**Refactoring Documentation for Project “Labyrinth - 4” of Team Neptunium**

1. Redesigned the project structure:

- More accurate names of classes were given (e.g. “KPK-PROEKT” to “LabyrinthGame”, “LabyrinthProcessor” to “LabirinthEngine”, etc.);

- ShowLabyrinth was moved from LabyrinthGame to LabyrinthEngine, because it has to print and manage all processes in the game.

- We have moved the input of the direction you are going in Handle Input

2. Reformatted the source code:

- A few wrongly implemented conditional statements were corrected (e.g. (!(matrix.MyPostionVertical == 6)) to (matrix.MyPostionVertical != 6));

- List<Tuple<uint, String>> scoreboard was made private in order to preserve it from external access;

- LabyrinthGame was changed to non-static in order to use it in LabyrinthEngine;

- Inconsistent use of the the Method of the tuple count() and it’s property Count was noticed and corrected;

- Added “this.” prefix in several places in order to follow the best practices (e.g. in ScoreBoard method Handle);

- Usage of “String” replaced with “string”;

- In HandleScoreBoard we have put “else if” instead of only “if”, because only one of them will

happen anyway;

- We have removed “.ToString()” from the “ShowScoreboard()” from the result which contains best players, because Console.Writeline calls it anyway;

- scoreBoard was changed from a jagged array to a two-dimensional array (7x7 matrix);

- We have simplified the “if” statements in the move Methods in the engine by adding some new variables;

- We have refactored the HandleInput method of the engine class and the four moving methods which is it related to. This way we manage to get rid of the “if” statements in the “switch case” construction;

- LabyrinthMatrix Line 22 - fixed loop - “i” replaced with row, “j” replaced with col.

- GetRandomSymbol() - Method Refactored ! Symbols are moved in array as readonly fields

- random renamed to randomGenerator and refactored to readonly field;

- Added **CreateRandomMatrix Method** - generate random matrix;

- Added **CountLabyrinthExits Method** - count all posible exits, used to check if generated random labyrint matrix have any exits;

- Modified **GenerateLabyrinthMatrix Method** - this method combines *CreateRandomMatrix* and *CountLabyrinthExits*. Generated labyrinth is guarantee to have an exit, so game can be finished in all cases.

3. Fixing errors with styleCop

- We have run StyleCop and fixed most of the meaningful mistakes (more than 150).

4. Object Oriented modifications:

- We have added 3 new interfaces, we increased the abstractions of the game, this way another matrix, another scoreboard, and another player can be used and they all will have the same structure.

- Class TopFiveScoreboard implements IScoreboard, LabyrinthPlayer implements IPlayer and LabyrinthMatrix implements IMatirx with the right methods.

- We have moved the showNameInputMassage from topFiveScoreBoard class to decrease the coupling from the console and scoreboard and to make it more testable, we have also included property of the scoreboard field to able to get it, but of course with private set to change it only from the current class.

5. Unit Testing

- We have made full testing of the methods of the Player class.

- We have added test for the scoreboard class, to test the public methods.

- Unit Testing Added To ***LabyrinthMatrixTest*** :

**- LabyrinthNoExitCountTest** - Testing results given by *CountLabyrintExits Method* in case of labyrinths with no actual exit ( starting point - middle of the labyrinth);

**- LabyrinthWithExitCountTest** - Testing results given by *CountLabyrintExits Method* in case of labyrinths with different amount of exits ( starting point - middle of the labyrinth)

- Added **LabyrinthRenderer Class** : used to draw on labyrint on console, independently from other classes. Inside this class added Console.Clear() so game have better looks;

- Unit Testing Added To ***LabyrinthRendererTest :***

- **TestLabyrinthConsolePrin**t - used to test how accurate the renderer print given matrix, with player on it;

- LabyrinthEngineTests, LabyrinthScorboardTests and UserInterfaceSimulation added