



## **RAPPORT FINAL DEVELOPPEMENT INFORMATIQUE AVANCE**

**THEME : Abalone**

- Joël YEPGANG





## Introduction

Dans le cadre notre apprentissage au cours de développement informatique avancé, il nous a été demandé de réaliser un projet au choix (application ou jeux) selon un canevas défini par nos enseignants respectant des consignes et des contraintes.

Commencé le 28 juin 2020 sous l'encadrement de Virginie Van den Schrieck , j'ai commencé par la rédaction du cahier de charge .

J'ai opté pour la création d'un jeu apprécié par toutes les générations : Abalone

Il se joue au minimum par 2 joueurs à l'aide des boules sur un hexagone

## Etapas de réalisations

Les étapes ci-dessous décrivent la procédure suivie tout le long de notre projet.

- Rédiger un cahier de charge validé par l'encadreur
- Mettre en place un diagramme UML (Unified Modeling language)
- Travailler de manière autonome et push sur GitHub

<https://github.com/projet2019/Abalone/tree/joel>

## Cahier de charge

A ce niveau, j'ai élaboré , l'objectif à atteindre et la description au travers des user stories que j'ai soumis à mon encadreur qui m'a fait un retour.

<https://github.com/projet2019/Abalone/wiki/Cahier-de-charge>

## Répartition des tâches

Le travail se faisant de manière autonome, la première des choses fut de définir les assets constituant le projet. Et donc la première des choses fut de réaliser un diagramme uml cependant au fur et à mesure de mon évolution dans le projet ce dernier se voyait modifier progressivement.

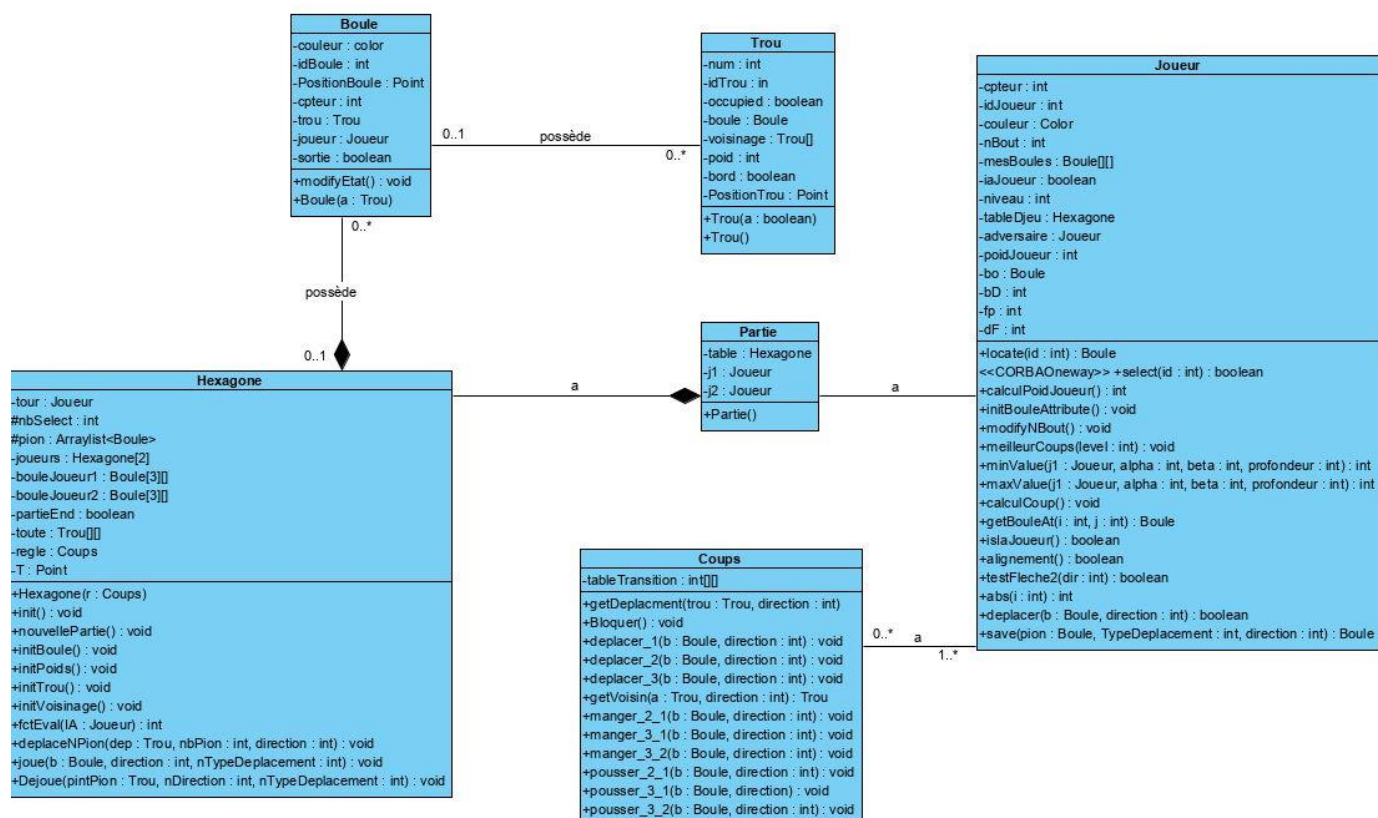


Puis vint la mise place du MVC ( model, vu , contrôleur). Ainsi, j'ai procédé à la mise en place des différentes dépendances notamment dans le modèle où j'ai créé les différentes classes.

Notons que à la différence d'un jeu classique entre deux joueurs ,j'ai voulu rajouter l'option permettant qu'un joueur puisse jouer avec la machine.

## I- Diagramme UML

Dans la phase d'analyse, j'ai mis en place le diagramme UML pour représenter l'architecture de mes différentes classes et leur interaction. Celle-ci m' a permis d'extraire les différentes fonctionnalités dont j'avais besoins.



N.B : Pour des raisons de lisibilités les getters et setters n'ont pas été représenté .

## Description



Le diagramme ainsi représenter , met en avant les différentes interactions entre elles.

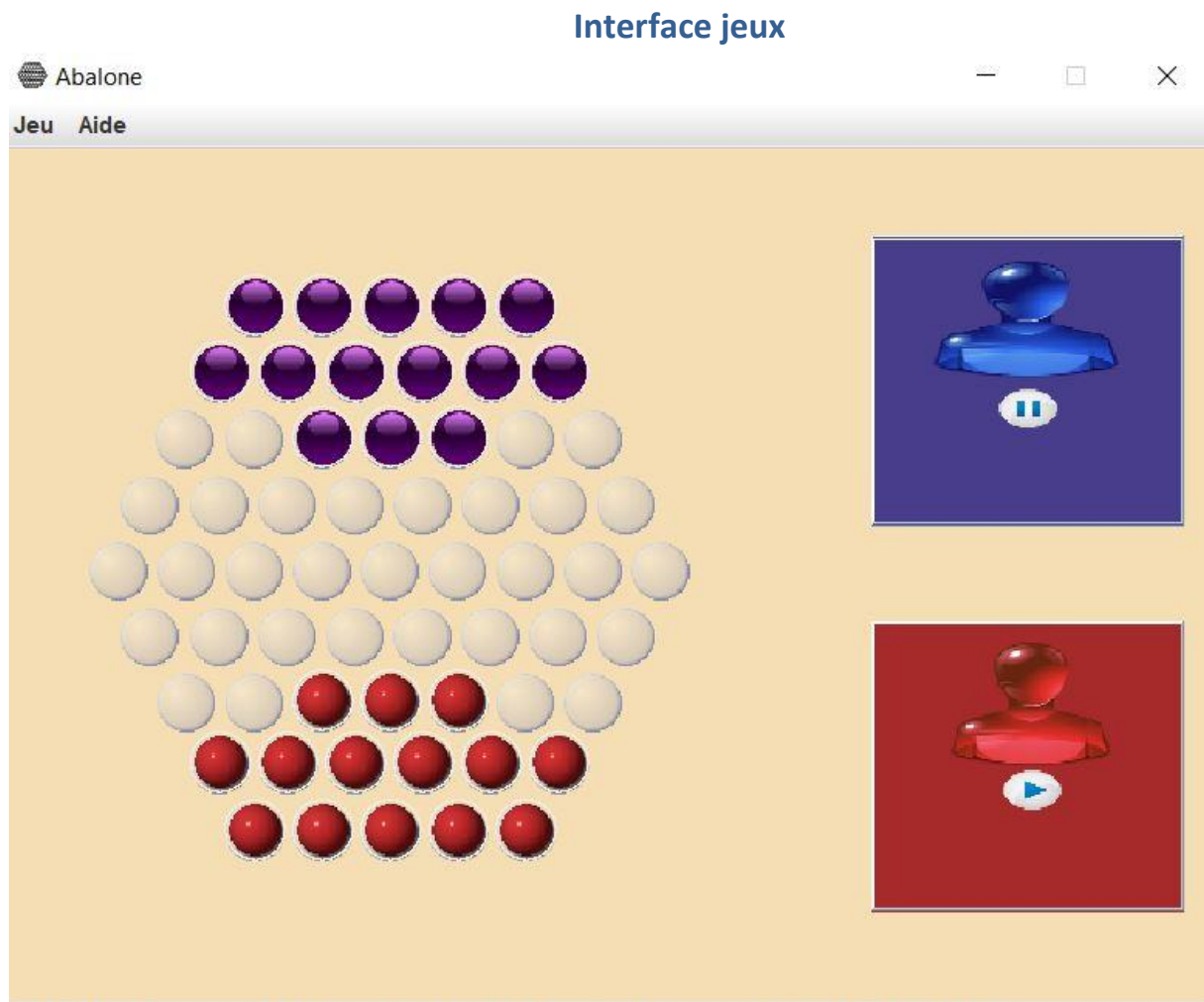
Ainsi, une **partie** ne peut être initialiser que par deux joueurs ou un joueur Vs machine et ces derniers sont autoriser à effectuer des **coups**.

La partie pourra se dérouler sur tapis (hexagone), qui contiendra **des trous, des boules**.

Package Controller : contient une classe Controller qui permet de faire lien entre le Model et la vue.

Package Vue : contient la classe console qui hérite les propriétés de la classe

## II- Description fonctionnelle et Modèle view



1- lancement du jeu : Lancer le package Abalone en run as java application.

2- lancement des clients(un ou deux





### III- Difficultés rencontrées

Après avoir défini les assets constituant notre package model, j'ai rencontré plusieurs problèmes au niveau de l'implémentation de l'interface des joueurs et de la mise en place d'une GUI ce qui a engendré des bugs à plusieurs niveaux de mon application. Que j'ai pu résoudre la plupart de mon problème.

L'autre difficulté résidait sur l'ampleur d'un tel projet donc je n'ai pas assez mesuré sa robustesse. Car au fur à mesure de mon évolution, de nouvelles solutions apparaissaient.

### IV- CONCLUSION

Ce projet s'est révélé très enrichissant dans la mesure où il a consisté en une approche concrète du métier d'informaticien. En la prise d'initiative, de plus c'est une première même pour moi développer une application tout seul.

De plus, il m'a permis d'appliquer les connaissances apprises au cours de nos différents travaux pratiques en classe et du cours théoriques ingénierie.

Pour des raisons académiques notamment l'échéance de soumission du travail, je n'ai pu terminer sa conception.

Cependant malgré les difficultés je reste positif à l'idée de le continuer.

