Design Pattern

Information Expert: This design has been implemented in the structural code of the ‘Game of Focus’ project. Specific responsibilities have been assigned class wise with their behavior transmitted throughout. For example, the “SaveStorage”, the “Pawn” and “Player” classes are responsible for updating attributes of their own instances via threading and these classes include unique methods which can hold information related to their working pattern. This is how the Information Expert is supported.

Low Coupling: Low coupling is present among most classes, and could be reduced by adding interfaces, for example, a CommandHolder.

High Cohesion: There is a good number of classes that maintain relationship with each other but perform their own responsibilities, beneficial for reuse and maintenance of code.

Controller: There are a few controllers designed within the current release, such as ‘Offboard’, which is responsible for Keeping track of pawns reserved or captured by players. Controllers for other system events such as changing color scheme or player count may be implemented in future releases.

Protected Variations: Some interfaces have been implemented, structure hiding by creating additional methods within classes to hold chained method calls has been used, supporting the pattern of Protected Variations.

Creator: The ‘Game’ class instantiates the classes for viewing game options or game rules and is also solely responsible for the creation andstorage(saving) of new games.