

## Saé 2.01 – Développement d'une application

### Chifoumi – Dossier d'Analyse et conception v1

## Sommaire :

### Version initiale :

- 1-Compléments de spécifications externe
- 2-Diagramme des cas d'utilisations
- 3-Scénarios
- 4-Diagramme de classe UML

### Version 0 :

- 1-Liste des fichiers sources
- 2-Tests des méthodes

### Version 1 :

- 1-Diagramme état transition
- 2-Liste des fichiers sources
- 3-Éléments d'interfaces
- 4-Test de l'application chifoumi

## 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

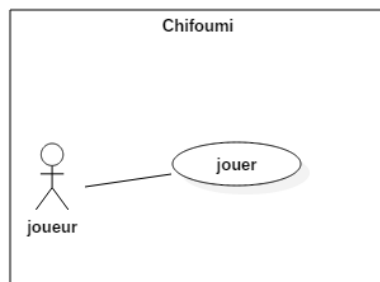


Figure 1 : Diagramme des Cas d'Utilisation du jeu Chifoumi

## 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

**Date de mise à jour :** --

**Créateur :** ...

| 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

## 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.

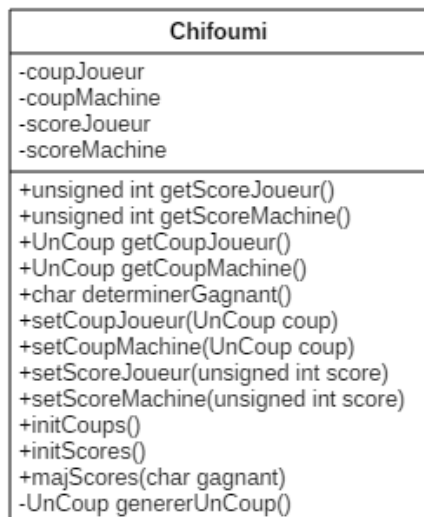


Figure 2 : Diagramme de Classes UML du jeu Chifoumi

(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.<br>Type énuméré<br>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
{
    ///  
    ///  
    public:
        enum UnCoup {pierre, papier, ciseau, rien};

        ///  
    ///  
    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 */

        ///  
    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 */

        ///  
    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

## Version 0 :

### 1-Liste des fichiers sources de cette version (et rôle de chacun)

Chifoumi.h : interface de l'application non graphique Chifoumi

```
    /** Une définition de type énuméré
public:
    enum UnCoup {pierre, papier, ciseau, rien};
    /** Méthodes 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);
        /* Mise à jour des scores en fonction des règles de gestion actuelles :
        - 1 point pour le gagnant lorsqu'il y a un gagnant
        - 0 point en cas de match nul */
    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
```

Chifoumi.cpp : corps de l'application non graphique Chifoumi

Main.cpp : fichier source contenant la boucle principale qui exécute l'application non graphique et la boucle secondaire qui se met en attente de événements

V0.pro : fichier contenant les chemins d'inclusions et l'arborescence des fichiers

Chifoumi\_dossierAnalyseConception\_V0 : dossier d'analyse & conception

## 2-Tests des méthodes :

| Test                                                                               | Résultat                                                                                  | Validation |
|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|------------|
| Méthodes get() associées aux attributs 'score'                                     | Score Joueur : 0      score<br>Machine : 0                                                | Ok         |
| Méthodes get() associées aux attributs 'coup'                                      | Coup Joueur :<br>Rien<br>Coup Machine : rien                                              | Ok         |
| Méthodes set() associées aux attributs 'score'                                     | Score Joueur : 1<br>Score Machine : 2                                                     | Ok         |
| Méthode initScores()                                                               | Score Joueur : 0<br>Score Machine : 0                                                     | Ok         |
| Méthodes set() et get() associées aux attributs 'coup'/'choix'                     | Coup Joueur : pierre<br>Coup Machine :<br>ciseau                                          | Ok         |
| Quelques tours de jeu pour tester l'identification du gagnant et la maj des scores | Coup Joueur : pierre<br>Coup Machine :<br>ciseau<br>Score Joueur : 1<br>Score Machine : 0 | Ok         |
| Appel du Constructeur                                                              | Scores à 0<br>CoupsJoueurs à rien                                                         | Ok         |

## Version 1 :

### 1-Classe Chifoumi : Diagramme états-transitions

#### (a) Diagramme états-transitions -actions du jeu

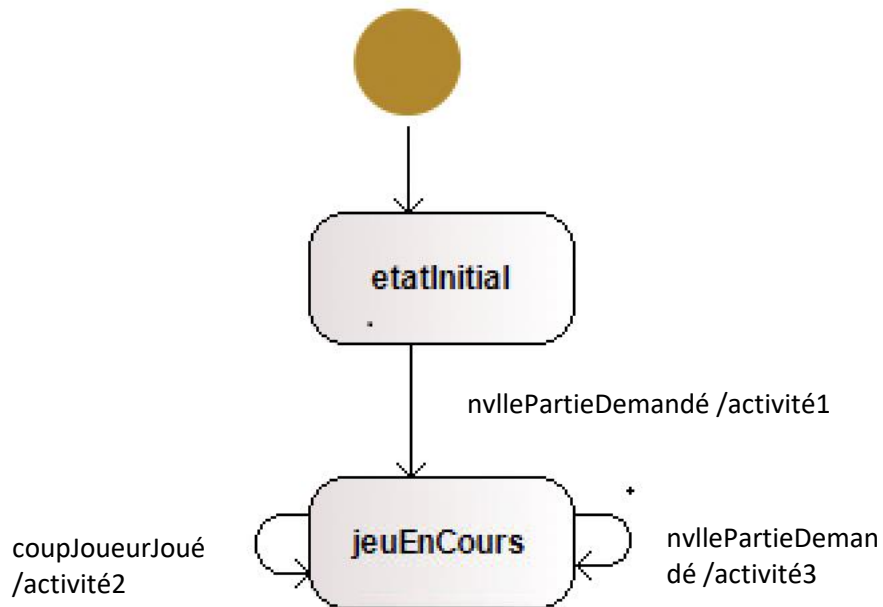


Figure 4 : Diagramme états-transitions



## (b) Dictionnaires des états, événements et Actions

### Dictionnaire des états du jeu

| <i>nomEtat</i> | <i>Signification</i>                                                                                                                                                  |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| etatInitial    | Le programme est lancé, les variables scores sont à 0 et les signes et la partie est prête à être jouée. Les signes sont initialisés et l’affichage des boutons aussi |
| jeuEnCours     | Le jeu est lancé, le joueur peut successivement avec la machine choisir son signe et les variables et fonctions seront misent à jour selon le résultat.               |

Tableau 3 : États du jeu

### Dictionnaire des événements faisant changer le jeu d’état

| <i>nomEvénement</i>  | <i>Signification</i>                                                                                                                                                                                                                                                                                                                            |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| nvlllePartieDemandée | Le joueur demande à jouer une nouvelle partie : si on était dans l’état initial alors on passe dans l’état jeu en cours en mettant à disposition la possibilité pour le joueur de choisir un signe.<br>Si on était dans l’état jeu en cours on reste dans cet état, les variables de coups et de scores sont remis respectivement à ‘rien’ et 0 |
| coupJoueurJoué       | Le joueur choisi un signe parmi les trois proposés (pierre, feuille, ciseau) et les variables de coups et de scores sont mis à jour en fonctions des choix du joueur et de la machine                                                                                                                                                           |

Tableau 4 : Événements faisant changer le jeu d’état

### Description des actions réalisées lors de la traversée des transitions

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Activité 1 : | <u>Système d’information :</u><br>Le jeu est initialisé avec les variables coups et scores du joueur et de la machine prenant les valeurs de l’état initial<br><u>Interface :</u><br>Les boutons de signes sont accessibles pour le choix de l’utilisateur<br>Les scores et nom des joueurs sont mis avec une couleur bleue pour indiquer que le jeu est en cours<br>Le focus est sur le bouton ‘NvlllePartie’                                                                                                           |
| Activité 2 : | <u>Système d’Information :</u><br>Le dernier coup du jour est mis à jour en fonction du coup joué.<br>La machine joue (tirage aléatoire) et le dernier coup de la machine est mis à jour en conséquence.<br>La machine détermine le gagnant et met à jour en conséquence les scores.<br><u>Interface :</u><br>Le signe choisi par le joueur s’affiche dans la case d’affichage prévu, de même pour celui de la machine<br>Les scores sont mis à jour en fonction du gagnant<br>Le focus est sur le bouton ‘NvlllePartie’ |

|              |                                                                                                                                                                                                                                                                                            |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Activité 3 : | <u>Système d'Information :</u><br>Les scores Joueur/Machine sont mis à 0.<br>Les derniers coups Joueur/Machine joués sont mis à rien.<br><u>Interface :</u><br>Les zones d'affichage sont mises en cohérences avec les propriétés précédentes.<br>Le focus est sur le bouton 'NvllePartie' |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

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

| <i>Événement</i> →<br><i>nomEtatJeu</i> | coupJoueurJoué         | nvllePartieDemandée  |
|-----------------------------------------|------------------------|----------------------|
| etatinitial                             |                        | jeuEnCours/activité1 |
| jeuEnCours                              | jeuEnCours / activité2 | jeuEnCours/activité3 |

Tableau 6 : Matrice d'états-transitions du jeu chifoumi

**Table T\_EtatsEvenementsJeu** avec les éléments d'interface prenant en charge les événements

|                                            |                        |              |              |                      |
|--------------------------------------------|------------------------|--------------|--------------|----------------------|
|                                            | boutonPierre           | boutonCiseau | boutonPierre | boutonNvllePartie    |
| <i>Événement</i><br>→<br><i>nomEtatJeu</i> | coupJoueurJoué         |              |              | nvllePartieDemandée  |
| etatinitial                                |                        |              |              | jeuEnCours/activité1 |
| jeuEnCours                                 | jeuEnCours / activité2 |              |              | jeuEnCours/activité3 |

Tableau 6 : Matrice d'états-transitions du jeu chifoumi AVEC éléments d'interface

## 2-Liste des fichiers sources :

chifoumimodele.h : interface du modèle de l'application Chifoumi

```
class ChifoumiModele : public QObject
{
    Q_OBJECT
public:
    enum UnCoup {pierre,papier,ciseau,rien};

    /*** PARTIE MODELE *****/

    /*** Méthodes du Modèle */
public:
    ChifoumiModele();

    // 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 */
    UnCoup genererUnCoup();
    /* retourne une valeur aléatoire = pierre, papier ou ciseau.
    Utilisée pour faire jouer la machine */

    // Setters
public slots:
    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);
        /* Mise à jour des scores en fonction des règles de gestion actuelles :
        - 1 point pour le gagnant lorsqu'il y a un gagnant
        - 0 point en cas de match nul
        */
    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
};
```

chifoumimodele.cpp : corp du modèle de l'application Chifoumi

chifoumivue.h : interface de la vue de l'application Chifoumi

```
class ChifoumiVue : public QMainWindow
{
    Q_OBJECT

public:
    ChifoumiVue(ChifoumiModele *m, QWidget *parent = nullptr);
    ~ChifoumiVue();

    /** Méthodes du Modèle
public:
    void miseAJour(ChifoumiModele::UnCoup, ChifoumiModele::UnCoup);
    //Met à jour les scores des joueurs et affiche dans les cases prévus les images de signes

//Getter
    ChifoumiModele* getModele();
//Setter
    void setModele(ChifoumiModele *m);
//Slots
public slots:
    void choixPapier();
    //Procédure qui active les différentes fonctions lorsque le joueur choisi la feuille
    void choixPierre();
    //Procédure qui active les différentes fonctions lorsque le joueur choisi la pierre
    void choixCiseau();
    //Procédure qui active les différentes fonctions lorsque le joueur choisi la ciseau
    void creerNvllePartie();
    //Procédure qui initialise les scores et les coups du joueur et de la machine, donne accès
    aux boutons de figures

private:
    ChifoumiModele *_leModele; // ptour vers le modèle
    Ui::ChifoumiVue *ui;
};
```

chifoumivue.cpp : corps de la vue de l'application Chifoumi

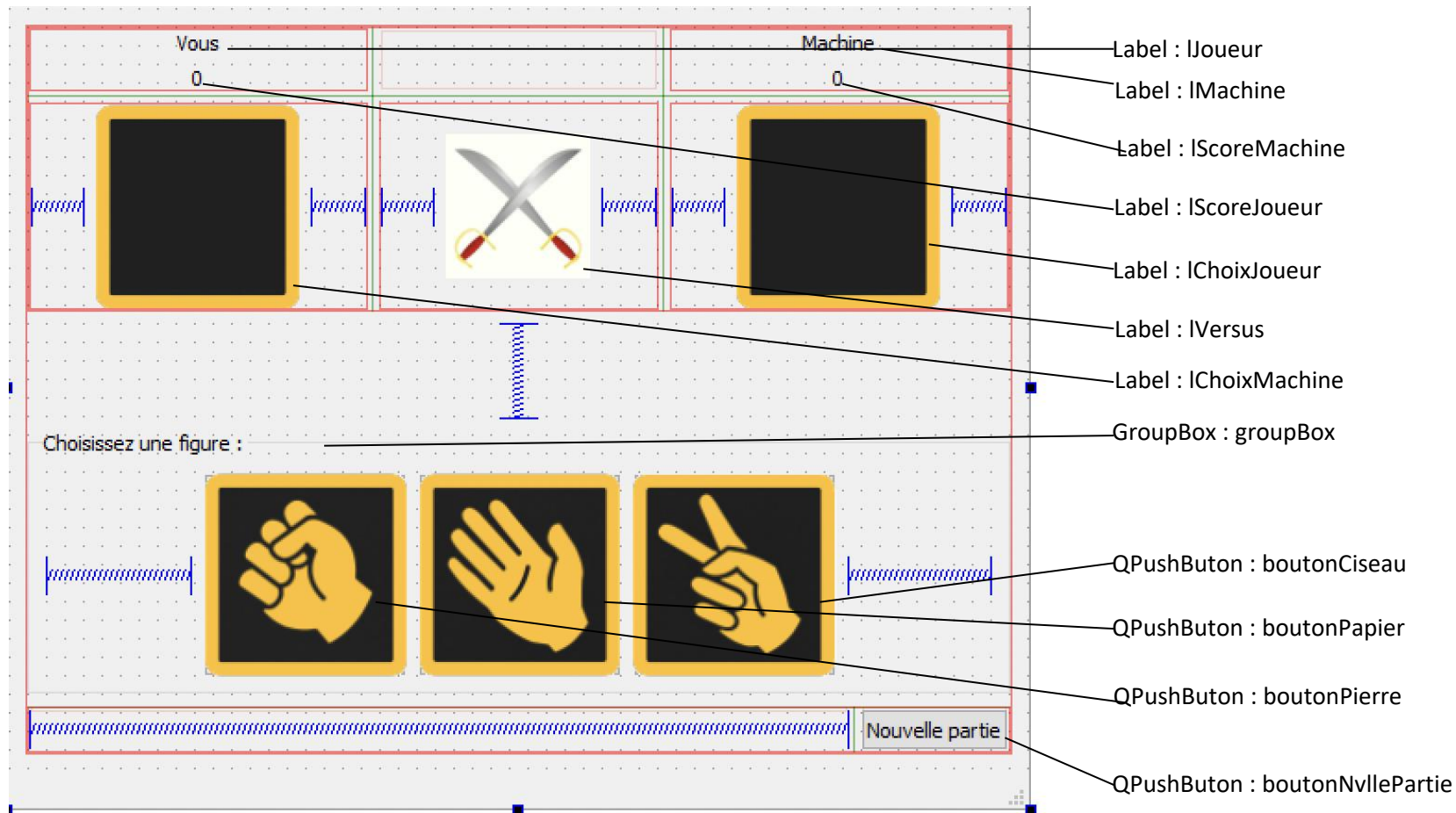
chifoumi.pro : fichier de construction du projet avec les chemins d'inclusion et l'arborescence des fichiers

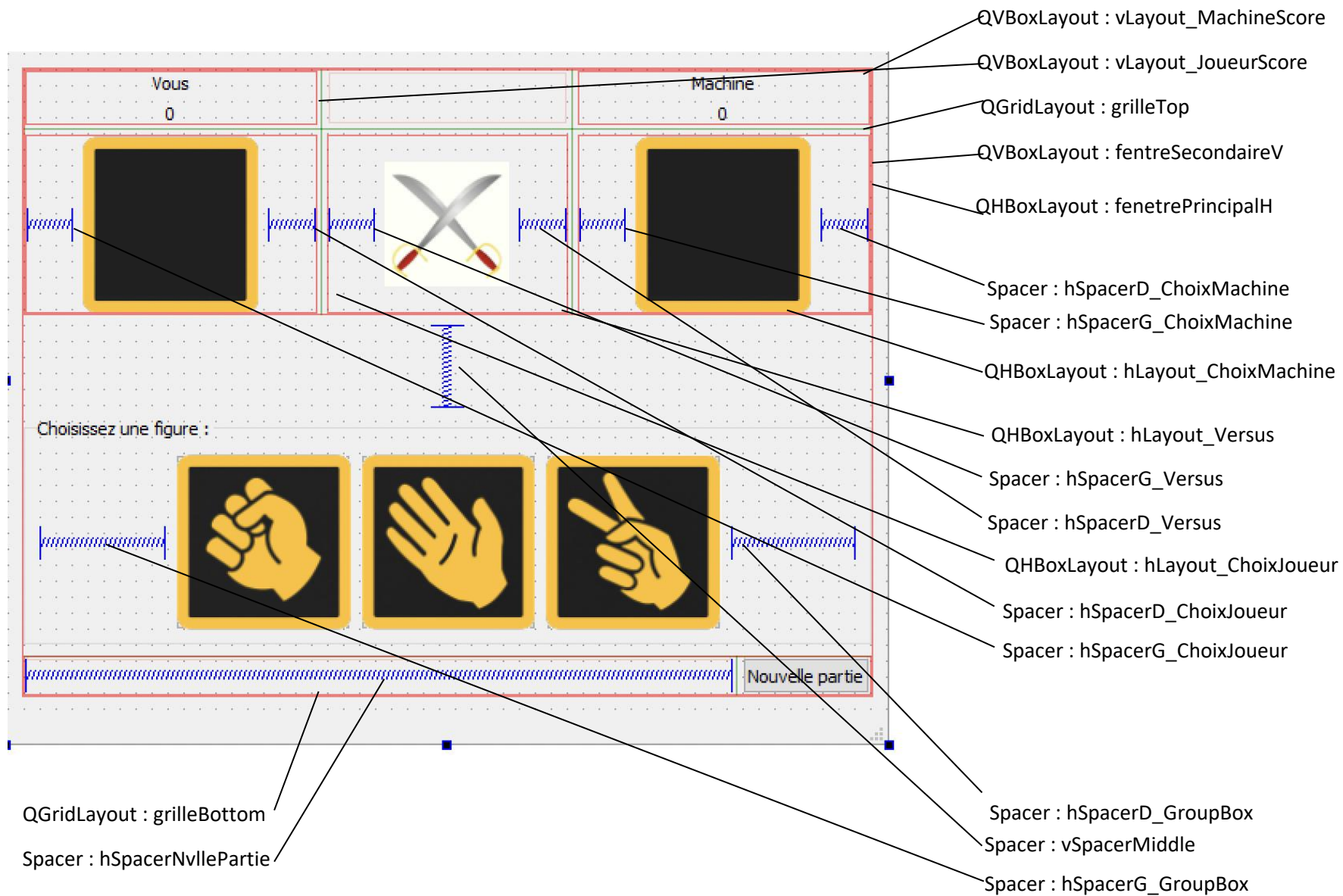
chifoumievue.ui : fichier répertoriant tous les éléments d'interfaces et leurs dispositions fait avec QDesigner

main.cpp : fichier source contenant la boucle principale d'exécution de l'application et secondaire d'attente des messages

ressourcesChifoumi.qrc : fichier qui répertorie toutes les sources de l'application et notamment des images

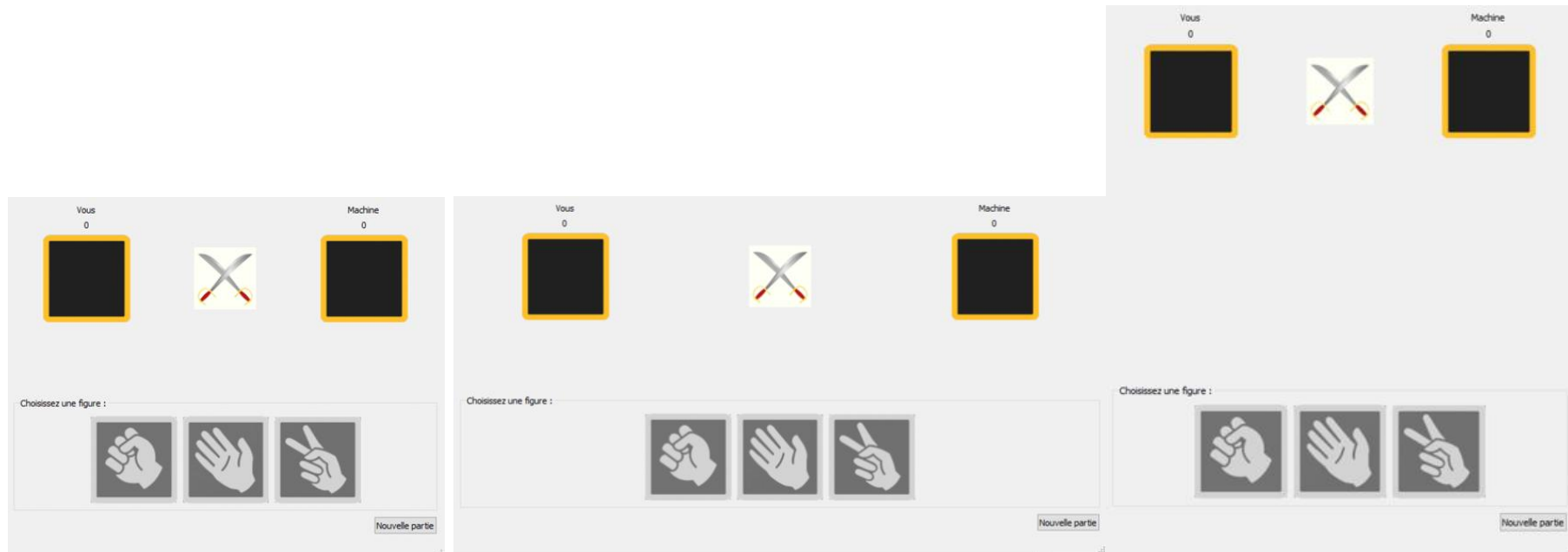
### 3-Element d'interfaces :





#### 4-Test de l'application Chifoumi :

Tests de redimensionnement :



Taille normale

Étirement horizontal

Étirement vertical



## Tests des activités :

| Test                  | Résultat attendu                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Validation |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| Activité 1            | <p><u>Système d'information :</u><br/>Le jeu est initialisé avec les variables coups et scores du joueur et de la machine prenant les valeurs de l'état initial</p> <p><u>Interface :</u><br/>Les boutons de signes sont accessibles pour le choix de l'utilisateur<br/>Les scores et nom des joueurs sont mis avec une couleur bleue pour indiquer que le jeu est en cours<br/>Le focus est sur le bouton 'NvllePartie'</p>                                                                                                             | Ok         |
| Activité 2            | <p><u>Système d'Information :</u><br/>Le dernier coup du jour est mis à jour en fonction du coup joué.<br/>La machine joue (tirage aléatoire) et le dernier coup de la machine est mis à jour en conséquence.<br/>La machine détermine le gagnant et met à jour en conséquence les scores.</p> <p><u>Interface :</u><br/>Le signe choisi par le joueur s'affiche dans la case d'affichage prévu, de même pour celui de la machine<br/>Les scores sont mis à jour en fonction du gagnant<br/>Le focus est sur le bouton 'NvllePartie'</p> |            |
| Activité 2.1 (Pierre) | //                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Ok         |
| Activité 2.2 (Papier) | //                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Ok         |
| Activité 2.3 (Ciseau) | //                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Ok         |

|            |                                                                                                                                                                                                                                                                                                              |    |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Activité 3 | <p><u>Système d'Information :</u><br/> Les scores Joueur/Machine sont mis à 0.<br/> Les derniers coups Joueur/Machine joués sont mis à rien.</p> <p><u>Interface :</u><br/> Les zones d'affichage sont mises en cohérences avec les propriétés précédentes.<br/> Le focus est sur le bouton 'NvlePartie'</p> | OK |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|