

Puissance 4

Objectif du TP

L'objectif de cette application est de mettre en place d'un jeu de puissance 4 pour deux joueurs (pas d'intelligence artificielle). L'application finale sera composée de 4 écrans : l'accueil, le plateau de jeu, un écran affichant les scores des 10 dernières parties et un écran permettant de paramétrer l'application (ajout de musique ou de son, modification des couleurs de l'arrière plan, etc.). Cette application devra, par ailleurs, être accessible, via notamment l'outil VoiceOver.

Partie I : Gestion du jeu

Question 1 : Téléchargez l'application « Puissance 4 » disponible sur Moodle. Cette application est composée d'une seule vue correspondant au plateau de jeu. Le plateau est composé de $6 * 7$ cases. Chaque case est associée à un `IBOutlet` via le tableau `caseOutlets` de type `IBOutletCollection`. Le tableau est indicé selon le schéma ci-dessous :

5	11	17	23	29	35	41
4	10	16	22	28	34	40
3	9	15	21	27	33	39
2	8	14	20	26	32	38
1	7	13	19	25	31	37
0	6	12	18	24	30	36

La case en haut à gauche est donc accessible via la variable `caseOutlets[5]` et celle en bas à droite grâce à `caseOutlets[36]`.

Chaque case est associé à un geste de type `UITapGestureRecognizer`. Les objets de type `UITapGestureRecognizer` sont stockés dans le tableau `taps`.

La méthode `handleTap` (fournie) est appelée lorsqu'un tap est détecté sur l'un des cases. Pour l'instant, l'appel de cette méthode affecte la variable d'instance `index_case_courante` avec la valeur de l'indice de la case sélectionnée.

Question 2 : Écrivez la méthode `premierIndiceLibre(indice: indice_case_courante) -> Int` qui renvoie l'indice de la première case vide dans la colonne sélectionnée par l'utilisateur. Par exemple, dans le plateau ci-dessous, si l'utilisateur clique dans la 5^{ie} colonne, par exemple sur la case 28, la fonction `PremierIndiceLibre` doit renvoyer l'indice 26.

5	11	17	23	29	35	41
4	10	16	22	28	34	40
3	9	15	21	27	33	39
2	8	14	20	26	32	38
1	7	13	19	25	31	37
0	6	12	18	24	30	36

Question 3 : Mettez à jour la méthode `handleTap` afin qu'elle modifie l'interface graphique lorsque l'utilisateur clique sur l'une des cases du plateau. Vous devez gérer l'alternance des couleurs (jaune / rouge). Pour cela, vous pouvez utiliser la variable d'instance `joueurCourant` qui peut prendre les valeurs 0 ou 1. Vous devez également mettre à jour la variable d'instance `nbCaseLibre` qui vous permettra de vérifier si le plateau est rempli ou non.

Question 4 : Proposez des éléments d'interface graphique permettant de visualiser qui doit jouer. Au début du jeu, c'est toujours le joueur qui a les pions jaunes qui commence.

Question 5 : Écrivez les méthodes permettant de vérifier si le plateau contient 4 pions alignés de même couleur. Pour cela vous pouvez décomposer le problème de la façon suivante :

1. Écrire une fonction `nbIdem(indiceMin : Int, indiceMax : Int, pas : Int) -> UIImage` qui renvoie l'image qui se répète plusieurs fois en comparant les cases dont les indices sont compris entre les indices `indiceMin` et `indiceMax` toutes les `pas` cases.

2. Écrire les méthodes permettant de tester les 4 directions (verticale, horizontale et les deux diagonales) en utilisant la méthode `nbIdem`.

5	11	17	23	29	35	41
4	10	16	22	28	34	40
3	9	15	21	27	33	39
2	8	14	20	26	32	38
1	7	13	19	25	31	37
0	6	12	18	24	30	36

Par exemple, pour tester si 4 pions de même couleur sont alignés sur la 1^{re} ligne, on peut appeler la méthode `nbIdem(indiceMin : 0, indiceMax : 36, pas : 6)`.

3. Écrire la méthode `gagne` qui appelle les 4 méthodes précédentes et renvoie le nom du joueur gagnant.

Question 6 : Écrivez une méthode `reinitialisation` qui ré-initialise le jeu (interface graphique, nombre de cases libres, etc.)

Question 7 : Complétez votre code afin de faire fonctionner le jeu :

- ◆ mise à jour de l'interface graphique lors du clique d'un joueur sur le plateau,
- ◆ vérification du plateau (plateau complet / plateau gagnant),
- ◆ affichage d'une alerte lorsque le jeu est terminé (cette alerte doit proposer aux joueurs de rejouer ou non),
- ◆ reinitialisation du plateau pour que les joueurs puissent rejouer
- ◆ etc.

Question 8 : Ajoutez un élément d'interface graphique offrant la possibilité de préciser le nom des deux joueurs avant le début de la partie. Pour cela, vous pouvez, par exemple, utiliser un objet de type `UIAlertTextView` contenant des champs de texte.

Partie II : Amélioration de l'application

Dans la première partie, vous avez mis en place un jeu de Puissance 4 entre deux joueurs. Dans cette seconde partie vous allez améliorer l'application en ajoutant la possibilité d'enregistrer les scores des parties et en offrant la possibilité aux utilisateurs de personnaliser l'apparence de l'application. Vous allez également ajouter la possibilité à un utilisateur mal voyant d'utiliser votre application grâce à l'outil VoiceOver.

Question 9 : Ajoutez les 3 écrans suivants à l'application :

- ◆ un écran « Accueil » composé de 3 boutons (Jeu, Scores, Réglages) renvoyant vers les vues correspondantes
- ◆ un écran « Score » qui affichera les scores des 10 dernières parties.
- ◆ un écran « Réglages » qui permettra de personnaliser l'application

Chacun de ces écrans doit être associés à son contrôleur de vue correspondant.

Question 10 : Ajoutez les mécanismes de segue entre les boutons de l'écran d'accueil et les 3 autres écrans de l'application.

Question 11 : Modifiez le code du fichier `viewController.swift` afin d'enregistrer les scores obtenus par les joueurs. Par exemple, Jean et Paul ont effectué 5 parties le 31 mai 2018 à 17h dont 3 ont été gagnées par Paul, 1 par Jean et 1 nulle. On souhaite que ces données soient pérennes, c'est à dire qu'elles ne soient pas effacées lorsque l'application est fermée. Il convient donc de les enregistrer dans un fichier.

Question 12 : Proposez un affichage des scores sur l'écran « Scores » de votre application.

Question 13 : Proposez différents réglages permettant de personnaliser l'interface graphique de votre application :

- ◆ ajout de musique,
- ◆ ajout de son lorsque l'utilisateur clique du le plateau,
- ◆ choix du fond d'écran de l'application,
- ◆ choix de la couleur des pions, etc.

Question 14 : Utilisez le protocole `UIAccessibility` pour rendre votre application accessible. L'objectif consiste à rendre « lisible » par VoiceOver chaque case du plateau de jeu pour informer l'utilisateur de l'indice de la case, si la case est vide, ou de la couleur du pion si celle-ci est remplie.

Question 15 : Ajoutez une icône à votre application.

Rendu : Déposez votre application sur Moodle au plus tard **le 17 juin à 23h.**