Problem:

We need to develop a framework through which developer can easily add picture boxes which fall under gravity.

Procedural solution:

In this case, I have to declare speed of every character whether it is moving top to down or left to right. I also have to change the position with respect to timer i.e., every time the character must automatically move and change its position. I also have to make sure that every character returns to its original position when game restarts.

Procedural code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace game_framework
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            resetGame();
        }

       int ghostSpeed1 =5;
       int ghostSpeed2 = 6;

        private void tick(object sender, EventArgs e)
        {
            enemy1.top+=ghostSpeed1;
            enemy2.top += ghostSpeed2;
        }

Public void resetGame()

{
       ghost1.Left = 445;

       ghost1.Top = 175;

       ghost2.Left = 350;
       ghost2.Top = 290;
      foreach(Control x in this.Controls)
             {
                if(x is PictureBox)
                {
                    x.Visible = true;
                }
            }
```

```
}
    }
}
```

Demerits:

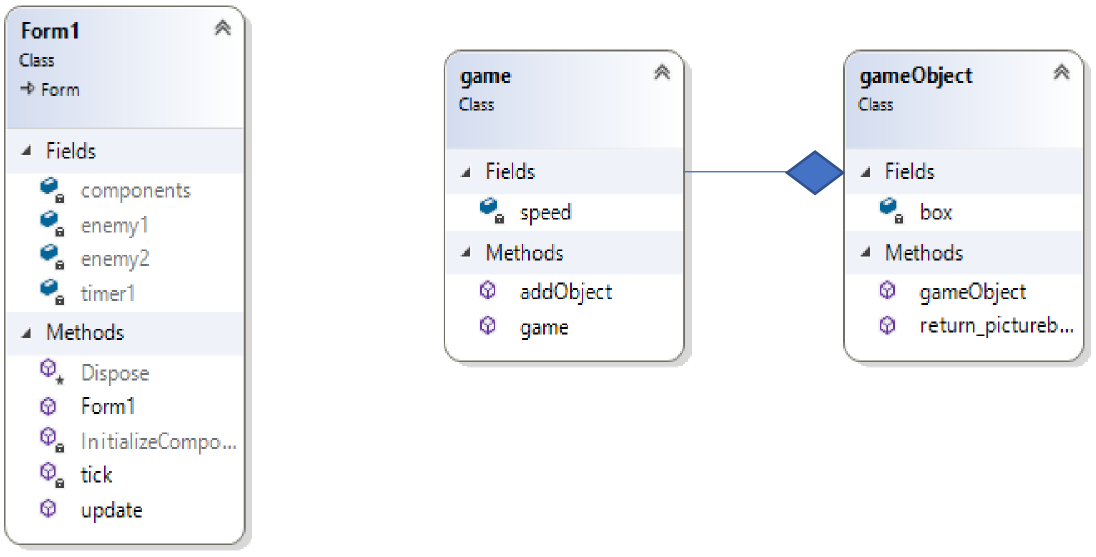
The procedural solution may seem simpler and easier at first sight but on a larger scale, it becomes problematic. We would have to declare speed of every character and then define its properties like top,left position and its visibility. So this approach not only makes our work complex but also makes it complex.

Oop:

Solution:

In this solution, we will try to use a dynamic way. First, we will create a class(gameObject) that will set attributes of all picture boxes at same time. We will create another class Game which will set the speed of all game characters. the game class will be responsible for changing speed of objects. This class will communicate with class gameObject to obtain picturebox and change its position.

Class diagram:



Demerits:

One problem is that we have to set speed of all characters at once and we are not able to set speed of each object separately.

Merit:

In this approach, we will save our time as we can create characters very easily just by using two classes. We just have to set the speed of characters one time and then pass the picture boxes to another class.

Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace game_framework
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void tick(object sender, EventArgs e)
        {
            update();
        }
        public void update()
        {
            game g = new game(5);

            gameObject obj1 = new gameObject(enemy1);
            gameObject obj2 = new gameObject(enemy2);

            g.addObject(obj1);
            g.addObject(obj2);
        }
    }
}


    class gameObject
    {
        PictureBox box;
        public gameObject(PictureBox pic)
        {
            pic.BackColor = Color.Red;
            pic.Visible = true;
            this.box = pic;
        }
```

```
            public PictureBox return_picturebox()
            {
                return this.box;
            }
        }
    }

        class game
        {
            int speed;
            public game(int move_speed)
            {
                this.speed = move_speed;
            }
            public void addObject(gameObject objs)
            {
                 PictureBox p= objs.return_picturebox();
                 p.Top =p.Top + this.speed;
            }
        }
    }
```