S1: Design of a new Agent: Wizard Doctor

**Behavior Description:**

Witch\_doctor is a new type of Agent that possesses the ability to heal other agents, and who shows similarity with Warriors as he has capabilities to defend himself if under attack. The Witch\_doctor can be in exactly one of “Attacking” or “Healing” at a time.

Automatic Behavior:

A Witch\_doctor will first look around in his environment, and then choose the agent with below full health within his range to be his patient. A Witch\_doctor is afraid of being caught in the middle of a battle, and therefore he will not consider healing any agents whose current status is “**Attacking**”. However, Witch\_doctor will heal Non-Warriors (i.e. Peasant) being under attack due to benevolence in his character. If two agents share the same distance from the Witch\_doctor, he will break the tie using alphabetic order. Once he has determined his patient, he will add 1 to the health of his patient and change his status to “**Healing**”. While his state is “**Healing**”, the Wizard\_doctor will continue healing his patient as long as the latter is not at full health. Once his patient is cured, his state will be changed to “**Not Healing**” and finding new Patient.

A Witch\_doctor will have the ability to defend himself once under the attack by counterattacking his opponent using poison. If he was healing a patient while being attacked, he will change his status from “**Healing**” to “**Attacking**”. Witch\_doctor shows similarity with Soldier in that he will continue attacking until his opponent is dead, or the latter is out of his range. In contrary to Warriors, Witch\_doctor shows no aggressive behavior that he cannot be ordered or actively participate in a combat.

**Initial Values:**

The Witch\_doctor is initialized with healing\_strength of 1, attacking\_strength of 1, and the range of 3, and outputs the message “**I’m Healing!**” when he cures his target, and “**Take poison!**” when counterattacking his opponent. Unless specified below, all behaviors should follow default Agent behavior.

**start\_attacking**: Output **“I don’t want to attack!**” and stays in his current state.

**update**: First, update the agent state, then follow the order below:

1. If not “**Alive**”, do nothing further.

2. If “**Attacking**”, follow the same behavior as Soldier.

3. If **Not healing**, find a new patient using logic described above. If found a patient, output “**Found a new patient!**” with the patient’s name. Otherwise, do nothing further. If no patient satisfying the conditions, then output nothing and stay in **Not Healing**.

4. If “**Healing**”, follow the order below:

1. Check if the patient is still “**Alive**”, if not, output “**Patient is Dead**” and change state to “**Not Healing**”, discard target pointer with no further action.

2. Check if the patient in in range, if not, output “**Patient out of range**” and change state to “**Not Healing**” discard target pointer with no further action.

3. Check if the patient is fully health, if so, output “**Patient is now perfectly healthy!”** and change his state to “**Not Healing**”.

4. Output the message “**I’m Healing!**”, and calls the target’s gain\_health() function with Witch\_doctor’s healing\_strength.

**start\_healing**:

Witch\_doctor is allowed to heal himself as long as he is not counter-attacking his aggressor. If so, error message of “**I'm busy counter-attacking!**” is output.

If the Target is “**Not Alive**”, throws error message that **“I can’t save dead patients!**”

If the distance between Witch\_doctor and the target is greater than range, throw an error that the “**patient is out of range**”.

If Witch\_doctor is asked to heal a patient whose state is “**Attacking**”, error message of **“I don't want to caught myself between Warriors!**” is output.

If the supplied patient has full health, then output error message “**This patient is perfectly healthy!**”

If all is OK, the pointer to patient is saved and set his state to “**Healing**”.

**take\_hit**: First, compute new health using Agent::lose\_health() function, then setting state to “**Not Healing**“ and “**Attacking**“. Then, stores the target\_ptr and output “**I’m counter-attacking!**” message.

**describe**: Output “Witch\_doctor” followed by his description, then followed by “**Attacking**” and “**Healing**” state information.

**Design Description:**

The Witch\_doctor class shares considerable similarity with Warrior class in such it has the ability to counter-attack once being attacked. Therefore, the Witch\_doctor class inherits publicly from Warrior base class and calls Warrior::update and Warrior::describe in support of his “**Attacking**” behaviors. In addition, one can treat Witch\_doctor as a “**Healing**” Warrior, and such feature is easily supported by adding a Boolean variable “**bool healing**” to support the addition of a such new state. Our Warrior class is fitted with a “**print\_attack\_word()**” function that is overridden by Witch\_doctor class for outputting his proper message while under attack without the need to rewrite Warrior::update function for such purpose.

Some new functionality has been added to the Agent base class. The addition of **Agent::gain\_health(int curing\_strength)** provides the ability for all agents to be able to be cured. Moreover, in order to determine if the Agent is in “**Attacking**” state, **bool is\_attacking()** function has been added as a “Fat Interface” approach whose default return value is false, and has been overridden in Warrior hierarchy to return the proper value. Such approach is adopted to avoid the need of exposing Warrior’s internal implementation. Finally, in order for Witch\_doctor to know the current health of the Agents, the function “**int get\_health()**” has been added in Agent base class. In Agent\_factory class, the changes are minimal to support the addition of the new Agent.

Few modifications has been done in Model and Controller. First, in order to get the Agent pointer who is within the Witch\_doctor’s range, whose state is “**Not Attacking**”, and whose health is weakest, function “**get\_weakest\_agent\_in\_range**” has been added with considerable reuse of functionality provided in “**get\_closest\_agent\_ptr**”. In Controller class, the addition of a new command “**heal**” is done with little effort for giving user command over the new Witch\_doctor Agent type.