CODE

```
#include <vector>

#include <iostream>

#include <conio.h>

#include <windows.h>

#include <ctime>

class Snake

{

private:

    int x,z,Dollars,y,Bonus,time,tail;

    // x - snake head position

    // z = for random number(pineapple generation)

    // y = tail coordinates time = time tail = snakelenght - 1

    char *map; //Map of game

    bool L,R,D,U,A; //L = LEFT R = RIGHT D = DOWN U = UP A = JUST A
BOOL

    std::vector <int> past; //Special vector for deleting old parts


public:

    Snake()

    {

        map = new char [2000];

        L = 0;

        R = 1; //bool 1 = true bool 0 = false

        D = 0;
        U = 0;
        A = 0;
        tail = 1; //Actually it will not have a tail until it eats first apple
        x = 1000; //Position of snake at the beggining
        Dollars = 0; //Money
        Bonus = 0;
        time = 40; //For bonus apples
        z = 1; //So first apple generation will not be bugged --
          }

    ~Snake()
    {
        delete [] map;
    }

    void Graphics (); //Drawing game
    void GameLogic(); //Does it need comment?
    void KEYBOARD (); //Checking for input
    void Pineapple(); //Generating new apple
    int GameOver (); //Game over ?.
    void Start(); //Start -_-
    friend void clearscreen(); //It's actually not like system("cls"),but works almost
same
    friend void sp();//Choose color function
    friend void s(); //Forget choose color choice
};
```

```
//////////////////////////////////////////////////////////////////////////////////// Windows
void clearscreen()
{
    HANDLE hOut;
    COORD Position;

    hOut = GetStdHandle(STD_OUTPUT_HANDLE);

    Position.X = 0;
    Position.Y = 0;
    SetConsoleCursorPosition(hOut, Position);
}

void sp(int choosecolor)
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
choosecolor); //FUNCTION OF COLOR
}
void s()
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
    //FUNCTION TO ..EHM FORGET COLOR,not sure how to say I know: Stop
using color
}
void Snake::Start()
{
    for(int p = 0;p < 2000;p++)
    {
        map[p] = ' ';
    }

    map[ x ] = char(219); //Let's make head - block
    Pineapple(); //Let's generate a pineapple
    Graphics(); //GOOOOOOO
}
////////////////////////////////////////////////////////////////////////////////////////
int Snake::GameOver()
{
    Sleep(2500);
    system("cls");
    std::cout <<     std::endl << "Oops...You earned " << Dollars + Bonus << "
Dollars...";
    //Better luck next time.
    Sleep(1800);
    return 0;
}
void Snake::Graphics()
{
    sp(697); //CHoosing color
    std::cout << Dollars + Bonus << " Dollars By Foxefde 2013 ";
    std::cout << "\n";
    s();
```

```
for(int u = 0;u < 50;u++) //Top border
{
   sp(750);
   std::cout << char(219);
   s();
}

     std::cout <<     std::endl;
for(int x1 = 0;x1 < 2000 ; x1++) //DRAWING BOARD!~
{
   if(x1 % 50 == 0 && x1 != 0)
   {
      std::cout <<     std::endl;
   }
   if(x1 % 50 == 0 || (x1-(x1 / 50)) % 49 == 0)
             {
      map[x1] = char(219);
      sp(750);
      std::cout << map[x1];
      s();
             }
     else if(map[x1]!=char(219) && map[x1]!=map[z])
{
   std::cout << map[x1];
}
```

```
else if(x1 == z)
{
   sp(10);
   std::cout << map[x1];
   s();
}
else
{
   sp(750);
   std::cout << map[x1];
   s();
     }


}


     std::cout <<    std::endl; //New line
for(int u = 0;u < 50;u++) //Bottom border
{
   sp(750);
   std::cout << char(219);
   s();
}
if(U == 1 || D == 1)
     {
   Sleep(19);
```

```
    }
    if(map[z] == char(5)) //If apple is bonus ,then start decreasing time to get it!
    {
        time--;
        if(time == 0) //If you were too slow ,you lose a dollar and a new apple is
generated
                    {
            Dollars++;
            time = 40;
            Pineapple();
        }
    }

    clearscreen();
    GameLogic();
}

void Snake::Pineapple()
{
    map[z] = ' ';
    if(Dollars % 8 != 0 || Dollars == 0)
    //Bonus apple - every 8 normal apples eaten ,so we need to check - generate an
apple or a bonus apple
    {
        while(map[z] != ' ' && z % 50 != 0 && (z-(z/50)) % 49!= 0);
    //Keep generating new coordinates of pineapple until that place is empty
```

```
        {
            z = rand()%2000 + 1;
                    }


              map[z] = char(229);
        //(z(z/50)) % 49 != 0 that means ,if z isn't 49+50n (49,99,149,199...)
    } //Logic ftw ,yeh?:D

    else //BONUS APPLE
    {
        while(map[z] != ' ' && z % 50 != 0 && (z-(z/50)) % 49 != 0);
        //Keep generating new coordinates of pineapple until that place is empty
        {
            z = rand()%2000 + 1;
        }
        map[z] = char(5);
    }

}

void Snake::KEYBOARD()
{
if(_kbhit()) //If player clicks something

{
```

```
char key;
key = _getch(); //Now this click is key
switch( key )
{
   case 'd':
      //cases below,nothing special ,you should understand that easily
      //,but let me give a simple explanation of first case:
   {
      if (L == 0) //So if a player has clicked 'd' ,then.1:We check if snake is not
      //going left,because how can she turn right,if she's going left?Teleporting?..
      {
         L = 0, U = 0, D = 0, //Left = false UP = false Down = false
         R = 1; //Right = true !
                    }
      break; //We break it,end of the case.Identic with other cases..
   }

   case 'w':
   {
      if (D == 0)
      {
         L = 0, D = 0, R = 0,
         U = 1;
      }
   break;
```

```
        }

        case 'a':
        {
            if (R == 0)
            {
                D = 0, U = 0, R = 0,
                L = 1;
                              }
        break;
        }

        case 's':
        {
            if (U == 0)
            {
                L = 0, U = 0, R = 0,
                D = 1;
            }
        break;
        }

    }

}
```

```
}
void Snake::GameLogic()
{
    past.insert(past.begin(),x); //Inserting past x position to vector
    KEYBOARD();
    if(R == 1) //If snake is going right
    {
        x++;
    }
    else if(L == 1) //If snake is going left
    {
        x--;
    }
    else if(U == 1) //If snake is going up
    {
        x-= 50;
    }
    else //If snake is going down(only case left)
    {
        x+= 50;
    }
    if(map[ x ] == char(219) || x % 50 == 0 || x > 2000 || x < 0 || (x-(x / 50)) % 49 ==
0)
        //If it hits herself or border...
    {
        GameOver();
    return;
}
if(map[x] == char(229))
//If it eats an pineapple also A becomes true,that means the
//very end of the snake(tail) will not be deleted ( line 276 ) for 1 frame
{
    A = true;
    tail++; //It eats a big apple ,so snake becomes heavier
    Pineapple(); //Let's generate new apple!
    Dollars++; //Apple had some dollars in it,congrulations!
      }
else if (map[x] == char(5)) //Same,but maybe it has just eaten bonus apple?
{
    A = true;
    tail++;
    Pineapple();
    Bonus+=time;
}

map[x] = char(219);
//When it touches apple - head becomes an apple,so we need to change it.
if(A == false) //If snake has just eaten an apple
{
    y = past[past.size() - tail]; //D E L E T I N G past tail from the map!
    map[y] = ' ';
    }

    A = false; //So the next time line 270 will work again,if apple is no eaten ///
    if(tail!=1)
    {
        for(int u = past.size() - 2;u > 0;u--)
                    {
            past[u+1] = past[u];
        } //Try your best to understand what's happening here,
        //let's say this is an exercise for you from Foxefde
    past.erase(past.end()-tail);
    }
    else
    {
        past.erase(past.begin()); //I could do it without erases,
    } //
    Graphics();
}
int main()
{
    srand((unsigned)time(0)); //So random numbers will be always random.
    Snake SNAKE; //creating class m
    SNAKE.Start(); //Starting main function/
}
```