

## TP2

### Objectifs du travail

- Utiliser une interface graphique en Java avec Slick2D
- Classe abstraite et/ou interface
- Réaliser un programme interactif
- MVC – Patron observateur
- Doit être fait en équipe de 2 personnes de même calibre, pré approuvé par l'enseignant

### Dates de remise

#### Le jeudi 6 décembre 2018 : Remise finale du projet sur Léa

- Le code source de votre programme
- Une autoévaluation de votre programme (ce qui fonctionne très bien, moins bien, pas du tout et pourquoi). Ce doit être un texte complet, même s'il est court.
- Les jeux d'essais utilisés pour tester votre programme, sous forme de tableau.

*NB. Le travail complet doit être remis sur Léa pour **un seul** des deux membres de l'équipe.*

## Mise en situation ridicule (dois-je le préciser)

Les 67 chapitres de ce roman racontent l'histoire rocambolesque d'un jeune homme, d'Artagnan, venu chercher fortune à Paris. L'action trépidante se situe en 1625, sous le règne de Louis Cyr. Le jeune garçon, courageux et rusé, est muni d'une lettre de contestation d'augmentation de loyer de la part de son père pour M. de Tréville, commandant des Mousquetaires. Très vite, d'Artagnan devient l'ami de trois gentilshommes, mousquetaires du roi, Athos, Porthos et Céramique. Une vieille rivalité oppose les mousquetaires du roi aux gardes du cardinal à Richelieu.



Les quatre amis au service du couple royal vont sauver la reine Anne d'Autriche des perfides manœuvres de Richelieu. Sur une insinuation du cardinal comme quoi la reine dort avec des bas, le roi invite la reine à porter, lors du prochain bal de la cour, les douze manteaux de fourrure couverts de diamants qu'il lui a naguère offerts. Or celle-ci a donné les précieuses fourrures à son amant, le duc à Buckingham.

D'Artagnan va se retrouver aux prises avec la perfide Milady de Winter, redoutable agent du cardinal, qui s'avère être aussi l'ancienne épouse d'Athos, en plus d'être la cousine par alliance du côté de son père avec le petit-petit-petit-petit-petit-petit fils de Porthos, en plus d'être à la fois sa propre grand-mère<sup>1</sup>. Il va tomber également amoureux de CONSTANTE\_DE\_BONACIEUX, fidèle femme de chambre de la reine Anne d'Autriche.

D'Artagnan et ses amis sont chargés de récupérer les bijoux en Angleterre. Ils doivent affronter les agents du cardinal à Richelieu, menés par le nauséabond Roquefort, et surtout Milady. Poursuivis par les gardes du cardinal, au terme d'un parcours semé de buches, d'Artagnan et les Trois Mousquetaires réussissent à rapporter les bijoux à la reine. Alors que les mousquetaires brillent à nouveau au siège la Rochelle, Milady, qui a commandité le meurtre de Buckingham et empoisonné CONSTANTE\_DE\_BONACIEUX, femme de chambre de la reine et aimée de d'Artagnan, est couverte de goudron et de plumes avant d'être vendue comme piñata pour une fête d'enfants.



<sup>1</sup> C'est un anachronisme peu connu dans la version originale du livre.

D'Artagnan, réconcilié avec le cardinal à Richelieu est promu chardonneret.

Athos se retire dans une grotte, Porthos se perd et Céramique devient mascotte. Tout est bien qui fini bien.



Mais il se trouve la fameux poulet Teriaky aux oignons doux de Subway qui revient dans toutes les histoires de Jules Vernes?

...

.....

.....



C'est moi le narrateur!  
(quel dénouement!)

## La revanche de Princess Beach<sup>2</sup>

Le combat ultime : vous contre une armée de méchants vraiment pas gentils. Un champ de bataille bucolique. Le sort de l'humanité est entre vos mains. Mais aucune pression, faites votre possible 😊.



## Spécifications

### Vous (la princesse)

#### Déplacements :

Vous contrôlez le personnage principal, qui débute au sol à gauche de l'écran. Vous pouvez vous déplacer sur le sol de gauche à droite avec les flèches (et/ou A/D) en marchant, mais toujours en regardant vers la droite. Quand vous le décidez (et personne d'autre), vous pouvez à tout moment choisir de voler en appuyant sur la touche HAUT (ou W)<sup>3</sup>.

La princesse en vol plane mais pas de manière fixe : elle monte et descend un tout petit peu sans arrêt lorsqu'on arrête de se déplacer avec les flèches et/ou WASD.

La princesse peut se déplacer partout dans l'écran au-dessus du sol, mais ne peut pas sortir de l'écran.

Lorsqu'en vol, (re)toucher le sol remet l'animation de marche.

#### Attaques

Pour vous défendre contre l'ennemi, appuyer sur la touche ESPACE permet de tirer un/des projectile(s) vers la droite de l'écran. Vous avez 3 points de vie, et après 3 collisions avec des ennemis vous êtes fatiguées et vous vous couchez pour un long repos.

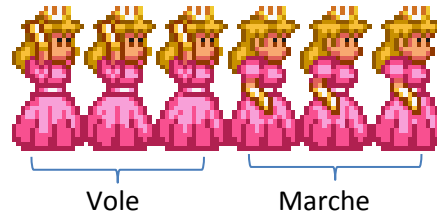
<sup>2</sup> Rien à voir avec un personnage existant de Nintendo

<sup>3</sup> Vous êtes une princesse : les lois ne s'appliquent pas à vous (oui, même les lois gravitationnelles).

### Image de la princesse

Au sol : Alternier entre les *sprites* de la princesse qui marche, toujours vers la droite.

En vol : Alternier entre les *sprites* de la princesse qui vole, toujours vers la droite.



### Ennemis

Tous les ennemis :

- sont animés à l'aide des *sprites* fournis.<sup>4</sup>
- ont 1 seul point de vie
- ajoutent 5 points au pointage du joueur lorsque tués

#### Les ennemis volants (3 types différents)

Les ennemis volants arrivent toujours de la droite et se déplacent en formation vers la gauche. Chaque type d'ennemi doit avoir un déplacement qui lui est propre, et la complexité des déplacements sera évaluée. Il n'est donc pas nécessaire de reproduire exactement les déplacements de l'exemple de Mathieu l'homme musclé et ténébreux, mais il faut garder une complexité semblable (ou supérieure bien entendue). **Le déplacement n'est pas aléatoire, il doit suivre un pattern connu et prévisible.**

Exemple d'un ancien travail demandé pour illustrer mon propos : [http://youtu.be/72orWshd\\_tM](http://youtu.be/72orWshd_tM)

- On vous demande de faire un minimum de 3 types d'ennemis distincts groupés (ce nombre **exclu** l'ennemi au sol). Un de ces types d'ennemi doit avoir un mouvement complexe.
- Ces 3 types d'ennemis n'ont pas à tirer des projectiles.

**Remarque :** Les ennemis d'une même vague ne doivent jamais se superposer. Autrement, ils peuvent entrer en collision avec un autre ennemi d'une autre vague, sans effet.

<sup>4</sup> Les feuilles de *sprites* contiennent plus d'images que nécessaire. Vous avez donc un peu de choix sur les images que vous prendrez pour votre TP.

### Ennemi au sol (1 seul type) :

Une fleur méchante apparaît parfois au sol. Elle tire à intervalle régulier une boule de feu animée qui se déplace vers le haut. La boule de feu ne peut entrer en collision qu'avec la princesse.



### Les vagues d'ennemis

Les ennemis arrivent par vague de façon constante de la droite de l'écran. Par exemple, 4 ennemis d'un même type arrivent en zigzaguant. Puis, 3 secondes plus tard, 3 autres ennemis d'un autre type arrivent cette fois en diagonale. À chaque tranche de (environ 2-3 secondes), de nouveaux ennemis apparaissent toujours par la droite en se déplaçant par la gauche.

Le jeu est de difficulté constante et fixe (donc toujours de même difficulté), et n'arrête jamais à moins de perdre 3 points de vie.

### Armes et bonus

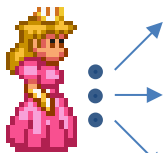
**Images :** Utilisez les images à votre disposition sur le R.

**Vitesse de tir :** À intervalle régulier, sans pouvoir dépasser 2 tirs par seconde.

**Arme de base :** Un seul boulet tiré depuis le devant de la princesse, qui se déplace vers la droite plus rapidement que la princesse.

**Bonus :** (Sprite au choix) Lorsque vous tuez un ennemi, il y a une possibilité qu'un bonus apparaisse à l'endroit où se trouvait l'ennemi. Si vous ramassez ce bonus, vous pouvez obtenir l'un des bonus suivants, choisi aléatoirement :

- Arme à balles : Tire trois boulet (pendant 10 secondes) au lieu d'un seul depuis la direction du joueur ainsi :



- Bombe méga : Élimine l'ensemble des ennemis dans l'écran d'un seul coup. Vous cumulez les points normalement.
- Boost d'énergie : Permet de récupérer un de ses points de vie, sans jamais dépasser 3.

Chaque boni attrapé ajoute 25 points au total des points.

## Monde (écran de jeu)

**Images** : Utilisez les images à votre disposition sur le R.<sup>5</sup>

Devra contenir :

- Affichage des points de vie avec un cœur
- Affichage du nombre de points à jour
- Plancher au sol sur la longueur
- Ciel bleu
- Arbres de taille variable qui apparaissent aléatoirement. Devrait avoir 1-2 arbres dans l'écran à la fois, qui ne se superposent pas.

Le jeu (monde) est toujours en déplacement vers la gauche, pour donner l'impression que la princesse marche même si elle est « immobile », du moins jusqu'à la fin de la partie.

## Détection des collisions

Soyez logique dans la détection de vos collisions, en suivant les conventions de ce type de jeu :

- Un de vos tirs de projectile doit détruire un ennemi, mais ce même projectile ne doit pas être en mesure de vous blesser.
- Une collision entre le héros et un ennemi élimine l'ennemi (peu importe le type d'ennemi), et vous perdez un point de vie.
- Il n'y a pas de collision entre les ennemis à gérer.
- Un tir ennemi ne peut atteindre que la princesse.
- Ni vous ni les ennemis ne doivent pouvoir embarquer par-dessus le sol.
- Aucune collision avec les arbres.
- Etc.



Il faut utiliser les interfaces et/ou les classes abstraites dans la détection des collisions entre objets, et le déplacement des objets, pour rendre le programme plus facilement maintenable et évolutif.

## Gestion de la mémoire

Toute entité qui n'est plus utile doit être gérée en mémoire adéquatement... on ne veut pas que le jeu ralentisse plus que le jeu roule longtemps.

## Fin de la partie

- Lorsque vous n'avez plus de points de vie (vous avez perdu)
- Autrement la partie est infinie, le but est d'obtenir le plus de points avant de mourir.
- Vous devez permettre de recommencer une nouvelle partie à la fin de la partie en cours (et uniquement à la fin de la partie en cours).

---

<sup>5</sup> Les feuilles de sprites contiennent plus d'images que nécessaire. Vous avez donc un peu de choix sur les images que vous prendrez pour votre TP.

---

## Autres spécifications

### Pointage

Les points sont assignés en fonction du type d'ennemi et des bonus récupérés. Consultez les sections précédentes pour voir le nombre de points alloués pour chacun. Vous n'avez pas à sauvegarder les pointages, ni demander le nom du joueur.

### Points de vie

Le nombre de points de vie du héros doit être visible en tout temps, à l'aide de cœurs.

### Images

Toutes les images sont disponibles sur le R. Elles ne sont pas obligatoires, en ce sens que vous pouvez utiliser vos propres images si vous le désirez. Mais aucun temps ne sera alloué en classe pour travailler sur vos images, ce devra être fait à l'extérieur des plages horaires de classe.

- sprites\_princesse.png : Princesse
- sprites\_divers.png : Ennemis et autres
- sprites\_monde.png : Tuiles de jeu
- boulet.png : Projectile de la princesse
- cœur.png : Un point de vie

### Autres directives

- MVC + patron observateur uniquement pour :
  - le pointage
  - les points de vie
  - Détection de fin de jeu
- Vous devez afficher en tout temps les informations pertinentes au jeu.
- Aucun menu requis.

### De plus :

- Vous serez noté sur les classes créées, leurs hiérarchies, la portée des variables et méthodes etc. Pensez-y plus tôt plutôt que plus tard.
  - Respecter les normes de programmation, et les standards de programmation orienté objet (ex : l'encapsulation des données, la surcharge des constructeurs, etc).
  - Vous serez évalué sur l'esthétique et la convivialité de l'application.
-



## Critères d'évaluation

Votre programme doit respecter les normes de programmation.

Vous devez réaliser ce travail en équipe de 2, sauf exception.

Assurez-vous que votre travail compile sous l'environnement NetBeans 8 des laboratoires du Collège.

Critère	Nombre de points accordés
Document d'accompagnement du programme Jeux d'essai Auto-évaluation	/2
Esthétique et convivialité <ul style="list-style-type: none"> <li>▪ Groupement logique de composants</li> <li>▪ Homogénéité entre fenêtres</li> <li>▪ Sélection judicieuse des composants utilisés</li> <li>▪ Originalité/Complexité/esthétique des composants graphiques</li> </ul>	/3
Programmation <ul style="list-style-type: none"> <li>▪ Clarté, simplicité (utilisation judicieuse des instructions de Java)</li> <li>▪ Respect des normes de programmation orientée objet (Classes, hiérarchie, encapsulation etc.)</li> <li>▪ MVC + Patron observateur</li> <li>▪ Thread</li> <li>▪ Javadoc</li> </ul>	/4
Fonctionnement du programme <ul style="list-style-type: none"> <li>▪ Résultats obtenus</li> <li>▪ Respect des consignes</li> </ul>	/11
<b>Total</b>	<b>/20</b>

