# Documentation for Project “Game 15”

*Team “Holmium”*

Линк към презентацията на проекта: <http://prezi.com/xhh4ojg1kgej/game-fifteen-7/>

Линк към проекта в GitHub:

<https://github.com/telerik-hqCode-teem/HQCodeProjectTelerikAcademy>

## Project Overview

Your task is to write an interactive **console-based implementation of the game “Fifteen”** in which the player tries to **arrange the numbers from 1 to 15 sequentially** in a square **matrix of size 4 x 4**. The game starts from a **randomly shuffled** 15 numbers in a matrix (4 x 4) and a random cell left empty (see the figure below). At each turn the player enters a number from the matrix, neighbor to the empty cell that should move over it. The game finishes when the numbers are arranged sequentially (see the figure). When the game is finished, a new game automatically starts.

## Redesign the Project Structure

* Create Console Application ***GameFiffteen.Demo*** for Project Main Execution
* Extract Existing Code in Class Library
* Create Project ***GameFiffteen.Test*** for Unit Testing the Project logic
* Referencing ***GameFifftien.Demo*** with ***GameFifftien*** Class Library and ***GameFifftien.Test*** with ***GameFifftien*** Class Library

## Code Formatting

* Use build in Visual studio code formatting tool for automating code formatting
* Character casing: variables and fields made **camelCase**; types and methods made **PascalCase**.
* Change wrong placed curly brackets with c# convention positioning

## Proper Code Naming

* Change single Latter Variable Names with their Full Name Equivalent like so:
  + ***a*** replaced with ***board***
  + ***топКандидати*** replaced with ***topPlayers***
  + ***x*** and ***y*** replaced with ***rows*** and ***cols***
  + ***flag2*** replaces with ***isNeighbou***
  + ***broqach*** replaced with ***movesCounte***
  + topCount replaced with ***topPlayersMovesCounter***
* Change Methods names with proper descriptive Names like so:
  + checkrenamed to ***isValideNeighbour***
  + Moverenamed to ***MoveNumber***
  + ***check2*** renamed to ***CheckBoardRange***
  + startagain renamed to ***StartNewGame***
  + ***move*** renamed to **SetTopPlayersList**

## Commenting and Code Documentation

* Comment all newly extracted and created methods with default Microsoft comments
* Add Extra Head Comments for separating Code Sections

## Class Refactoring

* Create new Class ***Game*** for handling Game Logic
  + Create Game constrictor for Game initialization
  + Add method ***GameLoop()***
  + Add method ***StartGame()***
  + Add method ***ReadInput()***
  + Add method ***IsNumeric Inp***ut()
  + Add method ***GameRestart(***)
  + Add method ***GameScore()***
  + Add method ***ScoreEnter()***
  + Add method ***SystemInput()***
* Create new Class **ScoreBoard**
  + Create new ScoreBoard Constructor
  + Extract method ***SetTopPlayersList() to ScoreBoard Class and renamed it to SetPlayerScore()***
  + Create a method ***ToString()*** that overwrite ***ToString()***
* Create new Class **PlayBoard** and extract the foaling Methods:
  + Create Constructor **PlayBoard**
  + Set private property gameBoard to hold Matrix 15/15
  + Create a method ***ToString()*** that overwrite ***ToString()***
  + Extract method ***DrawNumbersOnBoard()*** *to the new implementation of* ***ToString()***
  + Create indexation for handling matrix operations
  + Introduce new method ***PlayBoardGenerator()*** for random board generation
  + Introduce new method ***ShuffleBoardNumbers()*** for utilize random number generation
  + Introduce new Boolean method**IsMoveInBoardRange()** for checking the end of the game session
  + Refactor the **ChangePlayBoardLayout()** for proper number rearrangement
  + Introduce new method **BoardTraversRow** for finding the proper game board row
  + Introduce new method **BoardTraversCol** for finding the proper game board col

## Unit Testing

* Create Unit Tests for testing ScoreBoard public Methods
* Create Unit Tests for testing PlayBoard public Methods