

Présentation BONUS PROJET 3, LAURENT JUNDT

* Ajout d'un bonus qui permet de définir un type par personnage qui servira au calcul de l'attaque.

Les type sont les suivants :

- Magus de type healer,
- Combattant de type human
- Dwarf de type elf
- Colosse de type rock

* Ajout d'une probabilité de coup critique par attaque ou soin par personnage, si un coup est un critique la puissance d'attaque de l'arme du personnage est doublée.

Lorsque une attaque se produit entre 2 personnages, les 2 types sont comparés entre eux afin d'attribuer un coefficient d'attaque qui sera multiplié par la puissance d'attaque de l'arme et du coup critique éventuel.

Ce qui peut arriver pendant le cours du jeu est qu'un personnage découvre l'arme mystique avec 50 points de dégâts, si un coup critique est porté en même temps avec un facteur supérieur à 1 le personnage recevant le coup peut mourir en 1 coup. Cela apporte une part de chance si un joueur se fait mener il peut quand même remporter le match.

Partie technique et intégration du code

Intégration du coup critique et des types de chaque personnage :

```
class Character {  
    let name: String // name of the characters  
    var weapon: Weapon //character's weapon  
    var life: Decimal //life of character details in class girl  
    let maxLife: Decimal //constant to block over healing and avoid to pass under 0  
    var crit: Int //bonus var to multiply the hit by 2  
    let type: String //character type or race  
    let classType: ClassType //names of class girl  
  
    init(name: String, weapon: Weapon, crit: Int, type: String, maxLife: Decimal, classType: ClassType) {  
        self.name = name  
        self.weapon = weapon  
        self.crit = crit  
        self.type = type  
  
        self.maxLife = maxLife  
        self.life = maxLife  
        self.classType = classType  
    }  
}
```

Définition dans chaque class personnage le % de critique lui appartenant. Variable crit et constante type initialisées dans la class maitre Character

Les autres personnages héritent de la classe Character

```
class Combattant: Character {  
    init(name: String) {  
        super.init(name: name, weapon: Sword(), crit: 15, type: "human", maxLife: 100, classType: .figther)  
    }  
}
```

Ajout d'une constante random pour la partie critique dans la fonction attack() et healing() appartenant à la class Character en utilisant la valeur définie et différente pour chaque personnage. Le critique est indiqué au joueur par un print() et la variable critFactor est doublée.

```
if numberCrit == 1{  
    critFactor = 2  
    print("your attack has crit !!!")  
}
```

Définition des facteurs d'attaque en fonction du type qui attaque et du type qui reçoit l'attaque dans la fonction `factorType()` qui retourne une valeur décimale.

```
func factorType(attackFactorParameter: Decimal) -> Decimal{
    var factor: Decimal = 1
    factor = factor * attackFactorParameter
    print("attack factor : \(factor)")
    return factor
}
```

Le paramètre de la fonction est entré lors des choix du type à l'aide d'un if suite à la comparaison des 2 types. Toujours dans la class `Character` fonction `compareType()`, un facteur d'attaque sera retourné après la comparaison des 2 types. Cette fonction est appelée dans la fonction d'attaque principale.

```
func compareType (characterToAttack: Character)->Decimal{
    var attackFactor: Decimal = 1.00
    print("Type \(type) vs type \(characterToAttack.type)")

    if type == "human" {
        if characterToAttack.type == "elf"{
            attackFactor = factorType(attackFactorParameter: 1.5)
        }
        else{
            if characterToAttack.type == "rock"{
                attackFactor = factorType(attackFactorParameter: 0.25)
            }
        }
    }
    else{
        if type == "rock"{
            if characterToAttack.type == "human"{
                attackFactor = factorType(attackFactorParameter: 1.5)
            }
            else{
                if characterToAttack.type == "elf"{
                    attackFactor = factorType(attackFactorParameter: 1.8)
                }
            }
        }
        else{
            if type == "elf"{
                if characterToAttack.type == "rock"{
                    attackFactor = factorType(attackFactorParameter: 0.5)
                }
                else{
                    if characterToAttack.type == "human"{
                        attackFactor = factorType(attackFactorParameter: 0.75)
                    }
                }
            }
        }
    }
    return attackFactor
}
```