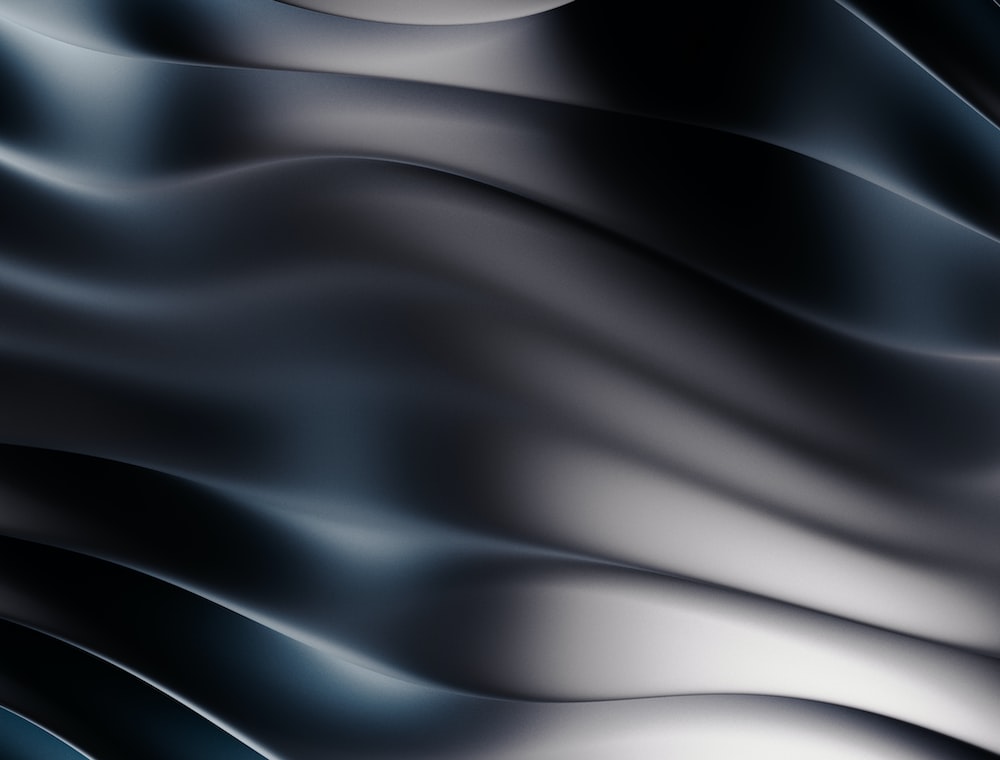


NimRL

Documentation Technique

Marco Rodrigues Marques

****

# Table des matières

[Table des matières 2](#_Toc134454397)

[Introduction 3](#_Toc134454398)

[Étude d’opportunité 3](#_Toc134454399)

[Analyse fonctionnelle 3](#_Toc134454400)

[Fonctionnalités l’application : 3](#_Toc134454401)

[Liste de fonctionnalités : 4](#_Toc134454402)

[Description de l’interface : 6](#_Toc134454403)

[Zone Principale : 6](#_Toc134454404)

[Analyse organique 7](#_Toc134454405)

[Méthodologie : 7](#_Toc134454406)

[ Exigences 7](#_Toc134454407)

[ Conception 7](#_Toc134454408)

[ Implémentation 7](#_Toc134454409)

[ Validation 7](#_Toc134454410)

[Architecture du projet : 8](#_Toc134454411)

[Apprentissage par renforcement : 9](#_Toc134454412)

[Améliorations possibles 9](#_Toc134454413)

[Conclusion 9](#_Toc134454414)

[Remerciements 9](#_Toc134454415)

[Table des illustrations 9](#_Toc134454416)

[Glossaire 9](#_Toc134454417)

[Sources 9](#_Toc134454418)

[Annexes 9](#_Toc134454419)

# Introduction

Ce document est un rapport des détails techniques de la réalisation du projet NimRL, destiné aux futures personnes impliquées dans ce projet.

NimRL c’est une application qui a pour but la visualisation d’un processus d’apprentissage par renforcement démontré sur le jeu de Nim.

# Étude d’opportunité

Ce projet porte sur le thème de l’Intelligence Artificielle, cette dernière est de plus en plus présente dans nos vies, ce qui fait que de plus en plus les personnes cherchent à comprendre le fonctionnement de ces technologies.

NimRL est fait pour les aider.

Il existe plusieurs méthodes d’apprentissage automatique, celle qui est démontrée dans l’application c’est la méthode par renforcement qui est utilisée dans des applications comme :

* Google Translate (Traitement automatique des langues)
* Netflix (Recommandations personnalisés)
* Voitures autonomes

NimRL aide à comprendre l’apprentissage automatique par renforcement grâce à la visualisation d’un processus d’apprentissage simple.

# Analyse fonctionnelle

## Fonctionnalités l’application :

Le jeu peut fonctionner selon trois modes :

* Joueur Humain contre joueur Humain ;
* Joueur Humain contre joueur IA ;
* Joueur IA contre joueur IA.

L’application permet à l’utilisateur de contrôler les utilisateurs Humains.

Les joueurs IA sont contrôlés par une Intelligence Artificielle, qui apprend grâce aux actions choisies lors des parties jouées.

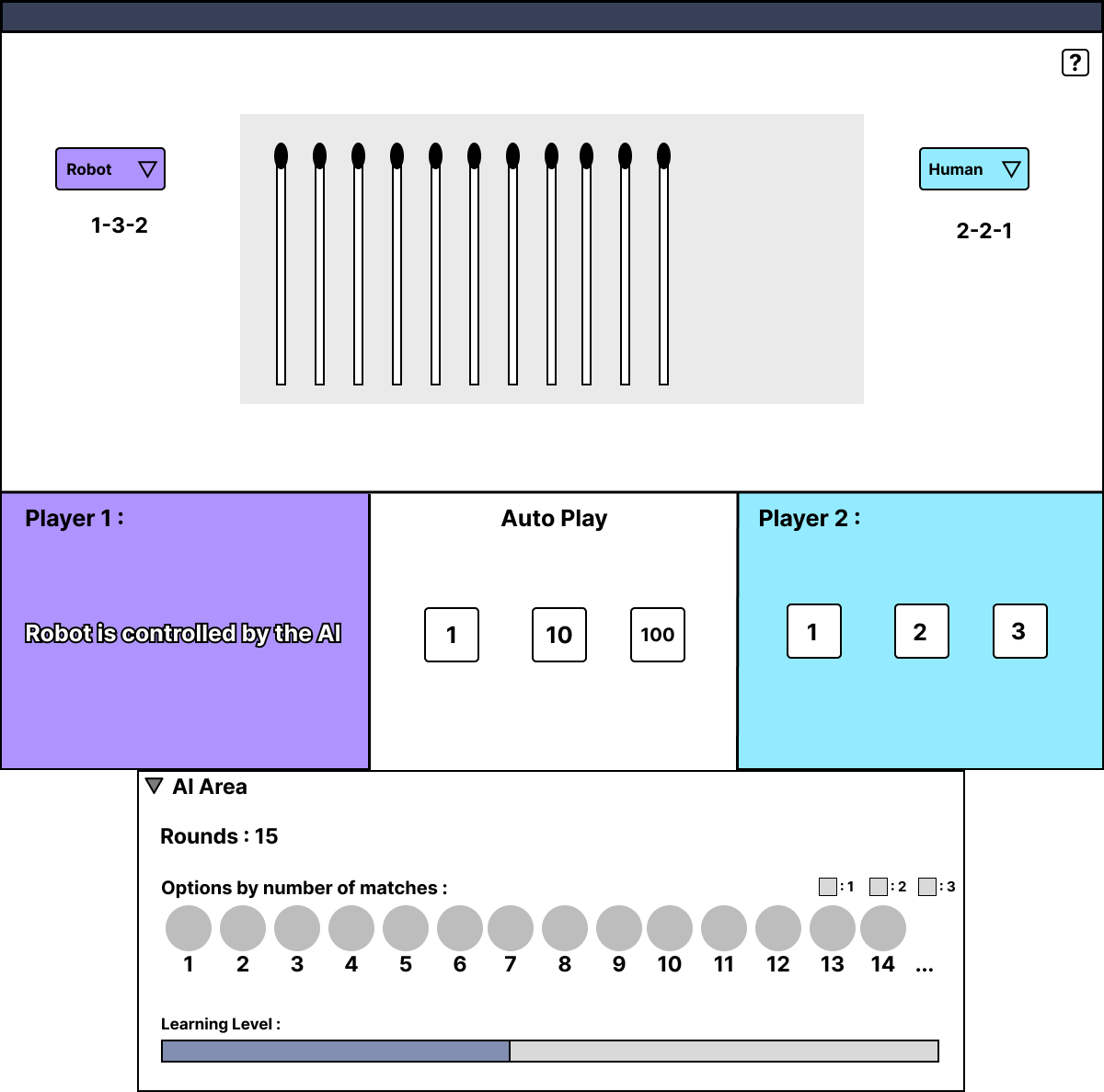
L’application permet également d’accélérer le processus d’apprentissage en faisant des parties automatiques.

Différentes informations sur l’Intelligence Artificielle et sur le processus d’apprentissage sont affichées.

## Liste de fonctionnalités :

|  |  |
| --- | --- |
| **Nom** | 01 : Modes de jeu |
| **Description** | En tant qu'utilisateur je dois pouvoir choisir entre trois modes de jeu différents :  - Joueur Humain contre Joueur Humain  - Joueur Humain contre Joueur IA  - Joueur IA contre Joueur IA |
| **Priorité** | Vital |
|  |  |
| **Nom** | 02 : Controller un Joueur Humain |
| **Description** | En tant qu'utilisateur je dois pouvoir contrôler un Joueur Humain |
| **Priorité** | Essentiel |
|  |  |
| **Nom** | 03 : Parties automatiques |
| **Description** | En tant qu'utilisateur je dois pouvoir faire des parties automatiques (mode Joueur IA contre Joueur IA) |
| **Priorité** | Essentiel |
|  |  |
| **Nom** | 04 : Niveau d'apprentissage de l'IA |
| **Description** | En tant qu'utilisateur je dois pouvoir voir le niveau d'apprentissage de l'IA |
| **Priorité** | Essentiel |
|  |  |
| **Nom** | 05 : Compréhension de l'apprentissage automatique (Décisions) |
| **Description** | En tant qu'utilisateur je dois pouvoir comprendre comment l'IA prend des décisions |
| **Priorité** | Confort |
|  |  |
| **Nom** | 06 : Compréhension de l'apprentissage automatique (Auto-amélioration) |
| **Description** | En tant qu'utilisateur je dois pouvoir comprendre comment l'IA s'auto améliore |
| **Priorité** | Confort |
|  |  |
| **Nom** | 07 : Sauvegarde de l'état de l'application |
| **Description** | En tant qu'utilisateur je veux que l'application garde son état lors de la fermeture |
| **Priorité** | Essentiel |
|  |  |
| **Nom** | 08 : Accès au manuel utilisateur |
| **Description** | En tant qu'utilisateur je veux avoir accès à un manuel utilisateur depuis l'application |
| **Priorité** | Confort |
|  |  |
| **Nom** | 09 : Affichage des allumettes |
| **Description** | En tant qu'utilisateur je dois avoir un affichage des allumettes de la partie courante |
| **Priorité** | Essentiel |
|  |  |
| **Nom** | 10 : Affichage du joueur courant |
| **Description** | En tant qu'utilisateur je dois avoir un affichage qui m'indique, pour la partie courante, le joueur qui doit jouer |
| **Priorité** | Essentiel |
|  |  |
| **Nom** | 11 : Nombre d'allumettes prises |
| **Description** | En tant qu'utilisateur je dois avoir un affichage du nombre d'allumettes que les joueurs ont pris |
| **Priorité** | Essentiel |

## Description de l’interface :



C

B

A

Fig. 1 : Interface générale de l’application

L’affichage est divisé en trois parties principales :

* Zone Principale (A) ;
* Zone des contrôles (B) ;
* Zone IA (C).

Ces zones ont été définies dans le but de rendre l’application plus ergonomique, chaque zone a une responsabilité, et la zone IA, qui contient beaucoup d’informations, peut être fermée ou ouverte par l’utilisateur.

Pour distinguer les deux joueurs, une couleur a été assignée à chacun, violet pour le premier et bleu clair pour le deuxième.

### Zone Principale :

Zone qui contient les informations principales du jeu.

Elle contient un affichage des allumettes en jeu, représentées par une image, cet affichage est mis à jour au cours du jeu.

On y voit également, pour chaque joueur, son type (humain ou robot) et les actions qu’il a pris pour la partie courante.

On peut également changer le type de chaque joueur grâce à deux boîtes combinées.

### Zone des contrôles :

Zone qui contient les principaux contrôles de l’application.

Elle est elle aussi divisée en trois parties : contrôles du joueur 1, contrôles de partie automatique et contrôles du joueur 2.

Les zones des contrôles des joueurs contiennent chacune trois boutons qui permettent de choisir une, deux ou trois allumettes lorsque le joueur correspondant est un humain, lorsqu’il est un robot ces contrôles sont remplacés par un message qui indique que le robot est contrôlé par l’IA.

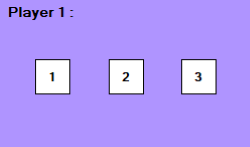


Fig. 2 : Zone des contrôles du joueur 1, lorsque le joueur est un humain

Une image contenant texte

Description générée automatiquement

Fig. 3 : Zone des contrôles du joueur 1, lorsque le joueur est un robot

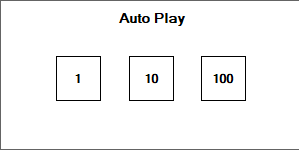
La zone qui contrôle les parties automatiques n’est qu’affichée si les deux joueurs sont des robots, elle a trois boutons qui permettent d’effectuer 1, 10 ou 100 parties automatiques.

Fig. 4 : Zone de contrôle des parties automatiques

### Zone IA :

# Analyse organique

## Méthodologie :

Pour ce projet, j'ai décidé d'organiser mes tâches selon une version simplifiée du

Modèle en cascade, qui est composée de 4 phases :

Une image contenant diagramme

Description générée automatiquement

Fig. 5 : Modèle en cascade utilisé

* Exigences : Analyse des besoins du projet selon le cahier des charges ;
* Conception : Élaboration de l'architecture logicielle (Diagramme de classes, Conception d'interfaces, Product Backlog) ;
* Implémentation : Réalisation du logiciel et des différents documents selon l’architecture définie précédemment ;
* Validation : Vérification du produit selon les exigences.

Normalement, le modèle en cascade comprend les phases "Analyse" et "Maintenance", mais dans le but d'adapter cette méthodologie au TPI j'ai décidé de simplifier les phases "Exigences" et "Analyse" en une seule vu que le cahier de charges du projet nous est déjà fourni, j'ai également exclu la phase "Maintenance" car le projet n'est pas publié.

## Architecture du projet :

Le projet est architecturé selon le modèle MVC, qui décompose le projet en trois couches, chacune avec un rôle précis, pour que le projet soit mieux organisé.

Ces couches sont :

* **M**odèle : Couche qui gère la logique métier de l’application ;
* **V**ue : Couche qui permet à l’utilisateur d’interagir avec l’application et l’affichage des données ;
* **C**ontrôleur : Gère les échanges entre le modèle et la vue.

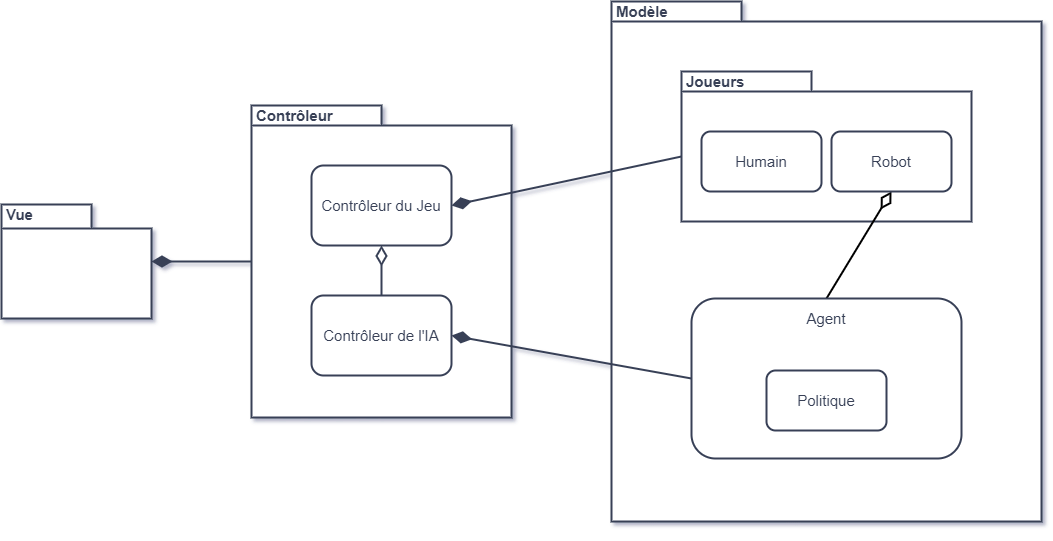


Fig. 6 : Architecture du projet simplifiée

Spécifiquement pour cette application j’ai prévu deux contrôleurs, un contrôleur pour le jeu et un contrôleur pour l’IA, les deux appartiennent à la vue.

Le contrôleur du Jeu contient le contrôleur de IA, il possède également deux joueurs, ces derniers peuvent être humains ou robots, les joueurs humains sont contrôlés par l’utilisateur alors que les joueurs robots sont contrôlés par l’Intelligence Artificielle.

Le contrôleur de IA possède un agent, qui est l’élément de l’apprentissage par renforcement qui apprend et prend des décisions, cet agent possède une politique qui définit la façon dont l’agent doit agir.

## Apprentissage par renforcement :

# Améliorations possibles

# Conclusion

# Remerciements

# Table des illustrations

# Glossaire

# Sources

# Annexes