

Jeu Dont Vous Êtes Le Héros (ou l'Héroïne)

On souhaite coder un jeu d'exploration en langage C. Il sera possible de voyager de chapitre en chapitre, de collecter ou perdre des pièces d'or ou des points de vie.

Consignes

Pour entamer le développement de chacune des fonctionnalités, il faut tirer une nouvelle branche ; sauf indication contraire, la fusion est réalisée à la fin de chaque exercice.

Avant de commencer, il convient :

- de supprimer le contenu du répertoire C:\Users\game\Documents\GitHub, et de déconnecter l'utilisateur précédent sur GitHub Desktop si nécessaire ;
- de créer un dépôt **public** et d'en envoyer l'URL à : selene.t.etpa@gmail.com

Exercice 1 - Choix par numéro

1. Tirer une branche "deplacement" pour ne pas développer directement sur le *master*.
2. Créer le fichier jdvelh.c dans le répertoire de travail local. Il contiendra l'ensemble du code de notre projet de JDVELH.
3. Coder un programme permettant d'afficher 2 destinations et permettant de les choisir par leur numéro. Suite au choix, la destination est annoncée comme atteinte.
4. Faire un commit "choix par numero". Le pousser (push) sur le repository distant. Fusionner votre branche avec le *master*.

Exercice 2 - Gestion des lieux

1. Tirer une nouvelle branche "gestion des lieux".
2. Créer une structure Chapitre et le type associé.
Chaque chapitre possède :
 - une description
 - un gain d'or
 - un gain de points de vie
 - un tableau d'id (les chapitres auxquels on aura accès depuis celui-ci)
3. Instancier trois chapitres au début du main(). Chaque chapitre sera désigné par un identificateur de format chapitre+numero (exemple : chapitre1, chapitre2, chapitre3).
4. Améliorer le programme de l'exercice 1 en codant une fonction "deplacement" permettant de changer de chapitre. On affiche alors le nouveau chapitre.
5. Commit. Push. Pull. Merge.

Exercice 3 - Aiguillage

1. Tirer une nouvelle branche "contraintes déplacement"
2. Créer un tableau de chapitres dans lequel on stocke les lieux précédemment instanciés. S'en servir pour permettre au joueur de naviguer entre les chapitres en les référençant par leur index dans le tableau. Ainsi, chapitre 12 devient chapitre[12].
3. Lors du déplacement, vérifier que le lieu de destination était effectivement éligible. S'il ne l'était pas, ne pas changer le chapitre courant, et afficher à la place un message disant que ce choix est impossible.
4. Commit. Push. Pull. Merge.

Exercice 4 - Ressources

1. Tirer une nouvelle branche "ressources"
2. Créer une jauge de points de vie, qui vaut 100 en début de partie. Créer un pécule d'or, au début égal à 10. Faire en sorte qu'à chaque nouveau chapitre, ces valeurs soient ajustées par la valeur du chapitre en question (en + ou en -). Le minimum d'or est de 0, le max de vie est de 100. En dessous de 0 de vie, c'est la fin de la partie.
3. Commit. Push. Pull. Merge.

Exercice 5 - Chemins conditionnels

1. Tirer une nouvelle branche "chemins conditionnels".
2. Faire en sorte que certains chemins entre deux chapitres ne soient accessibles que sous une condition précise. Exemple : "si vous avez au moins 20 pieces d'or, vous pouvez aller en 32. Sinon (ou si vous ne souhaitez pas dépenser d'or), allez en 323".
3. Commit. Push. Pull. Merge.

Exercice 6 - Bonus

Mettre en place un système permettant de sauvegarder sa progression (et de reprendre sa partie là où on l'a laissée), dans un document texte situé dans le même répertoire que l'exécutable. Se servir pour cela de la fonction `fopen()` accessible grâce à `stdio.h`.