Battles – Actions

Battle systems are challenging because they effect all of the other commands in the game. If you are in the middle of a battle with a monster, for example, you can't really take time to examine a book. Issuing an *examine book* command in the middle of a battle should respond differently than during other times in the game. In addition, you need to have something to battle. This could be any number of things including an NPC.

1. For most of you, the things you are battling fall under the category of Monster (goblins, tiger, bears, oh my!). A Monster class consists of the following fields:
   (a) a *name* so that you can tell the player what they are fighting
   (b) a *health* field so that you know how much damage the monster can take before he dies
   (c) a *damage* field so that you know how much damage the monster does when he hits you. It makes sense for this variable to vary in which case this would store the maximum damage.
   (d) a *hitProbabilty* so that you know how likely it is that the monster will hit you when you are battling.
2. If your player will eventually battle an NPC, your NPC class should extend your Monster class.
3. Your player class needs to have some additional features added to it before he can enter a battle:
   (a) Add a damage field to so that you know how much damage the player can inflict. It makes sense for this variable to vary in which case this would store the maximum damage. The accessor method for this field would also take into account any weapons the player is using
   (b) Add a hitProbability so you know how likely it is that the player hits his target when battling
4. Implementing a battle
   (a) Add a **boolean** variable (e.g. inBattle) into your game class to indicate whether or not the player is currently engaged in a battle. This variable is set to true whenever a monster is encountered and is set to false whenever the monster is killed
   (b) Review the action of every command in processCommand and decide what it's behavior should be if the player is currently in battle. Can you examine a book while a monster is attacking you? What should the behavior be?
      i. Break up the switch statement in processCommand into two, those that are available when inBattle is true and those that are available when inBattle is false.
      ii. Use the **default** case of the switch statement to inform the player that they have tried to use a command that is not available at the given time.
   (c) Create a Random variable in your game class to calculate hit probabilities and damage
   (d) Add appropriate commands for conducting your battle (e.g., swing, hit, poke, etc) and have them do the right thing (see below).
   (e) Use the following algorithm for each battle round in the main game loop:

```
Algorithm battle()
begin
    playerDied = false
    hit = random integer
    if (hit < hitProbability of the player) then
        Calculate how much damage player inflicts
        Subtract damage from monster's health
        if ( monster is dead ) then
            battle is over and inBattle = false
        else
            hit = random integer
            if ( hit < hitProbability of the monster) then
                Calculate how much damage monster inflicts
                Subtract damage from player's health
                if ( player is dead ) then
                    playerDied = true
                    returning true to wantToQuit in processCommand ends the game
                end
            end
        end
    end
    return playerDied
end
```

5. Be sure to test your project thoroughly and check your changes into Github.