

Chips Challenge Design Patterns

1. **Pattern Name:** Observer
 - a. **Class Name:** ChipsChallengeMap
 - i. **Role in pattern:** Observer
 - b. **Class Name:** Chip
 - i. **Role in pattern:** Observable
 - c. **Purpose**
 - i. The map needs to observe Chip so that it can know if Chip's movement will make him collect a key/chip, unlock a door, or finish the level. The map needs to update itself accordingly if Chip's movements cause any of these.
2. **Pattern Name:** Singleton
 - a. **Class Name:** ChipsChallengeMap
 - i. **Role in pattern:** Singleton
 - b. **Purpose**
 - i. We don't want to have more than one map loaded at a time because we only ever need one: the one we are currently playing. Thus, we only have one instance to avoid unused maps.
3. **Pattern Name:** Strategy
 - a. **Use Case 1**
 - i. **Class Name:** Unlockable
 1. **Role in pattern:** Base abstract class
 - ii. **Class Name:** CollectibleChipBlocker
 1. **Role in pattern:** Subclass
 - iii. **Class Name:** KeyWall
 1. **Role in pattern:** Subclass
 - iv. **Purpose:**
 1. Chip can attempt to unlock an unlockable with his collectibles, but the Chip class doesn't really need to know what kind of unlockable he's trying to open and how the unlockable determines whether or not the collectibles can unlock the door. He just cares whether they do unlock or not. This unlocking occurs in Chip's tryUnlocking() method and the tryUnlocking() methods of the subclasses.
 - b. **Use Case 2**
 - i. **Class Name:** Collectible
 1. **Role in pattern:** Base abstract class
 - ii. **Class Name:** CollectibleChip
 1. **Role in pattern:** Subclass
 - iii. **Class Name:** Key

1. **Role in pattern:** Subclass

- iv. **Purpose:**

1. When Chip tries to add a collectible to his collection and display its acquisition message, there's no need to know what kind of collectible it is. The Chip class only cares that it's added and displays its message. Chip doesn't actually care what the message is or even if a message is actually displayed at all. This acquisition and message display occurs in Chip's `pickUp()` method and the subclasses' `showCollectedMessage()` methods.