



Arco de Royale



Objectif

- Deux reines s'affrontent pour le contrôle de la région. Programmez votre intelligence artificielle pour construire votre base et détruire votre adversaire.

x: 1897
y: 953



Xyze

40

 **56** **+2**

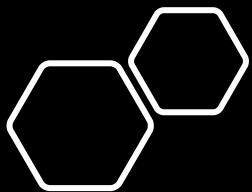
72 **+4**



RoboStac

40





Première séance

Présentation du jeu

Inscription et paramétrage

Formez des équipes de 2 et rejoignez la compétition

Tentez de battre le maximum de niveau de la ligue bois.

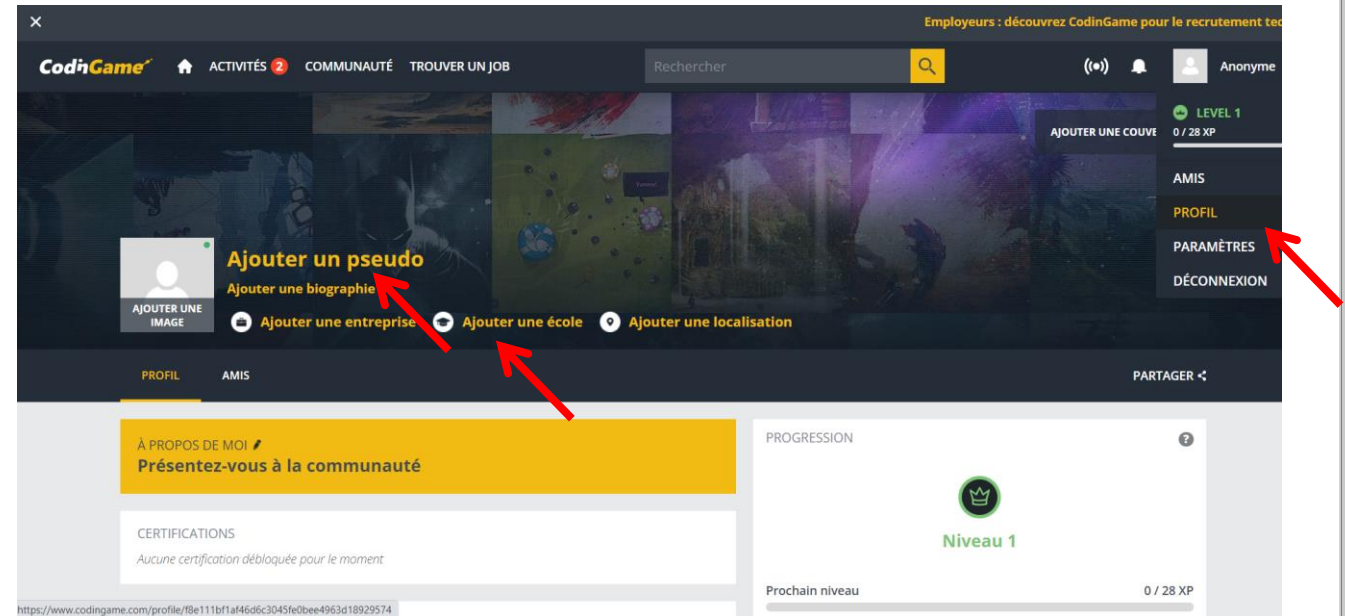
Inscription

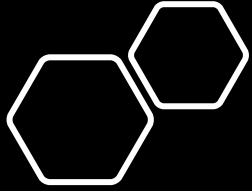
- Connectez vous sur :
<https://www.codingame.com/start>
- Cliquez sur s'inscrire
- Enregistrez vous.



Paramétrage

- Choisissez un pseudo. Il est possible que je vous demande de me l'envoyer alors prenez un public et simple.
- Vous pouvez mettre votre école si vous le souhaitez





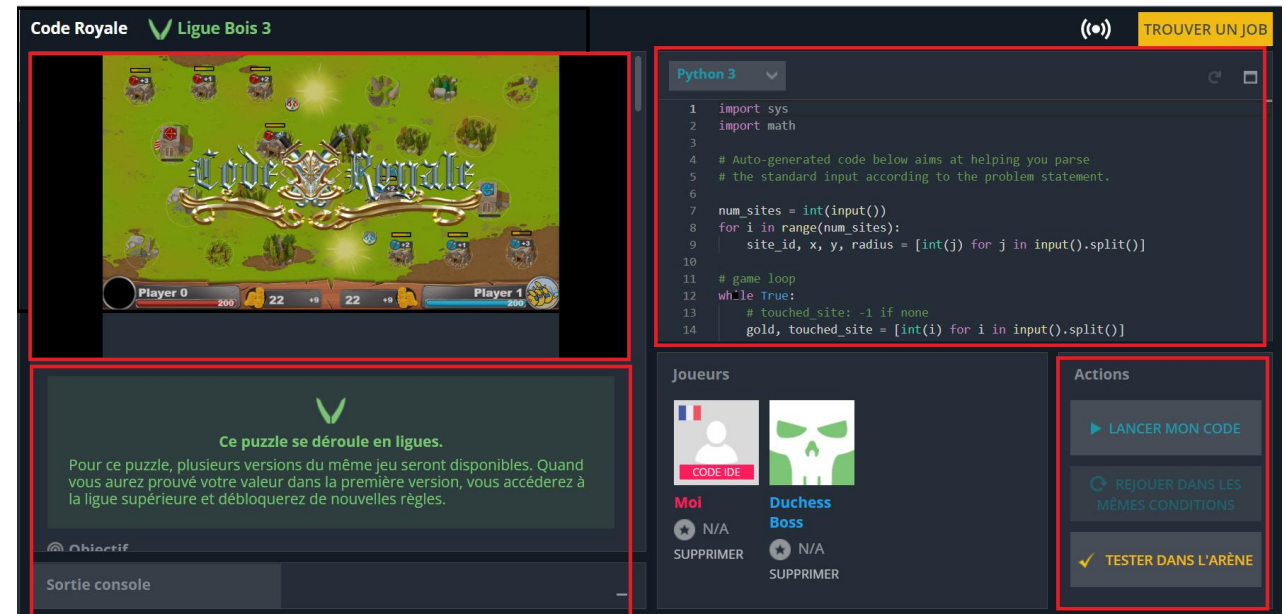
Rejoindre la compétition

- Connectez vous à <https://www.codingame.com/multiplayer/bot-programming/code-royale>
- Cliquez sur le bouton rejoindre

Organisation du jeu

Le site web divise la page en 4 grandes zones.

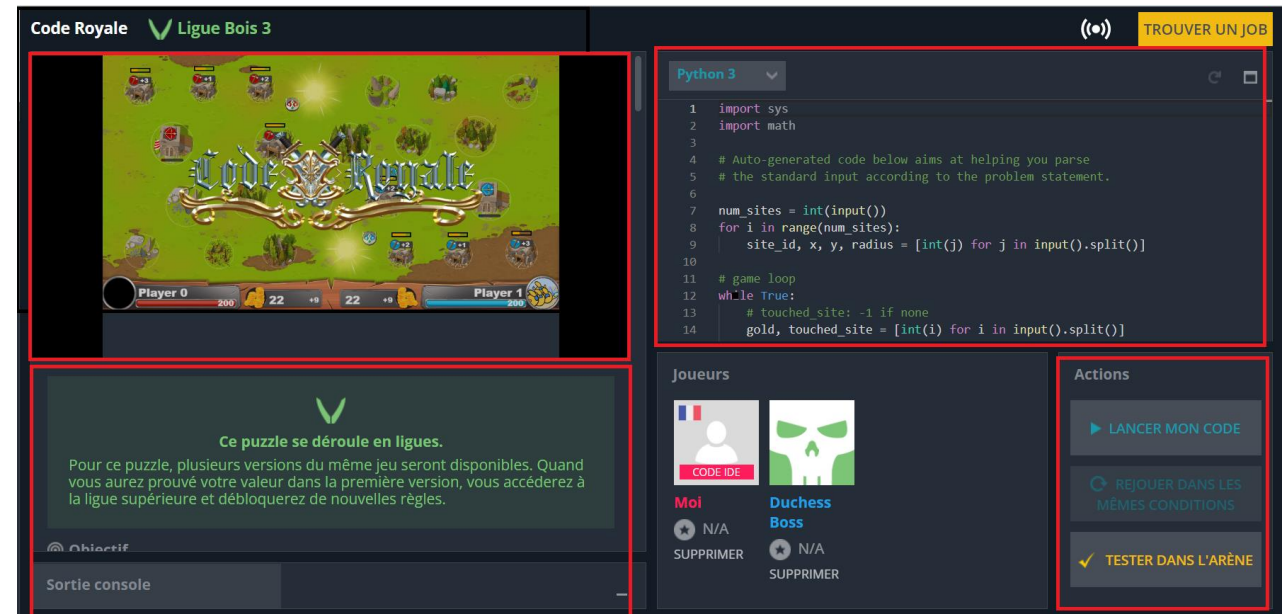
- En haut à gauche, une fenêtre qui vous permet de visualiser ce que fait votre code lors d'une simulation.
- En bas à gauche, l'intégralité de la consigne. Vous y verrez les règles du jeu ainsi que les entrants et les sortants.



Organisation du jeu

Le site web divise la page en 4 grandes zones.

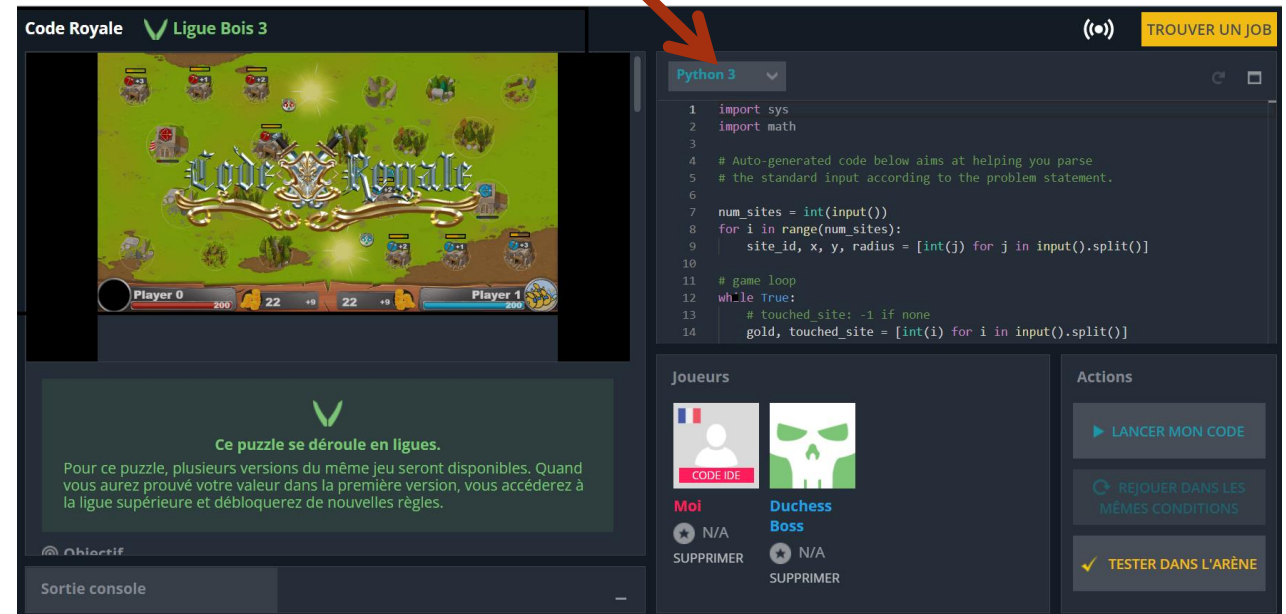
- En haut à droite, une sorte d'IDE simplifiée. C'est ici que vous déposerez votre code.
- En bas à droite, 3 boutons. Le premier permet de tester votre code sans modifier votre classement, le dernier permet de l'actualiser.



Modification du langage

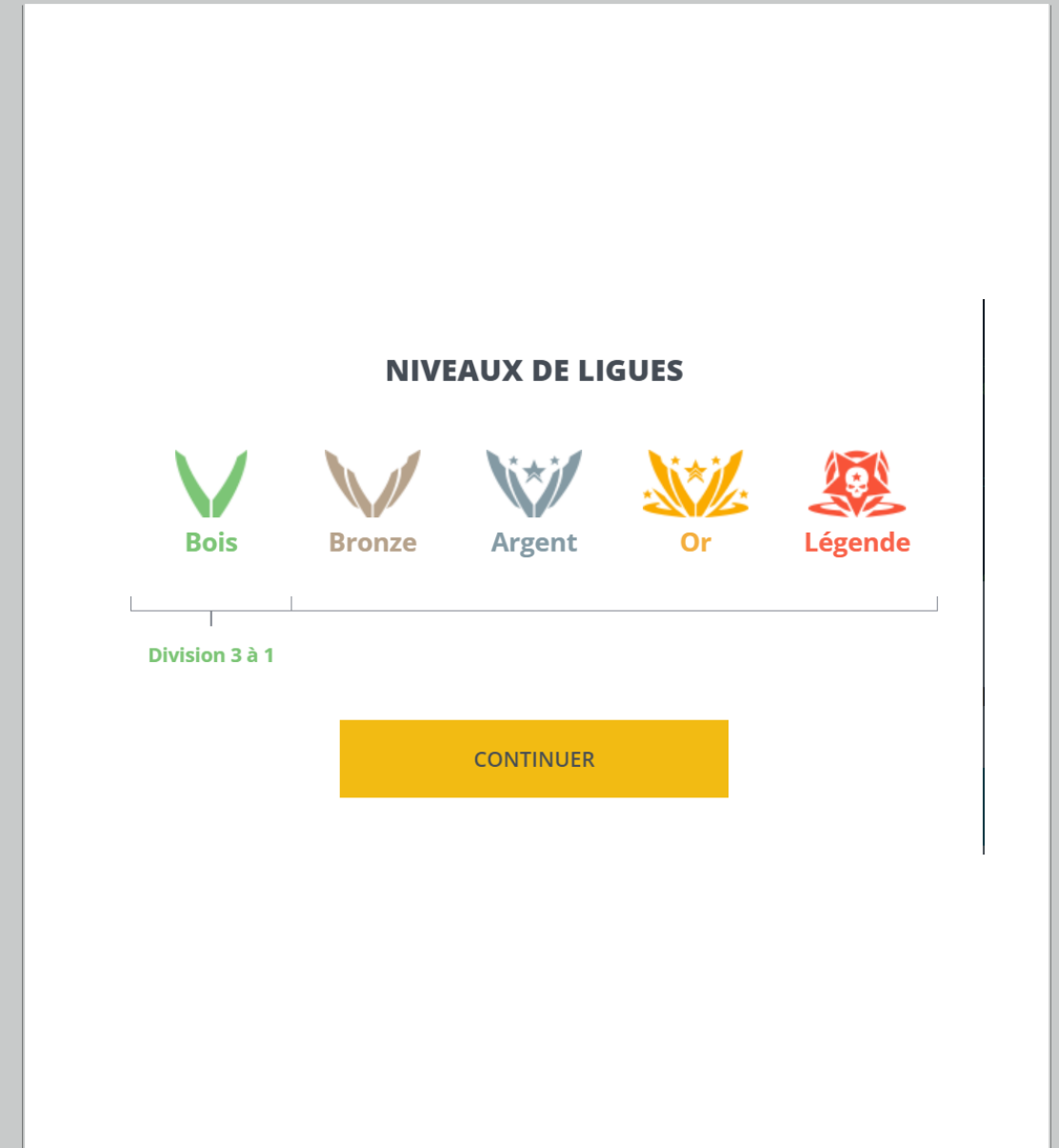
La toute première étape sera de modifier le langage pour le paramétrer en java.

Tout est prêt, vous pouvez commencer à jouer!



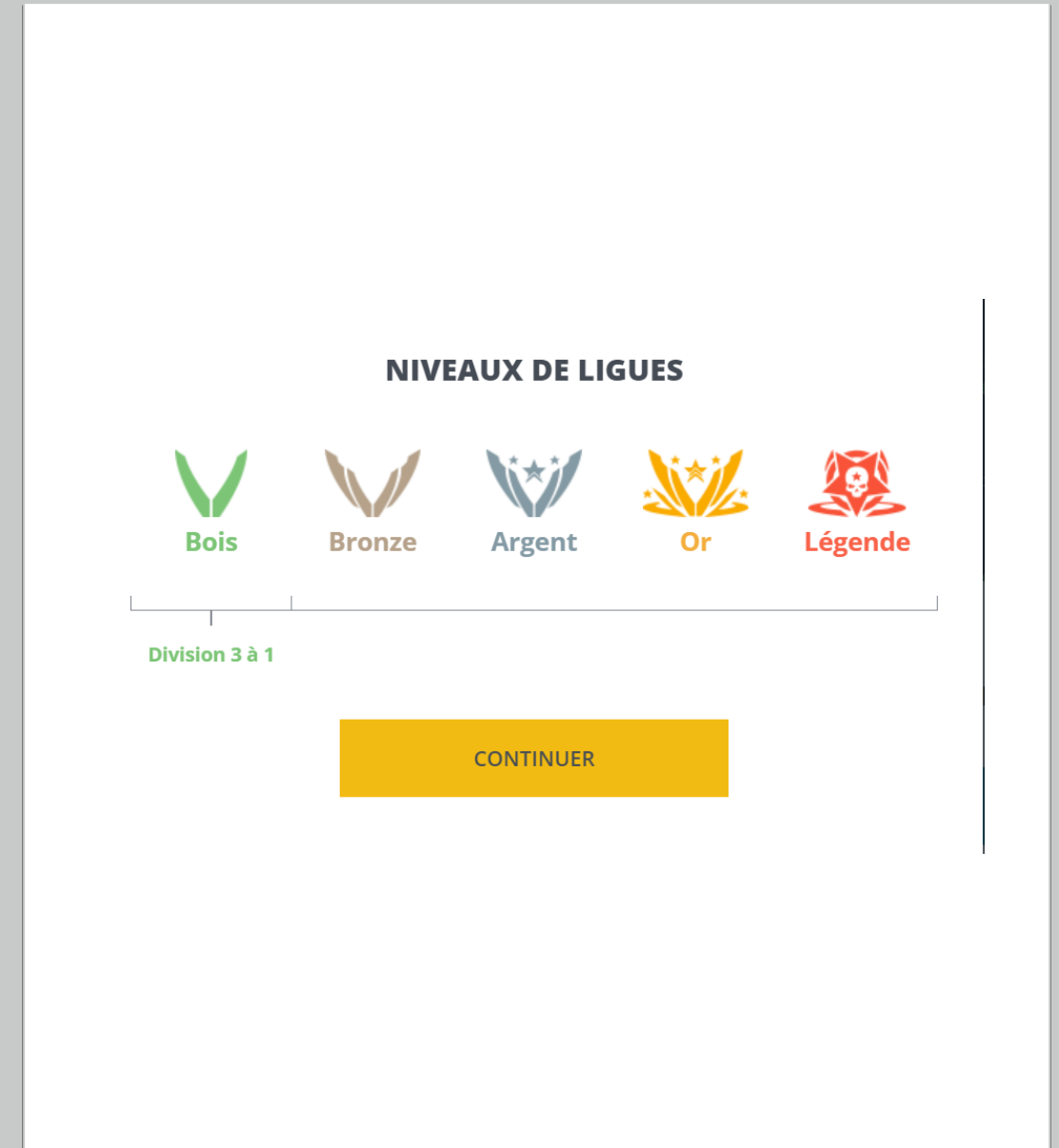
Objectif: Séance 1

- Le jeu est séparé en plusieurs ligue. Chaque ligue est défendue par un boss plus compliqué. Votre IA devra surpasser celle du niveau pour monter de ligue
- Pas d'inquiétude, il n'est pas question de se classer parmi le rang légende. Aujourd'hui, ceux qui atteindront le rang Bronze seront particulièrement forts.



Objectif: Séance 1

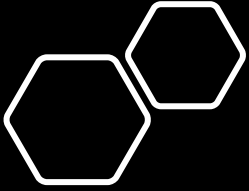
- Vous familiariser avec la syntaxe java, taper vos premières fonctions et créer vos premières class,
- L'autre objectif principal aujourd'hui est de vous amuser. L'entraide est autorisé.





Aide et astuces

- Ces astuces sont facultatives et ont pour objectifs de vous aider dans vos premiers pas.
- **Si vous ne devez retenir qu'un seul conseil: Faites simple! Vous n'avez que 2 heures, vous n'aurez pas le temps de tout modéliser.**



Premier essai

- Essayez de modifier la ligne 49:

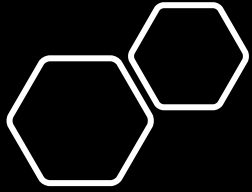
```
System.out.println("WAIT");
```

Par

```
System.out.println("BUILD 1 BARRACKS-  
KNIGHT");
```

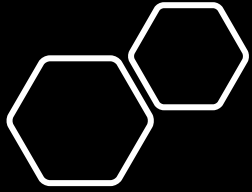
- Cliquez sur « Lancer mon code » en bas à droite.

Vous venez de lancer votre première suggestion!



Astuce 1: IDE

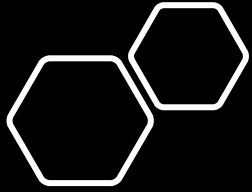
- Pour aujourd'hui un IDE n'est pas obligatoire, vous pouvez tout à fait coder dans la fenêtre du navigateur. Cependant un IDE propose différentes aides qui rendent la vie plus agréable.
- Personnellement j'utilise IntelliJ <https://www.jetbrains.com/fr-fr/idea/> mais eclipse fonctionne très bien et est très répandu en entreprise. Je déconseille Netbeans.
- Ici, exceptionnellement, il faudra coder toutes vos classes dans un seul fichier. Renommez votre classe principale « Player » et codez les autres classes du fichier sans le mot clé « public ».



Astuce 2: Réorganiser votre code

Dans un soucis de lisibilité, je conseille de réorganiser tout de suite votre code en 4 fonctions.

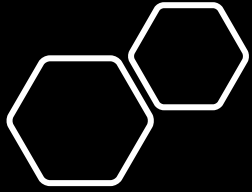
```
public static void main(String args[]) {  
    Scanner in = new Scanner(System.in);  
    instanciateGame(in);  
    int tour = 0;  
    while (true) {  
        System.err.println("    ****  TOUR : " + tour++ + "  
****  ");  
        actuliserRessources(in);  
        actualiserCarte(in);  
        actualiserArmee(in);  
  
        System.out.println("WAIT");  
        System.out.println("TRAIN");  
    }  
}  
  
public static void instanciateGame(Scanner in){ }  
// ...
```



Astuce 3: Modéliser les sites de la carte

- Votre premier objectif est de connaître l'état de la carte.
 - Créez un objet nommé Site. Ajoutez les attributs nécessaires à son bon fonctionnement. Si vous hésitez, pas d'inquiétude, vous pourrez toujours venir modifier votre code plus tard
 - Petit exemple de syntaxe:

```
class Site {  
  
    public int attribut;  
    public Site(int attribut) {}  
    public void fonction(int parametre) {  
        // code  
    }  
}
```

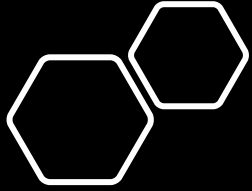



Astuce 4: Modelisez la carte

- Modifiez la méthode `initiateGame(in)` pour instancier la carte. Ici deux choix sont possibles:
 - La liste est un tableau dynamique (on ne définit pas une taille à son initialisation).
- La map permet de retrouver un élément du tableau via une clé, ici un `Integer` (on lui donnera `siteId` comme clé unique).

```
Map<Integer, Site> listeSite = new  
HashMap<>();
```

La map est un peu plus compliquée à parcourir via une boucle, mais offre l'avantage de trouver rapidement un élément via sa clé.



Astuce 5: Trouver le site le plus proche

- Toutes vos parties commenceront par demander à votre reine de construire un bâtiment. Pensez à coder une méthode vous permettant de trouver le site le plus proche.

```
public static int calculeSiteProche(xSite, ySite, xUnit,  
yUnit){ }
```