Department of Computer Science & CINSAM

NKU Summer Programming Workshop 2015

Project 4: Fight Game

We will implement a fight game today where you are facing off against some other fighter or monster. You have a sword which you can swing the sword 4 times in a turn. If you hit the other fighter, the sword can do between 1 and 8 points of damage to it. After you attack the monster, the other fighter gets a turn to attack you. The fighter (monster) uses its hands and tries to hit you twice per turn, doing between 1 and 12 points of damage. You fight until one of you dies. Write the program as follows:

- You start with 50 hit points and 10 points of protection, and the other fighter starts with a random amount of hit points between 30 and 40 and a random protection number between 12 and 15
- Repeat until one of you dies (has hit points of 0 or less) using a while loop
  - Output the turn number (e.g., round 7)
  - Your turn: repeat 4 times (use a for loop)
    - Randomly generate a number between 1 and 20, if the number is greater than or equal to the other fighter's protection, it's a hit, otherwise it's a miss
      - If you score a hit, randomly generate the damage (between 1 and 8) and deduct this from the other fighter's hit points
    - Output the result of the attack (hit or miss, and if hit, how much damage and the other fighter's remaining hit points
  - If the other fighter is still alive, it takes a turn: repeat 2 times (use a for loop)
    - Randomly generate a number between 1 and 20, if the number is greater than or equal to your protection, it's a hit, otherwise it's a miss
      - If a hit, generate damage (1-12) and deduct from your hit points
    - Output the result of the attack
  - Add 1 to round
- Output who won and the winner's ending hit points

Note that there is no user input in this program. So it will run quickly and all the output will appear too fast to read. Before or after adding 1 to the round, you can insert these two instructions to pause the game:

```
System.out.print("Press enter for the next round");
in.nextLine();    // in is a Scanner that you will have to declare and instantiate
```

Start your program from the fightgame.java skeleton on the website. Variables that you will need for the game are described in the comments. Once you have the basic game working, here are some modifications. You may have to add variables to handle these enhancements.

- Place a loop around the entire game to repeat until either the user dies or the user wants to quit. If at the end of the current match, the user is still alive, ask the user if he/she wants to fight again. If yes, restore the user's hit points by a random amount (add 11-30 points back) and then have the user face off against another fighter with a random number of hit and protection. Use a do-while loop for this outer loop.
- Before each fight begins, ask the user which type of skill he/she wants to have. The skills alter the number of attacks, damage and protection values. Use the following four choices (or make up your own)
  - Strength: swings only 1 time but can do damage between 11 and 24, protection of 13
  - Swordsman: swings 6 times, damage 1-6, protection of 12
  - Speed: swings 8 times, damage 1-4, protection of 8
  - Premonition: swings 2 times, damage 1-10, foresee attacks coming for protection of 16
- After each fight, if our fighter is still alive, there is a 50% chance that the last fighter left behind a magic potion. The potion will give you 10 extra hit points for the next fight.
- Make up other variations as you like!