Developing a Framework

Using C# and WinForms

What is Framework.

Here, In this context, framework is a line of code that has been written independently and can be re-used at multiple locations.

What is Framework.

For example, in every game, we write down code to move objects (some time with keyboard, some time move vertically, some time move right to left and then left to right).

If we write down this code to move objects once and then allow every other developer to reuse that code in his/her game without exposing the details of the code this is called framework..

What is Framework.

However, one may argue, this can we done through the simple OOP. Yes, framework is manifestation of OOP programming.

What is Framework.

Framework not only offers the re-usability but it should also offers two fundamental constructs

1. Constraints on the Development Side.
2. Extendability.

FrameWork Task 01

Problem: Let we need to add falling functionality for different objects (picture box) such as enemies and other players.

Solution Requirements:

We need to create a independent game framework that can be used by the other developer to make their picturbox fall under gravity.

The user of framework only decide the images of objects and type of motion for the object. All other code should be inside the framework as separate DLL file

Framework 01: Falling Under Gravity.

Solution Requirements:

- The game design (picture boxes) should be separated.
- There should be separated business logic from the form.

Framework 01 : Falling Under Gravity

How to do that ?

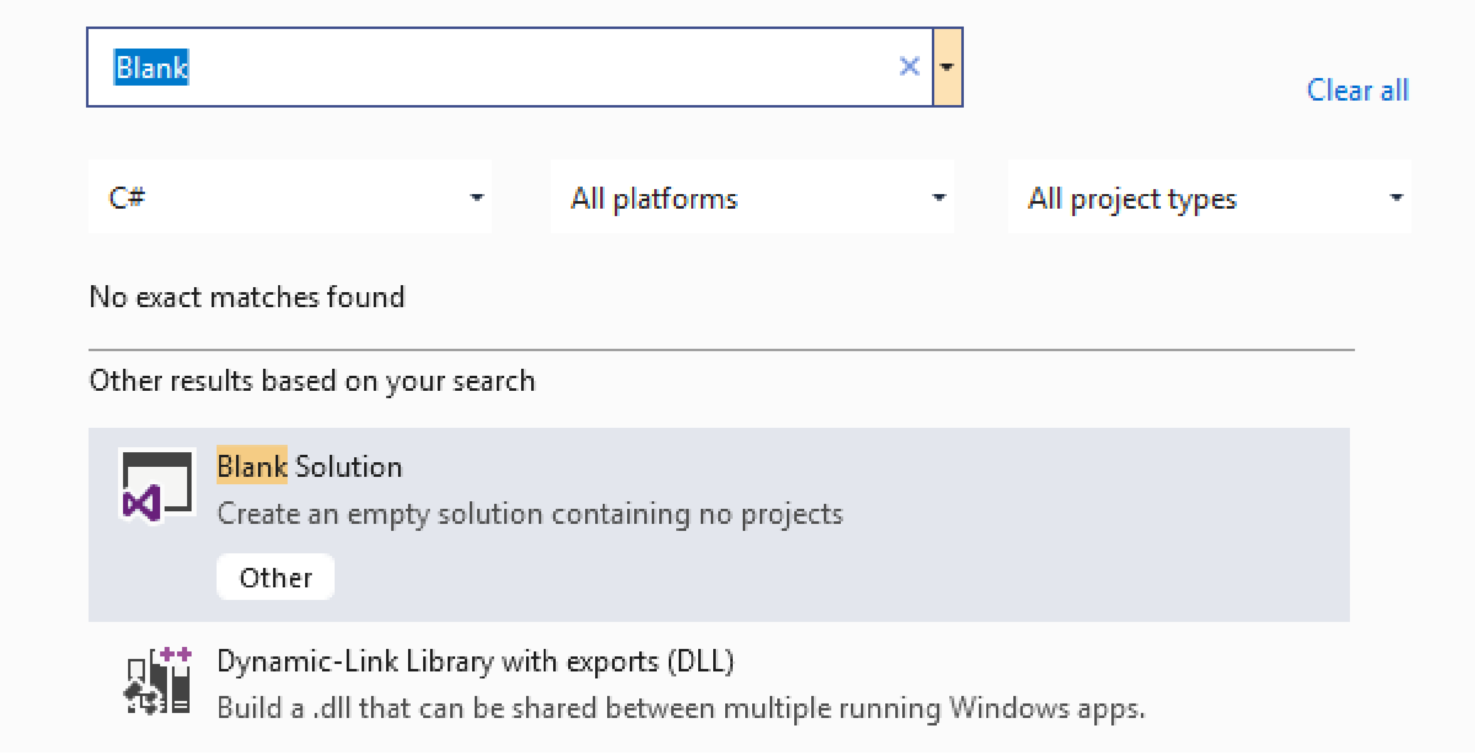
Framework 01 : Falling Under Gravity

First thing is to visualize if you are user of the framework and how will you want to use it ?

Framework 01: Falling Under Gravity

1. There should be some way that the developer tell the framework that the specific picture shall act as falling object.
2. Also, there should be a way to repeatedly update the position of the all the falling objects.

Framework 01: Create an Empty Project



No exact matches found

Other results based on your search

Blank Solution

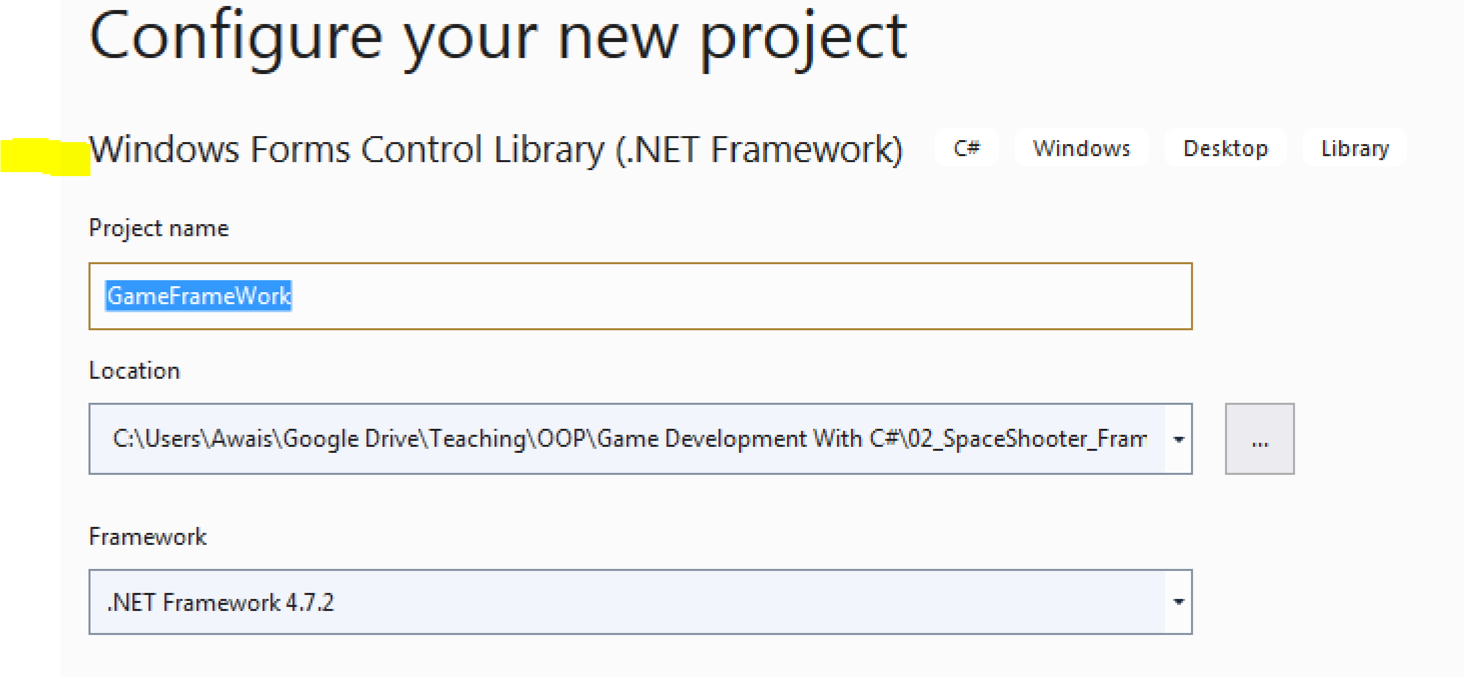
Create an empty solution containing no projects

Dynamic-Link Library with exports (DLL)

Build a .dll that can be shared between multiple running Windows apps.

Framework 01: Add Project for FrameWork

We have to use some graphical control such as picture box therefore the type is easier for reference management



Configure your new project

Windows Forms Control Library (.NET Framework)

Project name

GameFrameWork

Location

C:\Users\Awais\Google Drive\Teaching\OOP\Game Development With C#\02_SpaceShooter_Fram

Framework

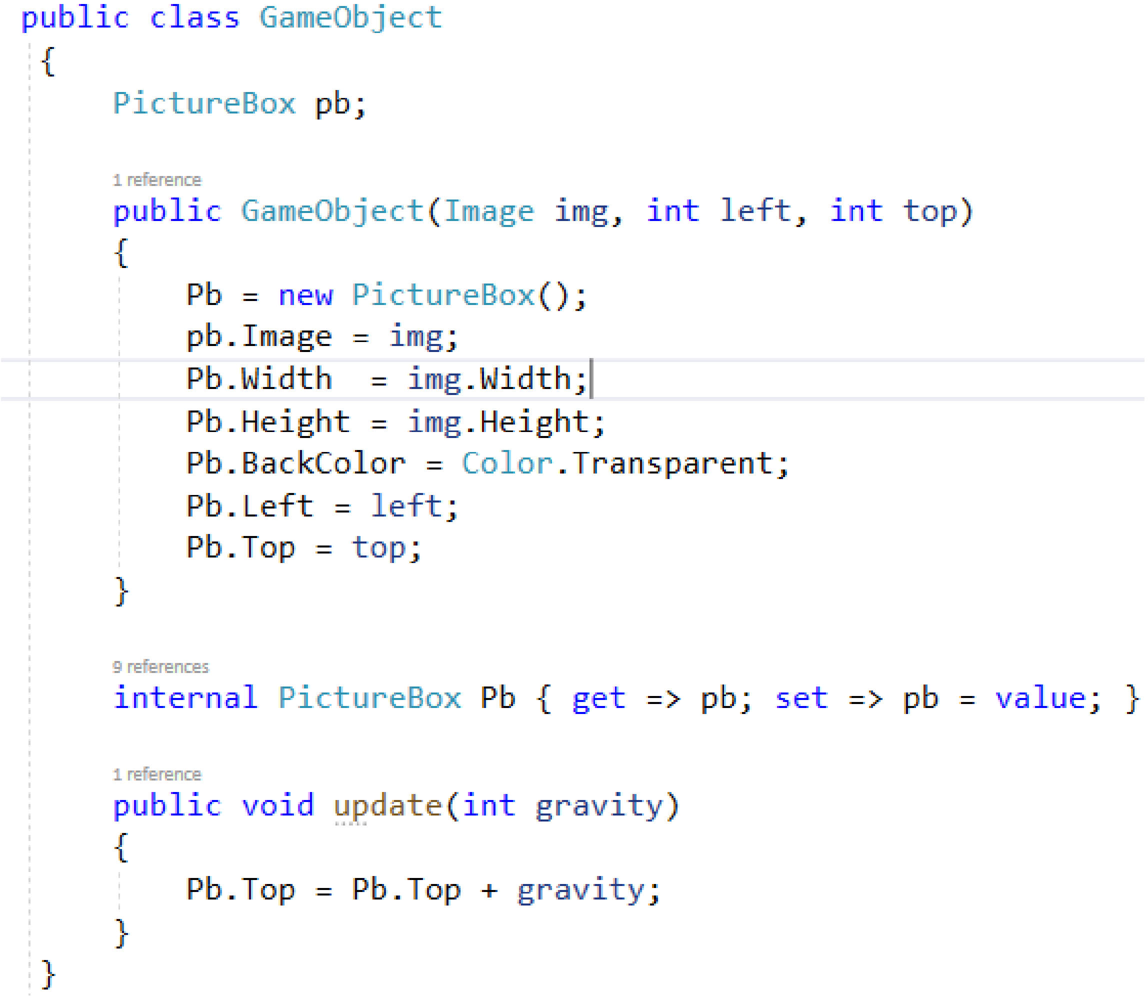
.NET Framework 4.7.2

Framework 01: GameObject class

we need to create a class that allows game objects to define

- Attributes (such as pictureBox, Movement Speed)
- Behavior of the game object (Movement Direction Etc).
- An Update method that will update the game object at every tick.

Framework 01: GameObject class



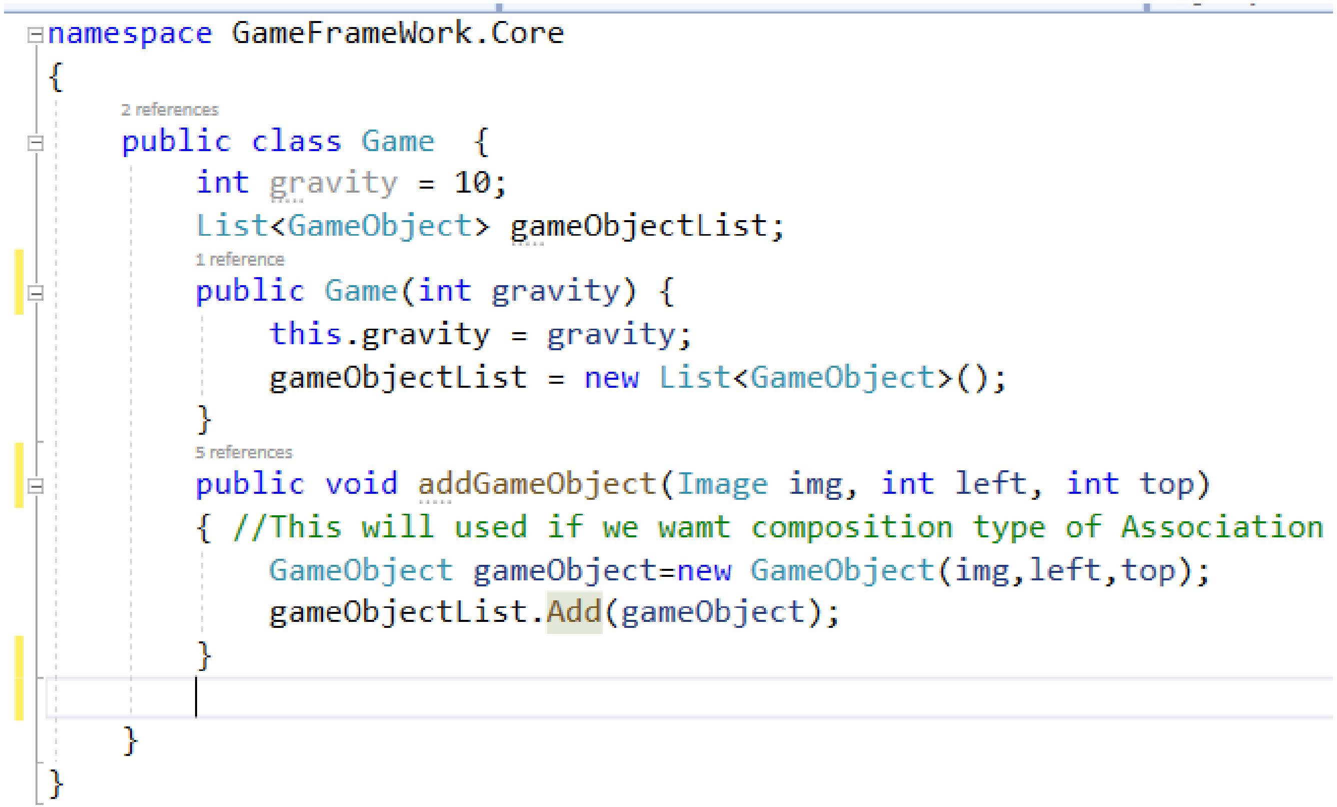
```
public class GameObject
{
    PictureBox pb;

    1 reference
    public GameObject(Image img, int left, int top)
    {
        Pb = new PictureBox();
        pb.Image = img;
        Pb.Width  = img.Width;
        Pb.Height = img.Height;
        Pb.BackColor = Color.Transparent;
        Pb.Left = left;
        Pb.Top = top;
    }

    9 references
    internal PictureBox Pb { get => pb; set => pb = value; }

    1 reference
    public void update(int gravity)
    {
        Pb.Top = Pb.Top + gravity;
    }
}
```

Framework 01: Game class

Now, we shall create a game class which will provide two services

- keep track of all the game objects
- Call update method of each game object.

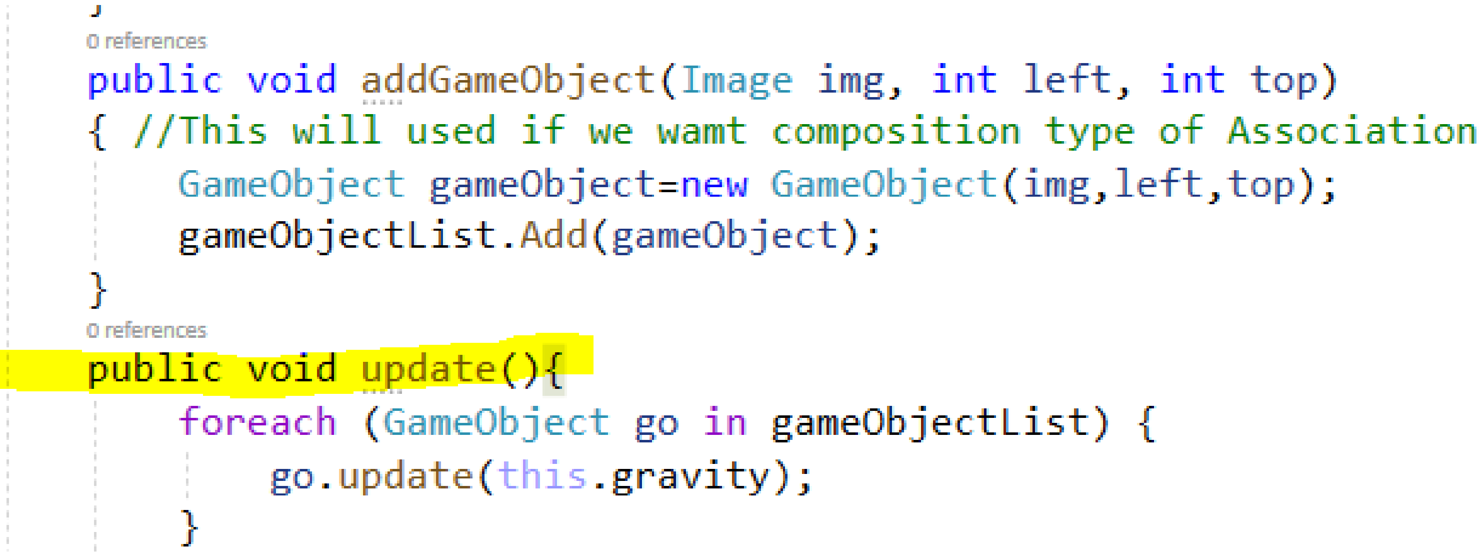
Framework 01: Game class



```
namespace GameFrameWork.Core
{
    2 references
    public class Game  {
        int gravity = 10;
        List<GameObject> gameObjectList;
        1 reference
        public Game(int gravity) {
            this.gravity = gravity;
            gameObjectList = new List<GameObject>();
        }
        5 references
        public void addGameObject(Image img, int left, int top)
        { //This will used if we wamt composition type of Association
            GameObject gameObject=new GameObject(img,left,top);
            gameObjectList.Add(gameObject);
        }
        
    }
}
```

Framework 01: Game class

Way to provide Update all game objects. Add update function in game class that subsequently call update function of all of it s game objects



```
}
0 references
public void addGameObject(Image img, int left, int top)
{ //This will used if we wamt composition type of Association
    GameObject gameObject=new GameObject(img,left,top);
    gameObjectList.Add(gameObject);
}
0 references
public void update(){
    foreach (GameObject go in gameObjectList) {
        go.update(this.gravity);
    }
```

Framework 01: Game class

Now, one more thing is left in our game framework that these game Objects are in ram but they are not graphically shown on the form.

How can Game class can show/add these objects on the form ?

Framework 01: First Approach

How can Game class can show/add these objects on the form ?

1. Pass form object reference to Game so when it will create gameObject all add that into the form

```
Game g = new Game(10, this);
g.addGameObject (meteoroid,10,20);
```

Inside the addGameObject

```
//code to add game object
container.Controls.Add(g.getPictureBox())
```

ANY PROBLEM ?

Framework 01: Second One

How can Game class can show/add these objects on the form ?

2. Create Gameobject from the Form and also add the picture box of the game Object into the Form. The could will look like this

```
Game g = new Game(10);
GameObject go1 = new GameObject (meteoroid,10,20)
g.addGameObject(go1);
this.Controls.Add(g.getPictureBox())
```

Framework 01: What we want

How can Game class can show/add these objects on the form ?

The problem with the second approach is sometimes we need to create the objects from within the framework and it is not always necessary that the main form has that object.

Also we do not want to pass the GUI (Form) reference to framework.

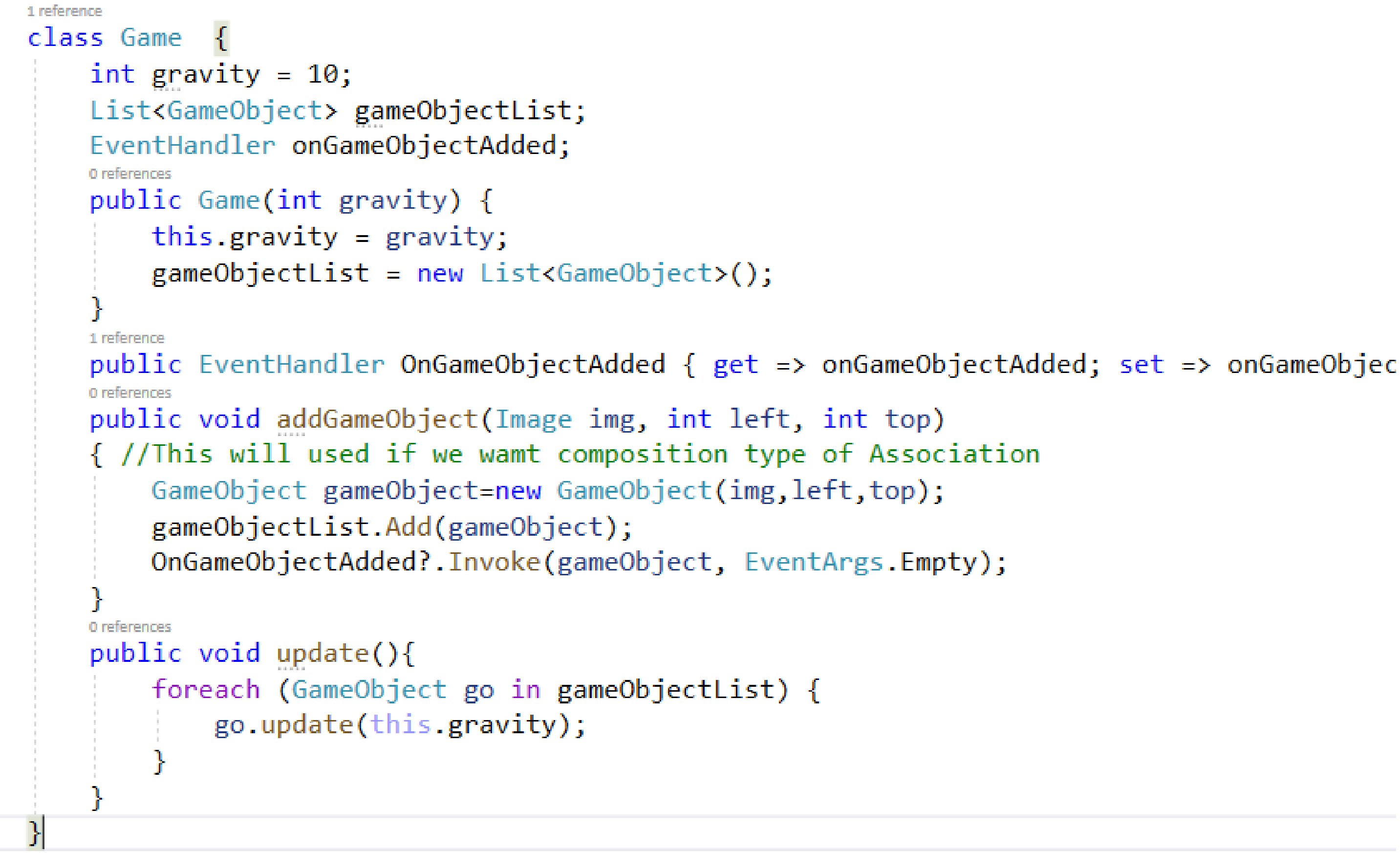
So how to combine these two approaches to get the required result ?

Framework 01: Event Based

How can Game class can show/add these objects on the form ?

Third approach is to add an event in the game class and when any game object is added, the event should raised and the form will handle this event to add the newly added object into the forms control.

Framework 01: Event Based



```
1 reference
class Game  {
    int gravity = 10;
    List<GameObject> gameObjectList;
    EventHandler onGameObjectAdded;
    0 references
    public Game(int gravity) {
        this.gravity = gravity;
        gameObjectList = new List<GameObject>();
    }
    1 reference
    public EventHandler OnGameObjectAdded { get => onGameObjectAdded; set => onGameObjec
    0 references
    public void addGameObject(Image img, int left, int top)
    { //This will used if we wamt composition type of Association
        GameObject gameObject=new GameObject(img,left,top);
        gameObjectList.Add(gameObject);
        OnGameObjectAdded?.Invoke(gameObject, EventArgs.Empty);
    }
    0 references
    public void update(){
        foreach (GameObject go in gameObjectList) {
            go.update(this.gravity);
        }
    }
}
```

Framework 01: Consumer Game

Now, let's consume this framework and see how easier the things are for developer.

- What will be the steps ?

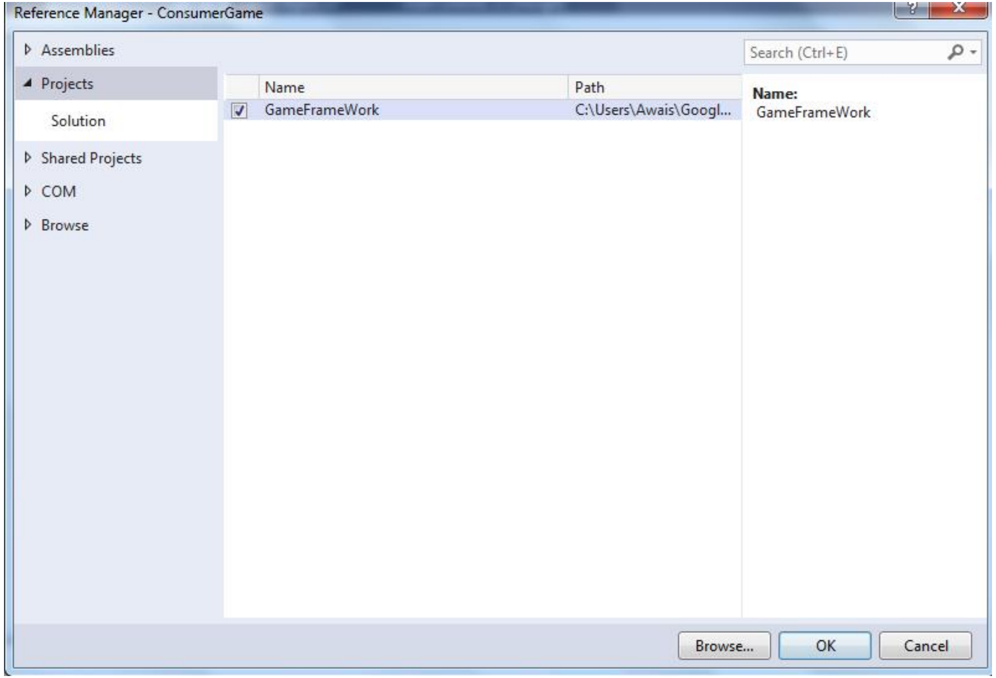
Framework 01: Consumer Game

To consume the framework follow these steps:

- Create a windows form project and add reference to framework
- In Consumer Project add Images into the Resources.
- Set the background color of the form.
- Make the form’s project as startup project
- Add timer on the form
- In Main Load Event create main instance of game and add different game objects.
- Handle OnAddGameObject event of the game to add pictureBox of the gameObject into the form controls .
- In time Update call the update method of game,

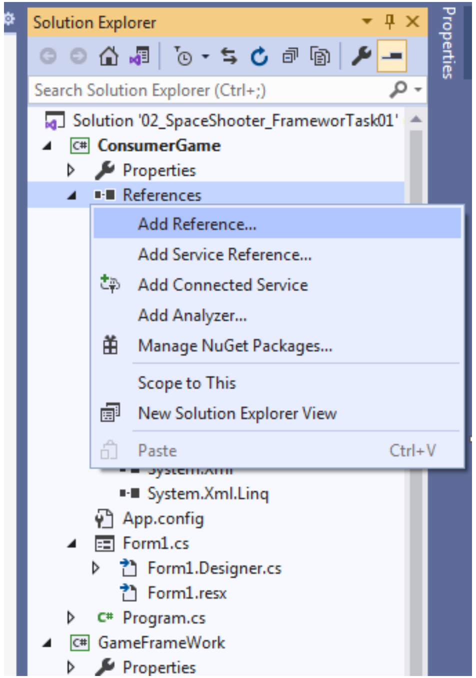
Framework 01 : Consumer Game

Create a windows form project and add reference to framework.



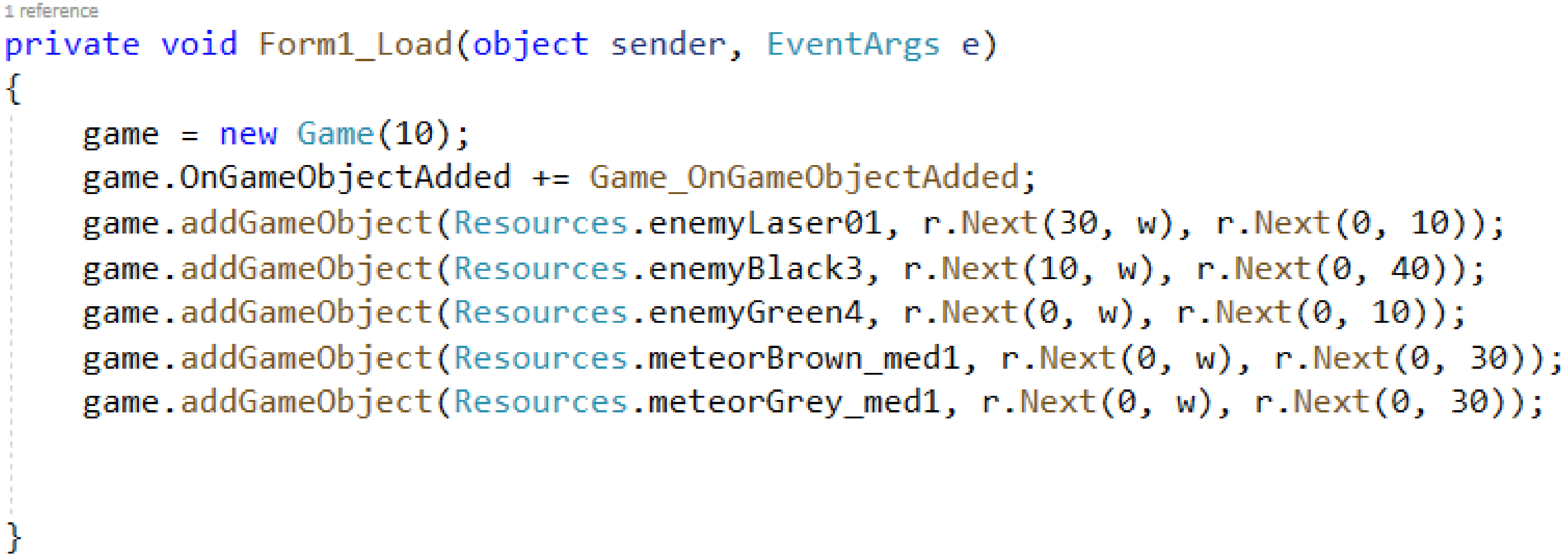
	Name	Path
☑	GameFrameWork	C:\Users\Awais\Googl...

Name: GameFrameWork



Framework 01: Consumer Game

Form Load will look like following



```
1 reference
private void Form1_Load(object sender, EventArgs e)
{
    game = new Game(10);
    game.OnGameObjectAdded += Game_OnGameObjectAdded;
    game.addGameObject(Resources.enemyLaser01, r.Next(30, w), r.Next(0, 10));
    game.addGameObject(Resources.enemyBlack3, r.Next(10, w), r.Next(0, 40));
    game.addGameObject(Resources.enemyGreen4, r.Next(0, w), r.Next(0, 10));
    game.addGameObject(Resources.meteorBrown_med1, r.Next(0, w), r.Next(0, 30));
    game.addGameObject(Resources.meteorGrey_med1, r.Next(0, w), r.Next(0, 30));


}
```

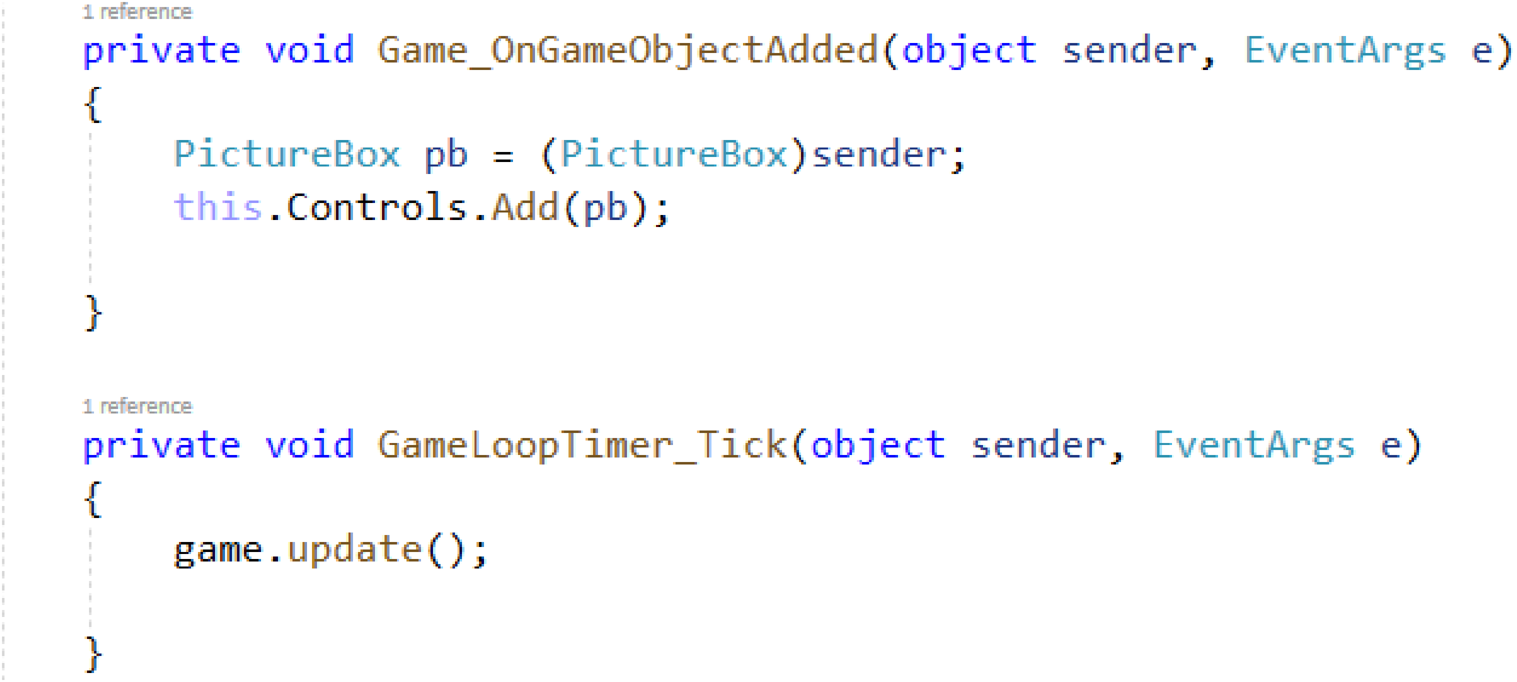
Framework 01: Falling Under Gravity

Within Timer Tick, we need to call the update method of our game object. May be like following..

```
game.update()
```

Framework 01: Consumer Game

On Tick Event of main game loop and onGameObjectAdded



```
1 reference
private void Game_OnGameObjectAdded(object sender, EventArgs e)
{
    PictureBox pb = (PictureBox)sender;
    this.Controls.Add(pb);

}

1 reference
private void GameLoopTimer_Tick(object sender, EventArgs e)
{
    game.update();

}
```

FrameWork Task 02 (Take Home)

1. Now extend the framework such that it allows developer to move any object using three types of motions
   a. Move Vertically (Falling)
   b. Periodically Move Left Right
   c. Move through KeyBoard

Note: we want the user of framework only decide the images of objects and type of motion for an object only. All other code should be inside the framework as separate DLL file

FrameWork Task 03 (Take Home)

Your framework should be developed in a way that if user of your framework want to extend the behaviour of the motion then it could do that easily without opening the code of the framework

For example user of the framework want to move an object zig zag or diagonally then it only need to write the code for this movement and then inject into your framework so now framework allow gameobjects to move diagonally. (Note this change will only for the user)