

Informatique tronc commun

Devoir n° 4 – Partie sur machine

26 mai 2018

Durée : 60 minutes, documents et internet interdits.

1. **Lisez attentivement tout l'énoncé avant de commencer.**
2. Ce devoir est à réaliser seul, en utilisant SQLite3, soit en ligne de commandes soit en utilisant sqlite browser.
3. Nous vous conseillons de commencer par créer un dossier au nom du DS dans le répertoire dédié à l'informatique de votre compte.
4. Nous vous rappelons qu'il est possible d'obtenir de l'aide dans l'interpréteur d'idle en tapant `help(nom_fonction)`.
5. Vous inscrirez vos réponses sur la feuille réponse fournie. Attention : lisez attentivement le paragraphe suivant.

Fonctionnement du devoir

Vos réponses dépendent d'un paramètre α , unique pour chaque étudiant, qui vous est donné en haut de votre fiche réponse. Notez-le bien !

Commencez à créer un dossier sur votre ordinateur au nom de ce devoir.

Vous trouverez un fichier `bdd_morpion_ α .sqlite` sur le site de classe ainsi qu'à l'adresse suivante (où X est à remplacer par 1 ou 2).

`~/groupes/mpsX/données/d04s/bdd_morpion_ α .sqlite`

Enregistrez la base de données portant votre numéro α dans le dossier du devoir.

Si vous êtes un utilisateur de Windows et si sqlitebrowser n'est pas installé sur votre ordinateur, vous trouverez un fichier `SQLiteDatabaseBrowserPortable_3.10.1_English.paf.exe` sur le site de classe ainsi qu'à l'adresse suivante (où X est à remplacer par 1 ou 2).

`~/groupes/mpsX/données/d04s/SQLiteDatabaseBrowserPortable_3.10.1_English.paf.exe`

Enregistrez ce fichier sur votre ordinateur, puis double-cliquez dessus. Installez sqlitebrowser dans votre dossier «Mes Documents». Vous devriez normalement avoir ainsi une version fonctionnelle de sqlitebrowser.

Vous trouverez en annexe les réponses pour le paramètre $\alpha = 1$, utilisez-les pour vérifier la correction de vos algorithmes.

Les classes de MPSI du lycée LMM organisent un tournoi d'intelligences artificielles de morpion. Une partie de morpion se déroule sur un damier de trois cases par trois cases. Un premier joueur choisit une case. Puis, le deuxième joueur choisit une autre case. Puis, le premier joueur choisit une troisième case, et l'on continue ainsi jusqu'à ce que :

- soit un joueur aligne trois cases, dans ce cas il gagne ;
- soit le damier est rempli, dans ce cas il y a match nul.

1	2	3
4	5	6
7	8	9

FIGURE 1 – Repérage des cases dans un damier de morpion.

Les professeurs de ces classes ont donc créé une base de données afin de gérer ce tournoi. Cette base contient deux tables :

- une table contenant les informations des joueurs : identifiant du joueur, pseudonyme, date de naissance, classe ;
- une table contenant les informations des parties : identifiant de la partie, identifiant du joueur n° 1, identifiant du joueur n° 2, résultat de la partie, coups joués, date de la partie.

Voici les instructions SQL ayant permis de créer ces deux tables.

```
CREATE TABLE joueurs (
    idj INTEGER,
    pseudo VARCHAR(50),
    datenaiss DATE,
    classe VARCHAR(50),
    PRIMARY KEY (idj)
);

CREATE TABLE parties (
    idp INTEGER,
    idj1 INTEGER,
    idj2 INTEGER,
    res INTEGER,
    coups INTEGER,
    date DATETIME,
    PRIMARY KEY(idp),
    FOREIGN KEY(idj1) REFERENCES joueurs,
    FOREIGN KEY(idj2) REFERENCES joueurs
);
```

Quelques conventions et rappels.

- Si le résultat d'une partie vaut 0, il y a match nul, s'il vaut 1 le joueur 1 a gagné et s'il vaut 2 le joueur 2 a gagné.
- Pour une partie, on lit dans l'entier **coups** les coups effectués par les joueurs. Ainsi, si **coups** = 54892, alors
 - le joueur 1 joue dans la case 5 ;

- le joueur 2 joue dans la case 4 ;
- le joueur 1 joue dans la case 8 ;
- le joueur 2 joue dans la case 9 ;
- le joueur 1 joue dans la case 2.

Ainsi, dans cette partie, le joueur 1 a gagné (alignement sur la deuxième colonne).

- Les dates (type `DATE`) sont au format `AAAA-MM-JJ`.
- Les dates-heures (type `DATETIME`) sont au format `AAAA-MM-JJ HH:MM:SS`.
- Les dates et dates-heures sont écrites comme des chaînes de caractères (exemple : "1515-09-13") et peuvent être comparées entre elles.
- Si une requête ne donne pas qu'une réponse (exemple : plusieurs lignes vérifient un critère), on inscrira toutes les réponses.
- Pour effectuer une jointure d'une table `T` avec elle même, on pourra lui donner un alias avec l'instruction `AS`. Voici un exemple.

```
SELECT *
FROM T
JOIN T AS Tbis ON T.[...] = Tbis.[...] ;
```

- Q1** Quel est le pseudonyme de l'étudiant d'identifiant n° 42 ?
- Q2** Quel est l'identifiant de l'étudiant dont le pseudonyme est "Galois" ?
- Q3** Combien de parties ont été jouées ?
- Q4** Combien y a-t-il eu de matchs nuls ?
- Q5** Combien de parties différentes (*i.e.* de successions de coups différentes) ont été jouées ?
- Q6** Quel est l'identifiant de l'étudiant le plus jeune ? Les donner tous s'il y en a plusieurs.
- Q7** Combien de parties ont été jouées par l'étudiant de pseudonyme "Tux" ?
- Q8** Combien de parties ont été perdues par l'étudiant de pseudonyme "Tux" ?
- Q9** Combien de parties ont été gagnées par des étudiants de MPS1 contre des étudiants de MPS2 ?
- Q10** Quels sont les pseudonymes des joueurs ayant joué le plus de parties en tant que joueur n° 1 ? Les donner tous s'il y en a plusieurs.
- Q11** Combien de parties ont été jouées en sept coups ou moins, en 2018 et entre deux étudiants d'une même classe ?

Informatique tronc commun
Devoir n° 4 – Partie sur machine
Fiche de test

$\alpha = 1$

R1 :	Serre
R2 :	80
R3 :	10 043
R4 :	3292
R5 :	5994
R6 :	104
R7 :	224
R8 :	72
R9 :	1732
R10 :	Coriolis
R11 :	747