

Devoir de Programmation Orientée Objet

Licence Informatique

Réalisation d'un Jeu d'Aventure *

Année 2018/2019

1 Le jeu

Votre tâche consiste à inventer et implémenter un jeu d'aventure. Comme point de départ, vous pouvez utiliser le logiciel *Zork* que vous pouvez télécharger depuis la page web du cours à :

<https://moodle.univ-paris13.fr/mod/resource/view.php?id=804>

Zork est un jeu très rudimentaire qui permet juste de se déplacer dans quelques pièces.

Avant toute chose, il est important de lire le code existant (au moins en partie) et d'essayer de comprendre ce qu'il fait. Dans un deuxième temps, afin d'améliorer votre compréhension du code apportez quelques modifications au code du jeu. Par exemple :

- changez le nom d'une pièce.
- modifiez la carte des pièces : prenez une pièce située actuellement à l'ouest d'une autre pièce et placez la au nord.
- ajoutez une ou plusieurs pièces.

2 Première partie : amélioration du logiciel *Zork*

2.1 Travail à faire

En partant du logiciel fourni, vous devez ajouter *toutes* les fonctionnalités suivantes :

1. Certaines pièces contiennent des objets. Chaque pièce peut contenir un nombre arbitraire d'objets. Certains objets peuvent être emportés par le joueur, d'autres non.
2. Le joueur peut transporter des objets avec lui. Chaque objet a un poids. Le joueur ne peut transporter des objets que dans la mesure où le poids total des objets transportés ne dépasse pas une certaine limite fixée à l'avance.
3. Le joueur peut gagner. Certaines situations doivent pouvoir être reconnues comme des situations gagnantes dans lesquelles le joueur sera averti qu'il a gagné.
4. Implémentez une commande "retour" qui fait revenir le joueur dans la pièce précédemment visitée.
5. Ajoutez au moins quatre nouvelles commandes (en plus de celles déjà présentes dans le code qui vous est fourni).

Date limite : Cette première partie du travail doit être rendue en déposant sur le site web du cours un fichier archive contenant l'ensemble de votre travail au plus tard le vendredi 2 novembre 2018.

2.2 Travail à rendre

Cf. 3.3

*M. Champesme – Département d'Informatique – Institut Galilée – Université Paris 13

3 Deuxième partie : création d'un nouveau jeu

3.1 Inventez votre propre version du jeu

Tout d'abord, vous devez décider quel est le but de votre jeu. Ce devrait être quelque chose du genre : *Vous devez trouver certains objets et les apporter dans une certaine pièce (ou à une certaine personne). Ensuite vous pouvez ramasser d'autres objets et si vous les amenez dans une autre pièce, vous gagnez.*

Par exemple : *Vous êtes étudiant à l'université Paris 13. Vous devez trouver votre salle de TP. Pour la trouver, vous devez vous rendre au secrétariat pour vous renseigner. A la fin du jeu, vous devez trouver la salle d'examen. Si vous y arrivez à temps, que vous avez récupéré un support de cours sur votre chemin et que vous avez assisté au cours, au TD et au TP, alors vous gagnez. Cependant, si vous êtes passé plus de cinq fois à la cafétéria, vous n'aurez que la moitié des points à l'examen.*

Ou bien : *Vous êtes perdu dans un donjon. Vous rencontrez un gnome. Si vous trouvez quelque chose de comestible à lui offrir, il vous indiquera où trouver une baguette magique. Si vous utilisez la baguette magique dans la salle des banquets, la sortie s'ouvre, vous sortez et vous gagnez.*

En fait, vous êtes entièrement libre dans la définition de votre jeu. Pensez au lieu dans lequel se déroulera votre jeu, déterminez à quoi correspondront les pièces. Imaginez les objets, les personnages ou les monstres qui peupleront les lieux. Définissez la mission que devra accomplir le joueur... Essayez de rendre votre jeu intéressant, mais ne le rendez pas trop compliqué.

3.2 Implémentez votre version du jeu

Une fois les caractéristiques de votre jeu définies, la phase d'implémentation peut commencer. Cette seconde partie du travail pourra indifféremment se baser sur le travail déjà effectué dans la première partie ou bien donner lieu à la réalisation d'un logiciel totalement nouveau. Dans tous les cas, le logiciel final devra posséder, au minimum, toutes les fonctionnalités (i.e. les fonctionnalités de base *et* les fonctionnalités avancées) présentées ci-après.

3.2.1 Fonctionnalités de base

Votre jeu doit implémenter les fonctionnalités de base ci-après (si vous réutilisez le logiciel fourni cette première partie du travail aura déjà été réalisée pour la première partie du devoir) :

1. Le jeu comporte plusieurs pièces.
2. Le joueur peut se déplacer d'une pièce à l'autre.
3. Certaines pièces contiennent des objets. Chaque pièce peut contenir un nombre arbitraire d'objets. Certains objets peuvent être emportés par le joueur, d'autres non.
4. Le joueur peut transporter des objets avec lui. Chaque objet a un poids. Le joueur ne peut transporter des objets que dans la mesure où le poids total des objets transportés ne dépasse pas une certaine limite fixée à l'avance.
5. Le joueur peut gagner. Certaines situations doivent pouvoir être reconnues comme des situations gagnantes dans lesquelles le joueur sera averti qu'il a gagné.
6. Implémentez une commande "retour" qui fait revenir le joueur dans la pièce précédemment visitée.
7. Ajoutez au moins quatre nouvelles commandes (en plus de celles déjà présentes dans le code qui vous est fourni).

3.2.2 Fonctionnalités avancées

Une fois les fonctionnalités de base implémentées, vous pouvez passer aux choses sérieuses :

1. Ajoutez à votre jeu des êtres animés. Les êtres animés (personnages, animaux, monstres...) sont des entités capable de se déplacer. Comme les objets ou le joueur, les êtres animés sont localisés dans des pièces, mais à la différence des objets, les êtres animés peuvent se déplacer de leur propre initiative.
2. Étendez l'analyseur syntaxique afin qu'il puisse reconnaître des commandes formées de trois mots. Vous pourriez, par exemple, avoir une commande
donner pain gnome
permettant de donner du pain (que vous transportez avec vous) au gnome.
3. Ajoutez une pièce de téléportation : chaque fois que le joueur pénètre dans cette pièce il est téléporté dans une autre pièce choisie au hasard.

Vous pouvez aussi imaginer d'autres fonctionnalités avancées, mais, dans ce cas, vous devrez en discuter avec votre enseignant et obtenir son approbation avant de les implémenter. Votre enseignant vous avertira alors s'il considère que cette fonctionnalité est trop complexe ou bien demande trop de travail.

Date limite : Cette deuxième partie du travail doit être rendue en déposant sur le site web du cours un fichier archive contenant l'ensemble de votre travail au plus tard le vendredi 11 janvier 2019.

3.3 Travail à rendre

Vous devez rendre le code source de l'ensemble des classes constituant votre projet ainsi qu'un rapport.

3.3.1 Code source

Pour rendre votre travail, vous devez déposer sur le site du cours (cf. "Déposer un fichier" en bas de la page du devoir (1ère ou 2ème partie)) un fichier archive (.zip ou .tar ou .tgz ...) contenant le code source (i.e. les fichiers .java). Tout votre code doit être de qualité "professionnelle" (commentaires complets au format javadoc décrivant le contrat pour toutes les classes et toutes les méthodes de chaque classe, indentation du code, choix réfléchi des identificateurs...). Votre code sera évalué entre autre sur :

- sa correction : pas d'erreurs de compilation ou d'exécution et implémentation conforme à l'interface (y compris le contrat) des classes.
- utilisation la plus large possible des concepts spécifiques à la programmation orientée objet, notamment la définition du contrat de chaque classe, l'héritage, la généricité, les exceptions, ...
- le style : commentaires, indentation, choix des identificateurs, ...
- difficulté : des points "bonus" seront accordés en fonction de la difficulté des extensions réalisées.

Pour l'évaluation de ce travail, une plus grande attention sera portée à la qualité du code (mise en œuvre des concepts objet, définition et respect du contrat, ...), qu'à la *jouabilité*.

3.3.2 Rapport

Vous devez aussi rendre un rapport (à inclure dans le fichier archive mentionné plus haut) contenant :

- le nom et une brève description de votre jeu. Cette description doit, au minimum, inclure une description à destination de l'utilisateur (que fait le jeu ?) et une brève description de l'implémentation (quelles sont les caractéristiques les plus importantes de l'implémentation ?) :
 - présentation générale ;
 - règle du jeu : liste de toutes les commandes (nom de la commande, paramètres éventuels, description détaillée du mode d'emploi et de ce que fait cette commande), but du jeu (comment gagner ?), ...
 - plan du jeu présentant toutes les pièces avec leurs noms, les passages entre les pièces et les objets présents dans chacune d'entre elles ;
 - au moins deux scénarios de test (par exemple : un scénario gagnant et un scénario perdant). Chaque scénario doit lister la totalité des commandes à effectuer pour aboutir à la fin du jeu.
- modifications apportées au logiciel :
 - liste des nouvelles classes, en indiquant, pour chacune, la raison pour laquelle cette classe a été créée, son rôle dans le fonctionnement global du logiciel, les caractéristiques communes des instances de cette classe ;
 - liste des classes modifiées : pour chacune des classes modifiées, indiquez quelles méthodes ont été modifiées ou ajoutées, la raison pour laquelle ces modifications ont été faites et la nature de ces modifications ;
 - le cas échéant, indiquez les erreurs présentes et les problèmes non résolus dans votre programme en donnant, pour chaque erreur, un scénario précis permettant de la reproduire (Note : les bugs répertoriés et documentés dans votre rapport seront beaucoup moins pénalisés dans l'évaluation que les bugs découverts par le correcteur).
- cf. section suivante si votre travail a été réalisé en groupe

3.3.3 Travail en groupe

Le travail en groupe est autorisé, à certaines conditions :

- Le rapport ne doit pas être réalisé en groupe.
- Que le travail soit fait seul ou en groupe l'évaluation est individuelle, il est donc indispensable que la contribution de chacun soit clairement indiquée. Dans ce but, vous devez indiquer dans votre rapport les noms de tous les étudiants ayant participé à la réalisation du devoir que vous rendez en précisant l'apport de chacun et en insistant sur votre contribution personnelle. Pour chaque classe, utilisez le tag `@author`, pour préciser dans le commentaire de classe toutes les personnes intervenues dans la réalisation de cette classe.
- Dans le cas où les règles précédentes ne seraient pas respectées, les similarités flagrantes seront assimilées à de la fraude et pénalisées.