# Sample Refactoring Documentation for Project “Balloons-Pop-6”

Team “Lawrencium”

1. Redesigned the project structure:
   * Renamed the project to **BallonsPopGame**.
   * Renamed the main class **Program.cs** to **BallonsPopMain.cs**.
   * Separated functionality for each class: **BallonsPopMain.cs**, **GameEngine.cs**, **Score.cs**, **Player.cs.**
   * Added Test project: **BalloonsPopTests.**
   * Added unit tests and data-driven tests classes to the **BalloonsPopTests** project:  **PlayerTests.cs, GameEngineTests.cs, ScoreTests.cs**
2. Refactored the source code:
   * Removed all unneeded empty lines in the project.
   * Inserted empty lines between the methods.
   * Cleared most style cop warnings.
   * Added unit testing project.
   * Split the lines containing several statements into several simple lines.
   * Formatted the curly braces **{** and **}** according to the best practices for the C# language.
   * Put **{** and **}** (when missing) after all conditionals and loops.
   * Character casing we changed variables and fields to **camelCase**, names of types and methods changed to **PascalCase**.
   * Formatted all other elements of the source code according to the best practices introduced in the course “[High-Quality Programming Code](http://codecourse.telerik.com/)”.
   * Balloon popping is now made with 1 recursive method instead of 4 (for each direction).
   * Separated main method from the engine.
   * Useless comments removed.
3. Renamed methods:

Original name 🡪 **Renamed**

* gen 🡪 **GenerateMatrix()**
* printMatrix 🡪 **GetMatrixImage()**
* checkLeft 🡪 **CheckNeighboringFields ()**
* checkRight 🡪 **CheckNeighboringFields ()**
* checkup 🡪 **CheckNeighboringFields ()**
* checkDown 🡪 **CheckNeighboringFields ()**
* change 🡪 **IsPopped()**
* doit 🡪 **IsWinner(), and DropDownMatrix()**
* sortAndPrintChartFive 🡪 **PrintScoreBoard()**
* signIfSkilled 🡪 **IsGoodEnough()**

1. Extracted the methods all methods from the method **Main()** made non static.
2. **NameValuePair** renamed and modified to **Player**
3. Introduced class **GameEngine()** and moved all related functionality in it (**GetMatrixImage(), GenerateMatrix(), CheckField(), CheckNeighboringFields(), IsPopped(), IsFinished(), DropDownMatrix(), ProcessGame(), isInputValid()**).
4. Added XML Documentation