

Lab 6 Improve function

```
@Override
public void Attack(RPGCharacter playerAT, RPGCharacter playerDF) {
    if (playerAT.wrist != null) {
        dmg = playerAT.damage + wrist.AtkBonus() - playerDF.defense;
    } else {
        dmg = playerAT.damage - playerDF.defense;
    }
    if (dmg <= 0) {
        dmg = 0;
    }
    playerDF.CurrentHp -= dmg;
    if (playerDF.CurrentHp < 0.5) playerDF.CurrentHp = 0;

    if (dmg == 0) {
        System.out.println("ATK : " + String.format("%.1f", dmg) + " dmg" + " [ "
            + playerDF.name + " ]" + " Defense : " + playerDF.defense);
    } else if (playerAT.wrist == null) {
        System.out.println("ATK : " + String.format("%.1f", dmg) + " dmg");
        System.out.println("[ " + playerDF.name + " ]" + " HP : " + String.format("%.1f", playerDF.CurrentHp));
    } else {
        System.out.println("ATK + ATK bonus : " + String.format("%.1f", dmg) + " [ "
            + playerDF.name + " ]" + " HP : " + String.format("%.1f", playerDF.CurrentHp));
    }
}
```

ฟังก์ชัน Attack คิดว่าสามารถพัฒนาได้
โดยจะมีทริค mana และ speed ของ
player มาคิด

เช่น player ที่มี speed มากกว่าสามารถเลือก
โจมตีได้ก่อน หรือสามารถใช้ mana เพื่อโจมตีได้
ทุกรอบขึ้น

```
private void Battle1_2(int x, RPGCharacter p1, RPGCharacter p2){
    if (x < 1 || x > 3) {
        System.out.println("Error!!! " + "[ " + p1.name + " ]" + " you enter wrong choice");
        SelectBattle1(p1, p2);
    }
    else if (x == 1){
        System.out.println("[ " + p1.name + " ] --> " + attacks + " --> [ " + p2.name + " ]"
            + " Attack(p1,p2);
        if (p2.CurrentHp <= 0){
            System.out.println("-----");
            System.out.println(" [ " + p1.name + " ] " + " is the winner!!!");
            System.out.println("-----");
        } else {
            SelectBattle2(p1, p2);
        }
    }
    else if (x == 2){
        System.out.println("[ " + p1.name + " ] --> " + heal ++ " ");
        p1.Heal();
        SelectBattle2(p1, p2);
    }
    else {
        System.out.println("[ " + p1.name + " ] don't do anything this turn" + "\n");
        SelectBattle2(p1, p2);
    }
}
```

```
private void Battle2_1(int x, RPGCharacter p1, RPGCharacter p2){
    if (x < 1 || x > 3) {
        System.out.println("Error!!! " + "[ " + p2.name + " ]" + " you enter wrong choice");
        SelectBattle1(p1, p2);
    }
    else if (x == 1){
        System.out.println("[ " + p2.name + " ] --> " + attacks + " --> [ " + p1.name + " ]"
            + " Attack(p2,p1);
        if (p1.CurrentHp <= 0){
            System.out.println("-----");
            System.out.println(" [ " + p2.name + " ] " + " is the winner!!!");
            System.out.println("-----");
        } else {
            SelectBattle1(p1, p2);
        }
    }
    else if (x == 2){
        System.out.println("[ " + p2.name + " ] --> " + Heal ++ " ");
        p2.Heal();
        SelectBattle1(p1, p2);
    }
    else {
        System.out.println("[ " + p2.name + " ] don't do anything this turn" + "\n");
        SelectBattle1(p1, p2);
    }
}
```

คิดว่าฟังก์ชัน Battle1_2 และ Battle 2_1
สามารถรวมเป็นฟังก์ชันเดียวกันได้
เพราะสองฟังก์ชันนี้ทำงานเหมือนกัน
แต่ player ไขนก่อนเลยคิดว่าสามารถรวม
ให้เป็นฟังก์ชันเดียวกันได้เลย

```
private void SelectBattle1(RPGCharacter p1, RPGCharacter p2){
    System.out.println("-----");
    System.out.println("[ " + p1.name + " ]" + " Choose your options ");
    System.out.println("[1] Attack [2] Heal [3] Don't do anything this turn ");
    Scanner bt = new Scanner(System.in);
    System.out.print("Select your options :");
    int scBattle = bt.nextInt();
    Battle1_2(scBattle, p1, p2);
}

//
+ Let player 2 choose action ( attack / heal / do nothing )
+ pparam p1 is player 1
+ pparam p2 is player 2
+ effect : call Battle2_1 function
//

private void SelectBattle2(RPGCharacter p1, RPGCharacter p2){
    System.out.println("-----");
    System.out.println("[ " + p2.name + " ]" + " Choose your options ");
    System.out.println("[1] Attack [2] Heal [3] Don't do anything this turn ");
    Scanner bt = new Scanner(System.in);
    System.out.print("Select your options :");
    int scBattle = bt.nextInt();
    Battle2_1(scBattle, p1, p2);
}
```

} สามารถ
รวมเป็น
อันเดียวได้
เหมือนกัน

```

public static void AddAccessories(Ring r, Clothing c, Wrist w) {
    AllRing[Rsize] = r;
    Rsize++;
    AllClothing[Csize] = c;
    Csize++;
    AllWrist[Wsize] = w;
    Wsize++;
    if (Rsize == 5 || Wsize == 5 || Csize == 5) {
        System.out.println("All accessories full");
    }
}

/**
 * Add sword and shield to each array ( sword , shield )
 * @param sw is sword that get from the client to push into sword array
 * @param sh is shield that get from the client to push into shield array
 * effect : ( r , c , w ) were pushed into sword array and shield array
 *          size of each array +1
 */
public static void AddSwordandShield(Sword sw, Shield sh) {
    sword[Ssize] = sw;
    Ssize++;
    shield[SHsize] = sh;
    SHsize++;
}

```

ข้อฟังก์ชันนี้ คิดว่า สามารถทำแบบ dynamic Array
ได้ โดยตัวรับของเข้ามาจนเต็ม ก็จะขยายขนาด Array
2 เท่า และรับ ของเพิ่มได้

และ คิดว่า ฟังก์ชันนี้ หกแน่นอนจากนี้ไปแล้ว แล ยังคิดวิธีพัฒนา ยังไม่ออก