



Lab 6

Review Code

#1 Avoid magic number

เราทำได้ค่อนข้างดี เพราะเคยที่เราเขียนส่วนใหญ่จะมีตัวแปรกำกับ ยกเว้น ค่าบางค่าที่เราตั้งขึ้นมาเป็น

ค่าเริ่มต้น ก็เราตั้งไว้เป็น ค่า fix

จุดที่คิดว่าสามารถแก้ไขได้

สามารถทำให้มันดีขึ้นได้โดยสร้างตัวแปรขึ้นมารับค่าบันทึกไว้ทำให้อ่านง่าย และ แก้ไขได้ง่ายในอนาคต

```
public double calDmg(Character1 Attacker) {  
    double dmg ;  
    if(equippedSword != null){  
        dmg = Attacker.equippedSword.getDamage();  
    }else{  
        dmg = 20 ;   
        // ตรงนี้เป็นที่สร้างเอง  
    }  
    if (equippedShield != null){  
        if (dmg - this.equippedShield.getDefense() > 0){  
            dmg -= this.equippedShield.getDefense();  
            return dmg ;  
        }else {  
            return 0;  
        }  
    }  
}
```

```

        }else {
            if(dmg >= hp){
                return hp ;
            }else{
                return dmg;
            }
        }
    }
}

```

#2 Don't repeat yourself (DRY)

โดยรวมคิดว่าค่อนข้างกลาง ๆ ในส่วนของตัว code ไม่ได้มีการใช้การ repeat variable ที่มีความหมายเดียวกัน neh

แต่ในส่วนของ methods ต่างๆที่ใช้ร่วมกันได้ทำการเก็บไว้ใน Class นั่งแล้วในส่วนนี้คิดว่าทำได้ดี

#3 Comment where needed

ในส่วนนี้ยังทำได้ไม่ดี เพราะว่า Code ในตอนแรก ไม่มีการ comment ใดๆเลยในส่วนนี้เพิ่มคิดว่า yังทำได้ไม่ดีมากนัก ยังสามารถที่จะทำได้ดีมากกว่านี้โดยการ comment เพิ่มเข้าไปให้อ่านได้ง่ายขึ้นและเข้าใจได้มากขึ้น

#4 Fail fast

ในส่วนของกฎข้อนี้คิดว่า yังไม่มีปัญหา เพราะในโปรเจคยังไม่ได้มีการใช้ค่าที่สามารถตั้งเป็น enum ได้

#5 One purpose for each variable

คิดว่าสามารถทำตามกฎได้ดี เพราะ each variable ใน code แบ่งแยกหน้าที่กันและการนำไปใช้อย่างชัดเจน

#6 Use good names

โดยรวมมองว่าค่อนข้างเลือกตั้งชื่อตัวแปรได้ดี เพราะสามารถอ่านชื่อตัวแปรแล้วรู้ได้เลยว่าหมายถึงอะไร

จุดที่คิดว่าสามารถแก้ไขได้

โดยจุดที่สามารถแก้ไขให้ดีขึ้นได้มองว่าการตั้งชื่อ function ในการทำงานควรจะตั้งชื่อเฉพาะให้สามารถอ่านแล้วเข้าใจได้เลย e.g., equipRing(), equipNecklace(), unEquipRing(), unEquipNecklace()

```
public void EquipAccessories(Ring ring){
    this.ring = ring ;
    ring.giveEffect(this);
}
public void UnEquipAccessories(Ring ring){
    this.ring = null ;
    ring.DeEffect(this);
}
public void EquipAccessories(Necklace necklace){
    this.necklace = necklace ;
    necklace.giveEffect(this);
}
public void UnEquipAccessories(Necklace necklace){
    this.necklace = null ;
    necklace.DeEffect(this);
}
```