VIT®
UNIVERSITY
(Estd. u's 3 of UGC Act 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING(SCOPE)

DATA STRUCTURES AND ALGORITHMS {CSE 2003} WINTER SEM PROJCET REPORT - 2016-2017

TOPIC: SNAKE GAME USING DATA STRUCTURES

UNDER THE GUIDENCE OF:

M. GOPINATH

SENIOR PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING VIT UNVERSITY
VELLORE-632 014

ACKNOWLEDGEMENT

I take this opportunity to present my votes of thanks to all those guidepost who really acted as lightening pillars to enlighten our way throughout this project that has led to successful and satisfactory completion of this study.

We are really grateful to our HOD Mr. SENTHIL KUMAR for providing us with an opportunity to undertake this project in this university and providing us with all the facilities. We are highly thankful to MR. BALA KRUSHNA TRIPATY for her active support, valuable time and advice, whole-hearted guidance, sincere cooperation and pains-taking involvement during the study and in completing the assignment of preparing the said project within the time stipulated.

Lastly, We are thankful to all those, particularly the various friends , who have been instrumental in creating proper, healthy and conductive environment and including new and fresh innovative ideas for us during the project, their help, it would have been extremely difficult for us to prepare the project in a time bound framework.

SUBMITTED BY:-

SR NAVYA SREE-16BCE0864

TABLE OF CONTENTS

INTRODUCTION

The following is an example game written in C based on the game called 'snake' which has been around since the earliest days of home computing (I can remember writing a version of it for my ZX81), and has re-emerged in recent years on mobile phones.

It isn't the world's greatest game, but it does give you an idea of what you can achieve with a relatively simple C program, and perhaps the basis by which to extend the principles and create more interesting games of your own.

.

ABSTRACT:

The client uses MS Excel, and maintains their records, however it is not possible them to share the data from multiple system in multi user environment, there is lot of duplicate work, and chance of mistake. When the records are changed they need to update each and every excel file. There is no option to find and print previous saved records. There is no security; any body can access any report and sensitive data, also no reports to summary report. This Snake Game In C Language is used to overcome the entire problem which they are facing currently, and making complete atomization of manual system to computerized system.

Playing the game

It is about 123k in length. Note that on faster PCs, it will be unplayably fast, so you will need to re-compile it with the pause_length constant set to a higher value. My PC is about 350MHz and it is OK. To move the snake, use up arrow key to move the snake , down arrow key for downwards movement of the snake , left arrow key for moving it left and right key to move it rightwards. Again, there are constants you can change if you want to alter these settings. Press 'esc' to exit the game at any time.

The aim of the game is to collect the dots (food) and avoid the obstacles (crosses, borders, and the snake itself). As you collect food, the snake gets longer, so increasing your likelihood of crashing into yourself. When you have collected enough food, you progress onto the next level, where your snake gets longer, and the amount of food to collect to progress through the level gets larger. You get scored according to the length of the snake and the number of 'x' obstacles on the screen. As soon as the snake eats the food There is no concept of lives. Once you hit an obstacle, that's it, game over.

PROPOSED SYSTEM

The following documentation is a project the "Name of the term paper allotted". It is a detailed summary of all the drawbacks of the old system and how the new proposed system overcomes these shortcomings. The new system takes into account the various factors while designing a new system. It keeps into the account the Economical bandwidth available for the new system. The foremost thing that is taken care of is the Need and Requirements of the User.

DESCRIPTION

Before developing software we keep following things in mind that we can develop powerful and quality software

PROBLEM STATEMENT

- Problem statement was to design a module:
- Which is user friendly
- Which will restrict the user from accessing other user's data.
- Which will help user in viewing his data and privileges.
- Which will help the administrator to handle all the changes.

FUNCTIONS TO BE PROVIDED:

The system will be user friendly and completely menu driven so that the users shall have no problem in using all options.

- The system will be efficient and fast in response.
- The system will be customized according to needs.
- It is a game which can play by anyone
- It is easy to use

SYSTEM REQUIRMENTS

Operating system: MS Windows XP or Windows Vista
Language: C Language
Processor: Pentium IV Processor
RAM: 512 MB
Hard disk: 5 GB

REQUIREMENT ANALYSIS

This process is adopted when management of the system development, Personnel decide that the particular system needs improvement. The system development life cycle is the set of activities, carried out by the analyst, designers and users to develop and implement a system. The systems that are present in the nature follow common life cycle pattern. For example consider the raining system. Initially the rain falls into the river, river flows into sea, the sea water evaporates to form vapors, the vapors form clouds which again bring rain. Similarly consider a man made system initially a system is analyzed, designed and made operational by the efforts of system analysis. After successful operation or a number of users, the system becomes less and less effective by change in the environment. So these changes have to be incorporated in to the system by minor modifications. So the general activities from the life cycle of the system are given below:

- Select ion and identification of the system to be studied
- Preliminary study
- Defining the system
- Design and development of the system
- Implementation of the system

Data structures and Algorithms used

1)STRUCTURES

In this we can define the various data structures ,in this linked list

2)DRAW

In this we will draw the boundary where the snake should be and snake and food also

3)GAME

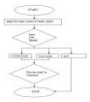
In this we will write code for geting new node when it eats food, and increasing score and size of the snake

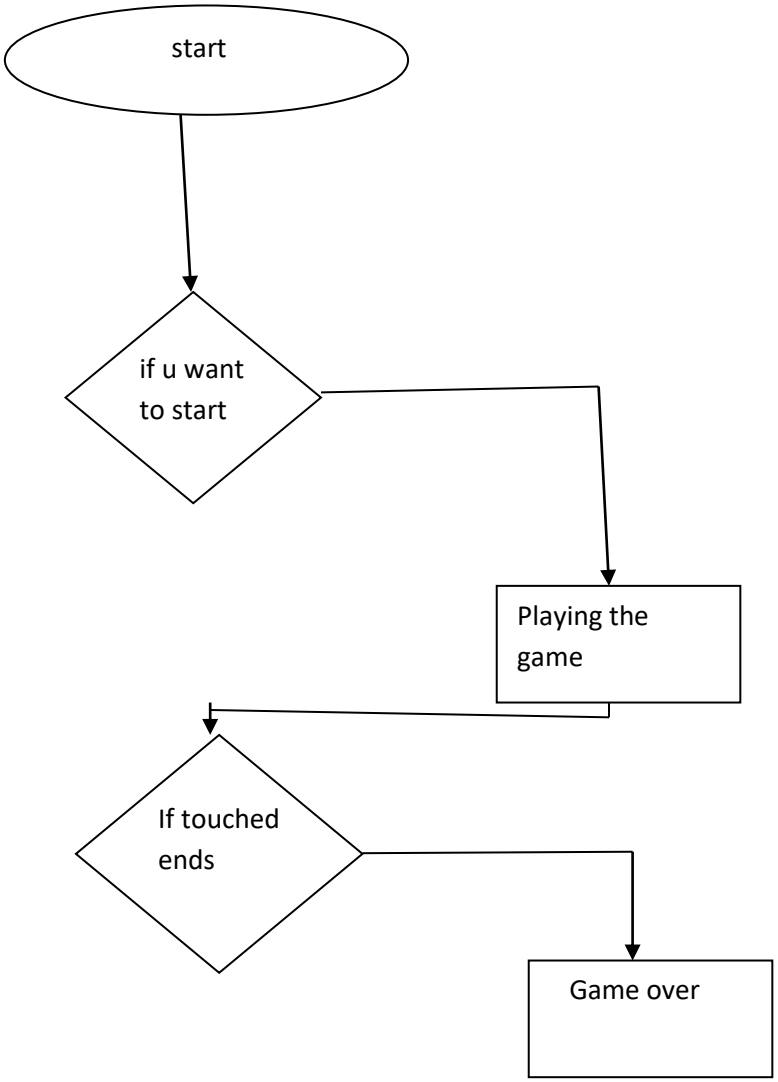
4) GAME OVER

This contains the code after the game over and conditions which leads to game over

SYSTEM DESIGN

Then we began with the design phase of the system. System design is a solution, a “HOW TO” approach to the creation of a new system. It translates system requirements into ways by which they can be made operational. It is a translational from a user oriented document to a document oriented programmers. For that, it provides the understanding and procedural details necessary for the implementation. Here we use Flowchart to supplement the working of the new system. The system thus made should be reliable, durable and above all should have least possible maintenance costs. It should overcome all the drawbacks of the Old existing system and most Important of all meet the user requirements.





SOURCE CODE

```
// -------------FILE INCLUDED----------------------

#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<memory.h>
#include<stdlib.h>
#include<dos.h>

//---------------REQUIRED STRUCTURE----------------




char key;

struct loc
{
        int x,y;
};

struct snake
{
        struct loc sloc;
        struct snake *link;
        char dir;
};

struct game_data
{
        int score;
        int no_food;

};
struct game_data gd={0,0};
```

```
struct limit
{
	int lx1,ly1,lx2,ly2;
};

struct limit l={96,96,404,404};


struct food
{
	struct loc floc;
	int number;
};

int n=0;

//---------------------------------------------------------------
//-----------------------FUNCTIONS-------------------------------
//---------------------------------------------------------------
void draw(struct snake *head,struct food *f)
{

	struct snake *temp;
	temp=head;



	rectangle(96,96,404,404);
	rectangle(98,98,402,402);
	rectangle(100,100,400,400);
	setfillstyle(9,13);
	bar(temp->sloc.x-6,temp->sloc.y-6,temp->sloc.x+6,temp->sloc.y+6);
	temp=temp->link;
	setfillstyle(9,2);
	while(temp->link!=NULL)
	{

		bar(temp->sloc.x-5,temp->sloc.y-5,temp->sloc.x+5,temp->sloc.y+5);
		temp=temp->link;
	}

	bar(temp->sloc.x-5,temp->sloc.y-5,temp->sloc.x+5,temp->sloc.y+5);

	circle(f->floc.x-2,f->floc.y-2,5);
	circle(f->floc.x+2,f->floc.y+2,5);
	circle(f->floc.x-2,f->floc.y+2,5);
```

```
        circle(f->floc.x+2,f->floc.y-2,5);

        delay(20);
        while(!kbhit()){goto e;}
        key=getche();
        e:

     cleardevice();

}



gameover(void)
{
        cleardevice();
        outtextxy(100,100,"---------- GAME OVER ----------");

        printf("\n\n\n\n\n\n\n\n");
        printf("\t\t# Score      - %d",gd.score);
        printf("\n\t\t# No of food - %d", gd.no_food);
        s1:
        sound(300);delay(300);sound(450);delay(150);sound(500);delay(150);
        sound(300);delay(200);sound(450);delay(100);sound(450);delay(200);
        while(!kbhit()){goto s1;}
        nosound();

     return(0);
}


void game(struct snake *head,struct food *f)
{
    struct snake *temp,pre,nxt;
    temp=head;

    while(key!='p')
    {
        if(head->sloc.x==l.lx1||head->sloc.x==l.ly2||head->sloc.y==l.ly1||head->sloc.y==l.ly2)
        {gameover();}



        if(head->sloc.x>=f->floc.x-5&&head->sloc.x<=f->floc.x+5&&head->sloc.y>=f->floc.y-
5&&head->sloc.y<=f->floc.y+5)
```

```
        {
                temp=head;
                sound(420);

                f->floc.x=150+random(245);
                f->floc.y=150+random(245);
                gd.score+=100;
                gd.no_food+=1;
                n=n+1;

              //  if(n==2)
              //  {
                while(temp->link!=NULL){temp=temp->link;}
                temp->link=(struct snake *)malloc(sizeof(struct snake));
                temp->link->link=NULL;
                temp->link->sloc.x= temp->sloc.x;
                temp->link->sloc.y= temp->sloc.y;
                temp->link->dir=temp->dir;
                n=0;
              //  }


              }



        switch(key)
        {
          case 'a': if(head->dir!='d'){head->dir='a';  head->sloc.x-=2; } else {key=head->dir;}
break;
          case 'w': if(head->dir!='s'){head->dir='w';  head->sloc.y-=2; } else {key=head->dir;}
break;
          case 'd': if(head->dir!='a'){head->dir='d';  head->sloc.x+=2; } else {key=head->dir;}
break;
          case 's': if(head->dir!='w'){head->dir='s';  head->sloc.y+=2; } else {key=head->dir;}
break;

        }
      //cleardevice();
      draw(head,f);
      nosound();
      temp=head;
      pre=*temp;

      while(temp->link!=NULL)
        {
```

```
                nxt.sloc.x=temp->link->sloc.x;
                nxt.sloc.y=temp->link->sloc.y;
                nxt.dir=temp->link->dir;
                temp->link->sloc.x=pre.sloc.x;
                temp->link->sloc.y=pre.sloc.y;
                temp->link->dir=pre.dir;
                temp=temp->link;
                pre=nxt;
        }


   }

}




void main(void)
{

        struct snake *s;
        struct food f;
        int gdriver = DETECT, gmode, errorcode;



  clrscr();

  s=(struct snake *)malloc(sizeof(struct snake));
  s->dir='w';
  s->link=NULL;
  s->sloc.x= s->sloc.y=300;
  f.floc.x=150+random(250);
  f.floc.y=150+random(250);
  //-------------------------------------------------------

  initgraph(&gdriver, &gmode, "c:\\tc\\bgi\\egavga.bgi");

  //-------- # ENTER REQUIREED PATH FOR EGAVGA.BGI----------

  outtextxy(300,50,"SNAKE");
  outtextxy(220,100,"Created By - Viral M Parekh");
  outtextxy(100,160,"# Instructions :");
```

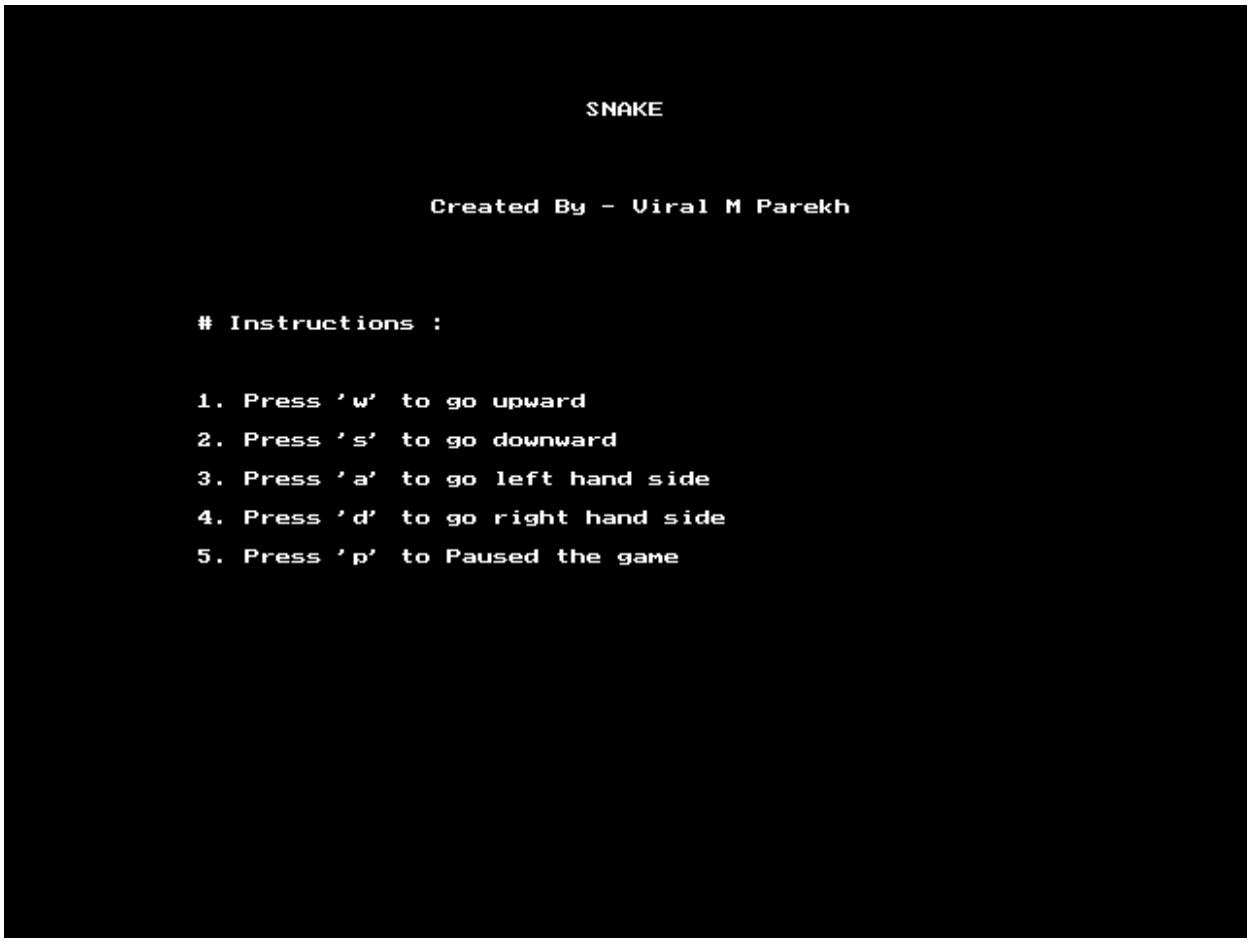
```
  outtextxy(100,200,"1. Press 'w' to go upward");
  outtextxy(100,220,"2. Press 's' to go downward");
  outtextxy(100,240,"3. Press 'a' to go left hand side");
  outtextxy(100,260,"4. Press 'd' to go right hand side");
  outtextxy(100,280,"5. Press 'p' to Paused the game");
  s:
  sound(700);delay(200);sound(450);delay(180);sound(300);delay(150);
  sound(350);delay(200);sound(450);delay(180);sound(500);delay(150);
  while(!kbhit()){goto s;}

  nosound();



  pause :
  draw(s,&f);
  game(s,&f);
  key='r';
  rectangle(60,60,600,200);
  outtextxy(260,100,"GAME PAUSED");
  s1:
  sound(400);delay(500);sound(450);delay(250);sound(300);delay(200);
  sound(400);delay(300);sound(450);delay(250);sound(300);delay(200);
  while(!kbhit()){goto s1;}
  goto pause;



  closegraph();
  clrscr();

  printf("\n\n\n\n\t\t\t\tThankyou.....");
  getch();

}
```

TESTING

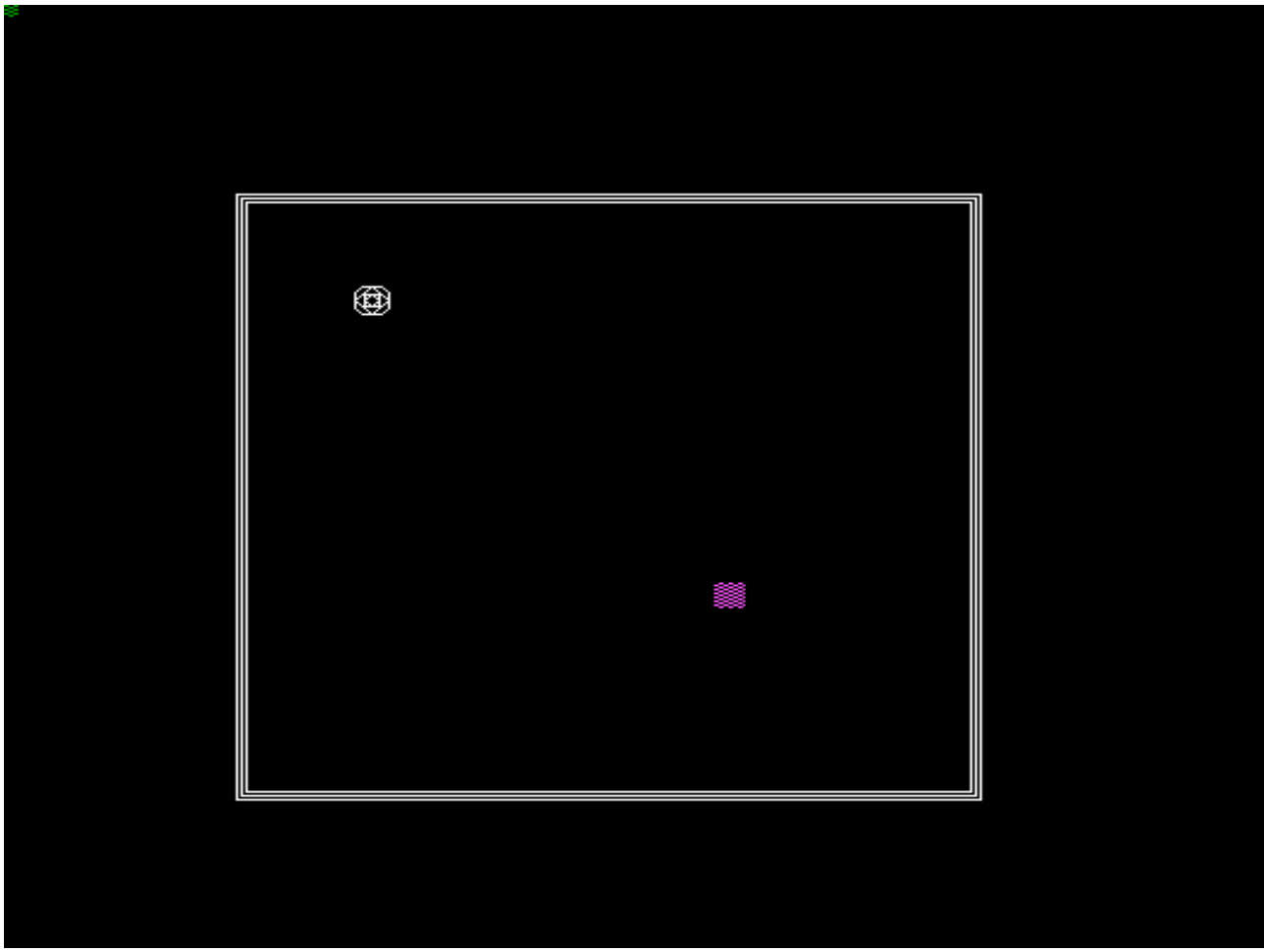
Testing is the major control measure used during software development. Its basic function is to detect errors in the software. During requirement analysis and design, the output is a document that is usually textual and no executable. After the coding phase, computer programs are available that can be executed for testing purpose. This implies that testing not only, has to uncover errors introduced during coding, but also errors introduced during previous phase. Thus the goal of testing is to uncover the requirements, design and coding errors in the programs. So after testing the outputs of my project are as follows:

RESULTS AND SNAPHOTS

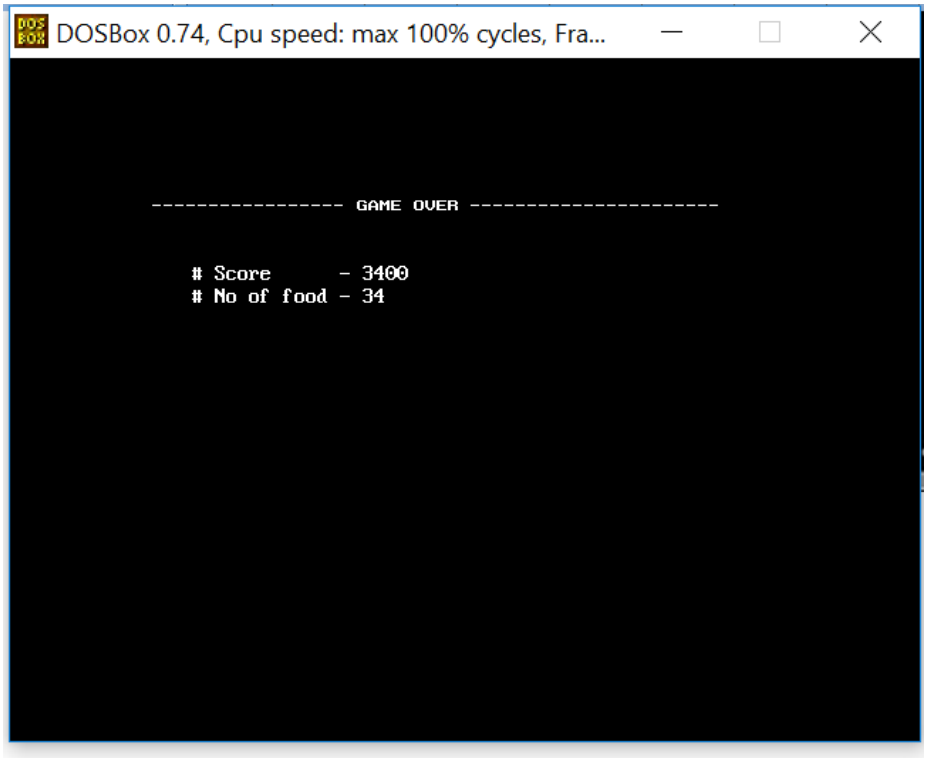
Main menu screen



If snake crosses the line then the life would decrease



If the snake touches the boarder game would be over.



The main limitation of Snake Game In C Language:

- The existing system only provides text-based interface, which is not as user-friendly as Graphical user Interface.
- Since the system is implemented in Manual, so the response is very slow.
- The transactions are executed in off-line mode, hence on-line data capture and modification is not possible.
- Off-line reports cannot be generated due to batch mode execution.

Hence, there is a need of reformation of the system with more advantages and flexibility. The Snake Game In C Language eliminates most of the limitations of the existing software. It has the following objectives:

Enhancement:

The main objective of Snake Game In C Language is to enhance and upgrade the existing system by increasing its efficiency and effectiveness. The software improves the working methods by replacing the existing manual system with the computer-based system.

Automation:

The Snake Game In C Language automates each and every activity of the manual system and increases its throughput. Thus the response time of the system is very less and it works very fast.

Accuracy:

The Snake Game In C Language provides the uses a quick response with very accurate information regarding the users etc. Any details or system in an accurate manner, as and when required.

User-Friendly:

The software Snake Game In C Language has a very user-friendly interface. Thus the users will feel very easy to work on it. The software provides accuracy along with a pleasant interface. Make the present manual system more interactive, speedy and user friendly.

Availability:

The transaction reports of the system can be retried as and when required. Thus, there is no delay in the availability of any information, whatever needed, can be captured very quickly and easily.

FUTURE SCOPE OF THE PROJECT

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to College. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly.

The present project has been developed to meet the aspirations indicated in the modern age. An attempt has been made through this project to do all work ease & fast. It provide current add, Update, Move Next, Move Previous, Move Last, Find & Delete all facilities to accomplish the desired objectives. The facility Include in this project and the suggested activities have been organized to impart knowledge & develop skill & attitude in the College official works.

REFERENCES

http://ba-programmer.blogspot.com/2014/03/mini-project-snake-game-in-c.html

http://viral-cs.blogspot.in/2013/03/snake-game-using-concept-of-linked-list.html

https://www.slideshare.net/azharniaz3/report-on-snake-game