Hong Kong Diploma of Secondary Education Examination School Based Assessment

Information and Communication Technology

Option D: Software Development

Topic: Puzzle & mini games

School: Cheung Sha Wan Catholic Secondary School

Content Page

Chapter 1 : Introduction

1.1 Background

Nowadays, in this Y-generation, children are described as the strawberries protected by the monster parents, that their life plan is already planned by their parents. They just like robots that following the instructions and orders by parents. As a result, most of the children are bookworms with low level of critical and logical thinking.

As a volunteer,I usually went to primary schools for sharing experience of my secondary school life. When I chat with some students in the primary school, their lives is very boring, while some of them even don’t know what are Sudoku or some other puzzle games that we used to play. Therefore I developed this program and delighted to provide an who are trapped in the cage of their parents for some entertainment.

1.2 Objectives

Aim

An entertaining program that consist of 3 games with different level or gaming experience, will be developed to provide a playing opportunity for people. Through this game, it is hoped that logical and critical think can be trained and this program can make people happy.

Target Users

I. People who feel lonely and bored. The games in the program are easy but exciting. It is also easy to open and operate. Therefore it will be a good choice for this group of people.
II. Children in primary school age. They can train up their critical and logical thinking through the games. Also, they can find entertainment outside of the tedious homework or tests.

Chapter 2 : Design and Implementation

2.1 The User interface

When I was writing the program, I tend to use the Command Line Interface (CLI) instead of the Graphical User Interface (GUI). It is because the command line interface is more suitable for the experience users. It is because the user has to remember the commands. So, the CLI is more suitable for the experiment users and the users with some ICT knowledge. Furthermore, using the CLI can help to save the resources because the photos are not needed for the CLI.

2.2 Modularization

In this program, there are 3 main different games, which are Number Guessing, Tic-Tac-Toe and Sudoku. Before entering the game menu, users have to register or login to assure the users are members and not robot. Also, after finished the games for a number of times, the program will be automatically bring the user to the game menu.

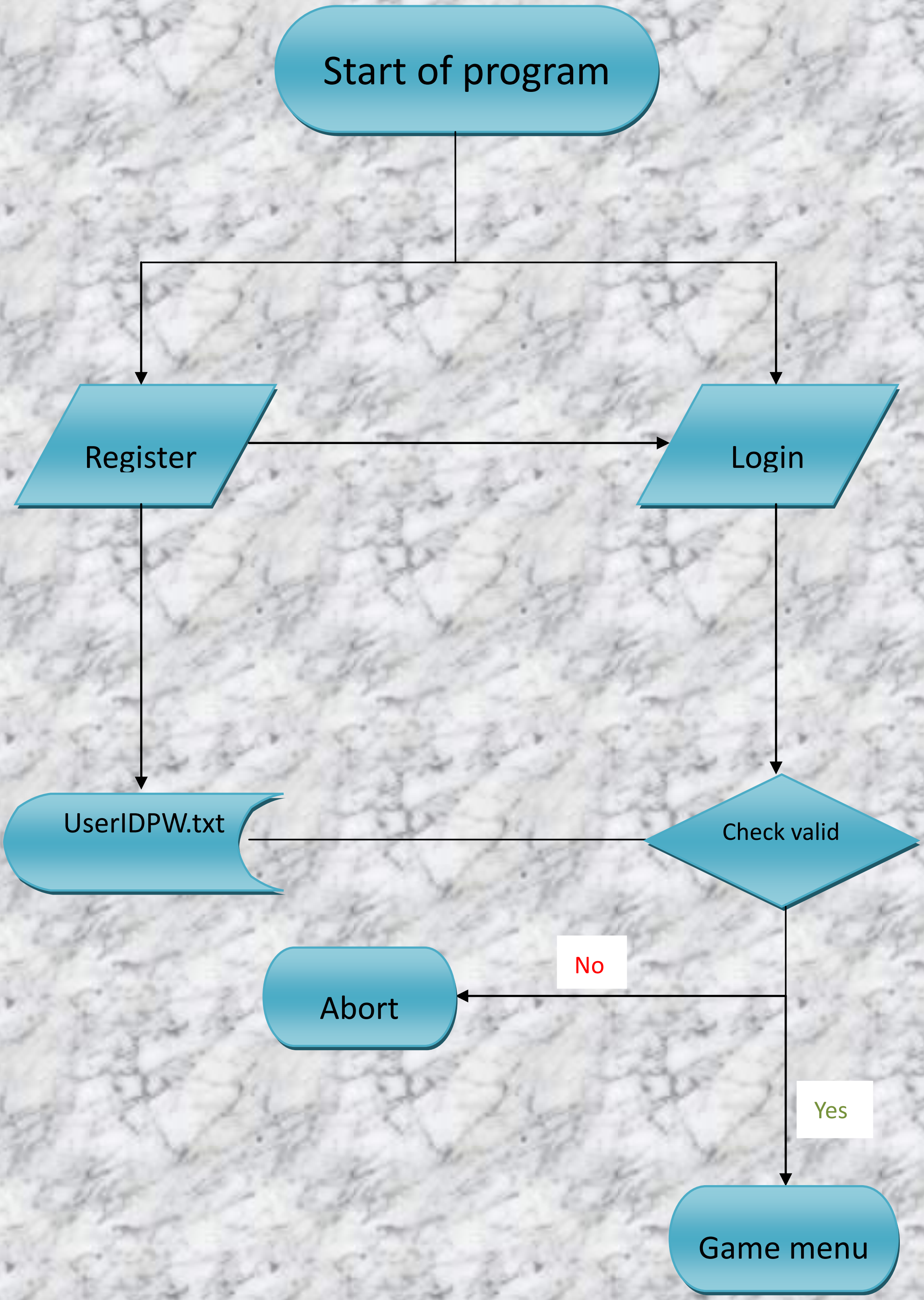


Fig.1 Login Menu

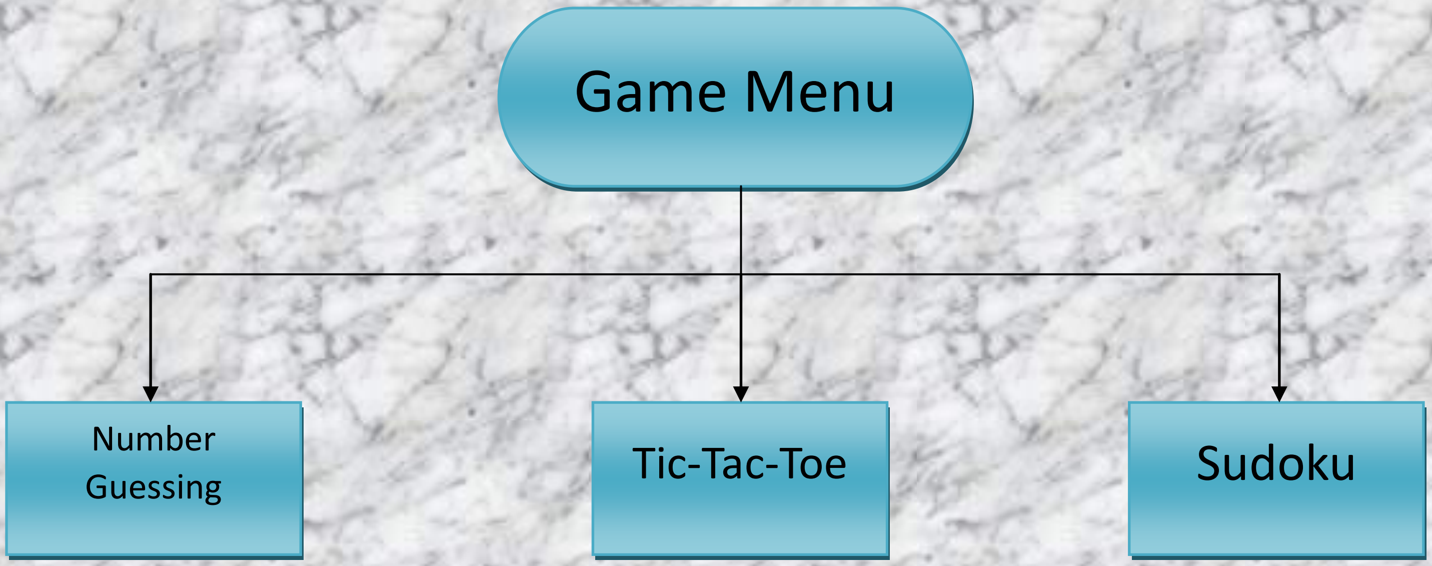


Fig.2 Game menu

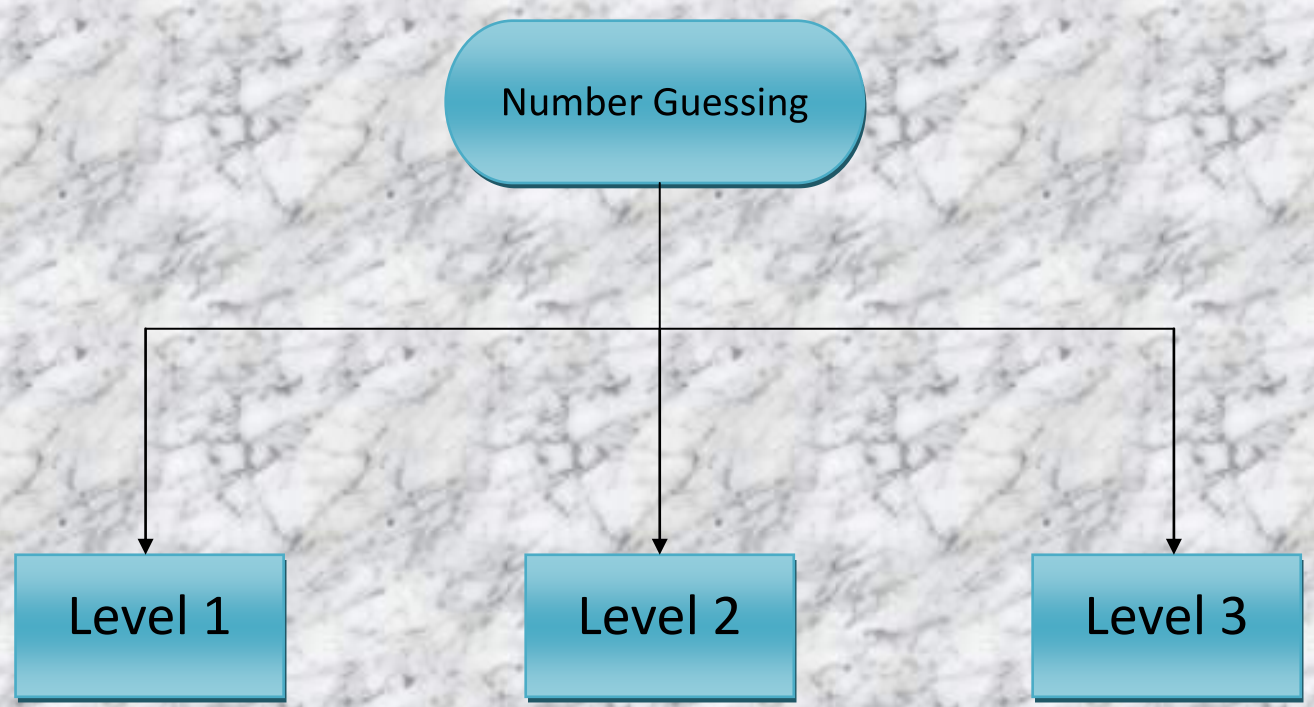


Fig.3 Number Guessing

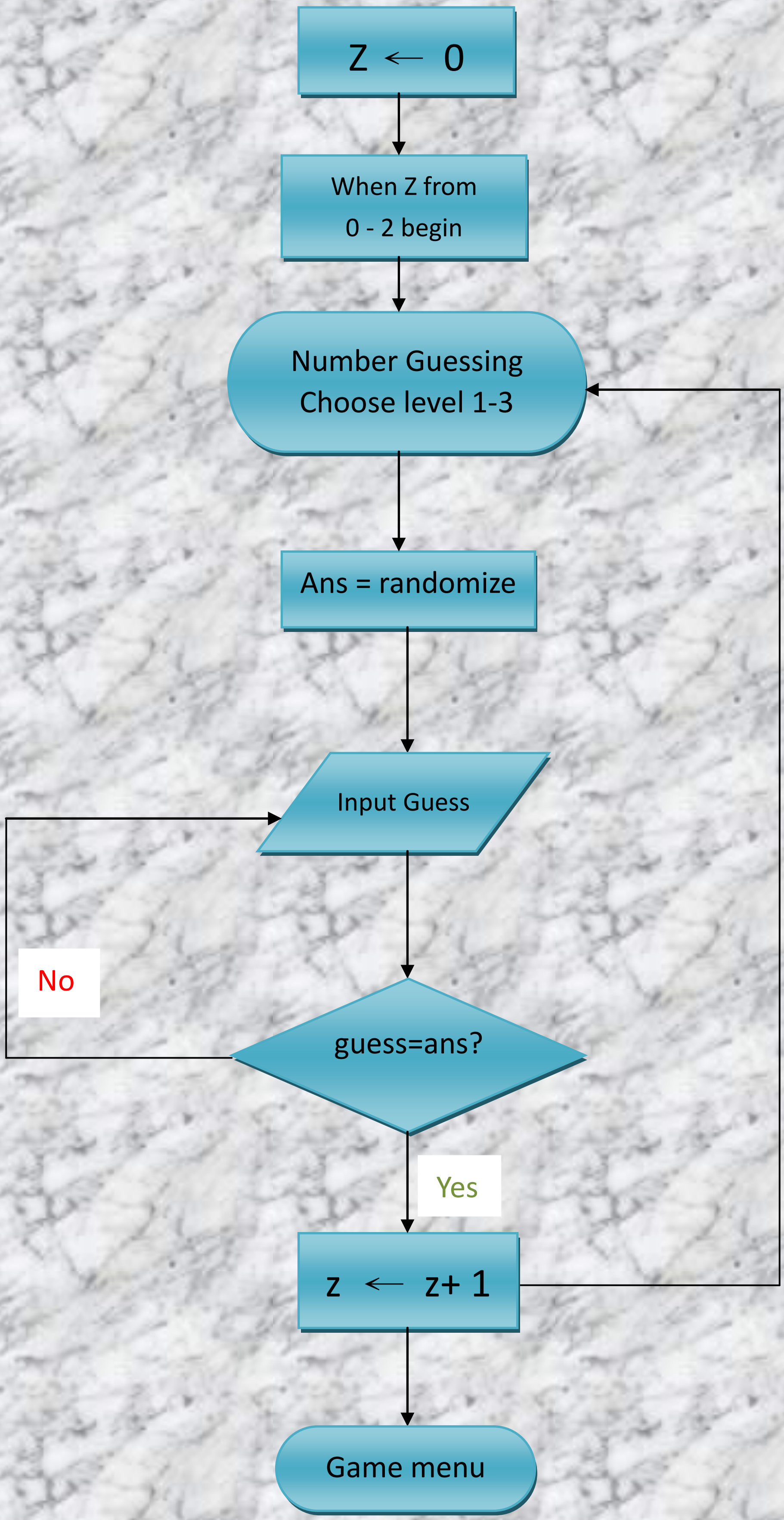


Fig.4 Flow of Number Guessing

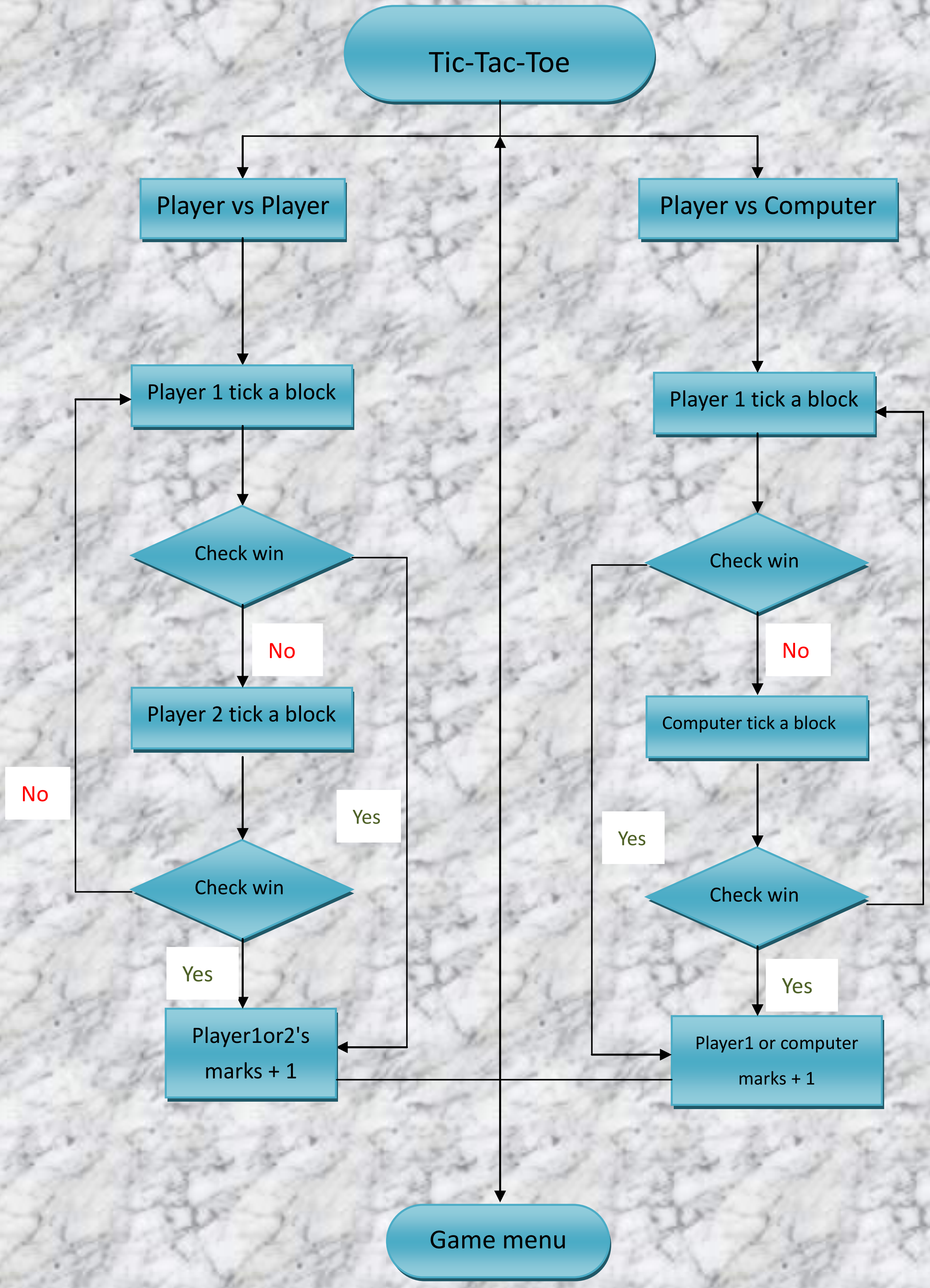


Fig.5 Flow of Tic-Tac-Toe

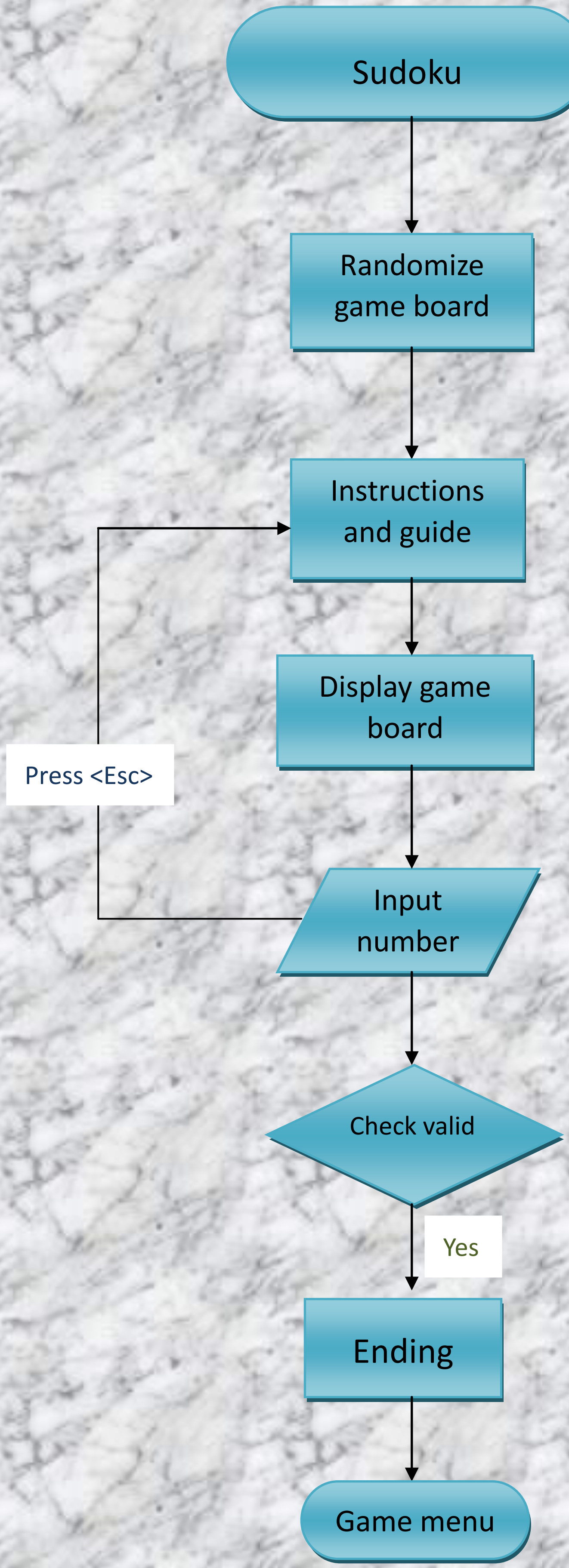
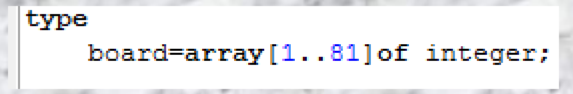


Fig.6 Flow of Sudoku

2.3 The Data Structure

In this program, I use the file to store the data of the users accounts' id and passwords. That is because using file to store is more efficient and convenient. the data in the file can be edited and deleted easily. File can also let the program become simple and more easily to debugging and data validation. File format is even easily to be controlled by a programming beginner. Therefore I prefer using file than record.

Besides, I have defined a new type, board to be 81 different arrays.



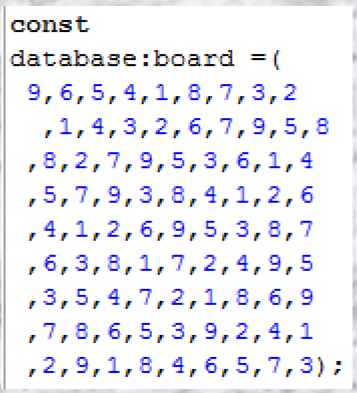
```
type
    board=array[1..81]of integer;
```

The arrays can be group together to be a complete Sudoku. This can be more effective and convenient so that redundant program coding is not exist.

2.4 Constants and Limitations

Firstly, in the game of number guessing, the maximum of range of numbers is only 1 - 1000. That is because the larger range of number, the more difficult of the game to the users. This will also make the game become tedious.

For the third game, constants are set for the game.



```
const
database:board =(
 9,6,5,4,1,8,7,3,2
  ,1,4,3,2,6,7,9,5,8
 ,8,2,7,9,5,3,6,1,4
 ,5,7,9,3,8,4,1,2,6
 ,4,1,2,6,9,5,3,8,7
 ,6,3,8,1,7,2,4,9,5
 ,3,5,4,7,2,1,8,6,9
 ,7,8,6,5,3,9,2,4,1
 ,2,9,1,8,4,6,5,7,3);
```

There are all together 81 arrays of constant numbers. When the numbers are grouped, a complete Sudoku can be formed. But there is also limitation that only 3x3 Sudoku can be formed while other sizesof Sudoku like 2x2 cannot be formed for the users.

2.5 Data Control

In order to reduce the input error, there is a validation check after the input of the data in some sub-program. It is because the typing error cannot prevent when the user are sleepy or pressing wrong button. So, using the validation check can help to minimize the input error.

Range check is also used in the first game, Number guessing. This can reduce the input error. In the game, the guessing times will be shown to players once they guess the right number. If there is no range check, players might be guess the numbers which may overlapped, affecting the guessing times and it might be inaccurate.

2.6 The Operations of Each Module

There are four main modules in the program. They are the login system, Number guessing, Tic-Tac-Toe and Sudoku.

First, in the login system, users can only choose 'Register' or 'Login', which means they must have a account before enjoying the games. When users chose register and typed his/her personal ID or password, the data will be stored a file named 'UserIDPW'. Users can register unlimited account as the data is appended to the file instead of rewrite. After registered, login interface will be automatically shown to the user for him/her to login with valid ID and password. When ID and password is typed, 'tmp' , a variable of the program will be substituted by the data in the file 'UserIDPW' to check whether the ID and the substituted tmp is identical. After the check of ID, tmp will be stored by data which is on the next line of the identical ID, and check the password with the same method. Once the ID and password are matched, the program will run continuously. But if the ID and passwords are not matched, the program will shut down without giving another chance for the users.

Secondly, in the Number Guessing Game, 3 levels can be chosen by the users which are 1-10, 1-100 and 1-1000. After users chose a level, the hidden answer will be randomized according to the level chosen. Then, users can try unlimited times for guessing the number. The guess cannot exceed the range. For example, if 1-100 level is chosen, 0 or 101, which are the numbers exceeded the range and typed, the program will ask users to try again. Even the guessing numbers is in the range of the level, some situations are still not allowed to guess some numbers. For example, it is known that the level is 1-100, the answer is 40, and a user guessed 50 in the first time guess, the user will know that the answer is between 1-50, and user cannot guess a number higher than 50, which can reduce looping. Lastly, when the correct number is guessed, the total guessing times will also be shown to the users to let he/she know his/her results.

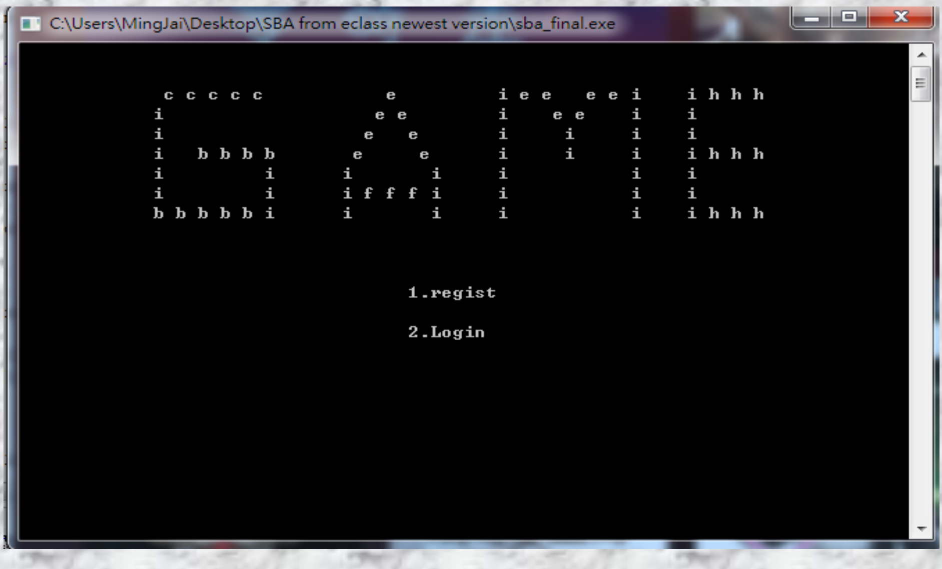
Thirdly, in the Tic-Tac-Toe game, 2 game modes can be chosen by the users which are Players vs Players and Players vs Computer. In the mode of Players vs Players, Players will take actions one by one that choosing one of the blocks among the nine blocks in the beginning. Number of blocks should be in the range of 1-9 or an error message will be occur. The chosen blocks are also cannot be chosen. After every action, the program will check whether the gameboard has fulfilled the victorious condition that 3 same symbols are in the same row, column or the oblique line. The game will be ended if the victorious condition is fulfilled, and 1 mark will be added to the winner. In the mode of Player vs Computer, the actions of computer will be randomize until player is nearly to win the game. All conditions are set that which block should the computer choose to block the way of players.

Lastly, in the Sudoku game, the numbers of the game board are randomized in the beginning. The principle is that change the rows and columns among 1 and 4, 2 and 5 , 3 and6. This can group up a new Sudoku board with different number every time the users open. In my program, 9 times of changes will be done to ensure the numbers will not repeat. Each time the user enter a number in the Sudoku board, the program will check once that whether the board is the complete one. Only when the users type all the correct number and formed a complete Sudoku can win the game.

Chapter 3 : Testing and Evaluation

3.1 Testing

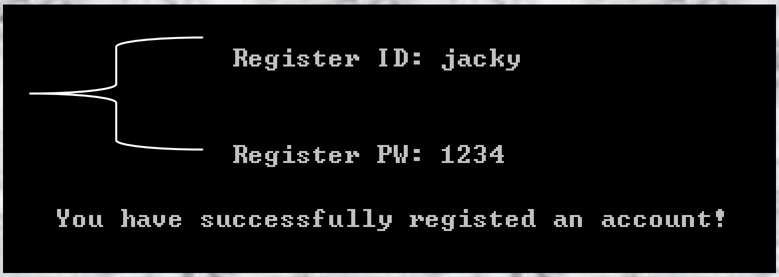
Login Menu


C:\Users\MingJai\Desktop\SBA from eclass newest version\sba_final.exe

```
1.regist

2.Login
```

choosing register or login

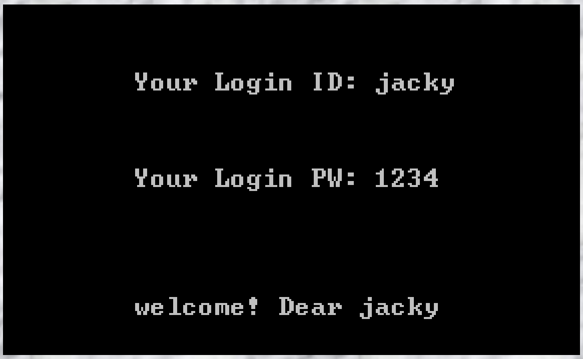
Fig.7 Login Menu



```
Register ID: jacky

Register PW: 1234

You have successfully registed an account!
```

Registering: inputting ID & PW

Fig.8 Registering

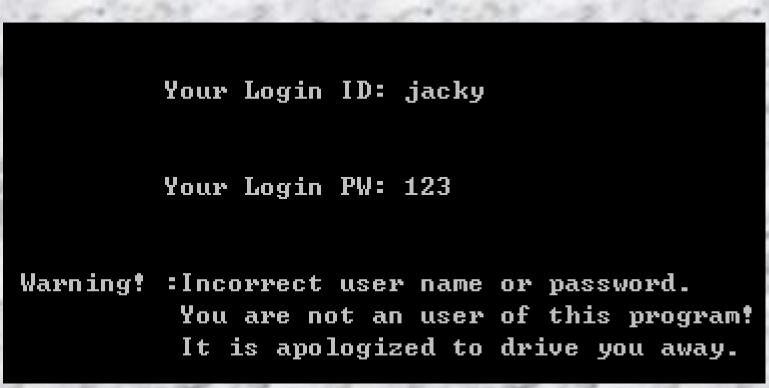


```
Your Login ID: jacky

Your Login PW: 1234


welcome! Dear jacky
```

Sample of correctly input of login ID&PW

Fig.9 Login(1)

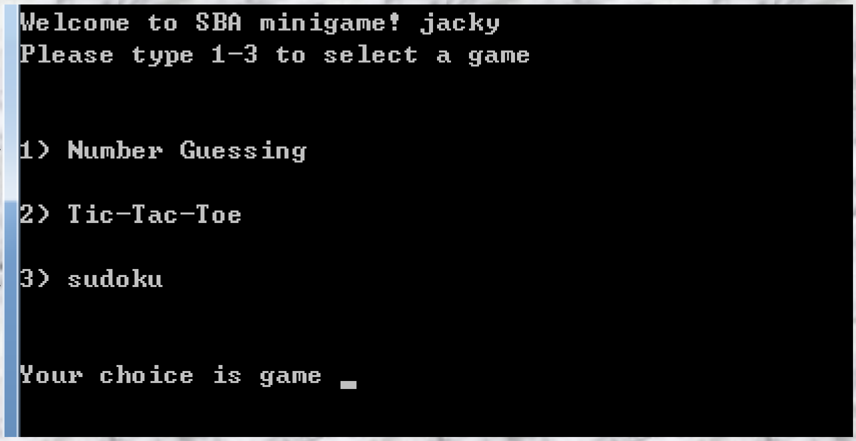
Sample of incorrectly input of login ID&PW



```
        Your Login ID: jacky


        Your Login PW: 123


Warning! :Incorrect user name or password.
          You are not an user of this program!
          It is apologized to drive you away.
```

Fig.10 Login(2)

Game Menu

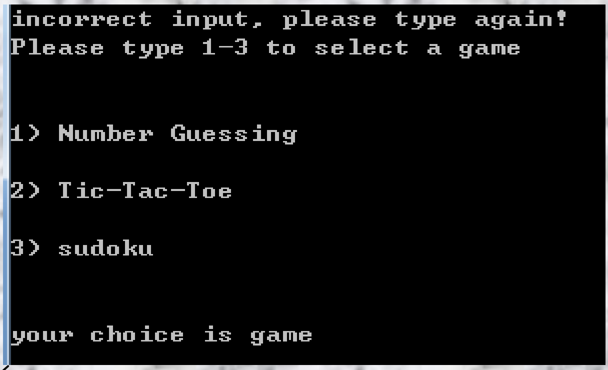


```
Welcome to SBA minigame! jacky
Please type 1-3 to select a game


1) Number Guessing

2) Tic-Tac-Toe

3) sudoku


Your choice is game _
```

Choices of games=ReadKey

Fig.11 Game Menu

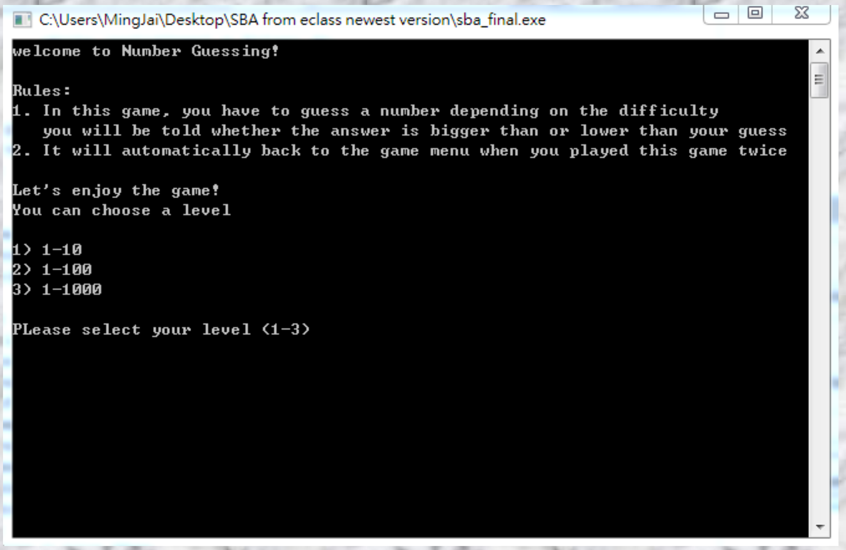
A reminder will be appear when choices are incorrect



```
incorrect input, please type again!
Please type 1-3 to select a game


1) Number Guessing

2) Tic-Tac-Toe

3) sudoku


your choice is game
```

Fig.12 Choosing game(invalid case)

Pressing '1'



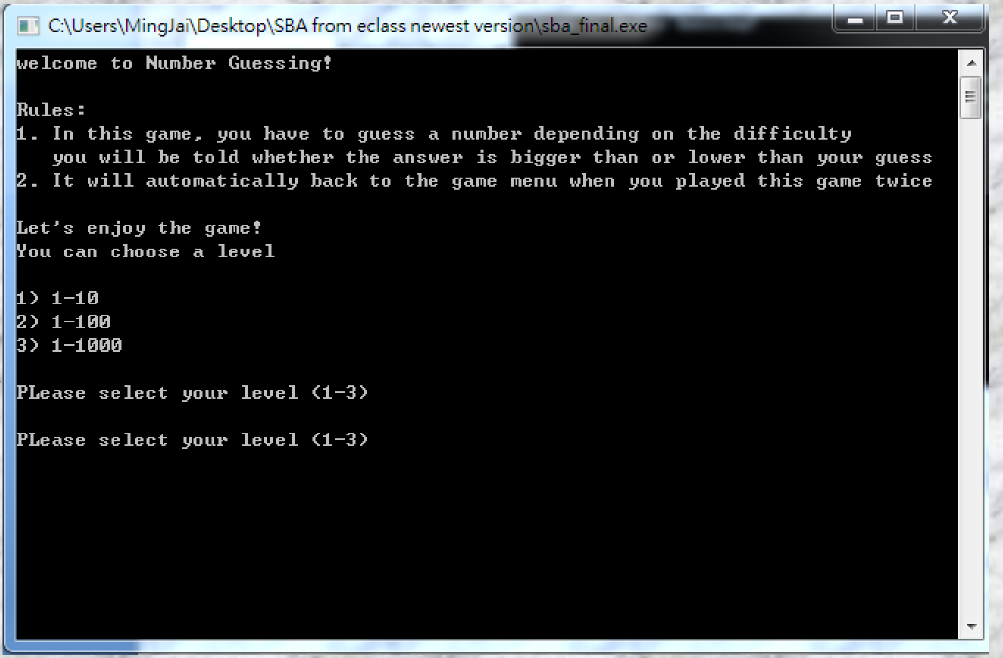
C:\Users\MingJai\Desktop\SBA from eclass newest version\sba_final.exe

```
welcome to Number Guessing!

Rules:
1. In this game, you have to guess a number depending on the difficulty
   you will be told whether the answer is bigger than or lower than your guess
2. It will automatically back to the game menu when you played this game twice

Let's enjoy the game!
You can choose a level

1) 1-10
2) 1-100
3) 1-1000

PLease select your level (1-3)
```

Fig.13 Interface of Number guessing

Game 1 : Number Guessing

The inquiry will be continuously asked until a valid input

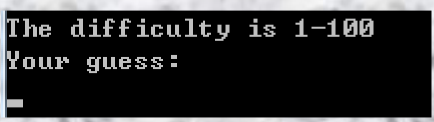


C:\Users\MingJai\Desktop\SBA from eclass newest version\sba_final.exe

```
welcome to Number Guessing!

Rules:
1. In this game, you have to guess a number depending on the difficulty
   you will be told whether the answer is bigger than or lower than your guess
2. It will automatically back to the game menu when you played this game twice

Let's enjoy the game!
You can choose a level

1) 1-10
2) 1-100
3) 1-1000

PLease select your level (1-3)

PLease select your level (1-3)
```

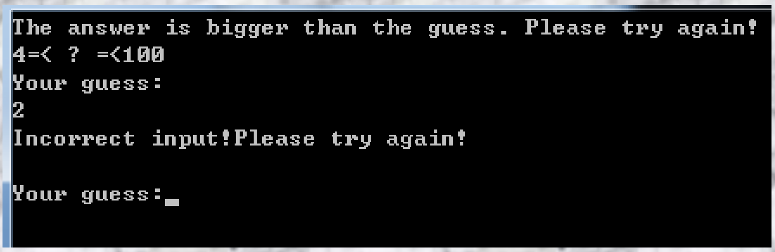
Fig.14 Choosing level(invalid case)

Fig.15 Choosing level(valid case) (e.g. Level 2)



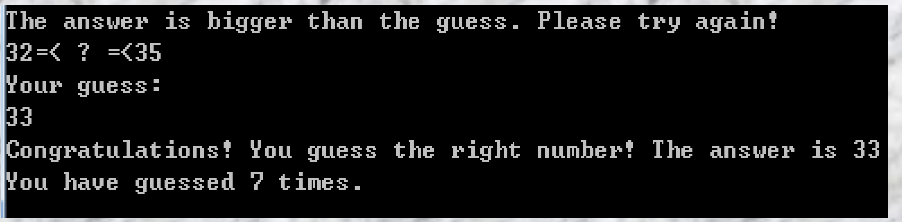
```
The difficulty is 1-100
Your guess:
```

Fig.16 Guessing(invalid case)



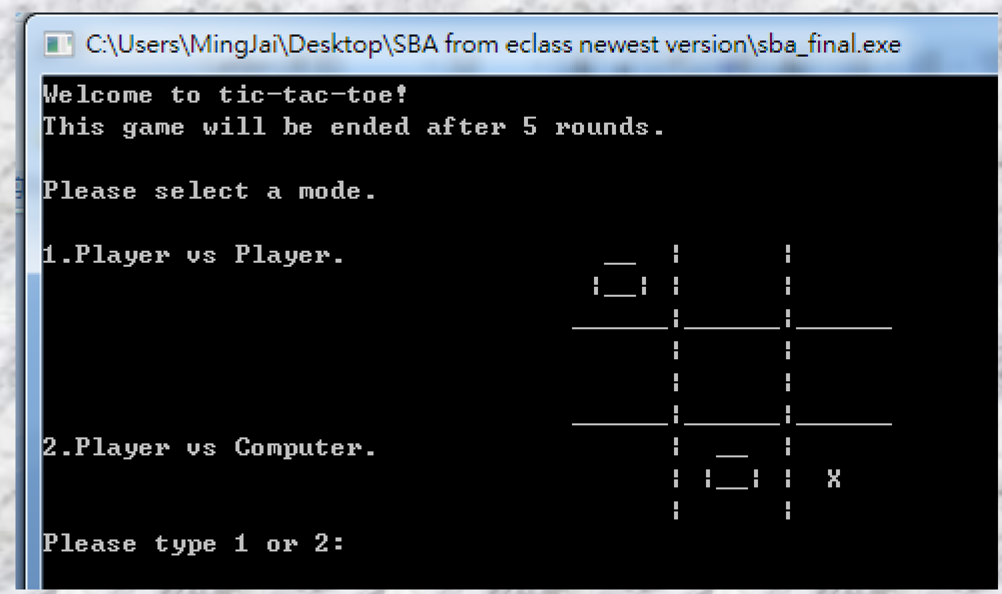
```
The answer is bigger than the guess. Please try again!
4=< ? =<100
Your guess:
2
Incorrect input!Please try again!

Your guess:
```

Fig.17 Guessing(correct answer)



```
The answer is bigger than the guess. Please try again!
32=< ? =<35
Your guess:
33
Congratulations! You guess the right number! The answer is 33
You have guessed 7 times.
```

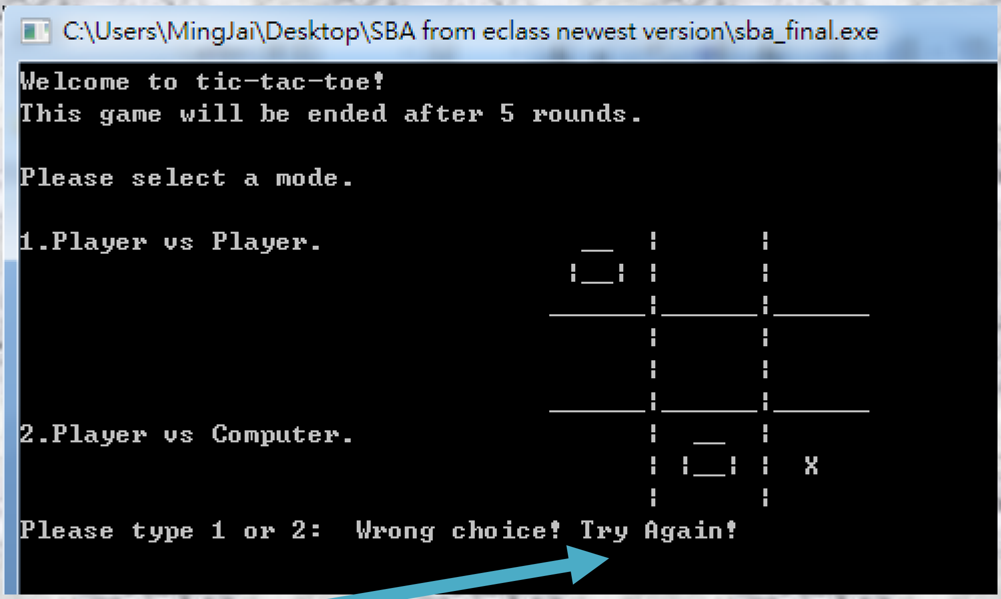
<u>Game 2 : Tic-Tac Toe</u>



C:\Users\MingJai\Desktop\SBA from eclass newest version\sba_final.exe

```
Welcome to tic-tac-toe!
This game will be ended after 5 rounds.

Please select a mode.

1.Player vs Player.

2.Player vs Computer.

Please type 1 or 2:
```

Fig.18 Interface of Tic-Tac-Toe



C:\Users\MingJai\Desktop\SBA from eclass newest version\sba_final.exe

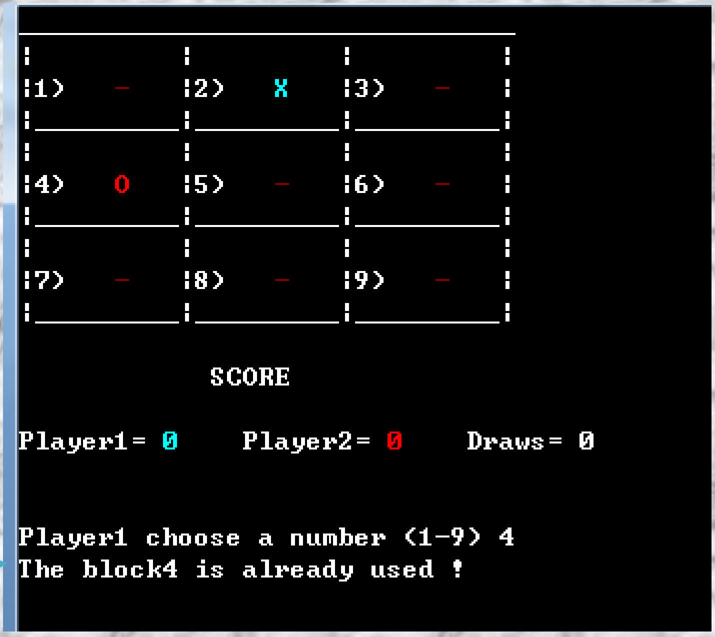
```
Welcome to tic-tac-toe!
This game will be ended after 5 rounds.

Please select a mode.

1.Player vs Player.

2.Player vs Computer.

Please type 1 or 2:  Wrong choice! Try Again!
```

Fig.19 Choosing mode(invalid case)

A signal will be appear

Fig.20 Choosing blocks(invalid case1)

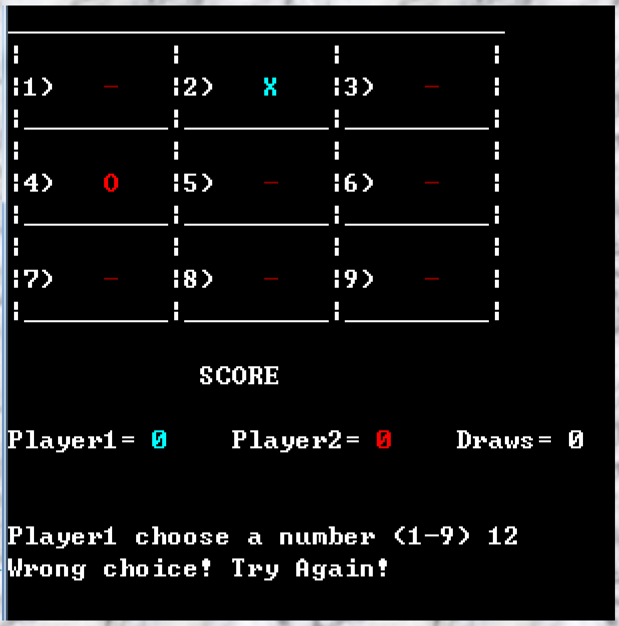
A signal will be appear



```
 ______________________________________________
|           |           |           |
|1)    -    |2)    X    |3)    -    |
|___________|___________|___________|
|           |           |           |
|4)    O    |5)    -    |6)    -    |
|___________|___________|___________|
|           |           |           |
|7)    -    |8)    -    |9)    -    |
|___________|___________|___________|

             SCORE

Player1= 0     Player2= 0     Draws= 0


Player1 choose a number (1-9) 4
The block4 is already used !
```

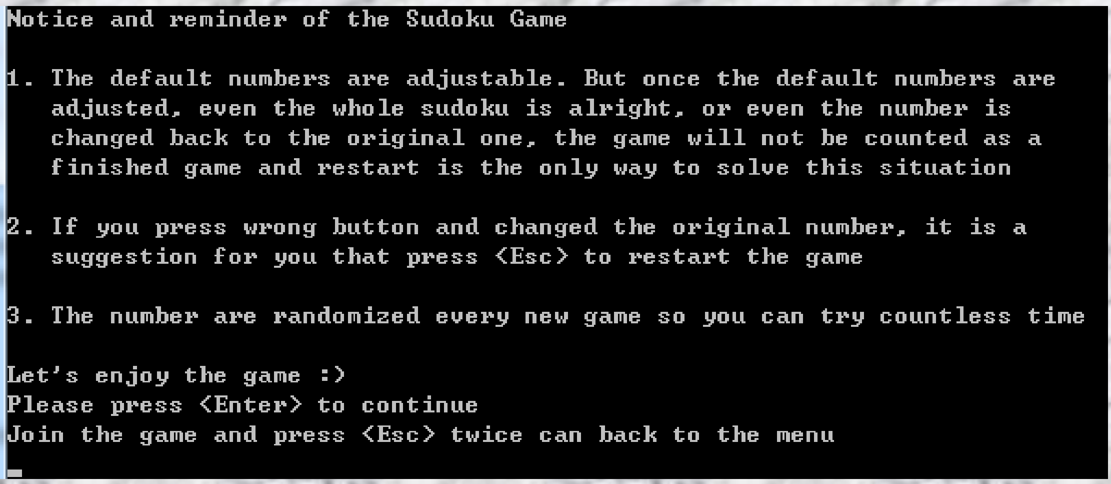
Fig.21 Choosing blocks(invalid case2)

A signal will be appear



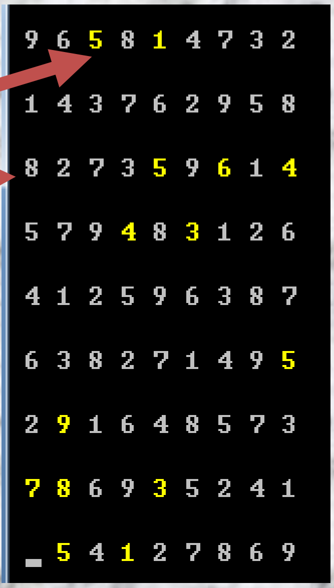
```
 ______________________________________________
|           |           |           |
|1)    -    |2)    X    |3)    -    |
|___________|___________|___________|
|           |           |           |
|4)    O    |5)    -    |6)    -    |
|___________|___________|___________|
|           |           |           |
|7)    -    |8)    -    |9)    -    |
|___________|___________|___________|

             SCORE

Player1= 0     Player2= 0     Draws= 0


Player1 choose a number (1-9) 12
Wrong choice! Try Again!
```

Game 3 : Sudoku



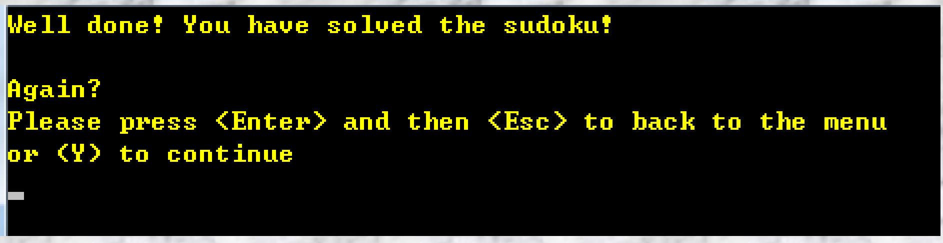
```
Notice and reminder of the Sudoku Game

1. The default numbers are adjustable. But once the default numbers are
   adjusted, even the whole sudoku is alright, or even the number is
   changed back to the original one, the game will not be counted as a
   finished game and restart is the only way to solve this situation

2. If you press wrong button and changed the original number, it is a
   suggestion for you that press <Esc> to restart the game

3. The number are randomized every new game so you can try countless time

Let's enjoy the game :)
Please press <Enter> to continue
Join the game and press <Esc> twice can back to the menu
```

Fig.22 Interface of Sudoku



```
9 6 5 8 1 4 7 3 2

1 4 3 7 6 2 9 5 8

8 2 7 3 5 9 6 1 4

5 7 9 4 8 3 1 2 6

4 1 2 5 9 6 3 8 7

6 3 8 2 7 1 4 9 5

2 9 1 6 4 8 5 7 3

7 8 6 9 3 5 2 4 1

_ 5 4 1 2 7 8 6 9
```

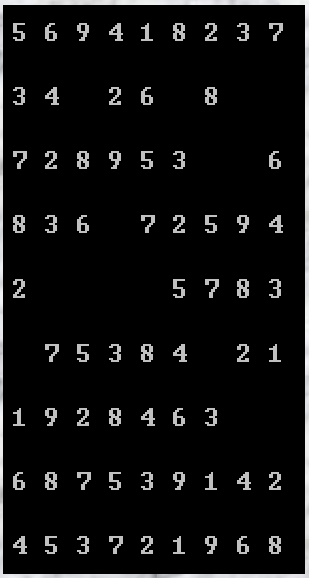
Fig.23 Game board of Sudoku



```
Well done! You have solved the sudoku!

Again?
Please press <Enter> and then <Esc> to back to the menu
or (Y) to continue
```

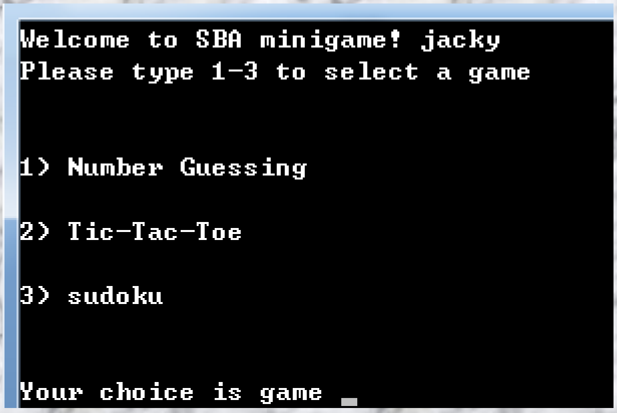
Fig.24 Finished interface

Fig.25 Test of when 'y' or 'Y' is typed



```
5 6 9 4 1 8 2 3 7

3 4   2 6   8

7 2 8 9 5 3     6

8 3 6   7 2 5 9 4

2         5 7 8 3

  7 5 3 8 4   2 1

1 9 2 8 4 6 3

6 8 7 5 3 9 1 4 2

4 5 3 7 2 1 9 6 8
```

Fig.26 Test of when <Enter> and <Esc> is pressed one after another



```
Welcome to SBA minigame! jacky
Please type 1-3 to select a game


1) Number Guessing

2) Tic-Tac-Toe

3) sudoku


Your choice is game _
```

3.2 Evaluate the program

I think that the program is seems not to be too user-friendly. For example, the program will directly abort when the users did not login with the correct user ID or Password. Besides, the display of the program is not well designed. So, the display of the program is also one of the flaw of the program.

But, the program consisted a login system, which seems to be a humanized design. Also, most of the basic functions inside the program can be called out by read key with typing. This can be more convenient and time can be saved. Furthermore, the games have different play modes and levels which can let people choose the one which is the most suitable to them.

Therefore, the program cannot be identify as the most user-friendly and most functional mini game program system.

Chapter 4 : Appendices

4.1 Program code

```
Program SBA;
uses crt, sysutils;
type
    board=array[1..81]of integer;

const
database:board =(
  9,6,5,4,1,8,7,3,2
   ,1,4,3,2,6,7,9,5,8
  ,8,2,7,9,5,3,6,1,4
  ,5,7,9,3,8,4,1,2,6
  ,4,1,2,6,9,5,3,8,7
  ,6,3,8,1,7,2,4,9,5
  ,3,5,4,7,2,1,8,6,9
  ,7,8,6,5,3,9,2,4,1
  ,2,9,1,8,4,6,5,7,3);

var table:array[1..9] of string;
    bool:array[1..9] of boolean;
    k,d, y , c, z,a,g :integer;
    Player1, Player2, Draw, Comp:boolean;
    yesorno:char;
    x, o, e:string;
    p12points, p1cpoints, p2points, cpoints, d1points, d2points, color:integer;
    gameBoard:board;
    solBoard:board;
    p,choice , listen:char;
    i,tempxy, tmp:integer;
    valid : array [1..81]of Boolean;
    registid , registpw : string;

  userid, userpw : string;
  f : textfile;
```

```
procedure regist;
begin
   assignfile(f, 'C:\Users\MingJai\Desktop\SBA from eclass newest
version\UserIDPW.txt');
   append(f);
   writeln;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln;
   write('                                        Register ID: ');
   readln(registid);
   writeln(f, registid);
   writeln;
   writeln;
   write('                                        Register PW: ');
   readln(registpw);
   writeln(f, registpw);
   close(f);
   writeln;
   writeln('                         You have successfully registed an account!');
   readln;
   clrscr;
end;

procedure cklogin;
var
verified : boolean;
tmp : string;
begin
   verified := false;
   writeln;
   writeln;
   writeln;
```

```
    writeln;
    writeln;
    writeln;
    writeln;
    write('                                              Your Login ID: ');
    readln(userid);
    writeln;
    writeln;
    write('                                              Your Login PW: ');
    readln(userpw);
    assignfile(f, 'C:\Users\MingJai\Desktop\SBA from eclass newest
version\UserIDPW.txt');
    reset(f);
  while not (eof(f)) or (verified=true) do
  begin
       readln(f,tmp);
         if (userid = tmp) then
         begin
                   readln(f,tmp);
                   if (userpw = tmp) then
                   begin
                   verified := true;
                   break;
                   end
                   else
                   begin
                   readln(f,tmp);
                   end;
         end
         else begin
         readln(f,tmp);
         end;

end;

  close(f);
if verified=true then
  begin
```

```
   writeln;
   writeln;
   writeln;
   writeln('                                          welcome! Dear ', userid);
   end
   else begin
   writeln;
   writeln;
   writeln('                            Warning! :Incorrect user name or password.');
   writeln('                                          You are not an user of this
program!');
   writeln('                                          It is apologized to drive you
away.');
   readln;
   abort;
   end;
   readln;

end;


procedure menu;
   begin
   clrscr;
      writeln('Welcome to SBA minigame! ',userid);
      writeln('Please type 1-3 to select a game');
      writeln;
      writeln;
      writeln('1) Number Guessing');
      writeln;
      writeln('2) Tic-Tac-Toe');
      writeln;
      writeln('3) sudoku');
      writeln;
      writeln;
      write('Your choice is game ');
     end;
```

```
procedure userbg;
var
  choice : char;
begin
    writeln;
    writeln;
    writeln('                ________          ▄         █▄▄▄  ▄▄
█   █▄▄▄▄    ');
    writeln('               █                ▄▄        █  ▄▄
█   █        ');
    writeln('               █               ▄  ▄       █    █
█   █        ');
    writeln('               █  ______      ▄    ▄      █    █
█   █▄▄▄▄    ');
    writeln('               █        █   █      █      █
█   █        ');
    writeln('               █        █   █▄▄▄▄▄▄█      █
█   █        ');
    writeln('               ________█   █      █      █
█   █▄▄▄▄    ');
    writeln;
    writeln;
    writeln;
    writeln('                                          1.regist');
    writeln;
    writeln('                                          2.Login');
    repeat
    choice := readkey;
    until (choice='1') or (choice='2');
    if (choice='1') or (choice='2') then begin
    if (choice='1') then
    begin
    clrscr;
    regist;
    end;
    if (choice='2') then
    clrscr;
    cklogin;
```

```
      end;
end;

procedure game3;


procedure sudokuReady;
begin

gameBoard:=database;
        for i := 1 to 81 do
        begin
        valid[i]:= true;

        end;
end;

procedure randomBoard(var gameBoard:board; var tempxy:integer);
var j,tmp,change:integer;
begin
tempxy:=random(2)+1;

repeat
            begin
            change:=random(9)+1;
            end;
until (change <> 2) and (change <> 5) and (change <> 8);
            if tempxy = 1 then
                  begin
                      for j:=1 to 9 do
                      begin
                          if ((change = 1) or (change = 3)) then
                          begin
                          tmp:=gameBoard[1+(9*(j-1))];
                          gameBoard[1+(9*(j-1))]:= gameBoard[3+(9*(j-1))];
                          gameBoard[3+(9*(j-1))] := tmp;
                          end;
                          if ((change = 4) or (change = 6)) then
```

```
                        begin
                        tmp:=gameBoard[4+(9*(j-1))];
                        gameBoard[4+(9*(j-1))]:= gameBoard[6+(9*(j-1))];
                        gameBoard[6+(9*(j-1))] := tmp;
                        end;
                        if ((change = 7) or (change = 9)) then
                        begin
                        tmp:=gameBoard[7+(9*(j-1))];
                        gameBoard[7+(9*(j-1))]:= gameBoard[9+(9*(j-1))];
                        gameBoard[9+(9*(j-1))] := tmp;
                        end;
                    end;
                end;
                if tempxy = 2 then
                begin
                    for j:=1 to 9 do
                    begin
                        if ((change = 1) or (change = 3)) then
                        begin
                        tmp:=gameBoard[j];
                        gameBoard[j]:= gameBoard[2*9+j];
                        gameBoard[2*9+j] := tmp;
                        end;
                        if ((change = 4) or (change = 6)) then
                        begin
                        tmp:=gameBoard[3*9+j];
                        gameBoard[3*9+j]:= gameBoard[5*9+j];
                        gameBoard[5*9+j] := tmp;
                        end;
                        if ((change = 7) or (change = 9)) then
                        begin
                        tmp:=gameBoard[6*9+j];
                        gameBoard[6*9+j]:= gameBoard[8*9+j];
                        gameBoard[8*9+j] := tmp;
                        end;
                end;
            end;
end;
```

```
procedure youWin;
begin
ClrScr;
writeln('Well done! You have solved the sudoku!');
writeln;
writeln('Again?');
writeln('Please press <Enter> and then <Esc> to back to the menu' );
writeln('or (Y) to continue');
yesorno := readkey;
if (yesorno='y') or (yesorno='Y') then
textcolor(15);
game3;
readln;
end;


procedure draw;
var i:integer;
begin
        for i:=0 to 80 do
        begin
                begin
                        if valid[i+1] = true then
                        begin
                        gotoxy(2+2*(i mod 9),2+2*(i div 9));
                        write(gameBoard[i+1]);
                        end else begin
                        gotoxy(2+2*(i mod 9),2+2*(i div 9));
                        write(' ');
                        end;
                end;
        end;
        gotoxy(2,2);
end;

procedure buildSol(var solBoard:board);
var i,tmp: integer ;
```

```
begin
solBoard:=gameBoard;
        for i := 1 to 15 do begin
                tmp:=random(81)+1;

                        while (valid[tmp] = false) do
                        begin
                        tmp:=random(81)+1;
                        end;
                gameBoard[tmp]:= 0;
                valid[tmp] := false;
        end;
end;

procedure checkValid;
var i:integer;
        done:boolean;
begin
        done := true;
        for i := 1 to 81 do
        begin
        if valid[i] = false then
              if gameBoard[i] = solBoard[i] then
              begin
              valid[i] := true;
              end
              else
              done := false;
        end;
        if done = true then
        youWin;
end;


begin
tempxy:=2;
ClrScr;
sudokuReady;
```

```
randomize;
for i:= 1 to 30 do begin
randomBoard(gameBoard, tempxy);
end;
buildSol(solBoard);
draw;
textcolor(14);
repeat
        listen:=readkey;
        if ord(listen)=0 then
        begin
              listen:=readkey;
              case ord(listen) of
              72:if wherey<>2 then
              gotoxy(wherex,wherey-2);
              80:if wherey<>18 then
              gotoxy(wherex,wherey+2);
              75:if wherex<>2 then
              gotoxy(wherex-2,wherey);
              77:if wherex<>18 then
              gotoxy(wherex+2,wherey);
              end;
              end;
              if (listen>'0') and (listen<='9') then
                  begin
                  write(listen);
                  tmp:=(wherex-1)div 2 + 9*((wherey div 2)-1);

                  gameBoard[tmp]:= strtoInt(listen);
                  gotoxy(wherex-1,wherey);
                  checkValid;
                  end;
until (ord(listen)=27);
textcolor(15);
end;
```

```
procedure game2;

procedure background;
        begin
        writeln('Welcome to tic-tac-toe!');
        writeln('This game will be ended after 5 rounds.');
        writeln;
        writeln('Please select a mode.');
        writeln;
        writeln('1.Player vs Player.                        __    |         |
');
        writeln('                                            |__| |         |
');
        writeln('                                          ______|______|______
');
        writeln('                                                 |         |
');
        writeln('                                                 |         |
');
        writeln('                                          ______|______|______
');
        writeln('2.Player vs Computer.                         |    __    |
');
        writeln('                                                 | |__| |    X
');
        writeln('                                                 |         |
');
        write('Please type 1 or 2:    ');
        end;


procedure wrongbutton;
        begin
        writeln('Wrong choice! Try Again!');
        delay(500);
        clrscr;
        end;
```

```
procedure enter;
    begin
    writeln;
    writeln('Press <Enter> to continue.');
    readln;
    clrscr;
    end;

procedure colors;
    begin
        if table[z]=x then color:=27
        else if table[z]=o then color:=60
        else if table[z]=e then color:=4;
    end;

procedure map;
    begin
    writeln('_____________________________________');
    writeln('|           |           |           |');
    write('|1)');
    z:=1;
    colors;
    textcolor(color);
    write(table[1]);
    textcolor(15);
    write('|2)');
    z:=2;
    colors;
    textcolor(color);
    write(table[2]);
    textcolor(15);
    write('|3)');
    z:=3;
    colors;
    textcolor(color);
    write(table[3]);
    textcolor(15);
    writeln('|');
    writeln('|_________|_________|_________|');
    writeln('|         |         |         |');
    write('|4)');
    z:=4;
    colors;
    textcolor(color);
    write(table [4]);
    textcolor(15);
    write('|5)');
    z:=5;
    colors;
    textcolor(color);
    write(table[5]);
    textcolor(15);
    write('|6)');
    z:=6;
    colors;
    textcolor(color);
    write(table[6]);
    textcolor(15);
    writeln('|');
    writeln('|_________|_________|_________|');
    writeln('|         |         |         |');
    write('|7)');
    z:=7;
    colors;
    textcolor(color);
    write(table [7]);
    textcolor(15);
    write('|8)');
    z:=8;
    colors;
    textcolor(color);
    write(table[8]);
    textcolor(15);
    write('|9)');
    z:=9;
    colors;
```

```
        textcolor(color);
        write(table[9]);
        textcolor(15);
        writeln('|');
        writeln('|_________|_________|_________|');
        writeln;
        writeln('                    SCORE');
        writeln;
            if (p='1') then begin
                          write('Player1= ');
                          textcolor(27);
                          write(p12points);
                          textcolor(15);
                          write('      Player2= ');
                          textcolor(60);
                          write(p2points);
                          textcolor(15);
                          writeln('      Draws= ', d1points);
                          end
                else if (p='2') then begin
                                   write('Player1= ');
                                   textcolor(27);
                                   write(p1cpoints);
                                   textcolor(15);
                                   write('      Computer= ');
                                   textcolor(60);
                                   write(cpoints);
                                   textcolor(15);
                                   writeln('      Draws= ', d2points);
                                   end;
        writeln;
        end;

    procedure P1;
        begin
            repeat
            clrscr;
            map;
            writeln;
            write('Player1 choose a number (1-9) ');
            readln(y);
                if (y>0) and (y<10) then
                    begin
                        if (table[y]=x) or (table[y]=o) then
                            begin
                            writeln('The block', y ,' is already used ! ');
                            delay(1000);
                            end;
                    end
                else wrongbutton;
            until (table[y]=e);
            table[y]:=x;
            bool[y]:=true;
        if (table[1]=x) and (table[2]=x) and (table[3]=x) or
           (table[4]=x) and (table[5]=x) and (table[6]=x) or
           (table[7]=x) and (table[8]=x) and (table[9]=x) or
           (table[1]=x) and (table[4]=x) and (table[7]=x) or
           (table[2]=x) and (table[5]=x) and (table[8]=x) or
           (table[3]=x) and (table[6]=x) and (table[9]=x) or
           (table[1]=x) and (table[5]=x) and (table[9]=x) or
           (table[3]=x) and (table[5]=x) and (table[7]=x) then
            begin
            clrscr;
            map;
            writeln('Player1 has won!');
            enter;
            clrscr;
                if (p='1') then p12points:=p12points+1
                else if (p='2') then p1cpoints:=p1cpoints+1;
            Player1:=true;
            end
       else if (Player1=false) and (bool[1]=true) and (bool[2]=true) and
    (bool[3]=true) and
               (bool[4]=true) and (bool[5]=true) and (bool[6]=true) and
    (bool[7]=true) and
               (bool[8]=true) and (bool[9]=true) then
```

```
                    begin
                    writeln('The game is Draw!');
                    enter;
                    clrscr;
cpoints:=0;
d1points:=0;
d1points:=0;
                              if (p='1') then d1points:=d1points+1
                              else if (p='2') then d2points:=d2points+1;
                    Draw:=true;
                    end;
          end;

procedure CompNumb;
          begin
          table[c]:=o;
          bool[c]:=true;
          writeln('Computer chose number ', c);
          enter;
          end;



begin
p12points:=0;
p1cpoints:=0;
p2points:=0;
a := 0;
for a := 0 to 3 do
begin
x:='      X      ';
o:='      O      ';
e:='      -      ';
for d:=1 to 9 do
          begin
          table[d]:=e;
          bool[d]:=false;
          end;
Player1:=false;
```

```
Player2:=false;
Comp:=false;
draw:=false;
cursoroff;
randomize;
background;


p := readkey;
    case p of '1':
        begin
        writeln;
        writeln;
        writeln('Your selection is ''Player vs Player.''');
        enter;
        clrscr;
            repeat
            P1;
                if (Player1=false) and (Draw=false) then
                begin
                    repeat
                    clrscr;
                    map;
                    writeln;
                    write('Player2 choose a number (1-9) ');
                    readln(y);
                        if (y>0) and (y<10) then
                            begin
                                if (table[y]=o) or (table[y]=x) then
                                    begin
                                    writeln('The Block', y ,' is already
used !');
                                    delay(1000);
                                    end;
                            end
                        else wrongbutton;
                    until (table[y]=e) ;
                table[y]:=o;
```

```
                        bool[y]:=true;
                        if (table[1]=o) and (table[2]=o) and (table[3]=o) or
                            (table[4]=o) and (table[5]=o) and (table[6]=o) or
                            (table[7]=o) and (table[8]=o) and (table[9]=o) or
                            (table[1]=o) and (table[4]=o) and (table[7]=o) or
                            (table[2]=o) and (table[5]=o) and (table[8]=o) or
                            (table[3]=o) and (table[6]=o) and (table[9]=o) or
                            (table[1]=o) and (table[5]=o) and (table[9]=o) or
                            (table[3]=o) and (table[5]=o) and (table[7]=o) then
                             begin
                             clrscr;
                             map;
                             writeln('Player2 has won');
                             enter;
                             clrscr;
                             p2points:=p2points+1;
                             Player2:=true;
                             end;
                        end;
                    until (Player1=true) or (Player2=true) or (Draw=true);
                end;
                '2':  begin
                    writeln('Your selection is ''Player vs Computer.''');
                    enter;
                    clrscr;
                        repeat
                        P1;
                            if (Player1=false) and (Draw=false) then
                                begin
                                clrscr;
                                map;
                                writeln;
                                if (table[1]=o) and (table[2]=o) and (table[3]=e)
then c:=3
                                else if (table[1]=o) and (table[3]=o) and
(table[2]=e) then     c:=2
                                else if (table[2]=o) and (table[3]=o) and
(table[1]=e) then     c:=1
```

```
            else if (table[4]=o) and (table[5]=o) and
(table[6]=e) then     c:=6
            else if (table[4]=o) and (table[6]=o) and
(table[5]=e) then     c:=5
            else if (table[5]=o) and (table[6]=o) and
(table[4]=e) then c:=4
            else if (table[7]=o) and (table[8]=o) and
(table[9]=e) then c:=9
            else if (table[7]=o) and (table[9]=o) and
(table[8]=e) then c:=8
            else if (table[8]=o) and (table[9]=o) and
(table[7]=e) then c:=7
            else if (table[1]=o) and (table[4]=o) and
(table[7]=e) then c:=7
            else if (table[1]=o) and (table[7]=o) and
(table[4]=e) then c:=4
            else if (table[4]=o) and (table[7]=o) and
(table[1]=e) then c:=1
            else if (table[2]=o) and (table[5]=o) and
(table[8]=e) then c:=8
            else if (table[2]=o) and (table[8]=o) and
(table[5]=e) then c:=5
            else if (table[5]=o) and (table[8]=o) and
(table[2]=e) then c:=2
            else if (table[3]=o) and (table[6]=o) and
(table[9]=e) then c:=9
            else if (table[3]=o) and (table[9]=o) and
(table[6]=e) then c:=6
            else if (table[6]=o) and (table[9]=o) and
(table[3]=e) then c:=3
            else if (table[1]=o) and (table[5]=o) and
(table[9]=e) then c:=9
            else if (table[1]=o) and (table[9]=o) and
(table[5]=e) then c:=5
            else if (table[5]=o) and (table[9]=o) and
(table[1]=e) then c:=1
            else if (table[3]=o) and (table[5]=o) and
(table[7]=e) then c:=7
```

```
                    else if (table[3]=o) and (table[7]=o) and
(table[5]=e) then c:=5
                    else if (table[5]=o) and (table[7]=o) and
(table[3]=e) then c:=3

                   else if (table[1]=x) and (table[2]=x) and (table[3]=e)
then c:=3
                   else if (table[1]=x) and (table[3]=x) and (table[2]=e)
then c:=2
                   else if (table[2]=x) and (table[3]=x) and (table[1]=e)
then c:=1
                   else if (table[4]=x) and (table[5]=x) and (table[6]=e)
then c:=6
                   else if (table[4]=x) and (table[6]=x) and (table[5]=e)
then c:=5
                   else if (table[5]=x) and (table[6]=x) and (table[4]=e)
then c:=4
                   else if (table[7]=x) and (table[8]=x) and (table[9]=e)
then c:=9
                   else if (table[7]=x) and (table[9]=x) and (table[8]=e)
then c:=8
                   else if (table[8]=x) and (table[9]=x) and (table[7]=e)
then c:=7
                   else if (table[1]=x) and (table[4]=x) and (table[7]=e)
then c:=7
                   else if (table[1]=x) and (table[7]=x) and (table[4]=e)
then c:=4
                   else if (table[4]=x) and (table[7]=x) and (table[1]=e)
then c:=1
                   else if (table[2]=x) and (table[5]=x) and (table[8]=e)
then c:=8
                   else if (table[2]=x) and (table[8]=x) and (table[5]=e)
then c:=5
                   else if (table[5]=x) and (table[8]=x) and (table[2]=e)
then c:=2
                   else if (table[3]=x) and (table[6]=x) and (table[9]=e)
then c:=9
                   else if (table[3]=x) and (table[9]=x) and (table[6]=e)
```

```
then c:=6
                    else if (table[6]=x) and (table[9]=x) and (table[3]=e)
then c:=3
                    else if (table[1]=x) and (table[5]=x) and (table[9]=e)
then c:=9
                    else if (table[1]=x) and (table[9]=x) and (table[5]=e)
then c:=5
                    else if (table[5]=x) and (table[9]=x) and (table[1]=e)
then c:=1
                    else if (table[3]=x) and (table[5]=x) and (table[7]=e)
then c:=7
                    else if (table[3]=x) and (table[7]=x) and (table[5]=e)
then c:=5
                    else if (table[5]=x) and (table[7]=x) and (table[3]=e)
then c:=3
                     else begin
                            repeat

                            c:=random(9)+1;
                            until (table[c]=e);
                        end;
                     CompNumb;
                     if (table[1]=o) and (table[2]=o) and (table[3]=o) or
                        (table[4]=o) and (table[5]=o) and (table[6]=o)
or
                        (table[7]=o) and (table[8]=o) and (table[9]=o)
or
                        (table[1]=o) and (table[4]=o) and (table[7]=o)
or
                        (table[2]=o) and (table[5]=o) and (table[8]=o)
or
                        (table[3]=o) and (table[6]=o) and (table[9]=o)
or
                        (table[1]=o) and (table[5]=o) and (table[9]=o)
or
                        (table[3]=o) and (table[5]=o) and (table[7]=o)
then
                         begin
```

```
                                        clrscr;
                                        map;
                                        writeln('Computer has won');
                                        enter;
                                        clrscr;
                                        cpoints:=cpoints+1;
                                        Comp:=true;
                                        end;
                                    end;
                        until (Player1=true) or (Comp=true) or (Draw=true);
                    end
                    else  wrongbutton;
                end;
    a := a + 1;
    end;
//until (yesorno='j') or (yesorno='k');

end;



procedure game1;
var
    ans: integer;
    guess: integer;
    x:integer;
    y:integer;
    z:integer;

      level:char;
begin
    clrscr;
    writeln('welcome to Number Guessing!');
    writeln;
    writeln('Rules:');
    writeln('1. In this game, you have to guess a number depending on the
difficulty');
    writeln('      you will be told whether the answer is bigger than or lower than
your guess');
```

```
    writeln('2. It will automatically back to the game menu when you played this
game twice');
    writeln;
    writeln('Let''s enjoy the game!');
    writeln('You can choose a level');
    writeln;
    writeln('1) 1-10');
    writeln('2) 1-100');
    writeln('3) 1-1000');
    level:='0';

    while ((level <> '1') and (level <> '2') and (level <>'3'))do
    begin
    writeln;
         writeln('PLease select your level (1-3)');
         level:=readkey;
    end;


    if    (level= '1')
    then begin
    clrscr;
    randomize;
    ans:= random(10)+1;
    x:=0;
    z:=1;
    y:=10;
    writeln('The difficulty is 1-10');
    repeat
    writeln('Your guess:');
    read(guess);
    if (guess <= 0) or (guess>=11) or (guess<x) or (guess>y) then begin
    writeln('Incorrect input!Please try again!');
    writeln;
    write('Your guess:');
    read(guess);
    end;
  if (guess >= 0) and (guess<=11) and (guess>x) and (guess<y) then begin
```

```
    if guess > ans    then begin
      clrscr;
      writeln('The answer is lower than the guess. Please try again!');
      y:=guess;
      writeln(x ,'=< ? =<', y);
      z:=z+1;
      end;
    if guess < ans    then begin
      clrscr;
      writeln('The answer is bigger than the guess. Please try again!');
      x:=guess;
      writeln(x ,'=< ? =<', y);
      z:=z+1;
      end;
    end;
    until guess = ans;
    writeln('Congratulations! You guess the right number! The answer is ',(ans));
    writeln('You have guessed ', z , ' times.');
    readln;
    readln;


end;
      if    (level= '2')
    then begin
    clrscr;
    randomize;
    ans:= random(100)+1;
    x:=0;
    z:=1;
    y:=100;

    writeln('The difficulty is 1-100');
   repeat
    writeln('Your guess:');
    read(guess);
    if (guess <= 0) or (guess>=101) or (guess<x) or (guess>y) then begin
    writeln('Incorrect input!Please try again!');
```

```
      writeln;
      write('Your guess:');
      read(guess);
      end;
      if (guess >= 0) and (guess <= 101) and (guess>x) and (guess<y) then begin
      if guess > ans    then begin
         clrscr;
         writeln('The answer is lower than the guess. Please try again!');
         y:=guess;
         writeln(x ,'=< ? =<', y);
         z:=z+1;
         end;
      if guess < ans    then begin
         clrscr;
         writeln('The answer is bigger than the guess. Please try again!');
         x:=guess;
         writeln(x ,'=< ? =<', y);
         z:=z+1;
         end;
       end;
      until guess = ans;
      writeln('Congratulations! You guess the right number! The answer is ',(ans));
      writeln('You have guessed ', z , ' times.');
      readln;
       readln;
end;
      if   (level= '3')
      then begin
      clrscr;
      randomize;
      ans:= random(1000)+1;
      x:=0;
      z:=1;
      y:=1000;
      writeln('The difficulty is 1-1000');
      repeat
      writeln('Your guess:');
      read(guess);
      if (guess <= 0) or (guess>=1001) or (guess<x) or (guess>y) then begin
      writeln('Incorrect input!Please try again!');
      writeln;
      write('Your guess:');
      read(guess);
      end;
      if (guess >= 0) and (guess<=1001) and (guess>x) and (guess<y) then begin
      if guess > ans    then begin
         clrscr;
         writeln('The answer is lower than the guess. Please try again!');
         y:=guess;
         writeln(x ,'=< ? =<', y);
         z:=z+1;
         end;
      if guess < ans    then begin
         clrscr;
         writeln('The answer is bigger than the guess. Please try again!');
         x:=guess;
         writeln(x ,'=< ? =<', y);
         z:=z+1;
         end;
      end;


      until guess = ans;
      writeln('Congratulations! You guess the right number! The answer is ',(ans));
      writeln('You have guessed ', z , ' times.');
      readln;
        readln;
 end;
 end;




      procedure callgame;
      begin
        choice:=readkey;
```

```
     if (choice='1')
     then
          begin
          game1;
        end
   else
   if    (choice='2')
   then
          begin
          clrscr;
          game2;
        end
   else
   if    (choice='3')
   then
       begin
        clrscr;
          writeln('Notice and reminder of the Sudoku Game');
          writeln;
          writeln('1. The default numbers are adjustable. But once the default
numbers are ');
          writeln('      adjusted, even the whole sudoku is alright, or even the
number is ');
          writeln('      changed back to the original one, the game will not be
counted as a ');
          writeln('      finished game and restart is the only way to solve this
situation');
          writeln;
          writeln('2. If you press wrong button and changed the original number, it
is a ');
          writeln('      suggestion for you that press <Esc> to restart the game');
          writeln;
          writeln('3. The number are randomized every new game so you can try
countless time');
          writeln;
          writeln('Let''s enjoy the game :)');
          writeln('Please press <Enter> to continue');
          writeln('Join the game and press <Esc> twice can back to the menu');
        readln;
        clrscr;
        game3;
      end
else repeat
    ClrScr;
    writeln('incorrect input, please type again!');
    writeln('Please type 1-3 to select a game');
    writeln;
    writeln;
    writeln('1) Number Guessing');
    writeln;
    writeln('2) Tic-Tac-Toe');
    writeln;
    writeln('3) sudoku');
    writeln;
    writeln;
    writeln('your choice is game ');
    choice :=readkey

  until (choice='1') or (choice='2') or (choice='3');
  if (choice='1')
  then
        begin
        game1;
      end
else
if    (choice='2')
then
        begin
        a := 0;
        clrscr;
        begin
        game2;
        end;
      end
else
if    (choice='3')
```

```
    then
        begin
          clrscr;
            writeln('Notice and reminder of the Sudoku Game');
            writeln;
            writeln('1. The default numbers are adjustable. But once the default
numbers are ');
            writeln('      adjusted, even the whole sudoku is alright, or even the
number is ');
            writeln('      changed back to the original one, the game will not be
counted as a ');
            writeln('      finished game and restart is the only way to solve this
situation');
            writeln;
            writeln('2. The restart button is <ESC>, but it is a reminder that you can
only restart');
            writeln('      ONCE only or the program will be shutted down if <ESC> is
pressed more than');
            writeln('      once');
            writeln;
            writeln('3. The number are randomized every new game so you can try
countless time');
            writeln;
            writeln('Let''s enjoy the game :)');
            writeln('Please press <Enter> to continue');
            writeln('Join the game and press <Esc> twice can back to the menu');
            readln;
            clrscr;
            game3;
          end;
      end;



begin
        userbg;
        for k:= 1 to 20 do
        begin
    menu;
    callgame;
    k:= k + 1;
    end;

end.
```

4.2 Reference

1.Free Pascal wiki
  http://wiki.freepascal.org/
2. Wikipedia Dev-Pascal
  https://en.wikipedia.org/wiki/Dev-Pascal
3. ICT Textbooks Program Development (D1, D2)
4. ICT Textbooks Compulsory 3

4.3 Working schedule

Date	Event
April-2017	Choice of Topic
May-2017	Background research + Define the objectives + Propose functions
June-2017	Design of solution
Oct-2017	Implementation
Nov-2017	Testing & Evaluation
Dec-2017& Jan-2018	Conclusion & Discussion + Final Report