

Evaluation 2

Le but de cette évaluation est de corriger un code existant et d'en étendre les fonctionnalités (en réarchitecturant, nettoyant... au besoin) en utilisant les différentes notions vues en cours.

Threshold 0

Le projet fourni s'inspire de *Space Invaders* s'exécutant dans une console Windows. Voici les différents éléments qui le compose :

- *Input* est une interface gérant les différentes actions possibles par le joueur (*Left, Right, Fire*).
- *RndInput* est une implémentation d'*Input* qui les gère de manière aléatoire.
- *GameObject* est l'abstraction d'un élément du jeu dans une scène
- *PlayField* est la scène en question regroupant tous les *GameObject*

- *Alien* est une instance de *GameObject* qui répond aux caractéristiques suivantes :
 - Se déplace à vitesse constante horizontalement jusqu'au bord de la scène avant de se déplacer d'un cran verticalement et recommencer à se déplacer horizontalement dans la direction opposée
 - Si un *Alien* atteint le bas de la scène, la partie est terminée
 - Tire des *AliensLasers* de manière aléatoire
 - S'il est touché par un *PlayerLaser*, l'*Alien* est détruit

- *PlayerShip* est une instance de *GameObject* qui répond aux caractéristiques suivantes :
 - Répond aux *Inputs*
 - Ne peut se déplacer qu'horizontalement

- *Laser* est une instance de *GameObject* qui répond aux caractéristiques suivantes :
 - Peut être issu d'un *Alien* (auquel cas il s'agira d'un *AlienLaser*) ou de *PlayerShip* (auquel cas il s'agira d'un *PlayLaser*)
 - Une collision entre deux lasers détruira les deux
 - Une collision avec un *Alien* ou un *PlayerShip* détruira le laser et fera perdre de la vie à l'autre *GameObject*
 - Ne se déplace que verticalement

Le but de ce Threshold est de réparer le projet existant pour qu'il fasse fonctionner les différents éléments mentionnés plus haut.

Threshold 1

A ce stade, le projet fourni doit fonctionner et il est temps l'améliorer. Voici donc une liste des axes d'améliorations et fonctionnalités à implémenter :

- Réarchitecturez le code existant. Comme vous avez pu le constater, il est sale, peu efficace, redondant... Rendez le plus clair, lisible... en faisant ce qui est nécessaire tout en gardant l'esprit du code de départ. **Une copie de *MyIsaac* n'est pas acceptable dans ce cas.**
- Changez de Renderer vers une version SFML (de même que les Inputs).
- Ajoutez la gestion des Inputs clavier
- Rendre le jeu totalement configurable (taille, nombre d'ennemi, seed, nombre de tir...). Le choix technique pour répondre à ces points reste votre.
- Ajoutez des Rochers lors de la génération du terrain bloquant tous lasers
- Améliorez les *Aliens* (plus de vie, changement de vitesse suivant le nombre d'ennemis restant, forme améliorée...)
- Optimisation en général

Threshold 2

Voici d'autres fonctionnalités que vous pouvez implémenter. A partir de ce point, si vous avez des idées non listées faites-vous plaisir :

- Changer le nom de la solution et du projet maintenant qu'il est tout propre
- Pour décider le renderer à utiliser
- Système de score (leaderboard local...)
- Amélioration du renderer SFML, du son...
- ...

Règles à suivre & informations notation

- Votre projet doit fonctionner sur Visual Studio 2019 (x64, Debug et Release). La version Release devant bien entendue être optimisée via Visual Studio. Vous pouvez travailler de Visual Studio sur la version que vous voulez mais votre rendu doit compiler sous Visual 2019 !
- La qualité du rendu fait une nouvelle fois partie de la notation. Cela comprend :
 - La présence de sous-dossier dans votre rendu final
 - La présence de certains fichiers (AUTHORS, README) et leur bon format
 - AUTHORS : Doit figurer votre nom et prénom. Rien de plus
 - README : Il s'agit de votre carnet de bord. Toutes vos actions doivent y être consignées (Toutes les erreurs trouvées au T0 doivent y être, les modifications faites, les choix que vous avez fait, les features implémentées... Il doit être le plus complet possible)
 - Le nettoyage de vos projets (fichiers temporaire...)
- Le rendu doit se faire avant le 06/03/2023 – 13:59:59. Deux modes de rendu sont autorisés :
 - Via une archive transmise par mail (maxime.lacombe@ynov.com) ou en message privé via Team. Dans ce cas, c'est un mail/message par personne. Soyez donc sûr de votre rendu au moment d'envoyer
 - Via git. Dans ce cas, le rendu doit se faire sur une branche à part « Submission ». Communiquez-moi le lien par mail/Teams au préalable. S'il s'agit d'un repo perso, vous pouvez ajouter AArzop. La configuration du repository, les droits... reposent sur vous !
- La note finale se compose de trois points :
 - La qualité du rendu comme énoncé plus haut
 - L'exécution :
 - Sans jeter un œil au code et simplement en lançant votre programme et en y jouant, une note sera faite en prenant en compte les bugs rencontrés, les features implémentées...
 - La qualité du code
 - Est pris en compte : la qualité du code, l'architecture, l'optimisation, la réflexion, les choix techniques...