

OK

Reprendre

### Méthode d'évaluation

Pour vous évaluer, nous allons faire une chose simple : vous faire vous « battre » entre vous. Nous vous fournissons une classe `Strategie` java qui sera une base de stratégie, et une classe `Strat0` qui est une stratégie peu évoluée. Par contre elle vous sert de base car il y a tout pour qu'elle renvoie des choses valides.

Du fait de la dynamique du jeu et de la forte dépendance d'une stratégie donnée par rapport à ou aux adversaires, nous avons décidé de simplifier la notation par rapport à ce qui avait été imaginé. En lieu et place de paliers, nous vous proposons de vous répartir en groupes selon les performances de votre stratégie jouant seule. Si votre stratégie dépasse 60 points en moyenne sur 1000 parties, vous dépassez 5/20. À 75 vous dépassez 10, à 90 vous dépassez 15. Ensuite, cela dépendra de votre performance. Tous ceux dans un palier vont concourir entre eux. Moitié des points (2.5) en combat 1 vs 1, moitié en combat à N joueurs. Pour information, un enseignant code une stratégie faisant 45 points en moyenne en 30 mins, 75 points en 3h.

### Travail à réaliser

Pour créer votre stratégie, rien de plus simple. D'abord, trouvez votre numéro de stratégie sur la page MLS dans le fichier Excel puis renommez votre `Strat0` java avec le refactor pour lui donner le bon nom (`Strat125` si vous avez le numéro 125). Changez la méthode `getName()` pour qu'elle corresponde (avec « **NOM, Prénom** » comme affichage) et la méthode de nom de Ville.

Ensuite, rien de plus simple, il vous suffit de coder les différentes méthodes qui sont données dans la classe et qui correspondent aux différentes étapes du jeu. Les différentes méthodes ne renvoient que des entiers ou des booléens correspondant au choix à faire. Attention à bien comprendre la structure du code du jeu (important d'avoir suivi les séances 5 et 6).

Vous écrivez en haut, en 10 lignes, en commentaire, une description rapide de votre stratégie.

### IMPORTANT

Quelques détails importants sur cette évaluation :

- Le code qui vous est donné (`Strat0`) compile. Ainsi, si vous rendez un code non compilable, ça sera 0/20 sans discussion possible.
- Idem pour un code qui ne s'exécute pas correctement ou fait planter le jeu par erreur de votre part. 0/20 direct.
- Nous avons des outils de détection de plagiat efficaces faits pour le langage Java. Ces outils sont capables de détecter des modifications simples (changement de place dans le fichier, changement de nom de variables, etc.) et même avancées (changement des indices d'une boucle etc.). Ces outils nous donnent une proximité sous forme d'un pourcentage entre deux fichiers et nous montre tout ce qui est repris de l'un à l'autre. Tout codes détectés avec une proximité > 80% seront notés ensemble, avec une note divisée par le nombre de codes similaires. Vous pouvez réfléchir ensemble, mais vous DEVEZ ÉCRIRE VOTRE CODE À VOUS.
- Si non respect de certaines règles (pas de commentaires explicatifs, pas votre nom au bon format) un malus de 2 points sera appliqué sur votre note.

### Travail sur ce projet

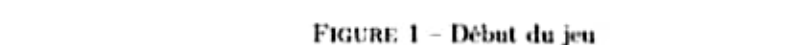
Pour plus d'informations, consultez le document de travail sur ce projet.

```
public static void exempleLanceIA() {  
    //On crée le jeu  
    try{  
        // recoit un tableau des numéros de strat (strat0 etc.).  
        // tous ensemble, n parties avec TOUT LE MONDE  
        TousEnsemble t = new TousEnsemble(new int[]{145}, _nbPartie:1000);  
  
        // championnatlv1 : les joueurs s'affrontent en duel sur n parties pour chaque duel  
        //Championnatlv1 t = new Championnatlv1(new int[]{145}, 10);  
  
        t.run();  
    }  
    catch(Exception e){  
        e.printStackTrace();  
        //System.out.println(e);  
    }  
}
```

[REDACTED]

## Le jeu complet

Figure 1: Map of 3 subcommunities (A, B, C) composed of 3 sub-plots.





## My NetBeans

### Recent Projects

<no recent project>

### Install Plugins

Add support for other languages and technologies by installing plugins from the NetBeans Update Center.

### Activate Features

NetBeans turns on functionality as you use it. Start creating and opening projects and the IDE will just activate the features you need, making your experience quicker and cleaner. Alternatively, you can activate features manually.

Output - WelcomeBis (run) X

Partie 500:

RandMan a fini avec 3 points.

Partie 1000:

RandMan a fini avec 12 points.

RandMan a fini en moyenne 1.0 avec un score moyen de 9.15 points

BUILD SUCCESSFUL (total time: 0 seconds)