# Rapport C++

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SAMSON Pierre

GRAFF Nicolas

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Ce qui a été traité

* Nous avons traité les deux premiers jeux à savoir les **Dominos carrés** ainsi que **Trax :** les deux jeux sont complets et fonctionnent.
* Une interface graphique satisfaisant utilisant sfml.
* Nous avons fait au mieux pour utiliser de l’héritage là où nous le pensions utiles.

## Les aspects les plus significatifs

Comme dit précédemment, les deux premiers jeux fonctionnent bien avec un graphisme sommaire mais tout de même.

Pour l’implémentation du Garbage Collector, nous avons décidé de ne pas faire comme dans les TP notés. C’est-à-dire de ne pas l’appeler dans le constructeur mais après chaque New. Ce n’est pas la manière de faire la plus optimale mais ça nous permet de gérer les objets de SFML sans créer de nouvelles classes pour surcharger le constructeur.

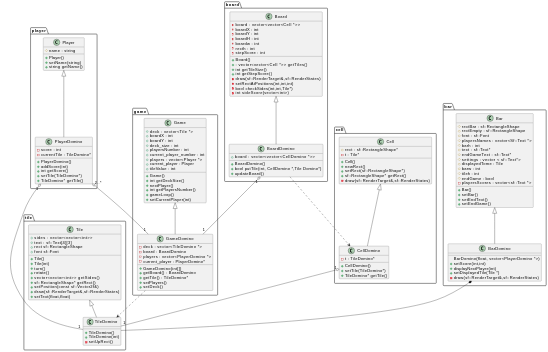
## Les quelques problèmes connus

* Nous savons que le contrat au niveau de l’héritage et de la réutilisation au maximum de code dans des jeux « conceptuellement proches » a été plus ou moins bien respecté…
* Nous avons trouvé que même si les jeux étaient conceptuellement proches il était assez difficile de réutiliser du code puisque les règles, les propriétés des tuiles ou encore les conditions de victoire sont très différentes d’un jeu à l’autre et il a été souvent nécessaire de réécrire du code.

## L’état des extensions que vous n’auriez pas eu le temps d’implémenter

Nous n’avons malheureusement pas eu le temps d’implémenter entièrement Carcassonne, néanmoins nous avons commencé à travailler dessus notamment le placement des tuiles est complètement fonctionnelle.

## Diagramme



On peut retrouver le diagramme de classe (volontairement incomplet) dans notre dépôt sous le nom de *class.plantuml* si vous voulez le consulter en meilleure qualité et en pouvant se déplacer et zoomer dans le diagramme.

Ici, il n’y a que les classes mères ainsi que les classes relatives au jeu de Domino, nous avons volontairement omis les classes relatives à Trax et à Carcassonne car les relations entre ces classes sont exactement les mêmes. N’utiliser que Domino ici permet d’avoir déjà une bonne idée de l’architecture du projet.