Rivière – Labo 3

Durée du travail :	**Du 18.04.2024 au 22.05.2024**
Auteurs :	**Demont Killian** **Graf Calvin**
Enseignant :	**Krähenbühl Grégoire**
Assistant :	**Decorvet Grégoire**
Domaine d'application :	**C++**
Lieu de travail :	**HEIG-VD | Yverdon-les-Bains**

Table des matières

1. Introduction

Ce laboratoire a pour objectif de concevoir un jeu de logique couramment appelée « La traversée de la rivière ». L'objectif est d'implémenter deux rives séparées par une rivière qui peut être traversée à l'aide d'un bateau. Il y a 6 personnages différents d'un côté de la rive (un père, une mère, deux fils, deux filles, un policier et un voleur) et le but est de tous les emmener sur la rive du côté opposé en respectant les règles établis (voir cahier des charges) pour chacun des personnages. Le bateau peut contenir 2 personnes par traversée et doit obligatoirement avoir un conducteur.

Nous devrons également gérer l'affichage des deux rives, du bateau et des personnages qui se met à jour en fonction des choix de l'utilisateur. Un menu permettant de réinitialiser le jeu, quitter ou faire les actions nécessaires au jeu doit être mis en place.

2. Développement

2.1 Instruction de compilation

Nom du compilateur : Mingw-w64 gcc

Version du compilateur : 13.2.0

2.2 Choix d'implémentation

Nous avons décidé d'intégrer les règles directement dans la classe de chacun des personnages. Comme les règles leurs sont directement lié, il est peu probable que nous réutilisions les mêmes règles dans le futur pour un nouveau personnage. De plus, cela nous permet de limiter le nombre de classes créer et de simplifier le programme.

Nous avons créé une classe Validation qui est utilisée par les personnes afin de vérifier si elles peuvent ou non se trouver dans un Container. Cette classe permet d'associer un message d'erreur à un état. Il est aurait été possible de simplement ajouter un méthode getErrorMessage() dans nos classes retournant un message d'erreur mais dans ce cas le message est séparé de l'état (il n'y a pas de réel lien entre les deux) et si l'on décide d'ajouter une condition supplémentaire à nos personnes alors ça forcerait d'avoir un message d'erreur long et peu clair précisant tous les cas.

Nous créons nos personnes dans le main car nous estimons que le Controller doit gérer le jeu et sa logique mais qu'il doit pouvoir être possible de jouer avec un ensemble différent de personnes. Une personne tierce souhaitant utiliser notre programme pourra alors uniquement modifier le main. Cela évite qu'il touche au Controller, où il pourrait malencontreusement modifier des parties du code qui rendrait le programme inutilisable.

2.3 Protocole de tests

Test	Résultat
Si l'utilisateur tape 'p' l'état actuel du jeu est affiché	OK
Si l'utilisateur tape 'e <nom>' la personne monte dans le bateau si les règles sont respectées	OK
Si l'utilisateur tape 'd <nom>' la personne descend du bateau si les règles sont respectées	OK
Si l'utilisateur tape 'm' le bateau se déplace s'il y a un conducteur	OK
Si l'utilisateur tape 'r' le jeu se remet à son point d'origine	OK
Si l'utilisateur tape 'q' le logiciel s'arrête	OK
Si l'utilisateur tape 'h' le menu est affichée	OK
S'il n'y pas de conducteur sur le bateau et que l'utilisateur souhaite le déplacer → message d'erreur	OK
Si la personne n'est pas sur la rive / le bateau et que l'utilisateur lui demande d'embarquer / de débarquer → message d'erreur	OK
Si l'utilisateur tape 'e' ou 'd' sans nom → message d'erreur	OK
Si le bateau est plein et qu'une personne essaie d'embarquer → message d'erreur	OK
Si le policier essaie de laisser seul le voleur avec d'autres personnes → message d'erreur	OK
Si le père essaie de laisser un de ses fils seul avec sa mère → message d'erreur	OK
Si la mère essaie de laisser une de ses filles seule avec son père → message d'erreur	OK
Si l'utilisateur entre un message vide ou un message qui ne correspond à aucune commande → message d'erreur	OK
Le nombre de tour réalisé ainsi que les rives et le bateau sont affichée à chaque tour	OK
Le nombre de tour augmente uniquement en cas d'embarquement, débarquement ou déplacement du bateau (réussi ou non)	OK
Si tous les personnages sont sur la rive de droite, le jeu s'arrête et un message de victoire s'affiche	OK

3. Conclusion

Tous les personnages ont pu être implémenté, le jeu fonctionne et le menu également. Toutes les règles ont été mis en place et l'affichage est respectée. La conception des classes a été factorisé le plus possible afin de faciliter l'implémentation d'éventuelles nouveaux personnages avec leurs propres règles dans le futur.

Tous les points du cahier des charges ont été remplis et testés avec succès.

4. Diagramme de classe