

## **SAE 1.01 – Implementation d'un besoin client**

### **Le Jeu du YAMS**

**Page 2** : Pseudo-code

**Page 4** : Explication des fonctions et procédures

**Page 8** : Schéma d'appel des fonctions et procédures

**Page 9** : Modification de la maquette

```

constante entier TAILLE_NOM_J := 14;      // taille maximum du nom du joueur
constante entier TAILLE_COMBI := 13;      // taille maximum d'une combinaison
constante entier CASE_VIDE := -1;        // valeur pour une case de tableau vide

constante entier I_BRELAN := 7;          // indice du brelan dans le tableau
constante entier I_CARRE := 8;
constante entier I_FULL_H := 9;
constante entier I_P_SUITE := 10;
constante entier I_G_SUITE := 11;
constante entier I_YAMS := 12;
constante entier I_CHANCE := 13;

type t_feuille : tableau[1..13] de entier; // feuille de marque
type t_des : tableau[1..5] de entier;      // tableau de dés
type nom : chaine(TAILLE_NOM_J);          // nom d'un joueur
type combi : chaine(TAILLE_COMBI);        // nom d'une combinaison

// Procédures et fonctions
procedure initialiser_feuilles(
    sortF t1 : t_feuille,
    sortF t2 : t_feuille);

procedure recuperer_nom_joueur(
    entF numero_joueur : entier,
    sortF nom_joueur : nom);

fonction nom_est_valide(
    entF nom_joueur : nom) delivre boolean;

fonction formater_nom_joueur(
    entF nom_joueur : nom) delivre chaine(TAILLE_NOM_J + 2);

procedure afficher_feuille_marque(
    entF nom_joueur : nom,
    entF feuille_joueur : t_feuille);

fonction lancer_des() delivre t_des;

procedure afficher_des(entF liste_des : t_des);

procedure choisir_combinaison(
    entF/sortF feuille_joueur : t_feuille,
    entF liste_des : t_des);

fonction rechercher_combinaisons(
    entF feuille_joueur : t_feuille,
    entF liste_des : t_des) delivre t_feuille;

fonction combi_vers_indice(
    entF combinaison : combi) delivre entier;

```

```

fonction indice_vers_combi(
    entF indice : entier) delivre combi;

fonction brelan(entF liste_des : t_des) delivre entier;
fonction carre(entF liste_des : t_des) delivre entier;
fonction full_house(entF liste_des : t_des) delivre entier;
fonction petite_suite(entF liste_des : t_des) delivre entier;
fonction grande_suite(entF liste_des : t_des) delivre entier;
fonction yams(entF liste_des : t_des) delivre entier;
fonction total_de(entF liste_des : t_des, entF n : entier) delivre entier ;

fonction trier_des(entF t : t_des) delivre t_des;

procedure afficher_gagnant(
    entF nom1 : nom,
    entF nom2 : nom,
    entF feuille1 : t_feuille,
    entF feuille2 : t_feuille);

programme YAMS c'est
debut
    nom_j1, nom_j2 : nom;
    feuille_j1, feuille_j2 : t_feuille;
    tour : entier;
    des : t_des;

    initialiser_feuilles(entE feuille_j1, entE feuille_j2);

    recuperer_nom_joueur(entE 1, sortE nom_j1);
    repeter
        recuperer_nom_joueur(entE 2, sortE nom_j2);
        si (nom_j1 == nom_j2) alors
            ecrireEcran("Nom de joueur déjà pris.");
        finsi
    jusqu'a (nom_j1 != nom_j2);

    pour tour de 1 à 13 faire
        // Joueur 1
        afficher_feuille_marque(entE nom_j1, entE feuille_j1);
        des := lancer_des();
        choisir_combinaison(entE feuille_j1, entE des);

        // Joueur 2
        afficher_feuille_marque(entE nom_j2, entE feuille_j2);
        des := lancer_des();
        choisir_combinaison(entE feuille_j2, entE des);
    finfaire

    afficher_gagnant(entE nom_j1, entE nom_j2, entE feuille_j1, entE feuille_j2);
fin

```

## **initialiser\_feuilles**

**Procédure** - Initialise les feuilles de marque des joueurs en les remplissant avec la valeur d'une case vide (CASE\_VIDE).

**Paramètres :**

- **t1** : la première feuille à initialiser
- **t2** : la seconde feuille à initialiser

## **recuperer\_nom\_joueur**

**Procédure** – Demande à l'utilisateur de saisir un nom de joueur. Une chaîne temporaire de 100 caractères est d'abord créée, puis vérifiée via l'appel de la fonction **nom\_est\_valide**. Les 14 premiers caractères sont ensuite transférés au programme principal.

**Paramètres :**

- **numero\_joueur** : le numero du joueur dont le nom est demandé
- **nom\_joueur** : la variable de sortie

## **nom\_est\_valide**

**Fonction** – Renvoie vrai si le nom saisi est valide (le nom n'est pas vide, il ne dépasse pas 14 caractères, \0 compris).

**Paramètre :**

- **nom\_joueur** : le nom saisi à vérifier

## **afficher\_feuille\_marque**

**Procédure** – Affiche la feuille de marque d'un joueur. N'affiche pas de valeur quand une case est vide (CASE\_VIDE), et s'occupe de calculer les totaux ainsi que le bonus. Appelle la fonction **formater\_nom\_joueur** pour insérer le nom du joueur dans l'en-tête.

**Paramètres :**

- **nom\_joueur** : le nom du joueur à afficher dans l'en-tête
- **feuille\_joueur** : le tableau des données, où chaque case représente le nombre de points

## **formater\_nom\_joueur**

**Fonction** – Renvoie le nom du joueur centré et complété par des – à droite et à gauche. Garde une marge d'un espace, expliquant pourquoi la chaîne retournée est de taille TAILLE\_NOM\_J + 2.

**Paramètre :**

- **nom\_joueur** : le nom du joueur à formater

## **lancer\_des**

**Fonction** – Propose à l'utilisateur de lancer à trois reprises 5 dés. Lors d'un relancé, la fonction se charge de demander au joueur quels dés relancer. Elle retourne à la fin les 5 dés choisis. La fonction appelle la procédure **afficher\_des** pour l'affichage des dés. Avant de retourner les 5 valeur, elle appelle **trier\_des** pour les trier.

**Pas de paramètres**

## **afficher\_des**

**Procédure** – Affiche les 5 dés l'un à côté de l'autre, chacun entouré par une bordure carrée.

**Paramètre :**

- **liste\_des** : le tableau contenant les 5 dés à afficher

## **choisir\_combinaison**

**Procédure** – Demande à l'utilisateur de choisir la combinaison à jouer en fonction de ses dés, et met à jour sa feuille de marque avec les points associés. Appelle la fonction **rechercher\_combinaisons** pour afficher à l'utilisateur les combinaisons qu'il peut jouer et les points qu'il peut obtenir selon la combinaison.

**Paramètres :**

- **feuille\_joueur** : la feuille de marque à mettre à jour
- **liste\_des** : les 5 dés choisis par le joueur

## **rechercher\_combinaisons**

**Fonction** – Renvoie une feuille de marque temporaire où chaque case représente le nombre de points que le joueur pourrait obtenir s'il choisissait la combinaison correspondante. Si une combinaison est impossible, la case est vide (CASE\_VIDE).

**Paramètres :**

- **feuille\_joueur** : la feuille de marque, afin de masquer les combinaisons déjà rentrées
- **liste\_des** : les 5 dés choisis par le joueur

## **combi\_vers\_indice**

**Fonction** – Transforme une chaîne de caractères représentant le nom d'une combinaison, en l'indice de cette combinaison dans la feuille de marque (ex : « brelan » renvoie 7). Renvoie CASE\_VIDE en cas d'erreur.

**Paramètre :**

- **combinaison** : le nom de la combinaison

## **indice\_vers\_combi**

**Fonction** – Transforme un entier, représentant l'indice d'une combinaison, en le nom de la combinaison (ex : 7 renvoie « brelan »).

**Paramètres :**

- **indice** : l'indice de la case de la feuille de marque

## **brelan, carre, full\_house, petite\_suite, grande\_suite, yams**

**Fonctions** – Renvoient le nombre de points obtenus avec la combinaison grâce aux valeurs des dés. Dans le cas où la combinaison n'est pas possible, renvoient 0.

**Paramètre :**

- **liste\_dés** : les 5 dés choisis par le joueur

## **total\_de**

**Fonction** – Additionne l'ensemble des dés dont la valeur est donnée en entrée (ex : total de 6). Dans le cas où aucun dé ne répond à la combinaison, renvoie 0.

**Paramètres :**

- **liste\_dés** : les 5 dés choisis par le joueur
- **n** : définit quel total effectuer (total de n)

## **trier\_des**

**Fonction** – Renvoie une copie de la liste des dés passée en entrée. La copie est triée, afin de faciliter le traitement par le programme (il est plus simple de vérifier l'état des dés si ces derniers sont regroupés et dans l'ordre). Cette copie n'a pas pour but d'être affichée.

**Paramètre :**

- **liste\_dés** : la liste des dés à trier

## **afficher\_gagnant**

**Procédure** – Affiche les totaux de chaque joueur sous forme de tableau, puis donne le nom du gagnant. Dernière procédure appelée par le programme, elle clôt le jeu.

### **Paramètres :**

- **nom1** : le nom du premier joueur
- **nom2** : le nom du second joueur
- **feuille1** : la feuille de marque du joueur 1, utilisée pour calculer ses totaux
- **feuille2** : la feuille de marque du joueur 2, utilisée pour calculer ses totaux

## Schéma d'appel des procédures

Seule le programme principal est décrit, et comme décrit plus haut, il n'appelle pas à lui seul l'ensemble des procédures et des fonctions. Certaines procédures en appellent d'autres. Aussi, pour comprendre quel rôle jouent ces procédures qui n'apparaissent pas dans le programme, voici un schéma d'appel des fonctions :

programme YAMS

- initialiser\_feuilles
- recuperer\_nom\_joueur
  - nom\_est\_valide
- afficher\_feuille\_marque
  - formater\_nom\_joueur
- lancer\_des
  - afficher\_des
  - trier\_des
- choisir\_combinaison
  - rechercher\_combinaisons
    - brelan
    - carre
    - full\_house
    - petite\_suite
    - grande\_suite
    - yams
    - total\_de
    - combi\_vers\_indice
  - indice\_vers\_combi
- afficher\_gagnant



## Modification de la maquette

Suite à l'avis émis sur la première version de la maquette, les modifications suivantes ont été apportées :

### Choix des dés :

```
Souhaitez-vous relancer les dés ?  
(oui / non) :  
  
Veuillez entrer les dés à relancer, un  
dé par ligne, puis terminez par "fin"  
(tous, 1, 2...) :
```

Dorénavant, seuls les chiffres seront utilisés pour le choix des dés, et non plus une chaîne (1e, 2e) comme avant.

### Affichage des dés :

```
Les dés sont lancés, vous obtenez :  
n°   1       2       3       4       5  
    [1]    [3]    [2]    [5]    [4]
```

Pour aider l'utilisateur à choisir correctement ses dés, chaque dé est numéroté de 1 à 5, et ces chiffres apparaissent durant l'affichage des dés.