

---

Projet: ARIMAA

## Objectif

Le but de ce projet est de développer une Intelligence Artificielle (IA) pour le jeu Arimaa. Arimaa est un jeu de plateau proche du jeu d'échecs (jouable sur le même plateau), dont les règles sont simples, mais dont la nature le rend très intéressant pour le développement d'une IA. Vous serez chargé de:

- Implémenter le jeu
- Créer une petite interface (plateau de jeu) pour pouvoir jouer
- Implémenter une IA pour jouer contre un humain

Le projet se finira sur une soutenance durant laquelle vous devrez nous exposer le code ainsi qu'un rapport (détails à la fin du document).

## Règles du jeu

- Règles officielles : <http://arimaa.com/arimaa/learn/rulesIntro.html>
- Tutoriel pour mieux comprendre : <http://arimaa.com/arimaa/learn/flash/jc/new/>
- Jouer au jeu : <http://arimaa.com/arimaa/>

Si les règles officielles du jeu peuvent paraître complexes, le tutoriel et le jeu en ligne permettent de les comprendre très vite.

## Soutenance

La soutenance se fera avec 4 groupes en même temps et durera 30 minutes (7-8 minutes par groupe). Durant celle-ci chaque groupe devra :

- Rendre son rapport imprimé
- Présenter son travail au chargé de TP et aux autres groupes
- Et faire une démonstration de son programme

A la suite de la présentation des questions pourront être posées. La soutenance aura lieu le jour de la dernière séance de TP comme indiqué sur le planning (groupes A et B mélangés).

## Rapport

Le rapport devra :

- Se composer de 2/3 pages (Max) hors page de présentation
- être rendu en version papier le jour de la soutenance
- ne pas contenir de code
- présenter votre réalisation ainsi que votre IA
- Répondre aux questions du type: difficultés rencontrées, points positifs et négatifs, choix dans l'implémentation des structures/variables/fonctions, ce que vous avez appris, améliorations si vous aviez eu plus de temps...

## Pour aller plus loin !

Le projet n'ayant pas pu être installé sur les machines de l'UTC, nous mettons à la disposition de ceux pouvant développer sur leur propre machine, une interface graphique et interactive en JavaScript. Depuis cette-ci, vous avez la possibilité de brancher votre IA et de jouer contre elle sur une jolie interface.

**Il est important de noter que les étudiants ne pouvant pas travailler sur leur machine et n'ayant pas accès à cette interface ne seront pas pénalisés. En effet, le projet sera noté sur une appréciation général du travail et de la compréhension de Prolog.**

### Mise en place du projet

Le squelette du projet est hébergé sur GitHub, il vous suffit de le télécharger et de suivre les instructions du README.

GitHub : <https://github.com/rlacazel/Prolog-Arimaa>

L'interface graphique est fournie, il suffit de remplir la fonction `get_moves` par le meilleur coup en fonction de l'état du jeu et des pions sur le plateau.

### Détails techniques

La fonction à implémenter : `get_moves(Moves, gamestate, board)`. Votre programme sera appelé avec le `gamestate` et le `board`, et il vous faudra renvoyer la variable `Moves`. La variable `Moves` devra renvoyer jusqu'à 4 mouvements unitairement, même si c'est la même pièce qui bouge (voir règles du jeu pour comprendre).

### Structure

`Gamestate`: `[side, [captured pieces]]`

`Board`: `[[row,col,piece_type,side],[row,col,piece_type,side],...]`

`Moves` : `[[[old_row,old_col],[new_row,new_col]],...,...,...]`

### Exemple d'appel

```
get_moves(Moves, [silver, []],
[[0,0,rabbit,silver],[0,1,rabbit,silver],[0,2,horse,silver],[0,3,rabbit,silver],[0,4,elephant,silver],
[0,5,rabbit,silver],[0,6,rabbit,silver],[0,7,rabbit,silver],[1,0,camel,silver],[1,1,cat,silver],
[1,2,rabbit,silver],[1,3,dog,silver],[1,4,rabbit,silver],[1,5,horse,silver],[1,6,dog,silver],
[1,7,cat,silver],[2,7,rabbit,gold],[6,0,cat,gold],[6,1,horse,gold],[6,2,camel,gold],[6,3,elephant,gold],
[6,4,rabbit,gold],[6,5,dog,gold],[6,6,rabbit,gold],[7,0,rabbit,gold],[7,1,rabbit,gold],[7,2,rabbit,gold],
[7,3,cat,gold],[7,4,dog,gold],[7,5,rabbit,gold],[7,6,horse,gold],[7,7,rabbit,gold]]).
```

**Return:** `Moves = [[[1,0],[2,0]],[[0,0],[1,0]],[[0,1],[0,0]],[[0,0],[0,1]]]`