

Projet	Jeu de rôle	
Nombre de personnes	1 binôme	
Date de remise	Vendredi 30 novembre 2012	
Pénalités	¼ point par jour de retard	

DESCRIPTION DU JEU

Le but est de mettre au point un jeu multi-joueurs où chaque joueur possède 2 héros : un Troll et un Magicien pourvus tout deux de pouvoirs magiques. Le jet d'un dé à 6 faces permet aux personnages de se déplacer sur un plateau constitué de cases. Partant de la case de départ, le but pour le joueur est d'amener à la dernière case ses deux héros afin de battre l'incontournable Boss de fin. Cependant pour y parvenir les héros devront réussir différentes épreuves au hasard de leurs déplacements, ainsi que des combats contre les héros des autres joueurs s'ils se rencontrent. En cas d'échec le héros devra tout recommencer !

RÈGLES DE JEU

Le jeu comporte des dés, 2 héros (Troll et Magicien) par joueur et un plateau de 22 cases. Le jeu peut se jouer de 2 à n joueurs. Le Boss de fin peut être géré par un joueur ou par l'ordinateur.

Les dés

Trois types de dés sont utilisés pour le jeu :

- **1 dé à 6 faces** pour les déplacements des héros sur le plateau ;
- **1 dé à 10 faces** pour les attaques des monstres et des héros ;
- **1 dé à 20 faces** pour les attaques du Boss de fin.

Les personnages

Chaque personnage est caractérisé par ses points d'expériences et, le cas échéant, par son pouvoir magique. Voici la liste des personnages avec leurs caractéristiques :

	Points d'expérience	Pouvoirs magiques
<i>Le Troll</i>	150 points	1 attaque massue
<i>Le Magicien</i>	100 points	1 invocation
<i>Le Monstre</i>	variables	aucun
<i>Le Boss de fin</i>	> 1000 points	1 déconcentration

- L'**attaque de massue** du Troll consiste à lancer une attaque de 40 jets du dé à 10 faces.
- L'**invocation** du magicien permet de placer un monstre sur sa case. Ce monstre ne pourra pas attaquer le Troll allié. Le nombre de points d'expérience du monstre sera défini au hasard par le programme. Il sera assez puissant pour ralentir l'adversaire qui croisera son chemin.
- La magie de **déconcentration** du Boss de fin fait sauter un tour à l'adversaire.

Attention : chaque magie ne peut être utilisée qu'une fois par partie ; à part le Boss de fin qui peut faire appel à sa magie une fois par combat.

Les cases

Les cases représentent les différentes épreuves que doivent accomplir les héros. Ils en rencontreront 4 types :

- La case **sanctuaire** (CS) : aucune action ne peut être engagée dans cette case. C'est-à-dire que les héros qui s'y rencontrent ne se combattent pas.
- La case **question** (CQ) : lorsqu'un héros s'arrête sur cette case, il doit répondre correctement à la question posée pour obtenir des points d'expérience supplémentaires. Il devra combattre les héros (des joueurs différents) qui se trouvent également sur cette case.

- La case **monstre** (CM) : si un héros s'arrête sur une telle case, il devra combattre le monstre qui s'y trouve. Lorsqu'un monstre est battu la case se transforme en case pause.
- La case **pause** (CP) : dans cette case aucune épreuve n'est proposée au héros, par contre il devra combattre les héros des joueurs différents s'il en rencontre.

Le plateau

Il s'agit d'un tableau qui doit comporter une première et une dernière case de type CS. Il est formé également de 8 CQ, 8 CM et 4 CP, comme par exemple :

CS	CQ	CM	CQ	CP	CM	CQ	CM	CP	CQ	CM	CQ	CP	CM	CQ	CM	CP	CQ	CM	CM	CQ	CS
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Les héros se déplacent sur le plateau de la case CS de gauche jusqu'à la case CS de droite. Tous les héros effectuent un déplacement par tour, d'une valeur définie à partir du dé à 6 faces. Après déplacement d'un héros l'action associée à la case est engagée immédiatement. Lorsqu'un héros arrive sur une case où est présent un adversaire, la première opération sera d'effectuer l'action de la case (répondre à une question ou combattre une monstre créé par l'adversaire) et en suite seulement le combat est engagé avec l'adversaire.

Les combats

Il y a trois types de combat :

- **Combat contre un adversaire ou un monstre** : à chaque tour, un jet du dé à 10 faces est effectué par chaque personnage. La valeur est soustraite aux points d'expérience de l'ennemi. Le combat se fini lorsque l'un des deux n'a plus de points d'expérience. En cas de victoire le capital expérience du héros augmentera d'une valeur valant la moitié de la valeur d'expérience de l'ennemi battu. En cas d'échec le héros se replacera sur la case de départ sans perdre ses points d'expériences déjà acquis mais sans en gagner.
- **Combat contre le Boss final** : le combat n'est engagé que lorsque les deux héros du joueur sont sur la case CS finale. Le Boss possède un dé à 20 faces d'attaque alors que les deux héros ont chacun un dé à 10 faces. Dans un même tour, le Troll, le magicien et le Boss de fin font leur attaque.
- La **magie** est toujours suivie d'une attaque.

MODALITÉS DU PROJET

Il vous appartient de mener le projet de manière autonome, que ce soit pour développer le programme ou pour utiliser les outils requis.

Le programme est à développer en langage C. Les outils à utiliser sont tous disponibles sur les machines d'enseignement (emacs, gcc, gprof, doxygen, make, tar, OpenOffice) et leur documentation sur le Web.

Vous pouvez utiliser toutes documentations nécessaires pour atteindre l'objectif (livres, manuels ou Internet). Vous pouvez aussi proposer des extensions du sujet. Néanmoins, vous devez au minimum réaliser ce qui demandé.

Si des points particuliers ne sont pas clairs ou pas assez précis, vous devez faire des choix pour lever ces ambiguïtés puis en rendre compte dans le rapport de projet.

Trois choses seulement sont imposées pour réaliser le logiciel : l'utilisation de la fonction rand() (avec srand()) pour la génération des nombres aléatoires, la définition de tests d'exécution et l'archivage des questions dans un fichier texte.

La génération des nombres aléatoires

On utilisera la fonction C rand() et srand() pour effectuer les tirages aléatoires.

- rand() retourne un entier aléatoire entre 0 et RAND_MAX (la valeur de RAND_MAX est définie dans le fichier stdlib.h) (voir \$ man rand).
- srand(x) initialise la graine du générateur de nombre aléatoire avec la valeur x. En général, on utilise srand(time(NULL)) pour initialiser la graine avec le nombre de secondes écoulées depuis le 1 janvier 1970. (voir man -s2 time)

Il est à noter que si la graine donnée comme paramètre de la fonction srand() est toujours la même, la suite des nombres aléatoires générés par la fonction rand() sera toujours la même.

Pour générer une suite de nombre aléatoire différente à chaque nouvelle exécution du programme, on utilise généralement srand(time(NULL)).

Les tests automatiques

Les tests ont pour objectif de vérifier que le programme fonctionne correctement.

Pour construire ces tests, il faut profiter du fait que la génération d'une même suite de nombres aléatoires permet de reproduire une même exécution du programme. Il suffit alors de mettre les réponses aux questions successives posées par le programme dans un fichier texte puis d'exécuter le programme avec comme entrée standard le dit fichier pour réaliser une exécution du logiciel sans intervention d'un utilisateur :

```
$ logiciel < test.txt
```

À partir de l'analyse de cette exécution, il est possible de vérifier que le programme fonctionne correctement et respecte les actions prévues.

Le fichier des questions

Les questions sont à inventer. Elles sont stockées dans un fichier de type ASCII dont la structure est la suivante :

question1|choix1|choix2|choix3|n° de la réponse.↵

question2|choix1|choix2|choix3|n° de la réponse.↵

...

Exemple de question :

Quel pourcentage de la surface de la Terre les océans recouvrent-ils?|30|55|71|3

LIVRABLES

Le rapport du projet devra se faire sous trois formes :

1) Un serveur Web

Ce serveur contient :

- la documentation en ligne sur le code.

Cette documentation est générée automatiquement par le logiciel Doxygen à partir des fichiers C.

2) Un courriel à votre enseignant de TP

- Un fichier archive tar avec tous les dossiers et fichiers de votre projet. Cette archive devra contenir :
 - un Makefile
 - un fichier README qui indique comment compiler le logiciel et comment l'exécuter.
 - Les fichiers avec les tests que vous avez réalisés.
 - Les codes sources C.

Il doit être possible de re-générer le logiciel à partir de cette archive. Vous prendrez soin de ne mettre dans l'archive que les fichiers nécessaires (pas de fichiers exécutables et ni de fichiers suffixés par '~' générés par emacs).

3) Un compte rendu papier (à faire avec OpenOffice ou Word)

- La présentation de ce qui a été réalisé et éventuellement ce qui n'a pas été réalisé ;
- L'analyse de l'exécution du programme par gprof pour générer un "profilage" du programme. L'analyse doit mettre en évidence les fonctions les plus exécutées et celles qui sont les plus longues en temps d'exécution;
- La description des types de tests construits (ce qui est analysé par chacun des tests).