Design Patterns

1. Information Expert – In the game, the information expert pattern has been used countless times, for example, the GameBoard class in the game contains information for a class like Domino.

2. Creator – In the game, the GameSetup class is the creator class. It creates instances of different classes in order to setup the game.

3. Controller – In the Use Case to place a domino, the class Gameboard is used as the controller that facilitates all the primary input/output in the use case.

4. Low Coupling – Most of the code in our game aims for low dependency among the classes but we could have focused more on lowering the coupling among the classes in our code.

5. High Cohesion – In our game we have made sure that each class is responsible for only one task. For example, we have a class solely to function the Gameboard, there's a class that solely functions the dominoes and etc.

6. Indirection – The GridSquare class in our code is an example of indirection, it is a form of Pure Fabrication that allows higher cohesion and lower coupling in the gameboard classes.

7. Protected Variations – There is heavy use of encapsulation in our code which is a form of protected variation, for example, to avoid clashing between components such as JButtons and JFrames on the game GUI.

8. Pure Fabrication – In our code, the class GridSquare is a pure fabrication in order to avoid complication in the GTPG and GFPG class