M3: Differentially Interacting Warriors

**Extension Document:**

In order to add a new Warrior class (called W hereon), we will have to first modify Agent, Soldier, Archer and Witch\_doctor classes. But the modification to each class is simple: first we add a **take\_hit(int, shared\_ptr<W>)** in Agent, and provide a default implementation. For example, we can implement by calling lose\_health(). Then in each of Soldier, Archer and Witch\_doctor class, we declare and define **take\_hit(int, shared\_ptr<W>)** if we want to override the default Agent behavior. After all these modifications, we add four take\_hit() functions in the W class: **take\_hit(int, shared\_ptr<Soldier>), take\_hit(int, shared\_ptr<Archer>), take\_hit(int, shared\_ptr<Witch\_doctor>) and take\_hit(int, shared\_ptr<W>).** Each one of these will implement the specific behavior when attacked by each kind of Warrior. Unless new information has to be acquired from Model, in which case Model has to be modified, no changes would be made to any of the other classes. Compared to other two methods for the Double Dispatch pattern, the change using this method is minimal and does not impose maintenance problems as serious as the other two methods.

Since each Warrior react depending on the type of the attacker, not specific attributes like strength, arbitrary interactions can be supported. And if we want Archer to have a reaction different from what we have now when attacked by a Soldier, we can simply re-implement the take\_hit(int, shared\_ptr<Soldier>) function in the Archer class without modifying any of the other classes (unless Model has to be queried for some new kind of information, in which case Model has to be modified).