

FRAMPLATEAULAND

1 Framework proposé

Le Framework est constitué de 3 entités :

- Une entité **CaseGeneric**

- Un plateau de jeux **FramPlateauLand** composé de CaseGeneric , offrant des fonctions propre aux jeux de plateau

- Une entité Affichage **Display** adaptable basé sur le design du 2048 permettant de jouer et d'afficher tous jeux basés sur un plateau du Framework

Chaque entité est faite pour être hérité et ainsi redéfinir certain mécanisme ou personnalisé certain comportement ou affichage

L'entité du **FramPlateauLand** fonctionne en parfaite transparence avec ces redéfinitions

CaseGeneric

Une simple classe abstraite qui permet de stocker une position et d'afficher le contenu hypothétique de la case.

FramPlateauLand

Une classe abstraite Template qui offre un étendard de fonction pour afficher, récupérer des valeurs sur l'entrée standard, modifier les valeurs du plateau

Offre un algorithme pour performer la chute de cases selon la gravité choisit à chaque tour par l'utilisateur

L'algorithme va faire chuter les cases qui ne sont pas fixe et pas vide, quand deux cases se touchent elles doivent indiquer si elles peuvent interagir ensemble, si oui elles vont à leur niveau d'abstraction interagir et renvoyer le résultat à l'algorithme qui va se contenter d'appliquer le changement dans le tableau.

Le plateau est un double vector de pointeur de CaseGeneric

Display

Une classe qui permet d'afficher un plateau de CaseGeneric grâce à SFML

Aucune attente active sur les événements

2 Respect du Sujet

2.1) Implémentation

L'implémentation d'un jeu **X** correspond à une classe **PlateauX** héritant de **FramPlateauLand**, une classe **CaseX** héritant de **CaseGeneric**

La définition d'une fonction de condition d'arrêt et une fonction pour Performer 1 tour de jeu.

Les 3 jeux sont implémentés dont notamment :

- Le Takin est générique : tant que le type définit l'opérateur de comparaison >
On peut ainsi jouer à un takin avec des Int, Char ...
- Le SOKOBAN permet d'importer des fichiers respectant le standard **xsb**
- Le 2048 normal
- Le 2048 Plugin qui est un héritage du 2048 normal et qui permet de jouer avec des case2048 personnalisés à l'infini, ainsi une nouvelle case est simplement un petit plugin à rajouter et le jeu n'a absolument pas besoin d'en connaître la logique interne.
- Un affichage dans le Terminal et un affichage en mode Fenêtre

2.2) Code :

- Entièrement Packager dans des sous-répertoire propre à chaque implémentation
- Code écrit en anglais, Nom de variable et fonction explicite
- Redéfinition des Constructeurs, définition des Visibilités
- MakeFile et Script, ressource UML et Doxygen
- Pattern Visitor pour le système Plug In
- Usage de vérification statique pour empêcher de jouer avec autre chose que des héritages de CaseGeneric

2.3) Difficultés rencontrées :

- Refactoring constant du code, pour parvenir à des Interfaces les plus claires et simples d'utilisation
- Contournement du problème **Multiple_dispatch** pour le système de Plug-In

2.4) Pistes d'extensions

Assurer la constCorrecness

Assurer plus de sécurité sur les Template avec les cast static

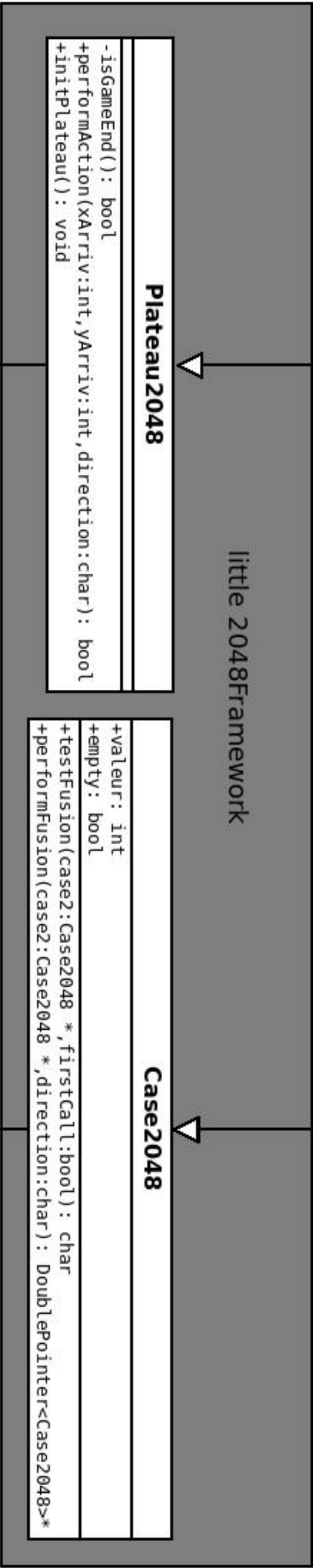
Vérifier les programmes à Valgrind et pousser la sécurité avec des smart Pointers

Case2048Operator
+oper: char

Case2048Negative

Case2048Destroy

Plateau2048PlugIn
+afterAction(newCase:Case2048 *): void



CaseTakin<T>
+valeur: T

casesOKOBAN
+valeur: char



PlateauTakin<T>
-AlreadySetBlankCase: bool
-BlankCaseAtTheEnd: bool
-isGameEnd(): bool
+performAction(xArriv:int,yArriv:int,direction:char): bool
+initPlateau(contenuJeu:vector<T>): void
+setBlank(i:int,j:int): void

PlateauSOKOBAN
-isGameEnd(): bool
+performAction(xArriv:int,yArriv:int,direction:char): bool
+initPlateau(contenuJeu:vector<char>): void
+readFile(nameFile:const char *): PlateauSokoban*



FramPlateauland<T>
+largeur: int
+longueur: int
#Plateau: vector<vector<T*> >plateau
#modeleJoueur: bool
#isJeuxPersonnage: bool
-modeleTerminal: bool
+isGameEnd(): bool
+performAction(xArriv:int=0,yArriv:int=0,direction:char=0)
+FramPlateauland(sizeX:int,sizeY:int,isJeuxPersonnage:bool)
+affiche(): void
+<<U>> applyInitPlateau(contenuJeu:U): void
+doAction(char:direction): void



CaseGeneric
+i: int
+j: int
+toString(): string
+Print(0:std::ostream&): void
+operator<<(0:std::ostream&,B:const CaseGeneric&): friend std::ostream&

Display<T>
+StartModewindow(): void
#initWindow(): void
-refreshWindow(): void
-quitOrGameOver(): void
-modeAuto(): void
-modeGamer(): void

FRAMEWORK