**Command Design Pattern**

**Advantages**

1. **Encapsulates Actions**: Each action is encapsulated as a command object, which includes the logic for executing and undoing the action.
2. **Flexibility**: It is easy to extend the system with new commands without changing existing code.
3. **Complex Actions**: Suitable for actions that require multiple steps or complex logic to undo.

**Use Case**

* **Complex Operations**: When actions are complex and involve multiple objects or steps (e.g., a series of operations in a text editor).
* **Redo Functionality**: Supports both undo and redo operations by maintaining a history of commands.
* **Transactional Systems**: Useful in systems where operations need to be applied and rolled back as atomic transactions.

**Memento Design Pattern (Snapshot Design Pattern)**

**Advantages**

1. **State Capture**: Captures and restores the internal state of an object without exposing its internal structure.
2. **Simple Undo**: Suitable for simple state changes where saving and restoring state is straightforward.
3. **Encapsulation**: Keeps the details of the state encapsulated within the memento object.

**Use Case**

* **Simple State Management**: When the state of an object is simple and can be easily captured and restored (e.g., restoring the state of a game).
* **Frequent State Changes**: Suitable for applications with frequent state changes that need to be undone (e.g., drawing applications).
* **Single Object**: Works well when dealing with the state of a single object or a small number of objects.

**Conclusion**

* **Use the Command Design Pattern** if your application involves complex actions that need both execution and undo logic, or if you need to support redo functionality. It is more flexible and can handle complex scenarios involving multiple objects or steps.
* **Use the Memento Design Pattern** if your application involves simple state changes that need to be captured and restored. It is more straightforward and easier to implement for simple undo scenarios involving single objects.