

# Rapport de Projet - QUARTO

Master 1 GIL - Université de Rouen

Année universitaire 2024-2025

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Objectifs du projet</b>	<b>2</b>
<b>3</b>	<b>Technologies utilisées</b>	<b>2</b>
<b>4</b>	<b>Structure de l'application</b>	<b>2</b>
<b>5</b>	<b>Fonctionnement du jeu</b>	<b>2</b>
5.1	Règles de base . . . . .	2
5.2	Moteur du jeu . . . . .	2
5.3	Affichage dynamique . . . . .	2
<b>6</b>	<b>Intelligences Artificielles (IA)</b>	<b>3</b>
6.1	Niveau 1 : IA Aléatoire . . . . .	3
6.2	Niveau 2 : IA Heuristique . . . . .	3
6.3	Niveau 3 : IA Minimax . . . . .	3
<b>7</b>	<b>Interface graphique</b>	<b>3</b>
<b>8</b>	<b>Fonctionnalités avancées</b>	<b>4</b>
<b>9</b>	<b>Difficultés rencontrées</b>	<b>4</b>
<b>10</b>	<b>Conclusion</b>	<b>4</b>

# 1 Introduction

Le projet **Quarto** s'inscrit dans le cadre de l'unité d'enseignement *Théorie des jeux*. L'objectif est de développer une application complète et interactive du jeu Quarto en utilisant le langage **Python** avec la bibliothèque **Pygame**.

## 2 Objectifs du projet

- Réaliser une interface graphique professionnelle et immersive.
- Implémenter toutes les règles du jeu Quarto.
- Créer différents niveaux d'intelligence artificielle (IA).
- Gérer les modes *Joueur vs Joueur* et *Joueur vs IA*.

## 3 Technologies utilisées

- **Python 3.12.3**
- **Pygame 2.6.1**
- **Git - GitHub** pour la gestion de version
- **Overleaf** pour la rédaction de ce rapport

## 4 Structure de l'application

L'application est divisée en modules :

- `main.py` : Menu principal (choix de la langue, lancement du jeu)
- `choix.py` : Sélection du mode de jeu
- `jeu.py` : Lancement de la partie avec IA ou non
- `partie.py` : Logique centrale du jeu Quarto
- `players/` : Implémentation des joueurs (humain et IA)

## 5 Fonctionnement du jeu

### 5.1 Règles de base

Le jeu est fidèle aux règles officielles : aligner 4 pièces partageant **une caractéristique commune**. Chaque joueur choisit une pièce pour l'autre.

### 5.2 Moteur du jeu

La logique repose sur deux grilles :

- Le *plateau* principal de jeu (4x4).
- La *réserve* de pièces à disposition.

Chaque pièce est définie par 4 attributs binaires.

### 5.3 Affichage dynamique

- Affichage de la pièce sélectionnée

- Mise à jour du plateau
- Affichage du vainqueur
- Avatars, messages stylisés

## 6 Intelligences Artificielles (IA)

### 6.1 Niveau 1 : IA Aléatoire

*IA\_level1* choisit et joue une pièce de manière aléatoire.

**Avantage** : rapide

**Limite** : très faible stratégie

### 6.2 Niveau 2 : IA Heuristique

*IA\_level2* applique des règles simples pour bloquer un alignement ou chercher une opportunité.

### 6.3 Niveau 3 : IA Minimax

*IA\_level3* utilise un algorithme de recherche minimax avec profondeur limitée pour anticiper les coups adverses.

## 7 Interface graphique

- Menu d'accueil avec sélection de langue
- Sélection du mode de jeu (dropdown)
- Plateau et réserve stylisés avec fond désert spatial
- Affichage des règles dans deux langues



FIGURE 1 – Interface de sélection du mode de jeu

## 8 Fonctionnalités avancées

- Mode multilingue (Fr, Es)
- **Rapport PDF accessible depuis le menu**
- **Navigation intuitive** et retour au menu avec ESC
- **Compatibilité Linux / Windows / Mac**

## 9 Difficultés rencontrées

- Synchronisation entre clics souris et logique du tour
- Intégration visuelle propre des dropdowns
- Positionnement et rafraîchissement de la carte
- Appels croissants entre les fichiers sans duplication

## 10 Conclusion

Ce projet nous a permis d'appliquer nos compétences en **Python orienté objet**, en **Pygame**, en écriture d'IA et en conception graphique. L'architecture du projet est claire, modulaire et facilement maintenable.