Design Patterns

# Singleton Pattern – FSGConverter.java (src/net/sf/freecol/tools/FSGConverter.java)

The FSGConverter class exemplifies the Singleton design pattern due to its embodiment of essential Singleton characteristics. Its private constructor, private static instance, lazy initialization, synchronized access, and global access point all align with the pattern's key principles.

Uma imagem com texto, captura de ecrã, software

Descrição gerada automaticamente

# Factory Pattern – ResourceFactory.java (src/net/sf/freecol/common/resources/ResourceFactory.java)

The ResourceFactory class is a central factory that creates different Resource objects based on the given parameters. This way, the creation process is hidden, and the code is organized. This pattern also increases flexibility, hides resource creation, and simplifies resource management in the FreeCol project.

Uma imagem com texto, captura de ecrã, software

Descrição gerada automaticamente

# Template Pattern – Resource.java (src/net/sf/freecol/common/resources/Resource.java)

The Resource Class is implemented by 8 other classes, as seen in the image. All the 8 implementations utilizes the common structure set by the Resource Class and make change to the methods if needed. That’s a clear use of the template pattern.

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente