

Name: **Session:**
Programming II
Lab Exercise 5/11/2023
Elimination Game

Elimination is a one-player game. The board consists of 12 tiles, numbered 1 to 12. The player rolls a pair of dice and removes tiles based on the numbers shown on the dice. For each roll the player can remove either the two tiles corresponding to the numbers shown on the dice or a single tile corresponding to the sum of the numbers on the dice. If a player rolls doubles (the same number on both dice), the player can only remove only the tile corresponding to the sum of the dice. Play continues until the player cannot make a legal move or all of the tiles have been removed. The sum of the remaining tiles is the players score. The goal is to have a low score.

Sample Game:

1	2	3	4	5	6	7	8	9	10	11	12
.

Dice Roll: 1 6 // Player removes both tiles

1	2	3	4	5	6	7	8	9	10	11	12
X	X

Dice Roll: 6 2 //Player removes sum of tiles

1	2	3	4	5	6	7	8	9	10	11	12
X	X	.	X

Dice Roll: 5 5 //Player removes sum of tiles

1	2	3	4	5	6	7	8	9	10	11	12
X	X	.	X	.	X	.	.

Dice Roll: 2 4 //Player removes both tiles

1	2	3	4	5	6	7	8	9	10	11	12
X	X	.	X	.	X	.	X	.	X	.	.

Dice Roll: 3 1 // Player quits since no legal moves

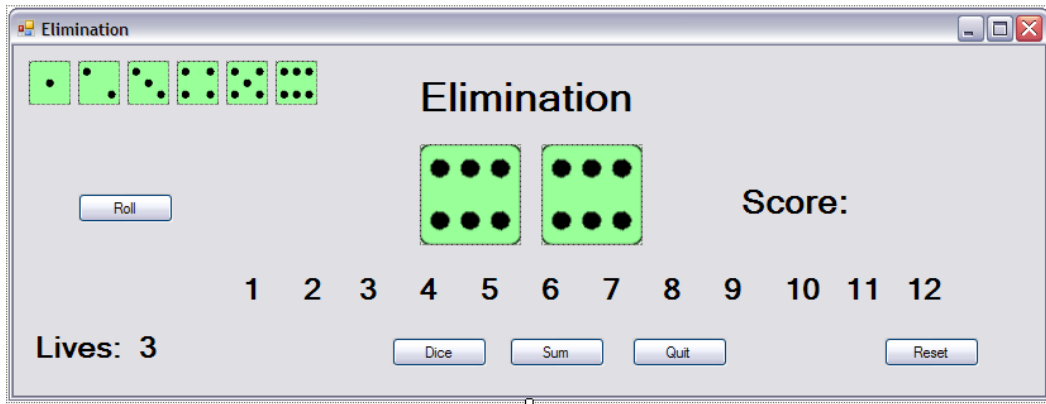
1	2	3	4	5	6	7	8	9	10	11	12
X	X	.	X	.	X	.	X	.	X	.	.

Create a program to play the game of elimination. Repeat the following steps until the user has removed all or the tiles or quit.

- Display the board
- Roll the dice

Let the user pick a move (Dice, Sum, Quit). In this version we will let the player have three illegal moves before the game is ended.

1. Create the following interface Using 2 PictureBox for the dice, 6 small PictureBox, 2 Labels (Score and Lives), 5 Buttons (Roll, Dice, Sum, Quit, and Reset), and 12 Labels (Numbers 1 – 12)



- i. The Dice and Sum buttons enabled property is set to FALSE initially.
 - ii. The 6 small picture boxes in the upper left corner of the form need to be named die1, die2, die3, die4, die5, die6. Note: Use StretchImage Mode for the Image property
 - iii. Their Visible property of die1 – die6 needs to be set to FALSE.
 - iv. The images for die1 – die6 should be placed in the Bin folder and can be found on the server in Summer Camp 2012\images\Elimination Game\.
 - v. You may use either the .bmp files or the .gif files.
 - vi. You will also need to add a Timer.
 - vii. The two PictureBoxes should be named pb1 and pb2
 - viii. Name the Number Labels lbl1, lbl2, lbl3, lbl4, lbl5, lbl6, lbl7, lbl8, lbl9, lbl10, lbl11, and lbl12.
 - ix. Name the buttons btnRoll, btnDice, btnSum, btnQuit, and btnReset.
2. Create the following global variables.

```
int roll1;
int roll2;
int rolls;
int illegalMoves = 0;
bool firstGood = false;
Random r = new Random();
```

3. Add the following to the Form_Load procedure.

```
pb1.Image = die6.Image;  
pb2.Image = die6.Image;
```

4. Add the following code to the btnRoll Click event procedure.

```
timer1.Enabled = true;  
btnRoll.Enabled = false;  
btnDice.Enabled = true;  
btnSum.Enabled = true;  
btnQuit.Enabled = true;
```

5. Add the following code to the timer1_Tick event procedure.

```
rollDie1();  
rollDie2();  
//increment the roll counter  
rolls++;  
//stop after 30 rolls (3 seconds)  
if (rollOver())  
{  
    //reset roll counter  
    rolls = 0;  
    //disable the rolling dice  
    timer1.Enabled = false;  
}
```

6. Create a rollDie1 function

```
private void rollDie1()  
{  
    //generate random roll for die 1  
    roll1 = r.Next(1,6);  
    //display the corresponding image (die 1-6)  
    switch (roll1)  
    {  
        case 1:  
            pb1.Image = die1.Image;  
            break;  
        case 2:  
            pb1.Image = die2.Image;  
            break;  
        case 3:  
            pb1.Image = die3.Image;  
            break;
```

```

        case 4:
            pb1.Image = die4.Image;
            break;
        case 5:
            pb1.Image = die5.Image;
            break;
        case 6:
            pb1.Image = die6.Image;
            break;
    }
}

```

7. Create a rollDie2 function

```

private void rollDie2()
{
    //generate random roll for die 1
    roll2 = r.Next(1,6);
    //display the corresponding image (die 1-6)
    switch (roll2)
    {
        case 1:
            pb2.Image = die1.Image;
            break;
        case 2:
            pb2.Image = die2.Image;
            break;
        case 3:
            pb2.Image = die3.Image;
            break;
        case 4:
            pb2.Image = die4.Image;
            break;
        case 5:
            pb2.Image = die5.Image;
            break;
        case 6:
            pb2.Image = die6.Image;
            break;
    }
}

```

8. Create a rollover function

```
private bool rollover()
{
    if (rolls > 30)
        return true;
    else
        return false;
}
```

9. Add the following code to the btnSum_Click event procedure.

```
btnRoll.Enabled = true;
btnDice.Enabled = false;
btnSum.Enabled = false;
btnQuit.Enabled = false;

switch (roll1 + roll2)
{
    case 2:
        if (lbl2.Text != "X")
            lbl2.Text = "X";
        else
        {
            MessageBox.Show("Illegal move!!!");
            illegalMoves++;
            lblIllegal.Text = "Lives: " + (3 - illegalMoves);
        }
        break;
    case 3:
        if (lbl3.Text != "X")
            lbl3.Text = "X";
        else
        {
            MessageBox.Show("Illegal move!!!");
            illegalMoves++;
            lblIllegal.Text = "Lives: " + (3 - illegalMoves);
        }
        break;
    case 4:
        if (lbl4.Text != "X")
            lbl4.Text = "X";
        else
        {
            MessageBox.Show("Illegal move!!!");
            illegalMoves++;
            lblIllegal.Text = "Lives: " + (3 - illegalMoves);
        }
        break;
}
```

```

case 5:
    if (lbl5.Text != "X")
        lbl5.Text = "X";
    else
    {
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = " Lives: " + (3 - illegalMoves);
    }
    break;
case 6:
    if (lbl6.Text != "X")
        lbl6.Text = "X";
    else
    {
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = " Lives: " + (3 - illegalMoves);
    }
    break;
case 7:
    if (lbl7.Text != "X")
        lbl7.Text = "X";
    else
    {
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = " Lives: " + (3 - illegalMoves);
    }
    break;
case 8:
    if (lbl8.Text != "X")
        lbl8.Text = "X";
    else
    {
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = " Lives: " + (3 - illegalMoves);
    }
    break;

```

```

case 9:
    if (lbl9.Text != "X")
        lbl9.Text = "X";
    else
    {
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = "Lives: " + (3 - illegalMoves);
    }
    break;
case 10:
    if (lbl10.Text != "X")
        lbl10.Text = "X";
    else
    {
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = "Lives: " + (3 - illegalMoves);
    }
    break;
case 11:
    if (lbl11.Text != "X")
        lbl11.Text = "X";
    else
    {
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = "Lives: " + (3 - illegalMoves);
    }
    break;
case 12:
    if (lbl12.Text != "X")
        lbl12.Text = "X";
    else
    {
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = "Lives: " + (3 - illegalMoves);
    }
    break;
} //end of switch (roll1 + roll2)

if (illegalMoves >= 3)
    endGame();

```

10. Add the following code to the btnDice_Click event procedure

```
btnRoll.Enabled = true;
btnDice.Enabled = false;
btnSum.Enabled = false;
btnQuit.Enabled = false;
switch (roll1)
{
    case 1:
        if (lbl1.Text != "X")
        {
            lbl1.Text = "X";
            firstGood = true;
        }
        else
        {
            MessageBox.Show("Illegal move!!!");
            illegalMoves++;
            lblIllegal.Text = " Lives: " + (3 - illegalMoves);
        }
        break;
    case 2:
        if (lbl2.Text != "X")
        {
            lbl2.Text = "X";
            firstGood = true;
        }
        else
        {
            MessageBox.Show("Illegal move!!!");
            illegalMoves++;
            lblIllegal.Text = " Lives: " + (3 - illegalMoves);
        }
        break;
}
```



```

case 3:
    if (lbl3.Text != "X")
    {
        lbl3.Text = "X";
        firstGood = true;
    }
    else
    {
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = " Lives: " + (3 - illegalMoves);
    }
    break;
case 4:
    if (lbl4.Text != "X")
    {
        lbl4.Text = "X";
        firstGood = true;
    }
    else
    {
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = " Lives: " + (3 - illegalMoves);
    }
    break;
case 5:
    if (lbl5.Text != "X")
    {
        lbl5.Text = "X";
        firstGood = true;
    }
    else
    {
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = " Lives: " + (3 - illegalMoves);
    }
    break;

```

```

case 6:
    if (lbl6.Text != "X")
    {
        lbl6.Text = "X";
        firstGood = true;
    }
    else
    {
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = " Lives: " + (3 - illegalMoves);
    }
    break;
} //end of switch roll1

switch (roll2)
{
    case 1:
        if (lbl1.Text != "X")
            lbl1.Text = "X";
        else
        {
            if (firstGood)
            {
                restore(roll1);
                firstGood = false;
                break;
            }
            MessageBox.Show("Illegal move!!!");
            illegalMoves++;
            lblIllegal.Text = " Lives: " + (3 - illegalMoves);
        }
        break;

```

```

case 2:
    if (lbl2.Text != "X")
        lbl2.Text = "X";
    else
    {
        if (firstGood)
        {
            restore(roll1);
            firstGood = false;
            break;
        }
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = " Lives: " + (3 - illegalMoves);
    }
    break;
case 3:
    if (lbl3.Text != "X")
        lbl3.Text = "X";
    else
    {
        if (firstGood)
        {
            restore(roll1);
            firstGood = false;
            break;
        }
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = " Lives: " + (3 - illegalMoves);
    }
    break;

```

```

case 4:
    if (lbl4.Text != "X")
        lbl4.Text = "X";
    else
    {
        if (firstGood)
        {
            restore(roll1);
            firstGood = false;
            break;
        }
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = " Lives: " + (3 - illegalMoves);
    }
    break;

case 5:
    if (lbl5.Text != "X")
        lbl5.Text = "X";
    else
    {
        if (firstGood)
        {
            restore(roll1);
            firstGood = false;
            break;
        }
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = " Lives: " + (3 - illegalMoves);
    }
    break;

```

```

case 6:
    if (lbl6.Text != "X")
        lbl6.Text = "X";
    else
    {
        if (firstGood)
        {
            restore(roll1);
            firstGood = false;
            break;
        }
        MessageBox.Show("Illegal move!!!");
        illegalMoves++;
        lblIllegal.Text = "Lives: " + (3 - illegalMoves);
    }
    break;
} //end of switch roll2

if (illegalMoves >= 3)
    endGame();

```

11. Add the following code to the btnQuit_Click event handler

```
endGame();
```

12. Add the following endGame function:

```

private void endGame()
{
    int score = 0;
    if (lbl1.Text != "X")
        score += Convert.ToInt32(lbl1.Text);
    if (lbl2.Text != "X")
        score += Convert.ToInt32(lbl2.Text);
    if (lbl3.Text != "X")
        score += Convert.ToInt32(lbl3.Text);
    if (lbl4.Text != "X")
        score += Convert.ToInt32(lbl4.Text);
    if (lbl5.Text != "X")
        score += Convert.ToInt32(lbl5.Text);
    if (lbl6.Text != "X")
        score += Convert.ToInt32(lbl6.Text);
    if (lbl7.Text != "X")
        score += Convert.ToInt32(lbl7.Text);
    if (lbl8.Text != "X")
        score += Convert.ToInt32(lbl8.Text);
}

```

```

    if (lbl9.Text != "X")
        score += Convert.ToInt32(lbl9.Text);
    if (lbl10.Text != "X")
        score += Convert.ToInt32(lbl10.Text);
    if (lbl11.Text != "X")
        score += Convert.ToInt32(lbl11.Text);
    if (lbl12.Text != "X")
        score += Convert.ToInt32(lbl12.Text);

    lblScore.Text = "Score: " + score;

    if (illegalMoves >= 3)
        lblIllegal.Text = "You're Toast!!!";
    else
        lblIllegal.Text = "You give up too easily";

    btnRoll.Enabled = false;
    btnReset.Enabled = true;
}

```

13. Add the following restore function:

```

private void restore(int die)
{
    switch (die)
    {
        case 1:
            lbl1.Text = "1";
            break;
        case 2:
            lbl2.Text = "2";
            break;
        case 3:
            lbl3.Text = "3";
            break;
        case 4:
            lbl4.Text = "4";
            break;
        case 5:
            lbl5.Text = "5";
            break;
        case 6:
            lbl6.Text = "6";
            break;
    } //end of switch
}

```

14. Add the following code the btnReset_Click event procedure:

```
btnReset.Enabled = false;
btnRoll.Enabled = true;
lbl1.Text = "1";
lbl2.Text = "2";
lbl3.Text = "3";
lbl4.Text = "4";
lbl5.Text = "5";
lbl6.Text = "6";
lbl7.Text = "7";
lbl8.Text = "8";
lbl9.Text = "9";
lbl10.Text = "10";
lbl11.Text = "11";
lbl12.Text = "12";
illegalMoves = 0;
lblIllegal.Text = "Lives: 3";
lblScore.Text = "Score: ";
timer1.Enabled = false;
```