

Programmation impérative

TP Bilan : Le Morpion

Victor Drouin Viallard

1. Résolution de “jouer”

1. R0

Sémantique : La procédure jouer fait cocher une case au joueur courant lorsque le jeu est en cours et détermine alors si le coup joué le fait gagner.

Préconditions : L'état du jeu est en cours, le damier contient au moins une case libre

Postconditions : Le damier contient une case libre de moins et une case du joueur courant (celui en entrée) en plus. Le joueur courant en sortie est l'inverse de celui en entrée. L'état du jeu est NUL si le damier est plein sans vainqueur, GAGNE si le joueur courant (celui en entrée) a joué un coup gagnant, et EN_COURS sinon.

Tests :

- jouer un coup gagnant
- jouer un coup non gagnant remplissant le damier
- jouer un coup non gagnant ne remplissant pas le damier

2. R1

Le joueur choisit la case à jouer

Déterminer si le coup fait changer l'état du jeu

Changer le joueur courant

3. R2

(Le joueur choisit la case à jouer *)*

choisir_case(le_damier, le_n, le_joueur)

(Déterminer si le coup fait changer l'état du jeu *)*

calculer_etat(le_damier, le_n, le_joueur, l_etat)

(Changer le joueur courant *)*

si le_joueur est JCROIX

le_joueur devient JROND

sinon

le_joueur devient JCROIX

2. Résolution de “choisir_case”

1. R0

Sémantique : fait choisir au joueur courant la case à jouer parmi celles qui sont libres sur le damier

Paramètres : le_damier (IN OUT, Damier), le_n (IN, integer), le_joueur (OUT, Joueur)

Preconditions : le_damier contient une case LIBRE au moins

Postconditions : le_damier contient une case LIBRE de moins et une case du joueur en cours de plus.

Tests :

- L'un des joueur doit choisir une case sur un damier avec quelques cases libres et d'autres non libres

2. R1

répéter

Afficher le damier avec les numéros des cases libres

Demander au joueur de choisir la case à jouer

jusqu'à ce que la case choisie soit bien libre

mettre à jour le damier

3. R2

répéter

(Afficher le damier avec les numéros des cases libres *)*

afficher_jeu_detaille(le_damier, le_n);

(Demander au joueur de choisir la case à jouer *)*

écrire("choisir la case à jouer")

lire(la_case)

jusqu'à ce que la case choisie soit bien libre

(mettre à jour le damier *)*

si le_joueur est JCROIX

la case choisie du damier reçoit CROIX

sinon

la case choisie du damier reçoit ROND

3. Résolution de "afficher_jeu_detaille"

1. R0

Sémantique : affiche le jeu en remplissant les cases LIBRE avec un numéro qui leur est associé (j'ai décidé de ne pas faire "choix colonne puis choix ligne" pour plus de convivialité). Un numéro de case est $(sa_ligne - 1) * le_n + (sa_colonne - 1)$ de façon à toutes les numérotées depuis 0 jusqu'à $le_n^2 - 1$ de haut en bas et de gauche à droite.

Paramètres : le_damier (IN, Damier), le_n (IN, integer)

Tests :

- Afficher un damier avec quelques cases remplies et d'autres libres
- Afficher un damier plein
- Afficher un damier vide

2. R1

Pour chaque ligne du damier

Pour chaque colonne du damier

Afficher le contenu de la case le_damier(ligne, colonne)

fin pour

sauter une ligne

fin pour

4. Résolution de “case_libre”

1. R0

Sémantique : détermine si la case en entrée est LIBRE dans le damier. Un numéro de case est $(sa_ligne - 1) * le_n + (sa_colonne - 1)$ de façon à tout les numéroté depuis 0 jusque $le_n^2 - 1$ de haut en bas et de gauche à droite.

Paramètres : le_damier (IN, Damier), le_n (IN, integer), la_case (IN, integer); return boolean

Postconditions : le retour est "true" si et seulement si le numéro la_case fourni est un numéro de case valide ET que la dite case est libre

Tests :

- appel sur une case non libre
- appel sur une case libre
- **appel sur une case en dehors du jeu**

2. R1

si la ligne est valide et la colonne est valide et la case est valide et alors si la case est libre

retourner vrai

sinon

retourner faux

5. Résolution de “calculer_etat”

1. R0

Sémantique : actualise l'état du jeu en cours en fonction du joueur qui viens de jouer

Paramètres : le_damier (IN, Damier), le_n (IN, integer), le_joueur (IN, Joueur), l_etat (IN OUT, Etat_Jeu)

Preconditions : l_etat est EN_COURS et le joueur dont ce n'est pas le tour n'a pas gagné

Postconditions : L'état est NUL si le damier est plein sans vainqueur, GAGNE si le joueur en entrée a gagné, EN_COURS sinon.

Tests :

- appel sur un jeu gagnant (pour le joueur courant)
- appel sur un jeu non gagnant

2. R1

Déterminer le symbole du joueur courant

Si l'une des lignes/colonnes/diagonales est gagnante

l_etat devient GAGNE

Sinon si le damier est plein

l_etat devient NUL

Sinon

l_etat reste EN_COURS

6. Résolution de “ligne/colonne/diagonale_gagne”

1. R0

Sémantique : détermine si l'une des colonnes du damier est gagnante pour le symbole fourni

Paramètres : le_damier (IN, Damier), le_n (IN, integer), le_symbole(IN, Joueur); return boolean

Postconditions : le retour vaut true si et seulement si une colonne du damier est remplie de le_symbole

Tests :

- ligne gagnante
- ligne non gagnante

2. R1

Répéter

Tant que la case de la ligne/colonne/diagonale parcouru contient le symbole fourni et qu'elle est dans le damier

passer à la case suivante

fin tant que

passer à la ligne suivante

jusqu'à la ligne en cours n'est pas dans le damier ou la case en cours est dans le damier

Si on a pu lire toutes les lignes

retourner faux

sinon

retourner vrai

7. Résolution des autres sous programmes

Voir les spécifications données dans le fichier .adb.