



KOFFI Jean-Jonathan  
ROBERT Quentin  
MOPTY Maxime  
TD2-TP3  
But Informatique première année

url GitHub : <https://github.com/jeanJonathan/Situation-d-analyse-et-d-evaluation-de-Developpement-D-application>

## SAE 2.01 – Développement d’une application

### Chifoumi – Dossier d’Analyse et conception

## Table des matières

1. Compléments de spécifications externes.....	6
1.1.....	6
2. Diagramme des Cas d'Utilisation.....	6
1.2 .....	6
3. Scénarios .....	6
1.3.....	6
4. Diagramme de classe (UML) .....	6
1.3.1 .....	9
Version v0 .....	10
5. Implémentation et tests.....	10
5.1 Implémentation .....	10
Liste des fichiers de cette version :.....	10
- chifoumi.h : .....	10
- chifoumi.cpp : .....	10
Respectivement spécification et corps de la classe Chifoumi décrite au paragraphe 4. ....	10
5.2 Test.....	10
Version v1 .....	11
6. Classe Chifoumi : Diagramme états-transitions .....	11
(b) Dictionnaires des états, événements et Actions .....	12
7. Éléments d'interface .....	13
8. Implémentation et tests.....	14
8.1 Implémentation .....	14
Liste des fichiers sources et rôle de chaque fichier .....	14
modele.h : .....	14
Contenant l'ensemble des informations permettant de programmer le jeu .....	14
chifoumivue.h : .....	14
Fichier servant de moteur pour le jeu. Il assure également la gestion de l'interface graphique.....	14
main.cpp.....	14
Fichier permettant d'exécuter le modèle. Et d'exécuter l'application graphique .....	14
chifoumivue.ui.....	14
Répertoire Ressources .....	14
Dossier regroupant l'ensemble des images servant à compléter l'interface graphique .....	14
Commenter brièvement les choix importants d'implémentation réalisés, comme par exemple, les signals/slots.....	15
Explication de l'implémentation des signals/slot Tableau ( .....	15
Signal : click .....	15
Slot : DemanderNewPartie.....	15

Signal : click .....	15
Slot : Choix figure (Changement de l'image figureJoueur en bouton selon le choix du joueur) .....	15
8.2 Test.....	15
les tests prévus / réalisés pour montrer : .....	15
- Le comportement fonctionnel du programme.....	15
- Le comportement de l'interface non lié aux aspects fonctionnels du programme.....	15
Version v2 .....	20
1. Classe Chifoumi : Diagramme états-transitions .....	20
(e) Dictionnaires des états, événements et Actions .....	21
2. Éléments d'interface .....	22
3. Implémentation et tests.....	23
8.1 Implémentation .....	23
Liste des fichiers sources et rôle de chaque fichier .....	23
modele.h : .....	23
Contenant l'ensemble des informations permettant de programmer le jeu. ....	23
chifoumivue.h : .....	23
Il assure également la gestion de l'interface graphique. ....	23
presentation.h : .....	23
Fichier servant de moteur pour le jeu. ....	23
main.cpp.....	23
Fichier permettant d'exécuter le modèle. Et d'exécuter l'application graphique .....	23
chifoumivue.ui.....	23
Répertoire Ressources .....	23
Dossier regroupant l'ensemble des images servant à compléter l'interface graphique .....	24
Commenter brièvement les choix importants d'implémentation réalisés, comme par exemple, les signals/slots.....	24
Explication de l'implémentation des signals/slot Tableau ( .....	24
Signal : click .....	24
Slot : DemanderNewPartie.....	24
Signal : click .....	24
Slot : Choix figure (Changement de l'image figureJoueur en bouton selon le choix du joueur) .....	24
8.2 Test.....	24
A faire : .....	24
Décrire les tests prévus / réalisés pour montrer : .....	24
- Le comportement fonctionnel du programme.....	24
- Le comportement de l'interface non lié aux aspects fonctionnels du programme.....	24
Version v3 .....	29
1. Classe Chifoumi : Diagramme états-transitions .....	29
(h) Dictionnaires des états, événements et Actions .....	30

2. Éléments d'interface) .....	31
3. Implémentation et tests .....	32
8.1 Implémentation .....	32
Liste des fichiers sources et rôle de chaque fichier .....	32
modele.h : .....	32
Contenant l'ensemble des informations permettant de programmer le jeu. ....	32
chifoumivue.h : .....	32
Il assure également la gestion de l'interface graphique. ....	32
presentation.h : .....	32
Fichier servant de moteur pour le jeu. ....	32
main.cpp.....	32
Fichier permettant d'exécuter le modèle. Et d'exécuter l'application graphique .....	32
chifoumivue.ui.....	32
Répertoire Ressources .....	32
Dossier regroupant l'ensemble des images servant à compléter l'interface graphique .....	33
Implémentation du signals/slots .....	33
connect(ui->actionA_propos_de,SIGNAL(triggered()),this,SLOT(aProposDe()));.....	33
8.2 Test.....	33
- Le comportement fonctionnel du programme .....	33
- Le comportement de l'interface non lié aux aspects fonctionnels du programme.....	33
Version v4 .....	39
1. Classe Chifoumi : Diagramme états-transitions .....	39
(k) Dictionnaires des états, événements et Actions .....	40
2. Éléments d'interface .....	41
3. Implémentation et tests.....	42
8.1 Implémentation .....	42
A faire : .....	42
Lister les fichiers impliqués dans cette version (répertoire, nom de fichier, rôle de chaque fichier) .....	42
Liste des fichiers sources et rôle de chaque fichier .....	42
modele.h : .....	42
Contenant l'ensemble des informations permettant de programmer le jeu. ....	42
chifoumivue.h : .....	42
Il assure également la gestion de l'interface graphique. ....	42
presentation.h : .....	42
Fichier servant de moteur pour le jeu. ....	42
main.cpp.....	42
Fichier permettant d'exécuter le modèle. Et d'exécuter l'application graphique .....	42
chifoumivue.ui.....	42
Répertoire Ressources .....	42

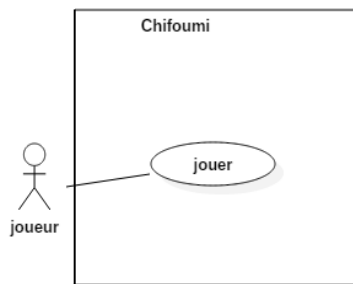
Dossier regroupant l'ensemble des images servant à compléter l'interface graphique .....	42
Commenter brièvement les choix importants d'implémentation réalisés, comme par exemple, les signals/slots .....	43
Explication de l'implémentation des signals/slot Tableau ( .....	43
Signal : click .....	43
Slot : DemanderNewPartie.....	43
Signal : click .....	43
Slot : Choix figure (Changement de l'image figureJoueur en bouton selon le choix du joueur) .....	43
8.2 Test.....	43
A faire : .....	43
Décrire les tests prévus / réalisés pour montrer : .....	43
- Le comportement fonctionnel du programme .....	43
- Le comportement de l'interface non lié aux aspects fonctionnels du programme.....	43

## 1. Compléments de spécifications externes.

*Rien à signaler*

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 création		
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 son 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) Remarque :

- *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.

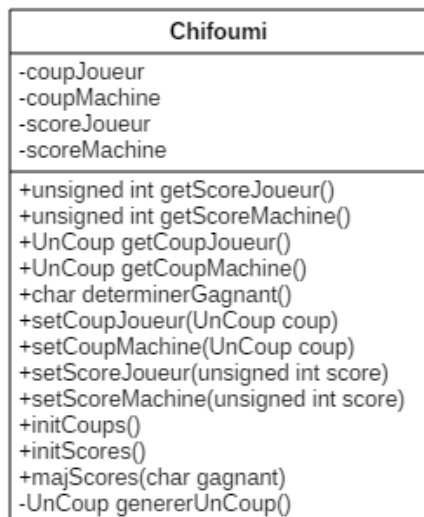


Figure 2 : Diagramme de Classes UML du jeu Chifoumi

(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

 $\} ;$ 

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 – 1h

#### 5.1 Implémentation

Liste des fichiers de cette version :

- chifoumi.h :
- chifoumi.cpp :

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

#### 5.2 Test

Test avec le programme fourni main.cpp

*Valeurs fournies / attendues... comme montré dans la ressource R2.03 (partie tests)*

<i>Classe</i>	<i>Description</i>	<i>Valeur fournie</i>	<i>Résultat attendue</i>	<i>Commentaire</i>
Valide n°1	Test les méthodes get() associer aux attribut 'score'	scoreJoueur : 0 scoreMachine : 0	scoreJoueur : 0 scoreMachine : 0	
Valide n°2	Test les méthodes get() associer aux attribut 'coup'	coupJoueur : rien coupMachine : rien	coupJoueur :rien coupMachine : rien	
Valide n°3	Test les méthodes set() associer aux attribut 'score'	setScore(1) setScore(2)	scoreJoueur : 1 scoreMachine :2	
Valide n°4	Test initScores()	scoreJoueur : 0 scoreMachine :0	scoreJoueur : 0 scoreMachine :0	
Valide n°4 Valide n°5	Test les méthodes set() et get() associés aux attributs 'coup/choix'	setCoupJoueur(pierre) setCoupMachine(ciseau)	coupJoueur :pierre coupMachine : ciseau	
Valide n°6	Test sur l'identification du gagnant et la maj des scores	coupJoueur : papier coupMachine : papier	scoreJoueur = 0 scoreMachine =0	
Valide n°7	Test sur l'identification du gagnant et la maj des scores	coupJoueur : ciseau coupMachine : papier	scoreJoueur = 1 scoreMachine =0	

## Version v1

### 6. Classe Chifoumi : Diagramme états-transitions – 1h 30

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

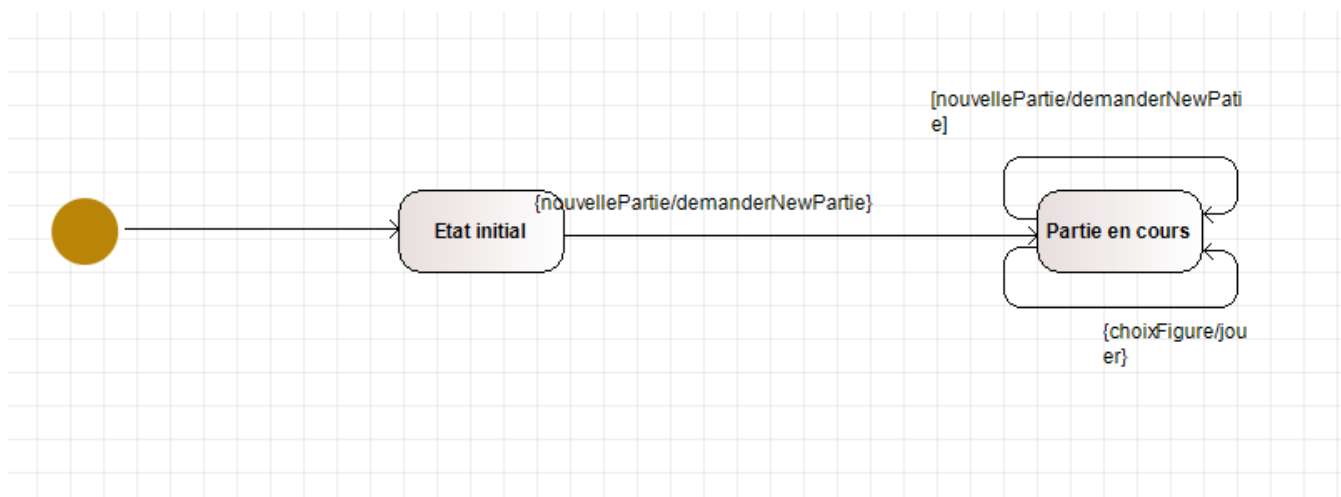


Figure 4 : Diagramme états-transitions

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

### Dictionnaire des états du jeu

<i>nomEtat</i>	<i>Signification</i>
Etat initial	L'état dans lequel l'application se trouve lors de son déclenchement et dans lequel elle se met en attente en réponse à tout événement extérieur
Partie en cours	L'état dans lequel l'application se trouve lorsque l'événement associé au bouton Nouvelle partie est déclenché et que les boutons figures sont actifs

Tableau 3 : États du jeu

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

<i>nomEvénement</i>	<i>Signification</i>
nouvellePartie	L'événement qui permet d'établir la transition entre l'état initial et partie en cours Ou entre la partie en cours à elle-même (lorsque le joueur veut recommencer une nouvelle partie)
choisirFigure	L'événement qui permet au joueur de jouer lorsque les boutons figures sont actifs

Tableau 4 : Événements faisant changer le jeu d'état

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

<i>ActionRealisé</i>	<i>Signification</i>
jouer	Cette action traduit le fait que lorsque le joueur clique sur Nouvelle partie, les boutons-figures sont actifs. Le nom et score du joueur sont de couleur bleue (la partie en cours)
newPartie	Cette action correspond à une affection qui traduit le fait que le joueur décide de recommencer la partie, les scores sont remis à 0
OuvrirFenetre	A la fin du tour du joueur, les nom et score du joueur machine sont de couleur bleue pour dire que c'est le tour de la machine et réciproquement pour le joueur
quitter	L'action conduisant à la sortie de l'application

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

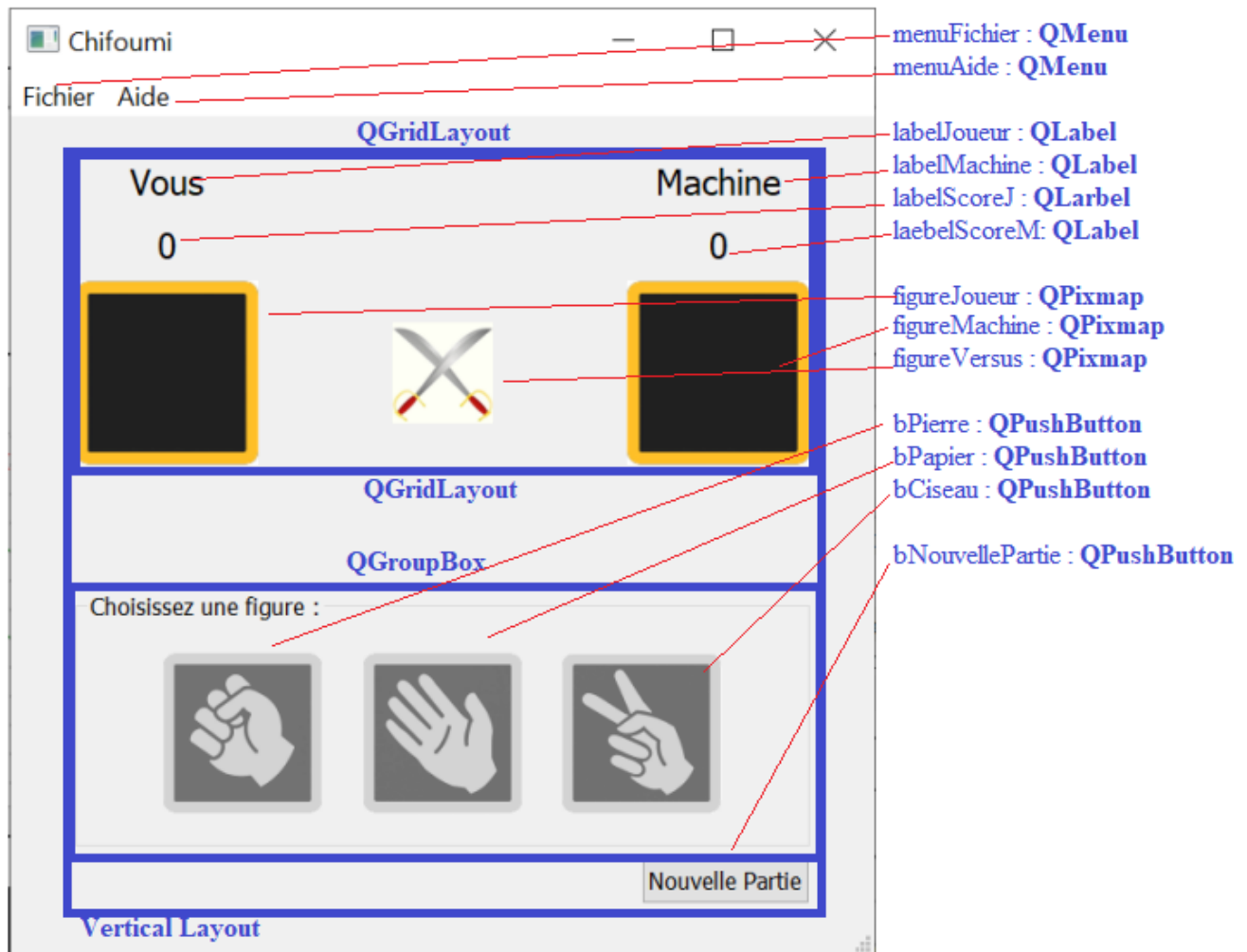
Événement → nomEtatJeu	bNouvellePartie	bPierre	bCiseau	bPapier
	Nouvelle	choixFigure		
Etat initial	EtatInitial / Activité1	-----		
Partie en cours	partieEnCours / activité3	partieEnCours /activité2		

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

Nom Activité	signification
Activité1 = nouvellePartie	Mettre les scores à 0, mettre les figures à rien Afficher ces éléments Rendre les boutons actifs, mettre le focus sur bouton Nouvelle Partie
Activité2 = choixFigure	Met à jour la figure du joueur Le programme choisir une figure aléatoire Mise à jour de la figure de la machine Détermination du gagnant Mise à jour les scores Affichage de ces valeurs (scores et figures) Mettre le focus sur bouton Nouvelle Partie
Activité3 = nouvellePartie	Mettre les scores à 0, mettre les figures à rien Afficher ces éléments Mettre le focus sur bouton Nouvelle Partie

*L'intérêt de cette vue matricielle est qu'elle permet une préparation naturelle et aisée de l'étape suivante de programmation.*

## 7. Éléments d'interface – 1h



## 8. Implémentation et tests – 2h

### 8.1 Implémentation

#### Liste des fichiers sources et rôle de chaque fichier

**modele.h :**

*Contenant l'ensemble des informations permettant de programmer le jeu*

**chifoumivue.h :**

*Fichier servant de moteur pour le jeu. Il assure également la gestion de l'interface graphique*

**main.cpp**

*Fichier permettant d'exécuter le modèle. Et d'exécuter l'application graphique*

**chifoumivue.ui**

#### Répertoire Ressources

*Dossier regroupant l'ensemble des images servant à compléter l'interface graphique*

*Commenter brièvement les choix importants d'implémentation réalisés, comme par exemple, les signals/slots*

*Explication de l'implémentation des signals/slot Tableau (*

*Signal : click*

*Slot : DemanderNewPartie*

*Signal : click*

*Slot : Choix figure (Changement de l'image figureJoueur en bouton selon le choix du joueur)*

## 8.2 Test

*les tests prévus / réalisés pour montrer :*

*- Le comportement fonctionnel du programme*

*- Le comportement de l'interface non lié aux aspects fonctionnels du programme*

Interaction réaliser	Action attendue	Action réaliser
Etat initial	Afficher les scores à 0 Afficher les coups à rien Rendre les figures actives Focus sur le bouton nouvellePartie	Voir la capture Figure 2
Partie en cours : ChoixFigure	Afficher le coup du joueur Choisir un coup pour le système et l'afficher Déterminer le gagnant, mettre à jour les scores et les afficher Focus sur le bouton nouvellePartie	Voir la capture Figure 3
Partie en cours : nouvelleParite	Afficher les scores à 0 Afficher les coups à rien Rendre les figures actives Focus sur le bouton nouvellePartie	Voir la capture Figure 2

Comportement fonctionnel du programme lors de la réalisation des tests

*--Etat des éléments graphiques lorsque le jeu d'état du programme est dans l'état « Etat initial »*

*Explication de l'inactivation des boutons avant le déclenchement du slot nouvelle partie*

-----

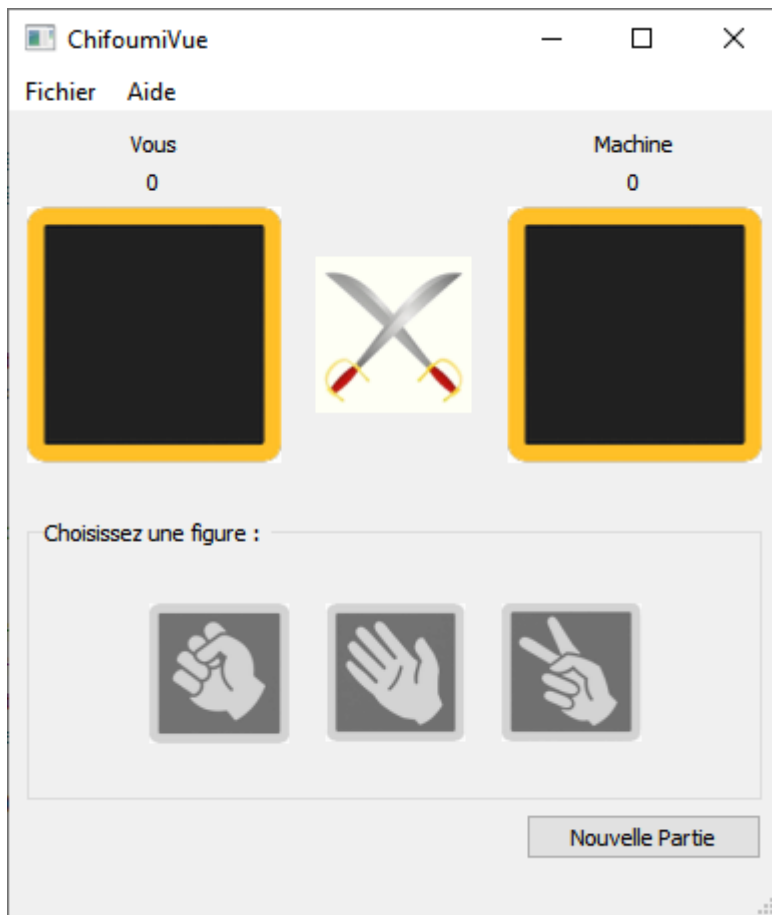


Figure 1 :

*Explication de l'activation des boutons après le déclenchement du slot nouvelle partie*



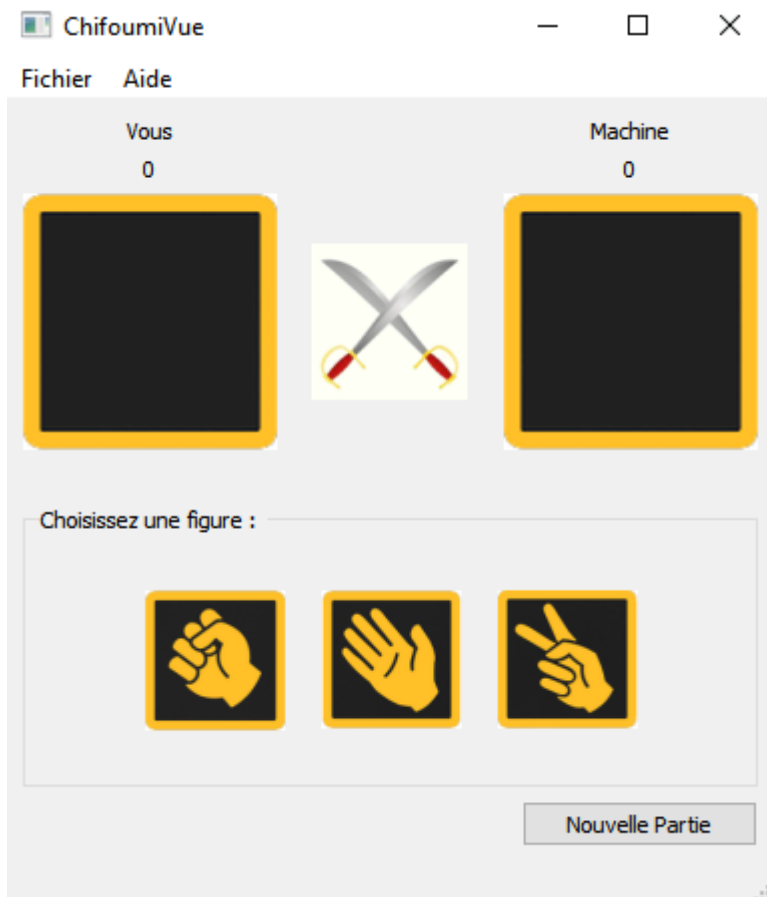


Figure 2 :

--Etat des éléments graphiques lorsque le jeu d'état du programme est dans l'état « Partie en Cours »

Cas d'exemple : Lorsque le joueur décide de jouer feuille (déclenchement du slot choixfeuille())

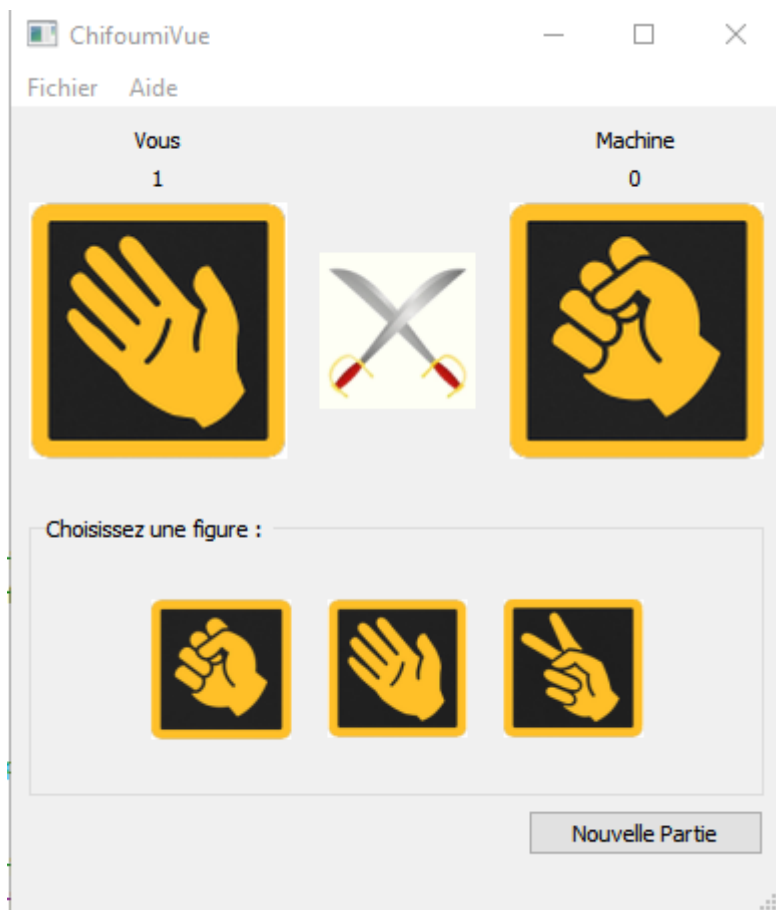


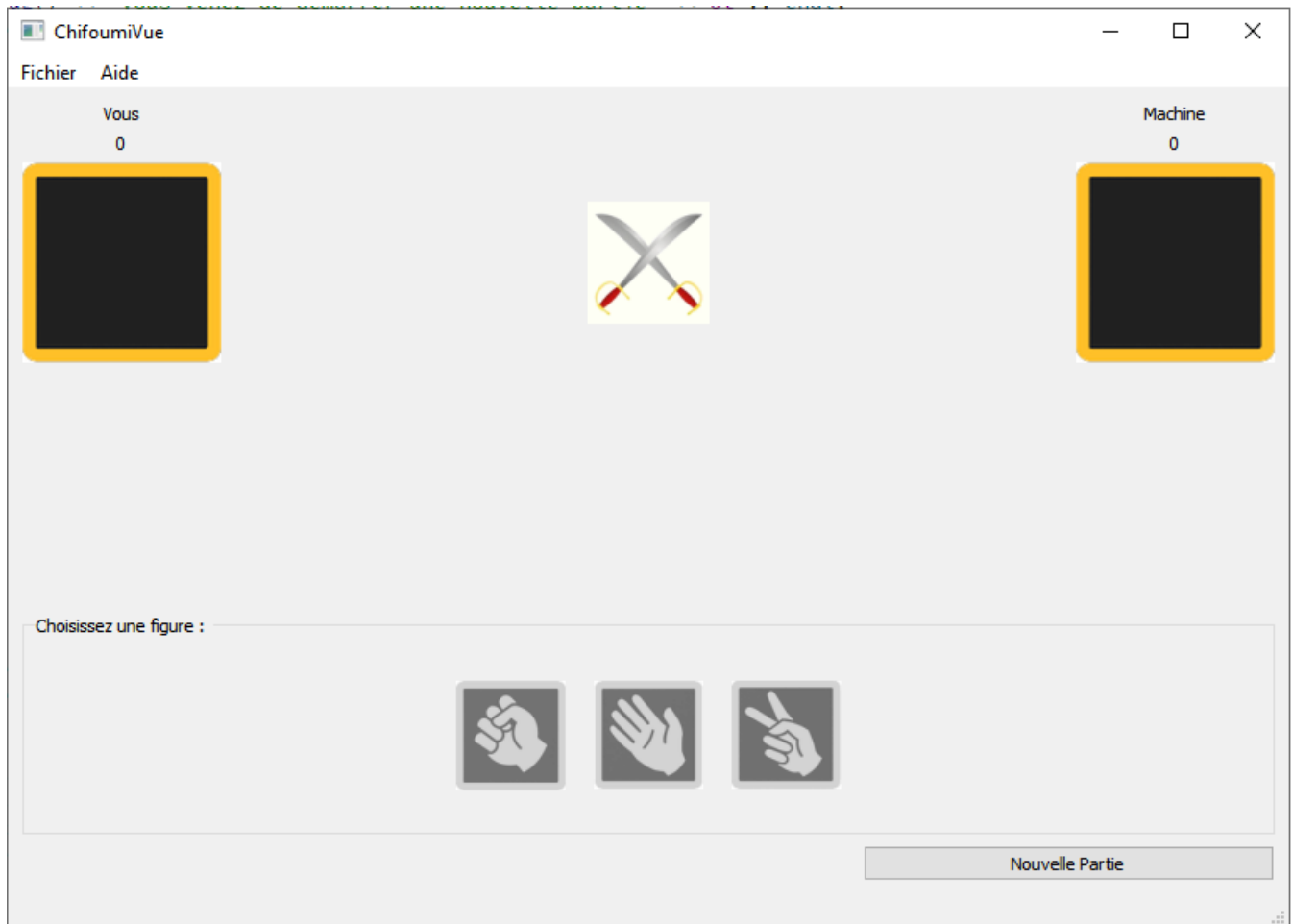
Figure 3 :

Comportement de l'interface (non lié aux aspects fonctionnels du programme) lors de la réalisation des tests

--Etat des éléments graphiques avant le redimensionnement

(Voir Figure 1 Pour observer l'Interface dans l'état initial de l'application)

--Etat des éléments graphiques lors du redimensionnement



*Figure 4 :*  
*Explication orale des règles ergonomiques qui ont motivé ce choix.*

## Version v2

### 1. Classe Chifoumi : Diagramme états-transitions – 2h

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

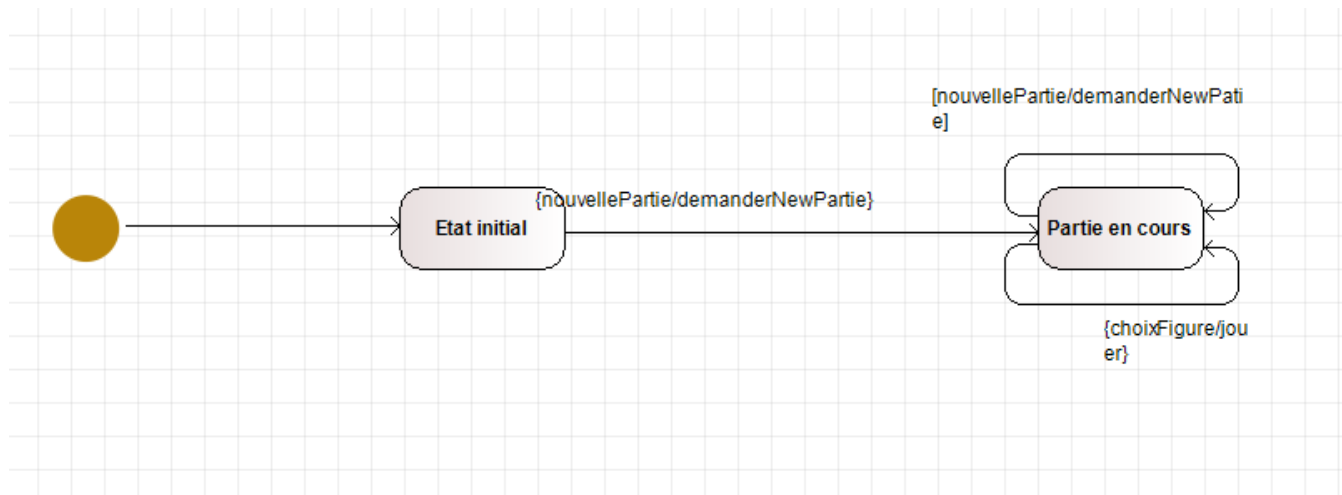


Figure 1 : Diagramme états-transitions

(e) Dictionnaires des états, événements et Actions

**Dictionnaire des états du jeu**

<i>nomEtat</i>	<i>Signification</i>
Etat initial	L'état dans lequel l'application se trouve lors de son déclenchement et dans lequel elle se met en attente et en réponse à tout événement extérieur
Partie en cours	L'état dans lequel l'application se trouve lorsque l'événement associé au bouton Nouvelle partie est déclenché et que les boutons figures sont actifs

Tableau 7 : États du jeu

**Dictionnaire des événements faisant changer le jeu d'état**

<i>nomEvénement</i>	<i>Signification</i>
CliquerSurNouvellePartie partie	L'événement qui permet d'établir la transition entre l'état initial et partie en cours Ou entre la partie en cours à elle-même (lorsque le joueur veut recommencer une nouvelle partie)
ChoisirFigure	L'événement qui permet au joueur de jouer lorsque les boutons figures sont actifs

Tableau 8 : Événements faisant changer le jeu d'état

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

<i>ActionRealisé</i>	<i>Signification</i>
jouer	Cette action traduit le fait que lorsque le joueur clique sur Nouvelle partie, les boutons-figures sont actifs. Le nom et score du joueur sont de couleur bleue (la partie en cours)
newPartie	Cette action correspond à une affection qui traduit le fait que le joueur décide de recommencer la partie, les scores sont remis à 0
OuvrirFenetre	A la fin du tour du joueur, les nom et score du joueur machine sont de couleur bleue pour dire que c'est le tour de la machine et réciproquement pour le joueur
quitter	L'action conduisant à la sortie de l'application

Tableau 9 : Actions à réaliser lors des changements d'état

(f) 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> → <i>nomEtatJeu</i>	bNouvellePartie	bPierre	bCiseau	bPapier
---	-----------------	---------	---------	---------

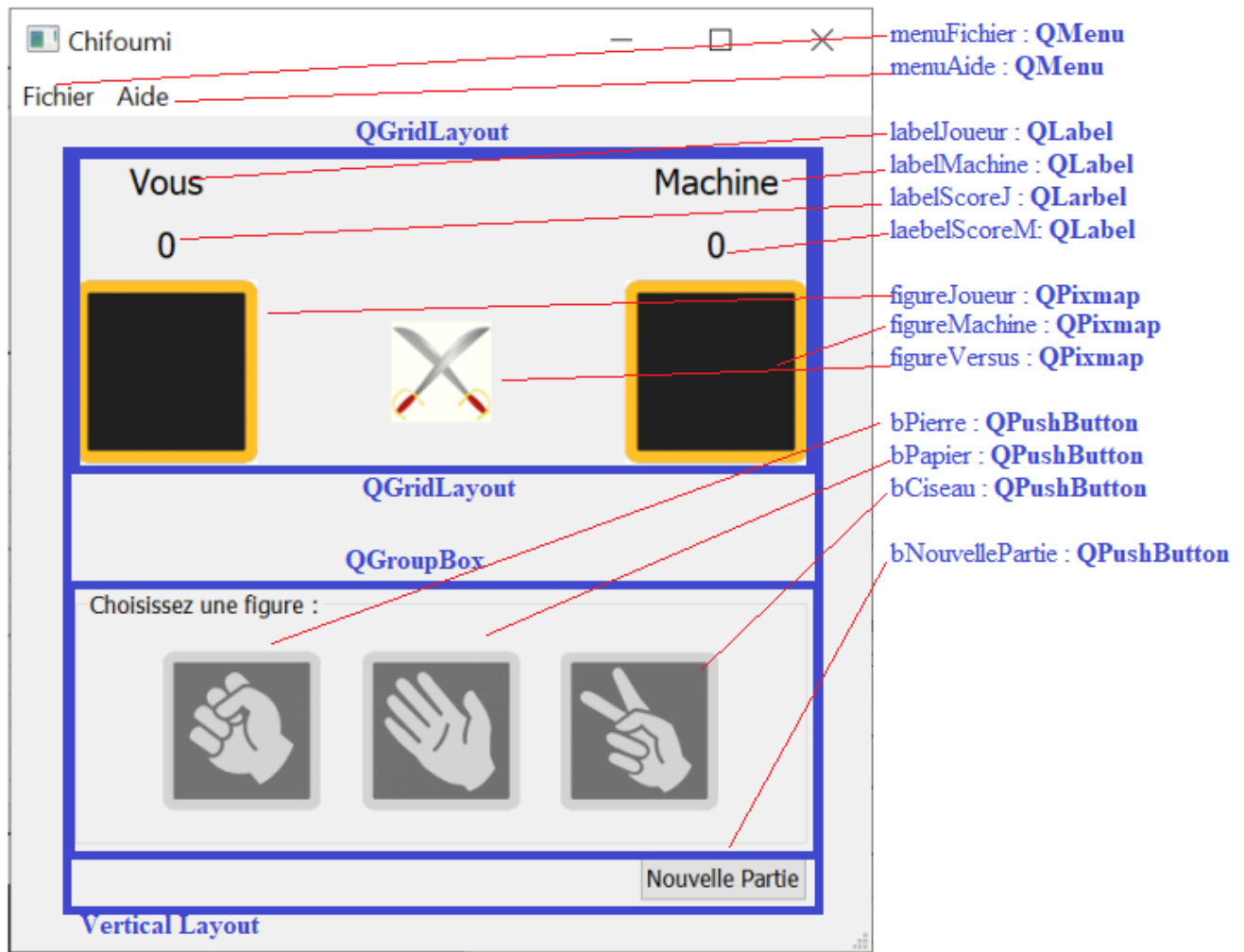
	nouvellePartie	choixFigure
Etat initial	partieEnCours / activité1	-----
Partie en cours	partieEnCours / activité3	partieEnCours / activité2

Tableau 10 : Matrice d'états-transitions du jeu chifoumi

Nom Activité	signification
Activité1	Mettre les scores à 0, mettre les figures à rien Afficher ces éléments Rendre les boutons actifs, mettre le focus sur bouton Nouvelle Partie
Activité2	Met à jour la figure du joueur On tire une figure aléatoire Met à jour la figure de la machine Déterminer le gagnant Met à jour les scores Afficher ces valeurs (scores et figures) Mettre le focus sur bouton Nouvelle Partie
Activité3	Mettre les scores à 0, mettre les figures à rien Afficher ces éléments Mettre le focus sur bouton Nouvelle Partie

*L'intérêt de cette vue matricielle est qu'elle permet une préparation naturelle et aisée de l'étape suivante de programmation.*

## 2. Éléments d'interface – 5 min



### 3. Implémentation et tests – 3h

#### 8.1 Implémentation

Liste des fichiers sources et rôle de chaque fichier

**modele.h :**

*Contenant l'ensemble des informations permettant de programmer le jeu.*

**chifoumivue.h :**

*Il assure également la gestion de l'interface graphique.*

**presentation.h :**

*Fichier servant de moteur pour le jeu.*

**main.cpp**

*Fichier permettant d'exécuter le modèle. Et d'exécuter l'application graphique*

**chifoumivue.ui**

#### Répertoire Ressources

## **Dossier regroupant l'ensemble des images servant à compléter l'interface graphique**

*Commenter brièvement les choix importants d'implémentation réalisés, comme par exemple, les signals/slots*

*Explication de l'implémentation des signals/slot Tableau (*

*Signal : click*

*Slot : DemanderNewPartie*

*Signal : click*

*Slot : Choix figure (Changement de l'image figureJoueur en bouton selon le choix du joueur)*

### **8.2 Test**

*A faire :*

*Décrire les tests prévus / réalisés pour montrer :*

- Le comportement fonctionnel du programme*
- Le comportement de l'interface non lié aux aspects fonctionnels du programme*

Interaction réaliser	Action attendue	Action réaliser
Etat initial	Afficher les scores à 0 Afficher les coups à rien Rendre les figures actives Focus sur le bouton nouvellePartie	Voir la capture Figure 2
Partie en cours : Choix de figure	Afficher le coup du joueur Choisir un coup pour le système et l'afficher Déterminer le gagnant, mettre à jour les scores et les afficher Focus sur le bouton nouvellePartie	Voir la capture Figure 3
Partie en cours : Click sur nouvelle partie	Afficher les scores à 0 Afficher les coups à rien Rendre les figures actives Focus sur le bouton nouvellePartie	Voir la capture Figure 2

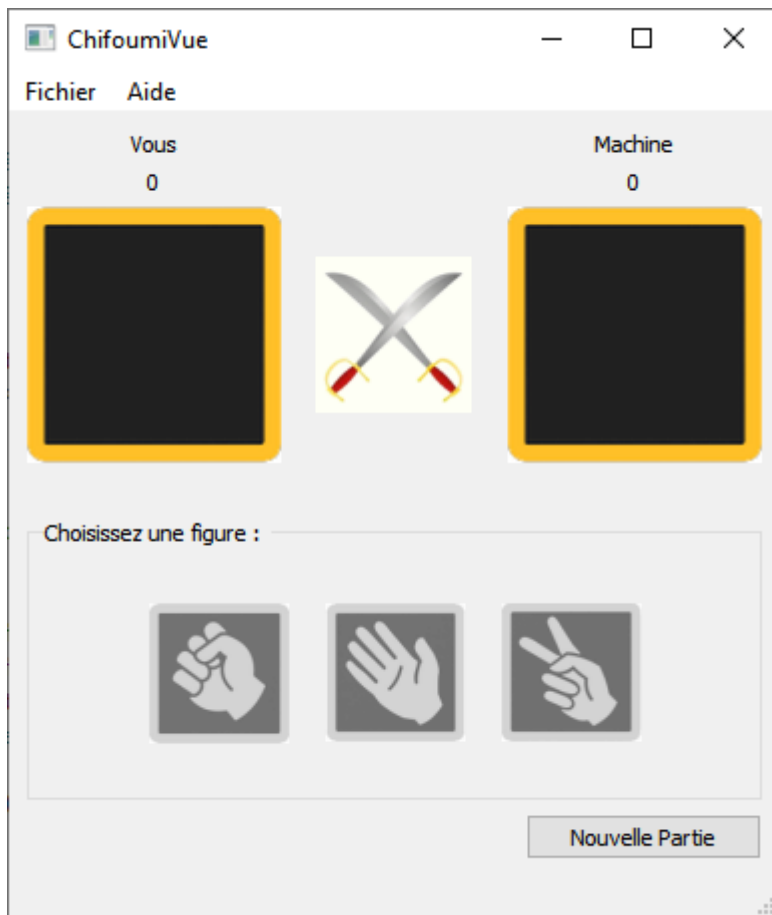
Comportement fonctionnel du programme lors de la réalisation des tests

*--Etat des éléments graphiques lorsque le jeu d'état du programme est dans l'état « Etat initial »*

*Explication de l'inactivité des boutons avant le déclenchement du slot nouvelle partie*

-----





*Figure 1 :*

*Explication de l'activité des boutons après le déclenchement du slot nouvelle partie*

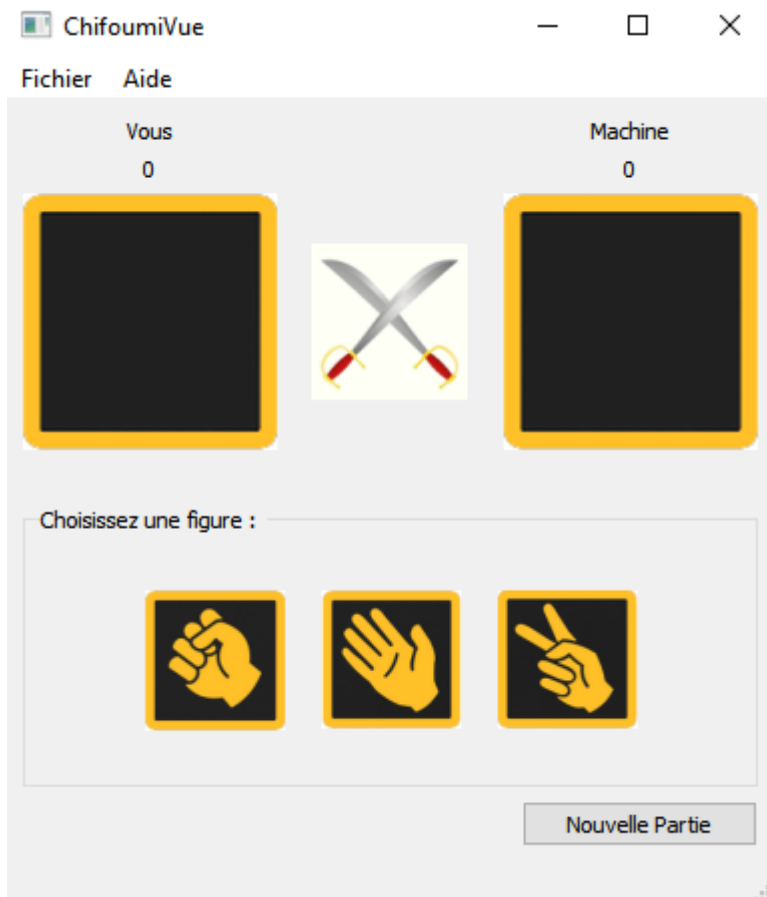


Figure 2 :

--Etat des éléments graphiques lorsque le jeu d'état du programme est dans l'état « Partie en Cours »

Cas d'exemple : Lorsque le joueur décide de jouer feuille (déclenchement du slot choixfeuille())



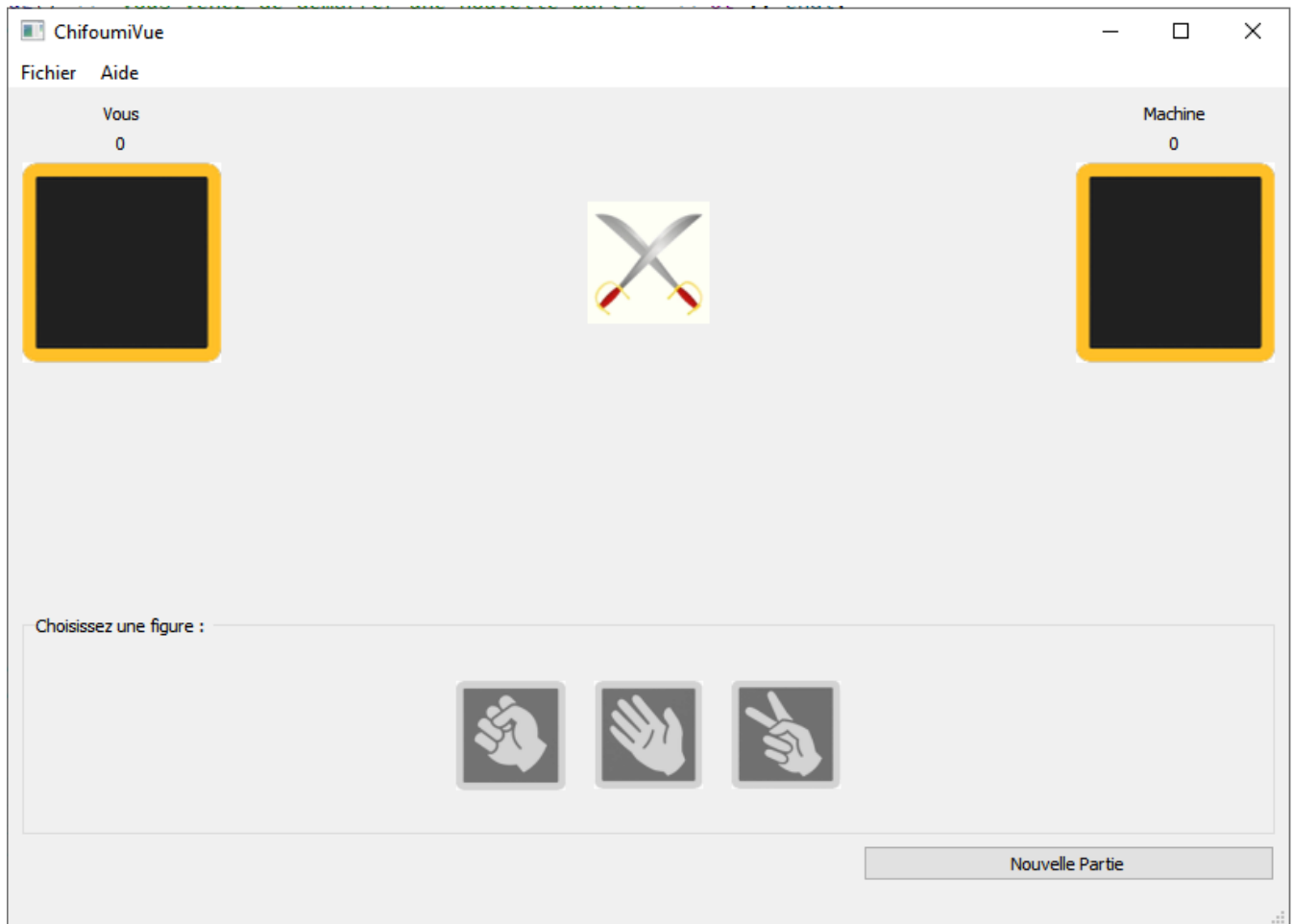
Figure 3 :

Comportement de l'interface (non lié aux aspects fonctionnels du programme) lors de la réalisation des tests

--Etat des éléments graphiques avant le redimensionnement

(Voir Figure 1 Pour observer l'Interface dans l'état initial de l'application)

--Etat des éléments graphiques lors du redimensionnement



*Figure 4 :*  
*Citer et Expliquer* les règles ergonomiques qui ont motivé ce choix.

## 1. Classe Chifoumi : Diagramme états-transitions – 1h 30

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

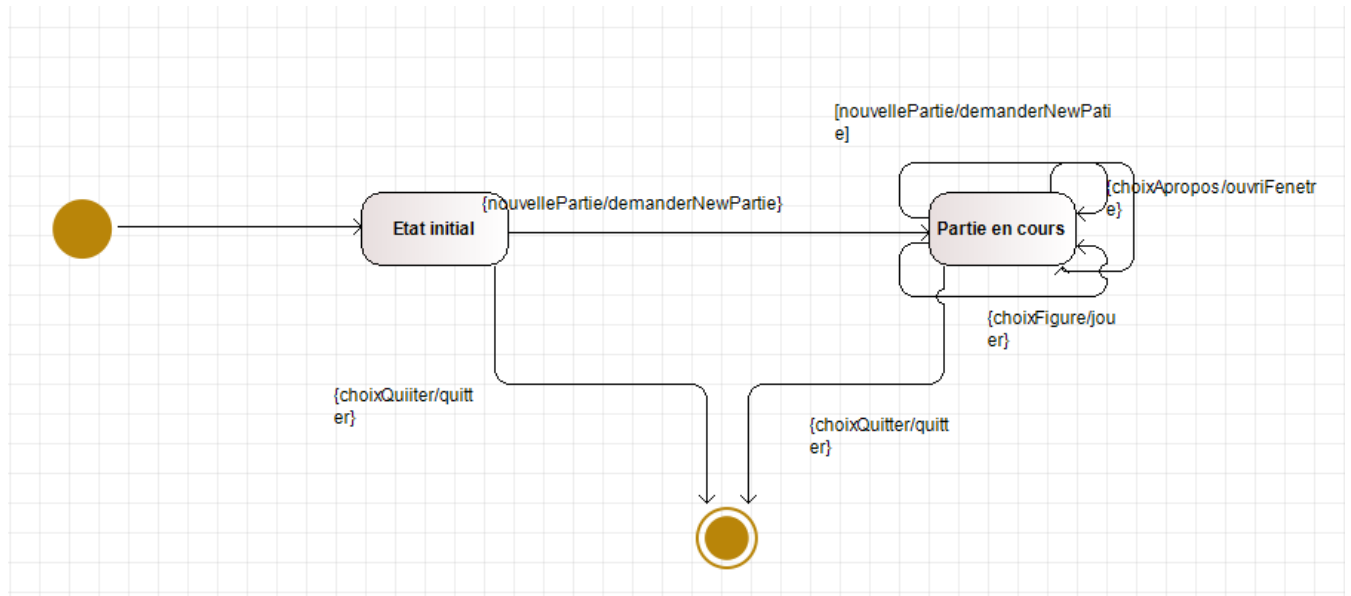


Figure 1 : Diagramme états-transitions

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

### Dictionnaire des états du jeu

<i>nomEtat</i>	<i>Signification</i>
Etat initial	L'état dans lequel l'application se trouve lors de son déclenchement et dans lequel elle se met en attente et en réponse à tout événement extérieur
Partie en cours	L'état dans lequel l'application se trouve lorsque l'événement associé au bouton Nouvelle partie est déclenché et que les boutons figures sont actifs
Etat final	L'état dans lequel l'application se trouve pendant son arrêt ( Etat simulant l'arrêt de l'application )

Tableau 11 : États du jeu

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

<i>nomEvénement</i>	<i>Signification</i>
nouvellePartie	L'événement qui permet d'établir la transition entre l'état initial et partie en cours Ou entre la partie en cours à elle-même (lorsque le joueur veut recommencer une nouvelle partie)
ChoixFigure	L'événement qui permet au joueur de jouer lorsque les boutons figures sont actifs

Tableau 12 : Événements faisant changer le jeu d'état

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

<i>ActionRealisé</i>	<i>Signification</i>
jouer	Cette action traduit le fait que lorsque le joueur clique sur Nouvelle partie, les boutons-figures sont actifs. Le nom et score du joueur sont de couleur bleue (la partie en cours)
newPartie	Cette action correspond à une affection qui traduit le fait que le joueur décide de recommencer la partie, les scores sont remis à 0
OuvrirFenetre	A la fin du tour du joueur, les nom et score du joueur machine sont de couleur bleue pour dire que c'est le tour de la machine et réciproquement pour le joueur
quitter	L'action conduisant à la sortie de l'application

Tableau 13 : Actions à réaliser lors des changements d'état

## (i) 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

Événement nt → nomEtatJeu	bNouvellePartie	bPierre	bCiseau	bPapier	bQuitter	bAPropos
	nouvellePartie	choixFigure			quitter	aPropos
Etat initial	partieEnCours / activité1	-----			Etat final / activité4	Etat initial / activité5
Partie en cours	partieEnCours / activité3	partieEnCours / activité2			Etat final / activité4	partieEnCours / activité5
Etat final	-----	-----			-----	-----

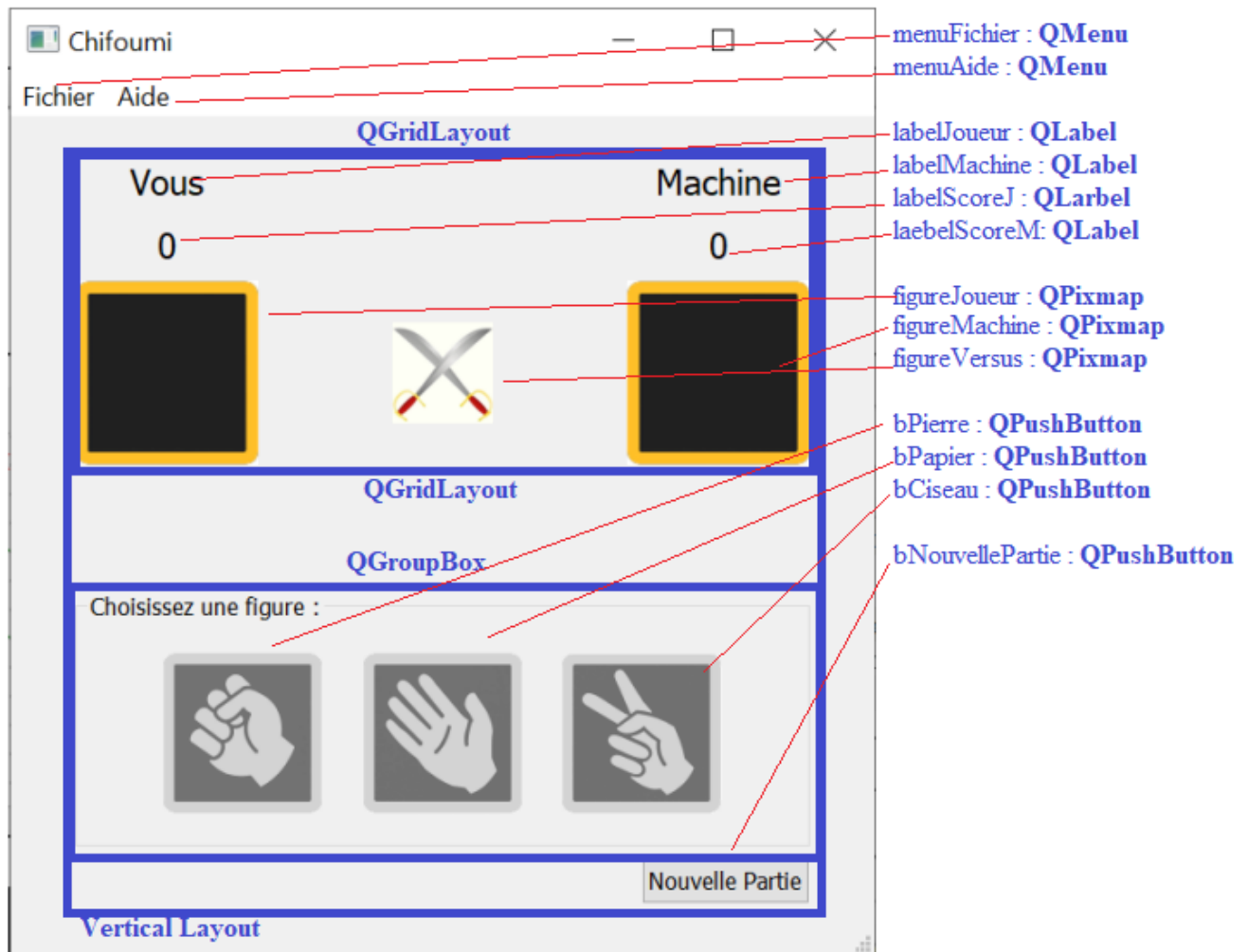
Tableau 14 : Matrice d'états-transitions du jeu chifoumi

Nom Activité	signification
Activité1	Mettre les scores à 0, mettre les figures à rien Afficher ces éléments Rendre les boutons actifs, mettre le focus sur bouton Nouvelle Partie
Activité2	Met à jour la figure du joueur On tire une figure aléatoire Met à jour la figure de la machine Déterminer le gagnant Met à jour les scores Afficher ces valeurs (scores et figures) Mettre le focus sur bouton Nouvelle Partie
Activité3	Mettre les scores à 0, mettre les figures à rien Afficher ces éléments Mettre le focus sur bouton Nouvelle Partie
Activité4	Quitter le jeu
Activité5	Ouverture d'une boîte de message(fenêtre modale ) donnant des informations sur la version de l'application, sur la date de création et les auteurs

*L'intérêt de cette vue matricielle est qu'elle permet une préparation naturelle et aisée de l'étape suivante de programmation.*

## 2. Éléments d'interface – 5 min

*Tout d'abord remarquons que dans cette version nous avons une fenêtre quasi standard puisqu'elle présente les 9 composants principaux d'une fenêtre standard en effet hormis Les barres de défilements vertical et horizontal nous avons*  
*Une poignée de redimensionnement*  
*Une barre de titre titrant le nom de l'application*  
*1 menu système composé du menu aide et du menu fichier*  
*On a le bouton de minimisation, le bouton de maximisation et de fermeture*  
*Une barre de menu / d'actions*  
*Et enfin une zone client*



### 3. Implémentation et tests – 30 min

#### 8.1 Implémentation

Liste des fichiers sources et rôle de chaque fichier

**modele.h :**

*Contenant l'ensemble des informations permettant de programmer le jeu.*

**chifoumivue.h :**

*Il assure également la gestion de l'interface graphique.*

**presentation.h :**

*Fichier servant de moteur pour le jeu.*

**main.cpp**

*Fichier permettant d'exécuter le modèle. Et d'exécuter l'application graphique*

**chifoumivue.ui**

#### Répertoire Ressources



*Implémentation du signals/slots*

```
connect(ui->bNouvellePartie, SIGNAL(clicked()), this,
        SLOT(demanderNouvellePartie()));
connect(ui->bPierre, SIGNAL(clicked()), this, SLOT(choisirPierre()));
connect(ui->bCiseau, SIGNAL(clicked()), this, SLOT(choisirCiseau()));
connect(ui->bPapier, SIGNAL(clicked()), this, SLOT(choisirPapier()));
connect(ui->action_Quitter, SIGNAL(triggered()), this, SLOT(choisirQuitter()));
connect(ui->actionA_propos_de, SIGNAL(triggered()), this, SLOT(aProposDe()));
```

**8.2 Test**

- *Le comportement fonctionnel du programme*
- *Le comportement de l'interface non lié aux aspects fonctionnels du programme*

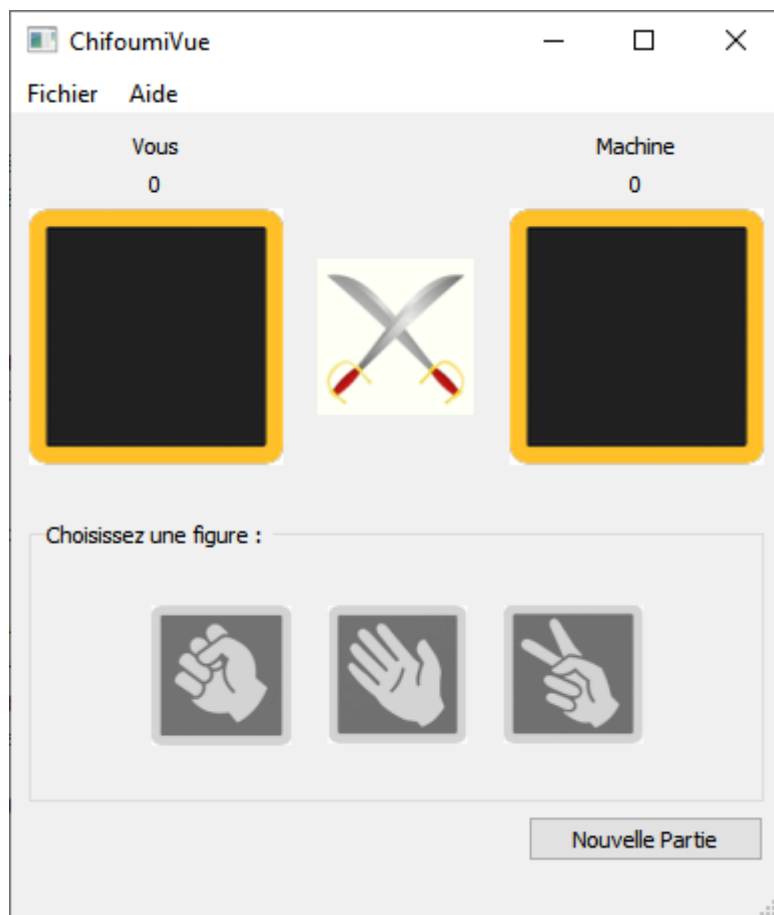
Interaction réaliser	Action attendue	Action réaliser
Etat initial	Afficher les scores à 0 Afficher les coups à rien Rendre les figures actives Focus sur le bouton nouvelle Partie	Voir la capture Figure 2
Partie en cours : Choix de figure	Afficher le coup du joueur Choisir un coup pour le système et l'afficher Déterminer le gagnant, mettre à jour les scores et les afficher Focus sur le bouton nouvellePartie	Voir la capture Figure 3
Partie en cours : Click sur nouvelle partie	Initialiser les scores à 0 Initialiser les coups à rien Focus sur le bouton nouvellePartie	Voir la capture Figure 2

Comportement fonctionnel du programme lors de la réalisation des tests

--Etat des éléments graphiques lorsque le jeu d'état du programme est dans l'état « Etat initial »

*Explication de l'inactivité des boutons avant le déclenchement du slot nouvelle partie*

-----



*Figure 1 :  
Explication de l'activité des boutons après le déclenchement du slot nouvelle partie*



Figure 2 :

--Etat des éléments graphiques lorsque le jeu d'état du programme est dans l'état « Partie en Cours »

Cas d'exemple : Lorsque le joueur décide de jouer feuille (déclenchement du slot choixfeuille())

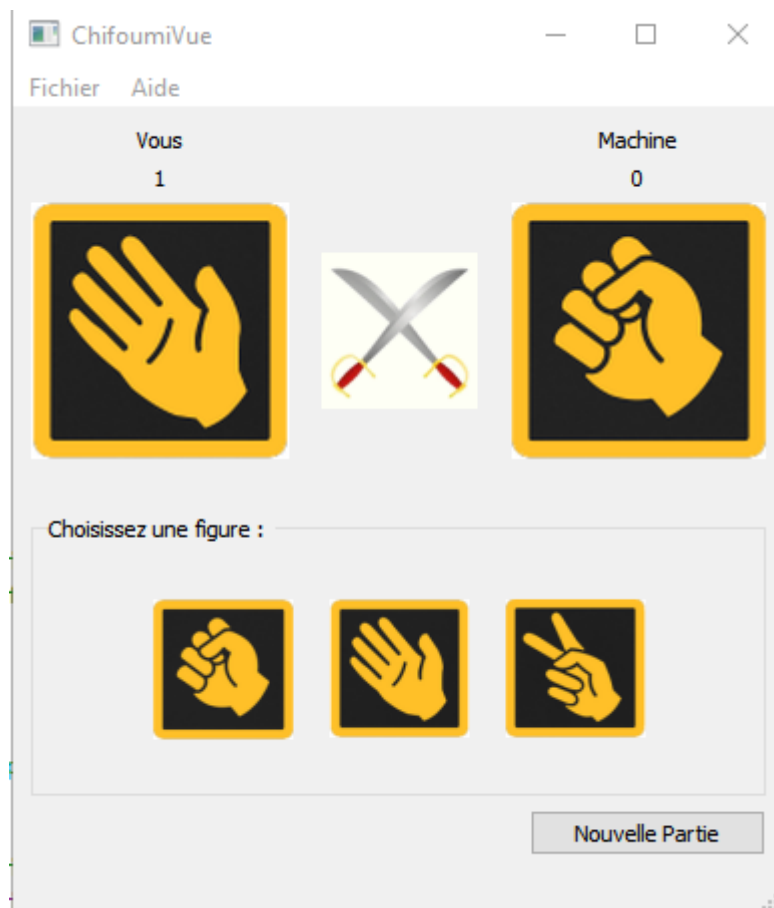


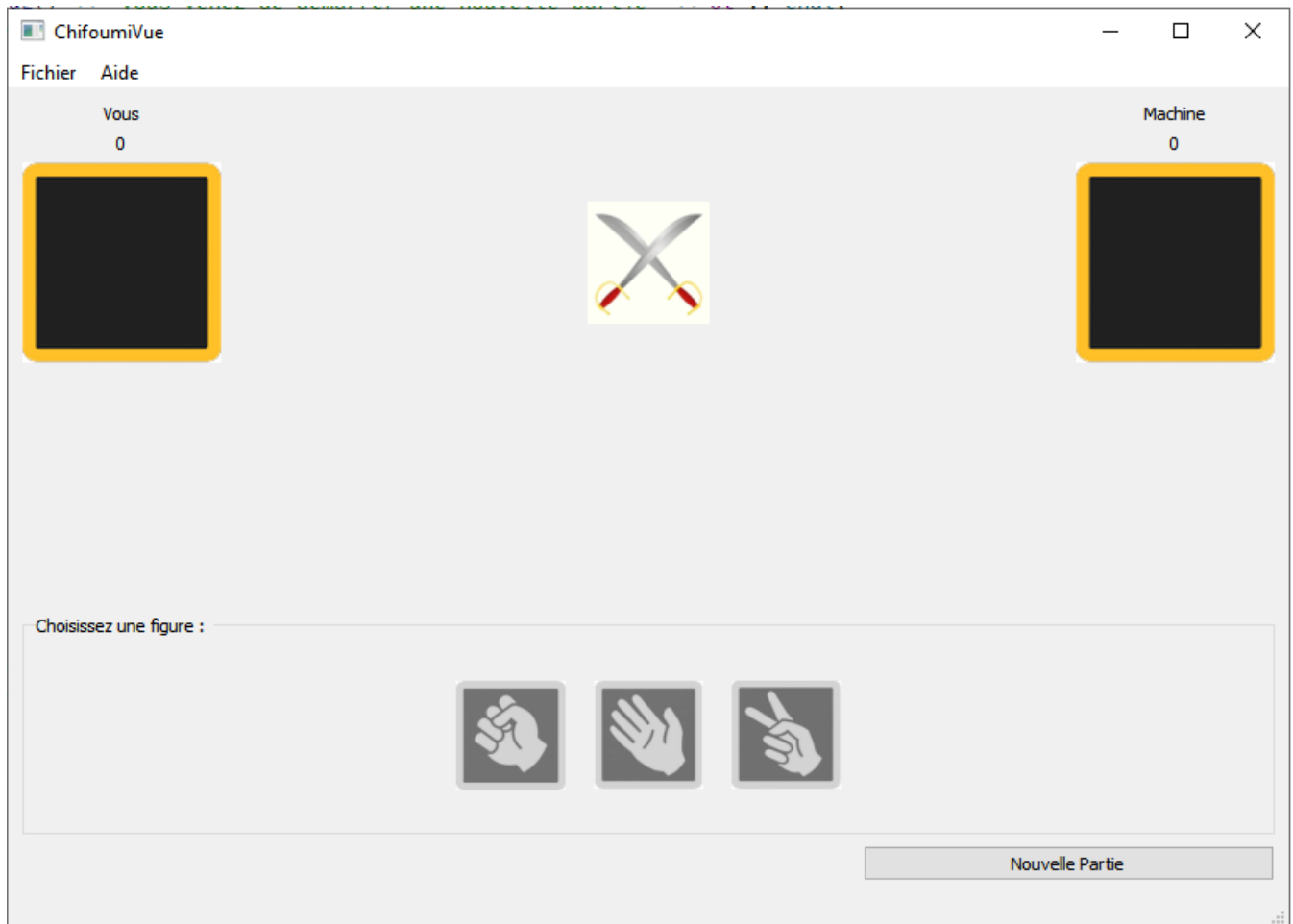
Figure 3 :

Comportement de l'interface (non lié aux aspects fonctionnels du programme) lors de la réalisation des tests

--Etat des éléments graphiques avant le redimensionnement

(Voir Figure 1 Pour observer l'Interface dans l'état initial de l'application)

--Etat des éléments graphiques lors du redimensionnement  
(voir figure 4)



*Figure 4 :*  
*Citer et Expliquer* les règles ergonomiques qui ont motivé ce choix.

--Etat des éléments graphiques avant et lors du redimensionnement  
(voir figure 5)

Nous remarquons qu'après l'ouverture du dialogue, l'utilisateur n'est plus autorisé à interagir en parallèle avec la première fenêtre en d'autre terme le dialogue est obligatoire mais est déplaçable par l'utilisateur pour laisser l'utilisateur voire la tâche amont (voir figure 6)

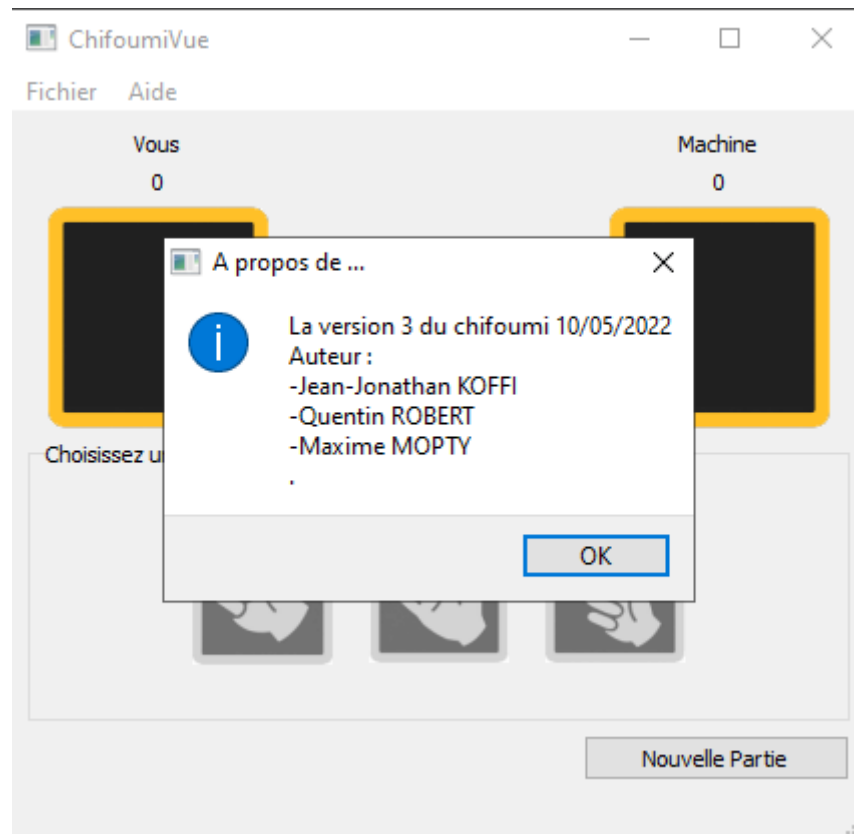


Figure 5 :

## 4 Bilan : - 10 min

*Bilan intermédiaire à compléter après chaque version : ce que vous avez appris, ce qui a été difficile, le TEMPS passé (On mettra le temps à coter des titres pour spécifié directement) par chaque membre de l'équipe sur la version (analyseConception et code)*

*Bilan final à compléter à la fin de la saé : ce que vous avez aimé / pas aimé, ce que vous auriez pu faire mieux (avec le recul), ce qui pourrait être amélioré dans la saé, ainsi que l*

## Version v4

### 1. Classe Chifoumi : Diagramme états-transitions – 30 min

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

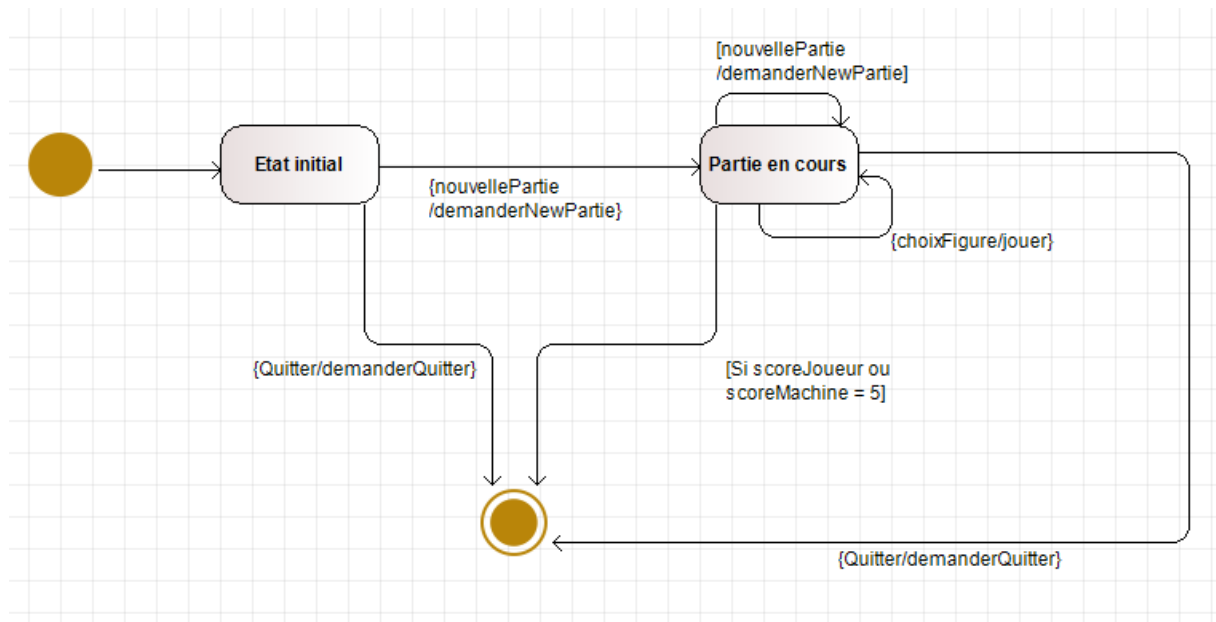


Figure 1 : Diagramme états-transitions

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

### Dictionnaire des états du jeu

<i>nomEtat</i>	<i>Signification</i>
Etat initial	L'état dans lequel l'application se trouve lors de son déclenchement et dans lequel elle se met en attente et en réponse à tout événement extérieur
Partie en cours	L'état dans lequel l'application se trouve lorsque l'événement associé au bouton Nouvelle partie est déclenché et que les boutons figures sont actifs
Etat final	L'état dans lequel l'application se trouve pendant son arrêt

Tableau 15 : États du jeu

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

<i>nomEvénement</i>	<i>Signification</i>
CliquerSurNouvellePartie partie	L'événement qui permet d'établir la transition entre l'état initial et partie en cours Ou entre la partie en cours à elle-même (lorsque le joueur veut recommencer une nouvelle partie)
ChoisirFigure	L'événement qui permet au joueur de jouer lorsque les boutons figures sont actifs

Tableau 16 : Événements faisant changer le jeu d'état

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

<i>ActionRealisé</i>	<i>Signification</i>
jouer	Cette action traduit le fait que lorsque le joueur clique sur Nouvelle partie, les boutons-figures sont actifs. Le nom et score du joueur sont de couleur bleue (la partie en cours)
newPartie	Cette action correspond à une affection qui traduit le fait que le joueur décide de recommencer la partie, les scores sont remis à 0
OuvrirFenetre	A la fin du tour du joueur, les nom et score du joueur machine sont de couleur bleue pour dire que c'est le tour de la machine et réciproquement pour le joueur
quitter	L'action conduisant à la sortie de l'application

Tableau 17 : Actions à réaliser lors des changements d'état

## (l) 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



Événement → nomEtatJeu	bNouvellePartie	bPierre	bCiseau	bPapier	bQuitter
	nouvellePartie	choixFigure			-----
Etat initial	partieEnCours / activité1	-----			Etat final / activité4
Partie en cours	partieEnCours / activité3	partieEnCours / activité2			Etat final / activité4
Etat final	-----	-----			-----

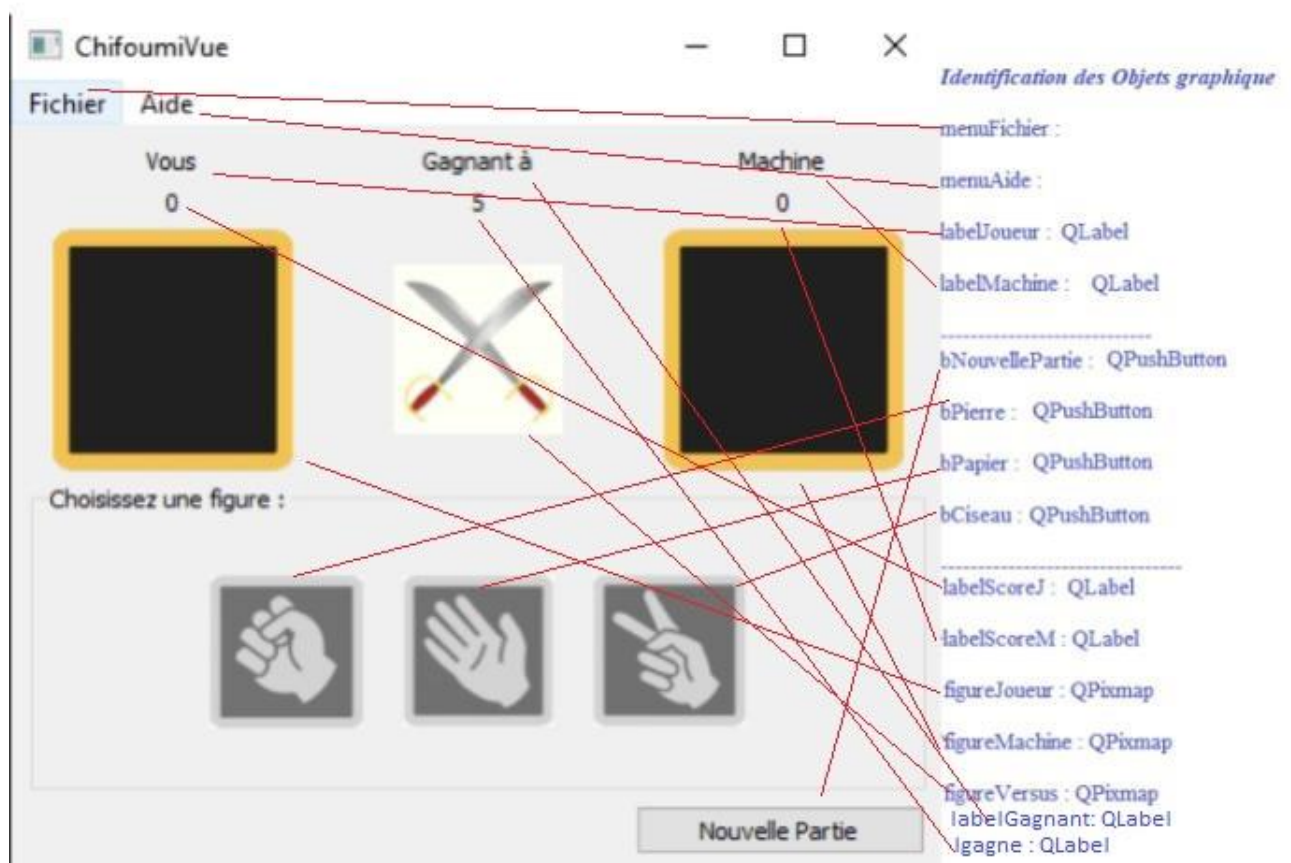
Tableau 18 : Matrice d'états-transitions du jeu chifoumi

Nom Activité	signification
Activité1	Mettre les scores à 0, mettre les figures à rien Afficher ces éléments Rendre les boutons actifs, mettre le focus sur bouton Nouvelle Partie
Activité2	Met à jour la figure du joueur On tire une figure aléatoire Met à jour la figure de la machine Déterminer le gagnant Met à jour les scores Afficher ces valeurs (scores et figures) Mettre le focus sur bouton Nouvelle Partie
Activité3	Mettre les scores à 0, mettre les figures à rien Afficher ces éléments Mettre le focus sur bouton Nouvelle Partie
Activité4	Quitter le jeu

*L'intérêt de cette vue matricielle est qu'elle permet une préparation naturelle et aisée de l'étape suivante de programmation.*

## 2. Éléments d'interface – 10 min

*A faire ici : description sommaire des éléments de l'interface, par exemple, avec une copie d'écran sur laquelle sont nommés les variables/objets graphiques et où les layouts sont positionnés et nommés.*



### 3. Implémentation et tests – 1h 30

#### 8.1 Implémentation

*A faire :*

*Lister les fichiers impliqués dans cette version (répertoire, nom de fichier, rôle de chaque fichier)*

Liste des fichiers sources et rôle de chaque fichier

**modele.h :**

*Contenant l'ensemble des informations permettant de programmer le jeu.*

**chifoumivue.h :**

*Il assure également la gestion de l'interface graphique.*

**presentation.h :**

*Fichier servant de moteur pour le jeu.*

**main.cpp**

*Fichier permettant d'exécuter le modèle. Et d'exécuter l'application graphique*

**chifoumivue.ui**

**Répertoire Ressources**

*Dossier regroupant l'ensemble des images servant à compléter l'interface graphique*

*Commenter brièvement les choix importants d'implémentation réalisés, comme par exemple, les signals/slots*

*Explication de l'implémentation des signals/slot Tableau (*

*Signal : click*

*Slot : DemanderNewPartie*

*Signal : click*

*Slot : Choix figure (Changement de l'image figureJoueur en bouton selon le choix du joueur)*

## 8.2 Test

*A faire :*

*Décrire les tests prévus / réalisés pour montrer :*

*- Le comportement fonctionnel du programme*

*- Le comportement de l'interface non lié aux aspects fonctionnels du programme*

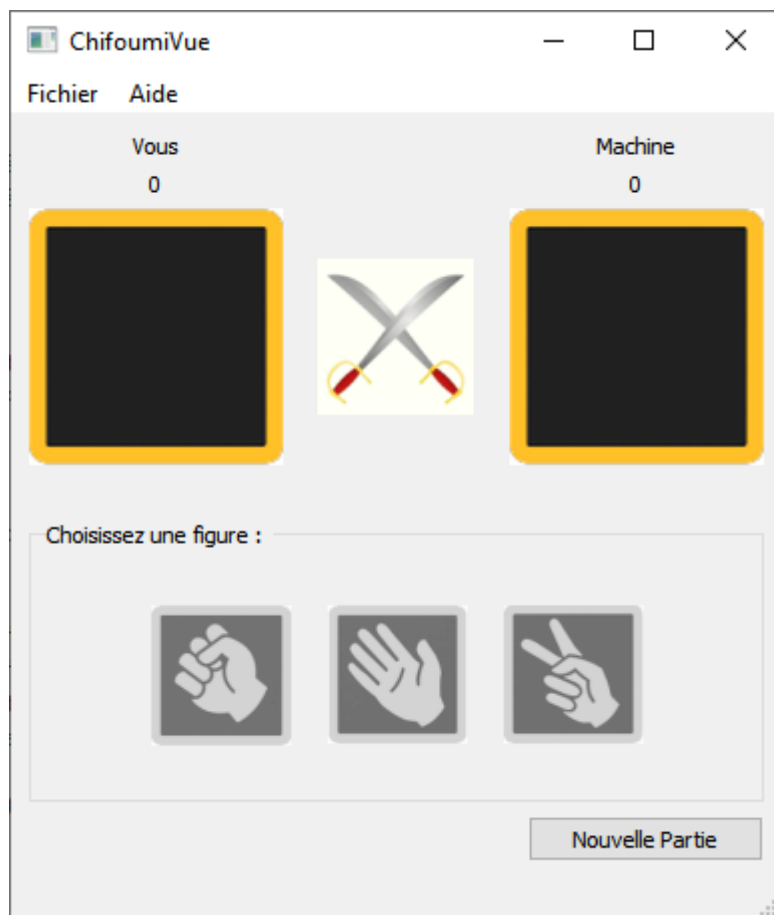
Interaction réaliser	Action attendue	Action réaliser
Etat initial	Afficher les scores à 0 Afficher les coups à rien Rendre les figures actives Focus sur le bouton nouvellePartie	Voir la capture Figure 2
Partie en cours : Choix de figure	Afficher le coup du joueur Choisir un coup pour le système et l'afficher Déterminer le gagnant, mettre à jour les scores et les afficher Focus sur le bouton nouvellePartie	Voir la capture Figure 3
Partie en cours : Click sur nouvelle partie	Afficher les scores à 0 Afficher les coups à rien Rendre les figures actives Focus sur le bouton nouvellePartie	Voir la capture Figure 2

Comportement fonctionnel du programme lors de la réalisation des tests

*--Etat des éléments graphiques lorsque le jeu d'état du programme est dans l'état « Etat initial »*

*Explication de l'inactivité des boutons avant le déclenchement du slot nouvelle partie*

-----



*Figure 1 :  
Explication de l'activité des boutons après le déclenchement du slot nouvelle partie*



Figure 2 :

--Etat des éléments graphiques lorsque le jeu d'état du programme est dans l'état « Partie en Cours »

Cas d'exemple : Lorsque le joueur décide de jouer feuille (déclenchement du slot choixfeuille())

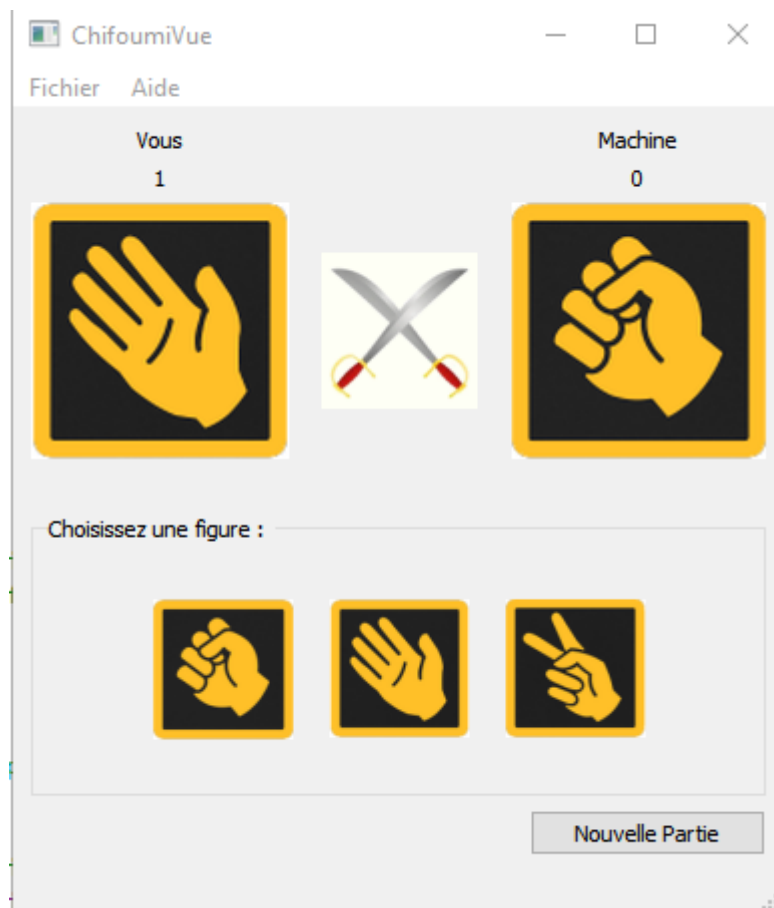


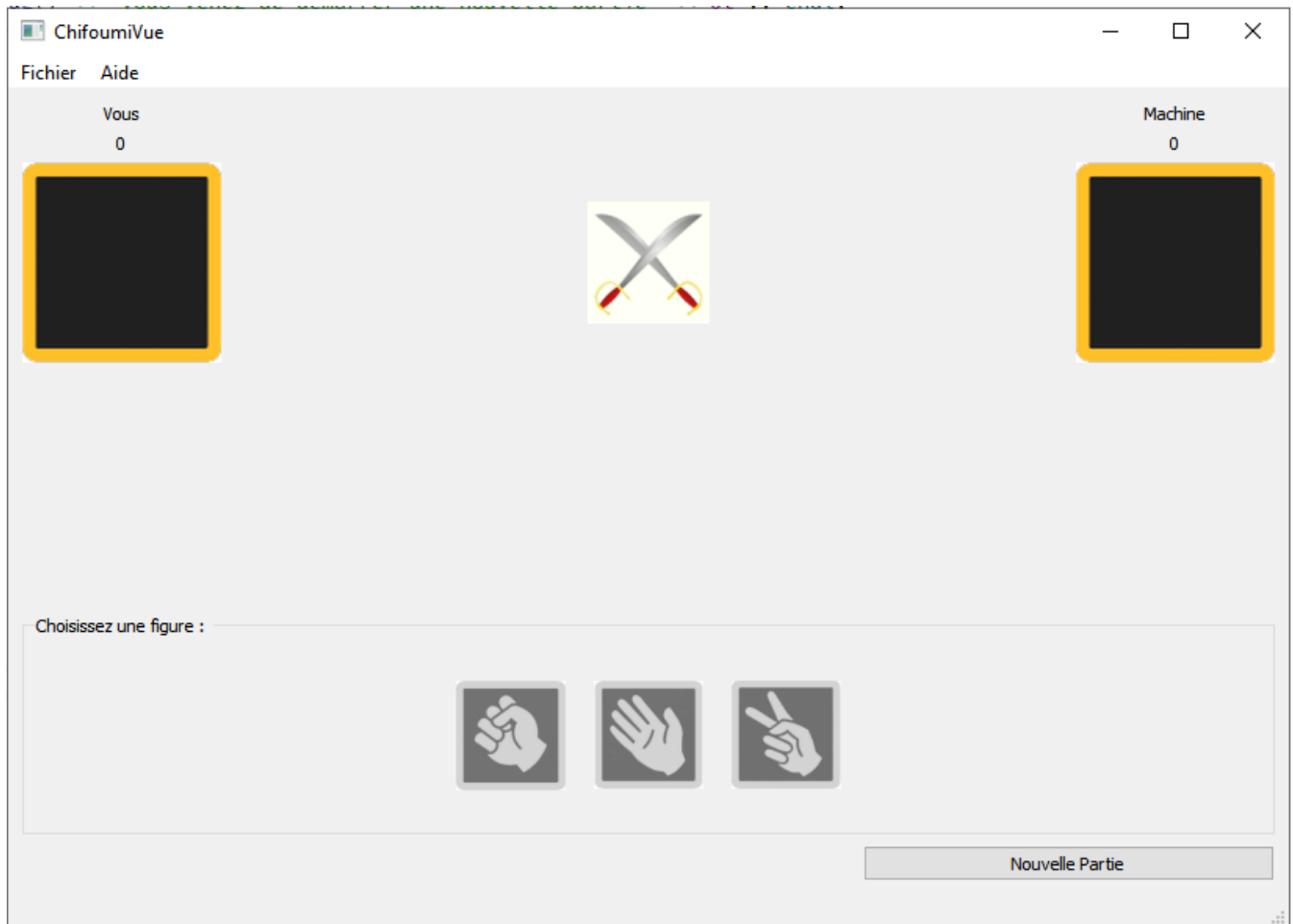
Figure 3 :

Comportement de l'interface (non lié aux aspects fonctionnels du programme) lors de la réalisation des tests

--Etat des éléments graphiques avant le redimensionnement

(Voir Figure 1 Pour observer l'Interface dans l'état initial de l'application)

--Etat des éléments graphiques lors du redimensionnement



*Figure 4 :*  
*Citer et Expliquer* les règles ergonomiques qui ont motivé ce choix.

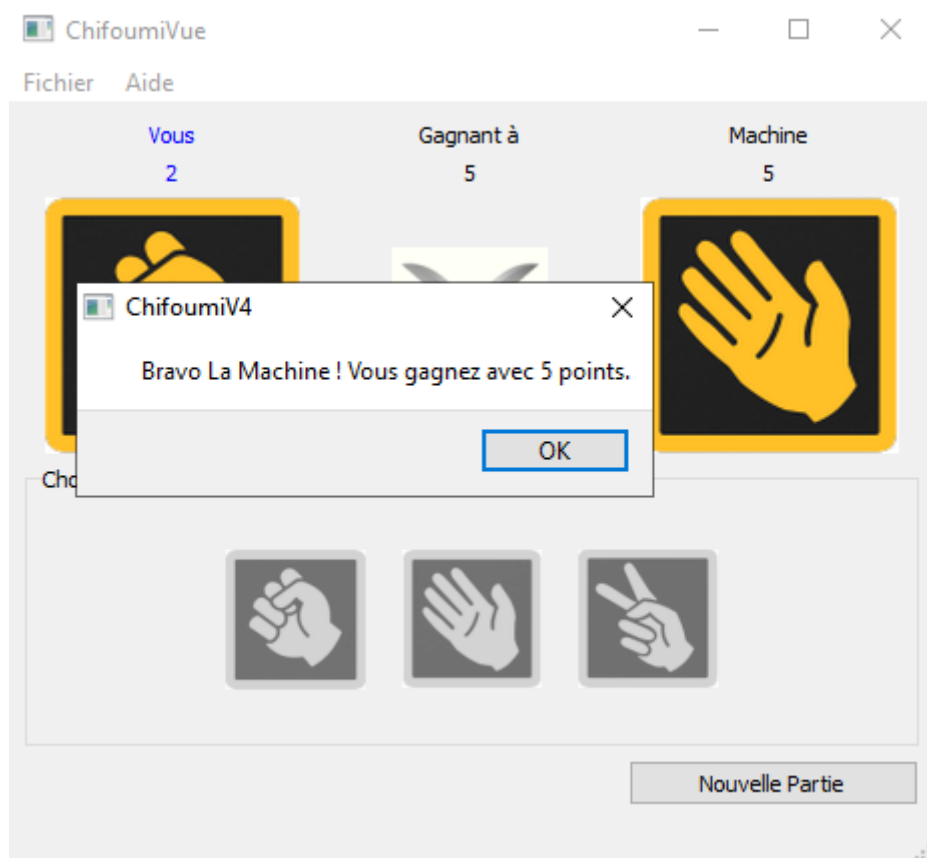


Figure 5 :



## 5 Bilan Final :

Ce que nous retenons de ce projet est l'utilisation du model MVP qui a rendu plus facile l'implémentation du jeu à partir de la V2, la version matricielle du diagramme état-transition à contribuer largement dans le développement des versions et aussi la documentation QT nous a aidés à manipuler les méthodes et fonctions. L'intervention du générateur de coups a allégé les interactions entre le joueur et la machine. Les différents temps passé sur chaque version sont renseignés dans le titre de chaque partie.