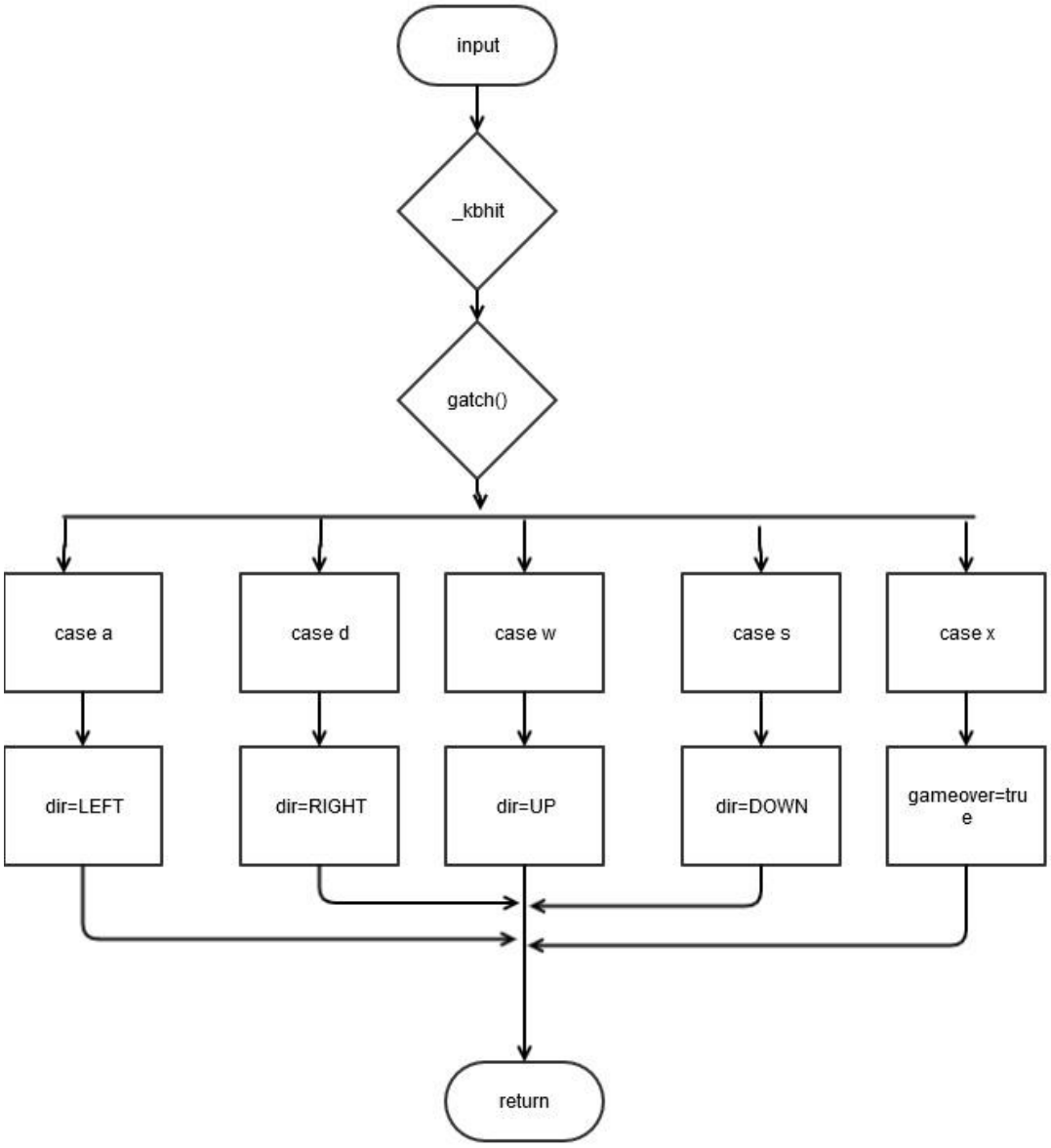
| type | name | argument | function | Location |
|---|---|---|---|---|
| Void | Draw | int width,int height,int x,int y, int fruitX,int fruitY,int nTail, int tailX,int tailY,int score | Draw the map, snake and fruit | 79 |
| Void | Input | bool gameover | input thing in the game | 140 |
| void | logic | int width,int height,int x,int y, int fruitX,int fruitY,int nTail, int tailX,int tailY,int score,bool gameover | | |

# 7. Frowchart


main
bool gameover
const int width
const int height
int x,y
setpoint fruit
int score
int tailX, tailY
int nTail
gameover
=false
dri=stop
x=width/2;
y=height/2;
fruit.x=rand() % width;
fruit.y=rand() % height;
score=0;
!gameover
No
draw()
input()
logic()
Yes
return 0


deaw()
system(clear)
int i=0
i<width+2
Yes
output " "
i++
No
int i=0
i<height
Yes
int j=0
j<width
No
i++
Yes
No
int i=0
i++
j==0
Yes
output "#"
No
i<width=2
Yes
output "#"
i==y&&j==x
Yes
output "O"
No
output score
No
i==fruitX &&
j==fruitY
Yes
output "F"
return
No
bool print =
false
int k=0
k++
k<nTail
Yes
tailX[k]==j&&
tailY[k]==i
output "o"
No


logic
int preX, preY
int pre2X pre2Y
tailX=x
tailY=y
inti=1
i<nTail
Yes
prev2X = tailX[i];
prev2Y = tailY[i];
tailX[i] = prevX;
tailY[i] = prevY;
prevX = prev2X;
prevY = prev2Y;
i++
switch(dir)
case left
x--
case right
x++
case up
y--
case down
y++
x>width||x<0||y
>height||y<0
Yes
gameover=ture
No
int i=0
i<nTail
Ys
tailX[i]==x&&
tailY[i]==y
No
i++
No
x==fruitX &&
y==fruitY
gameover=true
No
return
Yes
score+=10;
fruitX = rand()% width;
fruitY = rand()%
height;
nTail++;


input
_kbhit
gatch()
case a
case d
case w
case s
case x
dir=LEFT
dir=RIGHT
dir=UP
dir=DOWN
gameover=tru
e
return

# 8. Code

```
/*
 * File:      main.cpp
 * Author: Jimmy
 * Created on October 24, 2016, 4:13 PM
 * project 1
 */


//library
#include <iostream>
#include <ctime>
#include <cstdlib>
#include <stdlib.h>
#include <conio.h>
using namespace std;


//set the point x and y
struct setpoint{
      int x, y;
};


//set the enum to set the direction
enum eDirecton{STOP=0,LEFT,RIGHT,UP,DOWN};
eDirecton dir;


//the function to draw
//include the map , snake and the fruit
void draw(int width,int height,int x,int y,
                  int fruitX,int fruitY,int nTail,
                  int tailX,int tailY,int score);
//input thing in the game
//it use w a s d to control the snake
//w is up, s is down, a is left, d is right
void input(bool);
//the function for show the snake hand and body
//it also calculate the score
void logic(int width,int height,int x,int y,
                   int fruitX,int fruitY,int nTail,
                   int tailX,int tailY,int score,bool gameover);
```

```
int main(int argc, char** argv) {
	bool gameover;			//check game over
	const int width=20;		//set the width is 20
	const int height=20;		//set the height is 20
	int x, y;
	setpoint fruit;			//the fruit point
	int score;			//score in the game
	int tailX[100],tailY[100];	//the tail for the snake
	int nTail;			//number of tail


	gameover=false;
	dir = STOP;
	x=width/2;
	y=height/2;
	//random set the fruit
	fruit.x=rand() % width;
	fruit.y=rand() % height;
	score=0;


	//play	the game
	while(!gameover){
		draw(width,height,x,y,fruit.x,
			fruit.y,nTail,tailX,tailY,score);
		input();
		logic(width,height,x,y,fruit.x,
			fruit.y,nTail,tailX,tailY,score,gameover);
	}

	return 0;
}


//the function to draw
//include the map , snake and the fruit
void draw(int width,int height,int x,int y,
		int fruitX,int fruitY,int nTail,
		int tailX,int tailY,int score){
	system("clear");//clear

	//set the width wall
```

```
for(int i=0;i<width+2;i++)
	cout<<"#";
cout<<endl;


//set the height wall
for(int i=0;i<height;i++){
	for(int j=0;j<width;j++){
		//set the wall for left
		if(j==0)
			cout<<"#";

		//set the snake hand
		if(i==y && j==x)
			cout<<"O";

		//show the fruit
		else if(i==fruitX && j==fruitY )
			cout<<"F";

		//show the snake tail
		//the tail will be follow the snake hand to move
		else{
			bool print = false;
			for(int k=0; k<nTail;k++){
				if(tailX[k]==j&&tailY[k]==i){
					cout<<"o";
					print = true;
				}
			}
			//after the snake tail move
			//reset the tail to space
			if(!print)
					cout<<" ";
		}

		//set the wall for right
		if(j==width-1)
			cout<<"#";
	}

	cout<<endl;
}
for(int i=0;i<width+2;i++)
```

```
            cout<<"#";
        cout<<endl;

        //show the score
        cout<<"Score :"<<score<<endl;
}


//input thing in the game
//it use w a s d to control the snake
//w is up, s is down, a is left, d is right
void input(bool gameover){
        //get the    button from key board
        //it don't want pass the enter
        if(_kbhit()){
                //the key in key board
                //a mean turn left
                //d mean turn right
                //w mean go up
                //s mean go down
                switch(_getch()){
                        case 'a':
                                dir = LEFT;
                                break;

                        case 'd':
                                dir = RIGHT;
                                break;

                        case 'w':
                                dir = UP;
                                break;

                        case 's':
                                dir = DOWN;
                                break;

                        case 'x':
                                gameover = true;
                                break;
                }
        }
}
```

```
//the function for show the snake hand and body
//it also calculate the score
void logic(int width,int height,int x,int y,
                int fruitX,int fruitY,int nTail,
                int tailX,int tailY,int score,bool gameover){
    //mark down the point for the snake tail
    //and mark the number of tail
    //that mean how long is the tail
    int prevX = tailX[0];
    int prevY = tailY[0];
    int prev2X, prev2Y;
    tailX[0] = x;
    tailY[0] = y;
    //get the number for the tail
    //and mark all the point for the tail
    for(int i=1;i<nTail;i++){
        prev2X = tailX[i];
        prev2Y = tailY[i];
        tailX[i] = prevX;
        tailY[i] = prevY;
        prevX = prev2X;
        prevY = prev2Y;
    }
    //get the button from the key board
    //x is for left and right
    //if x -1, that mean turn left
    //if x +1, that mean turn right
    //y is fro up and down
    //if y-1. that mean up
    //if y+1, that mean down
    switch(dir){

        case LEFT:
            x--;
            break;

        case RIGHT:
            x++;
            break;

        case UP:
            y--;
            break;
```

```
            case DOWN:
                y++;
                break;

            default:
                break;

        }


        //if the snake hand in to the wall
        //the game is over
        if(x>width||x<0||y>height||y<0)
            gameover = true;

        //if the snake hand get in to snake tail
        //the game is over
        for(int i=0;i<nTail;i++)
            if(tailX[i]==x&&tailY[i]==y)
                gameover = true;

        //if the snake hand get in to the fruit
        //that mean the snake eat the fruit
        //the fruit will reset
        //and the tail of snake will +1
        if(x==fruitX && y==fruitY){
            score+=10;
            fruitX = rand()% width;
            fruitY = rand()% height;
            nTail++;
        }
    }
```