

Projet Othello - LA CONCEPTION PRÉLIMINAIRE

Groupe 1.5

22 novembre 2015

Première partie

Conception préliminaire des TAD

1 Conception préliminaire du TAD « Couleur »

- **fonction** blanc () : Couleur
- **fonction** noir () : Couleur
- **fonction** changerCouleur (couleur : Couleur) : Couleur

2 Conception préliminaire du TAD « Plateau »

- **fonction** creerPlateau () : Plateau
- **fonction** estCaseVide (plateau : Plateau, position : Position) : Couleur
- **procédure** viderCase (**E/S** plateau : Plateau, **E** position : Position)
 |précondition(s) non(estCaseVide(plateau,position))
- **procédure** poserPion (**E/S** plateau : Plateau, **E** position : Position, pion : Pion)
 |précondition(s) estCaseVide(plateau,position)
- **fonction** obtenirPion (plateau : Plateau, position : Position) : Pion
 |précondition(s) non(estCaseVide(plateau,position))
- **procédure** inverserPion (**E/S** plateau : Plateau, **E** position : Position)
 |précondition(s) non(estCaseVide(plateau,position))

3 Conception préliminaire du TAD « Coup »

- **fonction** creerCoup (position : Position, pion : Pion) : Coup
- **fonction** obtenirPositionCoup (coup : Coup) : Position
- **fonction** obtenirPionCoup (coup : Coup) : Pion

4 Conception préliminaire du TAD « Pion »

- **fonction** creerPion (couleur : Couleur) : Pion
- **fonction** obtenirCouleurPion (pion : Pion) : Couleur
- **procédure** retournerPion (**E/S** pion : Pion)

5 Conception préliminaire du TAD « Coups »

- **fonction** creerCoups () : Coups
- **procédure** ajouterCoups (**E/S** coups : Coups, **E** Coup)
- **fonction** nbCoups (coups : Coups) : Naturel
- **fonction** iemeCoup (coups : Coups, i : **NaturelNonNul**) : Coup
 |précondition(s) i ≤ nbCoups(coups)

6 Conception préliminaire du TAD « Position »

- **fonction** obtenirLigne (position : Position) : **NaturelNonNul**
- **fonction** obtenirColonne (position : Position) : **NaturelNonNul**
- **procédure** fixerPosition (**E** ligne, colonne : **NaturelNonNul**, **S** position : Position)
 |précondition(s) 1 ≤ ligne ≤ 8 & 1 ≤ colonne ≤ 8

Deuxième partie

Conception préliminaire des fonctions et procédures des analyses descendantes

1 Conception préliminaire de l'analyse descendante de « Faire une partie »

1.1 Types

- **Type** getCoup = **fonction**(plateau : Plateau, pionJoueur : Pion) : Coup

1.2 Sous-programmes

- **procédure** faireUnePartie (**E** coupJoueur1, coupJoueur2 : getCoup, **S** joueur : Couleur, estMatchNul : **Booleen**)
- **fonction** initialiserPlateau () : Plateau
- **procédure** jouer (**E** coupJoueur : getCoup, **E/S** plateau : Plateau, **S** aPuJouer : **Booleen**)
- **procédure** finPartie (**E** aPuJouerJoueur1, aPuJouerJoueur2 : **Booleen**, plateau : Plateau, **S** estFinie : **Booleen**, nbPionsBlancs, nbPionsNoirs : **Naturel**)
- **fonction** plateauRempli (plateau : Plateau) : **Booleen**
- **procédure** nbPions (**E** plateau : Plateau, **S** nbPionsBlancs, nbPionsNoirs : **Naturel**)
- **procédure** jouerCoup (**E** coup : Coup, **E/S** plateau : Plateau)
- **procédure** inverserLigne (**E** position : Position, pionJoueur : Pion, **E/S** plateau : Plateau)
- **procédure** inverserDiagonale (**E** position : Position, pionJoueur : Pion, **E/S** plateau : Plateau)
- **procédure** inverserColonne (**E** position : Position, pionJoueur : Pion, **E/S** plateau : Plateau)

2 Conception préliminaire de l'analyse descendante de « obtenirCoupIA »

- **fonction** obtenirCoupIA (plateau : Plateau, couleur : Couleur) : Coup
- **fonction** profondeur () :
- **fonction** listeCoupsPossibles (plateau : Plateau, couleur : Couleur) : Coups

- **fonction** coupValide (plateau : Plateau, coup : Coup) : **Booleen**
- **fonction** minMax (plateau : Plateau, coupsPossibles : Coups, couleur : Couleur, profondeur : Naturel) : **Entier**
- **fonction** scoreDUnCoup (plateau : Plateau, couleur : Couleur) : **Entier**
- **fonction** coupGagnant (plateau : Plateau, coup : Coup, couleur : Couleur) : **Booleen**
- **fonction** score (plateau : Plateau, couleur : Couleur) : **Naturel**
- **fonction** evaluerPlateau () :