**Code Smells:**

João Esteves 47994

1. **Primitive Obsession:**

Nesta classe podemos verificar que os tipos primitivos (como inteiros) são usados para representar conceitos específicos. Uma solução possível seria criar classes específicas para esses conceitos, tornando o código mais expressivo.

Como é o caso do int *saveGamePeriod* contido no método *autoSaveGame* na linha 886, da classe *InGameController.java* no package *src.net.sf.freecol.client.control,* este é usado para representar informações como períodos de salvamento. Usar um tipo de objeto que encapsula essa informação, como uma classe *SaveGamePeriod*, que poderíamos criar, de forma a tornar o código claro e permitir as validações mais robusto.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Fig1. Parte do código do método *autoSaveGame* onde se chama o inteiro *saveGamePeriod.*

1. **Long Method:**

**O método *moveDirection*  da classe** *InGameController.java* **contida no package** *src.net.sf.freecol.client.control* que começa na linha 1315 acabando só na linha 1496, **vê-se logo que é um método extenso e verifica-se que este faz muitas verificações condicionais e ações diferentes. Dividir esse método em métodos menores e mais específicos seria uma melhoria possível quer em termos de legibilidade quer na manutenção do código.**

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, menu, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Fig2 a 8 Representação completa do método  ***moveDirection.***

1. **Duplicated Code:**

Na classe Flag contida no package src.net.sf.freecol.client.gui.dialog há várias secções de código que se repetem, por exemplo os métodos drawStripes e o método drawQuarters possuem os dois a repetição das mesmas linhas de código g.setColor e o rectangle.setRect. Uma solução possível seria criar métodos auxiliares, para eliminar estes trechos de código repetido.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Fig9 Método *drawStripes*

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Fig10. Método drawQuarters