

## DELEGATION:

### 1. What is delegation?

- Delegation is an important concept that will use the “delegate” type to represent references to methods. We can simply understand that delegation is a “method-pointer” with its corresponding “signature”.

### 2. What is “signature”?

- It is the structure of the delegate method that others should follow if they want to be delegated.

### 3. Think of Delegation this way!

“Hey, here is a method I would love to use, but I’ll specifically decide which one later”.

- 4. **Example:** You are an owner of a car company and your company is on Boxing Day. All of the cars must be discounted on that day, the discount will be calculated either by 10% off or \$5 off. You do not know yet, but certainly there will be two ways. You are holding the Calculator (delegate) to calculate your new sales price for each product, you will decide which discount types you want on each car (which method to call). Here are the steps:

#### Step 1: Define Delegate:

```
public delegate double DiscountDelegate(double price);
```

=> Signature: “[modifier] double [name] (double price)”

#### Step 2: Create methods that match the delegate’s signature:

```
public class DiscountCalculator
{
    public double CalculatePercentageDiscount(double price) => price * 0.9;
    public double CalculateAmountDiscount(double price) => price - 5;
}
```

#### Step 3: Create a delegate and point it to a method:

```
public class Program
{
    static void Main()
    {
        DiscountCalculator discountCalculator = new DiscountCalculator();
    }
}
```

```
DiscountDelegate discountDelegate =  
discountCalculator.CalculatePercentageDiscount;  
Console.WriteLine(discountDelegate(100)); // Output: 90
```

Step 4: Change to the other method if you want the delegate point to:

```
discountDelegate = discountCalculator.CalculateAmountDiscount;  
Console.WriteLine(discountDelegate(100)); // Output: 95
```

1. An ambiguous reference -> means a type declared exists in 2 namespaces -> specify.

```
5 references
private List<Robot> _robots = new List<Robot>();
0 references
public Timer myTimer;

1 reference
public bool Quit
{
    get
    {
        return _player.Quit;
    }
}

1 reference
public RobotDodge (Window gameWindow) // Constructor
{
    if (gameWindow == null)
    {
        throw new ArgumentException(nameof(gameWindow), "Game window cannot be null");
    }

    _gameWindow = gameWindow;
    _player = new Player(gameWindow);
}
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

RobotDodge.cs 1

✗ 'Timer' is an ambiguous reference between 'SplashKitSDK.Timer' and 'System.Threading.Timer' (CS0104) [Ln 9, Col 12]

SplashKit.cs lib 1

ⓘ Not showing 700 further errors and warnings. [Ln 2740, Col 32]

```
0 references
public SplashKitSDK.Timer myTimer;

1 reference
```

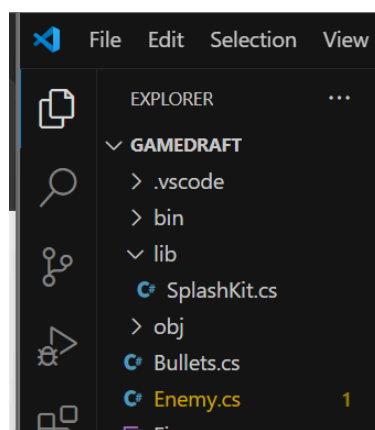
## 2. Bad Format Exception:

Issue:

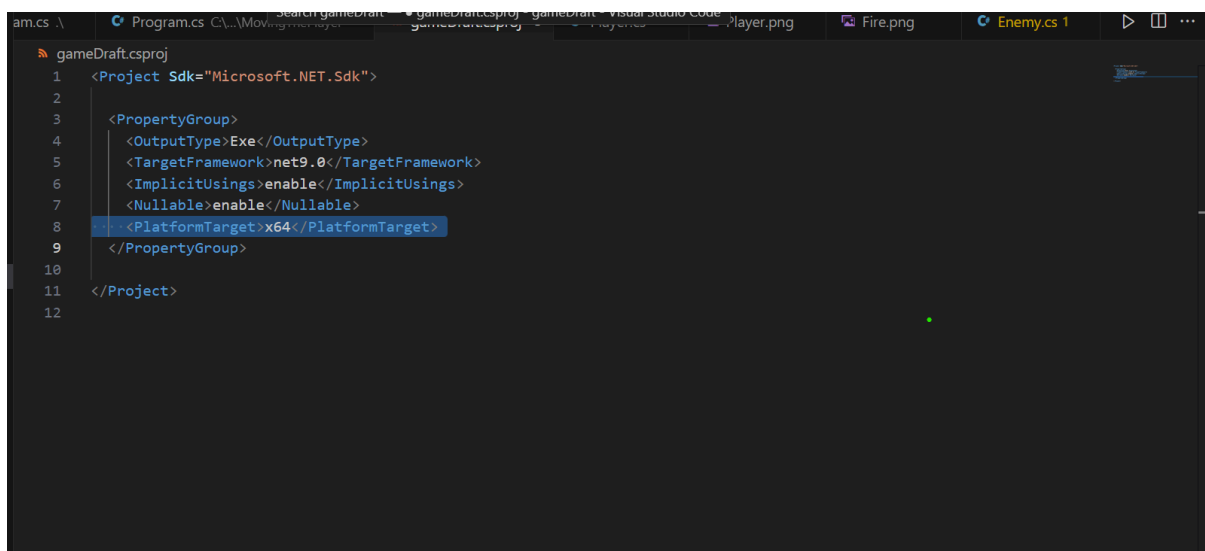
```
hdat2@tien MINGW64 /c/Users/hdat2/OneDrive/sit771/gameDraft
C:\Users\hdat2\OneDrive\sit771\gameDraft\Enemy.cs(22,12): warning CS8618: Non-nullable field '_gamewindow' must contain a non-null value when exiting constructor. Co
C:\Users\hdat2\OneDrive\sit771\gameDraft\Enemy.cs(10,20): warning CS0169: The field 'Enemy._gamewindow' is never used
Unhandled exception. System.BadImageFormatException: An attempt was made to load a program with an incorrect format. (0x8007000b)
   at SplashKitSDK.SplashKit.__sklib__open_window_string_int_int(__sklib_string caption, Int32 width, Int32 height)
   at SplashKitSDK.SplashKit.OpenWindow(String caption, Int32 width, Int32 height) in C:\Users\hdat2\OneDrive\sit771\gameDraft\lib\SplashKit.cs:line 15905
   at SplashKitSDK.Window..ctor(String caption, Int32 width, Int32 height) in C:\Users\hdat2\OneDrive\sit771\gameDraft\lib\SplashKit.cs:line 19795
   at gameDraft.Program.Main() in C:\Users\hdat2\OneDrive\sit771\gameDraft\Program.cs:line 12
```

Solution:

Go to your project file located at the left bar:



Add the `<PlatformTarget> *your computer's architecture* </PlatformTarget>`



```
1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <OutputType>Exe</OutputType>
5     <TargetFramework>net9.0</TargetFramework>
6     <ImplicitUsings>enable</ImplicitUsings>
7     <Nullable>enable</Nullable>
8     <PlatformTarget>x64</PlatformTarget>
9   </PropertyGroup>
10
11 </Project>
12
```