# Saé 2.01 – Développement d'une application

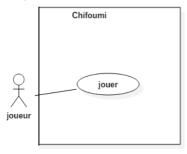
# Chifoumi – Dossier d'Analyse et conception

### 1. Compléments de spécifications externes.

On précise **uniquement** les points qui vous ont semblé flous ou bien incomplets. Rien de plus à signaler dans cette étude.

#### 1.1

## 2. Diagramme des Cas d'Utilisation



1.2

Figure 1 : Diagramme des Cas d'Utilisation du jeu Chifoumi

### 3. Scénarios

### (a) Exemple Scénario

| Cas d'utilisation   | JOUER   |   |  |
|---------------------|---|---|--|
| Résumé              | Le joueur joue une partie.                        |   |  |
| Acteur primaire     | Joueur  |   |  |
| Système             | Chifoumi  |   |  |
| Intervenants        |   |   |  |
| Niveau              | Objectif utilisateur                              |   |  |
| Préconditions       | Le jeu est démarré et se trouve à l'état initial. |   |  |
| Postconditions      |   |   |  |
| Date de oréation    |   |   |  |
| Date de mise à jour |   |   |  |
| Créateur            |   |   |  |
| Opérations          | Joueur  | Système   |  |
| 1                   | Démarre une nouvelle partie.                      |   |  |
| 2                   | ·   | Rend les figures actives et les affiche actives.  |  |
| 3                   | Choisit une figure.                               |   |  |
| 4                   |   | Affiche la figure du joueur dans la zone          |  |
|                     |   | d'affichage du dernier coup joueur.               |  |
| 5                   |   | Choisit une figure.                               |  |
| 6                   |   | Affiche sa figure dans la zone d'affichage de sor |  |
|                     |   | dernier coup.                                     |  |
| 7                   |   | Détermine le gagnant et met à jour les scores.    |  |
| 8                   |   | Affiche les scores. Retour à l'étape 3.           |  |
| Extension           |   |   |  |
| 3.A                 | Le joueur demande à jouer une nouvelle partie.    |   |  |
| 3.A.1               | Choisit une nouvelle partie                       |   |  |
| 3.A.2               |   | Réinitialise les scores.                          |  |
| 3.A.3               |   | Réinitialise les zones d'affichage des derniers   |  |
|                     |   | coups.  |  |
| 3.A.4               |   | Retour à l'étape 3.                               |  |

Tableau 1 : Scénario nominal

#### (b) Remarques:

- Le scénario est très simple.
- L'objectif est de mettre en évidence les actions de l'utilisateur, celles du système, sachant que ces actions sont candidates à devenir des méthodes du système

1.3

### 4. Diagramme de classe (UML)

(a) Le diagramme de classes UML du jeu se focalise sur les classes **métier**, cad celles décrivant le jeu indépendamment des éléments d'interface que comportera le programme.

|   | Chifoumi   |  |  |
|---|--|--|--|
|   | -coupJoueur<br>-coupMachine<br>-scoreJoueur<br>-scoreMachine |  |  |
| +unsigned int getScoreJoueur() +unsigned int getScoreMachine() +UnCoup getCoupJoueur() +UnCoup getCoupMachine() +char determinerGagnant() +setCoupJoueur(UnCoup coup) +setCoupMachine(UnCoup coup) +setScoreJoueur(unsigned int score) +setScoreMachine(unsigned int score) +initCoups() +initScores() +majScores(char gagnant) -UnCoup genererUnCoup() |  |  |  |

Figure 2 : Diagramme de Classes UML du jeu Chifoumi

#### (b) Dictionnaire des éléments de la Classe Chifoumi

| Nom attribut  | Signification  | Type     | Exemple |
|---------------|--|----------|---------|
| scoreJoueur   | Nbre total de points acquis par le joueur durant la  | unsigned | 1       |
| scoresoueur   | partie courante                                      | int      |         |
| scoreMachine  | Nbre total de points acquis par la machine durant la | unsigned | 1       |
| SCOTEMACITINE | partie courante                                      | int      | 1       |
|               | Mémorise la dernière figure choisie par le joueur.   |          |         |
| coupJoueur    | Type énuméré   | UnCoup   | papier  |
| Coapooucai    | enum unCoup {pierre, ciseau, papier,                 | oncoup   | papici  |
|               | rien};   |          |         |
| coupMachine   | Mémorise la dernière figure choisie par la machine.  | UnCoup   | Ciseau  |

Tableau 2 : Dictionnaire des éléments - Classe Chifoumi

(c) Dictionnaire des méthodes : intégrées dans l'interface de la classe : cf Figure 3

```
using namespace std;
class Chifoumi
    ///* ---- PARTIE MODÈLE ------
        ///* Une définition de type énuméré
    public:
        enum UnCoup {pierre, papier, ciseau, rien};
        ///* Méthodes publiques du Modèle
    public:
        Chifoumi();
        virtual ~Chifoumi();
        // Getters
        UnCoup getCoupJoueur();
            /* retourne le dernier coup joué par le joueur */
        UnCoup getCoupMachine();
            /* retourne le dernier coup joué par le joueur */
        unsigned int getScoreJoueur();
            /* retourne le score du joueur */
        unsigned int getScoreMachine();
           /* retourne le score de la machine */
        char determinerGagnant();
            /* détermine le gagnant 'J' pour joueur, 'M' pour machine, 'N' pour match nul
               en fonction du dernier coup joué par chacun d'eux */
         ///* Méthodes utilitaires du Modèle
    private :
        UnCoup genererUnCoup();
    /* retourne une valeur aléatoire = pierre, papier ou ciseau.
       Utilisée pour faire jouer la machine */
        // Setters
    public:
        void setCoupJoueur(UnCoup p coup);
            /* initialise l'attribut coupJoueur avec la valeur
              du paramètre p coup */
        void setCoupMachine(UnCoup p_coup);
             /* initialise l'attribut coupmachine avec la valeur
               du paramètre p_coup */
        void setScoreJoueur(unsigned int p score);
            /* initialise l'attribut scoreJoueur avec la valeur
               du paramètre p score */
        void setScoreMachine(unsigned int p score);
             /* initialise l'attribut coupMachine avec la valeur
               du paramètre p_score */
        // Autres modificateurs
         void majScores(char p_gagnant);
            /* met à jour le score du joueur ou de la machine ou aucun
               en fonction des règles de gestion du jeu */
        void initScores();
            /\star initialise à 0 les attributs scoreJoueur et scoreMachine
               NON indispensable */
        void initCoups();
            /* initialise à rien les attributs coupJoueur et coupMachine
               NON indispensable */
         ///* Attributs du Modèle
     private:
        unsigned int scoreJoueur; // score actuel du joueur unsigned int scoreMachine; // score actuel de la Machine
                                    // dernier coup joué par le joueur
        UnCoup coupJoueur;
        UnCoup coupMachine;
                                    // dernier coup joué par la machine
```

Figure 3 : Schéma de classes = Une seule classe Chifoumi

### (d) Remarques concernant le schéma de classes

- 1. On ne s'intéresse qu'aux attributs et méthodes métier. Notamment, on ne met pas, pour l'instant, ce qui relève de l'affichage car ce sont d'autres objets du programme (widgets) qui se chargeront de l'affichage. Par contre, on n'oublie pas les méthodes getXXX(), qui permettront aux objets métier de communiquer leur valeur aux objets graphiques pour que ceux-ci s'affichent.
- 2. On n'a mis ni le constructeur ni le destructeur, pour alléger le schéma.
- 3. D'autres attributs et méthodes viendront compléter cette vision ANALYTIQUE du jeu. Il s'agira des attributs et méthodes dits DE CONCEPTION nécessaires au développement de l'application.

1.3.1

### Version v0

## 5. Implémentation et tests

### 5.1 Implémentation

Liste des fichiers de cette version :

- chifoumi.h : déclaration des variables et des sous-programmes du chifoumi
- chifoumi.cpp : définition des corps des sous programmes du chifoumi

Respectivement spécification et corps de la classe Chifoumi décrite au paragraphe 4.

### 5.2 Test

Test avec le programme fourni main.cpp

| coupJoueur | coupMachine | Résultat attendu          |
|------------|-------------|---------------------------|
| pierre     | ciseau      | Incrémenter scoreJoueur   |
| papier     | ciseau      | Incrémenter scoreMachine  |
| ciseau     | ciseau      | Aucun changement de score |
| pierre     | papier      | Incrémenter scoreMachine  |
| papier     | papier      | Aucun changement de score |
| ciseau     | papier      | Incrémenter scoreJoueur   |
| pierre     | pierre      | Aucun changement de score |
| papier     | pierre      | Incrémenter scoreJoueur   |
| ciseau     | pierre      | Incrémenter scoreMachine  |