Project on

Guess Game Featured by Nim Game

Team Members

Md. Al Amin Bhuiyan
Id: 1804084

Md Mehedi Hassan
Id:1804106

Md. Anamul Hasan Rony
Id: 1804118

Md. Naeem
Id: 1804120

Course Teacher

Md. Mynul Hasan
Lecturer
Department of Computer Science & Engineering,
Chittagong University Engineering & Technology, Bangladesh.

Animesh Chandra Roy
Assistant Professor
Department of Computer Science & Engineering,
Chittagong University Engineering & Technology, Bangladesh.

Introduction:

A game of guessing the hidden numbers. There are two parts of the game:

Puzzle Game:
Puzzle game includes nine hidden buttons. The user has to find the targeted number from these buttons.

Nim Game:
The user has ten steps to find the hidden number. So the user has to search for the optimal number for each step.

Motivation:

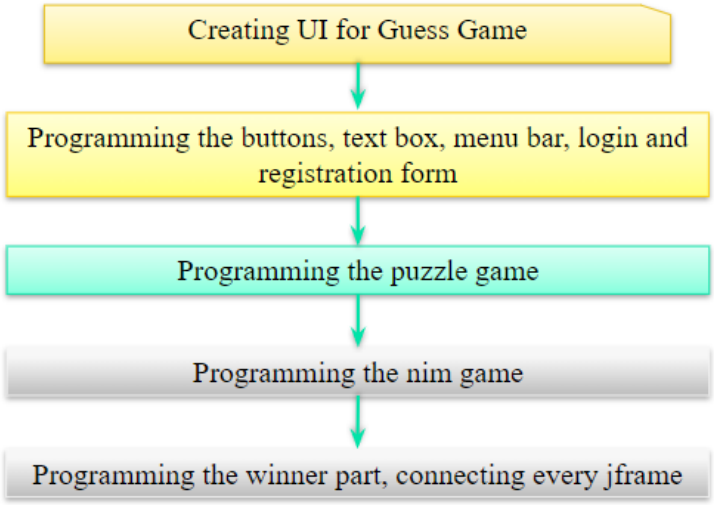
- Guess game sharpens the spatial visual intelligence.
- This game will sharpens memory.
- Puzzle Game sharpens the Little One's fine motor skills.
- Nim Game trains eye and hand coordination.
- Nim Game sharpens the ability to find solutions to problems.

Application:

- Puzzle game will help to increase guess power. It is full of suspense and uncertainty.

- The Nim game is created based on binary search. So in each step the player must choose the optimal answer otherwise the player might lost in the worst case

Flow Chart:



Function Description:

We used Java swing to create the game. For the UI components we used swing containers and swing control. For IDE we used NetBeans.
The functions we used are listed below:

Javax.swing.jFrame:
Jframe is a class type of container. It works like the main window where the swing containers and controls are added.
Javax.swing.JPanel:
Jpanel is a generic lightweight container. Jpanel is used here to enrich theUI. The label is hence used on JPanel.
Javax.swing.JLabel:
Label is used to name the game here. We have used Guess Game,Puzzle game, Nim Game etc as level here.
Javax.Swing.JComboBox:
Two combobox is used here for the game.
Javax.swing.JTextField:
JTextField is used to take input and show output. Here we have two JTextField. One textfield is editable and the other is non editable. The editable textfield is for taking input. The non-editable textfield is for showing output.
Javax.swing.JButton:
We used buttons here. Clear button, Exit button, Guess button, Puzzle buttons and so on.
Javax.swing.JMenu jMenu:

This is used to make a menu system in the game to easily access every possible frame of the game.

Action perform:

Connection:

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
        login supf=new login();
        this.dispose();
        supf.setVisible(true);
        supf.pack();
        supf.setLocationRelativeTo(null);
        supf.setDefaultcloseOperation(JFrame.EXIT ON CLOSE);
}
```

Exit:

```
    Generated Code

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        System.exit(0);
    }
```

Clear:

```
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1.setText("");
        jTextField2.setText("");
    }
```

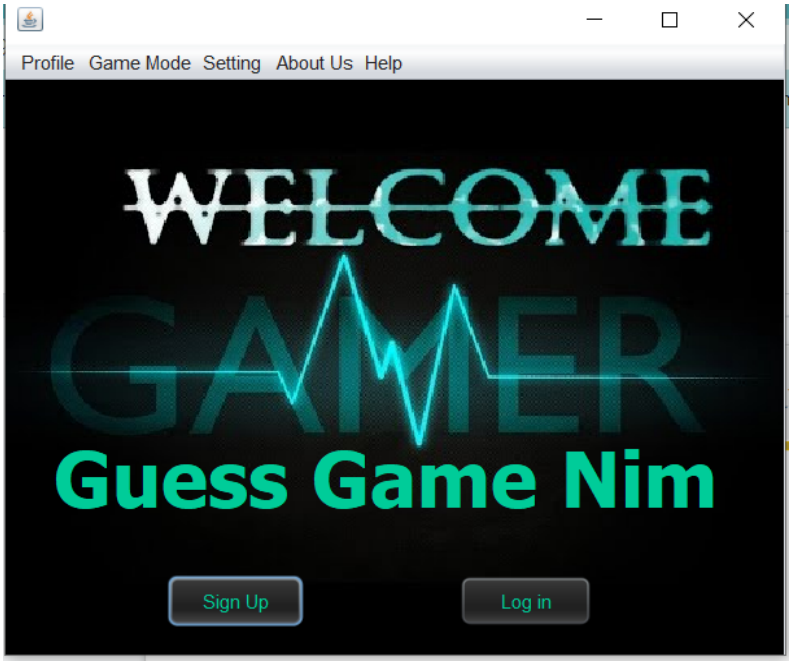
Puzzle Game Resulting Code:

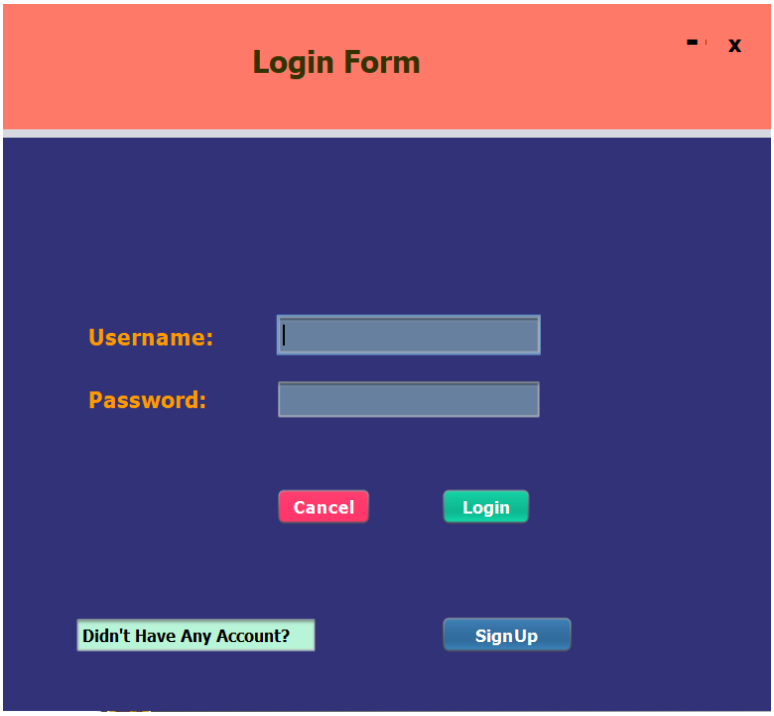
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int newNumber = randomNumberGenerator.nextInt(10);
    cnt++;
    if(cnt==9)
        jButton1.setText(String.valueOf(ans));
    else
        jButton1.setText(String.valueOf(newNumber));
    Score.setText(String.valueOf(cnt));
    if(cnt==9||ans==newNumber)
    {
        JOptionPane.showMessageDialog(null,"Congratualtions, You have won");
        Winner supf=new Winner();
         this.dispose();
         supf.setVisible(true);
         supf.pack();
         supf.setLocationRelativeTo(null);
         supf.setDefaultcloseOperation(JFrame.EXIT_ON_CLOSE);

        //JOptionPane.showMessageDialog(null,"Congratualtions, You have won");
    }
```

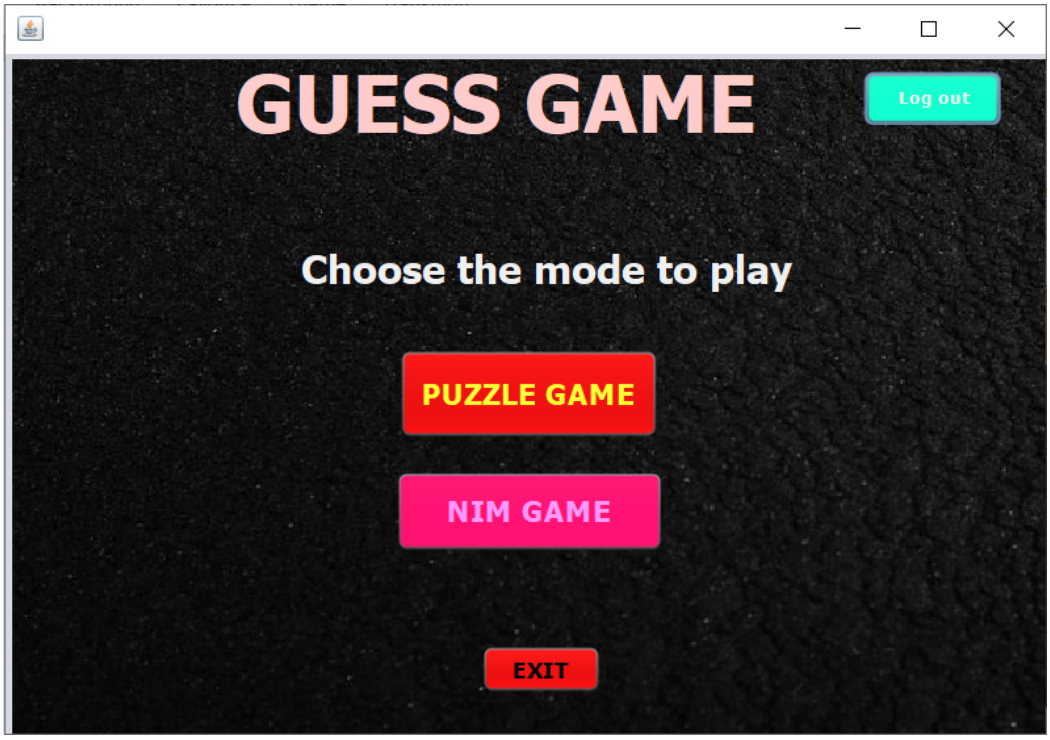
NIM Game Resulting code:

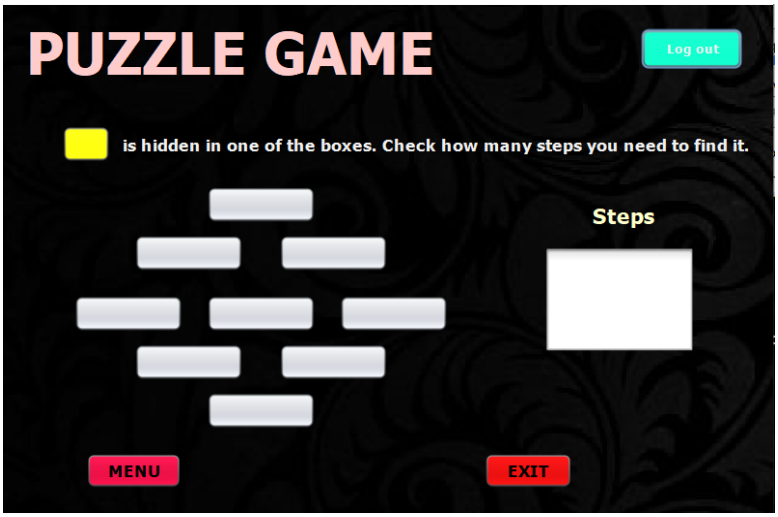
```
private void GuessMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int Guess=Integer.parseInt(txtUserGuess.getText());
    int temp=ans;
    cnt--;
    Score.setText(String.valueOf(cnt));
    if(Guess==temp)
    {
        JOptionPane.showMessageDialog(null,"Shabbash, You guessed the correct number");
        Winner supf=new Winner();
        this.dispose();
        supf.setVisible(true);
        supf.pack();
        supf.setLocationRelativeTo(null);
        supf.setDefaultcloseOperation(JFrame.EXIT_ON_CLOSE);

    }
    else if(Guess>temp)
    {JOptionPane.showMessageDialog(null,"The Number is lower than your guess");}
    else if(Guess<temp)
    {JOptionPane.showMessageDialog(null,"The Numeber is higher than your guess");}
    if(cnt==0)
    {
        JOptionPane.showMessageDialog(null,"OW NOOOOOOOO!! You have no steps left");
        MenuBar supf=new MenuBar();
    this.dispose();
    supf.setVisible(true);
    supf.pack();
    supf.setLocationRelativeTo(null);
    supf.setDefaultcloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

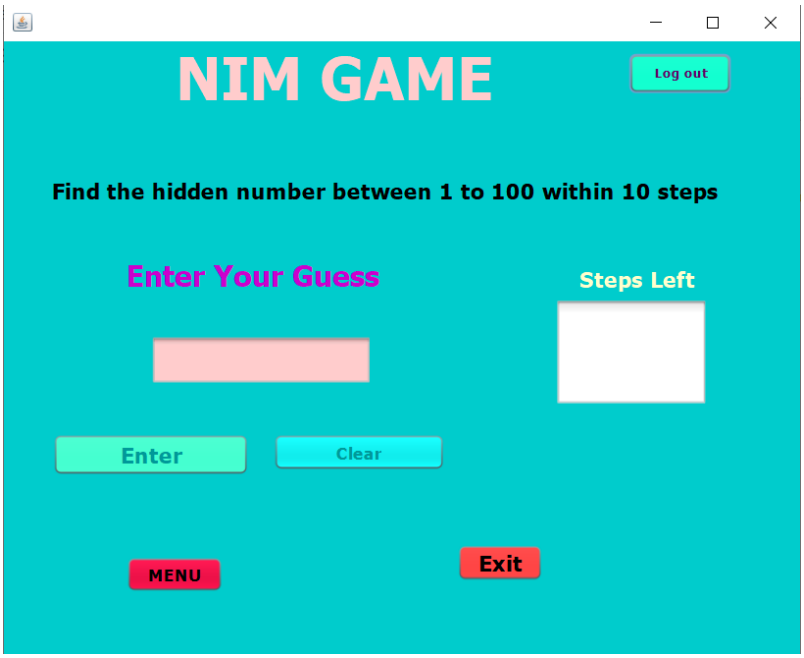
Project Output:

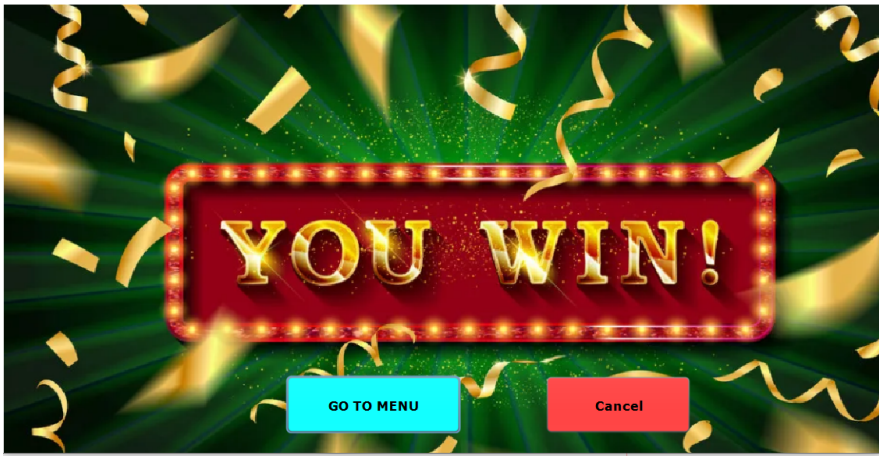



Login Form
Username:
Password:
Cancel
Login
Didn't Have Any Account?
SignUp


GUESS GAME
Log out
Choose the mode to play
PUZZLE GAME
NIM GAME
EXIT


PUZZLE GAME
Log out
is hidden in one of the boxes. Check how many steps you need to find it.
Steps
MENU
EXIT


NIM GAME
Log out
Find the hidden number between 1 to 100 within 10 steps
Enter Your Guess
Steps Left
Enter
Clear
MENU
Exit


YOU WIN!
GO TO MENU
Cancel

Conclusion:

- This game will improve the analytical skills of a player.
- Our game will help to understand nim game and optimal way to take every step.
- The game is not only for entertainment purposes but also improves our guess and thinking capabilities.

The End