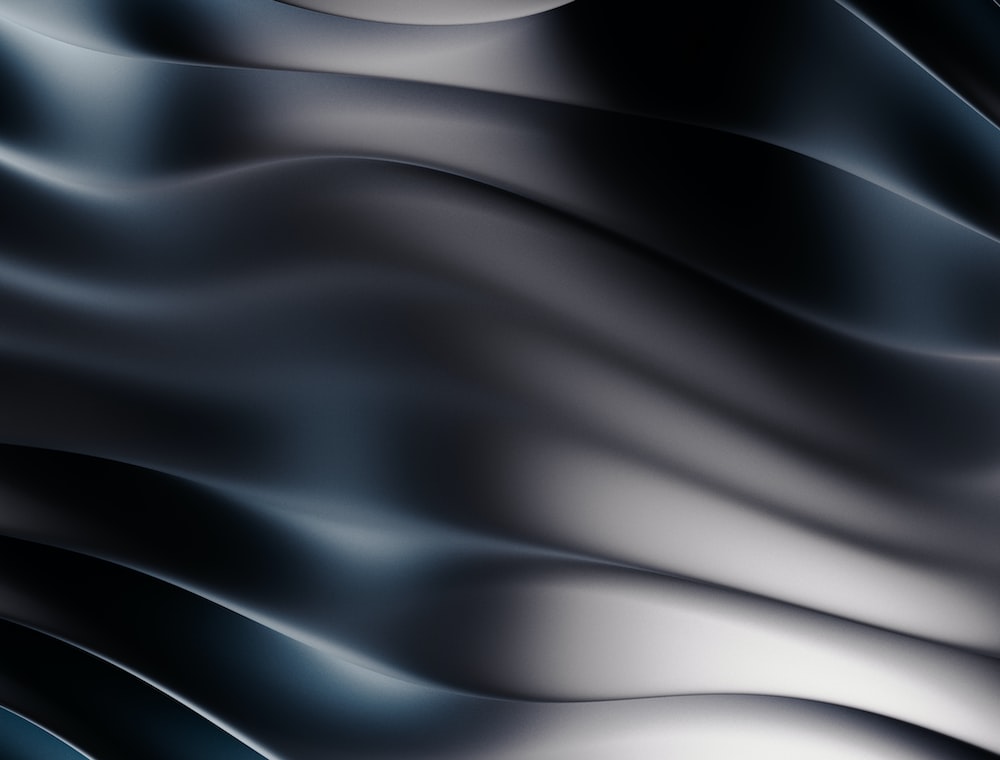


NimRL

Documentation Technique

Marco Rodrigues Marques

****

# Table des matières

[Table des matières 2](#_Toc134608756)

[Introduction 4](#_Toc134608757)

[Étude d’opportunité 4](#_Toc134608758)

[Rappel de l’énoncé : 4](#_Toc134608759)

[Organisation : 4](#_Toc134608760)

[Objectif : 4](#_Toc134608761)

[Règles du jeu : 5](#_Toc134608762)

[Contraintes : 5](#_Toc134608763)

[Livrables : 5](#_Toc134608764)

[Analyse fonctionnelle 6](#_Toc134608765)

[Fonctionnalités l’application : 6](#_Toc134608766)

[Liste de fonctionnalités : 6](#_Toc134608767)

[Description de l’interface : 8](#_Toc134608768)

[Zone Principale : 9](#_Toc134608769)

[Zone des contrôles : 9](#_Toc134608770)

[Zone IA : 10](#_Toc134608771)

[Analyse organique 11](#_Toc134608772)

[Méthodologie : 11](#_Toc134608773)

[ Exigences 11](#_Toc134608774)

[ Conception 11](#_Toc134608775)

[ Implémentation 11](#_Toc134608776)

[ Validation 11](#_Toc134608777)

[Architecture du projet : 11](#_Toc134608778)

[Apprentissage par renforcement : 12](#_Toc134608779)

[Généralités : 12](#_Toc134608780)

[Application dans le projet : 12](#_Toc134608781)

[Améliorations possibles 13](#_Toc134608782)

[Conclusion 13](#_Toc134608783)

[Remerciements 13](#_Toc134608784)

[Table des illustrations 13](#_Toc134608785)

[Glossaire 13](#_Toc134608786)

[Sources 13](#_Toc134608787)

[Annexes 13](#_Toc134608788)

# 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.

## Rappel de l’énoncé :

### Organisation :

* Élève : Marco Rodrigues Marques [marco.rdrgs2@eduge.ch](mailto:marco.rdrgs2@eduge.ch)
* Maître : Alec Beney [edu-beneya@eduge.ch](mailto:edu-beneya@eduge.ch)
* Experts : Damien Schorer [damienschorer@gmail.com](mailto:damienschorer@gmail.com)

Robin Bouille [robin.bouille@gmail.com](mailto:robin.bouille@gmail.com)

### Objectif :

Réalisation d’une application C# WindowsForm qui permet la visualisation d’un processus d’apprentissage automatique par renforcement sur le Jeu de Nim.

### Règles du jeu :

Le jeu se joue à deux joueurs au tout par tour. Chaque partie commence avec un certain nombre d’objets (dans ce cas 20) et chaque tour les joueurs peuvent prendre un, deux ou trois objets (dans ce cas des allumettes), celui qui prend le dernier est le perdant.

### Contraintes :

L’application doit être développée en respectant toutes les contraintes habituelles en matière de conception et de programmation d’application orientée objet (COO et POO) et respecter strictement la loi de Déméter.

Dans le code, les conventions de codage de l’école doivent être strictement respectées. La structure des classes est indiquée sous forme de commentaires et le choix des identificateurs doit être représentatif.

Le diagramme de classes doit être développé (par exemple avec la méthode des cartes CRC) et fourni dans la documentation en respectant strictement la norme UML.

L’interface utilisateur doit respecter tous les principes de base d’ergonomie dispensés durant le cursus de formation.

Les éléments d’interface et les composants doivent permettre une bonne compréhension du processus d’apprentissage automatique même pour un utilisateur inexpérimenté.

Le processus d’apprentissage par renforcement doit être détaillé dans la documentation technique.

### Livrables :

* Rapport de projet : 3 exemplaires
* Manuel utilisateurs : 3 exemplaires
* Résumé du rapport du TPI : 1 exemplaires
* Journal de bord : 1 exemplaires

# 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.

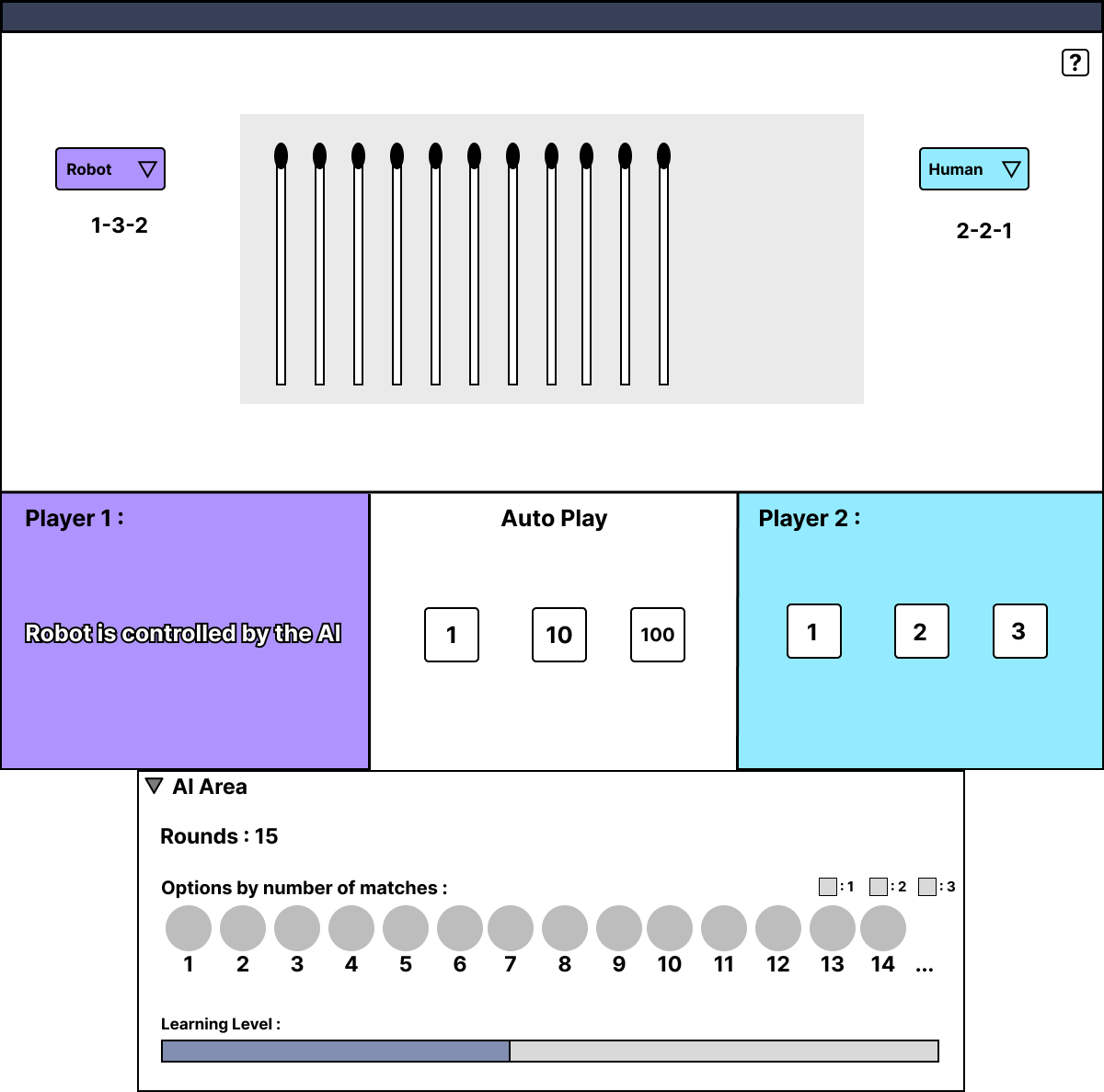
L’application permet également d’accélérer le processus d’apprentissage en faisant des parties automatiques.

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.

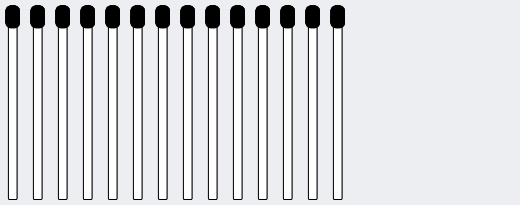


Fig. 2 : Affichage des allumettes

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.

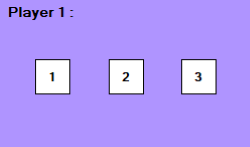


Fig. 2 : Zone des contrôles du joueur 1, lorsque le joueur est un humain

Une image contenant texte

Description générée automatiquement

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

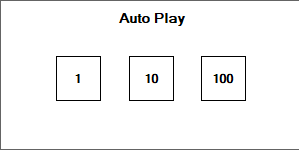
La zone qui contrôle les parties automatiques n’est qu’affichée si les deux joueurs sont des robots, elle a trois boutons qui permettent d’effectuer 1, 10 ou 100 parties automatiques.

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.

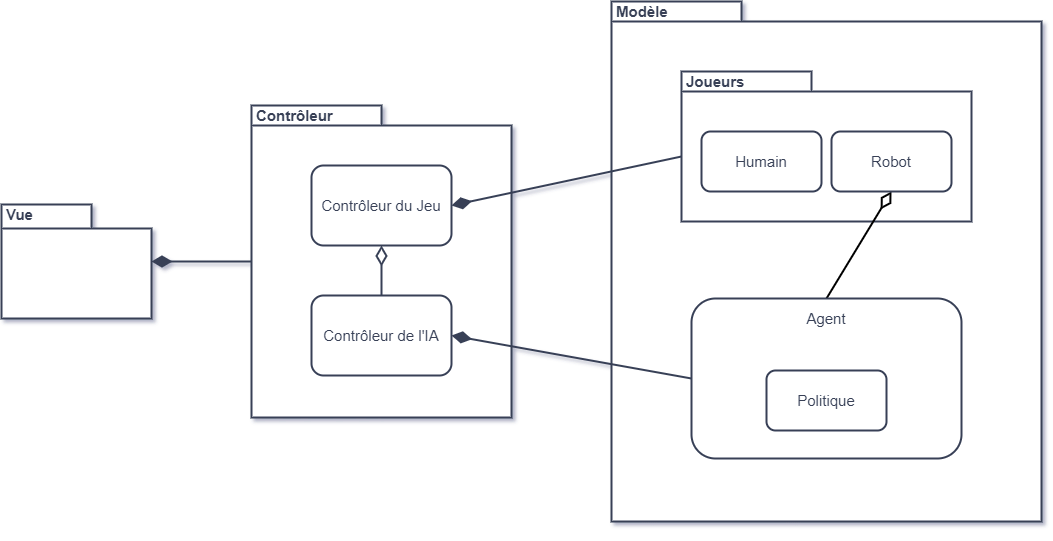


Fig. 6 : Architecture du projet simplifiée

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 :

### Généralités :

L’apprentissage par renforcement est une méthode d’apprentissage automatique ou on ne dit pas à l’apprenant quelles actions faire, il doit apprendre en explorant l’environnement et découvrant quelles actions sont le plus récompensées.

L’apprentissage par renforcement comprend quelques composants essentiels :

* Agent : Apprend et prend des décisions ;
* Politique : Détermine comment l’agent doit agir ;
* Récompenses : Objectifs dans un problème d’apprentissage par renforcement ;
* Fonction de valeur : Décrit ce qui est bon dans le long terme ;
* Modèle d’environnement : Imite l’environnement ou l’agent se trouve et permet de déduire comment l’environnement va se comporter.

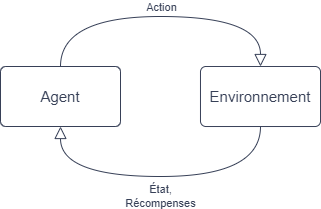
L’agent interagit constamment avec l’environnement, il prend une action selon l’état de ce dernier, cette action modifie l’environnement et l’agent reçoit en retour une récompense.

Fig. 7 : Interactions entre l’agent et l’environnement

### Une image contenant diagramme Description générée automatiquementApplication dans le projet :

# Améliorations

# Conclusion

# Remerciements

# Table des illustrations

# Glossaire

# Sources

# Annexes