

Dupin Rémi

Errecarret Leho

Cattarossi Thomas



**- S2.04 -**

# **Dossier analyse & conception V2**

Auteur : Rémi Dupin

Date : 21/05/2022

Dans cette version les fichiers sources sont :

images	17/05/2022 21:37	Dossier de fichiers	
.gitignore	17/05/2022 22:01	Document texte	1 Ko
chifoumi.pro	13/05/2022 15:15	Fichier PRO	1 Ko
chifoumi.pro.user	17/05/2022 21:39	Per-User Project O...	17 Ko
chifoumivue.cpp	17/05/2022 19:42	C++ Source File	5 Ko
chifoumivue.h	17/05/2022 18:02	Fichier H	1 Ko
chifoumivue.ui	11/05/2022 08:22	Fichier UI	12 Ko
main.cpp	13/05/2022 15:30	C++ Source File	1 Ko
modele.cpp	17/05/2022 19:42	C++ Source File	3 Ko
modele.h	17/05/2022 19:42	Fichier H	3 Ko
presentation.cpp	17/05/2022 19:43	C++ Source File	4 Ko
presentation.h	17/05/2022 19:42	Fichier H	1 Ko
ressource.qrc	11/05/2022 08:22	Fichier QRC	1 Ko

Figure 1 Listes de tous les fichiers sources de la V2

- Le dossier images contient toutes les images à pour rôles de contenir toutes les images dont nous avons besoin pour notre projet.
- Le fichier .gitignore générer à l'aide de ' fork git ' nous permet de filtrer les fichiers qu'on sauvegarde pour un commit sur git hub.
- Le fichier chifoumi.pro est le fichier de configuration de Qt lorsque qu'on change le projet d'ordinateur.
- Le fichier chifoumivue.cpp contient la définition de toutes les fonctions ou slots déclaré dans le .h
- Le fichier chifoumivue.h contient la déclaration des divers fonction utiles au programme et à chifoumivue.cpp.
- chifoumivue.ui est le fichier permettant de générer l'interface graphique de l'application
- main.cpp est le fichier C++ contenant la déclaration des fenêtre et l'association des fenêtre au programme.
- modele.cpp contient la définition de toutes les fonctions ou slots déclaré dans le .h
- modele.h contient la déclaration des divers fonction utiles au programme et à modele.cpp.
- présentation.cpp contient la définition de toutes les fonctions ou slots déclaré dans le .h
- presentation.h contient la déclaration des divers fonction utiles au programme et à modele.cpp.
- ressource.qrc est un fichier généré avec Qt qui nous permet d'ajouter les images contenues dans le dossier image aux ressources du projet.

```

#ifndef CHIFOUMIVUE_H
#define CHIFOUMIVUE_H

#include <QMainWindow>
#include "modele.h"
#include "Presentation.h"

QT_BEGIN_NAMESPACE
namespace Ui { class chifoumiVue; }
QT_END_NAMESPACE

class Presentation;
class chifoumiVue : public QMainWindow
{
    Q_OBJECT

public:
    chifoumiVue(QWidget *parent = nullptr);
    ~chifoumiVue();
    void setPresentation(Presentation *p);
    Presentation * getPresentation();
    void afficherScoreJoueur(unsigned int scoreJ);
    void afficherScoreMachine(unsigned int scoreM);
    void afficherCoupJoueur(Modele::UnCoup coupJ);
    void afficherCoupMachine(Modele::UnCoup coupM);
    void majInterface(Presentation::UnEtatJeu e);
    QString afficherNomCoup (Modele::UnCoup c);

private slots:
    void demarrerPartie();
    void coupPierre();
    void coupCiseau();
    void coupPapier();

    /** Attributs du Modèle
private:
    Presentation *laPresentation;
    Ui::chifoumiVue *ui;

};
#endif // CHIFOUMIVUE_H

```

Figure 2 Présentation du .h de chifoumivue

- La photo si contre nous présente le .h de la classe chifoumivue cette classe est donc la vue du modèle MVP.

- Dans cette classe le MVP est mis en œuvre de tel sorte que la vue communique avec la classe présentation par le biais du pointeur \*laPresentation.

- Dans la vue nous créons une « fausse » classe présentation qui nous permet de créer un lien et d'éviter que le programme n'accepte pas la déclaration du pointeur de la présentation.

- Ici nous avons défini 4 slots qui nous permet de démarrer la partie et de savoir quel coup le joueur à jouer.

- Pour effectuer le lien entre la vue et la présentation nous avons ici défini des méthodes setPresentation et getPresentation et nous nous servons du pointeur pour les initialiser.

- Pour toutes les autres fonctions déclarées elle nous serve au bon déroulement de la partie et elle nous permette de gérer l'affichage des différentes valeurs.

```

#ifndef PRESENTATION_H
#define PRESENTATION_H
#include "modele.h"

class chifoumiVue;
class Presentation
{
public:
    enum UnEtatJeu {etatInitial, partieEnCours};
    Presentation();
    void deroulementPartie();

    void setModele(Modele * m);
    Modele * getModele();

    void setVue(chifoumiVue * v);
    chifoumiVue * getVue();

    void demarrerPartie();
    void coupPierre();
    void coupCiseau();
    void coupPapier();
    void coupMachine();
    void majScore(char p_gagnant);

    void setEtat(UnEtatJeu e);
    UnEtatJeu getEtat();

private:
    UnEtatJeu etat; //état du jeu par rapport au diagramme E.T
    chifoumiVue *laVue;
    Modele *leModele;
};

#endif // PRESENTATION_H

```

Ci-contre nous avons le fichier présentation.h

Ce fichier nous montre comment la classe présentation gère le modèle MVP

- Nous avons ici une méthode setModele et une autre méthode getModele pour pouvoir faire le lien avec le fichier modele.h ces 2 méthodes utilisent tous deux un pointeur vers la classe modele.h, le pointeur est défini en bas de ce fichier dans les méthodes privée
- Nous avons dans ce fichier 2 autres méthodes, une qui s'appelle setVue et une autre qui s'appelle getVue, ces 2 méthodes nous servent a faire le lien vers la classe chifoumiVue à l'aide la aussi d'un pointeur vers la classe chifoumiVue.
- les autres méthodes de cette classe nous servent au bon déroulement du jeu.

Figure 3 Présentation du fichier présentation.h

- La classe modele.h n'a pas été changé pour la création du modèle MVP.

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

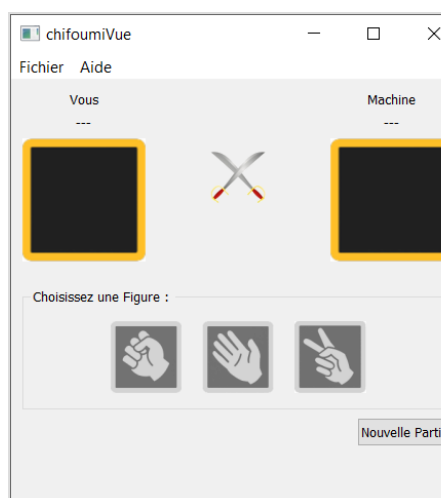


Figure 4 Etat initial de l'application

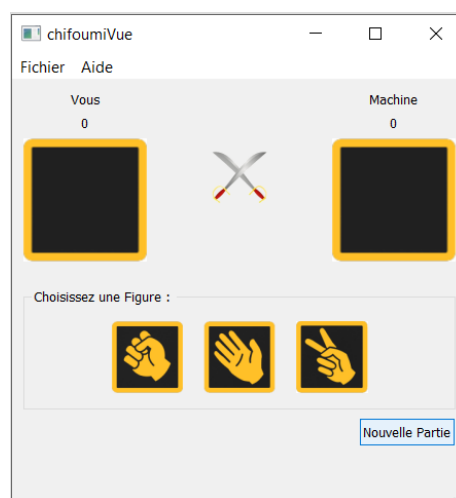


Figure 5 Bouton Nouvelle Partie cliqué

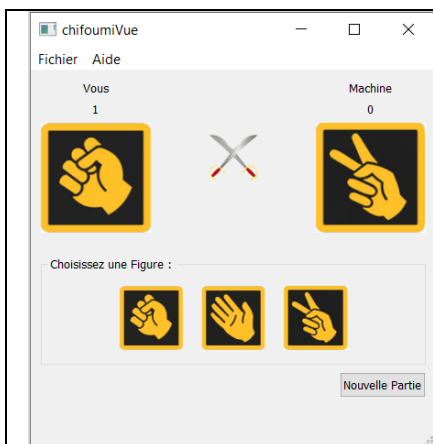


Figure 6 Cas où le joueur gagne avec pierre

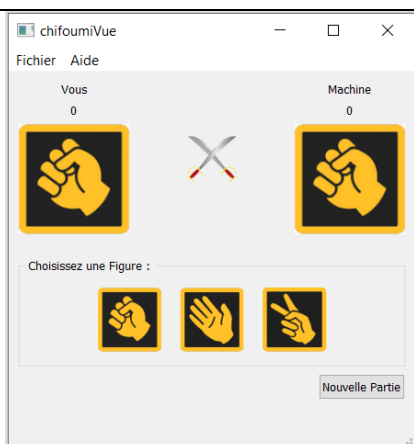


Figure 7 Cas où le joueur fait égalité avec pierre

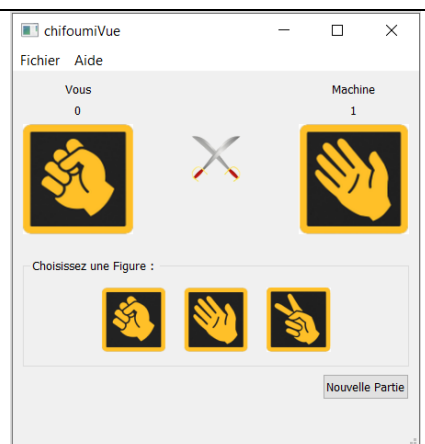


Figure 8 Cas où le joueur perd avec pierre

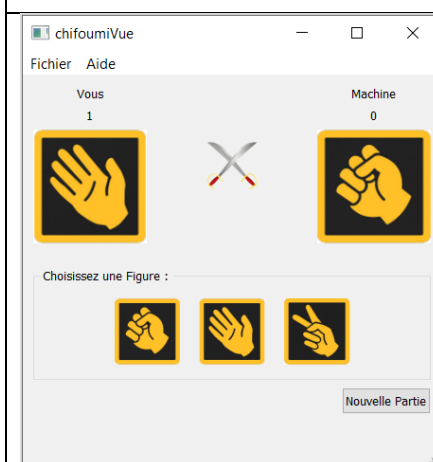


Figure 9 Cas où le joueur gagne avec feuille

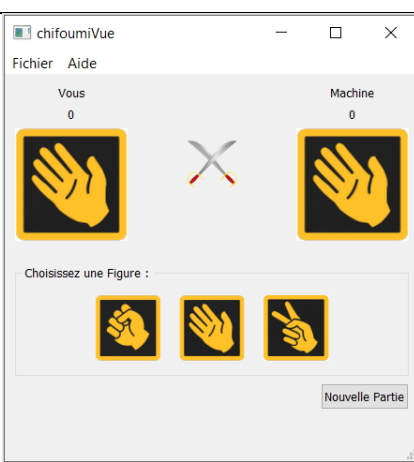


Figure 10 Cas où le joueur fait égalité avec feuille

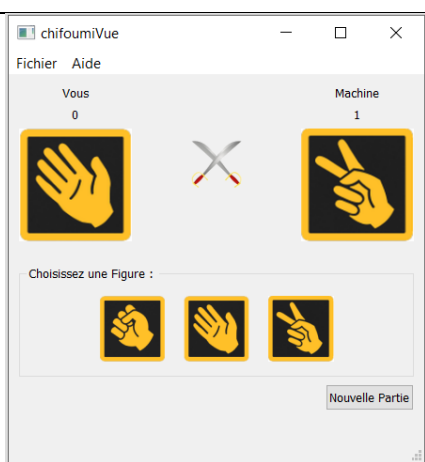


Figure 11 Cas où le joueur perd avec feuille

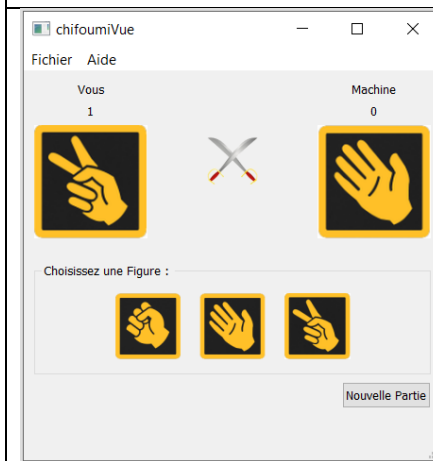


Figure 12 Cas où le joueur gagne avec ciseau

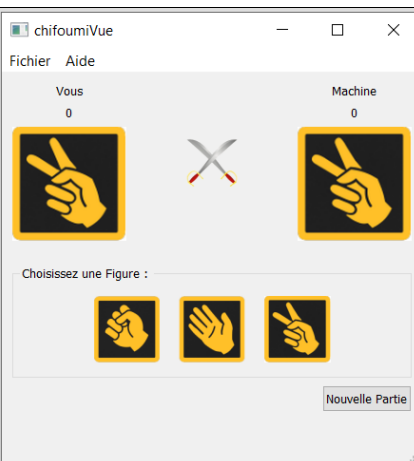


Figure 13 Cas où le joueur fait égalité avec ciseau

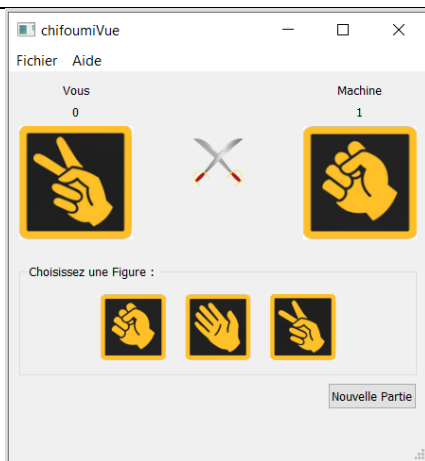


Figure 14 Cas où le joueur perd avec ciseau