

Projet Othello - LA CONCEPTION DÉTAILLÉE

Groupe 1.5

25 novembre 2015

Première partie

Conception détaillée des TAD

1 Conception détaillée des types

1.1 CD du type « Couleur »

— **Type** Couleur = {blanc, noir}

1.2 CD du type « Pion »

— **Type** Pion = **Structure**
couleur : Couleur
finstructure

1.3 CD du type « Position »

— **Type** Position = **Structure**
ligne : **Naturel**
colonne : **Naturel**
finstructure

1.4 CD du type « Plateau »

— **Type** Case = {blanc, noir, vide}
— **Type** Plateau = **Tableau**[1..8][1..8] **de** Case

1.5 CD du type « Coup »

— **Type** Coup = **Structure**
position : Position
pion : Pion
finstructure

1.6 CD du type « Coups »

— **Type** Coups = **Structure**
listeDesCoups : Liste<Coup>

finstructure

2 Conception détaillée des opérations des TAD

Deuxième partie

Conception détaillée des algorithmes compliqués de l'analyse « faireUnePartie »

1 La procédure « jouer »

procédure jouer (**E/S** plateau : Plateau, **E** coupJouer : getCoup, **S** aPuJouer : **Booleen**)

Déclaration i : Naturel
coups : Coups
joueurCourant : Couleur
res : **Booleen**

debut

joueurCourant ← obtenirCouleurPion(obtenirPionCoup(coupJoueur))

coups ← listeCoupsPossibles(plateau, joueurCourant)

pour i ← 1 **à** nbCoups(coups) **faire**

si iemeCoup(coups,i) = coup **alors**

res ← VRAI

jouerCoup(coupJoueur,plateau)

sinon

res ← FAUX

finsi

finpour

aPuJouer ← res

fin

Troisième partie

Conception détaillée des algorithmes compliqués de l'analyse « obtenirCoupIA »

1 La fonction « obtenirCoupIA »

fonction obtenirCoupIA (plateau : Plateau, couleur : Couleur) : Coup

Déclaration i, profondeurMinMax : Naturel
coupsPossibles : Coups
scoreCourant, meilleurScore : Entier
coupCourant, meilleurCoup : Coup

debut

profondeurMinMax ← profondeur()

coupsPossibles ← listeCoupsPossibles(plateau,couleur)

```

si nbCoups(coupsPossibles) > 0 alors
  meilleurCoup ← iemeCoup(coupsPossibles,1)
  meilleurScore ← scoreDUnCoup(plateau,meilleurCoup,couleur)
  pour i ← 2 à nbCoups(coupsPossibles) faire
    coupCourant ← iemeCoup(coupsPossibles,i)
    scoreCourant ← scoreDUnCoup(plateau,coupCourant,couleur)
    si scoreCourant > meilleurScore alors
      meilleurCoup ← coupCourant
      meilleurScore ← scoreCourant
    finsi
  finpour
finsi
fin

```