

Programmation asynchrone

■ ■ ■ Équité

Une société de paris en ligne vous contacte pour l'aider à mettre en place un système de courses virtuelles de chevaux.



- la course se déroule sur une distance de 900m ;
- il y a 5 « couloirs » ou identifiants de chevaux ;
- pour des raisons juridiques chaque cheval doit être modélisé par une fonction autonome qui doit prendre une décision aléatoire en interne et donner en sortie sa nouvelle position ;
- la fonction cheval fonctionne de la manière suivante :
 - ◇ elle possède :
 - * un numéro d'identification de 1 à 5 ;
 - * une constante de vélocité, V , sous forme d'une valeur entière de 1 à 5 choisie de manière aléatoire au **départ de la course** ;
 - * une variable de fatigue, F , sous forme d'une valeur flottante initialisée à $V/2$;
 - ◇ à chaque exécution, elle détermine :
 - * un tirage aléatoire, T , d'une valeur entière comprise entre 1 et V ;
 - * une valeur flottante de déplacement, D , correspondant au nombre de mètres parcourus, calculée de la manière suivante :

```

Si (T>V/2) alors
  F = F + (T-V/2)
  Si (F > V) alors F = V
Si (T<V/2) alors
  F = F - (V/2-T)
  Si (F < 0) alors F = 0
Si (F>V/2) alors D = T - F
Sinon D = T + F
    
```

vous pourrez mettre ce code dans une fonction et l'appeler dans votre programme.

- le cheval qui dépasse le premier les 900m est déclaré vainqueur.
La course prend fin et le cheval gagnant est donné par son numéro d'identification.
- chaque fonction « cheval » doit s'exécuter à tour de rôle pour garantir l'équité de la course.

Chaque cheval est gérée par une **fonction asynchrone**.

Pour afficher les positions de chaque cheval, une fonction `affiche_course` est appelée à la fin de l'exécution de toutes les fonctions « cheval », ce qui correspond à un « tour de jeu ».

Questions :

- a. Comment allez vous gérer les « tours de jeu », l'affichage et la détection du vainqueur ?
Vous indiquerez comment vous allez :
 - ◇ garantir l'équité pour l'appel des 5 fonctions « cheval » ;
 - ◇ partager les données de position des différents chevaux ;
 - ◇ appeler la fonction `affiche_course`.
- b. Donnez le code Javascript/algorithmique permettant de mettre en œuvre votre solution.
- c. On désire maintenant obtenir la liste des 3 premiers chevaux passant la ligne d'arrivée, indiquez quelles modifications vous devez apporter à votre programme.

- d. Suite à une modification législative, le tirage aléatoire utilisé pour déterminer T dans la fonction « cheval » doit être faite dans une fonction « jockey » dont le code est confidentiel. Il y aura autant de fonction « jockey » que de fonction « cheval ».



Indiquez comment il faudrait écrire la fonction « jockey » pour permettre les interactions avec la fonction « cheval » associée : exécution et passage de la valeur T.

Les parties confidentielles pour déterminer la valeur de T seront ignorées et notées sous forme du commentaire / calcul confidentiel de T */.*