



Department of Computer Science & CINSAM

Summer Programming Workshop 2015



Project 11: Fighting Game part II

You will re-implement the fighting game from week 1 as an object-oriented game. You will write two classes, the Fighter class representing a fighter and the FightGame itself. The FightGame will be similar to the original but will now declare two or more Fighter objects and pass them messages. The Fighter class will consist of these data variables: String name; int numAttacks, damagePerAttack, defense, hitPoints, potions, treasure; Random g; Scanner in. The Fighter class will need methods as follows:

1. a constructor (to assign name, defense, numAttacks, hitPoints, damagePerAttack, potions, the Random generator and the Scanner – this will be used in enhancements)
2. accessors for getDefense(), getName(), getPotions(), getTreasure()
3. an isAttacked(int damage) method which will subtract the damage from this Fighter's hit points (for instance, if f1.isAttacked(damage) then f1.hitPoints=hitPoints-damage)
4. an attack method which is passed another Fighter object (public void attack(Fighter f2) and for each of this Fighter's numAttacks, it will roll a 20-sided die and see if it successfully hit the other Fighter using f2.getDefense(). If the Fighter successfully hits the other Fighter, then generate a random amount of damage between 1 and damagePerAttack and then call f2.isAttacked(damage).
5. an isAlive method that returns true if this Fighter's hitPoints > 0, false otherwise.
6. a toString method to output the name of the Fighter and his/her current hit points, or if the Fighter is dead.

Implement a FightGame class which creates two Fighters. You can randomly assign values for numAttacks, damagePerAttack, etc, or assign them like we did with HorseRace. Now the two Fighters fight each other until one is dead.

Enhancements:

1. add a loop so that after one fight, another one starts until either this Fighter dies or there are until this Fighter survives some preset number of fights (say 8), output the results (how many fights this Fighter survived and whether the Fighter is still alive)
2. at the end of a fight, if this Fighter wins the fight, the Fighter takes the other Fighter's treasure, at the end of the game, output this Fighter's total treasure
3. after successfully winning a fight, the Fighter can choose to take a potion. This requires implement a takePotions method to check to see if the other Fighter has potions and if so, add the number that f2 has to your potions.
4. between fights, give the user a chance to input one of several choices:
 - drink potion – if the Fighter has potions, decrement this and add to the Fighter a 1 point of protection (90%) or heals the Fighter by 10-20 points (5%) or kills him (5%)
 - rest – heals the Fighter between 11-20 hit points but there is a 1 in 3 chance of being interrupted by a new Fighter who then gets one free turn before this Fighter begins to fight
 - practice – the Fighter has a 1 in 2 chance of not being interrupted and if successful, adds 2 points to his damagePerAttack but if interrupted, the other Fighter gets a turn before this Fighter begins to fight
 - quit – exit the game
5. create a graphical display of the game by outputting in a Graphics panel the fight information there (the Fighter's names, who is attacking whom, which attack he/she is on, and a rectangle indicating their hitPoints left (sort of like the amount of shields left), this rectangle/bar reduces in size as the fight continues