Use the ***Memento Pattern*** when you need to be able to return an object to one of its previous states; for instance, if your user requests an “undo”.

***Scenario – Saving the game state in interactive game***

GameMemento

savedGameState

The Memento has two goals:

* Saving the important state of a system’s key object.
* Maintaining the key object’s encapsulation.

MasterGameObject

gameState

Object getCurrentState() {

// gather state

return(gameState);

}

restoreState(Object savedState) {

// restore state

}

// do other game stuff

Client

// when new level is reached

Object saved = (Object) mgo.getCurrentState();

// when a restore is required

mgo.restoreState(saved);

***Benefits:***

* Keeping the saved state external from the key object helps to maintain cohesion.
* Keeps the key object’s data encapsulated.
* Provides easy-to-implement recovery capability.

***Uses and Drawbacks:***

* The Memento is used to save state.
* A drawback to using Memento is that saving and restoring state can be time consuming.
* In Java systems, consider using Serialization to save a system’s state.