TeamWork - Battle Field 7

Active contributors:

* Lokikko91 (forum: Siko, GitHub: Siko91) : Alexander Dinkov
* vbahovski (forum: ??? , GitHub: vnci) : Ventsislav Bahovski

**May 31, 2014**

* First readability improvements on the original source
* Fixed BattleFieldGame.cs's summary

### Jun 12, 2014

* Split the Game to UserInterface, Engine and Field. Filled most of them with //TODO notes and NotImplementedExeption-s
* Implemented a better method of explosion blast range calculations

### Jun 28, 2014

* Implemented most of the //TODO tasks in the Engine

### Jun 29, 2014

* Added some XML documentation
* Implemented ConsoleUI (child of IUserInterface) which displays the game in the console.  
  The Game is once again runnable
* Started the implementation of WinFormsUI

### Jun 30, 2014

* Added more XML documentation

### Jul 01, 2014

* Seperated the logic required for running the different versions of the game in a different class. It’s kind of like a factory, but doesn’t produce anything. I guess it’s more like a facade

### Jul 12, 2014

* Added a static Logger Class. The idea was to use it everywhere.

### Jul 13, 2014

* More XML documentation
* Converted the Logger to a non-static class. Only the Engine and the class that runs the versions will use it for now on.
* Removed the unnecessary methods defined in IUserInterface.
* Removed engine’s dependency on the Logger. Now it can run without it.

### Jul 15, 2014

* Added Field’s Unit tests.
* Added Logger’s Unit tests.
* Added the following interfaces:  
  - IField : IClonable  
  - IExplosionStrategy  
  - IBattleField  
  - IEngine
* Set the stage for a better design

### Jul 16, 2014

* Added SimpleField. It implements IClonable.
* Added SimpleExplosionStrategy. It accepts a position and explosive power, and returns a collection of coordinates to detonate.
* Added SimpleBattleField. It accepts a base IField and an IExplosionStrategy. The field is used via cloning. The explosion strategy is used on every detonation.
* Added Simple Engine. It Runs the game with an IUserInterface and an IBattleField. It also logs if given an ILogger.
* Added SimpleField’s Unit Tests
* Added SimpleExplosionStrategy’s Unit Tests
* Added SimpleBattleField’s UnitTests
* Fixed some of the discovered flaws in the code.

### Jul 17, 2014

* Fixed more of the discovered flaws in the code.
* Added a class Coords2D and started using it everywhere, where coordinates are required.
* Added [Obsolute] tag to the following:  
  - The initial class (the horrible one)  
  - Engine (the first version)  
  - Field (the first version)  
  - Field’s Unit tests

### Jul 18, 2014

* Fixed some of the discovered flaws in the code
* Replaced Coords2D with Tuple<int, int>
* Added [Obsolute] tag on Coords2D
* Added the following interfaces:  
  - ICountObserver  
  - ICountNotifier
* IEngine implements ICountObserver
* IBattleField implements ICountNotifyer
* Moved the “BombsCount” property from SimpleEngine to SimpleBattleField and made it so that the Engin would be notified about any changes on that property.
* Fixed 70+ JustCode warnings
* Generated Sandcastle documentation.
* Added a code dependency map.
* Added [this] documentation file to the project.