

CSCD349: Project Patterns Used

The α - Team

June 10, 2015

Contained in this document are the design patterns used when creating our game.

1 Strategy

Strategy Pattern is utilized in various places throughout the project. Some notable examples are:

- The Engagement class tells a character to select their action. Whether the character is a protagonist or antagonist, hunter or newt, all can be asked to select their action.
- Items have a use method. The use method behaves differently if the item is a healing item, a weapon, or a shield.
- Game has a reference to a GameClient interface. Game clients can be changed from console to GUI, etc, and the Game class doesn't have to behave any differently.

2 Template

Template pattern is used for protagonists' actions. Each protagonist's turn consists of the same template of actions, but those actions are different for each subclass. For example, a Hunter can choose to aim, shoot, or piercing shot. A Warrior can attack, block, or whirlwind strike. Hooks are also used in this template method (selectAction).

3 Factory

There are a couple factories in the project. There is a factory for creating characters (protagonists and antagonists), and a factory for creating parties of characters, the latter of which uses the former.

4 Singleton

The game logic for the project is encapsulated in the Game class, and delegated to the GameClient when need be. There should ever only be one instance of Game, but multiple classes should have a reference to it. Therefore, a singleton pattern was used for the Game class. Any class can get the game instance with Game.getInstance(), and it will always be the same instance.

5 Iterator

The Party class is iterable through the members of the party.