To make the game extensible, I introduced a GodCard interface, which defines methods for altering default rules for move, build, and checkWinCondition. Each specific GodCard (e.g., Minotaur, Demeter) implements this interface, encapsulating unique rules while adhering to a common structure. The game dynamically interacts with players' selected GodCards, delegating proper behavior based on each god card's rule, ensuring that new cards can be added without modifying the core game logic. This design uses the **Strategy Pattern**, allowing the game to delegation logic to interchangeable GodCard implementations, promoting **extensibility** and adherence to the **Open/Closed Principle**. Alternatives like hardcoding rules or creating separate logic classes were considered but wasn't used due to increased coupling and reduced maintainability. By encapsulating behavior in GodCards, the system achieves modularity, scalability, and alignment with the domain model, making it easier to test, maintain, and extend in the future.