

Le jeu “Donjons et Dragons”

Introduction

Le but de ce projet est de créer un jeu, inspiré des jeux de plateaux de l'univers « Donjons et Dragons », en utilisant le langage Java. Ce projet se découpe en plusieurs itérations, au cours desquelles de nouvelles fonctionnalités vont vous être demandées. Le détail du contenu des itérations se trouve en fin de document.

Notes: Ce projet doit être réalisé en îlot. En effet la quantité de fonctionnalités demandées est trop importante pour être réalisé individuellement sur le temps imparti.

Ce projet doit vous permettre, en plus d'acquérir des compétences en programmation, de progresser sur l'aspect collaboratif dans un projet conséquent (architecture, synchronisation, utilisation de GIT).

Pour chaque itération, veuillez à bien lire les fonctionnalités et livrables demandées et ainsi vous répartir le travail. Il est fortement conseillé de passer par une phase de réflexion sur “papier” avant de vous lancer dans le développement à proprement parlé.

LISEZ ATTENTIVEMENT LE SUJET ! :)

Règles du jeu

Le but du jeu est de faire avancer un personnage du début à la fin d'un plateau de jeu, en combattant des ennemis.

Le plateau de jeu

Le plateau de jeu est constitué de 64 cases. Chaque case pouvant soit être vide, soit contenir un ennemi à combattre ou contenir une caisse surprise.

Les caisses surprises peuvent contenir des objets (armes, sorts, potions)

Les personnages

Au début de chaque partie, l'utilisateur choisit son personnage (guerrier ou magicien) .

Chaque type de personnage est caractérisé par les attributs suivants :

- Guerrier :
 - Nom
 - Image
 - Niveau de vie (5 par défaut - 10 max)
 - Force d'attaque (5- 10)
- Magicien :
 - Nom
 - Image
 - Niveau de vie (3 - 6)
 - Niveau d'attaque (8 - 15)

Les équipements offensifs

Les armes/sorts sont caractérisés par un niveau d'attaque, qui lorsqu'ils sont utilisés, vont venir augmenter d'autant le niveau d'attaque du personnage.

On proposera dans une première version du jeu, les équipements offensifs décrits ci-dessous :

Les armes (spécifiques au guerrier) :

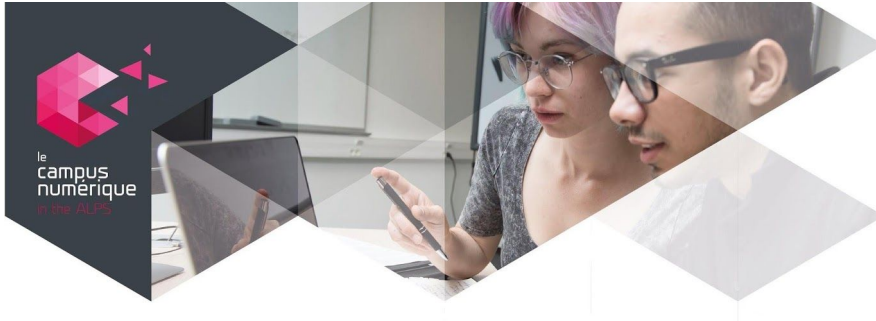
- Arc : augmente l'attaque de 1 point
- Massue : augmente l'attaque de 3 points
- Epée : niveau d'attaque augmenté de 5 points.

Les sorts (spécifiques au magicien) :

- Eclair: niveau d'attaque augmenté de 2 points.
- Boule de feu : niveau d'attaque augmenté de 7 points.

Les potions (utilisable par tous les personnages) :

- Potion de vie mineure : rends 1 point de vie.
- Potion de vie standard : rends 2 points de vie.
- Grande potion de vie : rends 5 points de vie.



Le jeu "Donjons et Dragons"

Les ennemis

Comme énoncé ci-dessus, les personnages peuvent être confrontés à des ennemis au cours du jeu. Ces derniers ont plus ou moins de points de vie selon leur type, et frappe également plus ou moins fort :

- Les gobelins: 6 points de vie, 1 point d'attaque.
- Les sorciers : 9 points de vie, 2 points d'attaque.
- Les dragons : 15 points de vie, 4 points d'attaque.

Déroulement du jeu

Ce jeu se déroule en mode tour par tour. À chaque tour, le joueur va lancer un dé virtuel à 6 faces afin de connaître le nombre de cases duquel il avance

- Si la nouvelle case atteinte est vide : on passe au tour suivant.
- Si la nouvelle case contient une caisse surprise :
 - o Équipement : le bonus est appliqué au personnage
 - o Potions: Le personnage récupère le nombre de points de vie défini par le type de potion.
- Si la nouvelle case atteinte contient un ennemi, le combat s'engage.

Règle des combats

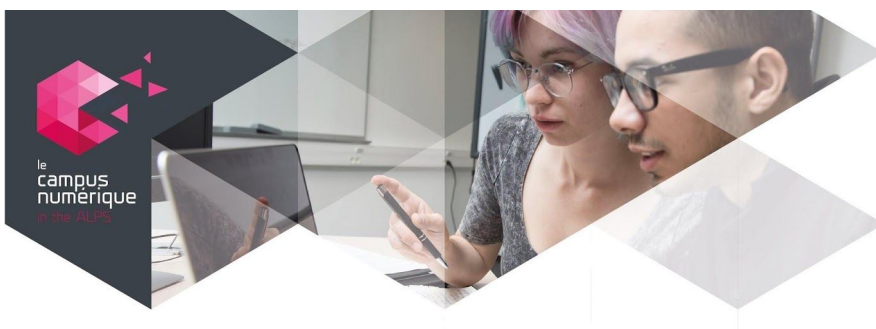
Le personnage frappe l'ennemi avec la force définie par son équipement (arme ou sorts) et le niveau de vie de l'ennemi diminue en conséquence.

- Si le niveau de vie de l'ennemi atteint 0, ce dernier meurt.
- Sinon l'ennemi réplique et le niveau de vie de du personnage diminue en fonction de la force de frappe de l'ennemi. Ce dernier s'enfuit lorsqu'il vous a frapper
- **Note :** Le niveau de vie d'un même ennemi (sur une même case) doit être persistante d'un tour à l'autre. Si le joueur parvient à vaincre totalement un ennemi (niveau de vie à 0), cet ennemi doit disparaître du plateau.

Règle de fin de partie

La partie est gagnée si le joueur arrive au bout du plateau.

En revanche si le joueur perd tous ses points de vie, la partie est perdue.



Le jeu “Donjons et Dragons”

Les différentes versions du jeu

Le développement de ce jeu va se faire de manière itérative. À chaque version, de nouvelles fonctionnalités seront attendues avec une difficulté croissante.

Dans les premières versions du jeu, nous travaillerons avec une application de type console afin de mettre en place le modèle de données et de la partie logique, mais avec un affichage minimaliste.

Par la suite, nous allons connecter un client web sur notre jeu pour plus de jouabilité :)

Dans les deux cas, **nous vous fournissons** les versions de base des clients console et web. Ces deux types de client vont s’interfacer avec le moteur du jeu (que vous allez devoir implémenter) via une API que nous vous fournissons également.

Sauf instructions contraires dans les itérations, **vous n’avez rien à modifier** dans le client console fourni ou dans l’API pour les première itérations.

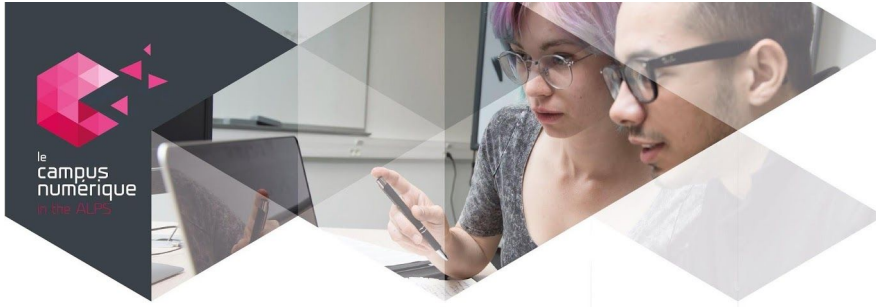
Itération 1

Le but de cette première itération est d’implémenter une première version du jeu où le joueur doit pouvoir :

- Entrer son nom
- Choisir son personnage (Guerrier ou magicien). Les personnages disponibles sont créés en “dur” par le jeu en utilisant les données énoncés plus haut dans ce document. Pour ce faire, vous pouvez ré-utiliser une grande partie du code écrit dans les mini-projets précédents.
- Commencer une partie et se déplacer sur le plateau grâce au dé tant qu’il n’est pas arrivé au bout du plateau. Pas de notions d’ennemis, de combats et de caisses surprises dans cette première version. Le programme doit afficher l’avancement du joueur sur le plateau (case n / 64).
- Une fois arrivé au bout du plateau, choisir de quitter le jeu ou de recommencer une partie.

Afin de vous faciliter le travail, nous vous fournissons le code console qui contient la boucle principal du programme ainsi que l’ordonnancement d’une partie décrite ci-dessus.

Vous pouvez faire un fork sur vos git du repository suivant :
https://github.com/le-campus-numerique/Java_Warriors_CodeDeBase



Le jeu "Donjons et Dragons"

Ce repository contient deux packages java contenant le code source de l'application console, la définition de l'API et la documentation associée.

Pour fonctionner, le client console fait des appels sur un objet du type d'une interface correspondant à l'API. En l'état, le projet ne compile pas car la classe qui doit implémenter l'API n'existe pas encore, c'est le point de départ de l'itération pour vous.

Nous vous conseillons de prendre le temps de bien étudier le code et la documentation qui vous sont fournis avant de démarrer votre réflexion.

Itération 2

Dans cette deuxième version, nous allons enrichir le plateau de jeu avec la présence des caisses surprises et des ennemis.

En plus des fonctionnalités mises en place lors de l'itération précédente, l'utilisateur doit pouvoir:

- Si il tombe sur une caisse surprise, pouvoir connaître son contenu. Les bonus sont alors appliqués dans la limite des caractéristiques des personnages.
- Savoir si il tombe sur un ennemi, et en connaître les détails.

Pour la mise en place du plateau de jeu, on choisira de positionner sur le plateau:

- 24 ennemis:
 - 4 Dragons (case 45, 52, 56 et 62)
 - 10 Sorciers (case 10, 20, 25, 32, 35, 36, 37, 40, 44 et 47)
 - 10 gobelins(case 3, 6, 9, 12, 15, 18, 21, 24, 27 et 30)
- 26 caisses surprises:
 - 5 arcs (case 2, 11, 14, 19 et 26)
 - 3 massues (case 5, 22, 38)
 - 2 épées (case 42 et 53)
 - 5 sorts "éclair" (case 1, 4, 8, 17 et 23)
 - 2 sorts "boules de feu" (case 48 et 49)
 - 5 potions mineures (case 7, 13, 28, 29 et 33)
 - 3 potions standards (case 31, 39, 43)
 - 1 grande potion (case 41)

Ces données sont un point de départ, n'hésitez pas à les adapter selon la jouabilité.

Itération facultative

Dans cette itération, les cases ne sont plus placées de manières fixes mais de manière aléatoire. Attention cependant le nombre d'ennemis et d'équipement doit rester le même que précédemment.

Itération 3

Maintenant que le plateau de jeu est complètement en place, vous devez dans cette itération ajouter la gestion des combats (selon les règles énoncées plus haut) si le joueur tombe sur un ennemi en se déplaçant sur le plateau.

Vous devez également gérer la fin de partie si le joueur perd tous ses points de vie ou si il arrive victorieusement au bout du plateau.

Itération 4

Dans cette itération, un mode debug doit être implémenté. Ce dernier permet de jouer un scénario prédéfini. Cela vous permettra de tester plus simplement votre jeu en retirant la partie aléatoire de vos tests(lancer de dé).

Votre moteur de jeu doit alors être capable de lire un fichier de scénario. Ce dernier défini au format CSV les nombres tirées au dé à chaque tour.

Exemple:

2,6,4

Le joueur a tiré 2 au premier tour, 6 au deuxième et 4 au troisième.

Note:

Ce mode de jeu doit pouvoir s'activer et se désactiver facilement. Vous devez utiliser les arguments de la boucle main() du client console afin de récupérer le chemin du fichier .csv, le lire, et transmettre les données au moteur du jeu qui les utilisera pour gérer l'avancement du personnage. **Attention**, vous devez réaliser cette opération sans modifier l'API fournie. En revanche, vous allez devoir rajouter du code dans le client console.

Itération 5

Le plateau de jeu doit maintenant être personnalisable par le joueur. Pour ce faire ce dernier doit fournir un fichier au format JSON. Le jeu doit alors être capable de lire ce fichier, de créer le plateau correspondant et de pouvoir dérouler une partie sur ce dernier. On pourra scanner un répertoire fixe contenant plusieurs fichiers représentant des plateaux de jeu et les proposer au joueur.

À vous de définir le formalisme du JSON afin d'avoir toutes les données nécessaires pour initialiser le plateau de jeu.

Nous vous conseillons l'utilisation de la librairie GSON : <https://github.com/google/gson>

Itération 6

Nous souhaitons maintenant que la liste des personnages disponibles et leurs attributs soient stockés dans une base de données. Au démarrage du jeu, vous devez récupérer toutes les informations nécessaires en base de données pour instancier les personnages que le joueur pourra choisir en début de partie.

Cette itération nécessite un travail préalable sur le mini projet **Connection à une base de donnée** afin d'appréhender les concepts de connection à une base de données.

Itération 7

En plus des personnages, nous souhaitons maintenant que les plateaux générés par fichier soit également stockés dans la base de données. Au démarrage du jeu les plateaux sont chargés puis affichés et le joueur peut sélectionner celui sur lequel il souhaite jouer.

Itération 8

Pour cette nouvelle itération, veuillez vous référer au sujet annexe 05 bis Partie Web.

Itération facultative

Le jeu, jusqu'à maintenant est plutôt linéaire. En effet le joueur ne dispose pas vraiment de choix. Dans cette itération nous proposons donc de changer quelques règles.

Nouvelle règle des combats

Les ennemis ne fuient maintenant plus le combat, ils se battent jusqu'à la mort.

Ainsi si le joueur frappe un monstre et que les points de vie de ce dernier sont supérieurs à 0, il inflige alors ces points de d'attaque au joueur.

Le joueur peut alors choisir de l'attaquer de nouveau ou de fuir:

- Si le joueur fuit il recule alors de deux cases
- S'il décide d'attaquer de nouveau un nouveau tour de combat est lancé

Gestion de l'inventaire

Le personnage possède maintenant un inventaire ne pouvant contenir que deux sorts pour les magiciens et trois armes pour les guerriers. Ainsi lorsque le personnage arrive sur une case contenant un équipement il peut décider de ramasser ou non cet objet.

Nouveaux ennemis

- Les orcs : Ces ennemis veulent prouver leur vaillance au combat, mais craignent la magie. Ainsi lorsque le personnage rencontre un orc, le combat n'est lancé que s'il est guerrier

Le jeu "Donjons et Dragons"

- Le mauvais esprit: Cet ennemi cherche à voler la magie de ces adversaires. Il n'a par contre aucun intérêt pour les créatures dénuées de magie. Ainsi le combat n'est lancé avec un esprit que si le personnage est un magicien.

Bien entendu, les exemples ci dessus ne sont pas limitatifs, il vous est possible d'implémenter les règles que vous souhaitez pour personnaliser votre jeu.

Livrables

- Code des différentes itérations demandées. A déposer sur git.
- Les diagrammes UML de classes et de Use case.
- Glossaire enrichi des différentes syntaxes utilisées. A déposer sous le nom **glossaire**, dans votre répertoire individuel sous **java→livrables**