**José Pereira 55204**

**Design Patterns**

1. **Command Pattern** (src/net/sf/freecol/client/gui/action/ActionManager.java)

A screenshot of a computer

Description automatically generated

**Abstract class location:** src/net/sf/freecol/client/gui/action/FreeColAction.java

A screen shot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

Since every action extends this abstract class and with every new action added we have the keyword “new” such as in “new AboutAction(freeColClient)”, it showcases a concrete implementation of a command. The calls to “add” are encapsulating a request as an object.

1. **Strategy Pattern** (src/net/sf/freecol/client/gui/action/ActionManager.java)

A screen shot of a computer code

Description automatically generated

The ActionManager class also uses glimpses of the Strategy pattern through its interaction with FreeColAction objects.

Different actions encapsulate different behaviors that can be executed.

It manages a collection of strategies (FreeColAction objects) and allows invoking them based on certain conditions or inputs.

1. **Facade Pattern** (src/net/sf/freecol/client/control/SoundController.java)

A screen shot of a computer program

Description automatically generated

The **SoundController** class simplifies the interface for playing sounds and music by providing methods like **playSound** and **playMusic**. It hides the complexities of the **SoundPlayer** class and the underlying audio system from the rest of the application - a key aspect of the Facade Pattern. Clients of SoundController don't need to deal with mixers, audio files, or sound resources directly.

A diagram of a sound controller

Description automatically generated