Exercice 6 : Héritage avec Galaga

Voici une petite app graphique où un sprite d'un jeu classique, Galaga, descend dans l'écran et une fois en bas, est téléporté en haut de l'écran, un peu comme ça se passe dans la version originale. http://game-oldies.com/play-online/galaga-demons-of-death-nintendo-nes# mais bon, je pense que vous connaissez vaguement le concept de ce jeu…

0-Prémisses

Vous pouvez observer le code récupéré. Il s'y trouve du code impliquant l'animation de spriteSheet. Vous n'aurez pas à y toucher, sauf pour changer le niveau de sprite pour effectivement changer le sprite d'une sous-classe. Pour ce qui est des techniques d'animation de sprite, nous y reviendrons dans les prochains cours mais vous pouvez observer le code quand même pour le moment.

Le sprite du Galaga de départ est celui du Butterfly.

1-Constructeurs

A- Le constructeur de la classe Galaga est définitivement à revoir. Certaines lignes de code n'y ont absolument pas leur place. Déterminez ces lignes de code et placez-les dans une méthode à part qui pourra être appelée subséquemment.

B- Une fois cela fait, utilisez les initialiseurs de la meilleure façon avec le constructeur.

C- Une fois le constructeur ajusté avec les initialiseurs, regardez les attributs de la classe Galaga. Est-ce que certains pourraient devenir des constantes? Si oui, ajustez ces variables en conséquence.

2-Héritage

Vous aurez à implémenter une hiérarchie d'héritage pour les Galagas.

A- Rendez Galaga abstrait pas la méthode deplacement. Par soucis de simplicité, remplacez private par protected dans Galaga.h

B- Créez la classe Butterfly. Celle-ci n'ajoute rien vraiment à la classe galaga de base, à part utiliser ce qui est déjà là et la rendre concrète.

C- Créez la classe Wasp qui héritera de Butterfly. La seule différence avec Butterfly (outre les sprites) est la surcharge de la méthode Déplacement, où on ajoute un mouvement d'aller-retour de 50 pixels de chaque côté en x. **Contrainte :** dans la méthode de

déplacement de Wasp, vous devez vous servir de la méthode de déplacement de Butterfly.

D- Créez la classe Catcher qui héritera de Butterfly. Ici son déplacement fait une orbite de 100 pixels de distance avec la position qu'aurait théoriquement un Butterfly équivalent. **Conseil :** Ceci est un petit défi algorithmique. Utilisez le déplacement de base du Galaga pour vous assurer d'avoir une classe fonctionnelle et travaillez à ajustez le mouvement correctement par la suite.

E- Créez un petit tableau de pointeurs de Galagas de taille 3 et placez-y un exemplaire de chaque galaga. Utilisez le tableau dans la game pour afficher les trois galagas.

3-Optimisation

Si vous avez simplement utilisé la classe Galaga telle quelle, pour créer vos trois sous-classes, vous avez certainement remarqué que vous chargez trois fois la même spritesheet. Ceci est une perte de ressource mémoire et de temps cpu inadmissible. Faites en sorte que la texture devienne un attribut static (qui doit rester protected) et ajustez le reste du code en conséquence.

Remettre le projet Exercice6, mis à jour sur LEA avant le cours 13.

Note 1 : Vous pouvez consulter le petit exécutable qui vient avec le projet pour avoir une image du résultat attendu.

Note 2 : Vous avez avec le projet d'autres sprites, des sons, et une font. Tout ce qu'il faut pour vous amuser à aller plus loin si vous voulez.