



Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-07-3-joue1-pt1

Concepts (extraits des sous-titres générés automatiquement) :

Couleurs des joueurs. Colonne d'un joueur. Séquences vidéo précédentes. Certaine colonne. Point de vue de la réponse. Structures de données. Initialisation de la grille. Séquence vidéo. Dimensions d'entiers. Valeur de retour. Premier paramètre. Type de retour. Point de vue des structures. Structures valides. Idée de son en-tête.



[vers la recherche de séquences vidéo](#)
(dans Initiation à la programmation (en Java).)



[vers la vidéo](#)

Center for Digital Education. Plus de matériel de soutien pédagogique ici :

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>

Puissance 4 : méthode joue 1ère version

(Partie 1)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s



A ce stade, nous avons vu :

- ▶ les structures de données choisies
- ▶ 2 fonctionnalités simples : `initialise` et `affiche`

🎮 Voyons maintenant comment jouer :

- ▶ demander à un joueur où il joue
- ▶ valider son coup
- ▶ demander à l'autre joueur
c'est-à-dire alterner les joueurs
- ▶ vérifier si l'un gagne (ou si le jeu est plein)

Continuons la programmation de notre jeux puissance 4. Dans les séquences vidéo précédentes, vous avez vu respectivement quelles sont les structures de données que nous vous proposons d'utiliser pour représenter ce jeu, représenter les couleurs des joueurs et 2 fonctionnalités de base que sont l'initialisation de la grille et son affichage. Intéressons-nous maintenant au fait même de jouer. De quoi avons-nous besoin ? Pour ça, il faut pouvoir demander au joueur l'endroit où il veut jouer il faut pouvoir valider son coup, valider aussi bien du point de vue de la réponse que du point de vue des structures des données, c'est-à-dire si le coup est correct, rendre les structures valides, mettre le pion dans la grille, puis ensuite il faut tour à tour demander à l'autre joueur, de jouer un coup, puis revenir, alterner entre les 2 joueurs, et puis décider si la partie se termine, soit si un des 2 joueurs a gagné, ou si la grille est pleine. Dans cette séquence vidéo-ci,

notes

résumé

0m 1s

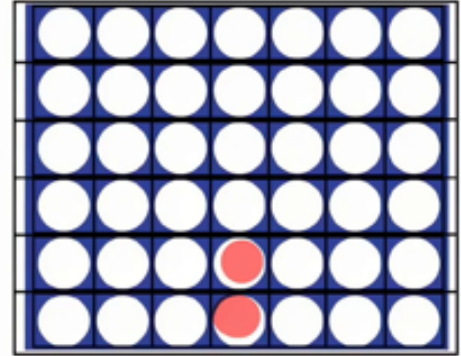


Méthode joue

```
static void joue(int[][] grille, int colonne, int couleur)
{
    |
}

...

joue(grille, 3, ROUGE);
joue(grille, 2, JAUNE);
joue(grille, 3, ROUGE);|
```



nous allons nous intéresser au second point, c'est-à-dire, recevoir une colonne d'un joueur et vérifier si elle est correcte et puis mettre à jour le jeu en fonction du coup joué. Nous sommes donc ici au clair sur ce que veut faire la méthode que l'on veut implémenter, donnons lui un nom, pour cela on va l'appeler la méthode "joue". Regardons maintenant comment on va la programmer. Commençons pour cela par son en-tête, pour avoir une idée de son en-tête, on va se demander comment on appellerait la méthode joue depuis le programme, par exemple on ferait de jouer sur la grille, à une certaine colonne, par exemple 3, Ici, on est dans le programme, les positions vont donc correspondre aux positions Java, la numérotation commence ici en zéro, et la colonne 3 serait donc...ici. Et puis il faut préciser la couleur du joueur : est-ce que c'est un joueur rouge ou est-ce que c'est un joueur jaune ? Voilà typiquement un appel que l'on pourrait faire, ça donnerait la situation suivante : en colonne 3, le joueur met un pion, on aurait un pion rouge qui viendrait se mettre ici. Puis on aurait après, typiquement des appels, où on aurait un pion, par exemple, jaune qui viendrait se mettre en colonne numérotée 2, donc en troisième colonne, et puis un joueur rouge qui rejoue dans la même colonne numéro 3, donc dans la quatrième colonne, ce qui correspond à cette situation-ci. Voilà, on est maintenant assez au clair pour pouvoir écrire l'en-tête de notre méthode joue. Donc joue va prendre comme premier paramètre la grille, donc on avait dit que c'était un tableau à 2 dimensions d'entiers, puis ensuite on va prendre une position qui indiquera la colonne, donc ici de nouveau un entier, et enfin on avait décidé

notes

résumé

1m 1s

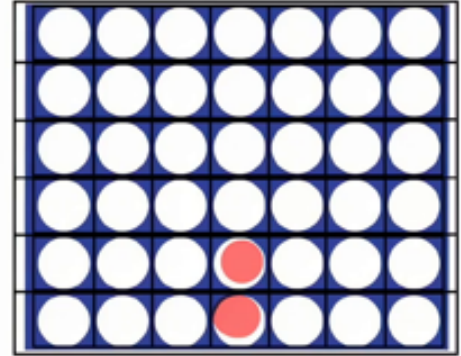


Méthode joue

```
static void joue(int[] grille, int colonne, int couleur)
{
    |
}

...

joue(grille, 3, ROUGE);
joue(grille, 2, JAUNE);
joue(grille, 3, ROUGE);|
```



de coder les couleurs par des entiers, donc on va prendre ici comme troisième argument la couleur. Voilà pour l'en-tête de la méthode joue. Maintenant, en ce qui concerne sa valeur de retour, on va commencer par le mot clé, static. Mais quel est son type de retour ? Donc le type de retour, comme on n'a pas écrit " z=joue(...)", on a simplement fait des appels, ici directement à joue, ça veut dire que la méthode joue ne renvoie rien, son type de retour va donc être void. Ce qui termine l'écriture de son en-tête, et nous permet maintenant de passer à la définition, au corps de cette méthode. Pour cela, commençons par nous demander ce que l'on veut vraiment faire, si, par exemple, on a un coup valide, que le joueur joue en colonne 3, imaginons le cas général, où on avait par exemple, déjà des pions dans la colonne 3

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

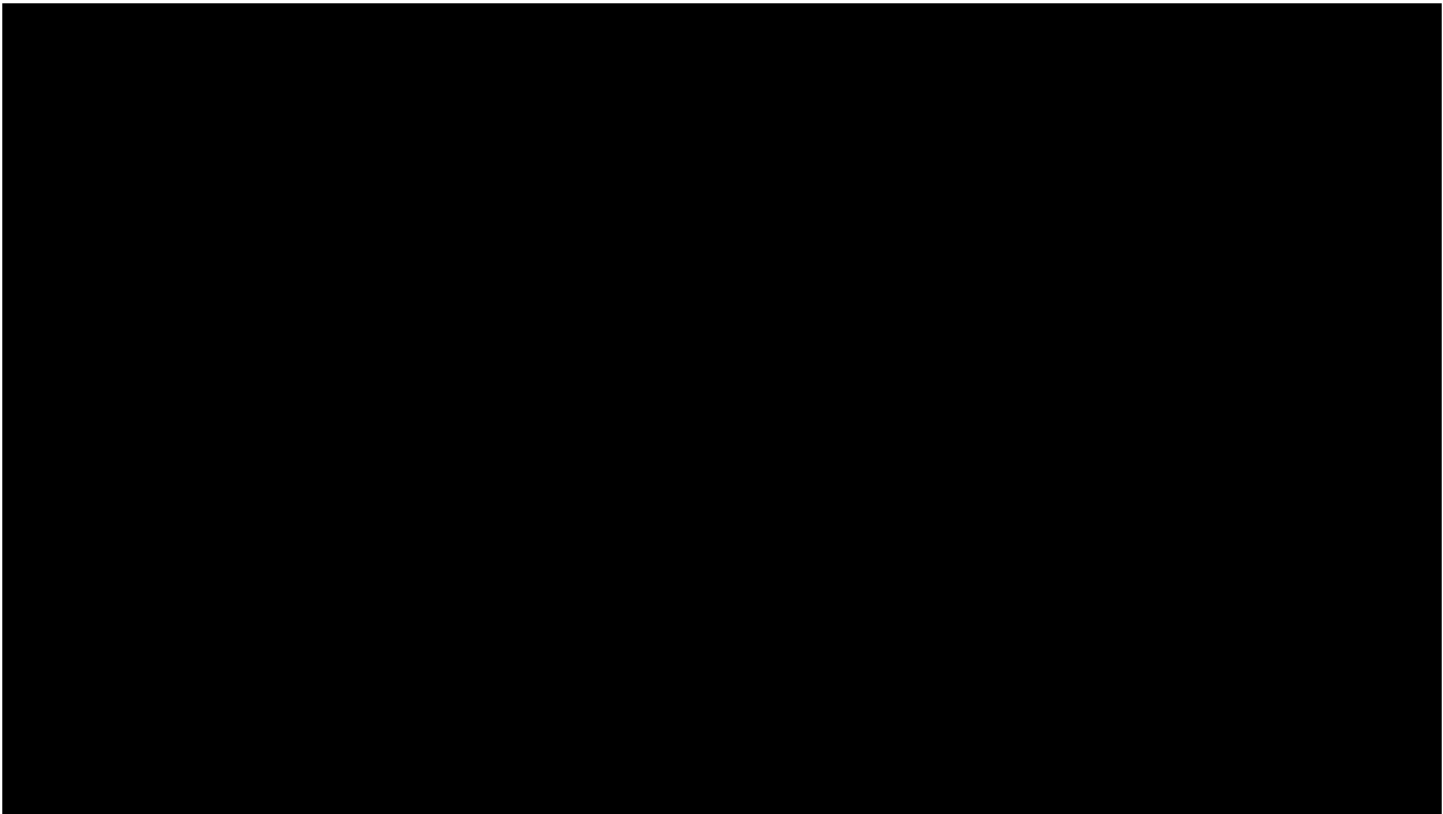
résumé

.....

.....

.....

.....



et le joueur va rejouer un pion dans la colonne 3 Que faut-il que l'on fasse ? Le pion va arriver sur la première case vide dans la même colonne. Alors, première, depuis où ? Première depuis le bas. L'algorithme va donc devoir rechercher depuis le bas, la première position vide. depuis le bas, la première position vide.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

4m 1s

