

1 Modalités

Le projet doit être écrit en *JavaFX*, et réalisé en binôme (éventuellement, en monôme). Les soutenances auront lieu en mai, la date précise sera communiquée ultérieurement. Pendant la soutenance, les membres d'un binôme devront chacun montrer leur maîtrise de la totalité du code.

Vous devrez déposer les projets sur Moodle sous la forme d'une archive se décompressant en un dossier portant les noms de votre binôme. Le dossier doit contenir le sous-dossiers correspondant au nom de votre package. L'archive doit contenir toutes les sources java et les fichiers fxml. (Si vous utilisez NetBeans ou IntelliJ Idea il suffit de compresser le répertoire qui se trouve dans le répertoire `src` du projet.)

La date limite du rendu sera la veille des soutenances. Seulement les archives `zip`, `tar`, `tar.gz`, `tbz2` seront acceptés (**pas de rar**).

La qualité du code, sa conception objet, ainsi qu'un usage approprié des fonctionnalités de JavaFX, auront un impact important sur la note. En particulier, partout où cela est possible nous préférons :

- l'utilisation de fxml et de fonctions « callback » déclarées dans fxml, et définies dans les contrôleurs à la place de handlers,
- des « bindings » plutôt que des `ChangeListener`s.

2 Jeu démons et pistolero

Le but du projet est de créer un jeu (genre shoot them up) où vous incarnez un pistolero chargé de tuer les démons.

Le jeu se joue sur une arène où évoluent les personnages. Vous dirigez le pistolero à l'aide du clavier. Prévoir les mouvements verticales et horizontales pour le pistolero et pour les démons, et peut-être, pourquoi pas, sur les diagonales, on peut imaginer que les 4 touches "flèches" sur le clavier provoquent les déplacements verticales et horizontales, mais modifiées avec CTRL elle permettent les déplacements sur les deux diagonales.

Les règles de jeux sont simples. Pistolero peut tirer une balle d'argent toujours dans la direction en face de lui. Si la balle touche un démon celui-ci est tué sur le champs. On pourra tirer en utilisant le clavier mais aussi avec un bouton de la souris.

Le trajet de la balle doit être visible de telle sorte qu'on puisse voir le trajectoire, et le démon touché par la balle devait changer brièvement forme avant de disparaître (il doit "saigné").

Le démon peut tuer (manger) le pistolero en le rattrapant (la collision avec le pistolero). Cela termine le jeu. Les démons changent la direction de mouvement s'il heurtent un obstacle (mettre des obstacles sur l'arène de jeu) ou spontanément (mais pas trop souvent et pas tous les démons au même moment). Utiliser un générateur de nombres aléatoires.

Avec MVC (modèle vue contrôleur) bien structuré on peut essayer plusieurs règles de mouvement de démons, par exemple

- les démons changent la direction de façon aléatoires, ils sont comme aveugles,
- ou bien ils ont une tendance de suivre le pistolero et se diriger vers lui, ils ne sont pas aveugles mais veulent bien sa peau et veulent le rattraper.
- on peut même penser de mettre les démons intelligents qui essaient d'éviter la ligne de tire.

Pour changer un peu par rapport à d'autres jeux de ce type on demande implementer les démons de deux sexes, mâles et femelles. Si deux démons mâles entrent en collision un seul survivra l'autre sera mangé, par contre une rencontre d'un démon mâle et un démon femelle fait naître un nouveau démon (choisir les icônes différentes pour les mâles et les femelles) .

Slider permettra de changer les vitesses de mouvement de personnages, on peut même prévoir deux **Sliders**, un pour la vitesse de démons et un autre pour la vitesse de pistolero.

Au début du jeu on donnera à l'utilisateur la possibilité de changer l'affectation de touches, il pourra choisir les touches qui lui conviennent pour les mouvement du pistolero. Le joueur pourra choisir le nombre de balles à sa disposition (le nombre infini c'est une de possibilité). Peut-être aussi il pourra choisir l'icône pour le pistolero.

Prévoir :

1. l'affichage en temps réel du nombre de démons qui restent à éliminer,
2. l'affichage du nombre de démons éliminés,
3. l'affichage de nombre de balles restants,
4. l'arrêt momentané de jeu avec le clavier (pour repartir par la suite),
5. sauvegarde de l'état de jeu avec la possibilité de charger la sauvegarde,
6. les positions des obstacles devaient changer d'une partie à l'autre (utiliser les générateur de nombre aléatoires pour trouver les positions des obstacles, les balles ne peuvent pas traverser les obstacles (le plus simple à implémenter ce sont les obstacles qui absorbent les balles mais si vous arrivez à implémenter les balles qui rebondissent sur l'obstacle cela devient plus intéressant).

Ceux qui veulent faire plus que le minimum peuvent penser à d'autres options comme :

- l'arène du jeu peut être plus grande que la partie visible, intuitivement on voit juste une fenêtre rectangulaire sur l'arène, et cette fenêtre suit le mouvement de pistolero, si le pistolero s'approche de bord de la partie visible alors une nouvelle partie de l'arène se dévoile devant lui,
- l'arène 3D ?
- toute astuce ou règle supplémentaires qui améliorent la « jouabilité ».