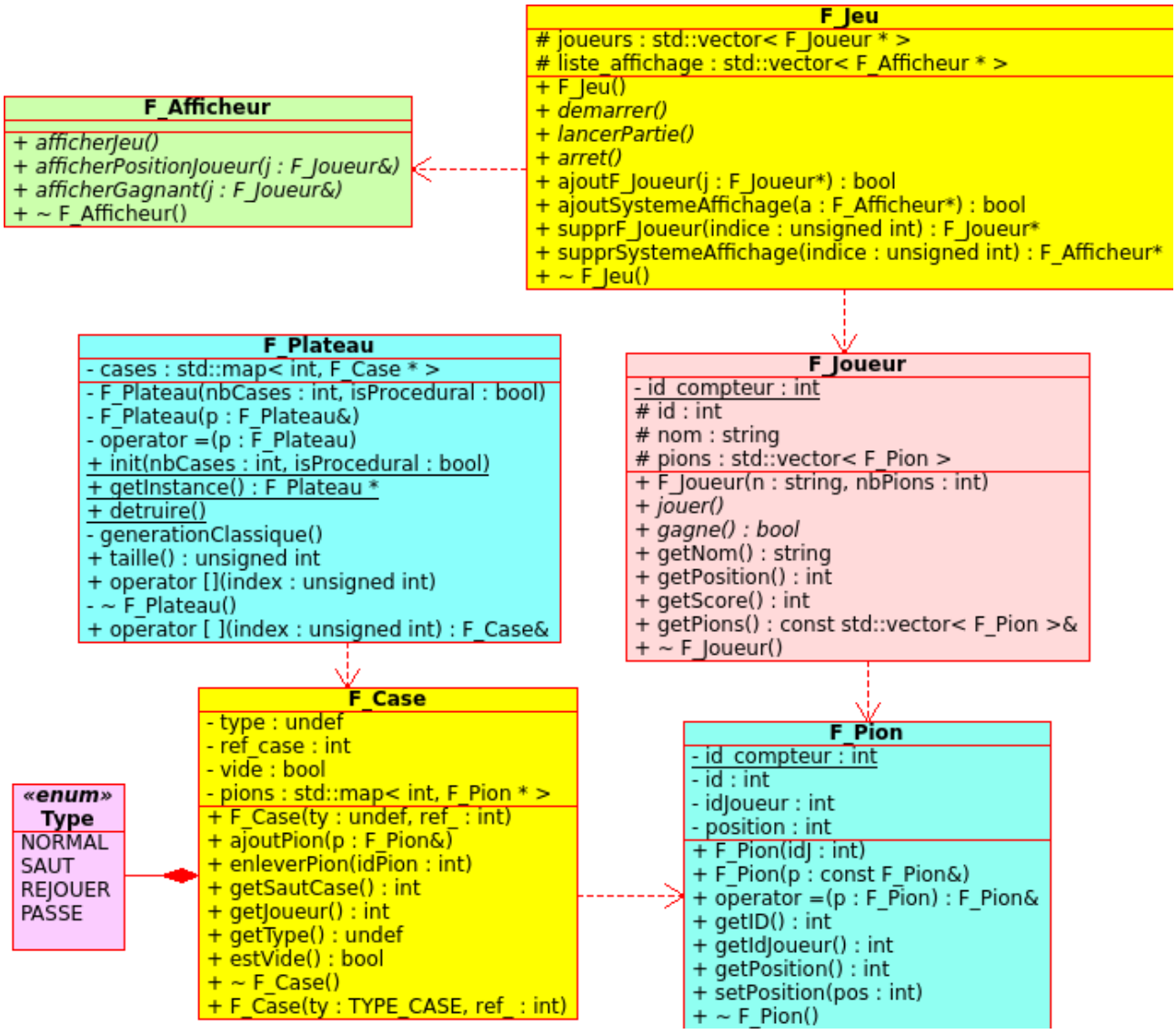
Rapport de projet C++
Master 1 Informatique

Framework de jeu
Auteurs : Célia SAIDANI & Luxon JEAN-PIERRE

Sommaire

I Architecture du projet

1 Architecture du framework



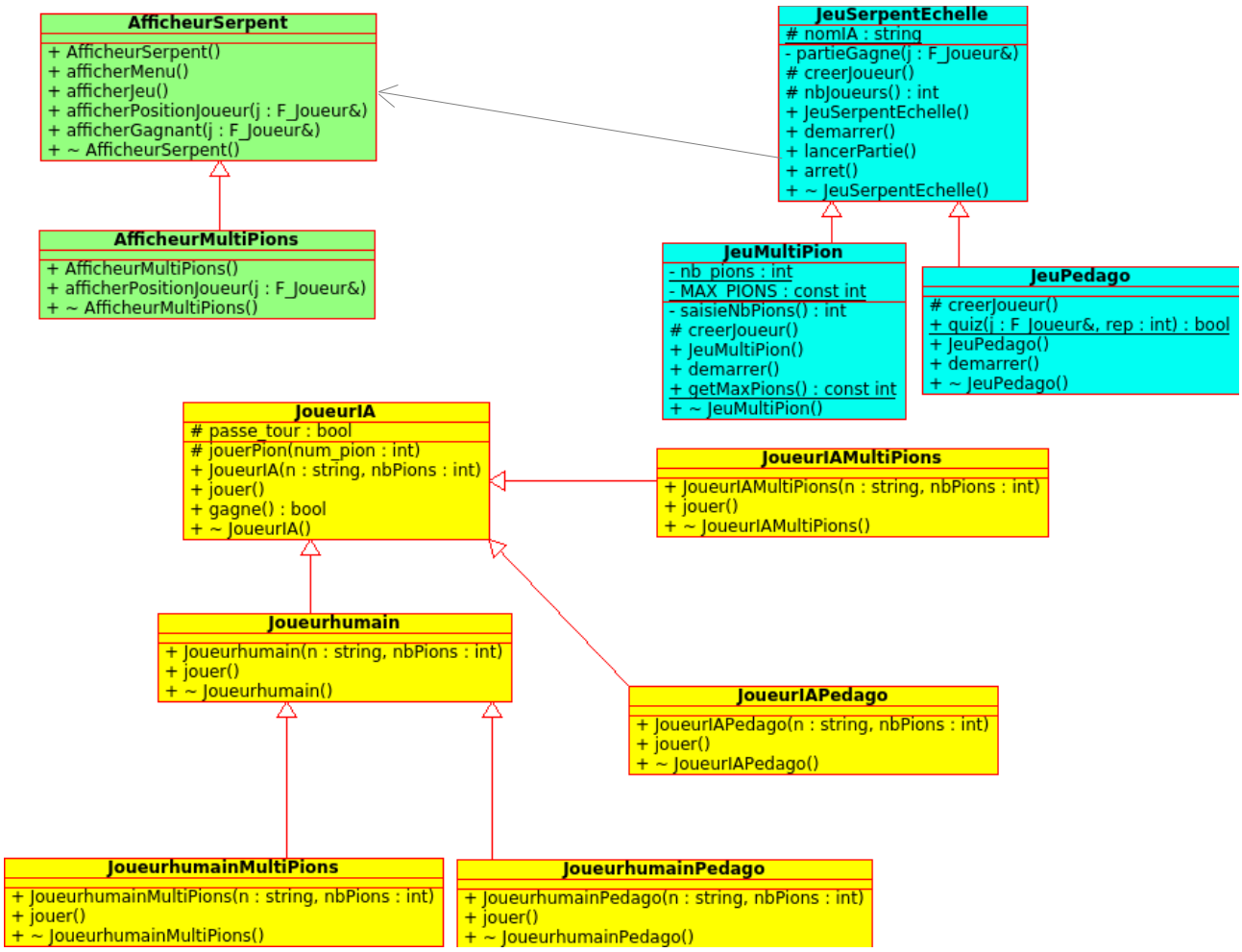
Le but recherché dans cette architecture est de séparer la structure responsable du jeu des éléments utilisés (case, plateau, joueur). Ici, La classe *F_Jeu* va interagir avec les joueurs et les systèmes d'affichages, tandis que *F_Joueur* interagira avec les *F_Pion*. En ce qui concerne l'affichage, *F_Afficheur* est une classe abstraite pure dans la mesure où les différents jeux n 'auront pas exactement le même type d'affichage.

Le plateau fourni par le framework prend la forme d'un singleton. En effet, cette structure sera partagée entre les joueurs, qui auront besoin de déplacer le/les pions(s), et le système d'affichage pourra récupérer les informations relatives a ce même plateau.

Chaque case du plateau se définit par son type, et, le cas échéant, l'indice de la case pointé par celle-ci sur le plateau. Cette indice (*ref_case*) n'est pas défini si la case n'est pas une échelle ou un serpent.

Au niveau de son utilisation, le framework s'utilise sous la forme d'un fichier bibliothèque (*libDiderot.a*) qui sera lié au code source du jeu.

2 Architecture du jeu



L'architecture du jeu a été conçu de manière à ce que le jeu, ainsi que ses variantes, soient intégrés dans un seul fichier exécutable de manière à minimiser les redondances lors de l'utilisation du framework. La sélection des jeux se fait à travers les paramètres donnés lors de l'exécution du programme.

JeuSerpentEchelle, AfficheurSerpent et *JoueurIA* héritent respectivement de *F_Jeu, F_Afficheur* et *F_Joueur. JeuSerpentEchelle* est le jeu de base sur lequel sont basés les deux variantes développées, en l'occurrence *JeuPedago* et *JeuMultiPion*.

On distingue bien les joueurs humains (*Joueurhumain*) des joueurs ordinateur (*JoueurIA*). Cependant, de part le comportement de ces deux types de joueur, on peut considérer les joueurs humains comme étant des « joueurs IA » dans la mesure où le comportement est identique, sauf du point de vue humain où les actions sont manuelles (lancé de dé, sélection d'un pion pour le déplacement). De plus, pour chaque jeu, un type de joueur lui est associé. En effet, de part les différences entre les jeux, il fallait faire en sorte que les joueurs puissent effectuer un éventail d'actions permises par chaque jeu.

II Jeux réalisées

Trois jeux ont été réalisés dans ce projet :

- Le jeu de base (avec les cases pour rejouer et passer son tour)
- Le jeu à variante pédagogique (valider le déplacement si la réponse à la question est correct)
- Le jeu multi-pions (chaque joueur peut avoir jusqu'à 16 pion dans le jeu)

III Bugs et limites du framework

1 Limite

Le framework est stable et ne souffre d'aucun bug grave. En terme de fonctionnalité, il fournit l'ensemble des fonctions et classes nécessaires pour réaliser une jeu du même type que le jeu du serpent et des échelles (jeu de l'Oie, ...).

Cependant, l'un des problèmes liés à son architecture concerne les joueurs ordinateurs. En effet, pour la variantes multi-pions, il y a plusieurs manières de choisir un pions à déplacer dans le jeu. Dans la version actuelle, le joueur ordinateur se contente juste de faire avancer en priorité le pion qui est le plus loin de l'arrivée. Il aurait été plus souhaitable de faire en sorte que le joueur ordinateur puisse sélectionner une stratégie de jeux afin de pouvoir choisir un pion aussi bien en fonction de l'objectif visé, que de la situation du jeu à un instant *t*, sans devoir appliquer un algorithme écrit en dur.

2 Pistes d'extensions

Une possibilité d'extension qui a été pensé mais qui n'a pas pu voir le jour était la généralisation procédurale de plateau. A chaque création de plateau, on donne la possibilité au joueur de choisir entre le plateau classique, défini sur le lien fourni dans le sujet du projet, et un plateau généré de manière procédurale. Cette génération procédurale est soumise à plusieurs contraintes, qui sont les suivantes :

- Le plateau contiendra entre 6 et 10 échelles et serpents.
- Le plateau contiendra entre 4 et 10 cases *PASSE* et *REJOUER*.
- Une échelle doit toujours mener vers une case normal.
- Un serpent doit toujours mener vers une case normal.
- Deux échelles ne doivent pas mener vers une même case.
- Deux échelles/serpents ne doivent pas partir d'une même case.
- La case normale pointée par une échelle ne doit jamais être celle qui précède une case serpent.
- La case normale pointée par un serpent ne doit jamais être celle qui précède une case échelle.
- Une case serpent et une case échelle ne doivent pas être consécutives.
- Les cases 0 et 100 sont toujours normales.