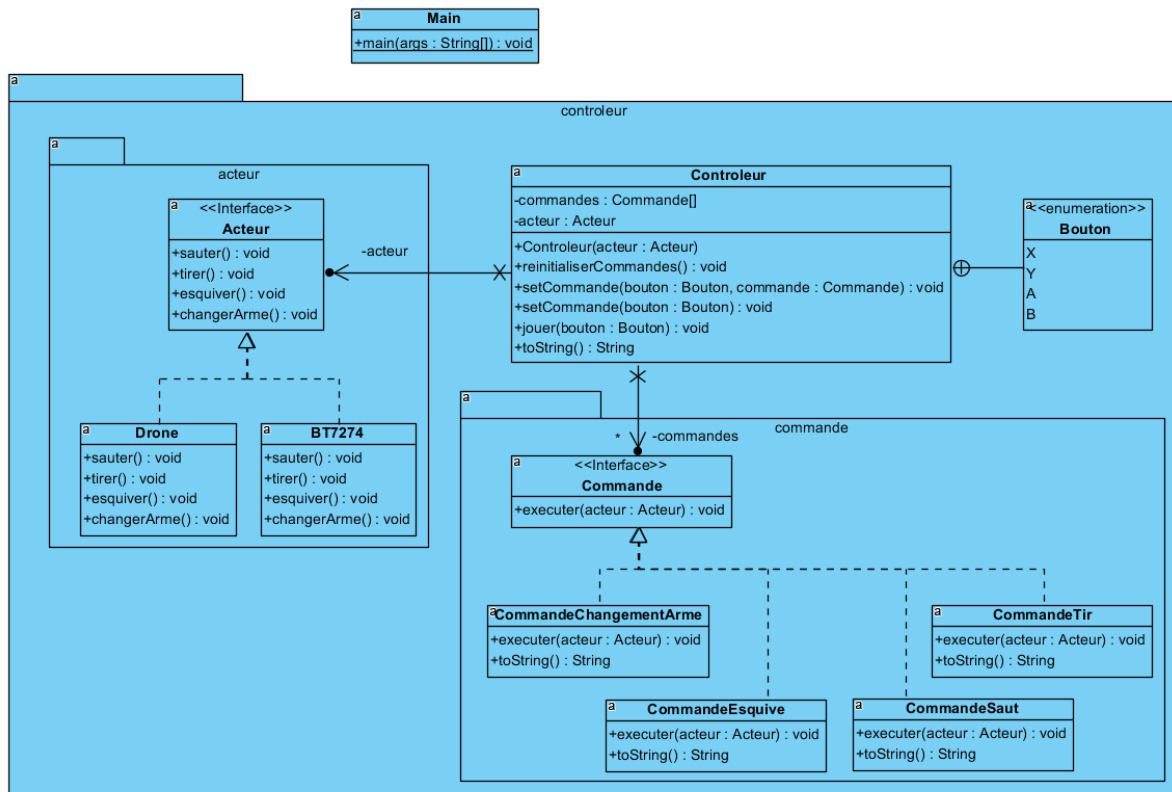


Patron #8 : Commande



Mise en contexte

Dans cet exemple de jeu vidéo, nous allons programmer un contrôleur de jeu (ex : une manette de Xbox) qui pourra être « personnalisable ». Nous allons utiliser le patron commande car les paramètres à « changer » seront des classes qui définissent des actions à réaliser.

De plus, on remarque l'ajout d'un acteur qui permet d'introduire une couche supplémentaire au modèle. On pourra donc personnaliser les commandes du contrôleur en fonction de l'état du jeu mais aussi en fonction de l'acteur du jeu qui sera contrôlé.



Contraintes

- Chaque acteur implémente l'interface « Acteur », voici un exemple de code pour la fonction « Drone.tirer »

```
@Override
public void tirer() {
    System.out.println("DRONE-TIRER");
}
```

- Contrôleur
 - La méthode « réinitialiserCommandes » (aussi appelé par le constructeur, associe les commandes suivantes :
 - X → Sauter
 - Y → Tirer
 - A → Esquiver
 - B → Changer d'arme
 - La méthode « setCommande » avec un seul paramètre remet la commande à *null*
 - La méthode « jouer » appelle la méthode « executer » de la commande correspondante ou pas (attention au null ;-)
- Commande
 - La méthode abstraite « executer » est implémenter dans chaque classe enfant. Par exemple, l'appel de la méthode « executer » de la classe « CommandeTir » lance l'exécution de la méthode « tirer » de l'acteur reçu en paramètre.
- Utiliser le code suivant pour la procédure « main »

```
Drone drone = new Drone();
BT7274 bt = new BT7274();
Contrôleur contrôleur = new Contrôleur(bt);

System.out.println("\nMODE DE JEU : NORMAL");

System.out.println(contrôleur);

contrôleur.jouer(Contrôleur.Bouton.X);
contrôleur.jouer(Contrôleur.Bouton.Y);
contrôleur.jouer(Contrôleur.Bouton.A);
contrôleur.jouer(Contrôleur.Bouton.B);

System.out.println("\nMODE DE JEU : AVENTURE");

contrôleur.setCommande(Contrôleur.Bouton.Y);
contrôleur.setCommande(Contrôleur.Bouton.B, new CommandeSaut());

System.out.println(contrôleur);

contrôleur.jouer(Contrôleur.Bouton.X);
contrôleur.jouer(Contrôleur.Bouton.Y);
contrôleur.jouer(Contrôleur.Bouton.A);
contrôleur.jouer(Contrôleur.Bouton.B);

System.out.println("\nMODE DE JEU : EXPLORATION");

contrôleur.setActeur(drone);
contrôleur.reinitialiserCommandes();
contrôleur.setCommande(Contrôleur.Bouton.X, new CommandeEsquive());
contrôleur.setCommande(Contrôleur.Bouton.B, new CommandeTir());

System.out.println(contrôleur);

contrôleur.jouer(Contrôleur.Bouton.X);
contrôleur.jouer(Contrôleur.Bouton.Y);
contrôleur.jouer(Contrôleur.Bouton.A);
contrôleur.jouer(Contrôleur.Bouton.B);
```

```
MODE DE JEU : NORMAL
Contrôleur{
    [X] {SAUT}
    [Y] {TIR}
    [A] {ESQUIVE}
    [B] {CHANGEMENT_ARME}
}
BT7274-SAUTER
BT7274-TIRER
BT7274-ESQUIVER
BT7274-CHANGER_ARME

MODE DE JEU : AVENTURE
Contrôleur{
    [X] {SAUT}
    [Y] null
    [A] {ESQUIVE}
    [B] {SAUT}
}
BT7274-SAUTER
BT7274-ESQUIVER
BT7274-SAUTER

MODE DE JEU : EXPLORATION
Contrôleur{
    [X] {ESQUIVE}
    [Y] {TIR}
    [A] {ESQUIVE}
    [B] {TIR}
}
DRONE-ESQUIVER
DRONE-TIRER
DRONE-ESQUIVER
DRONE-TIRER
```