

Rapport du projet de POOIG.

Thomas Bignon - Hind Hamila

Binôme 4

Année 2018-2019

I. Architecture du projet.

Classes abstraites :

- Jeu
- Case
- Piece
- Plateau

Classes non-abstraites héritant de Jeu :

- Puzzle
- Domino
- DominoGommette
- Saboteur

Classes héritant de Case :

- CaseDomino
- CaseSaboteur
- CasePuzzle

Classes héritant de Piece :

- CarteSaboteur
- CarteChemin
- CarteAction
- PieceDomino
- PiecePuzzle

Classes héritant de Plateau :

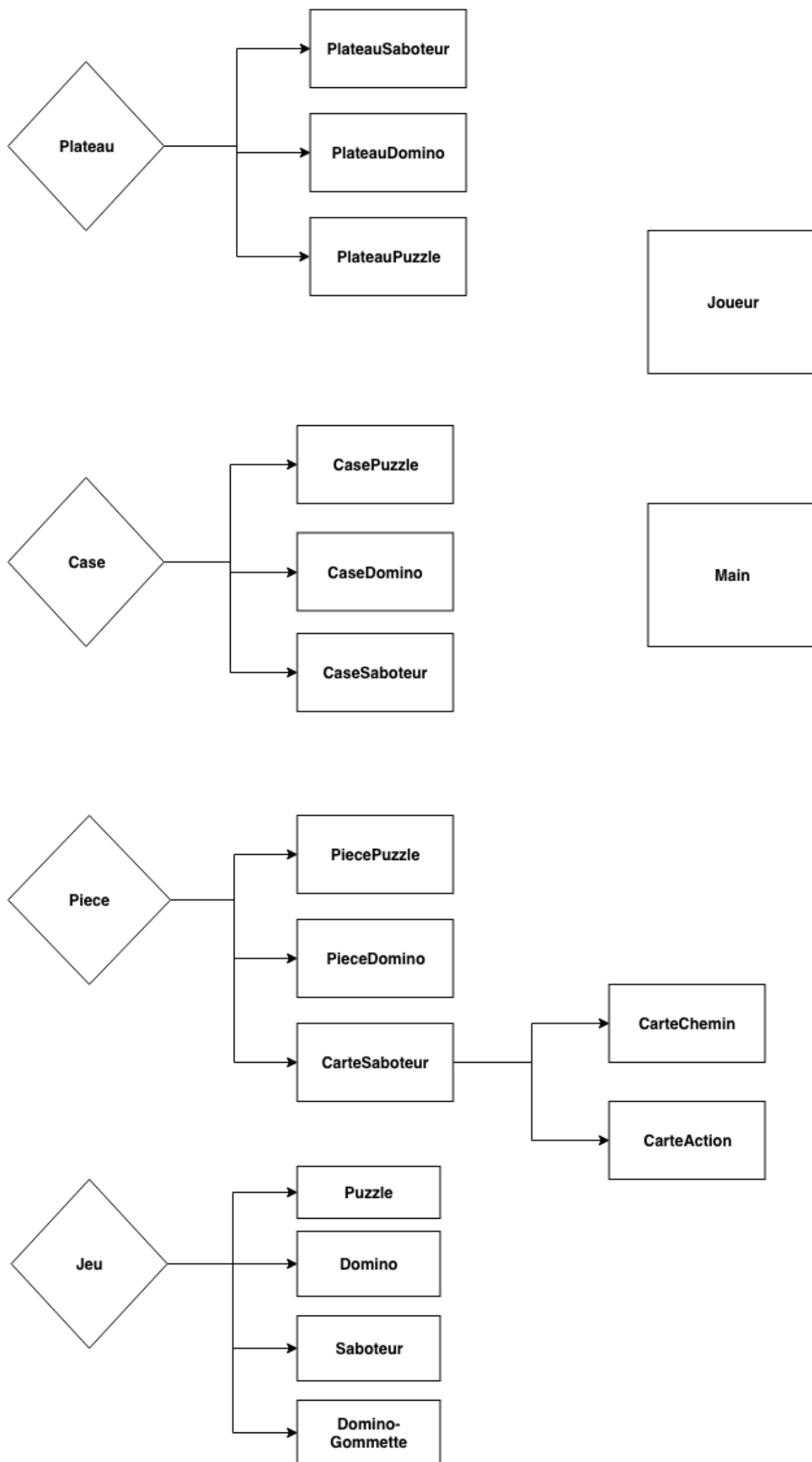
- PlateauDomino
- PlateauSaboteur
- PlateauPuzzle

Nous avons dans un premier temps tenté de généraliser les fonctions à utiliser pour les mettre dans des classes abstraites.

La classe Jeu par exemple contient les méthodes `lancerPartie()`, `joueUnTour()`, `setJoueur()` et `afficheIntro()`, tous les jeux doivent donc implémenter ces méthodes.

Elle possède également un attribut `Plateau` plateau ainsi qu'un attribut `Joueur []` participants qui est un tableau de joueurs.

Nous avons donc : `Main` qui lance un `Jeu` qui utilise forcément `Plateau`, `Case`, `Piece` et `Joueur`.



II. Parties Traitées

Nous avons commencé créer le code du jeu du Domino, celui-ci devait à la fois servir au domino et à sa variante, le domino-gommette.

Toutefois nous n'avons pas pu programmer le domino-gommette en interface graphique, faute de temps et de connaissances en interface graphiques. (En effet, nous pensons que les cours de POOIG au S3 ne sont pas suffisants pour pouvoir facilement créer une interface graphique propre). Mais la partie textuelle fonctionne pour les deux.

Le puzzle est totalement fonctionnel. Nous avons choisi de faire une variante de puzzle appelé "Taquin" ou pour résoudre le puzzle il faut coulisser les cases, le puzzle ayant une case vide. Pour la partie en interface graphique, le modèle a été séparé de la Vue (la classe interne `VuePuzzle` ainsi que la classe `JPanelCase`). La version textuelle du puzzle se fait donc sans image mais avec des chiffres avec plus ou moins le même comportement vis à vis du joueur.

Améliorations : on aurait pu rajouter des images supplémentaires pour plusieurs puzzle à résoudre à chaque fois.

Le saboteur n'a pas pu être complété, les bugs n'ont pas été fixés à temps, bien que le code était pratiquement complet.

L'affichage du plateau ne fonctionnant pas, il est difficile de jouer au jeu sans connaître les emplacements des cartes. Toutefois le dispositif de sabotage et de réparation de l'équipement des autres joueurs fonctionne, ainsi que toutes les autres mécaniques du jeu.

Nous aurions également pu créer une interface graphique au jeu, le rendant plus lisible et mieux jouable, ainsi qu'optimiser son code qui est, à notre avis, beaucoup trop long.