

- S2.01 -

Dossier analyse & conception

V0

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.

2. Diagramme des Cas d'Utilisation

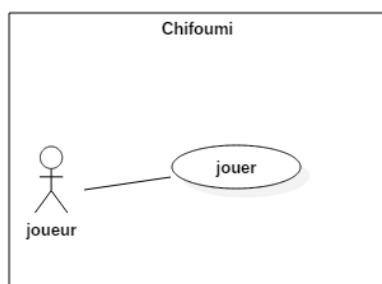


Figure 1 : Diagramme des Cas d'Utilisation du jeu Chifoumi

3. Scénarios

(a) Exemple Scénario

Titre : Jouer 2 coups
Résumé : Le joueur démarre une partie et joue 2 coups
Acteur : Utilisateur (acteur principal)
Pré-condition : Le jeu est démarré et se trouve à l'état initial.
Post-condition : néant
Date de création : 01/05/2021
Créateur : Thomas Cattarossi, Rémi Dupin, Leho Errezarret

Date de mise à jour : 13/04/2022

Utilisateur	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 son dernier coup.
	7. Détermine le gagnant et met à jour les scores.
	8. Affiche les scores.
9. Choisit une figure	10. Affiche la figure du joueur dans la zone d'affichage du dernier coup joueur.
	11. Choisit une figure.
	12. Affiche sa figure dans la zone d'affichage de son dernier coup.
	13. Détermine le gagnant et met à jour les scores.
	14. Affiche les scores.
15. Choisit une Nouvelle Partie	16. Réinitialise les scores.
	17. Réinitialise les zones d'affichage des derniers coups.
	19. Affiche les scores et les zones d'affichage des derniers coups.

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*

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.

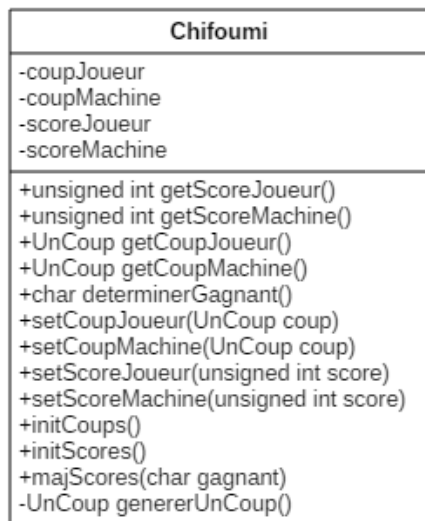


Figure 2 : Diagramme de Classes UML du jeu Chifoumi

(b) Dictionnaire des éléments de la Classe Chifoumi

Nom attribut	Signification	Type	Exemple
scoreJoueur	Nbre total de points acquis par le joueur durant la partie courante	unsigned int	1
scoreMachine	Nbre total de points acquis par la machine durant la partie courante	unsigned int	1
coupJoueur	Mémoire la dernière figure choisie par le joueur. Type énuméré enum unCoup {pierre, ciseau, papier, rien};	UnCoup	papier
coupMachine	Mémoire 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();
            /* 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
        UnCoup coupJoueur;           // dernier coup joué par le joueur
        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.

5. Classe Chifoumi : Diagramme états-transitions

(a) Diagramme états-transitions -actions du jeu

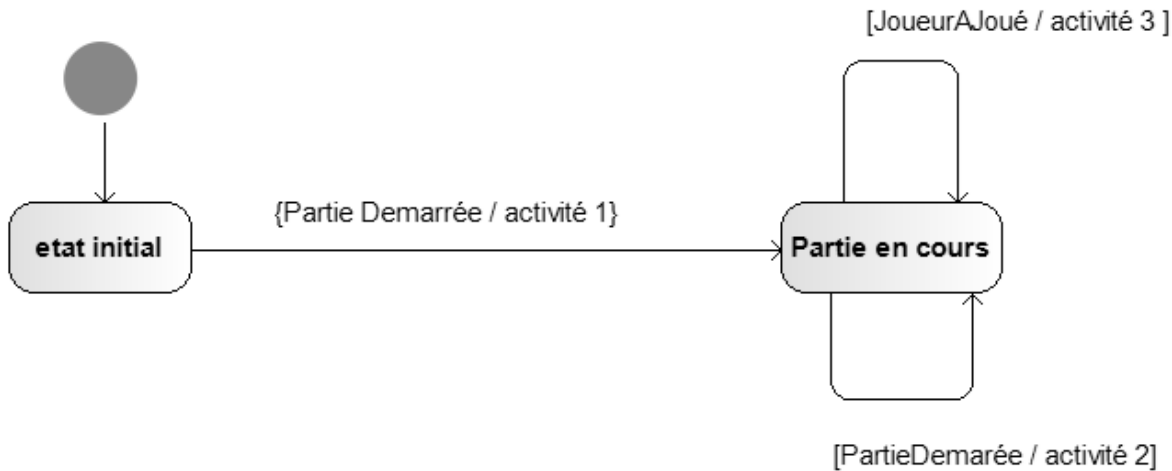


Figure 4 : Diagramme états-transitions

	BoutonNouvellePartie	BoutPierre	BoutCiseau	BoutPapier
	Partie Démarrée	JoueurAJoué		
étatInitial	partieEnCours / Activité 1			
PartieEnCours	PartieEnCours / Activité 2	PartieEnCours / Activité 3		

Tableau de la vue matricielle

(b) Dictionnaires des états, événements et Actions

Dictionnaire des états du jeu

<i>nomEtat</i>	<i>Signification</i>
etatInitial	La partie est prête a être lancé, mais le score n'apparait pas et les commandes sont grisé
partieEnCours	La partie est lancé, le score s'affiche

Tableau 3 : États du jeu

Dictionnaire des événements faisant changer le jeu d'état

<i>nomEvenement</i>	<i>Signification</i>
Bouton nouvelle partie	Enclenche une nouvelle partie et change l'état du jeu

Tableau 4 : Evénements faisant changer le jeu d'état

Description des actions réalisées lors de la traversée des transitions

nouvellePartieDemandée	RAS des scores Affiche les scores Affiche les coups Dégriser les boutons
relancerPartie	MAJ Coup joueur MAJ Coup Machine MAJ Scores
coupJoué	MAJ Coup joueur MAJ Coup Machine Maj Scores

Tableau 5 : Actions à réaliser lors des changements d'état

(c) Préparation au codage :

Table T_EtatsEvenementsJeu correspondant à la version matricielle du diagramme états-transitions du jeu :

- en ligne : les *événements* faisant changer le jeu d'état
- en colonne : les *états* du jeu

- Réalisation des tests pour le bon déroulement de la partie de jeu -

coup Joueur : papier coup Machine : ciseau
score Joueur : 2 score Machine : 2

Quitter ? (o/n)
coup Joueur : papier coup Machine : ciseau
score Joueur : 2 score Machine : 3

Quitter ? (o/n)
coup Joueur : pierre coup Machine : pierre
score Joueur : 2 score Machine : 3

Quitter ? (o/n)
coup Joueur : pierre coup Machine : pierre
score Joueur : 2 score Machine : 3

Quitter ? (o/n)
coup Joueur : ciseau coup Machine : pierre
score Joueur : 2 score Machine : 4

Quitter ? (o/n)
coup Joueur : papier coup Machine : papier
score Joueur : 2 score Machine : 4

Quitter ? (o/n)
coup Joueur : pierre coup Machine : ciseau
score Joueur : 3 score Machine : 4

Quitter ? (o/n)
coup Joueur : ciseau coup Machine : ciseau
score Joueur : 3 score Machine : 4

Quitter ? (o/n)
coup Joueur : ciseau coup Machine : pierre
score Joueur : 3 score Machine : 5

Quitter ? (o/n)
coup Joueur : ciseau coup Machine : pierre
score Joueur : 3 score Machine : 6

Quitter ? (o/n)
coup Joueur : pierre coup Machine : papier
score Joueur : 3 score Machine : 7