**Refactoring Documentation for Project “King Survival”**

*Team “Nickel”*

**Redesigned the project structure:**

* Renamed the project to KingSurvivalGame
* Renamed the main class Program to KingSurvivalGame.
* The whole logic of the game was divided into separate classes.
* Extracted each class in a separate file with a proper name:

- KingSurvivalGame.cs

- GameBoard.cs

- Engine.cs

- Figure.cs

- Pawn.cs

- King.cs

- Position.cs

- Direction.cs

**Reformatted the source code:**

* Removed all obsolete comments.
* Replaced all “bad” names of methods and variables with appropriate and meaningful ones.
* Improved code readability
* Removed "Magic numbers" - all used "magic" numbers are declared as constants with appropriate names.
* Improved code maintainability.
* Inserted empty lines between the methods.
* Removed all unnecessary empty lines.
* Formatted the curly braces {and } according to the best practices for the C# language.
* Put { and } after all conditionals and loops (when missing).
* Character casing:

-variables and fields made **camelCase**.

-types, methods and classes made **PascalCase**.

**Changed game logic:**

* Removed bug - When the king is blocked, the player gets a message that the king loses. When all pawns don't have available moves the same happens.
* Figure logic is now in separate classes.
* Added "Responsive design" to the game board - now it can be resized up to 26x26 (that's why the numbers for the coordinates are replaced with lower and upper English letters.
* GameBoard is now non-static: In the beginning of the game we create an instance of from the class GameBoard with all the figures on it. This way we can add new figures.
* Moved method *GenerateRandomNumber(int start, int end)* to separate class RandomUtils.