

2021 - 2022

Version 1

S2.01 - Développement d'une application

Nicolas DARGAZANLI

Alexandre MAURICE

Guillaume TRITSCH

Table des matières

[Présentation de la version 3](#_Toc104116423)

[Analyse 3](#_Toc104116424)

[Diagramme états-transitions du jeu 3](#_Toc104116425)

[Dictionnaires des états, événements et actions 3](#_Toc104116426)

[Éléments d’interface 5](#_Toc104116427)

[Hiérarchie des layouts et frames 6](#_Toc104116428)

[Les principaux layouts et frames 7](#_Toc104116429)

[Implémentation et tests 8](#_Toc104116430)

[Implémentation 8](#_Toc104116431)

[Détails des fichiers 8](#_Toc104116432)

[main.cpp 8](#_Toc104116433)

[chifoumi.h 8](#_Toc104116434)

[chifoumi.cpp 9](#_Toc104116435)

[chifoumivue.h 9](#_Toc104116436)

[chifoumivue.cpp 9](#_Toc104116437)

[chifoumivue.ui 9](#_Toc104116438)

[/Image 9](#_Toc104116439)

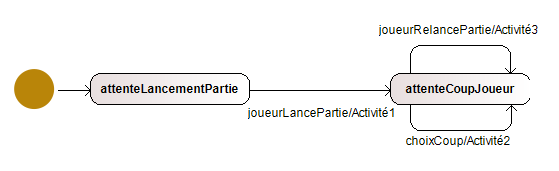
[Test de la version 9](#_Toc104116440)

# Présentation de la version

Cette version consiste à réaliser une analyse du programme, et la création d’une première version fonctionnelle du programme avec une fenêtre graphique créée sous Qt. Cette version réutilise les fichiers de la version 0.

# Analyse

## Diagramme états-transitions du jeu



## Dictionnaires des états, événements et actions

**Dictionnaire** **des états du jeu**

|  |  |
| --- | --- |
| *nomEtat* | *Signification* |
| attenteLancementPartie | L’application attend que le joueur lance la partie. |
| attenteCoupJoueur | Le programme attend que le joueur choisisse une figure. |

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

|  |  |
| --- | --- |
| *nomEvénement* | *Signification* |
| choixCoup | Clique sur un des boutons de figure (pierre, papier, ciseaux) |
| joueurRelancePartie | Clique sur le bouton boutonRejouer |
| joueurLancePartie | Clique sur le bouton boutonRejouer |

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

|  |  |
| --- | --- |
| Activité1 | Le joueur choisit un coup à jouer parmi « Pierre », « Feuille » et « Ciseau ». |
| Activité2 | Réinitialise le score et le coup du joueur et de la machine. |
| Activité3 | Initialise le score et le coup du joueur et de la machine. |

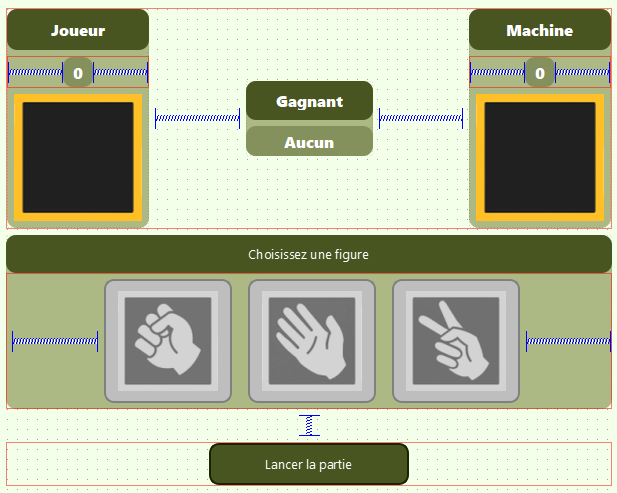
**Préparation au codage :**

**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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | boutonRejouer | boutonPierre | boutonPapier | boutonCiseaux |
| *Événement 🡪*  *nomEtatJeu* | joueurLancePartie | choixCoup | | |
| attenteLancementPartie | attenteCoupJoueur/Activité1 | // | | |
| attenteCoupJoueur | attenteCoupJoueur/Activité3 | attenteCoupJoueur/Activité2 | | |

## Éléments d’interface



La fenêtre est divisée en 4 zone distincte, chaque élément de ces zones sont contenue dans des frames. Pour ses 4 zones nous avons priorisé l’utilisation de frame car ils sont plus personnalisables que de simple layout.

Ces 4 zones sont frameSelect, frameJGauche, frameJDroit, frameJGagnant.

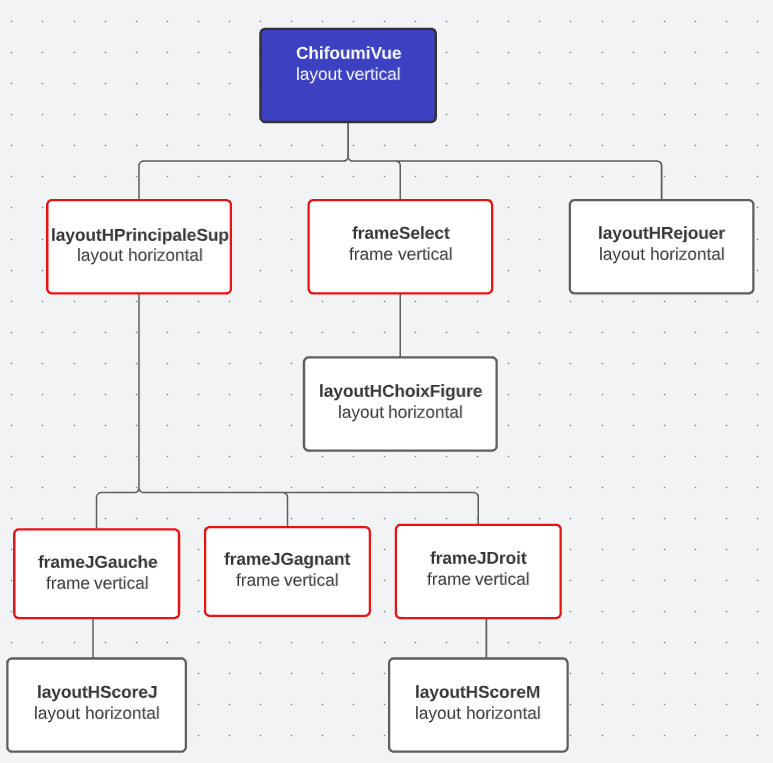
Chaque zone est détaillée plus bas.

Le bouton « lancer la partie » est en dehors de toutes les zones car il n’est pas pertinent de le regrouper avec d’autre widget.

### Hiérarchie des layouts et frames

Pour plus de compréhension voici la hiérarchie des layouts et des frames utilisés :

Les éléments entourés en rouge sont les principaux éléments, ils sont détaillés dans la section *Les principaux layouts et frames*, ce sont ceux qui regroupe le plus d’élément. Les autres layout sont moins primordiaux, et sont nécessaire afin de centrer le ou les éléments qu’ils contiennent.



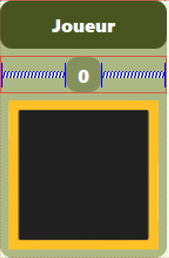
### Les principaux layouts et frames

**layoutHPrincipaleSup**

C’est un layout qui regroupe 3 zones distinctes, frameJGauche, frameJDroit, frameJDroit.

**frameJGauche**

frameJGauche est une zone qui regroupe les widgets donnant des informations sur le joueur humain, son nombre de point, la dernière figure sélectionné. Pour les versions futures, cette zone pourra accueillir plus d’information sur le joueur.

****

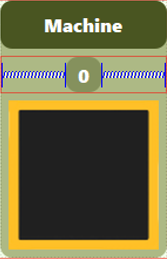
**frameJGagnant**

frameJGagnant est une zone qui regroupe les widgets donnant des informations relatives à la partie en cours, par exemple le dernier gagnant. Pour les versions futures, cette zone devrait accueillir les informations sur le timer.

****

**frameJDroit**

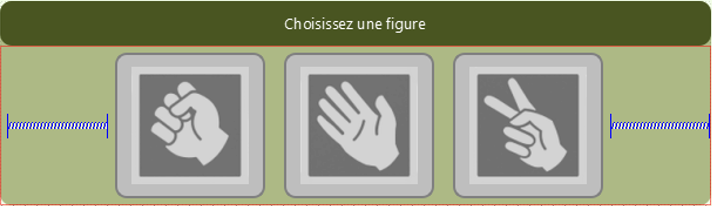
frameJDroit est une zone qui regroupe les widgets donnant des informations sur la machine, son nombre de point, la dernière figure sélectionné. Pour les versions futures, cette zone pourra accueillir plus d’information sur la machine.

****

**frameSelect**

frameSelect est la zone qui regroupe les interactions du joueur avec le programme durant une partie. Elle contient donc les boutons des différentes figures.

Nous avons préféré utiliser un frame avec un label plutôt qu’un group box pour une raison purement esthétique, le group box permet moins de personnalisation au niveau la feuille de style, et lors du redimensionnement de la fenêtre les éléments avait tendance à se chevaucher. Un frame et un label garantisse que chaque élément à un espace à lui, tout en nous permettant de garder le même style que sur le reste de la fenêtre.



## Implémentation et tests

### Implémentation

A faire :

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

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

# Détails des fichiers

## main.cpp

Programme permettant de vérifier le bon fonctionnement de l’interface et le corps des fichier intitulé chifoumi. Ce fichier est un fichier de test fournis lors de la présentation de la SAE.

## chifoumi.h

Ce fichier est l’interface de la classe chifoumi qui met à disposition la gestion d’une partie de chifoumi entre un joueur et l’ordinateur.

Cela comprend donc des attributs de classe gérant le score et le coup de chaque joueur, et des méthodes pour générer un coup, les comparer, gérer les scores, …

## chifoumi.cpp

Ce fichier est le corps de la classe chifoumi. Elle contient le corps de chaque méthode de la classe.

## chifoumivue.h

Ce fichier contient l’interface de la classe ChifoumiVue qui gère la fenêtre graphique et les interactions avec. Pour fonctionner la classe utilise la classe Chifoumi et le fichier chifoumivue.ui.

## chifoumivue.cpp

Ce fichier regroupe les définitions des sous programmes et slots déclaré dans l’interface.

## chifoumivue.ui

Ce fichier contient toutes les informations de l’interface utilisateur.

## /Image

Le dossier image contient des images utilisables par le programme, toutes ne sont pas utilisé.

# Test de la version

|  |  |  |  |
| --- | --- | --- | --- |
| Objet du test | Oracle | Résultat | Conclusion |
| Test du bouton rejouer | Actualisation des couleurs et du bouton\*. Réalisation des scores et des coups. | Actualisation des couleurs et du bouton. Réalisation des scores et des coups. | Test passé |
| Test du bouton pierre | Attribution du coup pierre au joueur et déroulement du tour. | Attribution du coup pierre au joueur et déroulement du tour. | Test passé |
| Test du bouton ciseau | Attribution du coup ciseau au joueur et déroulement du tour. | Attribution du coup ciseau au joueur et déroulement du tour. | Test passé |
| Test du bouton feuille | Attribution du coup feuille au joueur et déroulement du tour. | Attribution du coup feuille au joueur et déroulement du tour. | Test passé |

\*Le bouton doit passer du texte « Lancer la partie » à « Relancer la partie » au premier clique