**CODE SMELLS**

**Diogo Correia 62475**

**Enhancing Expressiveness:**

Within the confines of the codebase, there's an intriguing instance of primitive obsession. Take, for example, the utilization of a primitive type like an integer to encapsulate the concept of save game periods within the autoSaveGame method in InGameController.java. A more nuanced approach would involve crafting dedicated classes, like a SaveGamePeriod class, to encapsulate such information. By doing so, not only does the code become more lucid, but it also opens avenues for robust validation mechanisms.

**Refactoring Long Methods:**

Nestled within the labyrinth of code, a lengthy method named moveDirection in InGameController.java has surfaced. This method weaves through an array of conditional checks and diverse actions. A judicious approach here involves the surgical division of this behemoth into smaller, more specialized methods. Such a stratagem not only elevates the readability of the code but also facilitates easier maintenance down the line.

**Mitigating Duplicated Code:**

An intriguing case of duplicated code manifests within the Flag class in the src.net.sf.freecol.client.gui.dialog package. Specifically, both the drawStripes and drawQuarters methods exhibit repetitive lines of code pertaining to g.setColor and rectangle.setRect. A pragmatic antidote to this redundancy conundrum is to fashion handy helper methods, effectively excising the duplicated snippets and fostering a more streamlined and maintainable codebase.