

Project 8 - Frown

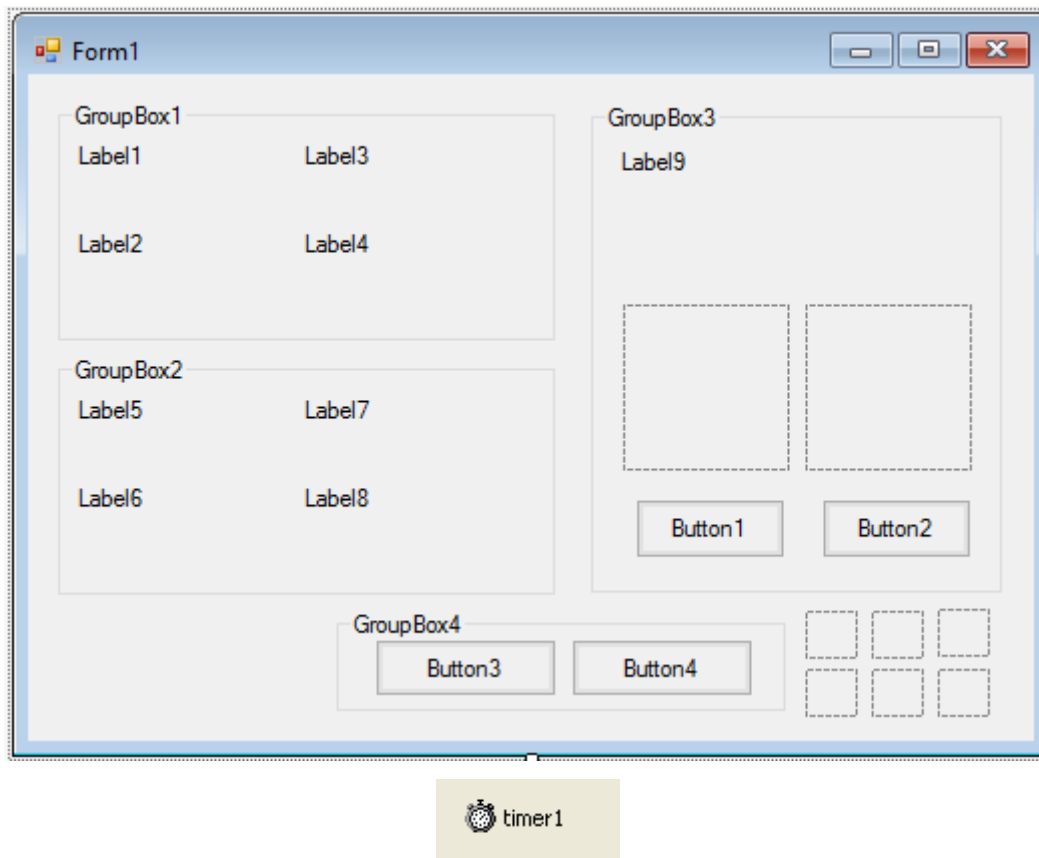
Project Design

Frown is a fun two-player dice game you play against the computer. You play with a set of two dice that are normal except the side of the die where the “1” would be is replaced by a frowning face. The object of the game is to achieve a score of 100 points. Players alternate turns, which consist of a series of at least one roll of the dice, perhaps many, subject to the following rules.

As long as no frown appears on either die, the roller builds a running score for the current turn. After each roll with no frown, the player can choose to continue rolling or pass the dice to the other player. If the player passes the dice, the current score is added to any previous total. If a frown appears, the player loses the points gained on the current turn. If two frowns appear, the player loses the current points and all saved points! There is a considerable amount of luck involved. However, the skill of deciding when to pass the dice to your opponent also figures prominently. The project you are about to build is saved as **Frown** in the project folder (**\BeginVCS\BVCS Projects**).

Place Controls on Form

Start a new project in Visual C#. There are lots of controls here. Place two group box controls on the form. Place four label controls in each of the group boxes. Put two buttons below the second group box. Add a panel control – in this panel, place a label, two picture box controls, and two buttons. Finally, add a timer control and six small picture box controls to the form. Set **AutoSize** to **False** for each label to allow resizing. When done, your form should look something like this:



Set Control Properties

Set the control properties using the properties window:

Form1 Form:

Property Name	Property Value
Text	Frown
FormBorderStyle	FixedSingle
StartPosition	CenterScreen

groupBox1 Group Box:

Property Name	Property Value
Name	grpYou
Text	You
BackColor	Blue
ForeColor	Yellow
Font Size	12
Font Style	Bold

label1 Label:

Property Name	Property Value
Text	Score This Turn
ForeColor	White
Font Size	10

label2 Label:

Property Name	Property Value
Text	Total Score
ForeColor	White
Font Size	10

label3 Label:

Property Name	Property Value
Name	lblYouScore
AutoSize	False
Text	[Blank]
TextAlign	MiddleCenter
BorderStyle	Fixed3D
BackColor	White
ForeColor	Black
Font Size	12

label4 Label:

Property Name	Property Value
Name	lblYouTotal
AutoSize	False
Text	0
TextAlign	MiddleCenter
BorderStyle	Fixed3D
BackColor	White
ForeColor	Black
Font Size	12

groupBox2 Group Box:

Property Name	Property Value
Name	grpComputer
Text	Computer
BackColor	Blue
ForeColor	Yellow
Font Size	12
Font Style	Bold

label5 Label:

Property Name	Property Value
Text	Score This Turn
TextAlign	MiddleLeft
ForeColor	White
Font Size	10

label6 Label:

Property Name	Property Value
Text	Total Score
TextAlign	MiddleLeft
ForeColor	White
Font Size	10

label7 Label:

Property Name	Property Value
Name	lblComputerScore
AutoSize	False
Text	[Blank]
TextAlign	MiddleCenter
BorderStyle	Fixed3D
BackColor	White
ForeColor	Black
Font Size	12

label8 Label:

Property Name	Property Value
Name	lblComputerTotal
AutoSize	False
Text	0
TextAlign	MiddleCenter
BorderStyle	Fixed3D
BackColor	White
ForeColor	Black
Font Size	12

panel1 Panel:

Property Name	Property Value
Name	pnlDice
BackColor	Red

label9 Label:

Property Name	Property Value
Name	lblMessage
Text	[Blank]
TextAlign	MiddleCenter
BorderStyle	Fixed3D
BackColor	Light Yellow
Font Size	10

pictureBox1 Picture Box:

Property Name	Property Value
Name	picDice1
BackColor	Green
SizeMode	StretchImage

pictureBox2 Picture Box:

Property Name	Property Value
Name	picDice2
BackColor	Green
SizeMode	StretchImage

button1 Button:

Property Name	Property Value
Name	btnRoll
Text	Roll Dice
BackColor	Light Red
Enabled	False

button2 Button:

Property Name	Property Value
Name	btnPass
Text	Pass Dice
BackColor	Light Red
Enabled	False

button3 Button:

Property Name	Property Value
Name	btnNew
Text	New Game

button4 Button:

Property Name	Property Value
Name	btnStop
Text	Exit

timer1 Timer:

Property Name	Property Value
Name	timComputer
Interval	2000

pictureBox3 PictureBox:

Property Name	Property Value
Name	picDots1
Image	frown.gif (in the \BeginVCS\BVCS Projects\Frown folder)
SizeMode	StretchImage
Visible	False

pictureBox4 PictureBox:

Property Name	Property Value
Name	picDots2
Image	dice2.gif (in the \BeginVCS\BVCS Projects\Frown folder)
SizeMode	StretchImage
Visible	False

pictureBox5 PictureBox:

Property Name	Property Value
Name	picDots3
Image	dice3.gif (in the \BeginVCS\BVCS Projects\Frown folder)
SizeMode	StretchImage
Visible	False

pictureBox6 PictureBox:

Property Name	Property Value
Name	picDots4
Image	dice4.gif (in the \BeginVCS\BVCS Projects\Frown folder)
SizeMode	StretchImage
Visible	False

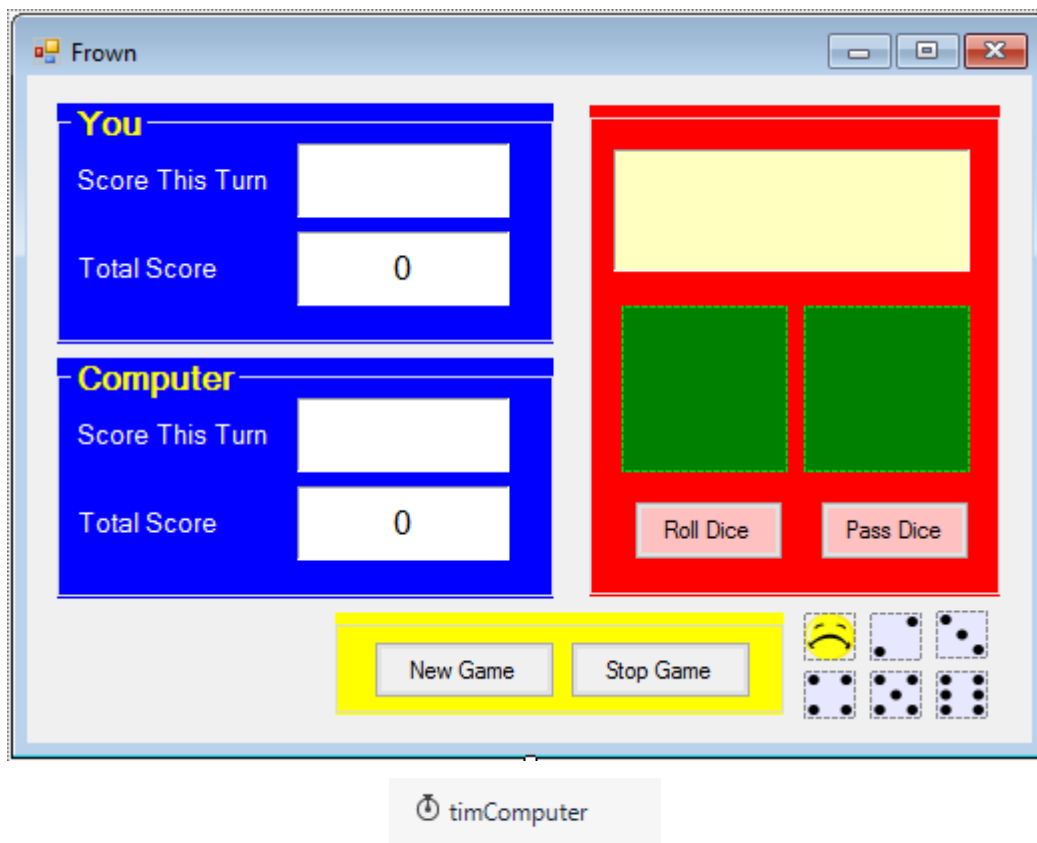
pictureBox7 PictureBox:

Property Name	Property Value
Name	picDots5
Image	dice5.gif (in the \BeginVCS\BVCS Projects\Frown folder)
SizeMode	StretchImage
Visible	False

pictureBox8 PictureBox:

Property Name	Property Value
Name	picDots6
Image	dice6.gif (in the \BeginVCS\BVCS Projects\Frown folder)
SizeMode	StretchImage
Visible	False

When done setting properties, my form looks like this:



Write Event Methods

Most of the code is involved with randomly rolling the two dice and passing control of the game from one player to the other. Study the logic carefully – it is used in many games where the human plays against the computer.

Add this code to the **general declarations** area:

```
bool gameOver;  
int whoseTurn, dice1, dice2;  
int youScore, computerScore;  
int youTotal, computerTotal;  
const int win = 100;  
Random myRandom = new Random();
```

The **Form1_Load** event method:

```
private void Form1_Load(object sender, EventArgs e)  
{  
    // Initialize dice to frowns  
    lblMessage.Text = "Click New Game To Start";  
    picDice1.Image = picDots1.Image;  
    picDice2.Image = picDots1.Image;  
    btnNew.Focus();  
}
```

The `btnNew_Click` event method:

```
private void btnNew_Click(object sender, EventArgs e)
{
    // Start new game
    gameOver = false;
    lblMessage.Text = "";
    btnNew.Enabled = false;
    btnStop.Text = "Stop Game";
    youScore = 0;
    lblYouScore.Text = "";
    computerScore = 0;
    lblComputerScore.Text = "";
    youTotal = 0;
    lblYouTotal.Text = "0";
    computerTotal = 0;
    lblComputerTotal.Text = "0";
    if (myRandom.Next(2) == 0)
    {
        // Computer goes first
        whoseTurn = 0;
        lblComputerScore.Text = "0";
        lblMessage.Text = "I'll roll first.";
        // must call instead of performclick since button is
        // is not enabled
        btnRoll_Click(null, null);
    }
    else
    {
        // You go first
        whoseTurn = 1;
        lblYouScore.Text = "0";
        lblMessage.Text = "You roll first.";
        btnRoll.Enabled = true;
        btnRoll.Focus();
    }
}
```

The **btnStop_Click** event method:

```
private void btnStop_Click(object sender, EventArgs e)
{
    if (btnStop.Text == "Exit")
    {
        this.Close();
    }
    else
    {
        // Stop current game
        timComputer.Enabled = false;
        btnNew.Enabled = true;
        btnStop.Text = "Exit";
        btnRoll.Enabled = false;
        btnPass.Enabled = false;
        if (!gameOver)
        {
            lblMessage.Text = "Game Stopped";
        }
        btnNew.Focus();
    }
}
```

The **btnRoll_Click** event method:

```
private void btnRoll_Click(object sender, EventArgs e)
{
    // Dice rolling
    // Roll Dice 1 and set display
    dice1 = myRandom.Next(6) + 1;
    switch (dice1)
    {
        case 1:
            picDice1.Image = picDots1.Image;
            break;
        case 2:
            picDice1.Image = picDots2.Image;
            break;
        case 3:
            picDice1.Image = picDots3.Image;
            break;
        case 4:
            picDice1.Image = picDots4.Image;
```

```

        break;
    case 5:
        picDice1.Image = picDots5.Image;
        break;
    case 6:
        picDice1.Image = picDots6.Image;
        break;
}
// Roll Dice 2 and set display
dice2 = myRandom.Next(6) + 1;
switch (dice2)
{
    case 1:
        picDice2.Image = picDots1.Image;
        break;
    case 2:
        picDice2.Image = picDots2.Image;
        break;
    case 3:
        picDice2.Image = picDots3.Image;
        break;
    case 4:
        picDice2.Image = picDots4.Image;
        break;
    case 5:
        picDice2.Image = picDots5.Image;
        break;
    case 6:
        picDice2.Image = picDots6.Image;
        break;
}
picDice1.Refresh();
picDice2.Refresh();
if (whoseTurn == 0)
{
    // Computer rolled
    if (dice1 > 1 && dice2 > 1)
    {
        // No frowns
        computerScore = computerScore + dice1 + dice2;
        lblComputerScore.Text =
Convert.ToString(computerScore);
        timComputer.Enabled = true;
        lblMessage.Text = lblMessage.Text + " Let me
think ...";
        return;
    }
}

```

```
        else if (dice1 == 1 && dice2 == 1)
        {
            // Two frowns - lose everything - must pass
            lblMessage.Text = lblMessage.Text + "\r\nI lost
all my points!\r\nYour turn.";
            computerTotal = 0;
            lblComputerTotal.Text = "0";
        }
        else
        {
            // One frown - must pass
            lblMessage.Text = lblMessage.Text + "\r\nI lost
my turn.\r\nYour turn.";
        }
        computerScore = 0;
        lblComputerScore.Text = "";
        whoseTurn = 1;
        btnRoll.Enabled = true;
        btnRoll.Focus();
    }
    else
    {
        // You rolled
        lblMessage.Text = "Still your turn.";
        btnPass.Enabled = true;
        if (dice1 > 1 && dice2 > 1)
        {
            // No frowns
            youScore = youScore + dice1 + dice2;
            lblYouScore.Text = Convert.ToString(youScore);
        }
        else if (dice1 == 1 && dice2 == 1)
        {
            // Two frowns - lose everything - must pass
            youScore = 0;
            youTotal = 0;
            lblMessage.Text = "You lost everything.\r\nYou
must pass to me.";
            btnRoll.Enabled = false;
            btnPass.Focus();
        }
    }
}
```

```

        else
        {
            // One frown - must pass
            youScore = 0;
            lblMessage.Text = "You lost your turn.\r\nYou
must pass to me.";
            btnRoll.Enabled = false;
            btnPass.Focus();
        }
    }
}

```

The **btnPass_Click** event method:

```

private void btnPass_Click(object sender, EventArgs e)
{
    // You passed dice to computer
    btnRoll.Enabled = false;
    btnPass.Enabled = false;
    whoseTurn = 0;
    youTotal = youTotal + youScore;
    youScore = 0;
    lblYouScore.Text = "";
    lblYouTotal.Text = Convert.ToString(youTotal);
    if (youTotal >= win)
    {
        gameOver = true;
        lblMessage.Text = "You win!!";
        btnStop.PerformClick();
    }
    else
    {
        lblMessage.Text = "I'll roll now.";
        // call btnroll routine, we can't use performclick
        // method since button is not enabled at this point
        btnRoll_Click(null, null);
    }
}

```

The `timComputer_Timer` event method:

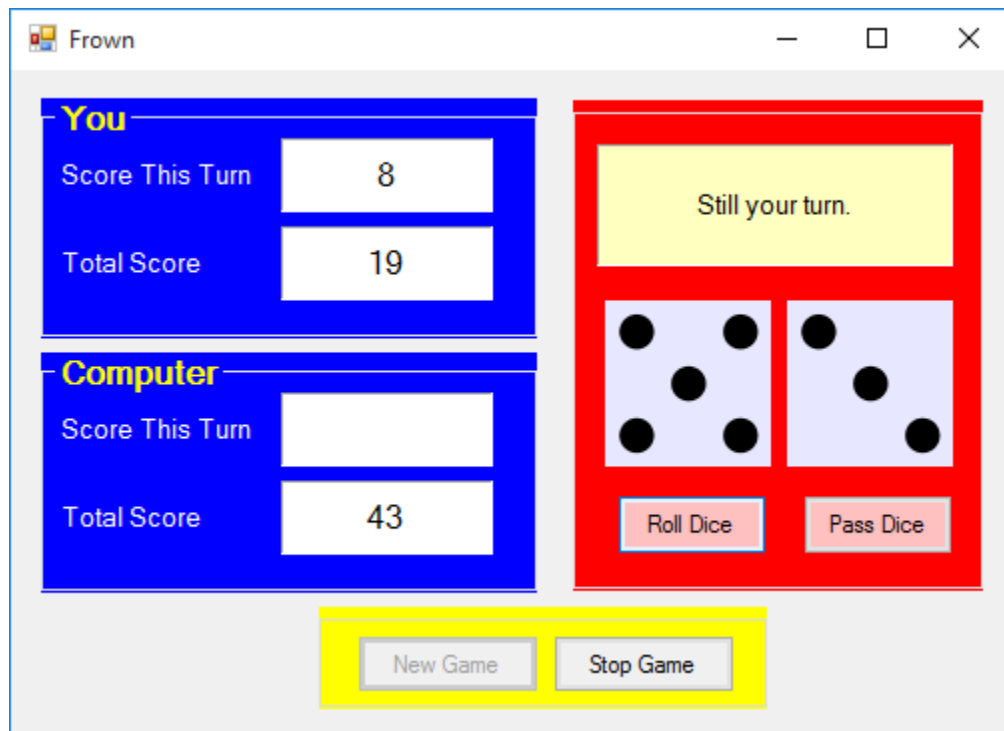
```
private void timComputer_Tick(object sender, EventArgs e)
{
    int v;
    int odds;
    // Computer turn - decide wheter to roll again or pass
    timComputer.Enabled = false;
    v = computerScore + computerTotal;
    if (v >= win)
    {
        // Computer wins!
        gameOver = true;
        lblComputerTotal.Text = Convert.ToString(v);
        lblMessage.Text = "I win!!";
        btnStop.PerformClick();
        return;
    }
    else if (win - youTotal <= 10)
    {
        // If you are close to win, computer rolls again
        lblMessage.Text = "I'll roll again.";
        btnRoll_Click(null, null);
    }
    else
    {
        if (computerTotal >= youTotal)
        {
            // If computer already ahead, less likely to roll
again
            odds = (int) (100 * ((double) computerScore) /
30.0);
        }
        else if (v < youTotal)
        {
            // If computer behind, more likely
50.0);
            odds = (int) (100.0 * ((double)computerScore) /
50.0);
        }
        else
        {
            odds = (int) (100.0 * ((double)computerScore) /
40.0);
        }
        if (myRandom.Next(100) > odds)
        {
```



```
        lblMessage.Text = "I'll roll again.";
        btnRoll_Click(null, null);
    }
    else
    {
        // Stick with roll and pass
        lblMessage.Text = "I pass to you.\r\nYour turn.";
        computerScore = 0;
        computerTotal = v;
        lblComputerTotal.Text =
Convert.ToString(computerTotal);
        lblComputerScore.Text = "";
        whoseTurn = 1;
        btnRoll.Enabled = true;
        btnRoll.Focus();
    }
}
}
```

Run the Project

Save your work. Run the project. You should figure out the game fairly quickly. The computer will decide who goes first. When its your turn, click 'Roll Dice'. After each roll, decide whether to roll again or pass the dice to the computer (click 'Pass Dice'). If you get a frown on any roll, your score will be adjusted accordingly and the dice passed to the computer. When it's the computer's turn, you will watch the computer roll and make its decisions using the same rules. The game is over when either you or the computer has a Total Score of at least 100 points. Click **Stop Game** at any time to stop the game before its end. Here's the middle of a game I played:



Other Things to Try

A first change to Frown would be to make it a two player game - eliminate the computer and play against a friend. You essentially need to have two group boxes and code like that for the human player. You might also like to have an adjustable winning score.

The computer logic used by the program is fairly simple – when it is far behind it tends to take more risks. This logic is in the **timComputer_Tick** event method. Study the logic and see if you can improve upon it. Have your computer play someone else's computer.