Exemple d'analyse/conception

IFT 159
Analyse et Programmation

Automne 2002

Le jeu de Nim

Spécification

Le jeu de Nim est un jeu qui possède plusieurs variantes dont une qui possède une stratégie gagnante intéressante. Deux joueurs sélectionnent tour à tour des pièces à partir d'une pile. Lors de son tour, un joueur choisi le nombre de pièces qu'il retire de la pile. Il doit retirer au moins une pièce et au plus la moitié de la pile. Le second joueur joue à la suite du premier avec le même choix. Les joueurs continuent à retirer de pièces de la pile jusqu'à ce qu'il en reste une. Le joueur qui prend la dernière pièce perd la partie.

Vous devez écrire un programme dans lequel l'ordinateur joue au Nim contre un joueur humain. L'ordinateur décide aléatoirement qui commence la partie. Le joueur décide du nombre de pièces (N) initialement sur la pile. Ce nombre doit être entre 10 et 100. Le joueur peut choisir aussi entre deux niveaux de difficulté. Il peut choisir le niveau «facile»dans lequel l'ordinateur choisi aléatoirement un nombre entre 1 et N/2 et retire ce nombre de pièces de la pile. Si le joueur choisi le mode «expert», alors l'ordinateur retire toujours (si cela est possible) suffisamment de pièces afin que le nombre de pièces qui restent sur la pile soit une puissane de 2 moins 1 ($2^K - 1$).

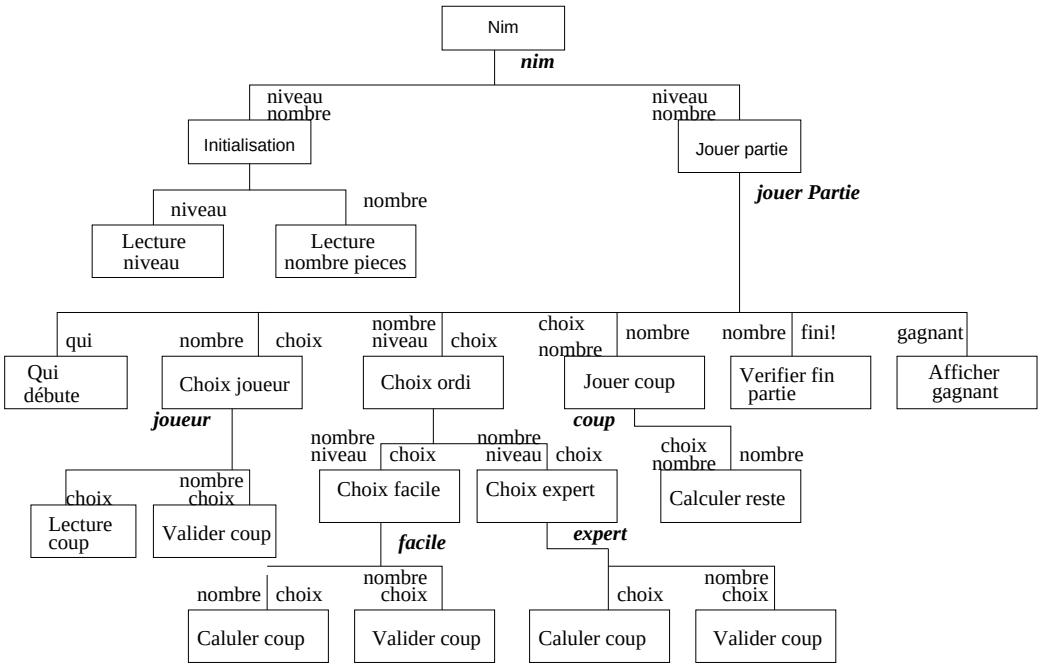
Analyse globale

Entrées : (clavier) Suite de nombres de pièces sur la pile (entiers)
(clavier) Suite de niveaux de jeu (chaînes de caractères)
(clavier) Suite de choix du joueur (entiers)

Sorties : (écran) Les résultats du jeu (entiers)

Formules : (1) Coup (facile) = (nombre aléatoire % (nombre de pièces restantes /2))+1
(2) Coup (expert) = nombre de pièces restantes - $(2^k - 1)$
tel que $2^k - 1 <$ nombre de pièces restantes
(3) Nombre de pièces restantes = nombre de pièces restantes - coup

Diagramme structurel



Conception

Module *Nim*

· Analyse
Entrées : (lecture) Nombre de pièces sur la pile (entier)
(lecture) Niveau de jeu (chaîne de caractères)

· Conception

1. Pout toutes les parties
   1.1. Lecture niveau (expert ou facile)
   1.2. Lecture nombre de pièces
   1.3. Jouer la partie (module jouerPartie)

Module *jouerPartie*

· Analyse
Entrées : (paramètre) Nombre de pièces sur la pile (entier)
(paramètre) Niveau de jeu (chaîne de caractères)
Sortie : (affichage) Gagnant (chaîne de caractères)

· Conception

1. Déterminer qui débute la partie
2. Pour chacun des coups de la partie
   2.1. Si c'est le tour de l'ordinateur
      i. Si niveau facile
         Obtenir coup facile ordinateur (CoupFacile)
      ii. Si niveau expert
         Obtenir coup expert ordinateur (CoupExpert)
   2.2. Si c'est le tour du joueur
      i. Obtenir coup du joueur (coupJoueur)
   2.3. Jouer le coup (module jouerCoup)
   2.4. Si la partie n'est pas finie $\longrightarrow$Changement (tour de l'autre) sinon on affiche le gagnant

Module *coupFacile*

· Analyse
Entrées : (paramètre) Nombre de pièces sur la pile (entier)
Sorties : (retour) Nombre de pièces à retirer (entier)
Formule : (1) Coup = (nombre aléatoire % (nombre de pieces restantes /2)) +1

· Conception

1. Calculer nombre de pièces à retirer (formule 1)
2. Valider le coup

Module *coupExpert*

· Analyse
Entrées : (paramètre) Nombre de pièces sur la pile (entier)
Sorties : (retour) Nombre de pièces à retirer (entier)
Formule : (1) Coup = nombre de pièces restantes - $(2^k - 1)$
tel que $2^k - 1 <$ nombre de pièces restantes

· Conception

1. Calculer nombre de pièces à retirer (formule 1)
2. Valider le coup

Module *coupJoueur*

· Analyse
Entrées : (lecture) Nombre de pièces sur la pile (entier)
Sorties : (retour) Nombre de pièces à retirer (entier)

· Conception

1. Lecture nombre de pièces à retirer
2. Valider le coup

Module *JouerCoup*

· Analyse

Entrées : (paramètre) Nombre de pièces à retirer (entier)
(paramètre) Nombre de pièces sur la pile (entier)
Sorties : (retour) Nombre de pièces restante sur la pile (entier)
Formule : (1) Nombre de pièces restantes = nombre de pièces restantes

· Conception

1. Calculer nombre de pièces restantes (formule 1)