

Jeu NOM DU JEU

Rapport de projet

Ce document présente les différentes phases du projet NOM DU JEU réalisé dans le cadre de l’électif Génie logiciel orienté objet

Dossier rédigé par *Hugo BORSONI et Vincent FLATTOT.*

17*/01/2021*

Table des matières

[01. Cahier des charges 2](#_Toc61805974)

[A. Présentation du jeu Flow Free 2](#_Toc61805975)

[B. Présentation d’une phase de jeu (version mobile) 2](#_Toc61805976)

[C. Notion de niveaux 3](#_Toc61805977)

[02. Expression des besoins 3](#_Toc61805978)

[03. Conception 3](#_Toc61805979)

[04. Réalisation 3](#_Toc61805980)

[05. Conclusion 3](#_Toc61805981)

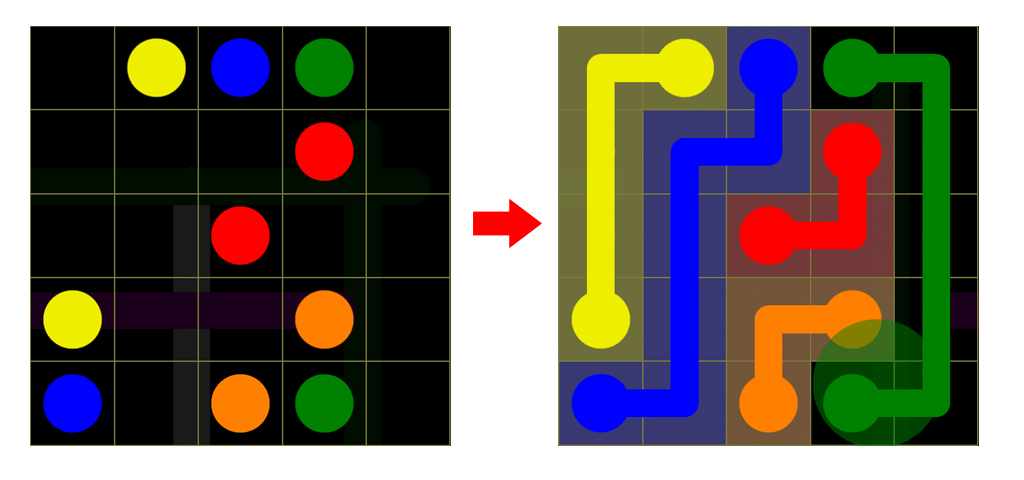
NB : les codes utilisés pour dessiner les différents diagrammes sont fournis dans le dossier code\_dossier rendu avec ce fichier. On pourra par exemple les entrer dans le site web plantuml.com afin d’obtenir les résultats graphiques.

# Cahier des charges

## Présentation du jeu Flow Free®

Le jeu dont nous nous inspirons est Flow Free®. Il s’agit d’un jeu de puzzle développé par le studio Américain Big Duck Games et sortit en juin 2012. Il s’agit initialement d’un jeu pour mobile et est disponible sur IOS et Android.

Le concept du jeu est assez simple. On part d’un plateau carré découpé en NxN cases. On dispose sur 2xn cases des « Plots » en forme de ronds. Il faut ensuite à l’aide de « Tuyaux colorés » réussir à relier chaque plot de la même couleur tout en s’assurant que l’ensemble {Tuyau+Plot} couvre bien l’ensemble des cases du Plateau et qu’aucun Tuyau ne croise un autre Tuyau .



*Une illustration d’une partie au début et à a fin*

## Présentation d’une phase de jeu (version mobile)

Dans la phase de jeu, l’utilisateur peut cliquer sur un plot puis tracer avec son doigt le chemin qu’il souhaite lui faire prendre. **Dans notre version, l’utilisateur cliquera avec sa souris sur un plot puis déplacera le tuyau grâce aux flèches de son clavier**

Si l’utilisateur lâche son doigt, le tuyau s’arrête sur la dernière case balayée par l’utilisateur. **De même, dans notre version, nous utiliserons ce procédé**

Si l’utilisateur fait croiser le nouveau tuyau avec un ancien, l’ancien tuyau est effacé jusqu’à la case précédant le croisement, le tuyau en construction reste celui que l’utilisateur était en train de construire. **Dans notre version du jeu, nous supprimerons le tuyau courant entièrement et l’utilisateur pourra recliquer sur un plot pour construire un nouveau tuyau**.

Lorsque l’utilisateur à réalisé le dernier mouvement avant la victoire, le jeu l’informe de la réussite et indique le nombre de tentatives qu’il a fallu pour réussir le niveau**. Nous allons essayer de reproduire ce comportement**

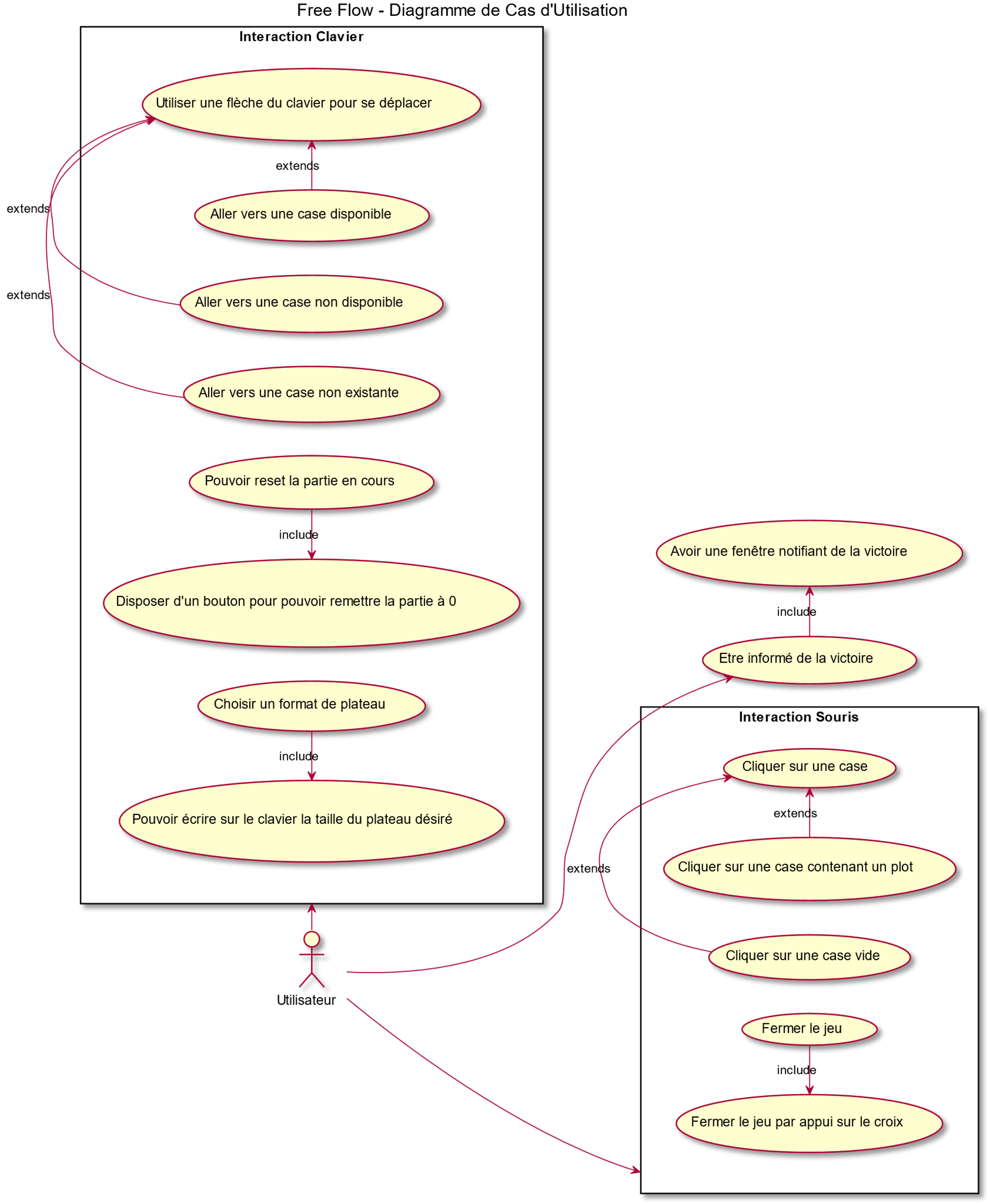
## Notion de niveaux

Dans le jeu, il existe différents niveaux auxquels on peut accéder lorsqu’on a réussi les niveaux précédents. **On essaiera dans ce projet de donner la possibilité à l’utilisateur de changer de « niveau » en changeant le nombre de cases du plateau**

# Expression des besoins

L’utilisateur du jeu a plusieurs attentes que nous récapitulons ici sous forme d’une liste et que nous illustrons via un diagramme de cas d’utilisation et de différents scénarios.

* Pouvoir sélectionner un plot pour créer un tuyau
* Pouvoir se déplacer dans le plateau avec les tuyaux
* Pouvoir remettre à zéro la partie
* Pouvoir savoir qu’il a gagné
* Pouvoir quitter le jeu

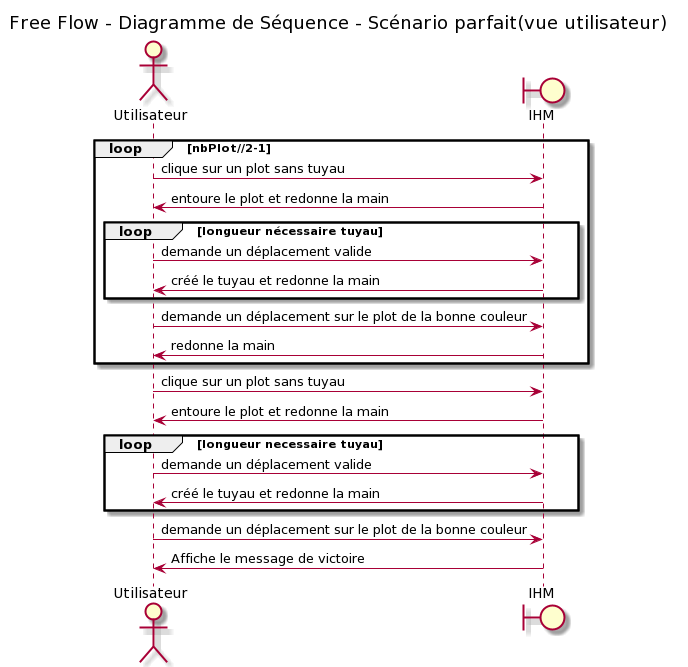


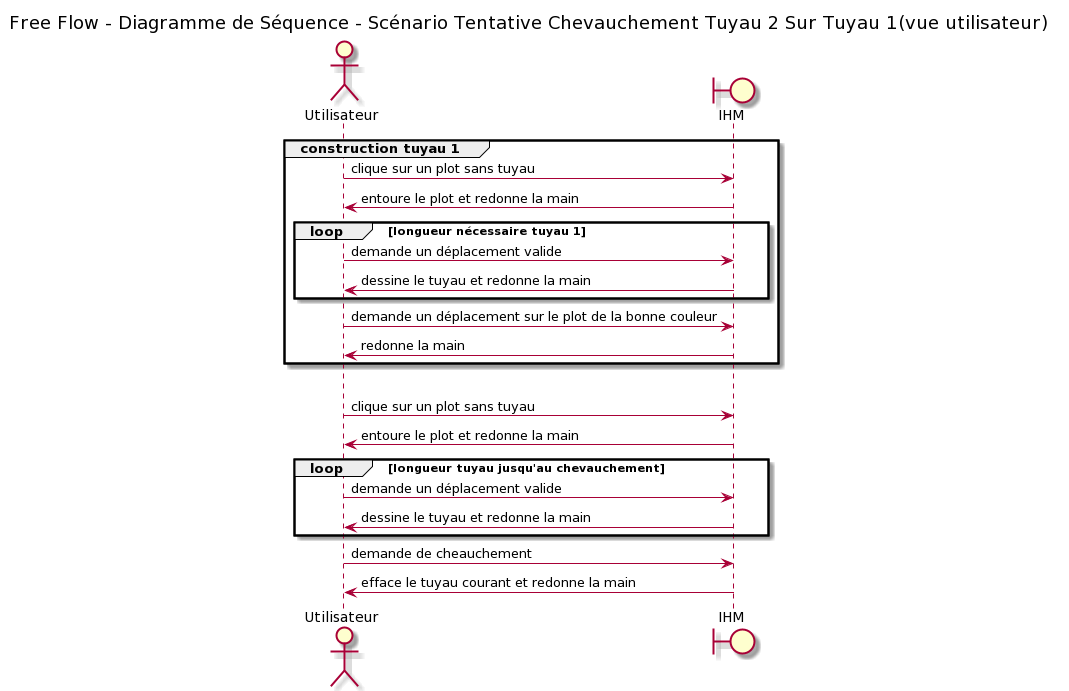
Ces cas d’utilisations peuvent être mis en parallèle avec les actions qu’ils vont nécessiter de la part de l’Interface Homme-Machine :

Une image contenant texte

Description générée automatiquement

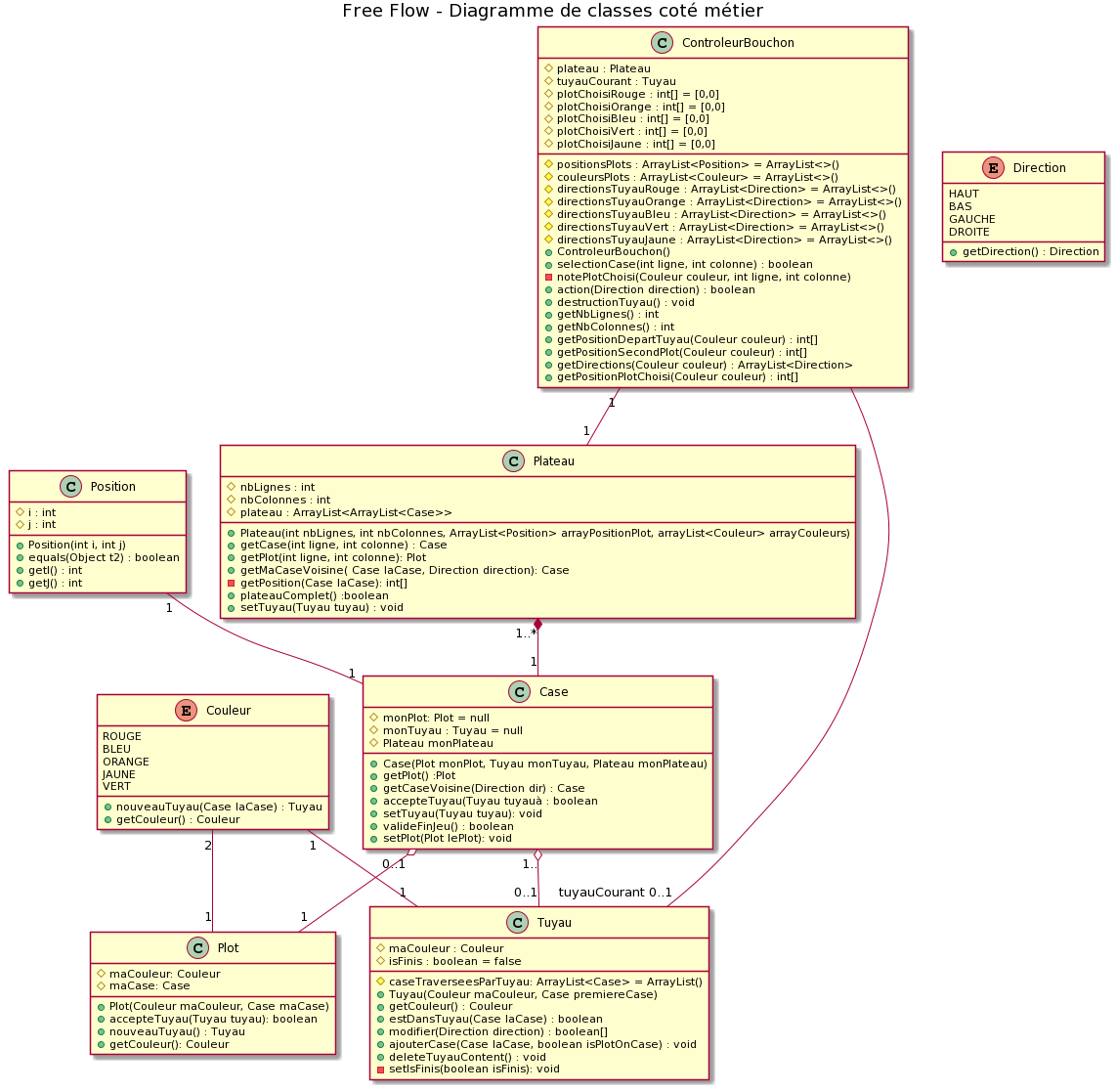
Afin de permettre une meilleure compréhension de ces éléments, on propose la modélisation d’un scénario « parfait » et d’un scénario avec une erreur du point de vue du joueur





# Conception

Pour commencer cette partie, on présente ci-dessous le diagramme de classe de notre application (coté machine) afin de pouvoir comprendre plus facilement les éléments en jeu.



# Réalisation

# Conclusion

Limites : lorsqu’on spamme le clavier ça chibre

# Annexe

Code pour le diagramme des cas d’utilisation :

diag\_usecase.txt

Code pour le diagramme de séquence des actions utilisateur

diag\_sequence\_useractions.txt

Code pour le diagramme de séquence scénario « parfait »

diag\_sequence\_perfectcase.txt

Code pour le diagramme de séquence scénario « erreur »

diag\_sequence\_errorcase.txt

Code pour le diagramme de classes :

diag\_classe.txt