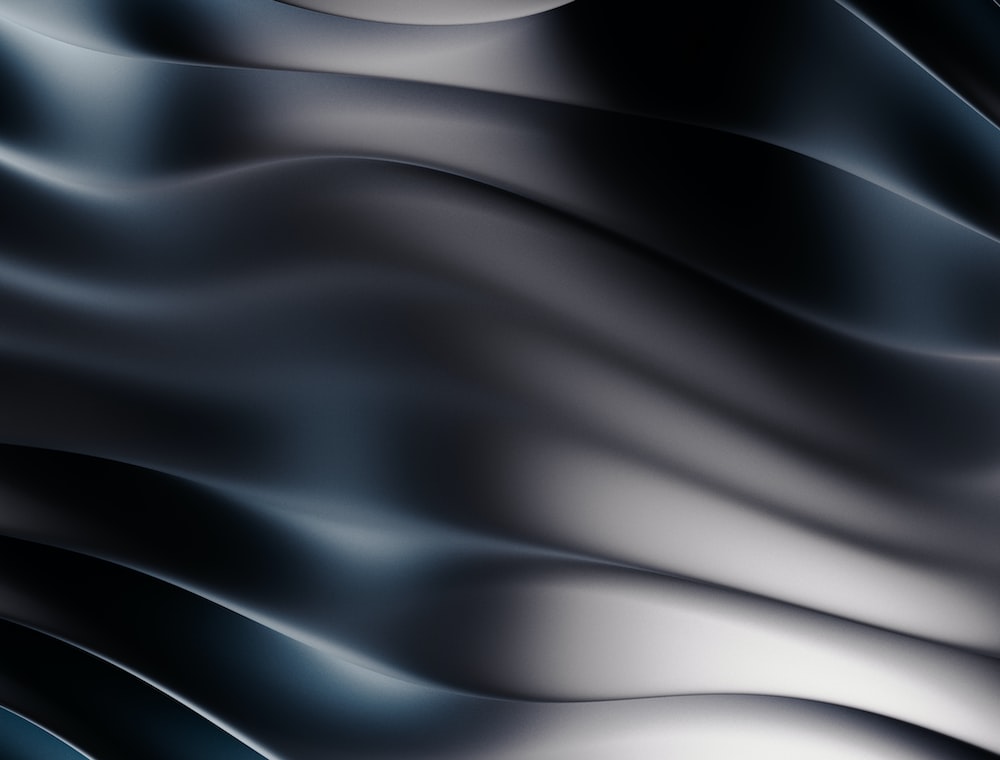


NimRL

Documentation Technique

Marco Rodrigues Marques

****

# Table des matières

[Table des matières 2](#_Toc133492499)

[Introduction 3](#_Toc133492500)

[Description de l’application : 3](#_Toc133492501)

[Étude d’opportunité 3](#_Toc133492502)

[Analyse fonctionnelle 3](#_Toc133492503)

[Analyse organique 3](#_Toc133492504)

[Méthodologie : 3](#_Toc133492505)

[ Exigences 4](#_Toc133492506)

[ Conception 4](#_Toc133492507)

[ Implémentation 4](#_Toc133492508)

[ Validation 4](#_Toc133492509)

[Améliorations possibles 4](#_Toc133492510)

[Conclusion 4](#_Toc133492511)

[Remerciements 4](#_Toc133492512)

[Table des illustrations 4](#_Toc133492513)

[Glossaire 4](#_Toc133492514)

[Sources 4](#_Toc133492515)

[Annexes 4](#_Toc133492516)

# 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, une meilleure compréhension du fonctionnement des algorithmes d’apprentissage automatique utilisés par ces intelligences aide à les démystifier.

Beaucoup d’applications existantes utilisent de l’apprentissage automatique, parmi elles :

* Google Translate (Traitement automatique des langues)
* Netflix (Recommandations personnalisés)
* Gmail (Détection de spam)

NimRL aide à comprendre l’apprentissage automatique par renforcement grâce à la visualisation d’un processus d’apprentissage, ce processus est évidemment plus simple que ceux utilisés par les applications citées ultérieurement.

# 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 a 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 |

# 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 :

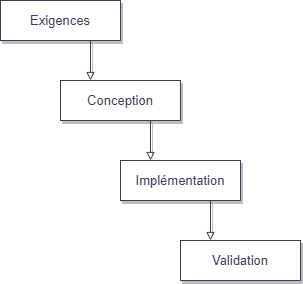


Fig. 1 : Modèle en cascade utilisé

* 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.

# Améliorations possibles

# Conclusion

# Remerciements

# Table des illustrations

# Glossaire

# Sources

# Annexes