# Object Oriented Programming
Lab Manual 12

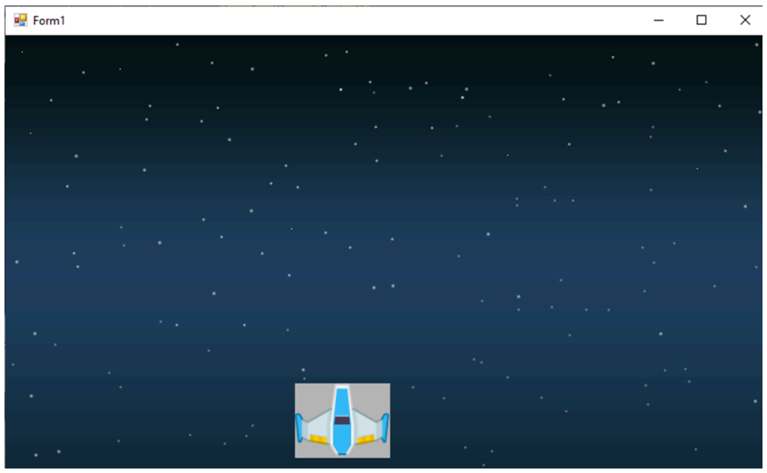
## Introduction
After a week of rigorous coding, Welcome back!

**You have learned all about Custom Controls and Desktop Application Development in the previous lab manuals. Let's move on to the next, new, and exciting concepts.**

In contrast to Object-Oriented Programming, students have another kind of programming paradigm known as **Event-Driven Programming**. Event-driven programming is a programming paradigm in which the flow of program execution is determined by events - for example, a user action such as a mouse click, keypress, or a message from the operating system or another program.

In this Lab, We will implement our first game using .NET Framework for Desktop Application.

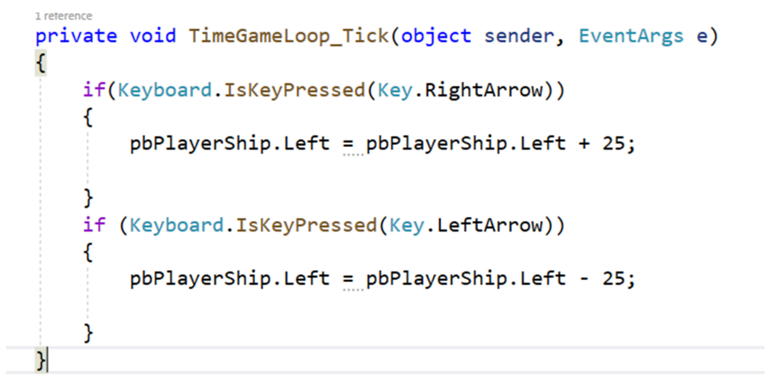
**Let's start the fun and Code.**

| sr | Snapshot | Description |
|---|---|---|
| 1. | Create a New Project by using **Windows Forms(.NET Framework)**. | |
| 2. |  | - Setup the background of the form.<br>- Add a Picture Box that shows the main player<br>Note: you need to have png for the background and main player. |
| We need to take a **continuous input** from the user. Therefore, we need to **add timer** | | |

**control** which will run its event after some time.
Therefore, we shall add a timer, set it **interval time to 500 milliseconds** and **write the code** inside the **Tick Event** of the timer

## Moving the Player

3.



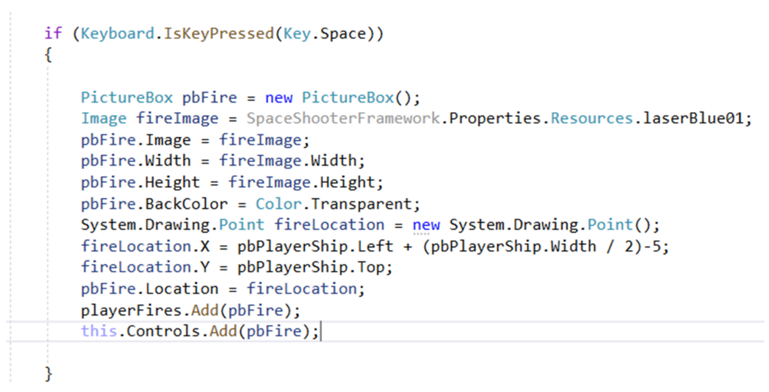
```
1 reference
private void TimeGameLoop_Tick(object sender, EventArgs e)
{
    if(Keyboard.IsKeyPressed(Key.RightArrow))
    {
        pbPlayerShip.Left = pbPlayerShip.Left + 25;

    }
    if (Keyboard.IsKeyPressed(Key.LeftArrow))
    {
        pbPlayerShip.Left = pbPlayerShip.Left - 25;

    }
}
```

Movement of the ship in Left and Right Direction.

Now, in order to remove the **blank space inside the picture box in Player**, set the **BackColor** as **Trasparent** and set the **doubleBuffer** property **True** to remove **Jitter**.

## Add Firing Behavior

4.

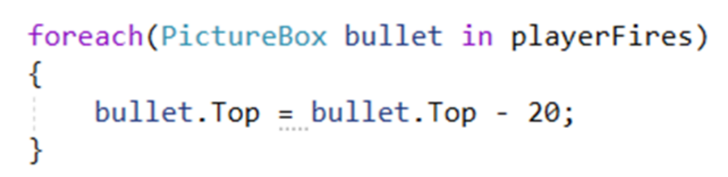


```
if (Keyboard.IsKeyPressed(Key.Space))
{

    PictureBox pbFire = new PictureBox();
    Image fireImage = SpaceShooterFramework.Properties.Resources.laserBlue01;
    pbFire.Image = fireImage;
    pbFire.Width = fireImage.Width;
    pbFire.Height = fireImage.Height;
    pbFire.BackColor = Color.Transparent;
    System.Drawing.Point fireLocation = new System.Drawing.Point();
    fireLocation.X = pbPlayerShip.Left + (pbPlayerShip.Width / 2)-5;
    fireLocation.Y = pbPlayerShip.Top;
    pbFire.Location = fireLocation;
    playerFires.Add(pbFire);
    this.Controls.Add(pbFire);

}
```

On spacebar press, we can create a **pictureBox** at runtime setup it location to the exactly the **middle of the player ship** and add into the list of fires so later on these fires may start moving.

In Order to **move the bullet**, just **decrement** the **anchor position from top**.

5.

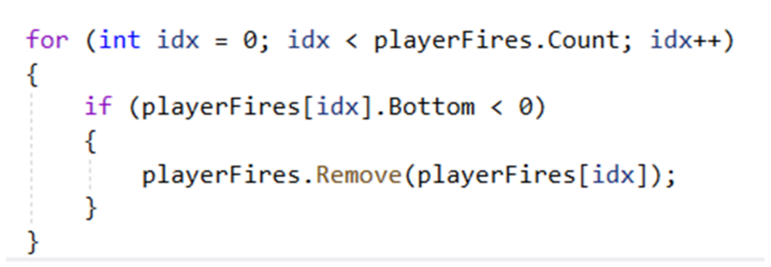


```
foreach(PictureBox bullet in playerFires)
{
    bullet.Top = bullet.Top - 20;
}
```

Include the code inisde the timer tick event.

## Removing Unnecessary Bullets from the Memory

6.

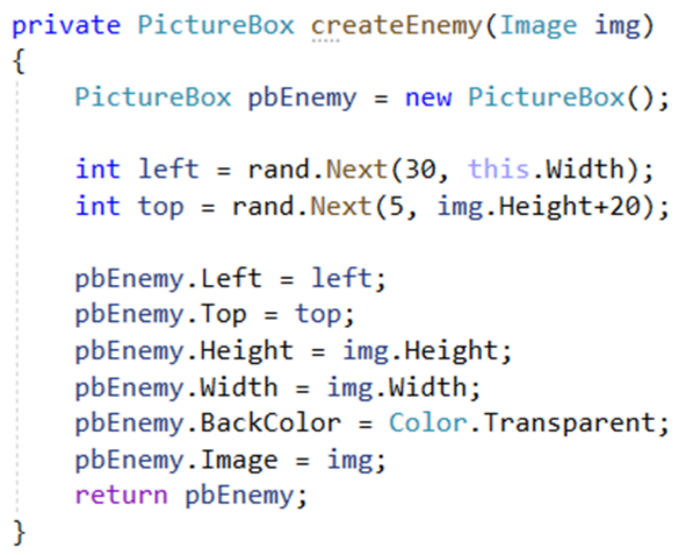


```
for (int idx = 0; idx < playerFires.Count; idx++)
{
    if (playerFires[idx].Bottom < 0)
    {
        playerFires.Remove(playerFires[idx]);
    }
}
```

Include the code inisde the timer tick event.

## Creating Multiple Enemies

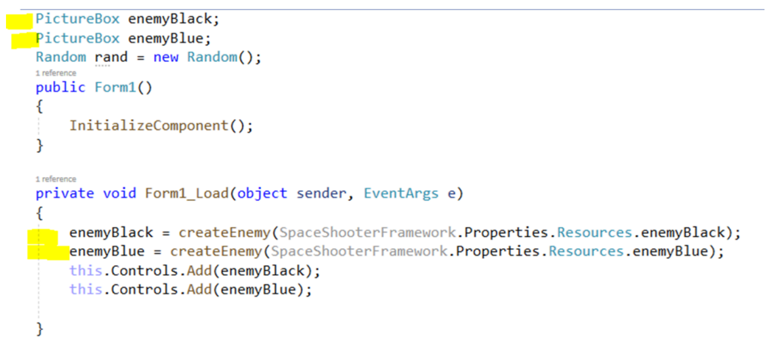
7.



```
private PictureBox createEnemy(Image img)
{
    PictureBox pbEnemy = new PictureBox();

    int left = rand.Next(30, this.Width);
    int top = rand.Next(5, img.Height+20);

    pbEnemy.Left = left;
    pbEnemy.Top = top;
    pbEnemy.Height = img.Height;
    pbEnemy.Width = img.Width;
    pbEnemy.BackColor = Color.Transparent;
    pbEnemy.Image = img;
    return pbEnemy;
}
```

Create a separate function for this functionality.

8.



```
PictureBox enemyBlack;
PictureBox enemyBlue;
Random rand = new Random();
1 reference
public Form1()
{
    InitializeComponent();
}

1 reference
private void Form1_Load(object sender, EventArgs e)
{
    enemyBlack = createEnemy(SpaceShooterFramework.Properties.Resources.enemyBlack);
    enemyBlue = createEnemy(SpaceShooterFramework.Properties.Resources.enemyBlue);
    this.Controls.Add(enemyBlack);
    this.Controls.Add(enemyBlue);

}
```

At the moment, we call the function from load form event to create two enemies.

## Moving Enemies

9.

```
//Moving Enemy Ship
if (enemyBlackDirection == "MovingRight")
{
    enemyBlack.Left = enemyBlack.Left + 10;

}

if (enemyBlackDirection == "MovingLeft")
{
    enemyBlack.Left = enemyBlack.Left - 10;
}

if ((enemyBlack.Left + enemyBlack.Width) > this.Width)
{
    enemyBlackDirection = "MovingLeft";
}

if (enemyBlack.Left <=  2)
{
    enemyBlackDirection = "MovingRight";
}
```

In order to move enemy from left to right and then right to left automatically.

While this is a **specific code for a single case only**, in case of **multiple enemies** there would be issue.
Therefore, let us create a function instead for moving the enemies.

10.

```
private void moveEnemy(PictureBox enemy, ref string enemyDirection)
{
    if (enemyDirection == "MovingRight")
    {
        enemy.Left = enemy.Left + 10;
    }
    if (enemyDirection == "MovingLeft")
    {
        enemy.Left = enemy.Left - 10;
    }
    if ((enemy.Left + enemy.Width) > this.Width)
    {
        enemyDirection = "MovingLeft";
    }
    if (enemy.Left <= 2)
    {
        enemyDirection = "MovingRight";
    }
}
```

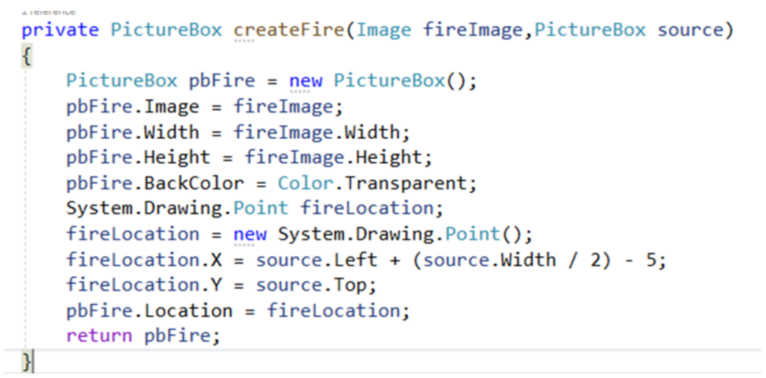
Create a separate function for this functionality.

```
//Moving Enemy Ship
moveEnemy(enemyBlack, ref enemyBlackDirection);
moveEnemy(enemyBlue, ref enemyBlueDirection);
```

Different enemies can be controlled this way.

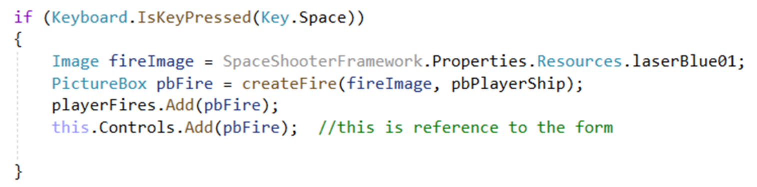
## Creating Fire Functionality

11.



```
1 reference
private PictureBox createFire(Image fireImage,PictureBox source)
{
    PictureBox pbFire = new PictureBox();
    pbFire.Image = fireImage;
    pbFire.Width = fireImage.Width;
    pbFire.Height = fireImage.Height;
    pbFire.BackColor = Color.Transparent;
    System.Drawing.Point fireLocation;
    fireLocation = new System.Drawing.Point();
    fireLocation.X = source.Left + (source.Width / 2) - 5;
    fireLocation.Y = source.Top;
    pbFire.Location = fireLocation;
    return pbFire;
}
```

Create a separate function for this functionality.

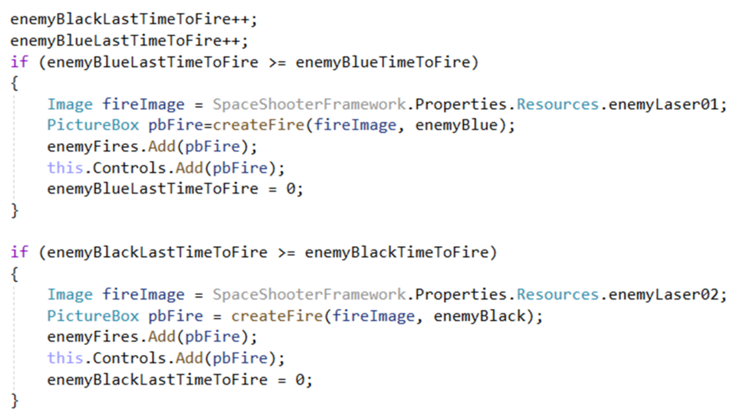


```
if (Keyboard.IsKeyPressed(Key.Space))
{
    Image fireImage = SpaceShooterFramework.Properties.Resources.laserBlue01;
    PictureBox pbFire = createFire(fireImage, pbPlayerShip);
    playerFires.Add(pbFire);
    this.Controls.Add(pbFire);  //this is reference to the form

}
```

Include this code inside Tick Event of Timer to call the

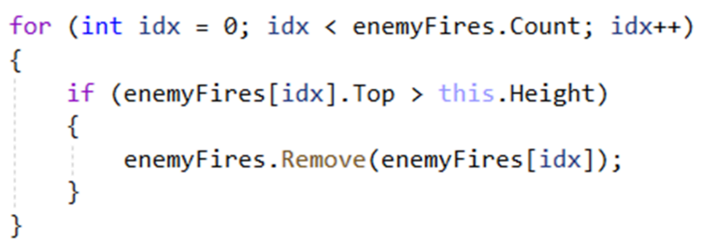
## Creating Enemy Fire

12.



```
enemyBlackLastTimeToFire++;
enemyBlueLastTimeToFire++;
if (enemyBlueLastTimeToFire >= enemyBlueTimeToFire)
{
    Image fireImage = SpaceShooterFramework.Properties.Resources.enemyLaser01;
    PictureBox pbFire=createFire(fireImage, enemyBlue);
    enemyFires.Add(pbFire);
    this.Controls.Add(pbFire);
    enemyBlueLastTimeToFire = 0;
}

if (enemyBlackLastTimeToFire >= enemyBlackTimeToFire)
{
    Image fireImage = SpaceShooterFramework.Properties.Resources.enemyLaser02;
    PictureBox pbFire = createFire(fireImage, enemyBlack);
    enemyFires.Add(pbFire);
    this.Controls.Add(pbFire);
    enemyBlackLastTimeToFire = 0;
}
```

13.



```
for (int idx = 0; idx < enemyFires.Count; idx++)
{
    if (enemyFires[idx].Top > this.Height)
    {
        enemyFires.Remove(enemyFires[idx]);
    }
}
```

**Note**: We also need to remove the fires from the memory as they are out from the width of the screen.

## Moving Enemy Fires

14.

```
foreach (PictureBox bullets in enemyFires)
{
    bullets.Top = bullets.Top + 20;
}
```

Include this code inside timer's tick event for moving the fires.

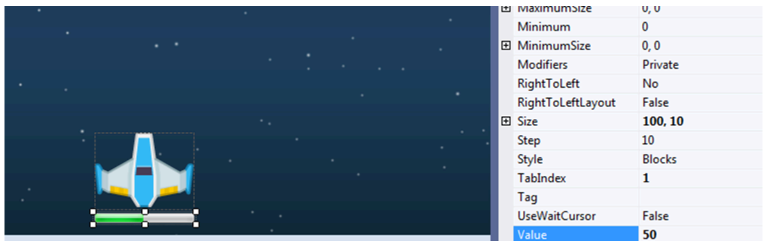
## Identification of the Collision

15.

```
foreach (PictureBox bulletes in enemyFires)
{
    if (bulletes.Bounds.IntersectsWith(pbPlayerShip.Bounds))
    {
     //Write code when player ship collide with player ship
    }
}
```

## Player Health (Progress Bar)

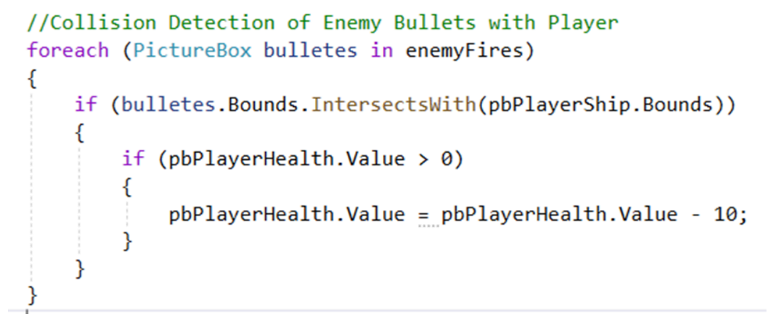
We can show health to user with help of the progress bar control. We add the control right under the ship and move it with the playership



| | |
|---|---|
| MaximumSize | 0, 0 |
| Minimum | 0 |
| MinimumSize | 0, 0 |
| Modifiers | Private |
| RightToLeft | No |
| RightToLeftLayout | False |
| Size | 100, 10 |
| Step | 10 |
| Style | Blocks |
| TabIndex | 1 |
| Tag | |
| UseWaitCursor | False |
| Value | 50 |

Add a Progress bar and set the mentioned property as required.

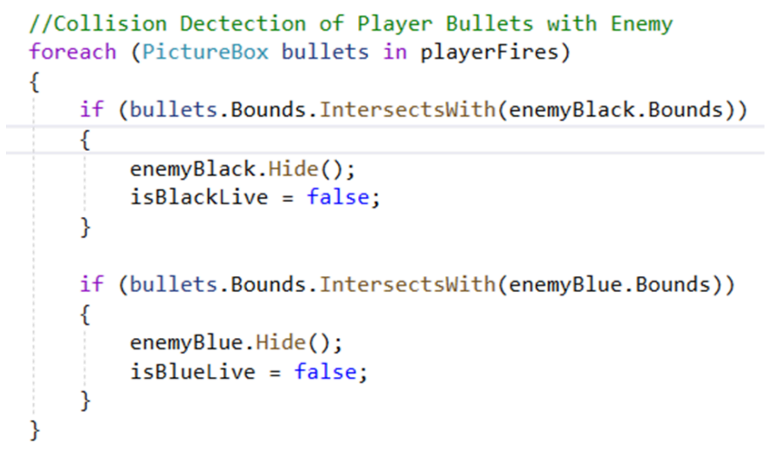
Now we can actully decrease the player health by 10 points till the value of the progress bar is not zero.



```
//Collision Detection of Enemy Bullets with Player
foreach (PictureBox bulletes in enemyFires)
{
    if (bulletes.Bounds.IntersectsWith(pbPlayerShip.Bounds))
    {
        if (pbPlayerHealth.Value > 0)
        {
            pbPlayerHealth.Value = pbPlayerHealth.Value - 10;
        }
    }
}
```

Note: You can end the game when the value is equal to or less than zero.

## Destroy Enemy

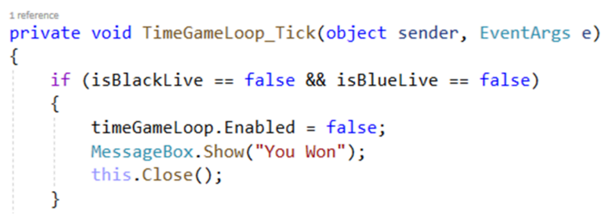


```
//Collision Dectection of Player Bullets with Enemy
foreach (PictureBox bullets in playerFires)
{
    if (bullets.Bounds.IntersectsWith(enemyBlack.Bounds))
    {
        enemyBlack.Hide();
        isBlackLive = false;
    }

    if (bullets.Bounds.IntersectsWith(enemyBlue.Bounds))
    {
        enemyBlue.Hide();
        isBlueLive = false;
    }
}
```

we can destroy enemy when player bullet hit it. For simplicity we are destroying enemy at single bullet.

Inlcude this code in tick event.

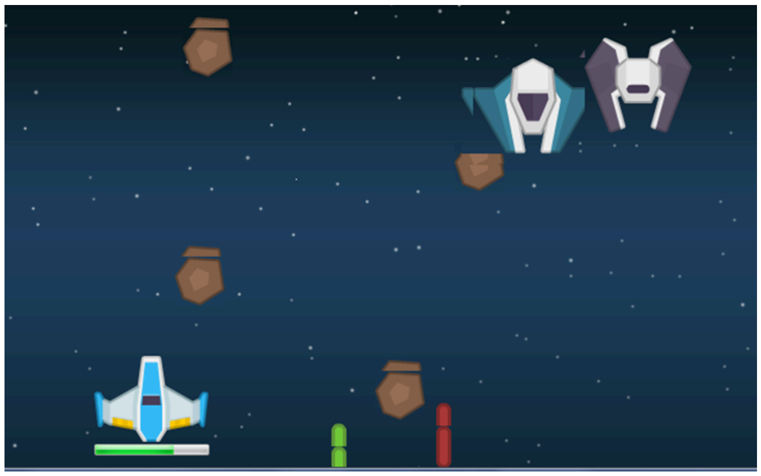
## Game Win



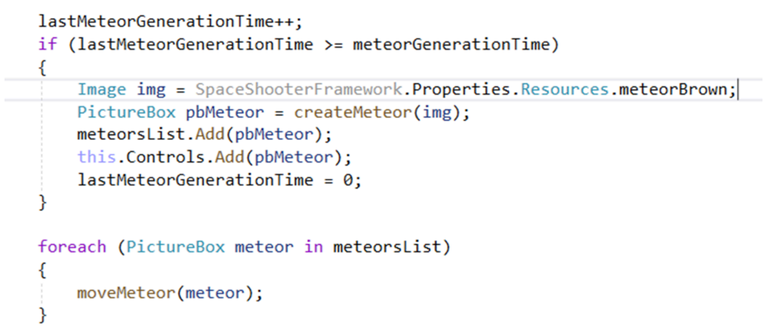
```
1 reference
private void TimeGameLoop_Tick(object sender, EventArgs e)
{
    if (isBlackLive == false && isBlueLive == false)
    {
        timeGameLoop.Enabled = false;
        MessageBox.Show("You Won");
        this.Close();
    }
```

When all enemies are destroyed game should be won by the user. For that we can write the code.

## Adding Meteriods



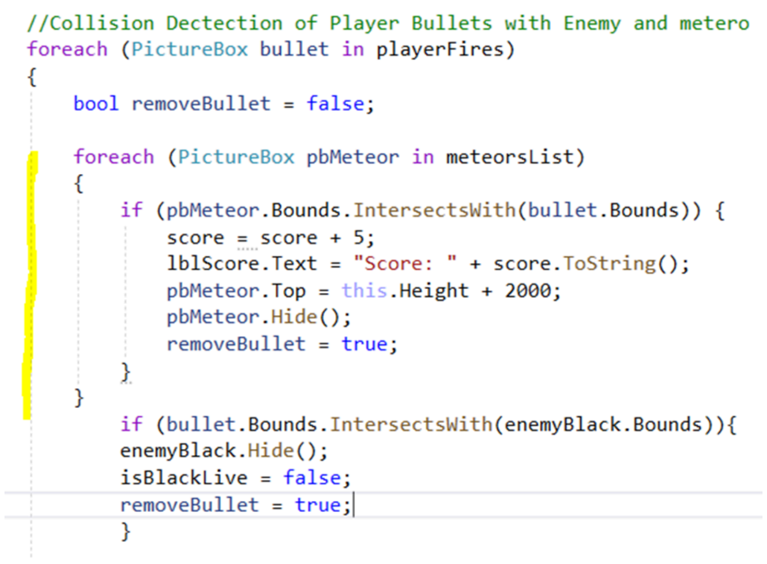
Note: We want to generate meteorite after some time and want to generate these randomly.



```
lastMeteorGenerationTime++;
if (lastMeteorGenerationTime >= meteorGenerationTime)
{
    Image img = SpaceShooterFramework.Properties.Resources.meteorBrown;
    PictureBox pbMeteor = createMeteor(img);
    meteorsList.Add(pbMeteor);
    this.Controls.Add(pbMeteor);
    lastMeteorGenerationTime = 0;
}

foreach (PictureBox meteor in meteorsList)
{
    moveMeteor(meteor);
}
```

Include the code in the Timer Tick Event.

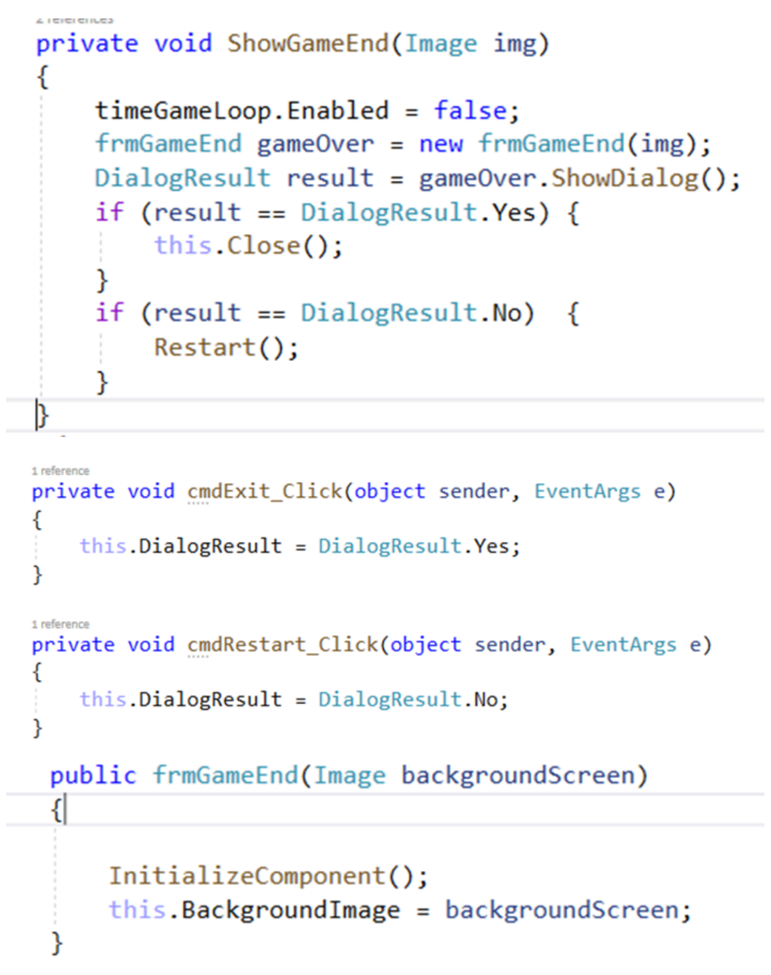
## Adding Scorig functionality



```
//Collision Dectection of Player Bullets with Enemy and metero
foreach (PictureBox bullet in playerFires)
{
    bool removeBullet = false;

    foreach (PictureBox pbMeteor in meteorsList)
    {
        if (pbMeteor.Bounds.IntersectsWith(bullet.Bounds)) {
            score = score + 5;
            lblScore.Text = "Score: " + score.ToString();
            pbMeteor.Top = this.Height + 2000;
            pbMeteor.Hide();
            removeBullet = true;
        }
    }
        if (bullet.Bounds.IntersectsWith(enemyBlack.Bounds)){
        enemyBlack.Hide();
        isBlackLive = false;
        removeBullet = true;
        }
```

**Note:** Now, if the player's bullet **hit the meteoroid**, we want to **add 5 to score**. Also, once **bullet is hit** to meteoroid it should be hide and **removed from the list**.

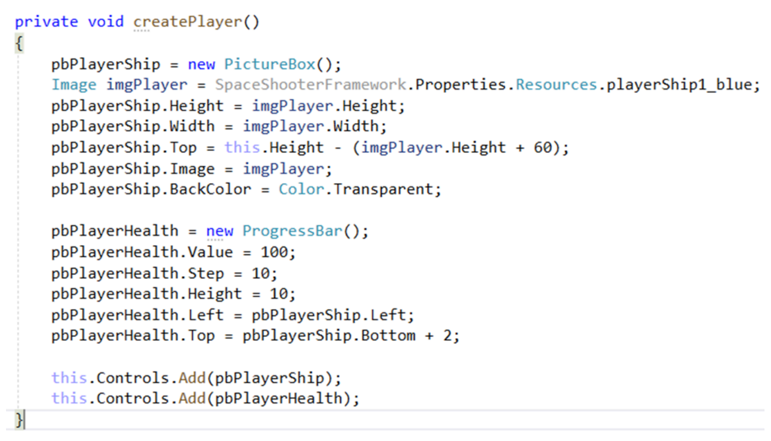
## Showing End Game Screen



```
private void ShowGameEnd(Image img)
{
    timeGameLoop.Enabled = false;
    frmGameEnd gameOver = new frmGameEnd(img);
    DialogResult result = gameOver.ShowDialog();
    if (result == DialogResult.Yes) {
        this.Close();
    }
    if (result == DialogResult.No)  {
        Restart();
    }
}

1 reference
private void cmdExit_Click(object sender, EventArgs e)
{
    this.DialogResult = DialogResult.Yes;
}

1 reference
private void cmdRestart_Click(object sender, EventArgs e)
{
    this.DialogResult = DialogResult.No;
}

 public frmGameEnd(Image backgroundScreen)
 {

     InitializeComponent();
     this.BackgroundImage = backgroundScreen;
 }
```

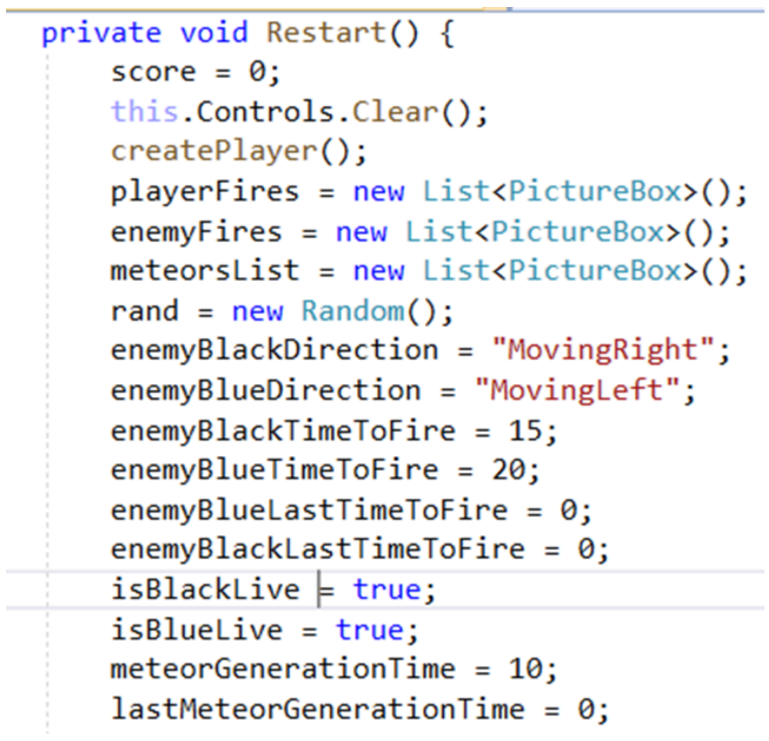
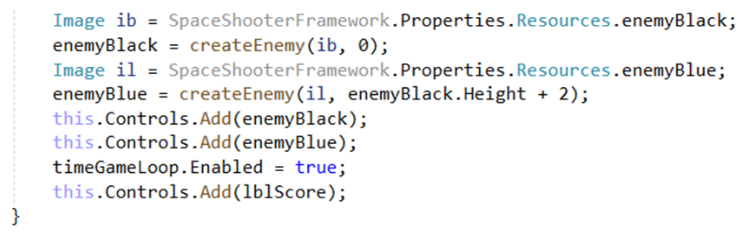
In this code, We have used **dialogue box** to show the game screen and use the **DialogResult** to decide what option user has **chosen**

While our code seems to be complete now, however, the playership is created at **design window** and needs to **created dynamically.**.
Now, we need to add Playership dynamically.



```
private void createPlayer()
{
    pbPlayerShip = new PictureBox();
    Image imgPlayer = SpaceShooterFramework.Properties.Resources.playerShip1_blue;
    pbPlayerShip.Height = imgPlayer.Height;
    pbPlayerShip.Width = imgPlayer.Width;
    pbPlayerShip.Top = this.Height - (imgPlayer.Height + 60);
    pbPlayerShip.Image = imgPlayer;
    pbPlayerShip.BackColor = Color.Transparent;

    pbPlayerHealth = new ProgressBar();
    pbPlayerHealth.Value = 100;
    pbPlayerHealth.Step = 10;
    pbPlayerHealth.Height = 10;
    pbPlayerHealth.Left = pbPlayerShip.Left;
    pbPlayerHealth.Top = pbPlayerShip.Bottom + 2;

    this.Controls.Add(pbPlayerShip);
    this.Controls.Add(pbPlayerHealth);
}
```

## Implementing Restart

| | | |
|---|---|---|
| | <br>`private void Restart() {`<br>`    score = 0;`<br>`    this.Controls.Clear();`<br>`    createPlayer();`<br>`    playerFires = new List<PictureBox>();`<br>`    enemyFires = new List<PictureBox>();`<br>`    meteorsList = new List<PictureBox>();`<br>`    rand = new Random();`<br>`    enemyBlackDirection = "MovingRight";`<br>`    enemyBlueDirection = "MovingLeft";`<br>`    enemyBlackTimeToFire = 15;`<br>`    enemyBlueTimeToFire = 20;`<br>`    enemyBlueLastTimeToFire = 0;`<br>`    enemyBlackLastTimeToFire = 0;`<br>`    isBlackLive = true;`<br>`    isBlueLive = true;`<br>`    meteorGenerationTime = 10;`<br>`    lastMeteorGenerationTime = 0;` | Create a separate function for restart and implement the functionaltiy of resetting all the required control components and variables. |
| | <br>`    Image ib = SpaceShooterFramework.Properties.Resources.enemyBlack;`<br>`    enemyBlack = createEnemy(ib, 0);`<br>`    Image il = SpaceShooterFramework.Properties.Resources.enemyBlue;`<br>`    enemyBlue = createEnemy(il, enemyBlack.Height + 2);`<br>`    this.Controls.Add(enemyBlack);`<br>`    this.Controls.Add(enemyBlue);`<br>`    timeGameLoop.Enabled = true;`<br>`    this.Controls.Add(lblScore);`<br>`}` | |

**Happy Coding ahead :)**