

Exercice 8 : Design pattern avec Galaga

Reprenez le corrigé de l'exercice 7. De là...

1-Fabrique statique

- a- Remplacez le tableau statique de Galagas par un vecteur de pointeurs de Galagas.
- b- Laissez le vecteur vide (donc pour le moment pas de Galagas qui affichent)
- c- Créez un Spawner, il s'agit d'un simple CircleShape de 25 de rayon, avec un trait d'épaisseur de 3, et un fond transparent. Il se déplace de 3 pixels par frame en x et s'il arrive à 100 pixels ou moins d'un rebord de fenêtre, il change de direction. Il ne fait rien d'autre qu'afficher, en fait c'est un repère visuel.
- d- Implémentez une fabrique statique pour les Galagas. Vous allez vous baser sur le modèle de départ, avec les éléments supplémentaires suivants.
 - a. L'enum de Galagas dans game devra à présent être situé dans la fabrique statique. Bien entendu, il sera lui aussi static.
 - b. Chaque attribut du constructeur de Galaga doit être représenté par une variable statique et privée appartenant à notre fabrique, sauf le niveau. Vous aurez donc.

private:

```
static Vector2f position;  
static int vitesse;  
static int nbrNiveaux;  
static int nbrAnims  
static RenderWindow* renderWindow;
```

- c. Une méthode void chargerData avec tous les paramètres ci-dessus en entrée pour les initialiser, SAUF POSITION.
- d. Une méthode setPosition pour « placer » la fabrique statique sur le spawner au moment où nous en aurons besoin. Un Galaga sera alors spawné à l'endroit même du cercle. Notez bien que cette méthode est utile. Dans un niveau de jeu nous pouvons avoir plusieurs spawners, mais une seule fabrique statique qui se "déplace" de spawner en spawner quand l'un d'eux doit générer un objet de jeu.

- e. La méthode de création a donc à présent tous ses paramètres; même si le constructeur de Galaga en a plusieurs, ils sont tous là, sauf le niveau, qui est décidé directement par le Galaga sélectionné (un Wasp aura bien entendu le niveau Wasp dans l'enum).
- f. Ainsi, c'est la fabrique statique qui ajoute un Galaga dans notre vecteur. Vous pouvez appeler la méthode "ajustementsVisuels" dans la fabrique avant que celle-ci ne retourne le Galaga créée.
- g. Implémentez une méthode Unload, qui va nettoyer le vecteur

2-Singleton

Soit la classe Player.h qui se trouve avec cet énoncé.

- A- Ajoutez-là au projet et complétez son implémentation en lui créant un .cpp approprié et intégrez-le dans le main, afin d'obtenir le comportement du démo.
- B- Une fois votre player fonctionnel, convertissez-le en Singleton. Notez bien le commentaire sur le constructeur dans le code pour pouvoir l'implémentez correctement.

3-Poids-mouche

Attention numéro bonus : +5 points si vous le réalisez (Vous pouvez donc avoir 15/10).

ET

Sauvegardez votre projet avant d'entreprendre ce numéro.

Convertissez le Galaga en poids-mouche (il n'hérite donc plus de sprite). Votre projet ne devrait avoir qu'une texture et un sprite (contenant la spritesheet des galagas), et ce sprite peut-être statique. ATTENTION. Du fait de l'animation de sprite, il vous faudra bien réfléchir à comment procéder.