

RPG Battle

On souhaite coder une interface de combats de RPG en HTML/CSS.

Ce travail sera évalué. Vos dépôts github seront vérifiés le dimanche 13/12/20 à 18h.

Vérifiez à cette fin que votre dépôt est bien public, et que je dispose de l'information de l'url de votre compte.

Consignes

Vous ne produirez pas d'assets par vous-mêmes ; à défaut, ils ne seront pas évalués.

Vous pourrez récupérer un pack d'assets libres de droit et d'utilisation sur [opengameart](https://opengameart.org/) ou sur tout autre site qui vous précise explicitement que vous avez le droit de les utiliser.

Vous n'utiliserez pas d'assets propriétaires qui ne vous appartiennent pas.

Cahier des charges (2 points par fonctionnalité)

1/ On ne voit pas les aventurier-e-s, mais on voit les monstres affrontés. On voit les caractéristiques des personnages (ainsi que leurs noms), mais pas celles des monstre-s

2/ Quand on passe le curseur sur un monstre, on voit son nom et ses points de vie actuels apparaître en info-bulle.

3/ On peut valider une action de chaque personnage parmi plusieurs proposées ; ces actions sont attaquer, se défendre, et une spéciale à chaque personnage (soin, poison...). Valider une action l'exécute immédiatement.

4/ Il y a quatre personnages.

L'attaque d'un monstre cible aléatoirement un personnage, et est réduite par la défense de ce dernier. Le soin soigne automatiquement le personnage possédant le moins de pv.

5/ Il y a trois monstres. Les actions offensives des personnages doivent déterminer quel monstre elles ciblent.

6/ Quand tous les personnages ont validé une action au cours de ce tour, les monstres attaquent, un par un. Les monstres ne peuvent faire qu'attaquer.

7/ Valider une action la rend indisponible. Quand un nouveau tour commence, cela libère les actions de tous les personnages, sauf celle qui a été utilisée par chacun d'entre eux au tour précédent.

TP HTML/CSS/JS

8/ Les actions spéciales coûtent de la mana et ne peuvent être utilisées que si le personnage en possède suffisamment (sinon l'option n'est même pas cliquable).

9/ Un monstre possède un certain nombre de points de vie, et à 0, il meurt (ce qui est visible, au moins par sa disparition). Il ne pourra évidemment plus attaquer. Quand tous les monstres meurent, on gagne la partie.

10 / Un personnage qui descend à 0 points de vie (ou moins) meurt, et on ne peut plus lui sélectionner d'actions ; si tous les personnages meurent, la partie est perdue.

Conseils

Commencez de façon modeste ! Les fonctionnalités ne doivent surtout pas être réalisées dans l'ordre. Faites d'abord un face-à-face entre un seul personnage et un monstre, et ne permettez que l'attaque. Vérifiez, dès ce moment, vos conditions de victoire et de défaite. N'ajoutez qu'ensuite les différentes actions. C'est à la fin que vous ajouterez les nouveaux monstres et différents personnages. Ces fonctionnalités ne comptent chacune que pour un dixième de la note !

Testez très régulièrement. Vous pouvez vous servir ponctuellement d'un outil en ligne ou de la console javascript pour faire vos tests, mais rien ne vaut ensuite l'implémentation au sein de vos fichiers de travail.

Travaillez avec des fichiers .html, .css et .js se situant dans le même répertoire, et synchronisez régulièrement votre travail avec git afin de ne pas risquer de le perdre et d'avoir un suivi de vos modifications.

Testez toujours très régulièrement, et particulièrement que votre fichier html inclue bien vos fichiers css et js.

Conservez vos assets dans un répertoire /img/ que vous n'oublierez pas d'inclure dans votre repository.

Testez vos fichiers avant la date et heure de rendu. Assurez-vous que votre travail est lisible et à jour. Tant que vous n'avez pas de certitudes, testez.