A Project Report
On
“Snake and Ladder”

For The Course

“Software Development Project-I”

Course Code: ICT-1210

Submitted By:

Name: Mst. Mostary Khatun
ID: IT-20036

Name: Ashik Ahmed Sajib
ID: IT-20037

Name: Tanvir Hasan
ID: IT-20038

Supervised By:

Dr. Md. Abir Hossain

Associate Professor

Dept. Of ICT ,MBSTU.

Mawlana Bhashani Science and Technology University

Department of Information and Communication Technology

Santosh ,Tangail-1902.

Declaration:

This is to certify that the work presented in this project is carried out by the candidate under the supervision of **Dr. Md. Abir Hossain** in the Department of Information and Communication Technology, MBSTU, Tangail, Bangladesh. It is also declared that neither of this project has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Signature of Supervisor

Dr. Md. Abir Hossain

Associate Professor

Department of ICT, MBSTU

ACKNOWLEDGEMENT:

We would like to express my gratitude towards guide **Dr. Md. Abir Hossain** for the useful instructions, remarks and for giving his valuable guidance and inspiration throughout the learning process of this report. Furthermore, we would like to thank him for making available all the facilities for the successful completion of this work and other staff members of the ICT Department for their valuable help. It is with humble gratitude & sense of indebtedness, we thank our respected and esteemed Dean for his valuable guidance, suggestion and constant support which lead towards successful completion of this work. We take this opportunity to express our sincere gratitude to all those who helped us in various capacities in undertaking this project and devising the report. We are privileged to express our sense of gratitude to our respected teacher **Dr. Md. Abir Hossain** whose unparalleled knowledge moral fiber and judgment along with her immense support in completing the project. We take this opportunity also to thank our friends and contemporaries for their cooperation and compliance.

CONTENTS

INTRODUCTION

1.1 Abstract

Snake and ladder is well known game among children even among matured people.

The rules and regulation of the game are as well-known as the game. The case study meant for implementing this game without losing its interesting and attraction. The game is in two modes-two player and one player mode. In the one player the computer itself will act as the second player.

The user can interact with the game using either keyboard or mouse. The number and position of the ladder and snake are generated fixed.

Random mode numbering is used for user action in which the user is able to press the button long time and the number goes to a loop and when user released the number was displayed.

SYSTEM STUDY

2.1 Existing System

Snakes and Ladders is an ancient Indian board game regarded today as a worldwide classic. It is played between two or more players on a gameboard having numbered, gridded squares.

A number of "ladders" and "snakes" are pictured on the board, each connecting two specific board squares. The object of the game is to navigate one's game piece, according to die rolls, from the start (bottom square) to the finish (Top Square), helped or hindered by ladders and snakes respectively.

The historic version had root in morality lessons, where a player's progression up the board represented a life journey complicated by virtues (ladders) and vices (snakes).

There was not a good GUI for Snake & ladder game for DOS operating system. We are trying to develop a system which makes, look and feel very interesting to play the game.

Snakes and Ladders originated in India as part of a family of dice board games, that included Gyan chauper and pachisi (present-day Ludo and Parcheesi). It was known as moksha patam or vaikunthapaali or paramapada sopaanam (the ladder to salvation). The game made its way to England and was sold as "Snakes and Ladders", then the basic concept was introduced in the United States as Chutes and Ladders (an "improved new version of England's famous indoor sport") by game pioneer Milton Bradley in 1943.

2.1 Proposed System

The proposed system was designed to give a professional look and feeling GUI for Snake & Ladder game.

The system has a splash screen, which was animated and designed with good look and feel. Game board was drawn by taking all initial screen conditions and mouse interaction. Interrupts and sound functions are used for fast mouse interactions.

The program was divided in to two screens:

Splash Screen

- Program Splash and Details

Game Window

- Game square board.
- Dice display board.
- Dice button for running the dice.
- Single / Double Mode.

Game Play

Each player starts with a token on the starting square (usually the "1" grid square in the bottom left corner, or simply, the imaginary space beside the "1" grid square) and takes turns to roll a single die to move the token by the number of squares indicated by the die roll.

Tokens follow a fixed route marked on the game board which usually follows a boustrophedon (ox-plow) track from the bottom to the top of the playing area, passing once through every square. If, on completion of a move, a player's token lands on the lower-numbered end of a "ladder", the player moves the token up to the ladder's higher-numbered square. If the player lands on the higher-numbered square of a "snake" (or chute), the token must be moved down to the snake's lower-numbered square.

If a player rolls a 6, the player may, after moving, immediately take another turn; otherwise play passes to the next player in turn. The player who is first to bring their token to the last square of the track is the winner.

SYSTEM SPECIFICATION

3.1 Software Requirements

Operating System	: DOS
Developing software	: Turbo C
Language used	: C++
Library	: Graphics

3.2 Hardware Requirements

Processor	: PC with a Pentium II-class.
Memory	: 256 MB RAM.
Hard disk space	: 2 .5 GB installation drive.
Monitor	: 15 inch Color Monitor.
Keyboard	: 105 Keys.
Display Type	: Super VGA (800x600).
Mouse	: Serial Mouse.

SYSTEM ANALYSIS

4.1 Feasibility Analysis

The main aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop the product.

The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on these data, the output data required to be produced by the system as well as various constraints on the behavior of the system.

In this project there are many details available related to the Snake & ladder game and its functions.

4.1.1 Economic & Technical Feasibility

This is concerned with specifying equipment and software that will successfully satisfy the user requirement, in examining technical feasibility, configuration of the system is given more importance than the actual make of hardware.

Snake & ladder game is developed for DOS operating system and most of the latest version of Windows series OS is supported with DOS OS. Tools development for the use of the system is Turboc C compiler.

C Programming Language

In computing, C is a general-purpose programming language initially developed by Dennis Ritchie between 1969 and 1973 at AT&T Bell Labs.[4] Like most imperative languages in the ALGOL tradition, C has facilities for structured programming and allows lexical variable scope and recursion, while a static type system prevents many unintended operations.

Its design provides constructs that map efficiently to typical machine instructions, and therefore it has found lasting use in applications that had formerly been coded in assembly, most notably system software like the Unix computer operating system.

Certain aspects of the ANSI C standard are not defined exactly by ANSI. Instead, each implementor of a C compiler is free to define these aspects individually. This chapter tells how Borland has chosen to define these implementation-specific standards. The section numbers refer to the February 1990 ANSI Standard.

Programming Tools

Turbo C**++** was a C++ compiler and integrated development environment and computer language originally from Borland. Most recently it was distributed by Embarcadero Technologies.

Graphics Library

C graphics using graphics' functions or WinBGIM (Windows 7) can be used to draw different shapes, display text in different fonts, change colors and many more. Using functions of graphics.h in turbo c compiler you can make graphics programs, animations, projects and games.

The program was completely developed with the support of **Graphics.h** library. *Graphics.h : Declare prototype for graphics functions*

Graphics in C Language to 16 bit C programming and MS DOS environment. In a C Program first of all you need to initialize the graphics drivers on the computer. This is done using the **initgraph** method provided in graphics.h library.

Initgraph initializes the graphics system by loading a graphics driver from disk (or validating a registered driver) then putting the system into graphics mode.Initgraph also resets all graphics settings (color, palette, current position, viewport, etc.) to their defaults, then resets graphresult to 0.

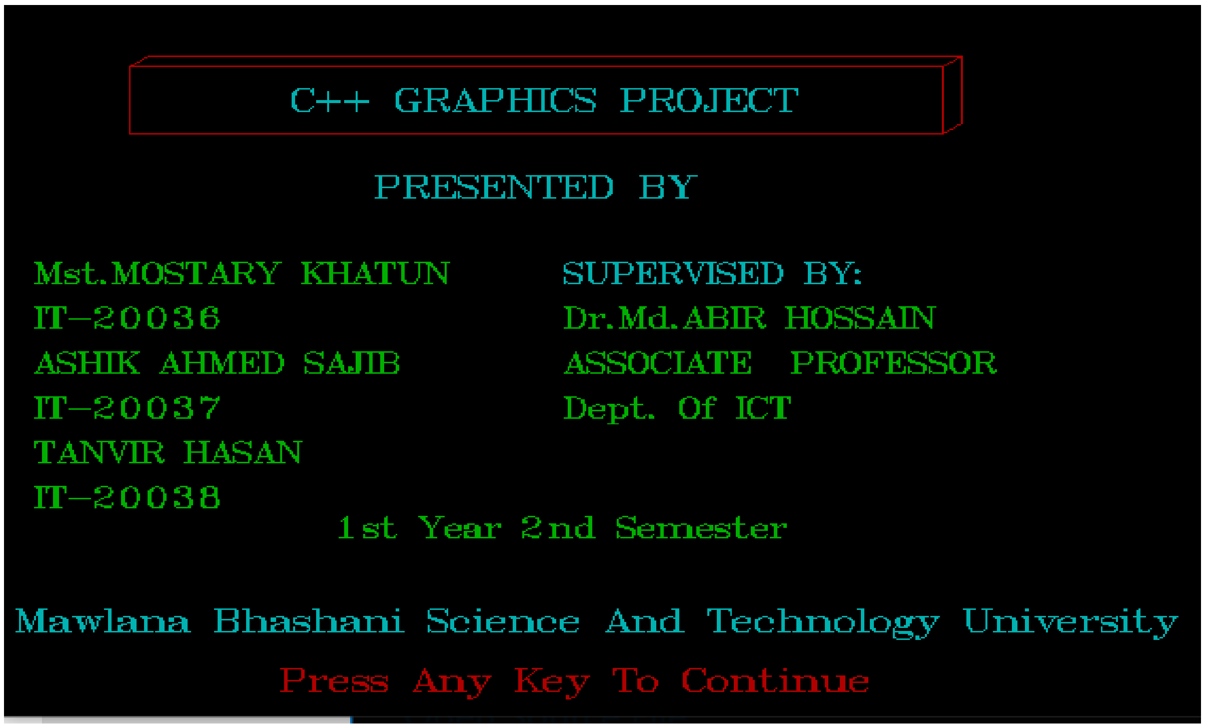
4.1.2 Social & Behavioral Feasibility

This concerned with the requirements and analysis of the program. Interface and layout was designed in a manner that the major part of the system was allocated to the game player board and the right side of the screen was allocated to dice running and score board display details.

Small square box was allocated to each player for moving there icon on game board with the dice score received. Each player icon was moved according to the score received and the final result f the player was shown at the end when the square reached the cell with number 100.

SYSTEM DESIGN

5.1: Front Face(First Windows):



C++ GRAPHICS PROJECT

PRESENTED BY

Mst.MOSTARY KHATUN
IT-20036
ASHIK AHMED SAJIB
IT-20037
TANVIR HASAN
IT-20038

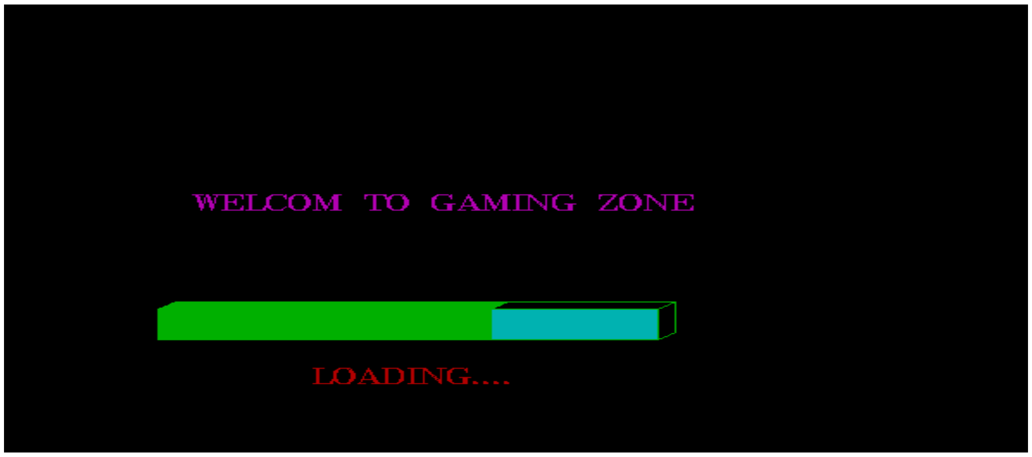
SUPERVISED BY:
Dr.Md.ABIR HOSSAIN
ASSOCIATE PROFESSOR
Dept. Of ICT

1st Year 2nd Semester

Mawlana Bhashani Science And Technology University

Press Any Key To Continue

5.1 Second Windows:



WELCOM TO GAMING ZONE

LOADING....

5.1 Input Design

Input for the Snake & ladder program is dice running from each player.

For player or user interaction the input design are designed with buttons and layout styles.

- **Game Mode**: Selecting player mode for the program there was two boxes allocated for players.

- **Dice Running:** For dice running two buttons are allocated for the players. Players need to hold the button and release when he needs to throw the dice. The received dice will be displayed on the on the dice box.

- **Exit / Stop Program**: The game can be stopped or exit with the click of Exit button.

- **New Game:** New game can be started with “New Game” button.

5.2 Output Design

The output design was created with board design and 100 square cells. The Snake and ladder was also drawn on the bard game for moving and jumping dice icon of each player.

Dice display board output the dice number received after each dice running action by each player. The total score is also displayed left side of the each dice running button.

5.3 Interface Design:

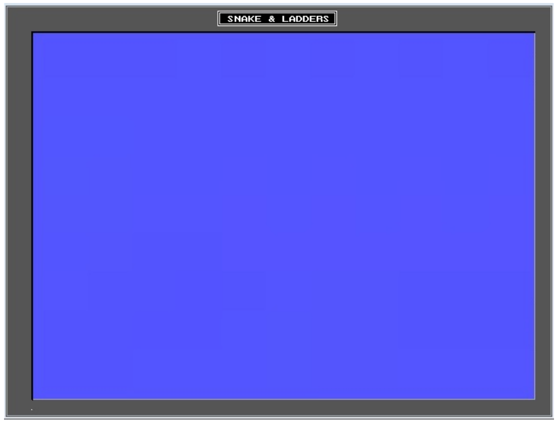
Interface design is the first step in the development phase for an engineering system. It is defined as “The process of applying various techniques and principles for the purpose of defining a process or a system in sufficient details to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used.

DOS operating system was not supported with an powerful GUI, so we tried to explore in giving a good GUI for our Snake & ladder program, which makes a good look and feel appearance for the players.

Game Window Design

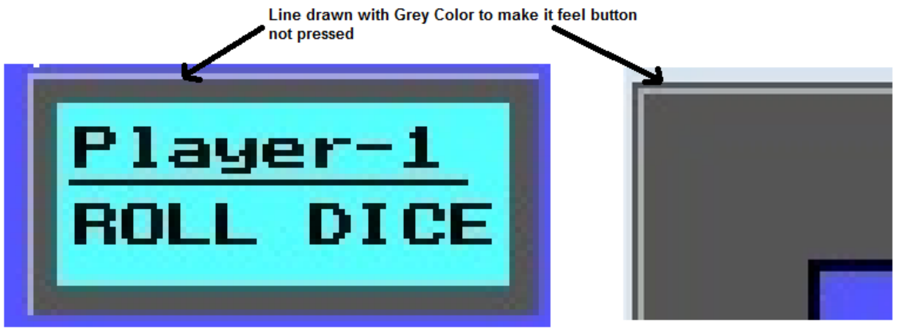
The main window of the game was designed by keeping in mind that the interface style and color used should express the main idea of the functionalities of the program.
The window was first created to make the outer section of game board to make all the controls and other sections inside the blue area.

Main Window



Border and Shading

Border and shading of windows and buttons are created with different lines drawn with grey and black color.


Line drawn with Grey Color to make it feel button not pressed
Player-1
ROLL DICE

Mouse Interaction

If a program that is meant to have an easy-to-use interface does not have mouse support, some would say it would not (and could not) be an easy-to-use interface.

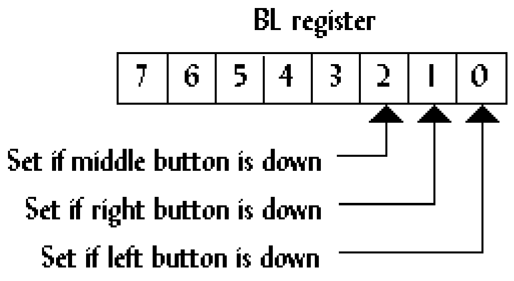
There are two ways to communicate with the mouse:

- with the serial port itself, or
- with the installed mouse driver through interrupt 0x33.

Reading the serial port can be cumbersome because the mouse must be detected, not to mention there are usually two or more serial ports on a computer, each of which could be connected to the mouse.

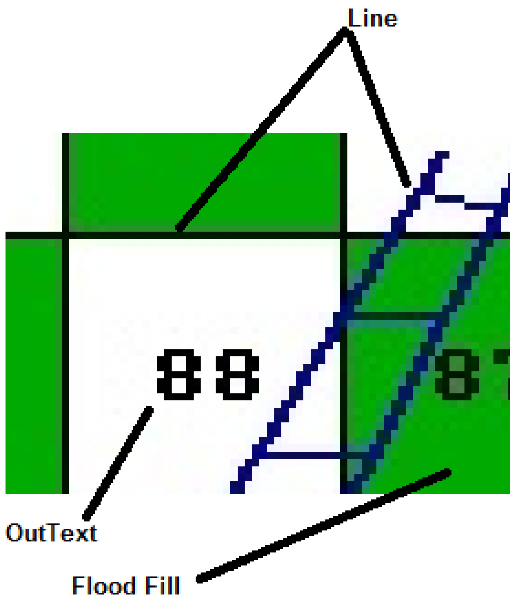
Mouse Register:

To get the mouse's current status, set AX to 3 and call interrupt 0x33. The x value is returned in CX and the y value is returned in DX. BX contains the status of the mouse buttons.

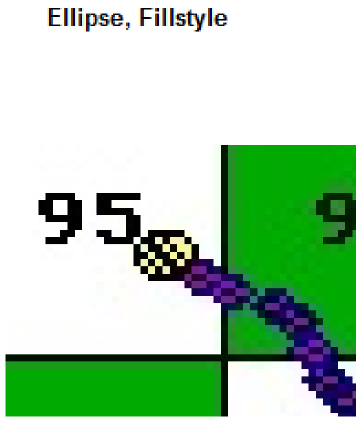

BL register
7
6
5
4
3
2
1
0
Set if middle button is down
Set if right button is down
Set if left button is down

Board

Game board was designed with **Lineto**, **Floodfill** and **Outtext** function. Snake was drawn with continuous **ellipse** function with fill patter style.



Game Board



Snake Head and Body

5.4 Procedural and Functional Design

Buttons are created for user interaction in game window. Below explains each buttons and its uses.

New Game & Exit

	Button is used for taking a new game and clear all dice values and player points. This will also rearrange the player points.
	Used for exiting the program

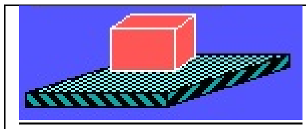
Game Mode

	Button is used for making program to work as a single player mode. Here the other player will be the computer program itself.
	Button is used for making program to work as a dual player mode. Here computer will not be entertained as second player.

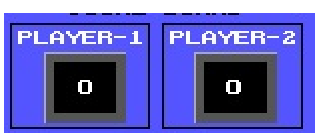
Dice Run Functions

	This can be used in Single and Double play mode. In single mode this button is selected for player. In double mode this button is used as dice running button for Player- 1
	This button is active only when the game mode is Double mode, otherwise the button is deactivated by the computer program. In double mode the Player-2 using this button for dice run.

Dice Value Display

	Display box is used for showing the values received after the

Score Board

	Score board display each players dice values. In Double mode Player-1 score is displayed in the first box and second player score in second boxes. In single mode the computer program score will be shown in the second box.

5.5 Flow Diagram

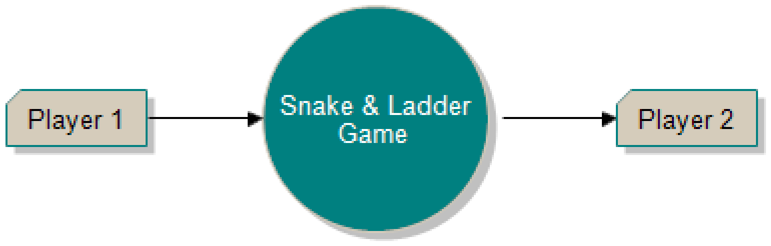
A flow diagram is a graphical representation of the "flow" of data through an information system, modeling its process aspects.

Often they are a preliminary step used to create an overview of the system which can later be elaborated. Flow Diagram can also be used for the visualization of data processing (structured design).We usually begin by drawing a context diagram, a simple representation of the whole system.

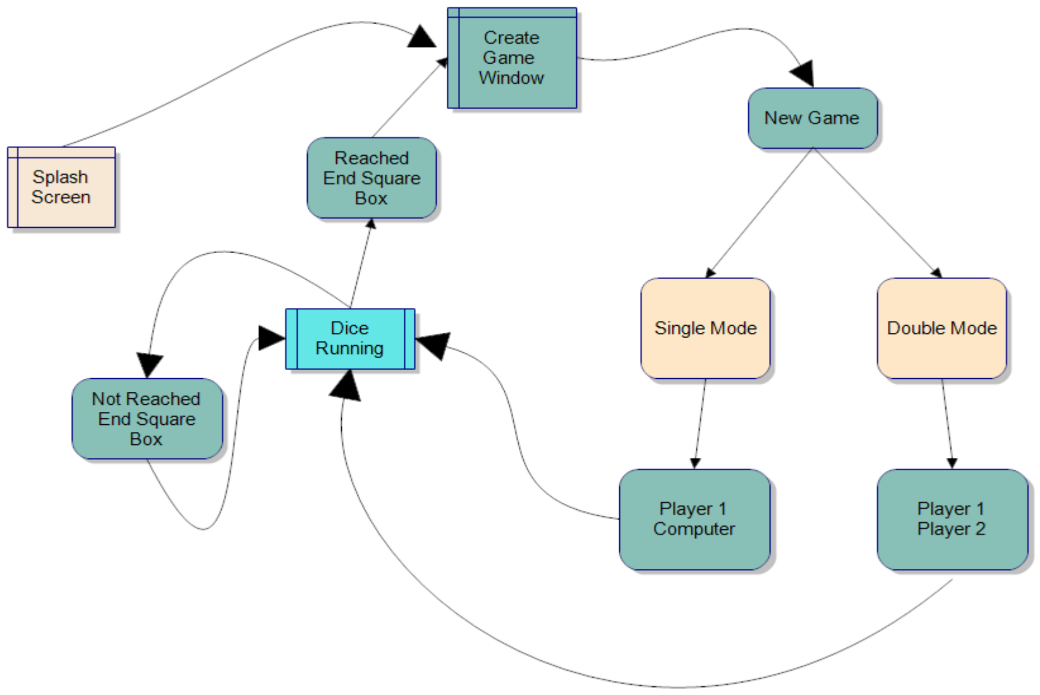
To elaborate further from that, we drill down to a level 1 diagram with additional information about the major functions of the system.

This could continue to evolve to become a level 2 diagram when further analysis is required. Progression to level 3, 4 and so on is possible but anything beyond level 3 is not very common.

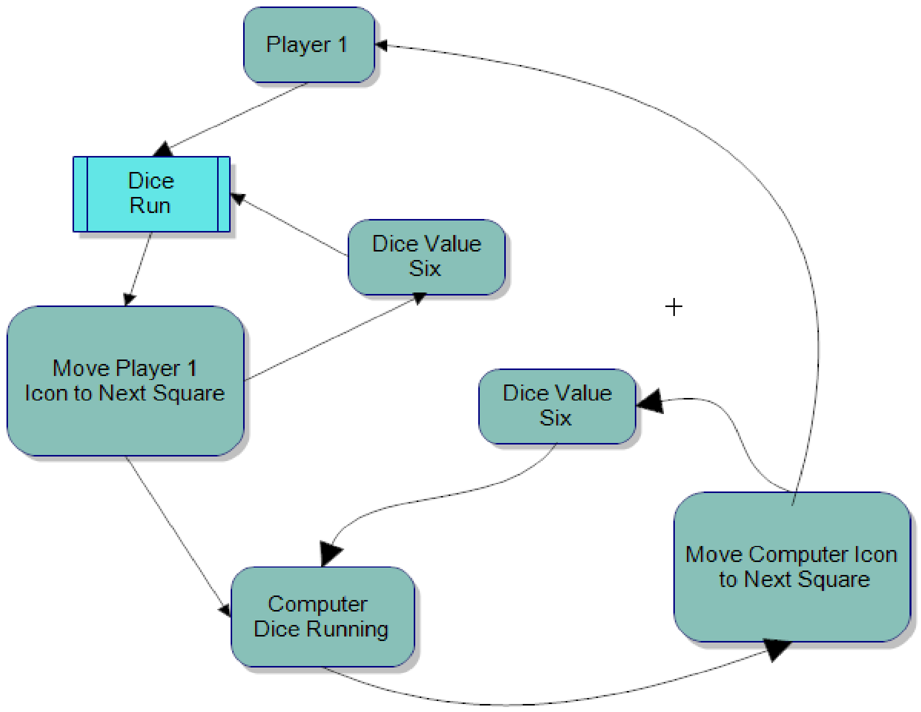
Level – 1



Level-2



Level -3


Player 1
Dice Run
Dice Value Six
Move Player 1 Icon to Next Square
Dice Value Six
Computer Dice Running
Move Computer Icon to Next Square

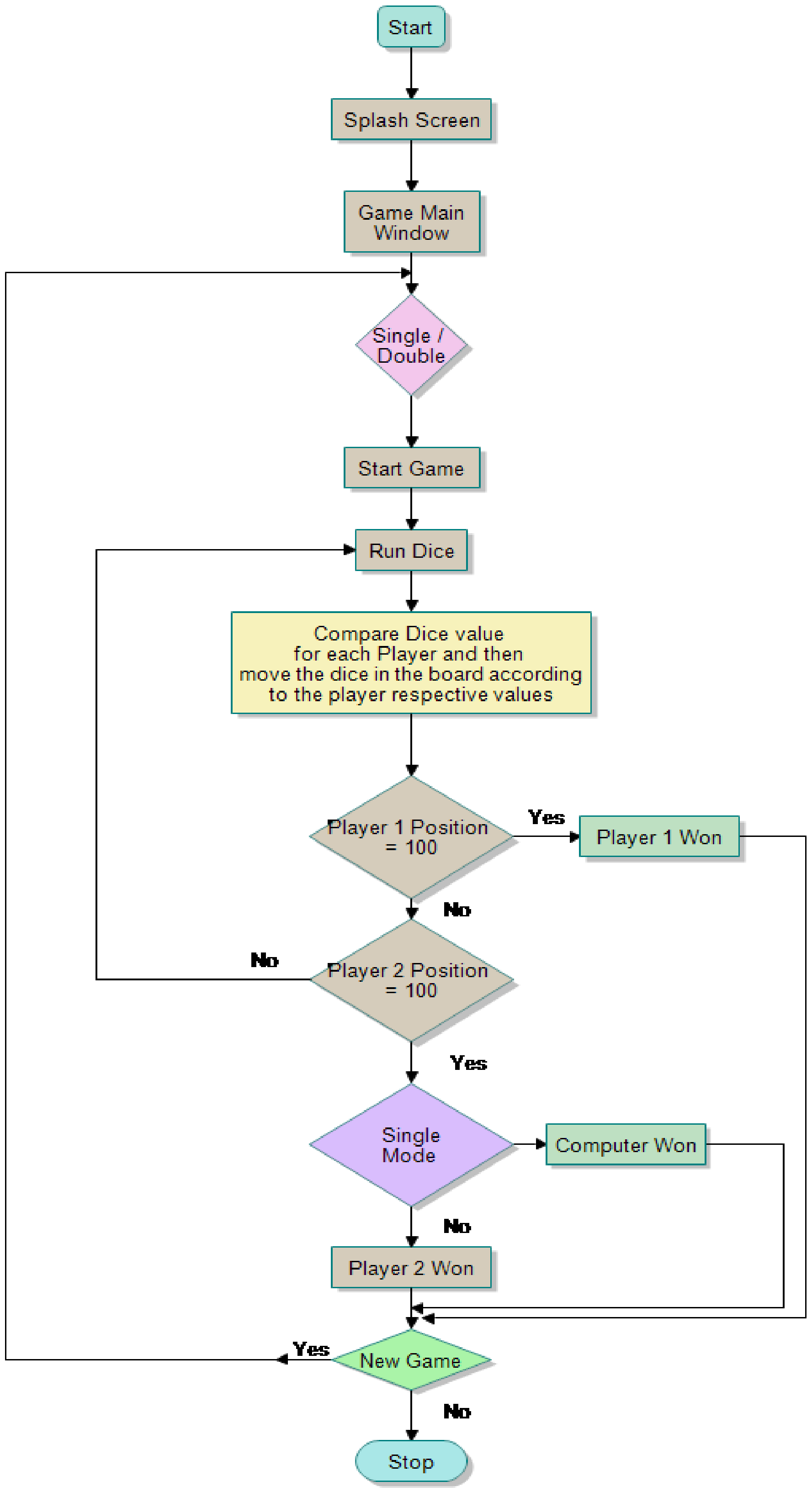
5.3 System Flow Chart

A flowchart (also spelled flow-chart and flow chart) is a schematic representation of an algorithm or a process. A flowchart is one of the seven basic tools of quality control, which include the histogram, Pareto chart, check sheet, control chart, cause-and-effect diagram, flowchart, and scatter diagram. See Quality Management Glossary. They are commonly used in business/economic presentations to help the audience visualize the content better, or to find flaws in the process.

- Start and end symbols: Represented as circles, ovals or rounded (fillet) rectangles, usually containing the word "Start" or "End", or another phrase signaling the start or end of a process, such as "submit inquiry" or "receive product".

- Input/Output, represented as a parallelogram. Examples: Get X from the user; usually containing the word "Start" or "End", or another phrase signaling the start or end of a process, such as "submit enquiry" or "receive product".
- Arrows, showing what's called "flow of display X.
- Generic processing steps: Represented as rectangles.
- Conditional or decision Represented as a diamond (rhombus) showing where a decision is necessary, commonly a Yes/No question or True/False test. The conditional symbol is peculiar in that it has two arrows coming out of it, usually from the bottom point and right point, one corresponding to Yes or True, and one corresponding to No or False. More than two arrows can be used, but this is normally a clear indicator that a complex decision is being taken, in which case it may need to be broken-down further.
- Arrows: Showing "flow of control". An arrow coming from one symbol and ending at another symbol represents that control passes to the symbol the arrow points to.

System Functional Flow Chart



IMPLEMENTATION AND TESTING

5.1 Testing

Testing is probably the most important phase for long-term support as well as for the reputation of the company. If you don't control the quality of the software, it will not be able to compete with other products on the market.

There are multiple types of testing and these are explained in this section. Each of these has its own importance.

Unit Testing

Unit testing is testing one part or one component of the product. The developer usually does this when he/she has completed writing code for that part of the product. This makes sure that the component is doing what it is intended to do. This also saves a lot of time for software testers as well as developers by eliminating many cycles of software being passed back and forth between the developer and the tester. When a developer is confident that a particular part of the software is ready, he/she can write test cases to test functionality of this part of the software.

In our system **Snake & Ladder** each modules are designed and tested separately. Modules designed and tested separately are

- initmouse() : Tested to initialize mouse functions and drivers
- down(int p): Tested to get the position of mouse coordinates
- getmousepos(int *button,int *x,int *y) : Tested to get mouse pointer positions which is read from registers
- createwindo(int x,int y,int w,int z,int col) : Tested to create main window of Snake and Ladder program
- board(): Tested to draw board for playing game

CONCLUSION

6.1 Conclusion

Throughout this thesis our aim was to develop a Snake & Ladder game that allows user to interact with the game using mouse.

The entire game is explained and displayed in a well arranged GUI. Two players are given equal chance for winning the game. Single mode and Double mode allows using computer as second player.

APPENDIX

A: Source Code:

```
#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <graphics.h>

#include <string.h>

#include <process.h>

#include<dos.h>

// path for graphics drivers

#define PATH "..\\bgi"


//

void down(int p);

//To get the position of mouse cordinates

void getpos(int sx,int sy,int b);

//used for drawing window of main board

void windo1();

// ladder position cells

int ladder[5][2]= {{10,28},{17,37},{32,62},{45,84},{78,97}};

// snake position cells

int snake[5][2]= {{34,16},{44,21},{68,47},{79,60},{95,73}};

// for storung dice values single/ double mode

int one_dice=0,prev1_dice=0,prev2_dice=0,sec_dice=0;

// void pointer for memmory allocation

void *d1,*d2;


unsigned int sd1,sd2;

// mouse interrupts functions and libraries

union REGS i,o;

void initmouse() // initinlizing mouse functions and drivers

{
    i.x.ax=0;
    int86(0x33,&i,&o);//raising interrupts
}
void showmouseptr()// interrupts for showing mouse pointer
{
    i.x.ax=1;
    int86(0x33,&i,&o);
}
void hidemouseptr() // interrupts for hiding mouse pointer
{
    i.x.ax=2;
    int86(0x33,&i,&o);
}
//interrupts to get mouse pointer positions which is read from registers
void getmousepos(int *button,int *x,int *y)
{
    i.x.ax=3;
    int86(0x33,&i,&o);

    *button=o.x.bx;
    *x=o.x.cx;
    *y=o.x.dx;
}
// restrict mouse pointer in a rectangle area,,
void restrictmouseptr(int x1,int y1,int x2,int y2)
{
    i.x.ax=7;
    i.x.cx=x1;
    i.x.dx=x2;
    int86(0x33,&i,&o);
    i.x.ax=8;
    i.x.cx=y1;

    i.x.dx=y2;

    int86(0x33,&i,&o);

}



// used for drawing outline for button to make up effect

void up(int x,int y,int w,int z)

{

    setcolor(7);

    line(x,y,w,y);

    line(x,y,x,z);

    setcolor(0);

    line(w,y,w,z);

    line(x,z,w,z);

}

// used for drawing outline for button to make down effect

void dow(int x,int y,int w,int z)

{

    setcolor(0);

    line(x,y,w,y);

    line(x,y,x,z);

    setcolor(7);

    line(w,y,w,z);

    line(x,z,w,z);

}

// function for creating button with color

void bcreate(int x,int y,int w,int z,int col)

{


    setcolor(7);
    rectangle(x,y,w,z);
    setcolor(8);
    setfillstyle(SOLID_FILL,8);
    rectangle(x+1,y+1,w,z);
    floodfill(x+5,y+5,8);
    setfillstyle(SOLID_FILL,col);
    bar(x+5,y+5,w-5,z-5);
}

////////////////////////////
// creating main window of Snake and Ladder program
void createwindo(int x,int y,int w,int z,int col)
{
    setcolor(8);
    rectangle(x,y,w,z);
    setcolor(7);
    setfillstyle(SOLID_FILL,8);
    rectangle(x+1,x+1,w-1,z-1);
    floodfill(x+3,y+3,7);
    setcolor(7);
    setfillstyle(SOLID_FILL,col);
    rectangle(x+30,y+30,w-20,z-20);
    floodfill(x+32,y+32,7);
    setcolor(BLACK);
    line(x+30,y+30,x+30,z-20);
    line(x+31,y+31,x+31,z-19);
    line(x+30,y+30,w-20,y+30);
    line(x+31,y+31,w-20,y+31);
    setfillstyle(SOLID_FILL,8);
    settextstyle(DEFAULT_FONT,0,1);
    setcolor(0);
```

```
rectangle(249,7,386,22);
setcolor(15);
rectangle(250,8,385,21);
rectangle(248,6,387,23);
setfillstyle(1,0);
floodfill(307,13,15);
outtextxy(260,12,"SNAKE & LADDERS ");
settextstyle(SMALL_FONT,0,4);
setcolor(2);

bcreate(530,53,615,85,15);
bcreate(530,90,615,111,15);

bcreate(528,400,614,440,11);
bcreate(528,348,614,388,11);
//mode
bcreate(468,150,562,170,0);
bcreate(468,175,562,195,0);

//SCORE
bcreate(475,290,515,325,0);
bcreate(550,290,590,325,0);

settextstyle(0,0,0);
setcolor(0);
line(526,220,596,220);
line(465,239,535,239);
line(526,220,465,239);
line(596,220,535,239);
rectangle(465,239,535,246);
line(535,246,596,226);
```

```
    line(596,220,596,226);

    setfillstyle(5,3);
    floodfill(473,244,0);
    setfillstyle(4,3);
    floodfill(541,240,0);
    setfillstyle(9,11);
    floodfill(524,230,0);
//dice on board
    setcolor(15);
    setfillstyle(1,12);
    rectangle(517,200,549,221);
    bar(518,201,548,220);
    rectangle(507,207,539,228);
    bar(508,208,538,227);
    line(517,200,507,207);
    line(539,208,549,200);
    line(539,228,549,221);
    floodfill(514,205,15);
    floodfill(541,223,15);
    setcolor(12);
    line(548,221,540,221);
    line(517,201,517,206);
    setcolor(0);



    outtextxy(487,264,"SCORE BOARD");
    outtextxy(472,139,"GAME MODE :");
//       /rectangle
    rectangle(458,275,527,328);
    rectangle(534,275,603,328);
```

```
//icon
setfillstyle(1,4);
bar(509,363,490,373);
setfillstyle(1,14);
bar(509,414,491,424);

setcolor(15);
outtextxy(492,304,"0");
outtextxy(566,304,"0");
outtextxy(492,157,"Single");
outtextxy(462,279,"PLAYER-1");
outtextxy(538,279,"PLAYER-2");

setcolor(7);
outtextxy(492,182,"Double");



setcolor(0);
outtextxy(536,357,"Player-1");
outtextxy(536,370,"ROLL DICE");
outtextxy(536,409,"Player-2");
outtextxy(536,422,"ROLL DICE");
outtextxy(541,65,"NEW GAME");
outtextxy(537,98,"EXIT GAME");

line(535,366,602,366);
line(535,418,602,418);
setcolor(15);
settextstyle(SMALL_FONT,0,4);
outtextxy(30,462,"Project Team : Tanvir, Mostary,Ashik..");
```

```
}

// drawing board for playing game
void board()
{
    int i,j,x=73,y=76;
    char cc[3],num=100;
    int m=2;

    setcolor(15);
    setfillstyle(1,2);
    bar(61,61,439,439);
    setcolor(0);
    rectangle(60,60,440,440);
    line(98,60,98,440);
    line(136,60,136,440);
    line(174,60,174,440);
    line(212,60,212,440);
    line(250,60,250,440);
    line(288,60,288,440);
    line(326,60,326,440);
    line(364,60,364,440);
    line(402,60,402,440);

    line(60,98,440,98);
    line(60,136,440,136);
    line(60,174,440,174);
    line(60,212,440,212);
    line(60,250,440,250);
    line(60,288,440,288);
    line(60,326,440,326);
    line(60,364,440,364);
    line(60,402,440,402);
    settextstyle(0,0,0);
    setcolor(0);
    setfillstyle(1,15);
    x=73;
    y=76;
    num=100;
    m=2;
// for printing cell number and white cells

    for(i=0; i<10; i++)
    {

        for(j=0; j<10; j++)
        {

            if(m==2)
            {
                if(j%2)
                    floodfill(x,y,0);

            }
            else
            {
                if(!(j%2))
                    floodfill(x,y,0);
            }

// printing number in graphics mode with  formatted output
```

```
            sprintf(cc,"%d",num);


            num--;
            outtextxy(x,y,cc);
            x+=38;
        }
        if(m==3)
            m=2;
        else


            m=3;
        x=73;
        y+=38;

    }
    setcolor(1);

//ladder 10->28
    line(70,414,147,335);
    line(71,414,148,335);
    line(86,427,165,344);
    line(87,427,166,344);

    line(80,405,95,419);
    line(80,404,95,418);
    line(100,384,116,396);
    line(100,383,116,395);
    line(118,366,135,376);
    line(118,367,135,377);
    line(139,344,155,355);
    line(139,345,155,356);
```

```
//ladder 17->37
    line(181,385,186,312);
    line(182,385,187,312);
    line(209,385,198,314);
    line(210,385,199,314);

    line(182,377,208,377);
    line(183,354,205,353);
    line(185,333,202,332);
    line(186,319,199,318);

//ladder 32->62
    line(369,300,384,194);
    line(370,300,385,194);
    line(385,302,397,198);
    line(386,302,398,198);

    line(370,295,387,295);
    line(373,276,388,279);
    line(376,256,391,260);
    line(377,241,392,245);
    line(381,222,395,226);
    line(383,207,396,209);

//ladder 45->84

    line(259,261,303,122);
    line(258,261,302,122);
    line(273,261,313,123);
    line(272,261,312,123);
    line(262,254,275,254);
    line(266,237,280,238);
    line(272,218,285,221);
    line(277,202,289,204);
    line(283,185,295,187);
    line(288,168,300,167);
    line(294,148,305,149);
    line(299,131,310,131);

//ladder 78->97

    line(150,147,187,87);
    line(151,147,188,87);
    line(165,152,199,87);
    line(166,152,200,87);

    line(153,141,170,144);
    line(163,128,178,128);
    line(174,109,188,109);
    line(185,92,196,95);

///////snakes
    setfillstyle(9,5);
    setcolor(1);
    fillellipse(242,382,2,2);
    fillellipse(246,379,2,2);
    fillellipse(247,379,2,2);
    fillellipse(249,378,2,2);
    fillellipse(252,375,2,2);
    fillellipse(251,370,2,2);
    fillellipse(250,365,2,2);
    fillellipse(248,354,3,2);
    fillellipse(248,362,3,2);
    fillellipse(245,358,3,2);
    fillellipse(252,351,3,2);
    fillellipse(261,351,3,2);
    fillellipse(266,353,3,2);
    fillellipse(284,348,3,2);
    fillellipse(288,340,3,2);
    fillellipse(275,351,3,2);
    fillellipse(285,336,3,2);
    fillellipse(283,329,3,2);
    fillellipse(284,319,3,2);
    fillellipse(287,317,3,2);
    fillellipse(257,350,3,2);
    fillellipse(270,352,3,2);
    fillellipse(281,350,3,2);
    fillellipse(286,345,2,2);
    fillellipse(285,333,2,2);
    fillellipse(283,323,2,2);
    fillellipse(291,316,2,2);
    fillellipse(295,316,2,2);

//head fillellipse(299,313,3,2);

//snake 21->
    fillellipse(420,358,3,2);
    fillellipse(415,357,3,2);
    fillellipse(404,350,3,2);
    fillellipse(398,343,3,2);
    fillellipse(392,335,3,2);
    fillellipse(374,319,3,2);
    fillellipse(362,315,3,2);
    fillellipse(346,317,3,2);
    fillellipse(333,313,3,2);
    fillellipse(325,304,3,2);
    fillellipse(319,293,3,2);
    fillellipse(315,284,3,2);
    fillellipse(316,289,3,2);
    fillellipse(315,280,3,2);
    fillellipse(321,298,3,2);
    fillellipse(327,308,3,2);
    fillellipse(323,301,3,2);

    fillellipse(330,311,3,2);
    fillellipse(337,315,3,2);
    fillellipse(341,317,3,2);
    fillellipse(352,317,3,2);
    fillellipse(357,316,3,2);
    fillellipse(368,313,3,2);
    fillellipse(373,315,3,2);
    fillellipse(374,323,3,2);
    fillellipse(374,326,3,2);
    fillellipse(375,330,3,2);
    fillellipse(378,334,3,2);
    fillellipse(383,334,3,2);
    fillellipse(388,335,3,2);
    fillellipse(393,337,3,2);
    fillellipse(397,339,3,2);
    fillellipse(399,348,3,2);
    fillellipse(408,352,3,2);
    fillellipse(411,354,3,2);
```

```
//snake 47
    fillellipse(155,205,3,2);
    fillellipse(158,207,3,2);
    fillellipse(162,209,3,2);
    fillellipse(170,227,3,2);
    fillellipse(169,219,3,2);
    fillellipse(169,236,3,2);
    fillellipse(169,247,3,2);
    fillellipse(173,257,3,2);
    fillellipse(181,264,3,2);
    fillellipse(175,260,3,2);
    fillellipse(165,212,3,2);
    fillellipse(168,216,3,2);
    fillellipse(171,222,3,2);
    fillellipse(170,231,3,2);
    fillellipse(170,241,3,2);
    fillellipse(172,252,3,2);
    fillellipse(178,262,3,2);
    fillellipse(184,266,3,2);
//snake 60

    fillellipse(86,225,3,2);
    fillellipse(90,221,3,2);
    fillellipse(95,217,3,2);
    fillellipse(101,212,3,2);
    fillellipse(105,204,3,2);
    fillellipse(106,196,3,2);
    fillellipse(106,189,3,2);
    fillellipse(106,179,3,2);
    fillellipse(109,170,3,2);
    fillellipse(112,164,3,2);
    fillellipse(109,167,3,2);
    fillellipse(107,174,3,2);
    fillellipse(106,184,3,2);
    fillellipse(106,194,3,2);
    fillellipse(105,200,3,2);
    fillellipse(103,208,3,2);
    fillellipse(98,215,3,2);
    fillellipse(92,219,3,2);
    fillellipse(88,222,3,2);
//snake 73->94
    fillellipse(285,87,3,2);
    fillellipse(289,89,3,2);
    fillellipse(295,91,3,2);
    fillellipse(300,97,3,2);
    fillellipse(305,104,3,2);
    fillellipse(311,109,3,2);
    fillellipse(320,113,3,2);
    fillellipse(330,117,3,2);
    fillellipse(338,122,3,2);
    fillellipse(342,127,3,2);
    fillellipse(345,132,3,2);
    fillellipse(347,138,3,2);
    fillellipse(350,142,3,2);
    fillellipse(350,145,3,2);
    fillellipse(298,93,3,2);
    fillellipse(302,101,3,2);
    fillellipse(307,107,3,2);
    fillellipse(315,113,3,2);
    fillellipse(325,112,3,2);
    fillellipse(334,119,3,2);
    fillellipse(340,125,3,2);
    fillellipse(328,114,3,2);
//heads...
    setfillstyle(6,14);
    setcolor(0);
    fillellipse(299,312,4,3);
    fillellipse(314,276,4,3);
    fillellipse(152,202,4,3);
    fillellipse(116,161,4,3);
    fillellipse(280,84,4,3);
//player button






}

// splash screen
void splash()
{
    int j,t,sx,sy,b,nex,star;
    char dd[15];
    setfillstyle(1,15);
    bar(105,105,getmaxx()-125,getmaxy()-105);
    setfillstyle(1,10);
    bar(105,105,195,getmaxy()-105);
    setcolor(0);
    rectangle(106,106,getmaxx()-126,getmaxy()-106);
    setfillstyle(1,12);
    setcolor(0);
    rectangle(245,145,290,190);
    rectangle(266,166,310,210);
    line(245,145,266,166);
    line(245,190,266,210);
    line(290,145,310,166);
    floodfill(261,196,0);
    floodfill(296,159,0);
    bar(245,145,290,190);
    bar(266,166,310,210);
// drawing dice
    rectangle(266,166,310,210);
    line(245,145,266,166);
    line(245,190,266,210);
    line(290,145,310,166);
    line(245,145,290,145);
    line(245,145,245,190);
    line(310,210,331,182);
    line(331,182,310,182);
    setfillstyle(1,7);
    setcolor(7);
    floodfill(315,195,0);
    line(310,210,331,182);
    line(331,182,311,182);

    setfillstyle(5,14);
    fillellipse(275,259,2,2);
    setcolor(12);
    setfillstyle(1,15);
    fillellipse(275,176,4,4);
    fillellipse(299,176,4,4);
    fillellipse(275,199,4,4);
    fillellipse(299,199,4,4);
    fillellipse(287,187,4,4);

    fillellipse(252,160,3,4);

    fillellipse(261,197,3,4);

    fillellipse(256,179,3,4);

    fillellipse(262,150,4,3);

    fillellipse(284,150,4,3);

    fillellipse(273,160,4,3);

    fillellipse(296,160,4,3);


// drawing big snake body


    setcolor(0);


    moveto(338,239);

    lineto(325,238);

    lineto(314,233);

    lineto(301,229);

    lineto(293,227);

    lineto(285,229);

    lineto(277,234);

    lineto(275,237);

    lineto(278,245);

    lineto(270,249);

    lineto(259,250);

    lineto(254,252);

    lineto(253,258);

    lineto(255,264);

    lineto(260,268);

    lineto(264,272);

    lineto(270,274);

    lineto(280,276);
    lineto(292,275);
    lineto(304,272);
    lineto(316,270);
    lineto(322,271);
    lineto(325,275);
    lineto(323,283);
    lineto(320,290);
    lineto(307,314);
    lineto(299,321);
    lineto(295,322);
    lineto(294,321);
    lineto(301,313);
    lineto(306,301);
    moveto(308,306);
    lineto(304,300);
    lineto(295,297);
    lineto(283,298);
    lineto(268,304);
    lineto(272,300);
    lineto(286,294);
    lineto(300,292);
    lineto(309,289);
    lineto(314,281);
    lineto(319,275);
    moveto(298,273);
    lineto(301,276);
    lineto(309,276);
    lineto(312,278);
    lineto(308,285);
    lineto(306,290);
    moveto(302,299);
```

```
lineto(301,308);

lineto(289,322);

lineto(291,327);

lineto(303,326);

lineto(313,316);

lineto(321,301);

lineto(329,287);

lineto(338,281);

moveto(284,276);

lineto(274,273);

lineto(268,270);

lineto(264,264);

lineto(261,260);


lineto(257,259);

lineto(253,259);

moveto(282,248);

lineto(287,251);

lineto(294,255);

lineto(304,258);

lineto(310,255);

lineto(315,249);

lineto(315,245);

lineto(310,241);

lineto(304,241);

lineto(298,245);

lineto(294,247);

lineto(288,249);

lineto(282,247);


ellipse(304,248,0,360,4,4);
```

```
moveto(321,244);
lineto(314,237);
lineto(307,234);
lineto(301,236);
lineto(294,241);
lineto(288,244);
moveto(280,262);
lineto(281,257);
lineto(278,254);
lineto(273,254);
lineto(270,257);
ellipse(275,259,0,360,3,3);
moveto(297,230);
lineto(290,231);
lineto(284,233);
lineto(279,237);
lineto(282,242);
moveto(273,276);
lineto(277,284);
lineto(284,290);
lineto(280,281);
lineto(281,276);
moveto(263,272);
lineto(264,280);
lineto(270,288);
lineto(267,278);
lineto(268,274);
moveto(329,287);
lineto(332,280);
lineto(336,272);
lineto(333,267);
```

```
moveto(338,267);
lineto(379,265);
lineto(415,265);
lineto(442,260);
lineto(459,255);
lineto(478,247);
lineto(483,239);
moveto(334,284);
lineto(388,279);
lineto(457,265);
lineto(479,254);
lineto(485,240);
lineto(486,219);
lineto(480,206);
lineto(451,175);
lineto(410,155);
lineto(381,142);
lineto(380,133);
lineto(392,128);
lineto(421,119);
lineto(446,111);
lineto(467,107);
moveto(335,239);
lineto(361,240);
lineto(404,242);
lineto(425,241);
lineto(438,239);
lineto(447,235);
lineto(455,229);
lineto(457,225);
lineto(456,221);
lineto(449,212);
lineto(409,186);
lineto(369,168);
lineto(353,150);
lineto(355,128);
lineto(364,118);
lineto(381,109);
lineto(388,106);
moveto(334,239);
lineto(340,244);
lineto(340,253);
lineto(388,253);
lineto(387,247);
lineto(382,241);
moveto(407,242);
lineto(413,247);
lineto(416,255);
lineto(442,250);
lineto(460,242);
lineto(474,232);
lineto(479,220);
lineto(475,207);
lineto(466,196);
lineto(458,197);
lineto(451,202);
lineto(458,209);
lineto(461,216);
lineto(459,224);
lineto(452,231);
moveto(456,181);
lineto(449,184);
```

```
    lineto(445,191);
    lineto(416,171);
    lineto(397,164);
    lineto(398,157);
    lineto(408,154);
    moveto(403,107);
    lineto(406,112);
    lineto(402,117);
    lineto(383,119);
    lineto(373,125);
    lineto(370,133);
    lineto(372,143);
    lineto(382,151);
    lineto(375,153);
    lineto(371,158);
    lineto(360,151);
    lineto(357,140);
    lineto(359,124);
// filling color for snke after drawig outline....



    setfillstyle(9,2);
    floodfill(398,254,0);

    setfillstyle(1,4);
    floodfill(314,290,0);

    setfillstyle(9,14);
    floodfill(366,248,0);
    floodfill(439,245,0);
    floodfill(420,166,0);
    floodfill(365,133,0);

    setfillstyle(6,0);
    floodfill(262,266,0);

    setfillstyle(1,9);
    floodfill(297,254,0);

    setfillstyle(1,12);
    floodfill(304,248,0);
    setcolor(0);
    setfillstyle(1,15);
    fillellipse(304,248,2,2);

    settextstyle(0,0,2);
    setcolor(2);
    outtextxy(116,129,"S");
    outtextxy(116,149,"N");
    outtextxy(116,169,"A");
    outtextxy(116,189,"K");
    outtextxy(116,209,"E");
    outtextxy(116,229,"S");

    outtextxy(141,217,"&");
    outtextxy(169,221,"L");
    outtextxy(169,241,"A");
    outtextxy(169,261,"D");
    outtextxy(169,281,"D");
    outtextxy(169,301,"E");
    outtextxy(169,321,"R");
    outtextxy(169,341,"S");
```

```
setcolor(0);
outtextxy(114,128,"S");
outtextxy(114,148,"N");
outtextxy(114,168,"A");
outtextxy(114,188,"K");
outtextxy(114,208,"E");
outtextxy(114,228,"S");

outtextxy(140,215,"&");
outtextxy(167,220,"L");
outtextxy(167,240,"A");
outtextxy(167,260,"D");
outtextxy(167,280,"D");
outtextxy(167,300,"E");
outtextxy(167,320,"R");
outtextxy(167,340,"S");
settextstyle(0,0,0);
setcolor(7);
outtextxy(203,137,"THE UPS AND DOWN");
outtextxy(259,218,"OF VIRTUAL LIFE..");

setcolor(0);
outtextxy(201,135,"THE UPS AND DOWN");
outtextxy(257,216,"OF VIRTUAL LIFE..");

settextstyle(2,0,3);
settextstyle(2,0,0);
setcolor(0);
outtextxy(138,122,"'Dr.Md.Abir Hossain'");
setcolor(8);
outtextxy(108,352,"Project Team:   Tanvir, Mostary,Ashik");
```

```
    setcolor(0);

    line(138,134,193,134);

    line(138,136,193,136);

    outtextxy(119,110,"Idea from");

    outtextxy(187,344,"1.0");

    setcolor(1);

    setfillstyle(1,9);

//ladder


    bar(392,127,400,241);

    bar(392,278,400,359);


    bar(448,110,456,229);

    bar(448,266,456,350);


    bar(397,137,448,142);

    bar(397,176,448,181);

    bar(397,216,448,221);

    bar(397,286,448,291);

    bar(397,337,448,342);


    setfillstyle(1,1);

    bar(392,127,398,241);

    bar(392,278,398,361);


    bar(448,110,454,231);

    bar(448,266,454,352);


    bar(397,137,448,140);

    bar(397,176,448,179);

    bar(397,216,448,219);
    bar(397,286,448,289);
    bar(397,337,448,340);

    setfillstyle(6,10);
    bar(337,269,348,280);
    bar(351,269,359,280);
    bar(362,269,370,278);
    bar(373,268,380,277);
    bar(383,268,391,276);
    bar(393,267,400,275);
    bar(403,267,410,272);
    bar(413,267,421,271);
    bar(424,266,430,269);
    bar(433,264,441,266);
    bar(443,262,451,264);
    bar(452,259,460,262);
    bar(462,256,467,258);
    bar(468,253,474,255);
    bar(476,248,480,251);
    setcolor(0);
    rectangle(223,364,380,369);
    setfillstyle(1,0);
//progress bar
    nex=224;
    star=2;
    for(j=10,t=10; j<49;)
    {
        t++;
        j++;
        bar(nex,365,nex+star,368);
        nex+=star;
        nex+=2;
        delay(190);

    }
}




// move dice in board to next cell
void move(int x,int y,int sel)
{
    if(sel==1)
    {
        setfillstyle(1,4);
        setcolor(14);
        rectangle(x-1,y-1,x+10,y+10);

    }
    else
    {
        setfillstyle(1,14);
        setcolor(12);
        rectangle(x-1,y-1,x+10,y+10);

    }
    bar(x,y,x+9,y+9);

}

//change position of dice in board
void changeposition(int num,int player)
```

```
{
    num--;
    if(player==1)
        move(417-(num%10*38),427-(num/10*38),1);
    if(player==2)
        move(417-(num%10*38),405-(num/10*38),2);
}

// save image portion before any paint function
void saveimageportion(int num,int player)
{



    if(player==1)
    {
        sd1=imagesize(416-(num%10*38),426-(num/10*38),(417-
(num%10*38))+10,(427-(num/10*38))+10);
        d1 = malloc(sd1);
        getimage(416-(num%10*38),426-(num/10*38),(417-(num%10*38))+10,(427-
(num/10*38))+10,d1);
    }

    if(player==2)
    {

        sd2=imagesize(416-(num%10*38),404-(num/10*38),(417-
(num%10*38))+10,(407-(num/10*38))+10);
        d2 = malloc(sd2);
        getimage(416-(num%10*38),404-(num/10*38),(417-(num%10*38))+10,(407-
(num/10*38))+10,d2);


    }
```

```
}

// restore save image to correct location
void putimageportion(int num,int player)
{

    if(player==1)
    {
        putimage(416-(num%10*38),426-(num/10*38),d1,0);
    }

    if(player==2)
    {
        putimage(416-(num%10*38),404-(num/10*38),d2,0);
    }


}

// drawing dice after playing
void dicerun(int num)
{
    setcolor(12);
    setfillstyle(1,12);
    bar(508,209,537,226);
    setcolor(15);
    setfillstyle(1,15);

    if(num==1)
        fillellipse(522,217,3,1);
```

```
if(num==2)
{
    fillellipse(511,211,3,1);
    fillellipse(533,224,3,1);
}

if(num==3)
{
    fillellipse(511,211,3,1);
    fillellipse(533,224,3,1);
    fillellipse(522,217,3,1);

}
if(num==4)
{
    fillellipse(511,211,3,1);
    fillellipse(511,224,3,1);
    fillellipse(533,211,3,1);
    fillellipse(533,224,3,1);

}
if(num==5)
{
    fillellipse(511,211,3,1);
    fillellipse(511,224,3,1);
    fillellipse(533,211,3,1);
    fillellipse(533,224,3,1);
    fillellipse(522,217,3,1);

}
if(num==6)
```

```
    {
        fillellipse(511,211,3,1);
        fillellipse(511,217,3,1);
        fillellipse(511,224,3,1);
        fillellipse(533,211,3,1);
        fillellipse(533,217,3,1);
        fillellipse(533,224,3,1);
    }

}



void loading ()
{
    cleardevice();

    settextstyle(1,0,3);
    setcolor(5);
    outtextxy(130,200,"WELCOM TO GAMING ZONE");
    setcolor(4);
    outtextxy(200,400-30,"LOADING....");
    setcolor(GREEN);
    for(int i=0; i<290; ++i)
    {
        setfillstyle(1,3);
        bar3d(110+i,350-30,400,380-30,10,1);
        delay(10);
    }
    cleardevice();
```

```
}

void front_page()
{


    setcolor(RED);
    setfillstyle(1,0);
    bar3d(70,45,500,90,10,1);
    setcolor(3);
    settextstyle(1,0,3);
    int yy=30;
    outtextxy(200,140-yy,"PRESENTED BY ");
    outtextxy(120,52,"   C++ GRAPHICS PROJECT  ");
    setcolor(GREEN);
    settextstyle(1,0,2);
    outtextxy(20,170,"Mst.MOSTARY KHATUN");
    outtextxy(20,200,"IT-20036");
    outtextxy(20,230,"ASHIK AHMED SAJIB");
     outtextxy(20,260,"IT-20037");
    outtextxy(20,290,"TANVIR HASAN");
     outtextxy(20,320,"IT-20038");
    outtextxy(180,340,"1st Year 2nd Semester");

    //outtextxy(350,170,"TANVIR HASAN");
    //outtextxy(350,200,"IT-20038");
    //outtextxy(350,230,"1st Year 2nd Semester");

    setcolor(CYAN);
    outtextxy(300,170,"SUPERVISED BY:");
    setcolor(GREEN);
    outtextxy(300,200,"Dr.Md.ABIR HOSSAIN");
    outtextxy(300,230,"ASSOCIATE  PROFESSOR");
    outtextxy(300,260,"Dept. Of ICT");

    setcolor(CYAN);
    settextstyle(1,0,3);

    //----------------
    outtextxy(10,400,"Mawlana Bhashani Science And Technology University");
    setcolor(3);
    setcolor(RED);
    outtextxy(150,440,"Press Any Key To Continue");

    getch();

}

// main function...

void main()
{
    // detect graphics driver and graphics mode....
    int gd=DETECT,gm;
    int i,j,b,sx,sy,z,col,x1,y1,w1,z1;
    char dd[20],dc[5];
    int num;
    int dice=0,k,l,chance=1,selemode=1,end=0;
    int one_dice=0,prev1_dice=0,prev2_dice=0,sec_dice=0;

    void *d1,*d2,*result;
    unsigned int sd1,sd2,resarea;
```

```
// initilizing mouse functions
initmouse();
// these libraries for making stand alon program by inserting
// graphics libraries,, so no need to put the BGI path in the program
//The program will work without graphics driver in the system..
// for all steps refere document inside TC\DOC

/*registerfarbgidriver(EGAVGA_driver_far);
  registerfarbgifont(sansserif_font_far);
  registerfarbgifont(triplex_font_far);
  registerfarbgifont(small_font_far);
  initgraph(&gd,&gm,NULL);
*/
// initilizing graphics driver
initgraph(&gd,&gm,PATH);

front_page();
loading();

// getting maximum x and y size of active or current display mode
w1=getmaxx(),z1=getmaxy();
x1=0;
y1=0;

// drawing splash screen
splash();
// clearing display device and its memory for main window drawing
cleardevice();
// drawing snake and ladder game outline window
createwindo(x1,y1,w1,z1,9);
// drawing game board cells for playing
```

```
board();
// showing mouse pointer after drwaing all graphics windows
showmouseptr();

/* main loop and only one infinite loop for the program to run every time
 for getting input from the devices.. this loop will be exited
 when the "Exit" button is clicked.
 */

while(1)
{



    // this was used to get the mouse position every time to
    // identify the current position of mouse for each functions
    // mouse cordinates was stored in sx and sy variables..

    getmousepos(&b,&sx,&sy);
    // used to give action for single nd double button clicked actions
    // and selemode variable is set
    // single

    // b variable is used for checking buton status..
    // b=1 left button clicked
    if(b==1&&sx>468&&sy>150&&sx<562&&sy<170)
    {
        settextstyle(0,0,0);
        hidemouseptr();
        setcolor(7);
        outtextxy(492,182,"Double");
        setcolor(15);
```

```
            outtextxy(492,157,"Single");
            showmouseptr();
            selemode=1;
            end=0;
            one_dice=0;
            sec_dice=0;
            putimageportion(prev1_dice,1);
            putimageportion(prev2_dice,2);
            chance=1;
            setcolor(15);
            rectangle(528,348,614,388);
            setcolor(9);
            rectangle(528,400,614,440);
            setcolor(0);
            outtextxy(484,304,"ллл");
            outtextxy(556,304,"ллл");
            setcolor(15);
            outtextxy(484,304,"0");
            outtextxy(558,304,"0");
            setcolor(12);
            setfillstyle(1,12);
            bar(508,209,537,226);
        }
// double
        if(b==1&&sx>468&&sy>175&&sx<562&&sy<195)
        {
            hidemouseptr();
            setcolor(15);
            outtextxy(492,182,"Double");
            setcolor(7);
            outtextxy(492,157,"Single");
```

```
    showmouseptr();
    selemode=2;
    end=0;
    one_dice=0;
    sec_dice=0;
    putimageportion(prev1_dice,1);
    putimageportion(prev2_dice,2);
    chance=1;
    setcolor(15);
    rectangle(528,348,614,388);
    setcolor(9);
    rectangle(528,400,614,440);
    setcolor(0);
    outtextxy(484,304,"ллл");
    outtextxy(556,304,"ллл");
    setcolor(15);
    outtextxy(484,304,"0");
    outtextxy(558,304,"0");
    setcolor(12);
    setfillstyle(1,12);
    bar(508,209,537,226);
}

if(b==1&&sx>530&&sy>53&&sx<615&&sy<85)
{
    settextstyle(0,0,0);
    end=0;
    one_dice=0;
    sec_dice=0;
    putimageportion(prev1_dice,1);
    putimageportion(prev2_dice,2);
```

```
            chance=1;
            setcolor(15);
            rectangle(528,348,614,388);
            setcolor(9);
            rectangle(528,400,614,440);
            setcolor(0);
            outtextxy(484,304,"ллл");
            outtextxy(556,304,"ллл");
            setcolor(15);
            outtextxy(484,304,"0");
            outtextxy(558,304,"0");
            setcolor(12);
            setfillstyle(1,12);
            bar(508,209,537,226);

        }

        if(b==1)
        {
////////////////////////////////////////////////////////

            if(end==0&&chance==1&&b==1&&sx>528&&sy>348&&sx<614&&sy<388)
            {

                //  player 1   bcreate(528,348,614,388,11);

                //loop runs contnously when mouse button was pressed and stops
                //when mouse button is released..

                dice=0;
                while(b==1)
                {
                    getmousepos(&b,&sx,&sy);
                    dice++;
                    if(dice==7)
                        dice=1;
                }
                dicerun(dice);
                // hiding mouse and waiting for one more chance / click when dice
= 6


                hidemouseptr();
                if(dice==6)
                {
                    setcolor(15);
                    rectangle(528,348,614,388);
                    setcolor(9);
                    rectangle(528,400,614,440);
                    chance=1;
                }
                else
                {
                    setcolor(9);
                    rectangle(528,348,614,388);
                    setcolor(15);
                    rectangle(528,400,614,440);
                    chance=2;
                }
                // show mouse pointer
                showmouseptr();
                one_dice+=dice;
```

```
for(i=0; i<5; i++)
    for(j=0; j<1; j++)
    {
        if(ladder[i][j]==one_dice)
            one_dice=ladder[i][1];
        if(snake[i][j]==one_dice)
            one_dice=snake[i][1];

    }
sound(400);
delay(500);
nosound();

if(one_dice<=100)
{
    num=one_dice;
    num--;
    putimageportion(prev1_dice,1);
    saveimageportion(num,1);
    changeposition(one_dice,1);
    prev1_dice=one_dice-1;
}
else
{
    one_dice-=dice;
}

sprintf(dc,"%d",one_dice);

setcolor(0);
outtextxy(484,304,"ллл");
```

```
setcolor(15);
outtextxy(485,304,dc);



if(one_dice==100)
{
    end=1;


    hidemouseptr();
    resarea=imagesize(190,170,435,320);
    result=malloc(resarea);
    getimage(190,170,435,320,result);


    setfillstyle(1,0);
    setcolor(15);
    rectangle(190,170,435,320);
    bar(191,171,434,319);
    outtextxy(258,242,"PLAYER 1 WON");
    rectangle(369,272,415,280);
    rectangle(378,261,404,271);
    line(378,261,384,255);
    line(384,255,377,209);
    line(405,209,399,255);
    line(399,255,404,261);
    ellipse(391,209,0,360,14,3);
    setfillstyle(1,8);
    floodfill(391,234,15);
    setfillstyle(1,4);
    floodfill(386,267,15);
    setfillstyle(1,1);
    floodfill(386,276,15);
```

```
                        setfillstyle(1,7);
                        floodfill(391,208,15);
                        settextstyle(2,0,0);
                        setcolor(14);
                        outtextxy(280,303,"press any key to continue..");
                        getch();
                        putimage(190,170,result,0);
                        free(result);
                        showmouseptr();
                    }
                    settextstyle(0,0,0);
                }

//////////////////////////////////////////////////////////////
// double mode for two player

if(selemode==2&&end==0&&chance==2&&b==1&&sx>528&&sy>400&&sx<614&&sy<440)
                {

                    dice=0;
                    while(b==1)
                    {
                        getmousepos(&b,&sx,&sy);
                        dice++;
                        if(dice==7)
                            dice=1;
                    }
                    dicerun(dice);

                    hidemouseptr();
                    if(dice==6)
                    {
```

```
    setcolor(9);
    rectangle(528,348,614,388);
    setcolor(15);
    rectangle(528,400,614,440);
    chance=2;
}
else
{
    setcolor(15);
    rectangle(528,348,614,388);
    setcolor(9);
    rectangle(528,400,614,440);
    chance=1;
}

showmouseptr();
sec_dice+=dice;

for(i=0; i<5; i++)
    for(j=0; j<1; j++)
    {
        if(ladder[i][j]==sec_dice)
            sec_dice=ladder[i][1];
        if(snake[i][j]==sec_dice)
            sec_dice=snake[i][1];

    }
sound(400);
delay(500);
nosound();
```

```
if(sec_dice<=100)
{
    num=sec_dice;
    num--;
    putimageportion(prev2_dice,2);
    saveimageportion(num,2);
    changeposition(sec_dice,2);
    prev2_dice=sec_dice-1;
}
else
{
    sec_dice-=dice;
}


sprintf(dc,"%d",sec_dice);
setcolor(0);
outtextxy(556,304,"ллл");
setcolor(15);
outtextxy(557,304,dc);


if(sec_dice==100)
{
    end=1;
    hidemouseptr();
    resarea=imagesize(190,170,435,320);
    result=malloc(resarea);
    getimage(190,170,435,320,result);
```

```
            setfillstyle(1,0);
            setcolor(15);
            rectangle(190,170,435,320);
            bar(191,171,434,319);
            outtextxy(258,242,"PLAYER 2 WON");
            rectangle(369,272,415,280);
            rectangle(378,261,404,271);
            line(378,261,384,255);
            line(384,255,377,209);
            line(405,209,399,255);
            line(399,255,404,261);
            ellipse(391,209,0,360,14,3);
            setfillstyle(1,8);
            floodfill(391,234,15);
            setfillstyle(1,4);
            floodfill(386,267,15);
            setfillstyle(1,1);
            floodfill(386,276,15);
            setfillstyle(1,7);
            floodfill(391,208,15);
            settextstyle(2,0,0);
            setcolor(14);
            outtextxy(280,303,"press any key to continue..");
            getch();
            putimage(190,170,result,0);
            free(result);
            showmouseptr();

        }

    }
```

```
//////////////////////////////////////////computer mode play
            else if(selemode==1&&end==0&&chance==2)
            {

                // computer mode
                dice=random(6);
                if(dice==0)
                    dice=1;
                dicerun(dice);
                hidemouseptr();
                if(dice==6)
                {
                    setcolor(9);
                    rectangle(528,348,614,388);
                    setcolor(15);
                    rectangle(528,400,614,440);
                    chance=2;
                }
                else
                {
                    setcolor(15);
                    rectangle(528,348,614,388);
                    setcolor(9);
                    rectangle(528,400,614,440);
                    chance=1;
                }

                showmouseptr();
                sec_dice+=dice;

                for(i=0; i<5; i++)
```

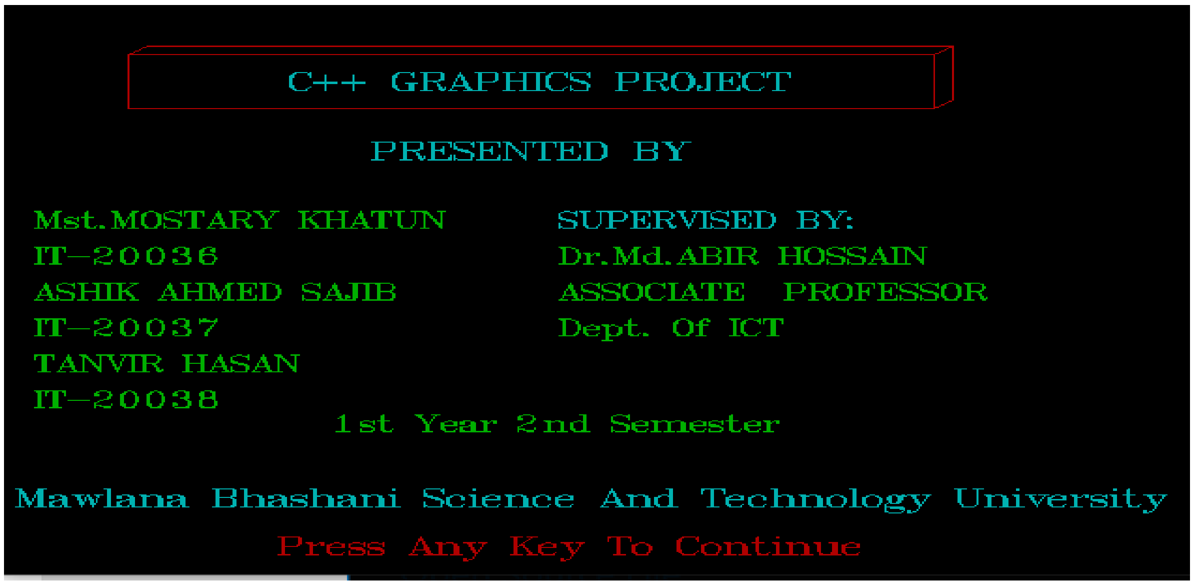
```
    for(j=0; j<1; j++)
    {
        if(ladder[i][j]==sec_dice)
            sec_dice=ladder[i][1];
        if(snake[i][j]==sec_dice)
            sec_dice=snake[i][1];

    }
sound(200);
delay(500);
sound(300);
delay(200);
sound(500);
delay(100);
nosound();


if(sec_dice<=100)
{
    num=sec_dice;
    num--;
    putimageportion(prev2_dice,2);
    saveimageportion(num,2);
    changeposition(sec_dice,2);
    prev2_dice=sec_dice-1;
}
else
{
    sec_dice-=dice;
}
```

```
sprintf(dc,"%d",sec_dice);
setcolor(0);
outtextxy(556,304,"ллл");
setcolor(15);
outtextxy(557,304,dc);


if(sec_dice==100)
{
    end=1;
    hidemouseptr();
    resarea=imagesize(190,170,435,320);
    result=malloc(resarea);
    getimage(190,170,435,320,result);

    setfillstyle(1,0);
    setcolor(15);
    rectangle(190,170,435,320);
    bar(191,171,434,319);
    outtextxy(258,242,"COMPUTER WON");
    rectangle(369,272,415,280);
    rectangle(378,261,404,271);
    line(378,261,384,255);
    line(384,255,377,209);
    line(405,209,399,255);
    line(399,255,404,261);
    ellipse(391,209,0,360,14,3);
    setfillstyle(1,8);
    floodfill(391,234,15);
    setfillstyle(1,4);
```

```
                    floodfill(386,267,15);
                    setfillstyle(1,1);
                    floodfill(386,276,15);
                    setfillstyle(1,7);
                    floodfill(391,208,15);
                    settextstyle(2,0,0);
                    setcolor(14);
                    outtextxy(280,303,"press any key to continue..");
                    getch();
                    putimage(190,170,result,0);
                    free(result);
                    showmouseptr();
                }

            }

////////////////////////////////////////////////////
        }


        // exit button pressed
        if(b==1&&sx>530&&sy>90&&sx<615&&sy<111)
        {
            break;
        }
    }

// close graphics mode
    closegraph();
// restore to CRT mode  normal text mode
    restorecrtmode();
}
```

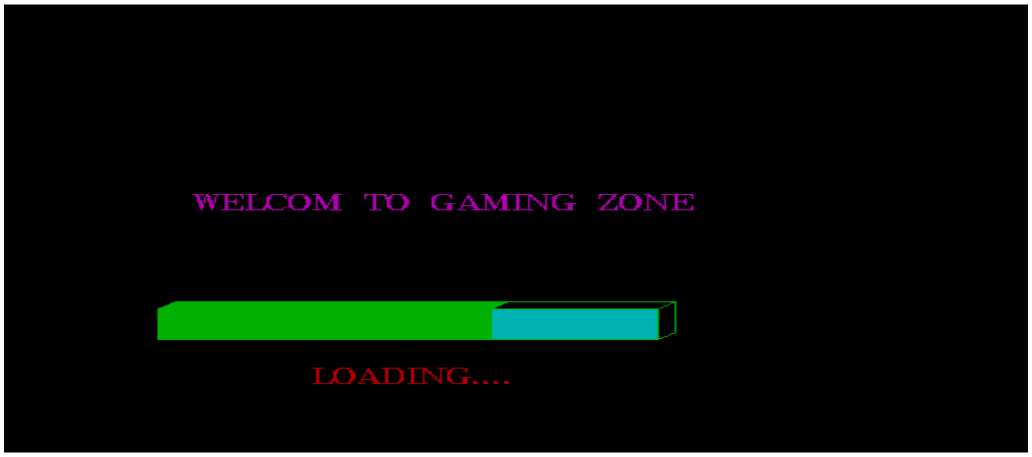
B: Screen Shots:

Front Face:

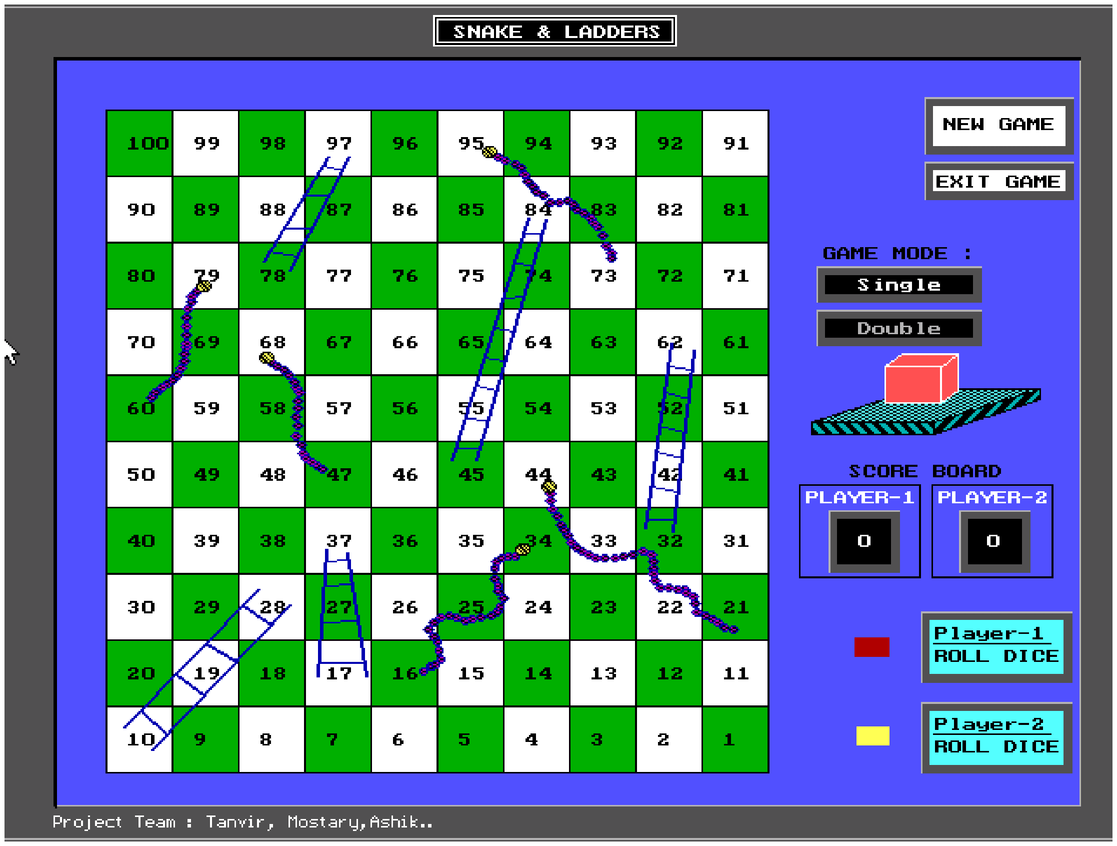


```
C++ GRAPHICS PROJECT

PRESENTED BY

Mst.MOSTARY KHATUN          SUPERVISED BY:
IT-20036                    Dr.Md.ABIR HOSSAIN
ASHIK AHMED SAJIB           ASSOCIATE  PROFESSOR
IT-20037                    Dept. Of ICT
TANVIR HASAN
IT-20038
           1st Year 2nd Semester

Mawlana Bhashani Science And Technology University
        Press Any Key To Continue
```

Second windows:



Game Window



Project Team : Tanvir, Mostary,Ashik..

BIBILIOGRAPHY & REFERENCE :

https://www.geeksforgeeks.org/include-graphics-h-codeblocks/

https://www.programmingsimplified.com/c/graphics.h

Books:

1. **Programming in ANSI C by**
   E. Balagurusamy
2. **Object Oriented Programming With C++ by**
   E. Balagurusamy