

# RAPPORT D'ANALYSE

## Introduction :

Le Risk est un jeu de société qui se joue de 2 à 6 joueurs, de type jeu de guerre compétitif. L'objectif final de ce projet est de proposer une modélisation de ce jeu à l'aide de la programmation orienté objet, en utilisant le langage Java pour développer et les diagrammes UML pour l'analyse du sujet.

## Mécanismes du jeu :

Le Risk est un jeu au tour par tour. Ici, nous allons commencer par considérer qu'il n'y a pas d'intelligence artificielle en guise de joueur (mais on a quand même implémenté un attribut qui permet de le spécifier). On veut modéliser une partie de Risk.

Le plateau de jeu est constitué de différents territoires, rassemblés en continents.

## Initialisation du jeu :

Les territoires sont répartis aléatoirement entre tous les joueurs. Ces derniers reçoivent un certain nombre de troupes (composées de pions) qu'ils peuvent utiliser pour renforcer leurs territoires.

## Lors d'un tour :

Au début d'un tour, le joueur reçoit des renforts proportionnellement au nombre de territoires qu'il possède, avec un bonus éventuel s'il contrôle un continent entier.

Le joueur peut, pour chaque pion, effectuer une attaque ou un déplacement, à la condition qu'il reste au moins un pion sur chaque territoire.

## Attaque :

Le joueur peut prendre jusqu'à 3 pions d'un territoire pour lancer une attaque sur un territoire ennemi, adjacent et accessible. Le résultat de l'attaque dépend d'un lancer de dés. L'attaquant jette autant de dés qu'il a de troupes engagées. Le défenseur, quant à lui, jette un dé s'il possède 1 ou 2 pion(s), et 2 sinon. On compare les valeurs du dé le plus fort de chaque participant. Celui qui a la valeur la plus faible perd un pion, et en cas d'égalité, l'attaquant perd un pion.

Si le territoire défendu n'a plus de pions, l'attaquant peut placer autant de pions qu'il veut dessus, puis attaquer à nouveau.

## Déplacement :

Chaque pion peut, par tour, se déplacer d'un seul territoire. Alors, il ne peut plus attaquer.

## Objectifs :

A l'aide des mécanismes et des règles du jeu précédemment décrites, l'on va implémenter plusieurs classes (Partie, Territoire, Continent, Pion, Joueur), ainsi qu'une classe Main.

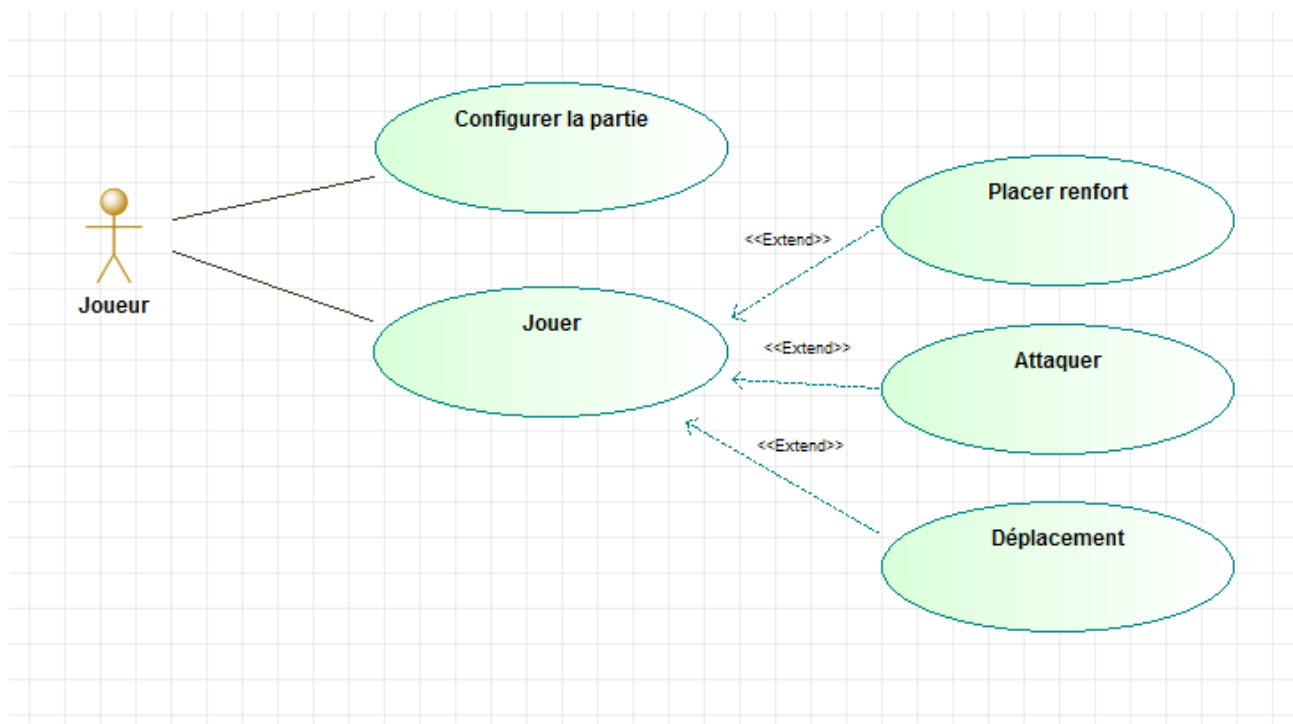
L'interaction entre ces différentes classes doit permettre de jouer une partie avec différents joueurs, qui peuvent tous bouger leurs pions sur différents territoires.

A la fin du projet, nous ambitionnons de créer une interface graphique simple.

Enfin, une fois que le code uniquement avec des joueurs « humains » fonctionne, nous allons implémenter la possibilité de jouer contre une intelligence artificielle.

Les différentes caractéristiques des classes vont permettre de jouer, par exemple la classe Pion doit posséder une méthode « déplacer », qui va permettre de gérer le déplacement du pion, d'un territoire vers un autre (adjacent et accessible).

Pour modéliser les possibilités d'utilisation de l'application nous avons réalisé un diagramme des cas d'utilisation.



*diagramme des cas d'utilisation*

Ainsi, le joueur peut configurer la partie, c'est-à-dire par exemple choisir s'il y a une intelligence artificielle (et son niveau de difficulté), la carte sur laquelle il va jouer ...

De plus, il peut jouer : ses actions peuvent être attaquer un territoire d'un adversaire dans le but de l'obtenir, déplacer de ses troupes dans le but d'attaquer plus tard ou de défendre une zone par exemple, et enfin il peut placer les renforts qu'il a reçu en début de tour de plateau.

Le déroulement d'une partie peut se modéliser à l'aide d'un diagramme d'activité.

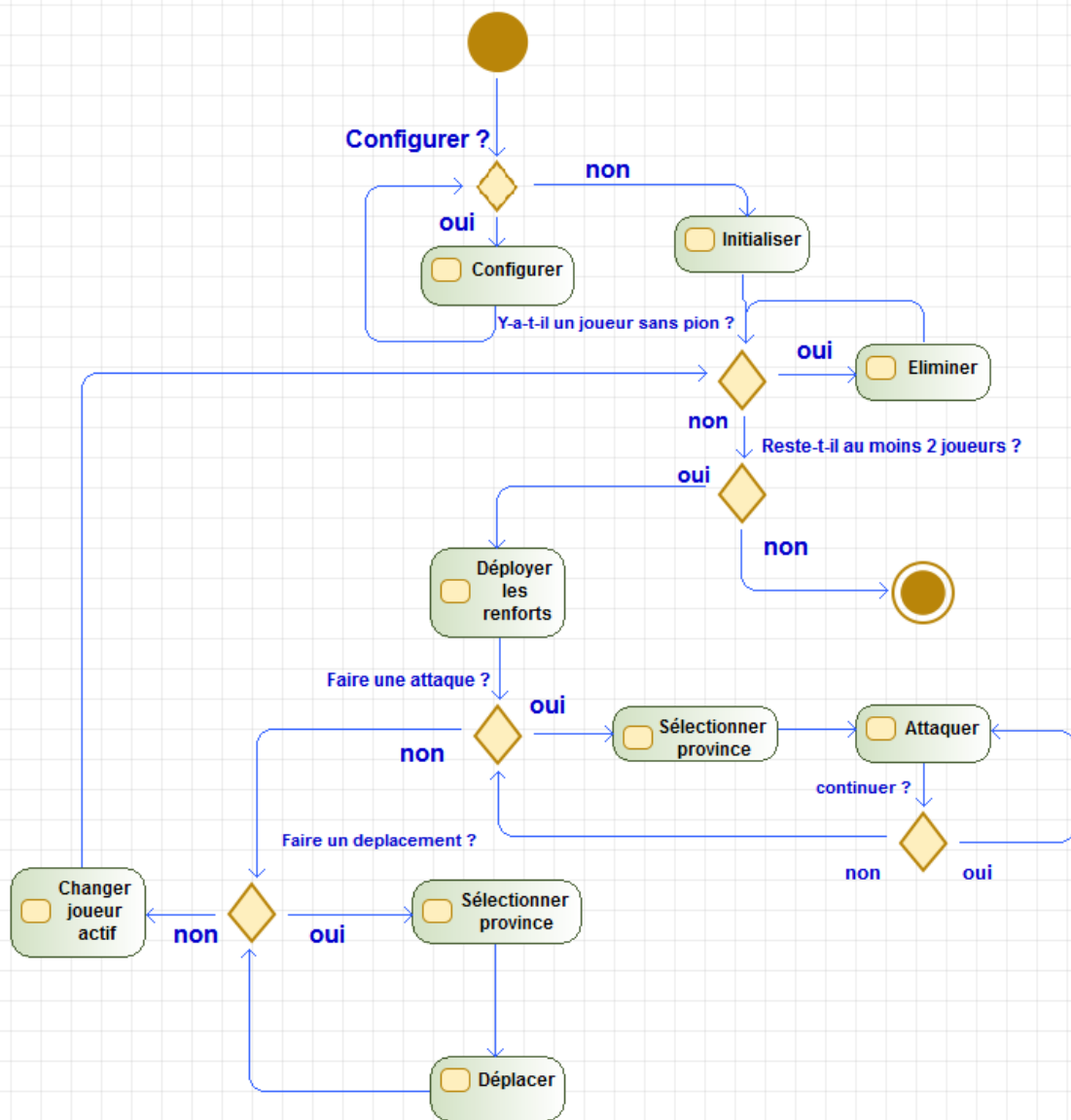


diagramme d'activité

Enfin, la modélisation en orienté objet peut être représentée par ce diagramme de classes.

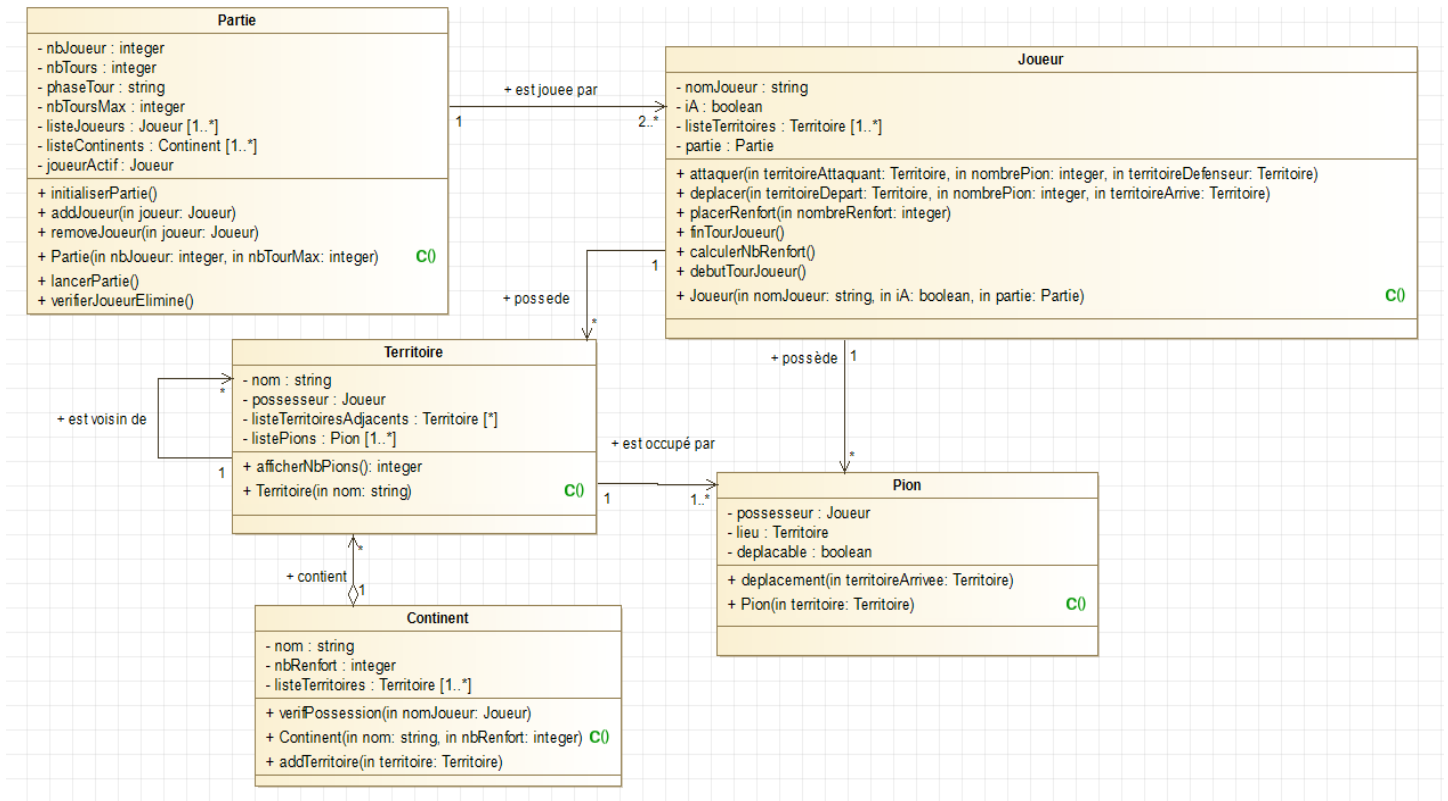


diagramme de classes

L'on a bien implémenté 5 classes qui représentent les différents objets du jeu. En effet, une partie peut être lancée entre plusieurs joueurs, qui vont déplacer des pions sur des territoires, eux-mêmes rassemblés en continents.

Afin de nous organiser, nous avons réalisé un diagramme de GANTT.

01/04 matin	01/04 après-midi	03/04 matin	03/04 après-midi	08/04 matin	08/04 après-midi	12/04 matin
Analyse du Sujet						
Rapport d'analyse				Finaliser rapport		
	Diagrammes UML					
		Squelette du programme	Coeur Du Programme			

08/04 après-midi	12/04 matin	12/04 après-midi	19/04 matin	19/04 après-midi	07/05 matin	09/05 matin
Coeur Du Programme						
		Interface Graphique				
					Ajout Cosmétique	
						Préparation oral

diagramme de GANTT