

*Objectif de la séance :*

Réaliser la conception détaillée d'un petit programme manipulant des tableaux.

## 1 Analyse descendante du jeu Mastermind

Vous allez réaliser la conception détaillée d'un jeu de Mastermind opposant 2 joueurs. Les règles du Mastermind sont les suivantes :

- Le premier joueur choisit une combinaison ordonnée de 4 couleurs parmi les couleurs autorisées : jaune, orange, rose, rouge, vert, bleu. Cette combinaison n'est pas connue du second joueur qui devra la deviner
- Le second joueur dispose de 10 tentatives maximum. Chacune de ses tentatives se déroule comme suit :
  1. le second joueur propose une combinaison de 4 couleurs
  2. le premier joueur lui répond par un nombre de pions noirs et un nombre de pions blancs. Le nombre de pions noirs correspond aux couleurs bien placées. Le nombre de pions blancs correspond aux couleurs présentes dans la combinaison mais présentes à une autre place. **Attention** : une couleur à une place donnée ne peut pas entraîner l'apparition de plus d'un pion (qu'il soit noir ou blanc).
- Le jeu se termine quand le second joueur a trouvé la combinaison cachée (il gagne) ou quand il arrive à la fin de ses tentatives (il perd).

### Exercices

1. Définissez un type de données spécifique pour représenter une couleur, et un type de données pour représenter une combinaison.

### Solution

1. Type Couleur = {jaune, orange, rose, rouge, vert, bleu}  
Type Combinaison = Tableau[1..4] de Couleur

La conception globale a permis d'identifier un ensemble de procédures et fonctions pour réaliser ce programme.

#### 1.1 Saisie d'une combinaison

Il s'agit dans cette procédure de demander à l'utilisateur de rentrer une combinaison de 4 couleurs.

{Demande à l'utilisateur de saisir une combinaison de 4 couleurs}

procédure saisie (S c : Combinaison)

---

## Exercices

---

1. Écrivez la conception détaillée de la procédure saisie.
- 

### 1.2 Scores d'une combinaison

Le score obtenu par une combinaison est exprimé comme indiqué ci-dessus par un nombre de pions blancs et noirs. Chacun de ces nombres sera donné par une fonction :

*{Renvoie le nombre de pions noirs obtenus sur un joueur propose la combinaison tentative et qu'il faut trouver objectif}*

**fonction** nbNoirs (objectif, tentative : Combinaison) : Naturel

*{Renvoie le nombre de pions blancs obtenus sur un joueur propose la combinaison tentative et qu'il faut trouver objectif}*

**fonction** nbBlancs (objectif, tentative : Combinaison) : Naturel

---

## Exercices

---

1. Écrivez la conception détaillée de ces 2 fonctions.
- 

### 1.3 Tour de jeu

Un tour de jeu consiste à saisir une proposition, afficher son résultat en nombre de pions noirs et blancs et indiquer si la combinaison du premier joueur a été trouvée.

*{Exécute un tour de jeu quand la combinaison à trouver est objectif. victoire sera vrai si le tour est victorieux et faux sinon}*

**procédure** tour (**E** objectif : Combinaison, **S** victoire : **Booleen**)

---

## Exercices

---

1. Écrivez la conception détaillée de la procédure tour.
- 

### 1.4 Programme principal

Le programme principal doit demander au premier joueur une combinaison, puis itérer les tours de jeu jusqu'à ce que la combinaison soit trouvée ou que le second joueur ait atteint 10 tentatives infructueuses.

---

## Exercices

---

1. Écrivez la conception détaillée du programme principal.
- 

*Solution*

1. **procédure** saisie (**S** c : Combinaison)

**Déclaration** i : Naturel

**debut**

---

```

    pour i ← 1 à 4 faire
        ecrire('Saisissez la couleur no', i, ' svp :')
        lire(c[i])
    finpour
fin

2. fonction nbNoirs (objectif, tentative : Combinaison) : Naturel
    Déclaration i,resultat : Naturel
debut
    resultat ← 0
    pour i ← 1 à 4 faire
        si objectif[i] = tentative[i] alors
            resultat ← resultat+1
        finsi
    finpour
    retourner resultat
fin
fonction nbBlancs (objectif, tentative : Combinaison) : Naturel
    Déclaration i,resultat : Naturel ; col : Couleur ; nbCouleurs : Tableau[jaune..bleu]
    de Naturel
debut
    resultat ← 0
    pour col ←jaune à bleu faire
        nbCouleurs[col] ← 0
    finpour
    pour i ← 1 à 4 faire
        si objectif[i] ≠ tentative[i] alors
            nbCouleurs[objectif[i]] ← nbCouleurs[objectif[i]] + 1
        finsi
    finpour
    pour i ← 1 à 4 faire
        si (objectif[i] ≠ tentative[i]) et (nbCouleurs[tentative[i]] > 0) alors
            nbCouleurs[tentative[i]] ← nbCouleurs[tentative[i]] - 1
            resultat ← resultat + 1
        finsi
    finpour
    retourner resultat
fin

3. procédure tour (E objectif : Combinaison, S victoire : Booleen)
    Déclaration tentative : Combinaison ; blancs, noirs : Naturel
debut
    ecrire('Entrez votre combinaison')
    saisie(tentative)
    noirs ← nbNoirs(objectif,tentative)
    blancs ← nbBlancs(objectif,tentative)
    ecrire('Vous obtenez ', noirs, ' pions noirs et ', blancs, ' pions blancs.')
    si noirs = 4 alors
        victoire ← Vrai
    sinon

```

```

    victoire ← Faux
  fin
fin
4. Déclaration: obj : Combinaison ; round : Naturel ; gagne : Booleen
debut
  ecrire('Joueur 1, entrez la combinaison a deviner')
  saisie(obj)
  gagne ← Faux
  round ← 1
  ecrire('Joueur 2, a vous de deviner la combinaison')
  repete
    ecrire('Tentative no', round)
    tour(obj,gagne)
    round ← round + 1
  jusqu'a ce que gagne ou (round > 10)
  si gagne alors
    ecrire('Bravo vous avez trouve la bonne combinaison!')
  sinon
    ecrire('Game over, vous avez perdu...')
  fin
fin

```