Code Smells

Code Smell 1 – Long Method

Path: net.sf.freecol/client/gui/action/ActionManager.java

O método “initializeActions”, apresentado a seguir, tem mais de 100 linhas de código, ou seja, tem uma grande complexidade e dificuldade de leitura por parte de programadores.

```
public void initializeActions(InGameController inGameController,
                              ConnectController connectController) {
    /**
     * Please note: Actions should only be created and not initialized
     * with images etc. The reason being that initialization of actions
     * are needed for the client options ... and the client options
     * should be loaded before images are preloaded (the reason being that
     * mods might change the images).
     */

    /**
     * Possible FIXME: should we put some of these, especially the
     * move and tile improvement actions, into OptionGroups of
     * their own? This would simplify the MapControls slightly.
     */

    // keep this list alphabetized.
    add(new AboutAction(freeColClient));
    add(new AssignTradeRouteAction(freeColClient));
    add(new BuildColonyAction(freeColClient));
    add(new CenterAction(freeColClient));
    add(new ChangeAction(freeColClient));
    add(new ChangeWindowedModeAction(freeColClient));
    add(new ChatAction(freeColClient));
    add(new ClearOrdersAction(freeColClient));
    for (PanelType panelType : PanelType.values()) {
        add(new ColopediaAction(freeColClient, panelType));
    }
    add(new ContinueAction(freeColClient));
```

...

Para resolver o code smell, o método poderia ser separado em vários métodos menores, facilitando a sua leitura e compreensão.

Code Smell 2 – Large Class

Path:net.sf.freecol/common/model/Tile.java

A seguinte classe contém mais de 2800 linhas, o que o torna muito complexo e difícil de ler. Para além disso, também viola a regra da única responsabilidade, fazendo demasiadas ações para apenas uma classe.

```
/**
 * Represents a single tile on the {@code Map}.
 *
 * @see Map
 */
Michael Pope +15
public final class Tile extends UnitLocation implements Named, Ownable {

    4 usages
    private static final Logger logger = Logger.getLogger(Tile.class.getName());

    public static final String TAG = "tile";

    /** Comparator to sort tiles by increasing distance from the edge. */
    1 usage
    public static final Comparator<Tile> edgeDistanceComparator
        = Comparator.comparingInt(Tile::getEdgeDistance);

    /** Comparator to find the smallest high seas count. */
    1 usage
    public static final Comparator<Tile> highSeasComparator
        = Comparator.comparingInt(Tile::getHighSeasCount);

    /** Predicate to identify ordinary sea tiles. */
    1 usage
    public static final Predicate<Tile> isSeaTile = t ->
        !t.isLand() && t.getHighSeasCount() >= 0;
```

...

Para resolver o code smell, a classe poderia ser dividida em classes menores, separando as responsabilidades por cada nova classe. Dessa forma, não existiria uma classe que faz demasiadas açoes.

Code Smell 3 – Magic Number

Path: net.sf.freecol/server/model/ServerPlayer.java

Na linha 606, o número 7 é chamado de "magic number". Não tem explicação direta no código e pode ser difíceis de entender a sua utilidade.

```
// OK, the situation is poor, but the REF hangs on if it has
// a minimal number of land and naval units
final int landREFUnitsRequired = 7; // FIXME: magic number
if (land < landREFUnitsRequired) {
    logger.info( msg: "REF surrenders due to land army collapse ("
        + land + " < " + landREFUnitsRequired + ")");
    return true;
}
```

Para resolver o code smell, a variável “landREFUnitRequired” deveria ser inicializada com “private static final int”, anteriormente.