

Projet Java III - 2023-2024

HELBTower

Introduction :

Dans le cadre du cours de Java III, il vous est demandé d'implémenter un jeu d'arcade : HELBTower.

Dans un monde fantastique en proie aux ténèbres, le valeureux héros se leva pour combattre les forces du mal qui menaçaient son royaume. Pour réussir sa quête, il devait pénétrer dans la cour des anciens dont les profondeurs renfermaient des trésors magiques capables de bannir l'obscurité. Cependant, ce lieu était peuplé de gardiens maléfiques qui surgissaient des sombres tours du château.



Description des éléments du jeu :

L'espace de jeu :

La cour du château est un espace en deux dimensions, rectangulaire, dont les délimitations sont des obstacles pour les personnages qui s'y déplacent. Cet espace est composé d'un certain nombre de cases, chaque case pouvant contenir un élément du jeu.

Les personnages :

Deux types de personnages se déplacent dans la cour du château : Le héros, contrôlé par le joueur et les gardiens qui sont les adversaires du joueur.

Les pièces :

Au début de la partie, chaque case qui n'est pas occupée par un autre élément du jeu, contient une pièce, qui peut être collectée par le joueur. Le joueur gagne la partie et peut accéder au niveau supérieur s'il ramasse toutes les pièces présentes sur l'espace de jeu.

Les murs :

La cour du château contient un certain nombre de murs, fixes, qui sont des obstacles rectangulaires pour les personnages. Ces derniers ne peuvent pas les traverser. Les murs ont une taille plus ou moins égales aux personnages qui se déplacent dans la grille de jeu. Les murs peuvent être placés les uns à côté des autres pour former des structures plus complexes à l'intérieur de la cour.

Les tours :

Les tours sont les habitacles des gardiens. Une tour est habitée par un gardien. Il y a quatre tours présentes dans la cour, une tour par type de gardien. Initialement, les gardiens sont prisonniers au sein de leur tour. Lorsque le joueur a ramassé 25 pourcents des pièces présentes sur l'espace de jeu, les gardiens sortent de leur tour. Les tours sont des obstacles pour tous les personnages du jeu. Une fois sortis de leur tour, les gardiens ne peuvent plus y pénétrer.

Les gardiens :

Il existe 4 types de gardiens, chacun avec un comportement qui lui est propre.

Le gardien orange a un comportement **hasardeux**. Chacun de ses déplacements se fait sur une case aléatoire.

Le gardien bleu a un comportement **exploratoire**. Il se déplace toujours vers une case, aléatoire, choisie parmi toutes celles qu'il n'a pas encore explorée, jusqu'à avoir exploré toutes les cases du plateau de jeu. Une fois toutes les cases explorées, il recommence.

Le gardien mauve a un comportement **baladeur** et une vitesse variable. Il se déplace toujours vers une prochaine tour, choisie de manière aléatoire. Sa vitesse initiale est la même que celle des autres gardiens mais celle-ci augmente pour chaque pièce ramassée par le joueur. Il va donc de plus en plus vite.

Le gardien rouge a un comportement **prédateur**. Il se déplace constamment dans la direction de la case où se situe le héros.

Déplacements et vitesses :

Sauf exception, les gardiens n'arrêtent jamais leur déplacement dans la cour du château.

Les gardiens, ainsi que le héros, peuvent se déplacer horizontalement ou verticalement. Pas en diagonale. Tous les personnages ont la même vitesse de base.

Potions et Capes magiques :

Au lancement du jeu, la cour du château peut contenir un certain nombre de potions et de capes magiques. Ces éléments sont fixes et placés de façon aléatoire dans la cour du château.

Quand le héros collecte une potion, il la boit, ce qui a pour effet d'augmenter temporairement sa vitesse. Il existe trois types de potions, rouge, orange et jaune. La seule différence entre ces trois types de potions est la durée du bonus de vitesse qu'elles procurent.

Quand le héros collecte une cape magique, il s'en équipe, ce qui lui permet de traverser, une fois, un mur.

Période de jeu :

Il existe trois périodes de jeu, matin, jour, et soir, qui représentent la difficulté du jeu.

Dans la configuration matin, les gardiens ont un malus de vitesse par rapport à leur vitesse de base, ce qui les rends plus lents que le héros.

Dans la configuration soir, les gardiens ont un bonus de vitesse par rapport à leur vitesse de base, ce qui les rends plus rapides que le héros.

Le changement de période est visible grâce à un changement de la couleur de background du jeu.

Initialement, le jeu commence toujours à la période jour.

Téléporteurs :

Chacun des quatre bords de l'espace de jeu contient un téléporteur. Les téléporteurs fonctionnent par pair. Quand le joueur rentre dans un des téléporteurs, il ressort par l'autre téléporteur correspondant. Seul le héros peut les emprunter.

Systèmes de niveaux et fin de partie :

Quand le joueur termine un niveau, il passe au niveau suivant. La configuration de l'espace de jeu reste la même dans tous les niveaux. Cependant, la vitesse de base des gardiens augmente à chaque niveau. Il n'y a pas de niveau final.

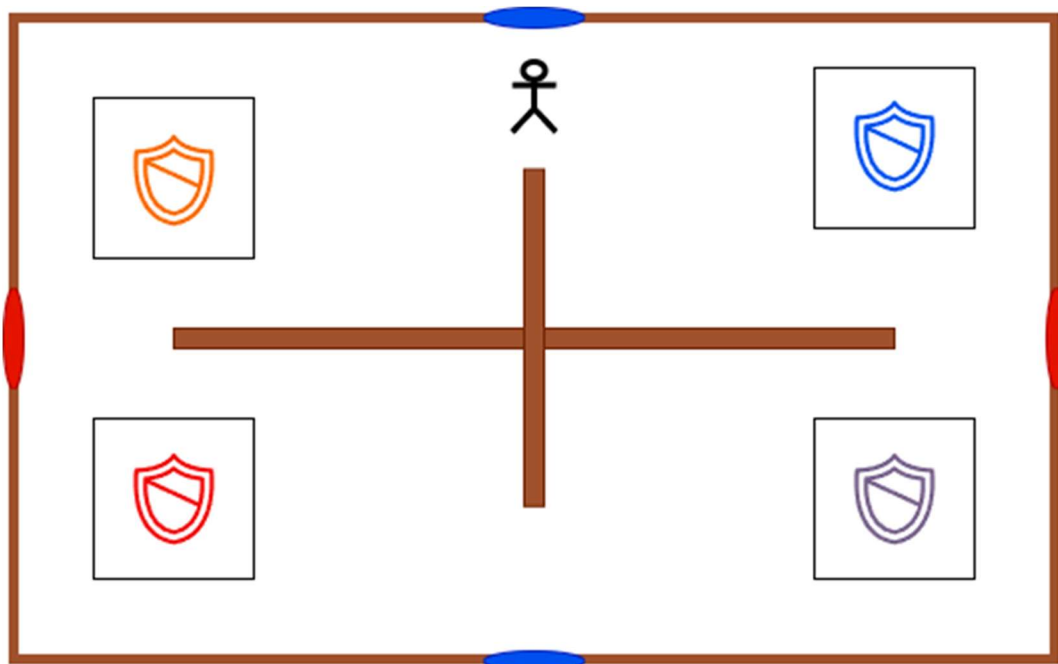
Le jeu se termine quand le joueur est attrapé par un gardien. Un message de fin de partie est alors affiché au joueur qui peut voir l'affichage du meilleur score.

Systèmes de scores :

Le jeu doit proposer de retenir le meilleur score de l'utilisateur. Ce score doit persister, même si le programme du jeu a été arrêté. Il faut donc sauvegarder le score du joueur.

Configuration du jeu :

La conception logicielle du jeu doit pouvoir permettre son évolution. Notamment, il doit être possible de modifier le nombre de cases de l'aire de jeu (en hauteur et largeur) à partir du code. Il vous est également demandé de proposer, la configuration suivante des éléments du jeu :



Dans cette configuration, on peut voir que les murs intérieurs de la cour forment une croix centrale. Vu qu'il s'agit d'un début de partie, les gardiens sont encore dans leurs tours. Les gardiens orange et bleu occupent initialement la partie supérieure du plateau, tandis que le rouge et le mauve occupent la partie inférieure. On peut également voir les téléporteurs et leur correspondances, placés sur les bords de l'espace de jeu. Le joueur commence la partie dans l'espace supérieur du plateau, entre les tours des gardiens orange et bleu.

Commandes interactives – Cheat codes :

Afin de permettre des interactions avec le jeu, il existe un système de commandes permettant d'influencer la partie. Quand l'utilisateur appuie sur une touche, cela déclenche un évènement précis :

0 : Reset la partie.

1 : Définir la période de jeu sur matin.

2 : Définir la période de jeu sur jour.

3 : Définir la période de jeu sur soir.

4 : Ajouter à une position aléatoire une potion.

5 : Ajouter à une position aléatoire une cape magique.

6 : Rajoute un gardien, de type aléatoire, à une position aléatoire dans la cour.

r : Cela permet de stopper le gardien rouge.

b : Cela permet de stopper le gardien bleu.

m : Cela permet de stopper le gardien mauve.

o : Cela permet de stopper le gardien orange.

s : Activation/(Désactivation) de la fonctionnalité supplémentaire

Fonctionnalité supplémentaire :

Additionnement, il vous est demandé d'implémenter au minimum une fonctionnalité supplémentaire originale – par exemple :

- Nouveau type de collectable.
- Nouveau type de gardien avec comportement propre.
- Nouvel environnement.
- Etc...

Cette fonctionnalité doit toutefois pouvoir être optionnelle et activable seulement si la touche « s » est appuyée. L'originalité et la complexité de cette fonctionnalité seront prises en compte pour son évaluation.

Notes Importantes :

Les contraintes et fonctionnalités du projet sont **susceptibles d'évoluer au cours du temps**. Pensez donc à adapter une stratégie de développement adéquate.

Certains points de la description ne sont pas précisés ou sont laissés volontairement vagues. Il revient à vous de faire certains choix d'interprétations. Veuillez toutefois à ce que votre approche soit logique et justifiée.

Contrainte de développement :

- Votre programme devra être compilable et exécutable dans un environnement Linux Ubuntu tel qu'une des machines virtuelles utilisées en cours et disponible sur le SharePoint d'eCampus. Un script bash nommé « run.sh » devra permettre la compilation et l'exécution de l'application en utilisant seulement la commande suivante « bash run.sh » en terminale. Si le fichier n'est pas fourni, ou si la compilation et l'exécution ne fonctionnent pas dans l'environnement spécifié, le projet ne sera pas corrigé et sanctionné d'un zéro. **Testez donc avant que tout fonctionne !**
- Votre code devra respecter les principes de designs orientés objet comme vu au cours. Pensez donc à faire des choix logiques de design de classes afin de produire un **code propre et maintenable**.
- Votre code devra présenter **une structure correcte et maintenable**.
Notamment :
 - Evitez la duplication de code.
 - Evitez les constantes magiques.
 - Evitez le code mort.
 - Commentez intelligemment et suffisamment votre code
 - Nommez correctement vos variables et méthodes et organisez correctement votre code.

Il est fortement conseillé de consulter la partie du cours sur les bonnes pratiques de développement. Un code dont la qualité sera jugée insuffisante **sera sanctionné d'un zéro**.

- Votre jeu devra être développé en utilisant **exclusivement la librairie graphique JavaFX** comme vu en cours **et à partir du jeu du SnakeFX** qui devra être utilisé comme base fonctionnelle.
- A l'exception des commentaires, l'entièreté de votre programme **devra être codé en anglais** (nom de variables/fonction/classes/etc...).

Rapport :

Il vous est demandé de rédiger un rapport décrivant votre projet. Votre rapport devra contenir au minimum les sections suivantes :

Introduction : Cette section devra introduire votre projet. Décrire ce qui a été réalisé et présenter brièvement la structure de votre rapport.

Fonctionnalités de base : Cette section devra expliquer les fonctionnalités offertes par votre application d'un point de vue à la fois fonctionnel et technique.

Fonctionnalités supplémentaires : Cette section devra expliquer les fonctionnalités supplémentaires offertes par votre application d'un point de vue à la fois fonctionnel et technique.

Analyse : Cette section devra expliquer la structure de votre implémentation en utilisant les outils d'analyses déjà vus durant votre parcours. Si aucun outil n'a été vu pour l'instant, considérez qu'il vous est demandé d'élaborer des schémas explicatifs sur la structure de votre code. Attention : Tous les diagrammes doivent être commentés !

Limitations : Les limites de votre application, par exemple : dans quels cas d'utilisation votre application pourrait ne pas fonctionner comme prévu ? Y a-t-il des aspects techniques qui n'ont pas été traités ? Si vous aviez plus de temps pour le projet, qu'auriez-vous amélioré ? Plusieurs points de vue sont possibles, il revient au groupe d'étudiant de choisir les points qu'il considère les plus pertinents pour réaliser son autocritique.

Conclusion : Votre conclusion sur le projet. Ce que vous avez réussi à faire ou non durant le projet et les apprentissages que vous en tirez.

Le rapport sera notamment évalué sur la qualité écrite et l'effort de présentation ainsi que la pertinence et la complétude des points abordés. **Un rapport dont la qualité écrite est jugée très insuffisante sera sanctionnée d'un zéro.**

Maitrise des productions :

L'évaluation du projet vise à attester de la bonne maîtrise, des concepts liés au cours et au développement du projet, par l'étudiant. Toute réalisation pour laquelle l'étudiant ne peut pas démontrer une maîtrise suffisante lors de l'évaluation orale ne sera pas prise en considération.

Deadline et remise :

La date limite pour la remise du projet est le **lundi 8 Janvier à 23h59**. Le projet devra être déposé sur eCampus à l'intérieur d'un fichier **.zip** contenant toutes les sources de votre projet ainsi que le rapport au format **.DOCX et PDF**.

Développement et Triche :

- Tout acte de triche sera sanctionné par **une note de fraude au bulletin et sera notifié à la direction qui pourra possiblement décider de sanctions supplémentaires**. Des parties de code réutilisés d'un projet existant (d'un autre étudiant ou disponible sur le net) sans références dans votre rapport et sans mention de l'utilité du code utilisé est considéré comme une fraude.
- Pour ce projet, **vous ne pouvez pas reprendre des parties du code d'un autre étudiant, de cette année, ou d'une année précédente**.
- Pour ce projet **vous ne pouvez pas vous inspirer/servir d'un jeu/code disponible sur internet**.
- Pour ce projet **vous ne pouvez pas vous inspirer/servir d'un code généré par des outils de génération de code tels que, par exemple, ChatGPT**.
- Si vous avez un doute, contactez l'enseignant le plus tôt possible afin d'éviter du refactoring inutile, ou pire, **une note de zéro/fraude**.

Conseil pratique :

Voici quelques conseils qui j'espère pourront vous aider.

- Veillez à ce que votre code ne contienne pas de constantes magique et/ou de duplication qui serait facilement évitable avec l'utilisation de méthodes.
- Veillez à effectivement implémenter les différents comportements demandés.
- Réfléchissez en terme d'« attribution de responsabilités » en accord avec les principes vu au cours (segmentation logique en classes).
- Ne négligez pas la théorie du cours (vous serez interrogés dessus).
- Ne négligez pas votre rapport. Tachez d'y expliquer/justifier explicitement vos choix d'implémentation. (Exemple : pourquoi un héritage ici ? pourquoi l'implémentation comme ceci ? quel avantage en termes de structure ? ...)
- Prenez le temps de bien comprendre tout l'énoncé avant de vous lancer (et lisez la FAQ).
- Vérifiez que votre code compile et run effectivement via le script bash prévu sur la machine Ubuntu présente dans le SharePoint du cours.

FAQ :

- **Puis je ajouter d'autres sections ou sous-sections dans le rapport ?**

Oui. La partie rapport de ce document donne seulement la structure minimum.

- **Puis je coder ou rendre mon rapport en anglais ?**

Oui. Pour ce qui est du code vous devez toutefois respecter les usages corrects des conventions de nommage. Codez en anglais.

- **Puis je programmer sur Windows avec Eclipse ?**

Oui vous pouvez programmer comme vous le désirez mais vous devez respecter les contraintes de ce document, notamment : votre code doit être exécutable sur un environnement linux Ubuntu via un script run.sh que vous devez fournir en même temps que vos sources (voir les contraintes de développement). Une machine préconfigurée sera disponible sur le SharePoint. **C'est cette machine qui sera utilisée pour l'évaluation de votre projet.**

- **Le rapport est-il important ?**

Oui. Le rapport est une **pièce centrale de votre projet** et c'est le premier outil de communication qui me servira à juger de la bonne réalisation du projet, pas seulement du point de vue du code mais également de la méthodologie utilisée.

- **Quel est le niveau de complexité que vous attendez pour les fonctionnalités supplémentaires ?**

La complexité de ce qui été développé sera prise en compte dans l'évaluation. Toutefois, entamer le dev. d'une fonctionnalité trop ambitieux risque de vous pénaliser si cela implique que vous deviez bâcler votre rapport pour compenser ou omettre des fonctionnalités demandées.

- **Que voulez-vous dire par « tous les diagrammes doivent être commentés ».**

Les diagrammes doivent servir à illustrer et appuyer vos explications sur la structure de votre implémentation. Ils ne remplacent aucunement un texte explicatif revenant sur les points d'attention.

- **Je n'ai pas réussi à tout réaliser. Est-ce que ça vaut la peine de vous rendre le projet ?**

Oui veuillez toutefois à être claire sur les parties non implémentées. Il est très déconseiller de dissimuler ou d'« oublier » de mentionner qu'une partie n'a pas été réalisée. Veuillez toutefois à bien respecter les consignes. Par exemple, votre code doit pouvoir compiler avec le script bash demandé, la qualité du code doit être suffisante, etc...

- **Puis je réaliser le projet en groupe ?**

Non le projet doit être réalisé individuellement.

- **Que voulez-vous dire par « Votre code devra respecter les principes de designs orientés objet comme vu au cours. »**

L'orienté objet fait intervenir certains principes comme l'héritage ou les méthodes statiques. Il revient à vous de décider quand les mettre en œuvre ou non. Votre approche devra toutefois être logique et justifiée. Cela implique notamment, de faire apparaître de l'héritage quand cela a du sens, de rendre une classe abstraite quand cela a du sens, d'utiliser intelligemment l'encapsulation etc...

- **Dois-je vraiment faire de l'orienté objet ? Mon programme peut fonctionner sans.**

Vous devez absolument mettre en œuvre l'orienté objet pour ce projet. Cela fait partie des contraintes du projet. Un non-respect de ces contraintes sera pénalisé. L'utilisation de l'orienté objet n'est pas une contrainte faible. Veillez donc à la respecter.

- **Je ne peux vraiment pas utiliser de code venant d'internet ?**

Non appart pour ce qui peut être considéré comme des briques fonctionnelles. (Par exemple le code permettant de lire/écrire un fichier, le code relatif à l'utilisation des listes ou autres structures de données). Dans tous les cas ne prenez aucun risque et contactez l'enseignant le plus tôt possible si vous avez un doute !

- **L'aspect graphique du jeu et la jouabilité sont-ils des critères importants ?**

Ces critères peuvent être pris en compte dans l'évaluation mais sont nettement moins importants que l'implémentation des fonctionnalités et le respect des contraintes. Dis grossièrement : Mieux vaut un jeu moche mais avec toutes les fonctionnalités implémentées qu'un jeu magnifique mais avec des fonctionnalités manquantes.

- **Quels sprites puis je utiliser pour le projet ? Dois-je vraiment trouver une image de tour ?**

Vous êtes libre d'utiliser toutes les sprites/images que vous désirez. Pensez toutefois à spécifier leur source dans le rapport. Vous pouvez également utiliser des images abstraites. Vous êtes vraiment libre en ce qui concerne l'aspect purement graphique.

- **Puis je utiliser un outil comme Scene Builder pour la création d'interfaces graphique ?**

Non, vous ne pouvez utiliser que les bases fonctionnelles du SnakeFX partagées par l'enseignant sur eCampus.