

RAPPORT PROJET OTHELLO

Victorin Turnel, Paul Thulliez, Chen Yang, Ahmed Zarki, Sacha Wojciechowski

27 novembre 2022

1 Analyse

1.1 TAD COULEUR :

Nom: Couleur
Utilise: Booleen
Opérations: blanc: \rightarrow Couleur
noir: \rightarrow Couleur
estBlanc: Couleur \rightarrow Booleen
changerCouleur: Couleur \rightarrow Couleur
Axiomes: - estBlanc(blanc())
- non(estBlanc(noir()))
- changerCouleur(blanc())=noir()

1.2 TAD PION :

Nom: Pion
Utilise: Couleur
Opérations: creerPion: Couleur \rightarrow Pion
obtenirCouleurSuperieure: Pion \rightarrow Couleur
retournerPion: Pion \rightarrow Pion
Axiomes: - obtenirCouleurSuperieure(retournerPion(creerPion(col))) != col

1.3 TAD POSITION :

Nom: Position
Utilise: 1..8
Opérations: position: $1..8 \times 1..8 \rightarrow$ Position
obtenirX: Position \rightarrow 1..8
obtenirY: Position \rightarrow 1..8
Axiomes: - obtenirX(Position(x,y))=x
- obtenirY(Position(x,y))=y

1.4 TAD COUP :

Nom: Coup
Utilise: Pion, Position
Opérations: coup: $\text{Pion} \times \text{Position} \rightarrow \text{Coup}$
obtenirPionCoup: $\text{Coup} \rightarrow \text{Pion}$
obtenirPositionCoup: $\text{Coup} \rightarrow \text{Position}$
Axiomes: - obtenirPionCoup(coup(pion,pos))=pion
- obtenirPositionCoup(coup(pion,pos))=pos

1.5 TAD COUPS :

Nom: Coups
Utilise: Coup, NNN, Naturel
Opérations: Coups: $\rightarrow \text{Coups}$
nbCoups: $\text{Coups} \rightarrow \text{Naturel}$
ajouterCoups: $\text{Coups} \times \text{Coup} \rightarrow \text{Coups}$
iemeCoup: $\text{Coups} \times \text{NNN} \rightarrow \text{Coup}$
Préconditions: iemeCoups(coups,position): position \leq nbCoups(coups)
Axiomes: - Coups()
- nbCoups(Coups())=0
- nbCoups(ajouterCoups(coups,coup)) = nbCoups(coups) + 1
- iemeCoup(ajouterCoups(coups,coup),nbCoups(coups)+1) = coup

1.6 TAD PLATEAU :

Nom: Plateau
Utilise: Pion, Position, **Booleen**
Opérations: plateau: $\rightarrow \text{Plateau}$
poserPion: $\text{Position} \times \text{Pion} \times \text{Plateau} \rightarrow \text{Plateau}$
obtenirPion: $\text{Position} \times \text{Plateau} \rightarrow \text{Pion}$
caseVide: $\text{Position} \times \text{Plateau} \rightarrow \text{Booleen}$
retournerPion: $\text{Position} \times \text{Plateau} \rightarrow \text{Plateau}$
enleverPion: $\text{Position} \times \text{Plateau} \rightarrow \text{Plateau}$
Préconditions: obtenirPion(pos,pl): non (caseVide(pos,pl))
retournerPion(pos,pl): non (caseVide(pos,pl))
poserPion(pos,pion,pl): caseVide(pos,pl)
enleverPion(pos,pl): non (caseVide(pos,pl))
Axiomes: - caseVide(pos,plateau())
- obtenirPion(pos,poserPion(pos,pion,plateau()))= pion
- caseVide(pos,enleverPion(pos,plateau()))
- non caseVide(pos,poserPion(pos,pion,plateau()))
- retournerPion(pos,retournerPion(pos,pl))=pl

««««< HEAD

1.7 Analyse descendante de la fonction faireUnePartie

=====

2 Conception préliminaire

»»»> 585e0ceb2db50707af665320b383fcd15dfe4231

2.1 Analyse descendante de la fonction obtenirCoup

3 Conception préliminaire

3.1 TAD : Couleur

fonction noir () : Couleur
fonction blanc () : Couleur
fonction estBlanc (c : Couleur) : **Booleen**
procédure changeCouleur (**E/S** c :Couleur)

3.2 TAD : Pion

fonction creerPion (couleur : Couleur) : Pion
fonction obtenirCouleurSuperieure (pion : Pion) : Couleur
procédure retournerPion (**E** pion : Pion,**S** Pion)

3.3 TAD : Coup

fonction coupCoup (pion :Pion) : Coup
fonction coupObtenirPionCoup (coup :Coup) : Pion
fonction coupObtenirPositionCoup (coup :Coup) : Position

3.4 TAD : Plateau

fonction plateau () : Plateau
procédure poserPion (**E/S** plateau : Plateau,**E** pos : Position , pion : Pion)
|**précondition(s)** estCaseVide(plateau , pos)
fonction obtenirPion (pos : Position , plateau : Plateau) : Pion
|**précondition(s)** non (estCaseVide(plateau , pos)
fonction estCaseVide (pos : Position , plateau : Plateau) : **Booleen**
fonction retournerPion (pos : Position , plateau : Plateau) : Plateau
|**précondition(s)** non (estCaseVide(plateau , pos)
procédure enleverPion (**E/S** plateau : Plateau,**E** pos : Position)
|**précondition(s)** non (estCaseVide(plateau , pos)
«««< HEAD

3.5 faireUnePartie

fonction faireUnePartie (couleurJoueur : Couleur , profondeur : Naturel) : Couleur
fonction plateauPartie () : Plateau
fonction resultat (plateau : Plateau) : Couleur
fonction nbPions (plateau :Plateau) : **Naturel,Naturel**
fonction verifierFini (plateau :Plateau) : **Booleen**
fonction estTotalelementRempli (plateau :Plateau) : **Booleen**
fonction aucunPosition (plateau :Plateau) : **Booleen**
procédure jouer (**E/S** plateau : Plateau , **E** pos : Position , couleur : Couleur , profondeur : **Naturel**)
fonction verifier (plateau :Plateau , pos : Position) : **Booleen**, Position
procédure flipCoup (**E/S** plateau : Plateau , **E** pos1 : Position, pos2 : Position)

3.6 obtenirCoup

fonction obtenirCoups (unPlateau : **Plateau**, joueur : **Pion**, profondeur : **Naturel**) : **Coup**

 |**précondition(s)** non plateauComplet(unPlateau)

fonction scoreDUnCoup (unPlateau : **Plateau**, joueurRef, joueurCourant : **Pion**, unCoup : **Coup**,profondeur : **Naturel**) : **Entier**

fonction alphaBeta (unPlateau : **Plateau**, joueurRef, joueurCourant : **Pion**, profondeur : **Naturel**, alpha, beta : **Entier**) : **Entier**

fonction plateauComplet (unPlateau : **Plateau**) : **Booleen**

fonction coupGagnant (unPlateau : **Plateau**, unCoup : **Coup**) : **Booleen**

fonction évaluer (unPlateau : **Plateau**, joueurRef : **Pion**) : **Entier**

fonction score (unPlateau : **Plateau**, joueur : **Pion**) : **Entier**

fonction obtenirCoupsPossibles (unPlateau : **Plateau**) : **Coups**

4 Conception détaillé

=====

4.1 CP : faireUnePartie

```
fonction faireUnePartie (couleurJoueur : Couleur , profondeur : Naturel) :  
Couleur  
fonction plateauPartie () : Plateau  
fonction resultat (plateau : Plateau) : Couleur  
fonction nbPions (plateau :Plateau) : Naturel,Naturel  
fonction verifierFini (plateau :Plateau) : Booleen  
fonction estTotalelementRempli (plateau :Plateau) : Booleen  
fonction aucunPosition (plateau :Plateau) : Booleen  
procédure jouer (E/S plateau : Plateau , E pos : Position , couleur : Couleur  
, profondeur : Naturel)  
fonction verifier (plateau :Plateau , pos : Position) : Booleen, Position  
procédure flipCoup (E/S plateau : Plateau , E pos1 : Position, pos2 : Posi-  
tion)
```

```
»»»> 585e0ceb2db50707af665320b383fcd15dfe4231
```