The Use of Design Patterns

The use of specific design patterns are essential to help build a clean and organized code structure. Although we tried to maintain all GRASP guidelines, we were unable to capture some of these design aspects due to the lack of time. The design patterns used and potential improvements are outlined below:

Information Expert(the expert principle) has been used in our code to delegate specific responsibilities to methods, computer fields and so on.

We tried to code using **Low coupling** and **High cohesion** design patterns by creating separate classes for different game objects - the board, players, settings etc. If we had ample time, we would have tried to demonstrate a simpler code design with fewer dependencies on the classes.

The **Protected Variations Pattern** protects elements from the variations on other elements which given more time we could've implemented by maintaining a cohesive gameplay for all the different settings of the game.

We used a **Controller** class(*GameBoard*) which reacts to the input events and provides appropriate response.

Polymorphism has also been implemented in our code as we tried to use common methods for both the human and computer players.

The "Save the Game" and "Load the Game" implementation uses **Pure Fabrication** as it connects to a .csv database where data can be stored and retrieved from.