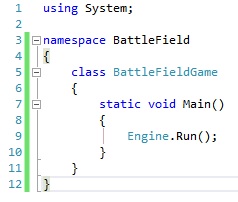
Refactoring Documentation for Project “BattleField”-1

Team “Arsenic”

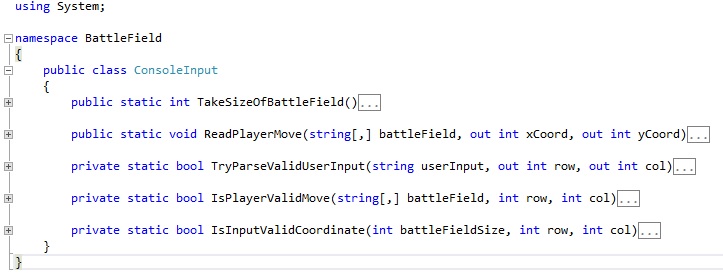
1. *Redesigned the project structure:*
   * Renamed the project from **Bombi** to **BattleField**.
   * Extracted each class in a separate file with a good name: **ConsoleInput.cs, Engine.cs, ExplosionGenerator.cs, Renderer.cs, MineType.cs, BattleFieldGame.cs.**
   * From the **Methods** class we extracted the classes: ExplosionGenerator.cs, ConsoleInput.cs, Engine.cs, ExplosionGenerator.cs, Renderer.cs; From **BattleFieldGame** was created the Engine.cs.
   * From the ExplosionGenerator class was created the MineType enumeration which holds the type (size) of the Mine.
2. *Reformatted the source code:*
   * First we get rid of all the unneeded empty spaces.
   * Formatted the code with StyleCop, JustCode and decreased the problems to minimum.
   * Inserted empty lines between the methods.
   * Put **{** and **}** after all conditions and loops.
   * Formatted the curly braces **{** and **}** according to the best practices for the C# language.
   * Changed the characters casing. Replaced variables with **camelCase**; Classes, enums, methods with **PascalCase**.
   * Formatted all other elements of the source code according to the best practices introduced in the course “High-Quality Programming Code”.

3. *Extracted methods in the new classes.*

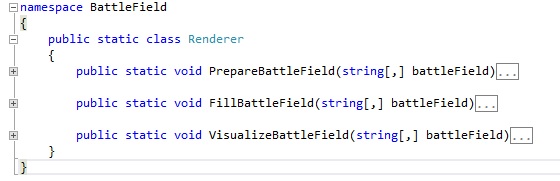
* In the BattleField: Only one method: Main() method which calls the Run method in Engine.cs:

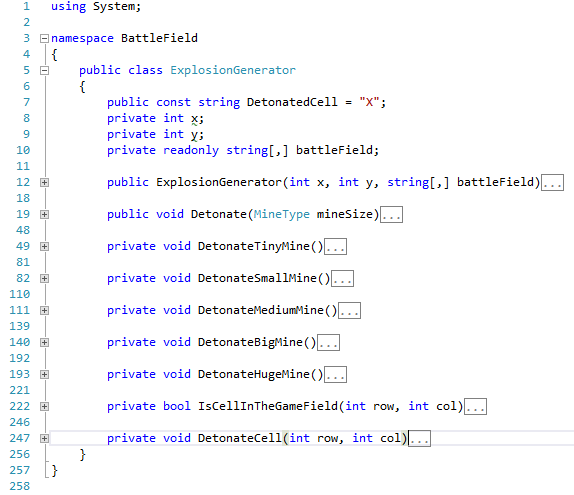


* In the Engine was created two methods: Run() and IsEndOfTheGame()
* In the ConsoleInput:



* In the Render:



* In The ExplosionGenerator:
* 

4. *Introduced const:*

* DetonatedCell = "X";

5. *Renamed variables:*

* In all the class names of the variables are changed.
* Example in Engine.cs:
* Method "Krai" => IsEndOfTheGame();
* Boolean variable: край => isEndOfTheGame;