

12/05/2022

## Rapport Laboratoire 3 : Rivière



Alexandre Jaquier et Jonathan Friedli



## Introduction

Dans le cadre de ce laboratoire, nous devons modéliser un problème de traversée de rivière.

Une famille composée d'un père, d'une mère, de deux filles et de deux garçons est accompagnée d'un policier et d'un voleur menotté et doivent tous traverser une rivière à l'aide d'un bateau.

Ci-dessous les contraintes devant être satisfaites en tout temps sur les deux rives et sur le bateau :

- Il ne peut y avoir que deux personnes au maximum sur le bateau.
- Les enfants et le voleur ne peuvent pas piloter le bateau (mais peuvent y monter lorsqu'il est à quai).
- Le voleur ne peut pas rester en contact avec un membre de la famille si le policier n'est pas présent.
- Les fils ne peuvent rester seuls avec leur mère si le père n'est pas présent (surement à cause de violence domestique) et réciproquement, les filles ne peuvent rester seules avec leur père si la mère n'est pas présente.

Le but du laboratoire est donc de créer une application en C++ en mode console permettant à l'utilisateur d'introduire les commandes pour embarquer et débarquer des personnes et déplacer le bateau et autre afin de faire passer toutes les personnes de l'autre côté de la rivière.

## Choix de modélisation et d'implémentation

### Person

Afin de modéliser les différentes personnes présentes dans le problème, nous avons décidé de créer une classe **Person** afin de factoriser au maximum le comportement de ces dernières. Ensuite 3 classes vont hériter de **Person** : **DependantPerson**, **Driver** et **Thief**.

### Driver

Les deux parents et le policier sont instanciés en tant que **Driver** car c'est les 3 seules personnes pouvant conduire le bateau.

### DependantPerson

Cette classe nous permet de factoriser le fonctionnement des enfants, surtout au niveau de la présence/absence de leurs parents. De cette classe, héritent les classes **Boy** et **Girl**. Ces dernières sont utiles afin de personnaliser le message d'erreur (Fille avec son père mais sans sa mère ou Garçon sans son père mais avec sa mère).

### Thief

Nous avons choisi de faire une classe séparée pour le voleur car ce dernier ne dépend que d'une seule autre personne (le policier) et non de deux comme **Boy** et **Girl**. Nous aurions pu faire un héritage plus complexe afin de plus factoriser les paramètres des personnes dépendant d'autres mais cela aurait trop complexifié le problème.

### Container

Comme indiqué dans la consigne, nous avons créé une classe **Container** dont héritent les classes **Boat** et **Banks**. Cela nous permet de factoriser le fonctionnement de ces containers. Afin d'itérer sur la liste des personnes et de ne pas la mettre à disposition nous avons créé des méthodes **begin()** et **end()** renvoyant des itérateurs constants sur la liste.

### Controller

Cette classe permet de gérer toute la logique du jeu. Elle va créer tous les composants du jeu lors de sa création, à savoir 8 personnes, 2 rives et un bateau. **Controller** va aussi s'occuper de la fluidité du jeu en tuant dans l'œuf toute tentative d'appel illégale aux composantes du jeu, afin de les empêcher de lancer une exception à la moindre entrée utilisateur invalide.

Elle s'occupe aussi de l'affichage, il aurait été mieux de confier ce rôle à une classe spécifique mais ne voulant pas complexifier plus le projet, nous avons décidé de ne pas le faire.

## Tests effectués

Nous avons effectué divers tests afin de vérifier que toutes les règles sont respectées :

Test	Résultat attendu
Il y a au maximum 2 personnes dans le bateau. Si nous tentons d'embarquer une 3 <sup>ème</sup> personne, un message d'erreur s'affiche.	OK
Le bateau peut bouger uniquement si le père, la mère ou le policier est dedans.	OK
Si la mère laisse une des filles avec le père, un message d'erreur s'affiche.	OK
Si le père laisse un des fils avec la mère, un message d'erreur s'affiche.	OK
Si le voleur n'est pas tout seul en présence du policier, un message d'erreur s'affiche.	OK
Il est impossible d'embarquer une personne qui ne serait pas sur la même rive que le bateau.	OK
Si l'utilisateur entre une input ne correspondant à aucune des commandes prédéfinies, un message d'erreur s'affiche.	OK
Si l'utilisateur entre l'une des input suivante (e <nom> ou d <nom>) mais donne un nom inconnu. Cela résultera en un input invalide	OK
Le compteur de tour s'incrémente si le joueur bouge un personnage ou bouge le bateau. Il ne s'incrémente pas s'il y a une erreur (laisser le voleur sans le policier)	OK
Le compteur de tour se remet à zéro et toutes les personnes retournent sur la rive gauche si on reset le jeu.	OK
Si toutes les personnes sont sur la rive droite, l'utilisateur à gagné et un message s'affiche	OK