

Descriptif du Bonus

Le concept

Le bonus choisi est celui de l'auto-Destruction.

Ce bonus se déclenchera à la suite de l'attaque lancée, dans le cas où l' "enemySelected" n'a plus que 4 points de vie ou moins, le personnage lancera automatiquement l'attaque "self destruction".

Par cette attaque, le personnage s'auto-détruit, et ainsi ôtera des points de vie à tous les membres de l'équipe adverse (1/2 des points de vie actuels)

Ce bonus servira à renverser la partie.

Toutefois le nombre de point de vie à avoir pour "bénéficier" du bonus étant faible, il sera difficile à le déclencher par la stratégie, relèvera plus de la chance (ou de la mal chance).

L'integration

Dans le dossier Game+Bonus.swift, j'y ai créé une extension de la classe Game, avec une fonction selfDestruction() avec comme paramètres, une victime de type Character et une équipe attaquante de type Team.

Lorsque la fonction est appelée, on initialise la valeur de points de vie de la victime à 0, afin qu'elle soit considérée comme décédée, puis on divise par deux le nombre de points de vie pour tous les personnages de l'équipe attaquante

```
11 extension Game {
12     // increment the the victim life point to 0, and remove half of the life point from all characters
    of the opposite team
13     func selfDestructionBonus(victim: Character, attackerTeam: Team) {
14         victim.life = 0
15         for character in attackerTeam.characters {
16             character.life /= 2
17         }
18         print("""
19
20             !!! Due to his low level of life points, \(victim.name) the \(victim.characterType)
                \(victim.characterType.getEmoji()) choose by himself, to use self Destuct, by this
                action, he fainted 🏴‍☠️ but he also inflict a loose of half to the life points of all the
                characters in the Team \(attackerTeam.name) !!!
21
22         """)
23
24     }
```

Cette fonction est appelée dans le fichier Game.swift à la ligne 122, elle est intégrée à la fonction battleTurn(), lorsque l'ennemie a été sélectionné et attaqué.

Dans un premier temps une fois l'attaque faite, une première vérification est réalisée afin de savoir si l'enemySelected est toujours vivant ou non, puis si l'enemySelected est toujours en vie, une seconde vérification est faite, à savoir si le nombre de points de vie est égal ou inférieur à 4, dans le cas où la condition est remplie, alors la fonction selfDestruction() est appelée.

```
114         if enemySelected.life > 0 {
115             print("""
116
117                 \(characterSelected.name) \(characterSelected.characterType.getEmoji()) attacked
118                 \(enemySelected.name) \(enemySelected.characterType.getEmoji())
119                 \(enemySelected.name) \(enemySelected.characterType.getEmoji()) has now
120                 \(enemySelected.life) life points ♥
121             """)
122             // use to verify if enemySelected.life <= 4 using the selfDestructionBonus()
123             if enemySelected.life <= 4 {
124                 selfDestructionBonus(victim: enemySelected, attackerTeam: team)
125             }
126         }
```

Recommandation

Pour tester avec facilité le bonus, je peux vous conseiller de commenter la ligne 95 du fichier Game.swift, ce qui n'appellera pas la fonction magicBoxAppearance().

Puis lancez le jeu et constituez la première équipe avec un Warior, un Colossus, et un Midget, et la deuxième à votre guise avec toutefois un Midget dans l'équipe.

Choisissez le Warior pour attaquer le Midget, puis dans le second tour, le Midget pour attaquer le Midget et enfin dans le dernier tour, le Colossus pour attaque le Midget.

De là, le message s'affichera.

Lien GitHub

ClaireRimel:

<https://github.com/ClaireRimel/Projet-3-OC-Game>