

## Refactoring Documentation for Project “Minesweeper-2”

---

<https://github.com/TeamMinesweeper2/Minesweeper>

Name	Team	
	Telerik ID	Github
Jivko Rusev	neutrino	neutrino-git
Nikolay Nachev	nikolay.n.nachev	Nachev
Jasen Bashov	triplexsa	jasonbashov
Viktoria Lyomcheva	viktoria_lyomcheva	amelia77
Lachezar Nikolov	l.nickolov	lsnickolov

### Implemented design patterns:

#### ✓ *Creational:*

- Singleton: [ RandomGeneratorProvider ]
- Factory Method: [ MinesweeperGameEasy : MinesweeperGame ]

#### ✓ *Behavioral:*

- Command: [ CommandExecutor, MinesweeperGame, ICommand: CmdBoom, CmdExit, CmdOpenCell,... ]
- Strategy: [ IEnumerable<T>.Shuffle method takes IRandomGeneratorProvider ]

#### ✓ *Structural:*

- Adapter: [ IRenderer: ConsoleRenderer(System.Console) ]
- Facade: [ MinesweeperGame is Facade for UIManager, Minefield, ScoreBoard ]
- Bridge: [ UIManager is bridged with IRenderer and IUserInputReader ]

### Followed SOLID principles:

**Single responsibility:** all classes have narrowly defined responsibilities;

**Open-closed:** realized through the Factory Method and Command pattern;

**Liskov substitution:** the two derived classes (MinesweeperGameEasy and MinefieldEasy ) follow the contract of their base classes;

**Interface segregation:** all interfaces define only the minimum number of methods needed;

**Dependency inversion:** realized with Strategy pattern (passing the dependency in the constructor).

### Followed DRY principle:

- The minefield is drawn by the same method regardless of the game state (initial, playing or exploded);
- Opening and flagging of cell goes through the same method, which takes a delegate to perform the specific action
- Some methods on arrays implemented as extension methods to avoid repetition

**Code coverage of 90% with unit tests.**

## Refactoring history

1. Reformatted code, removed labels, removed comments.
2. Removed IsFoundInRandomNUmbers.
3. Removed int[,] state, and added isAlive.
4. Removed unnecessary matrix initialization.
5. Renamed variables and methods.
6. Changed mineField to bool[,].
7. Top scores done with SortedDictionary.
8. Formatted curly braces.
9. Added Position class to hold the position in the game matrix.
10. Implemented Position in CountNeighborCell method.
11. Added and implemented class ConsoleManager and enumeration ErrorType.
12. Removed board drawing from Main.
13. Minesweeper game made non-static.
14. Position class renamed to Cell and made struct.
15. Extracted CommandReader class.
16. ConsoleManager class made non-static.
17. Renamed methods in ConsoleManager class.
18. Extracted Minefield class.
19. Minefield class decoupled from ConsoleManager.
20. Minefield size is set in constructor.
21. ConsoleManager initialized in constructor.
22. Added IUserInputReader interface.
23. Added BoardDrawer class.
24. Added IRandomGeneratorProvider interface.
25. RandomGeneratorProvider implemented as Singleton and used in minefield.
26. Added ICell interface and implemented Cell class.
27. Removed hard coded messages from ConsoleManager.
28. Moved all interfaces to Interfaces folder.
29. Add constants for all commands in CommandReader class.
30. Used Split string and TryParse when parsing user input.
31. Added restart, show score and exit game methods.
32. Modified Minefield class to use ICell and shuffle the mines.
33. Added IRender and implemented ConsoleRenderer.
34. Fixed bug in Minefield - missing cell initialization.
35. Fixed bug with exception in Shuffle method in the Minefield class.
36. Deleted the rows/cols properties/fields from Cell class.
37. Added ClearLines method to IRenderer interface.
38. UIManager is now decoupled from the System.Console.
39. Added 'boom' command which reveals all mines as if a mine was opened.
40. Moved independent classes to a class library Minesweeper.Lib

41. Added CellImage enumeration and GetImage method to the Minefield class.
42. Initial, game and final minefield are now drawn with one method.
43. Implemented Command Pattern – commands are now classes of type ICommand.
44. Added FlagCell command to enable marking of cells.
45. Added method for game end when all empty cells are opened.
46. Renamed method GetValueCount to ConvertMinefield and changed its return type to Generic.
47. Scoreboard limited to 5 places (add a new score, sort and take the top five).
48. Added setters for rowCount, columnsCount, numberOfMines in Minefield class.
49. Converted method GetOpenedCells to property in Minefield class.
50. First cell will not have mine (by transferring to a random empty cell).
51. Open empty cells recursively (save the user extra effort to open obvious empty cells).
52. Added interface for CellPosition.
53. Renamed ICellPos to ICellPosition.
54. Changed input type for FlagCellHandler to ICellPosition.
55. Added unit tests and xml documentation.
56. Ran StyleCop with default rules and corrected warnings.
57. Moved ICellPosition to Interfaces folder.
58. BoardDrawer constructor takes 1 parameter instead of 4
59. UIManager ctor takes 1 parameter instead of 3
60. UIManager has two constructors
61. CommandParser.ParseCommand made virtual
62. CommandParser.Game made protected
63. Extracted shuffling from Minefield to ArrayExtensions class in Minesweeper.Common
64. Renamed method GetRandomNumber to Next.
65. Refactored Minefield class - combined methods OpenCellHandler and FlagCellHandler.
66. TopScores functionality extracted to ScoreBoard class.
67. All game messages stored as constants in a static Messages class
68. MinefieldState enum renamed to CellActionResult
69. MinesweeperGame uses Factory Method to create Minefield.
70. Moved recursive opening of cells to MinefieldEasy class.

commit 446437b