Singleton AudioManager:

Due to the Singleton's ability to work throughout the lifetime of the game, it is good for preserving information regarding to the player's audio preferences. It could be used to save the preferred volume of the player ship's attack sound, the sound of enemies attacking, and the sound of both background music as well as menu elements.

Observer Player/Enemies:

Because the Observer allows multiple different game elements to read the status of one, on a subscription basis, you could use the observer to implement new attacks/abilities for the player which would allow them to, for example: On item pickup, destroy all enemies of a certain type. It would be most efficient to use an observer for the enemies so that instead of filtering through a list of all current enemies and destroying the ones that meet the specifications/requirements, you could instead use have the relevant enemies be subscribed to the corresponding observer that would affect them.

Factory Method Patterns:

You could use factory methods patterns to "improve" enemies during game runtime. By having multiple different sets of preset enemy AI, programmers could use factory method pattern to switch between the varying sets easily by having the different parts subdivided. This could be used to make the enemies smarter when the player has higher score, rather than just making them move/attack faster.