HENTRICS LOISTINE Samuel TD2 TP3
ETCHEPARE Cédric TD2 TP3
FAKHFAKH Ahmed TD2 TP3

S2.01 – Développement d'une application

Chifoumi – Dossier d'Analyse et conception - Version 1

BUT Informatique Semestre 2 2021-2022
IUT de Bayonne et du Pays Basque

01/05/2022

Sommaire :

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

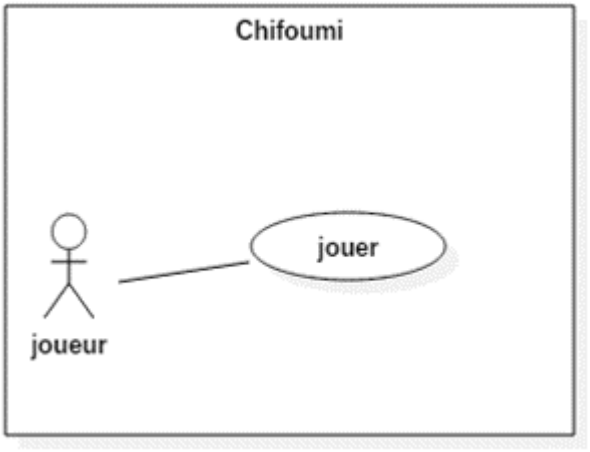


Figure 1 : Diagramme des Cas d'Utilisation du jeu Chifoumi

3. Scénarios

(a) Exemple Scénario

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.

(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.

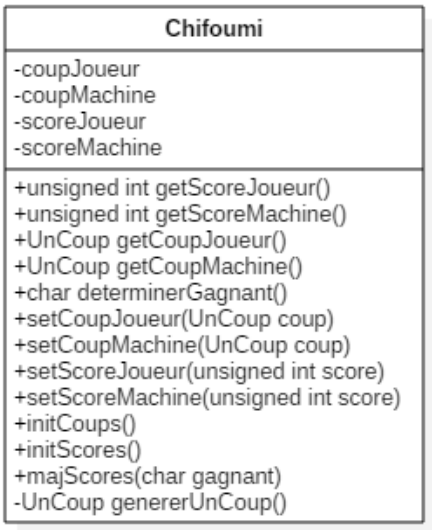


Figure 2 : Diagramme de Classes UML du jeu Chifoumi

(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	Mémorise la dernière figure choisie par le joueur. Type énuméré enum unCoup {pierre, ciseau, papier, rien};	UnCoup	papier
coupMachine	Mémorise 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 4

```
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 4 : 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.

Version v0

5. Implémentation et tests

5.1 Implémentation

Liste des fichiers de cette version :
- chifoumi.h : Déclaration de la classe Chifoumi
- chifoumi.cpp : Méthodes de la classe Chifoumi

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

5.2 Test

Test avec le programme fournit main.cpp

Testeur(s): Samuel HENTRICS LOISTINE Date: 14/04/2022
Ahmed FAKHFAKH
Cédric ETCHEPARE

Élément testé : main.cpp Version: 0

Classe	Description	Valeurs en entrée	Résultat(s) attendu(s)
Valide n°1	Le joueur joue pierre. La machine joue pierre.	coupJoueur=pierre coupMachine=pierre	Aucun gagnant
Valide n°2	Le joueur joue pierre. La machine joue feuille.	coupJoueur=pierre coupMachine=feuille	Machine
Valide n°3	Le joueur joue pierre. La machine joue ciseau.	coupJoueur=pierre coupMachine=ciseau	Joueur
Valide n°4	Le joueur joue feuille. La machine joue pierre.	coupJoueur=feuille coupMachine=pierre	Joueur
Valide n°5	Le joueur joue feuille. La machine joue feuille.	coupJoueur=feuille coupMachine=feuille	Aucun gagnant
Valide n°6	Le joueur joue feuille. La machine joue ciseau.	coupJoueur=feuille coupMachine=ciseau	Machine
Valide n°7	Le joueur joue ciseau. La machine joue pierre.	coupJoueur=ciseau coupMachine=pierre	Machine
Valide n°8	Le joueur joue ciseau. La machine joue feuille.	coupJoueur=ciseau coupMachine=feuille	Joueur
Valide n°9	Le joueur joue ciseau. La machine joue ciseau.	coupJoueur=ciseau coupMachine=ciseau	Aucun gagnant

Version v1

6. Classe Chifoumi : Diagramme états-transitions

(a) Diagramme états-transitions -actions du jeu

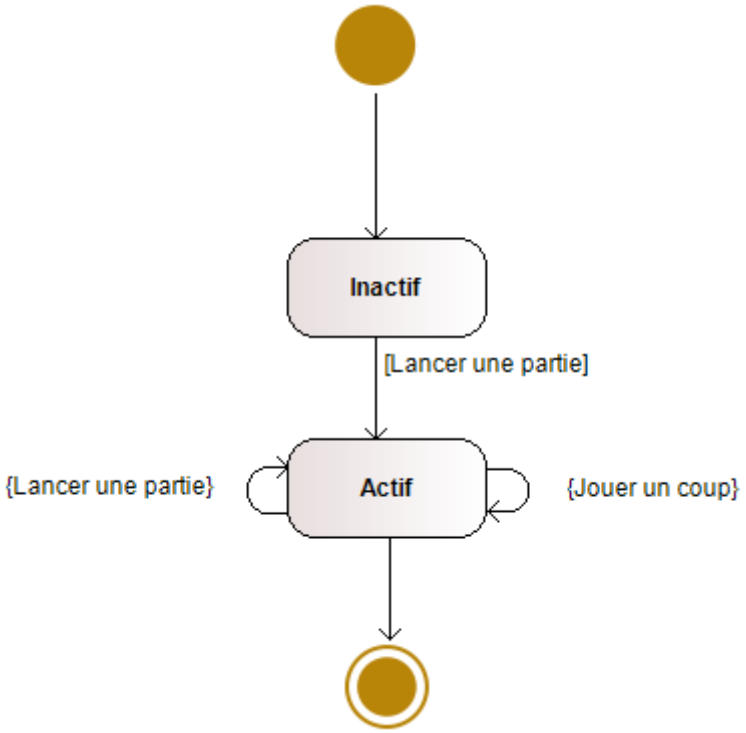


Figure 9 : Diagramme états-transitions

(b) Dictionnaires des états, événements et Actions

Dictionnaire des états du jeu

nomEtat	*Signification*
Inactif	Le jeu avant qu'on lance une partie
Actif	Le jeu pendant que la partie est en cours

Tableau 2 : États du jeu

Dictionnaire des événements faisant changer le jeu d'état

nomEvénement	*Signification*
Lancer une partie	Met le jeu en état actif avec les scores des deux joueurs à zéro et leurs coups à "rien".
Jouer un coup	Fait tourner le jeu en déterminant à chaque partie qui est le gagnant en augmentant les scores.

Tableau 3 : Evénements faisant changer le jeu d'état

Description des actions réalisées lors de la traversée des transitions

Inactif -> Actif	Le joueur vient d'arriver sur l'application et souhaite lancer une partie
Actif (Jouer une partie)	Le joueur choisit un coup et joue contre la machine.
Actif (Lancer une partie)	Permet de mettre le jeu à zero et de recommencer une partie

Tableau 4 : 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

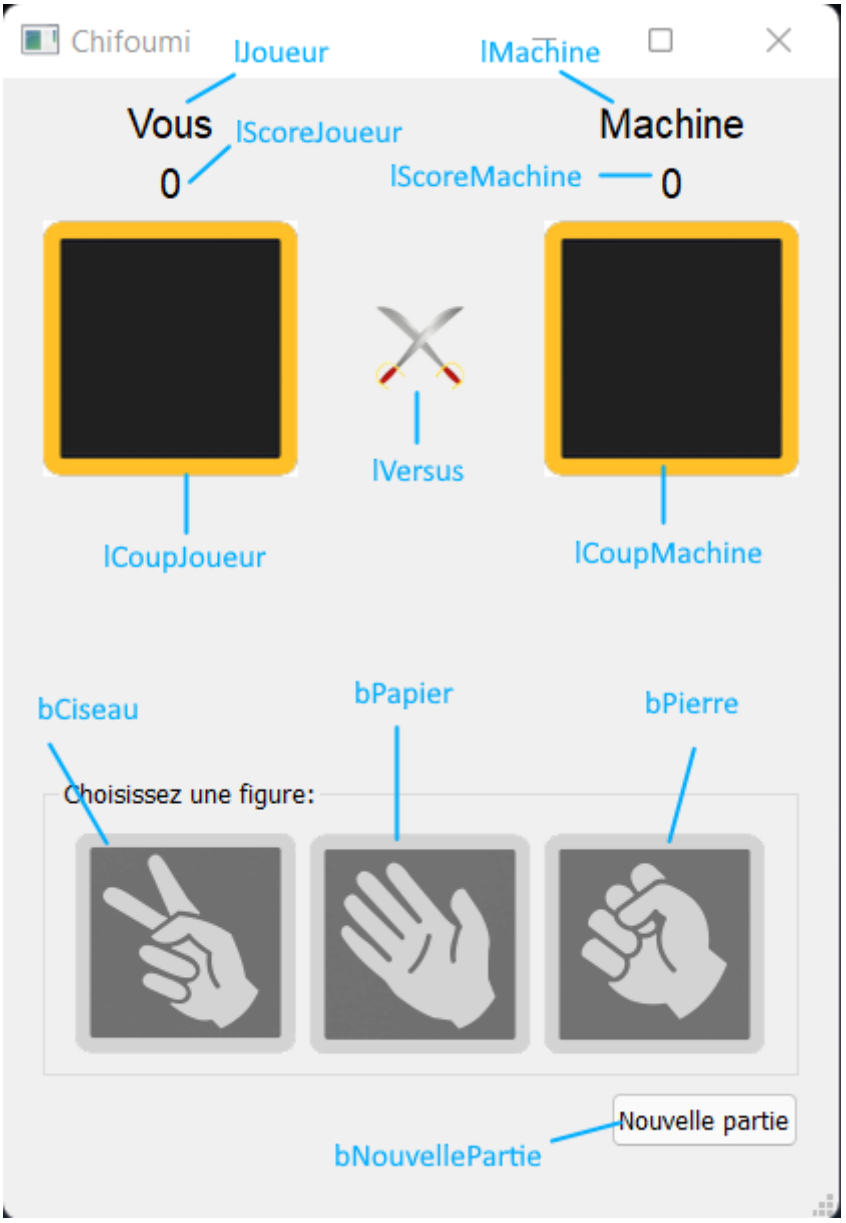
Événement / *nomEtatJeu*	Jouer un coup	Lancer une partie
Inactif		Actif
Actif	Actif	Actif

Tableau 5 : Matrice d'états-transitions du jeu chifoumi

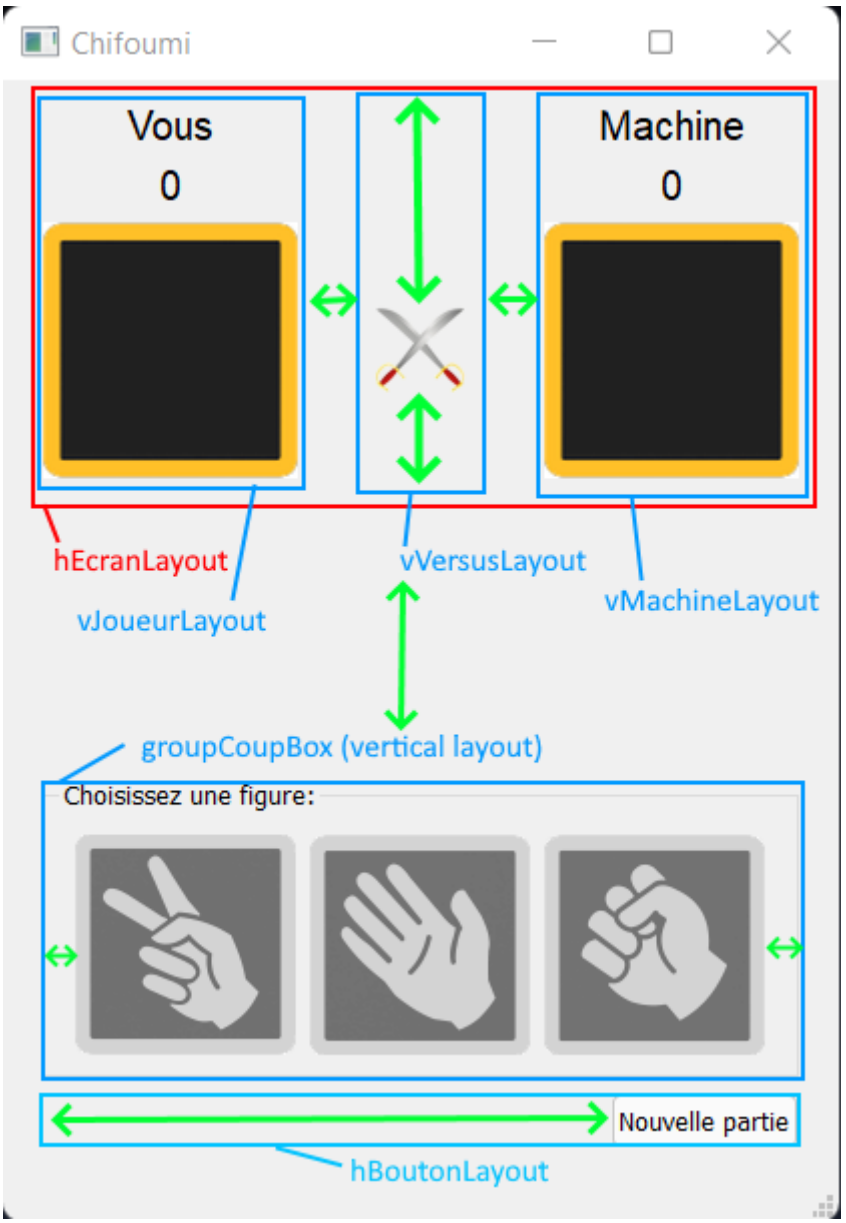
L'intérêt de cette vue matricielle est qu'elle permet une préparation naturelle et aisée de l'étape suivante de programmation.

7. Éléments d'interface

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.



Objets graphiques



Les layouts (les “stretch” sont représentés en vert)

8. Implémentation et tests

8.1 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

chifoumi.pro : Fichier du projet Qt
main.cpp : Fichier permettant l'exécution du programme
chifoumi.cpp : Fichier contenant la définition des fonctions / procédures (disponibles dans le h) nécessaire à la bonne exécution du programme
chifoumi.h : Fichier contenant la déclaration des fonctions / procédures nécessaire à la bonne exécution du programme

chifoumi.ui : Fichier permettant la réalisation et le placement des éléments graphiques du jeu.
chifoumires.qrc : Fichier de ressources Qt
images/ciseau.gif : Permet la sélection du coup “Ciseau” pour que le joueur joue.
images/ciseau_115.png : Permet l'affichage du coup “Ciseau” pour le coup joué de la machine ou du joueur.
images/papier.gif : Permet la sélection du coup “Papier” pour que le joueur joue.
images/papier_115.png : Permet l'affichage du coup “Papier” pour le coup joué de la machine ou du joueur.
images/pierre.gif : Permet la sélection du coup “Pierre” pour que le joueur joue.
images/pierre_115.png : Permet l'affichage du coup “Pierre” pour le coup joué de la machine ou du joueur.
images/rien_115.png : Permet l'affichage d’aucun coup à chaque lancement d’une partie pour la machine ou le joueur.
images/versus.gif : Permet l’affichage entre les deux coups.

8.2 Test

A faire :

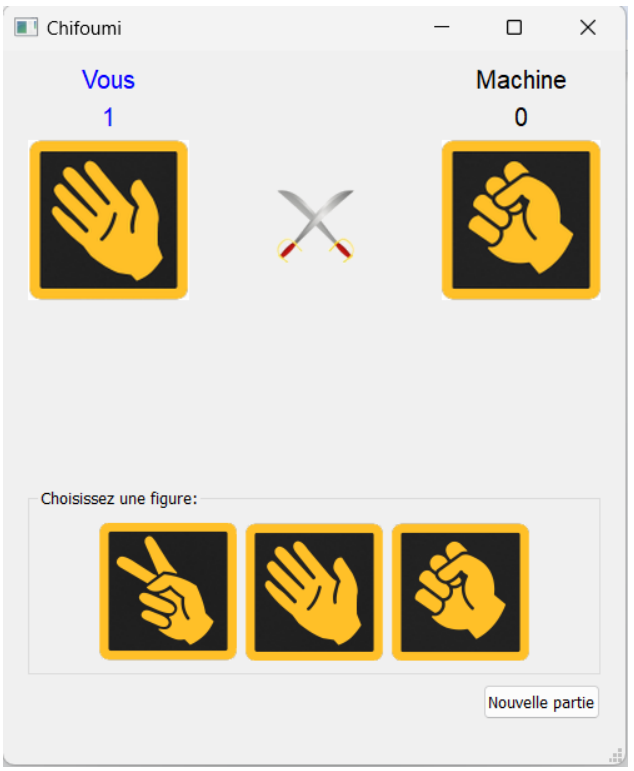
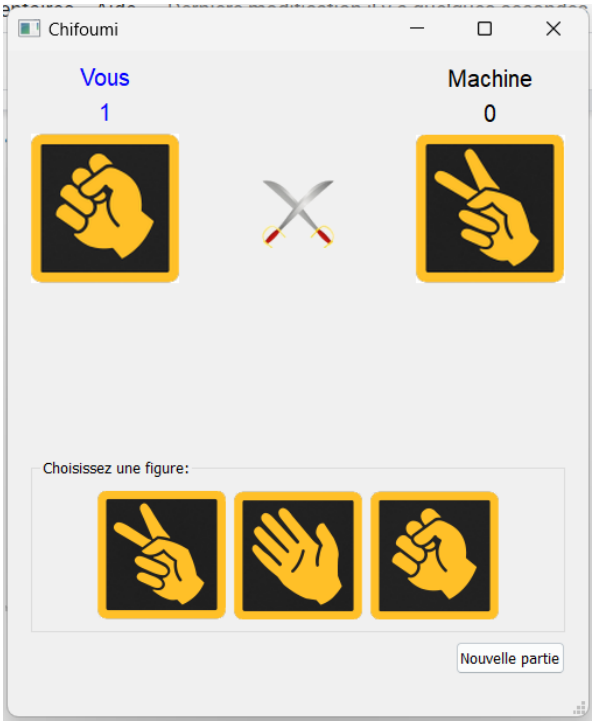
Décrire les tests prévus / réalisés pour montrer :

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

Le joueur joue ciseau La machine joue ciseau Egalité coupJoueur = ciseau coupMachine = ciseau	

Le joueur joue ciseau La machine joue papier Victoire du joueur coupJoueur = ciseau coupMachine = papier	
Le joueur joue ciseau La machine joue pierre Victoire de la machine coupJoueur = ciseau coupMachine = pierre	

Le joueur joue papier La machine joue ciseau Victoire de la machine coupJoueur = papier coupMachine = ciseau	
Le joueur joue papier La machine joue papier Egalité coupJoueur = papier coupMachine = papier	

Le joueur joue papier La machine joue pierre Victoire du joueur coupJoueur= papier coupMachine = pierre	
Le joueur joue pierre La machine joue ciseau Victoire du joueur coupJoueur=pierre coupMachine=ciseau	

Le joueur joue pierre La machine joue papier Victoire de la machine coupJoueur = pierre coupMachine = papier	
Le joueur joue Pierre La machine joue Pierre Egalité coupJoueur = pierre coupMachine = pierre