

1 Implémentation d'un jeu de l'oie

L'objectif de ce TD est de travailler sur la modélisation d'un jeu de l'oie 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.

1.1 Présentation du jeu

DEPART		?		?
	ARRIVEE	↙	?	↻

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

- le joueur 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 DEPART.
- si la case d'arrêt présente un symbole particulier, alors il exécute les instructions concernant celle-ci,

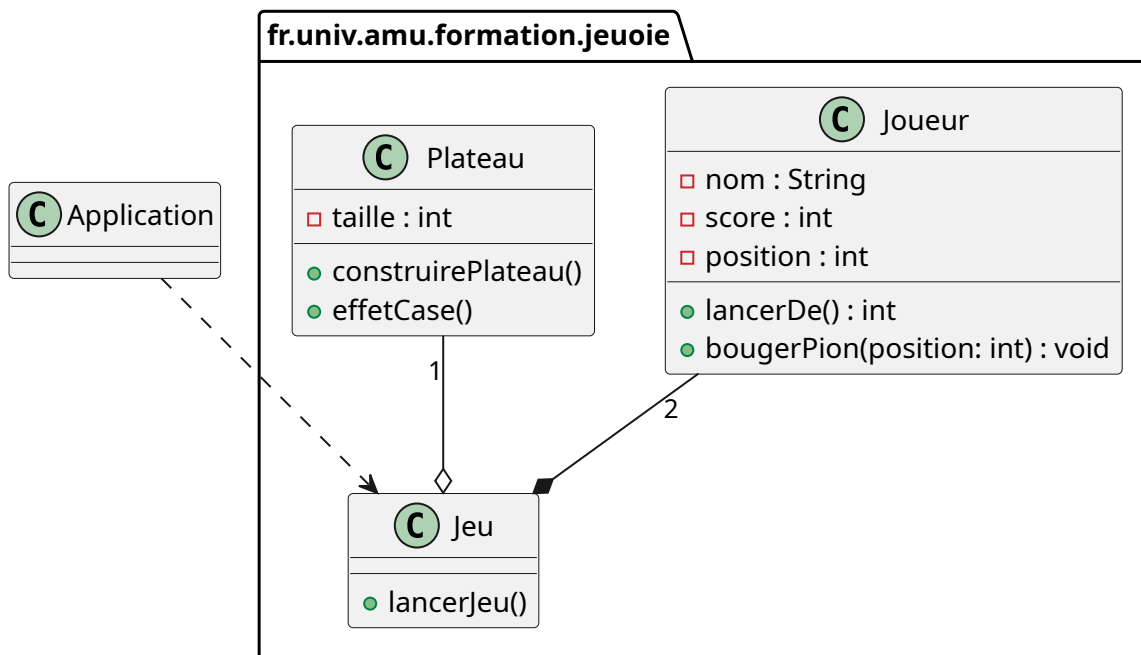
Case	Effet
↙	retour à la case départ
↻	recul d'une case
Case vide	rien de particulier

Attention les effets des cases n'est pas rétroactif (reculer sur une case avec un effet ne fait rien de plus). Le gagnant est le joueur arrivant le premier sur la case ARRIVEE.

Remarque : il peut y avoir plusieurs joueurs (ou pions) sur la même case.

1.2 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 seul deux joueurs sont présents.



Quelques explications sur le diagramme :

- Le score des joueurs représente le nombre de parties gagnées par ce joueur.
- Les constructeurs ne sont pas explicités.
- Les joueurs sont construits par la classe Jeu.

1.3 Extensions

La première modélisation étant un peu limitée dans les possibilités de jeu. Nous proposons plusieurs extensions ou améliorations :

1. La limitation de 2 joueurs est dommage. Proposer une façon de lever la limitation pour avoir un nombre quelconque de joueurs. Comment gérer la saisie des joueurs ?
2. 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.
3. Le plateau proposé en début de TD est petit. Nous aimerions pouvoir utiliser des plateaux de taille donnée avec des cases générées aléatoirement. Comment faire que le jeu reste jouable ?
4. Comment gérer proprement l'affichage du déroulement du jeu ? Quelles sont les limites d'une approche *simple* ?

2 Questions

Question 1 : Quels retours ou critiques pouvons-nous formuler sur la modélisation actuelle ?

Proposez des solutions pour améliorer.

Question 2 : Quels sont les effets de chaque extension sur la modélisation ? Vous justifier vos choix technologiques et présenter ainsi que justifier l'architecture choisie.

Question 3 : On souhaite pouvoir sauvegarder une partie. Comment s'y prendre et quelles sont les modifications à faire dans la modélisation ?