

To make the game extensible, I implemented a GodCard interface, which defines methods for modifying rules such as move, build, and checkWinCondition. Each specific GodCard (e.g., Minotaur, Demeter) implements this interface and provides its own rules. Using the Decorator Pattern, GodCards wrap the default behavior of workers, and it extends their actions like movement and building. This approach allows the game to support new GodCards by adding them as decorators without changing the core game logic. By decorating workers with GodCards, the design follows the Open/Closed Principle, making it possible to extend functionality without modifying existing code. Alternatives like subclassing workers or embedding rules directly in the game class were considered but would have led to higher coupling and less flexibility. The Decorator Pattern ensures that each GodCard focuses on its own behavior and can be combined dynamically, creating a flexible and scalable design.