Support de cours

Cours:

## Initiation à la programmation (en C++)

Vidéo:

## Puissance 4 - fonction joue 1ère version (partie 1)

Concepts (extraits des sous-titres générés automatiquement) :

**Numéro de colonnes. Grille de jeu. Fonctions de base. Structures de données. Nombres positifs. Programmation de notre jeu de puissance. Joueur rouge. Séquences vidéo précédentes. Type entier. Premier temps. Joueur jaune. Seconde séquence. Structure de données valide. Nombre positif. Éléments d'un tableau.**

vers la recherche de séquences vidéo
(dans Initiation à la programmation (en C++).)

vers la vidéo

EPFL

# Puissance 4 : fonction `joue` 1ère version
## (Partie 1)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s

Continuons la programmation de notre jeu de puissance 4,

notes

résumé

0m 1s

## Jouer...

EPFL

A ce stade, nous avons vu :

- les structures de données choisies (`Grille`, `Couleur`)
- 2 fonctionalités 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)

dans les séquences vidéo précédentes vous aviez respectivement vu comment représenter le jeu, les couleurs, donc les structures de données utilisées par le programme et puis dans une seconde séquence les fonctions de base pour initialiser la grille de jeu et puis pour l'afficher. Voyons maintenant comment jouer. Pour pouvoir jouer qu'est-ce qu'il faut faire, il faut être capable de demander à l'utilisateur de placer un pion dans la grille, il faut valider que sa réponse était correcte, et garantir que les structures de données restent valides par rapport au coup joué, c'est-à-dire jouer effectivement le coup dans la grille, puis ensuite il faut demander à l'autre joueur de placer un pion, valider, etc donc boucler comme ça tant qu'aucun des deux joueurs n'a gagné ou que la grille n'est pas remplie. Nous allons dans cette séquence ci s'intéresser au second point, c'est-à-dire valider son coup et du point de vue de la structure de données rendre la structure de données valide en jouant effectivement dans la grille. Décidons donc dans un premier temps d'en faire une fonction. La première étape c'est de savoir très précisément ce que cette fonction doit faire de s'intéresser au « quoi », et donc ici on a une idée assez précise l'idée c'est de recevoir un numéro de colonnes qui nous indique où le joueur veut jouer, et de vérifier que ce coup est valide puis de mettre le jeu à jour en fonction donc du coup indiqué par le joueur.

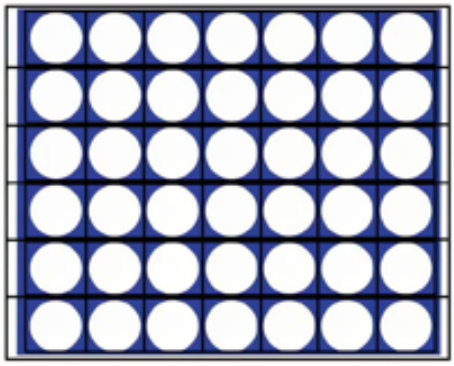
notes

résumé

0m 5s

## Fonction joue

```
joue(grille, 3, rouge
```



On peut donc la nommer cette fonction par exemple on voudrait l'appeler « joue » et on va donc maintenant s'intéresser à son prototype, quels sont les arguments que l'on doit passer à la fonction. Pour ça on va imaginer donc quels sont les appels typiques que l'on ferait de cette fonction donc typiquement on écrirait « joue », il faut bien sûr jouer sur un jeu donc on passerait une grille et puis on va indiquer la colonne dans laquelle on veut jouer donc par exemple jouer dans la colonne 3 et la couleur du joueur est ce que c'est le joueur rouge ou est ce que c'est le joueur jaune qui joue

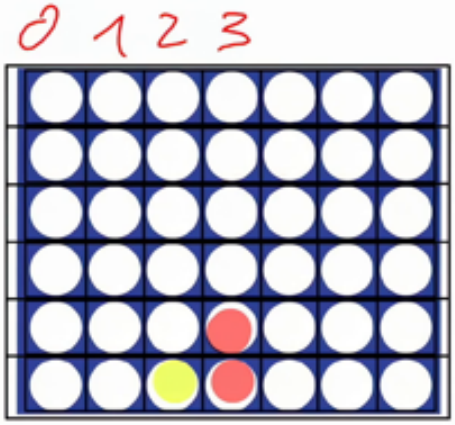
notes

résumé

1m 25s

## Fonction joue

```
joue(grille, 3, rouge);
joue(grille, 2, jaune);
joue(grille, 3, rouge);
```




donc par exemple ici on indiquerait que c'est le joueur rouge qui a joué, ce qui aurait donc pour effet de mettre un pion rouge ici en colonne 3, on pourrait faire ensuite jouer le joueur jaune par exemple le joueur jaune a envie de jouer dans la colonne 2 puis c'est de nouveau le joueur rouge qui joue dans la colonne 3. Donc on voit bien ici que l'on souhaite recevoir trois paramètres

notes

résumé

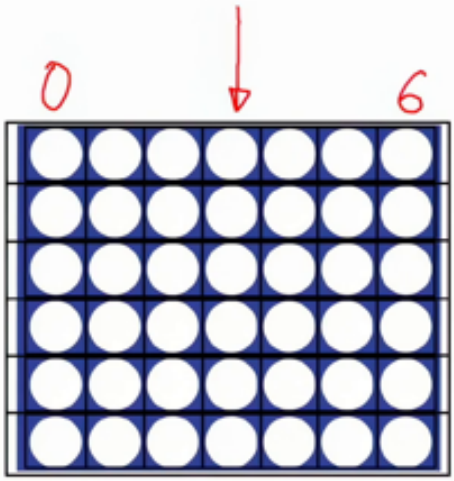
2m 1s

## Fonction joue

```
joue(Grille grille, int colonne, Couleur couleur)

...

joue(grille, 3, rouge);
joue(grille, 2, jaune);
joue(grille, 3, rouge);
```




passer trois arguments : la grille, un numéro de colonne et puis une couleur. Donc on peut commencer à écrire le prototype, donc « joue » c'est le nom de la fonction, ensuite on passe une grille qu'on va appeler grille de façon un peu originale, puis ensuite on va récupérer un numéro de colonne donc typiquement un type entier et puis enfin on va passer donc une couleur de joueur et on avait pour ça défini le type « Couleur ». Voilà donc une première version de ce prototype. Commençons par la critiquer un petit peu, déjà donc ici la colonne on a mis un entier, mais les colonnes à priori vont aller de 0 jusqu'à 6 et donc sont des nombres positifs, donc ici on va pouvoir vouloir forcer que c'est un nombre positif, et qui plus est, cette colonne va en fait nous servir à indiquer des positions dans un tableau, typiquement si on a la grille ici

notes

résumé

2m 25s

vous avez vu lors de l'affichage qu'on allait faire des « grille[i][j] » où 'j' est justement la colonne. Donc cette colonne nous servant à indexer des éléments d'un tableau, on va respecter donc les conventions qui sont que les tailles de tableau et tous ces indices, on utilise le type « size_t » donc ici on va utiliser un type « size_t », donc on corrige cet aspect là et puis ensuite deuxième correction qui est très importante, concerne la grille. concerne la grille.

notes

résumé

3m 25s