Bryan Anders
Sami Awwad
Samir Ouahhabi

Team Abeeseesdee

Final Project Design Pattern Choices

1) Strategy
   - Items, Equipment, Abilities, Characters and Parties all implement strategy to make them easily swapable with other objects of their type.
2) Singleton
   - The Output and InfoHandler hierarchy tree implement singleton to smoother interactions with output streams and limit the number of objects that will query the database.
3) Factory
   - All of our equipment and usable item factory classes are not quite factories in the conventional sense. They are intended to create random items from a variety of possibilities. For the UsableItemFactory this very much resembles a simple factory, while others rely heavily on our database to retrieve their information and build their objects.
4) Abstract Factory
   - Abilities are created using an abstract factory with Concrete factories for each class and for hostiles.
5) Template
   - EquipmentFactory subclasses implement a template method to gather together the various stats, name, description, etc that are required to build a piece of equipment, partially using database calls. The equipment factories are thus not pure factories, but factory-template hybrids.
6) Null
   - We use a NullEquipment object to place on characters so that they have an objects that can be used for checks such as equip/unequip.
7) Prototype
   - Used when creating abilities to avoid querying the database very often.