

# Battle-Field-4

High-Quality Code - Team Work

**Адриана Диманова**

**Ангел Раев**

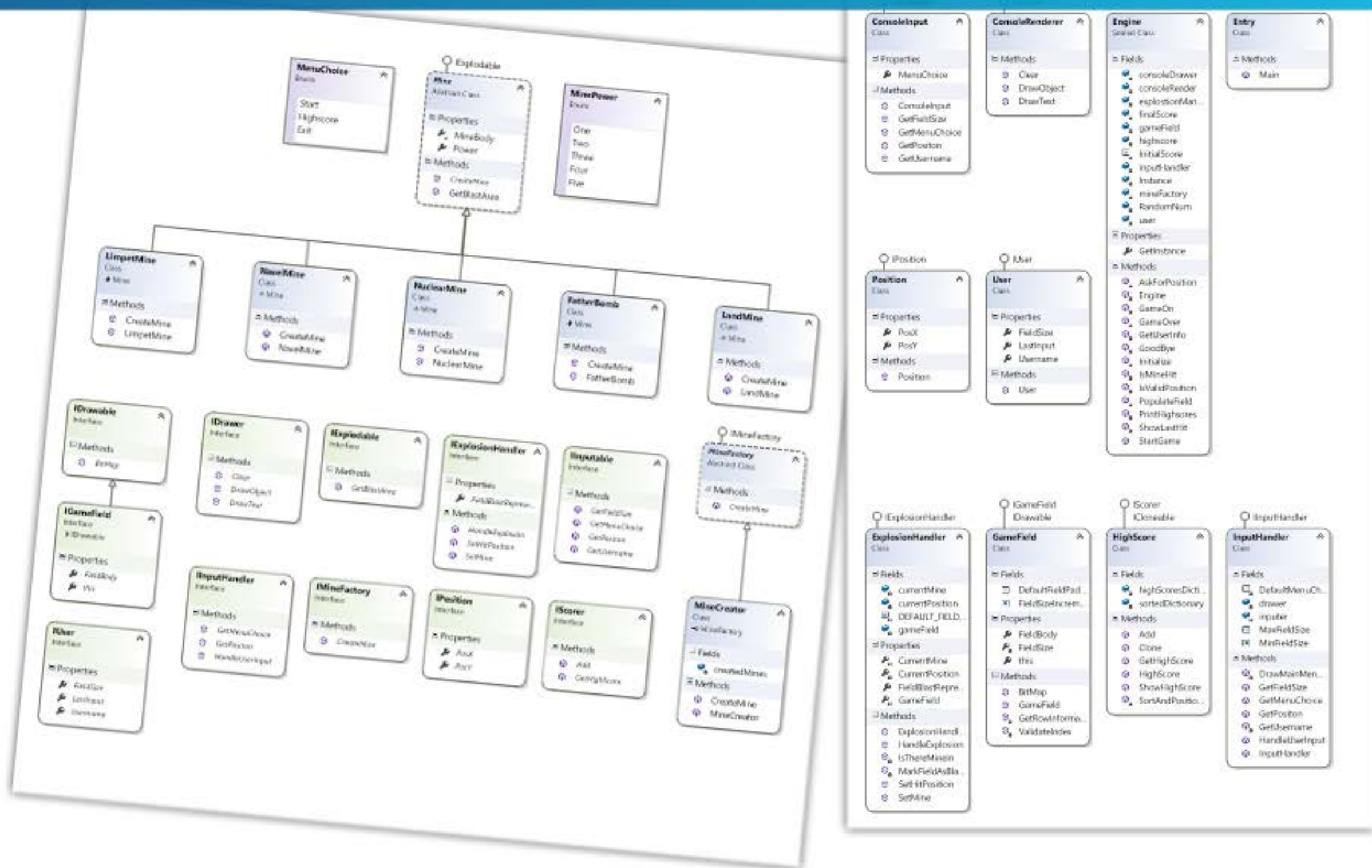
**Валерия Димитрова**

**Димитър Костов**

**Мартин Маринов**

**Пламен Петков**

# REFACTORING



# TOTAL GAME REDESIGN



# DESIGN PATTERNS

- + Strategy
- + Template
- + Singleton
- + Facade
- + Flyweight
- + Factory
- + Prototype
- + Dependency Injection

```
while (minesOnFieldCount > 0)
{
    this.consoleDrawer.Clear();
    this.ShowLastHit();
    this.consoleDrawer.DrawObject(this.gameField);

    do
    {
        this.AskForPosition();
        this.user.LastInput = this.inputHandler.GetPositon();
    }
    while (!this.IsValidPosition());

    this.finalScore++;

    if (this.IsMineHit())
    {
        // Generate mine
        string mineHitOnField = this.gameField[this.user.LastInput.PosX, thi
        int mineHit = int.Parse(mineHitOnField);
        var currentMine = this.mineFactory.CreateMine((MinePower)mineHit);

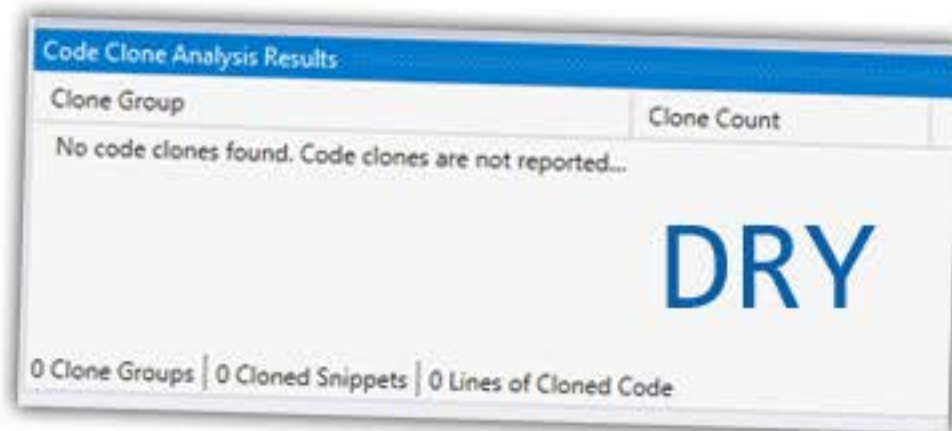
        // Configure(reconfigure) the explosion manager
        this.explosionManager.SetMine(currentMine);
        this.explosionManager.SetHitPosition(this.user.LastInput);

        // Blow the mine up
        int minesTakenOut = this.explosionManager.HandleExplosion();

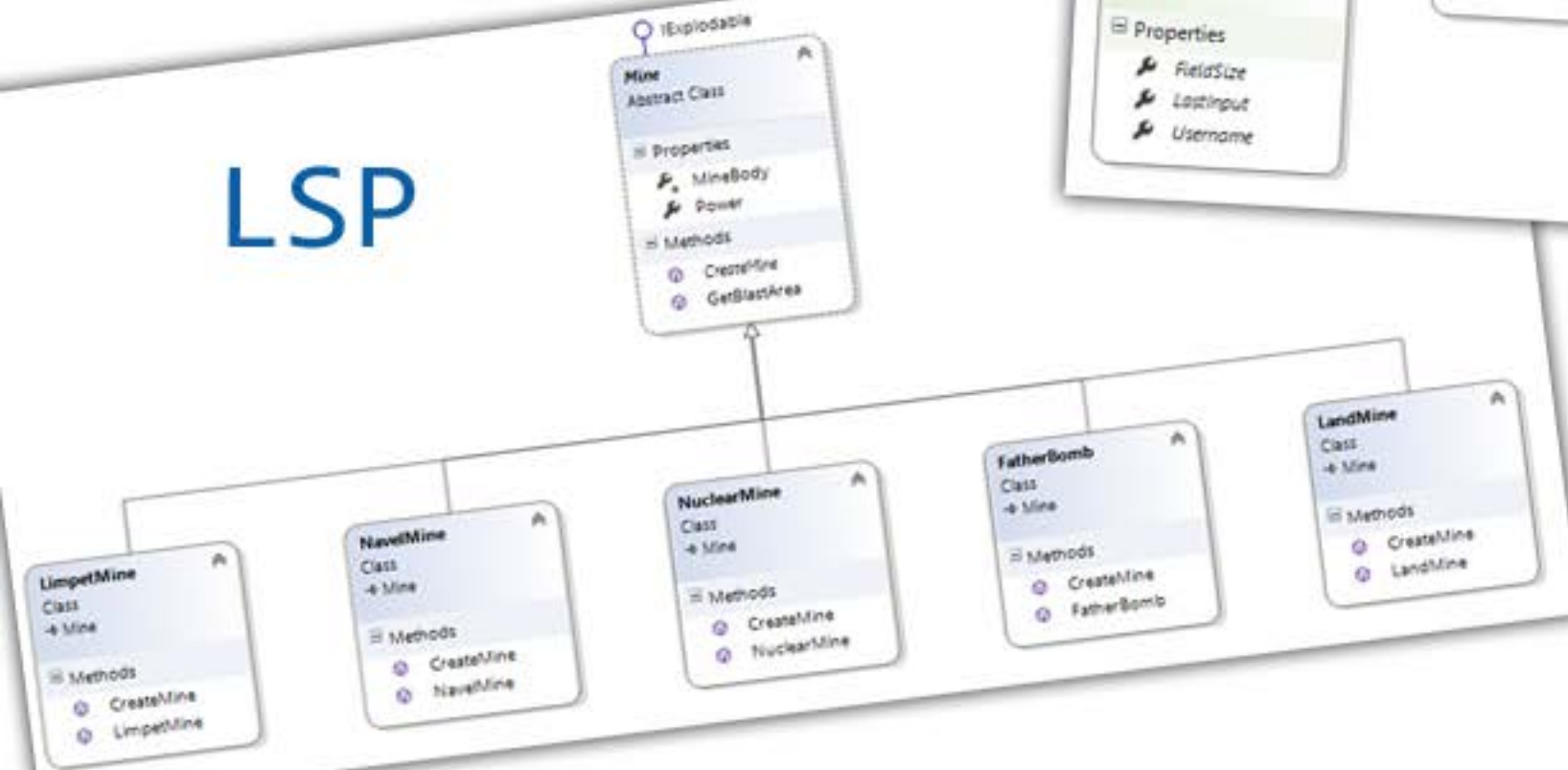
        minesOnFieldCount -= minesTakenOut;
    }
}

this.consoleDrawer.Clear();
this.ShowLastHit();
this.consoleDrawer.DrawObject(this.gameField);
}
```

# SOLID / DRY PRINCIPLES



**LSP**



**DIP**

```
private IDrawer consoleDrawer;
private IInputable consoleReader;
private IInputHandler inputHandler;
private IMineFactory mineFactory;
private IGameField gameField;
private IExplosionHandler explosionManager;
private IUser user;
```



# DETAILED DOCUMENTATION

The image displays a software documentation interface for a project named "BattleFieldGameLib". The interface is divided into several sections:

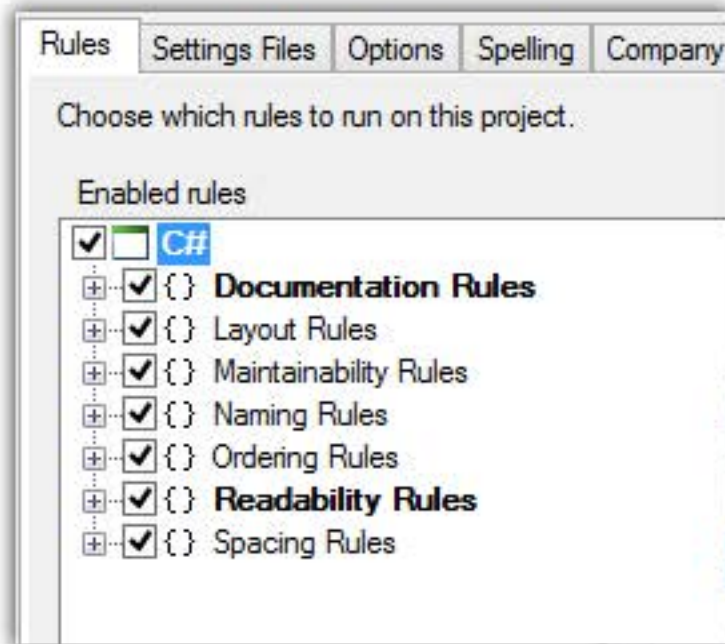
- Sidebar (Left):** A tree view showing the project structure. The "BattleFieldGameLib.Core" namespace is selected, listing various classes and interfaces.
- Main Pane (Top):** Displays the "BattleFieldGameLib.Core Namespace" and a "Send Feedback" link.
- Classes Table:** A table listing the classes in the namespace:

Class
<a href="#">Engine</a>
<a href="#">ExplosionHandler</a>
<a href="#">MineCreator</a>
<a href="#">MineFactory</a>
- Code Snippet (Right):** A snippet of C# code with XML documentation comments. The code defines two methods: `SetHitPosition` and `SetMine`. The `SetHitPosition` method sets the current position to the specified position. The `SetMine` method sets the current mine to the specified mine. The `HandleExplosion` method calculates the blast area and returns the number of mines taken out.

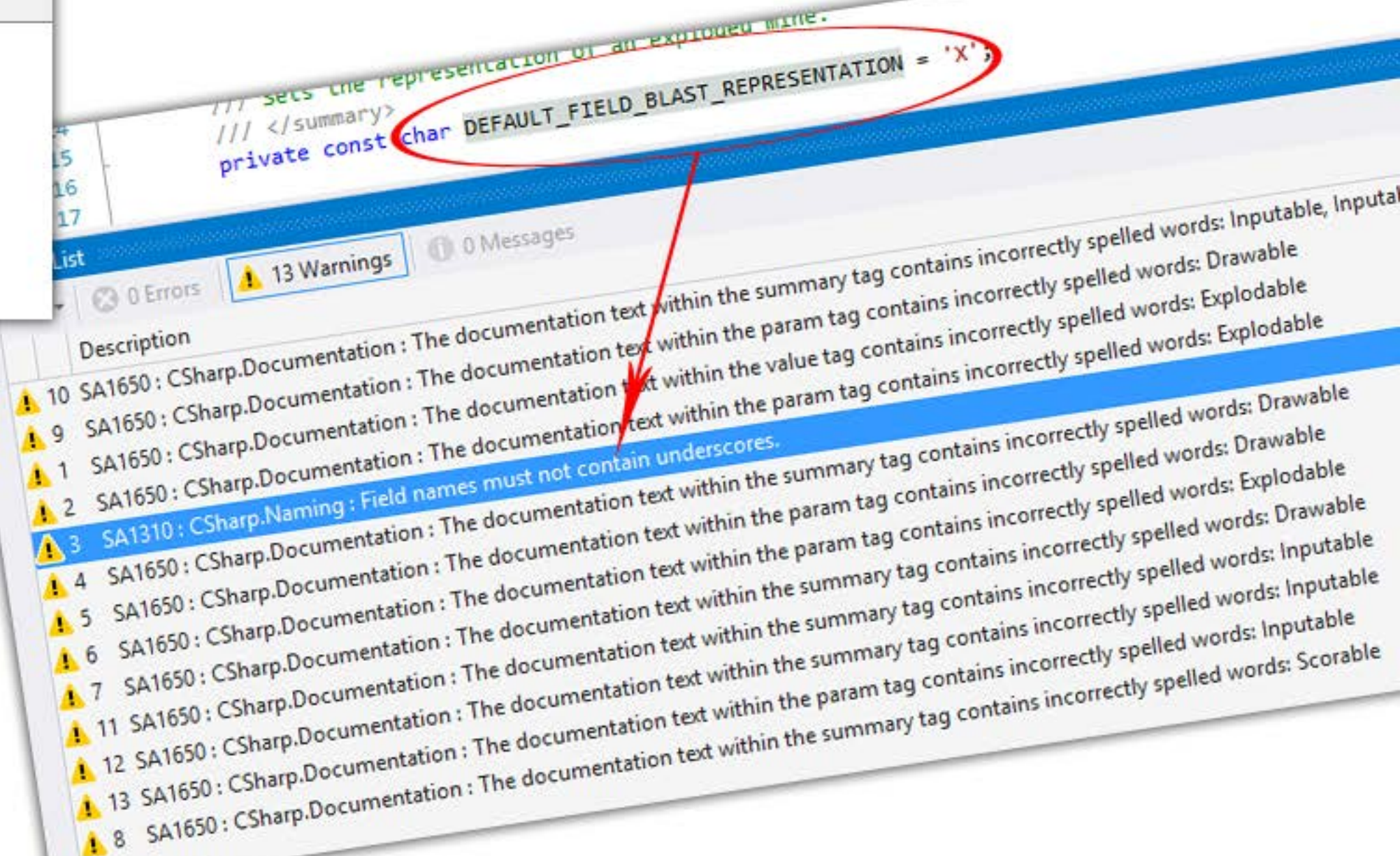
```
/// <summary>  
/// Sets the mine that will explode.  
/// </summary>  
/// <param name="mine">An instance of IExplodable.</param>  
3 references | 0/1 passing  
public void SetMine(IExplodable mine)  
{  
    this.CurrentMine = mine;  
}  
  
/// <summary>  
/// Walks through every field of the mines' blast area, if there is a mine  
/// blasted.  
/// </summary>  
/// <returns>The number of mines taken out by the current mine blast area.</returns>  
3 references | 0/1 passing  
public int HandleExplosion()  
{  
    try  
    {  
        int fieldLength = this.GameField.FieldBody.GetLength(0) - 1;  
        int offsetX = this.CurrentPosition.PosX - 2;  
        int offsetY = this.CurrentPosition.PosY - 2;  
        int[,] mineBody = this.CurrentMine.GetBlastArea();  
        int minesTakenOut = 0;  
        // Walks through every field  
        for (int row = 0; row < mineBody.GetLength(0); row++)  
        {  
            for (int col = 0; col < mineBody.GetLength(1); col++)  
            {  
                if (mineBody[row, col] > 0)  
                {  
                    minesTakenOut++;  
                }  
            }  
        }  
    }  
    catch { }  
    return minesTakenOut;  
}
```



# CODE QUALITY



ALL  
OPTIONS



StyleCop Warnings



# UNIT TESTING

```
/// <summary>  
/// Tests ExplosionManager class.  
[TestClass]  
0 references  
public class TestsExplosionManager  
{  
    ...  
}
```

Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
bf4-unittests.dll	68	14.59 %	398	85.41 %
bf4-unittests.dll	10	4.26 %	225	95.74 %
battlefieldgame.lib.exe	58	25.11 %	173	74.89 %

```
public void ExplosionConstructorShouldThrowExceptionOnNullGameFieldPassedAsArgument()  
{  
    var em = new ExplosionHandler(null);  
}  
  
/// <summary>  
/// Tests exceptions in SetHitPosition.  
[TestMethod]  
[ExpectedException(typeof(ArgumentNullException))]  
0 references  
public void SetHitPositionShouldThrowExceptionOnNullPassed()  
{  
    var explosionManager = new ExplosionHandler(new GameField(7));  
    explosionManager.SetHitPosition(null);  
}  
  
/// <summary>  
/// Tests exceptions in SetMine.  
[TestMethod]  
[ExpectedException(typeof(ArgumentNullException))]  
0 references  
public void SetMineShouldThrowExceptionOnNullPassed()  
{  
    ...  
}
```

# GITHUB

