



## Interface – Refactoring – Sensibilisation à la séparation des responsabilités

### ❶ Dernier rappel : Utilisation de Git

#### *Récupération du dépôt distant*

A chaque fois que vous commencez à travailler chez vous ou à l'université, dans votre explorateur de fichier cliquer droit sous votre dossier 'LesGaulois' puis ouvrir 'Git Bash Here'.

En ligne de commande tapez la commande 'git pull'.

#### *Historisation périodique de votre projet*

Rappel : lorsque vous avez fini une question du sujet de TP (ce qui vous a amené à créer ou modifier plusieurs classes), vous devez procéder à l'historisation de votre dépôt local et la mise à jour du dépôt distant.

Dans votre explorateur de fichier cliquer droit sous votre dossier 'LesGaulois' puis ouvrir 'Git Bash Here' . Il faut réaliser trois commandes successives :

- Enregistrement des modifications à historiser dans l'index au moyen de la commande 'git add .' (n'oubliez pas le point)
- Enregistrement dans le dépôt local des fichiers mis à l'index avec la commande 'git commit -m <Intitulé des modifications>'  
Exemple : `git commit -m "TP1 fin de l'exercice 1"`  
De manière plus professionnelle il faudrait donner la nature des modifications effectuées.
- Mise à jour du dépôt distant avec la commande 'git push'<sup>1</sup>

---

<sup>1</sup> A la première utilisation de la commande, une authentification par le navigateur peut être demandée. Complétez cette identification dans votre navigateur et revenez sur votre terminal continuer votre processus Git.



## ② Fin du TP précédent – Des soldats sournois : l’embuscade

### 1. Préparer le combat :

Equiper les deux soldats avec casque, plastron et bouclier.

Affichage de cette partie :

```
Les soldats s'étaient bien préparés :  
- Le soldat Quintilius s'équipe avec un casque.  
- Le soldat Quintilius s'équipe avec un plastron.  
- Le soldat Quintilius s'équipe avec un bouclier.  
- Le soldat Minus s'équipe avec un casque.  
- Le soldat Minus s'équipe avec un plastron.  
- Le soldat Minus s'équipe avec un bouclier.
```

### 2. Décrire le combat :

Tant qu'il n'y a pas toute une équipe KO. Aléatoirement un soldat et un promeneur qui n'ont pas abandonné sont choisis. Puis d'une manière aléatoire on choisit qui frappe et qui reçoit le coup.

Comme dit plus haut cette méthode demande un peu plus de refactoring que précédemment.

Ici on a besoin aussi de découper cette méthode en méthodes privées simples :

- une méthode permettant de sélectionner un promeneur qui n'a pas abandonné,
- une méthode pour sélectionner un soldat qui n'a pas abandonné,
- une méthode qui choisit qui frappe et qui reçoit le coup

Extrait de déroulement possible pour cette partie :

```
Le gaulois Astérix donne un grand coup de force 5 au soldat Minus.  
Le casque absorbe 2 du coup  
Le plastron absorbe 3 du coup  
Le bouclier absorbe 3 du coup  
Le soldat Minus : « Aïe ».  
  
Le soldat Minus donne un grand coup de force 2 au gaulois Ordralfabétix.  
Le gaulois Ordralfabétix : « Aïe ».  
  
Le soldat Quintilius donne un grand coup de force 2 au gaulois Ordralfabétix.  
Le gaulois Ordralfabétix : « Aïe ».
```



3. Donner le résultat :

Si les gaulois gagnent, on note les promeneurs encore debout, par exemple :

```
Malgré cette sournoise attaque, nos promeneurs s'en sont sortis indemnes.  
Ils pouvaient compter sur la force de Cétautomatix, Obelix et Astérix.
```

Pour cet affichage il faudra probablement écrire une méthode privée afin de trouver les gaulois valident. On remarque que cette sortie fait appel à la même présentation que pour la méthode du choix des combattant (l'affichage des promeneurs qui n'ont pas abandonné sont séparés de virgule, avec un « et » entre les deux derniers et terminant par un point). Arranger votre code afin de factoriser cette partie.

Sinon, si les romains gagnent, on note dans quel camp seront ramenés les prisonniers :

```
L'attaque fut tellement rapide et inattendue que nos malheureux gaulois n'ont pas eu le  
temps de réagir.  
Ils furent ligotés et emmenés dans le camp de Chorus.
```