

## 1 Introduction

L'objectif de ce TD est de travailler sur la modélisation d'un jeu de l'oie (*game of the goose*) simplifié. Ce dernier est un jeu de plateau à information complète (les joueurs connaissent les possibilités des adversaires). Le jeu proposé est différent du véritable jeu de l'oie avec des cases non bloquantes.

## 2 Présentation du jeu

DEPART		?		?
ARRIVEE	↩	?	↪	

Le jeu (en version basique) comporte un plateau (*board*) de 10 cases (*spaces*) représenté la figure ci-dessous, des pions pour les joueurs (*players*) et d'un dé (*die*) à six faces. Au début de la partie les joueurs décident de l'ordre de passage, placent leur pion sur la case DÉPART, puis chacun à son tour,

1. le joueur (*player*) lance le dé, ensuite il avance son pion du nombre de cases indiqué par le dé, si le pion sort du plateau alors, il met son pion sur la case DÉPART.
2. si la case d'arrêt présente un symbole particulier, alors il exécute les instructions concernant celle-ci,

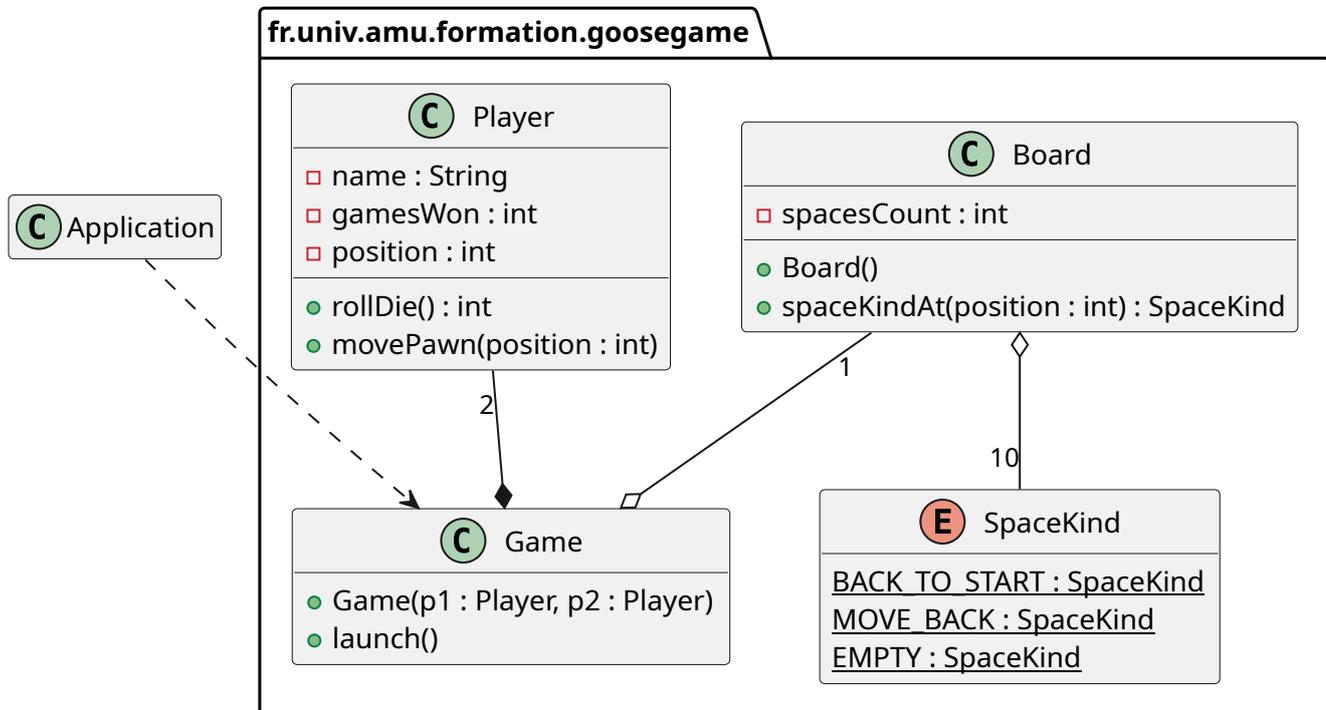
Case	Effet
↩	retour à la case départ
↪	recul d'une case
?	question (détails après)
Case vide	rien de particulier

Attention seul l'effet de la première case atteinte lors du tour d'un joueur s'applique (reculer sur une case avec un effet ne fait rien de plus). Le gagnant est le joueur arrivant le premier sur la case ARRIVÉE.

Remarque : il peut y avoir plusieurs joueurs (ou pions) sur la même case (contrairement au jeu de l'oie classique).

### 3 La modélisation actuelle

Une première modélisation UML a été proposée par une équipe dont voici le diagramme de classes. Cette modélisation utilise le plateau présenté au début et suppose que le jeu se joue uniquement avec deux joueurs.



Quelques explications sur le diagramme :

- Le `gamesWon` des joueurs représente le nombre de parties gagnées par ce joueur.
- Les joueurs sont construits par la classe `Game` (relation composition).

### 4 Critique de la modélisation

**Question 1 :** Quels retours ou critiques pouvons-nous formuler sur la modélisation actuelle ?  
Proposez des solutions pour améliorer l'organisation du code.

### 5 Limitation du nombre de joueurs

La limitation de 2 joueurs est dommage et on souhaiterait pouvoir créer des parties avec 3 joueurs ou plus.

**Question 2 :** Proposez une façon de lever la limitation pour avoir un nombre quelconque de joueurs. Comment gérer la saisie des joueurs ?

## 6 Nouveau type de case : question

Nous aimerions ajouter un nouveau type de case : la case *Question*. Cette case a l'effet suivant : si un joueur tombe dessus, il doit répondre à une question, s'il répond juste, il reste sur la case, sinon il recule de 3 cases. Les questions sont en clair dans un fichier texte qui serait chargé au lancement. Évidemment, les questions sont avec leur réponse dans le fichier.

**Question 3 :** Proposez une réorganisation du code permettant de gérer convenablement ce nouveau type de case.

## 7 Limite de la taille du plateau

Le plateau proposé en début de TD est petit. Nous aimerions pouvoir proposer aux joueurs

- de jouer avec le plateau original,
- ou bien d'utiliser un plateau de taille fixée par le joueur créant la partie avec des cases générées aléatoirement.

**Question 4 :** Comment faire pour proposer ce choix ? Donner une nouvelle modélisation permettant de gérer les deux possibilités de créations de plateau.

## 8 Sauvegarde de partie

On souhaite pouvoir sauvegarder une partie dans un fichier pour pouvoir ensuite la reprendre en chargeant le contenu du fichier.

**Question 5 :** Comment s'y prendre et quelles sont les modifications à faire dans la modélisation ? Comment peut organiser le code pour permettre de changer facilement le format de stockage de la partie ?