

Enoncé :

Programmer un combat à mort entre un chevalier et un dragon
Le combat continue **jusqu'à** la mort d'un des 2 protagonistes

A chaque **round** de jeu, déterminez aléatoirement la vitesse du dragon et du chevalier

Celui qui a la plus grande vitesse attaque, (suivant l'arme choisie, l'armure choisie et la difficulté)

Armes

Le joueur peut choisir 1 arme parmi 3 armes au début du combat, ou alors l'arme est choisie aléatoirement (à vous de voir, soit prompt, soit random)

Les armes appliquent un **ratio** sur les dégâts de base du joueur (les dégâts du joueur augmentent suivant le ratio)

Armure

Le joueur peut choisir 1 armure parmi 3 armures au début du combat, ou alors l'armure est choisie aléatoirement (à vous de voir, soit prompt, soit random)

Les armures appliquent un **ratio** sur les dégâts de base du dragon (cette fois ci, réduit les dégâts de base du dragon)

Difficulté

La **difficulté** du jeu est choisie par le joueur, entre facile, normal ou difficile.

La difficulté augmente les dégâts de base du dragon et réduit les dégâts de base du joueur

Données de base :

Dégâts de base du joueur et du dragon : 10

Ratio arme : 1,2,3

Ratio armure : 1,2,3

Ratio difficulty : 1,2,3

!!\ NOTEZ CE QUE VOUS FAITES EN CONSOLE.LOG POUR AVOIR UN SUIVI CLAIR DE CE QUE VOUS CODEZ

Etape 1 - LA BASTON :

- Déclarer 2 variables : dragon & chevalier
- Déclarer 1 fonction combat()
- Dans cette fonction de combat(), le dragon et le joueur doivent jeter un dé (math.random())
- Le plus grand Math.random() attaque son adversaire

Etape 2 - AJOUTONS DE LA SUBSTANCE

- Donner 150 points de vie au héros
- Donner 125 points de vie au dragon
- Lorsque le dragon ou le chevalier se fait attaquer, retirer 10 points de vie.

Etape 3 - LOGIQUE DU JEU

- Faire une boucle qui appelle la fonction combat() **TANT QUE** le dragon **OU** le chevalier est encore en vie.
- Compter le nombre de rounds de combat que le jeu effectue et les afficher dans la console

Etape 4 - RENDU DU VAINQUEUR

- Lorsque la boucle est terminée, afficher une image en HTML du vainqueur (chevalier ou dragon, images non fournies et vendues séparément :p)

Etape 5 - SURCOUCHE ET EQUILIBRAGE

- Demander au joueur le niveau de difficulté voulu entre 1) facile 2) moyen 3) difficile
- La difficulté augmente les dégâts de base du dragon (10) et réduit les dégâts de base du joueur (10)
- Demander au joueur de choisir une arme ou lui assigner en fonction de la diff choisie (a vous de voir, je préfère le 2e personnellement :))
- Les armes appliquent un **ratio (*1.25)** sur les dégâts de base du joueur (les dégâts du joueur augmentent suivant le ratio)
- Demander au joueur de choisir une armure ou lui assigner en fonction de la diff choisie
- Les armures appliquent un ratio (***0.75**) sur les dégâts de base du dragon