

Patron de Conception MVC :

//VueGlobale : Vue

[...]

```
viewOnShip.setOnAction(e->{
    vc.center();
});
```

[...]

```
//VueController : Controller
public void center() {
    if(engine.getVaisseau()!=null) {
        engine.centerOnShip();
    }
}
```

```
//Engine : Modele
public void centerOnShip() {
    corpFocus=this.vaisseau;
    changeObserverFlag();
    notifyObservers(corpFocus);
}
```

Iterateur :

//Exemple de l'utilisation d'un iterateur pour le parsing des données du fichier .txt ou .astro

[...]

```
Iterator<String> paramIt = param.iterator();
```

```
while (paramIt.hasNext()) {
    if (!pass) {
        paramIt.next();
        pass = true;
    }
    stockParam(paramIt);
}
```

[...]

Observateur/Observable :

//Utilisation du patron de conception Observateur/Observable pour la centralisation de la deuxième vue sur le vaisseau

```
public void centerOnShip() {
    corpFocus=this.vaisseau;
    changeObserverFlag();
    notifyObservers(corpFocus);
}
```

Encapsulation :

```
//Encapsulation de la classe CorpsCeleste dans Engine
public class Engine extends Observable{
    private List<CorpsCeleste> systeme;
    [...]
}
```