

**Rapport de projet Bibliothèque de développement multimédia**

Version du 19/03/19 - Spécifications

Royaume de la Programmation

Table des matières

[Spécifications 2](#_Toc3882827)

[Conception 2](#_Toc3882828)

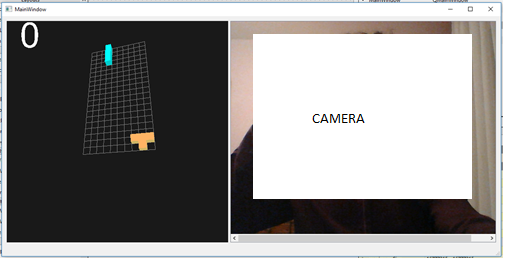
[Etat lors du rendu 2](#_Toc3882829)

# Spécifications

Le projet devra respecter le cahier des charges suivant :

* Affichage en perspective du plateau constitué d’une grille de 10×20 cellules représentant le plateau
* Apparition d’un tétriminos et descente de ce tétriminos en glissant sur le plateau incliné
* Déplacement latéral du tétriminos commandé par un geste de rotation des mains du joueur vers la droite ou vers la gauche
* Rotation du tétriminos commandé lorsque les poings du joueur sont frappés l’un contre l’autre
* Arrêt du tétriminos lorsqu’il arrive en bas et apparition du tétriminos suivant en haut
* Détection d’une ligne remplie et incrémentation des points
* Disparition des blocs associés à la ligne remplie et descente des blocs situés au-dessus
* Détection de la fin de la partie et réinitialisation pour la partie suivante.
* Descente rapide de du tétriminos commandée par un geste de déplacement des 2 poings vers le bas

Le design de l’application sera au format de l’image suivante.



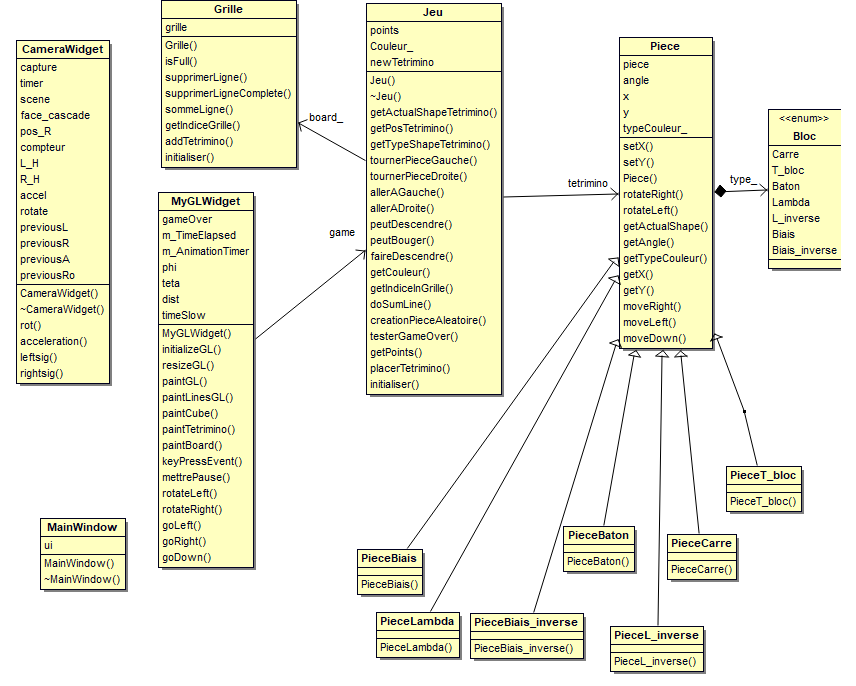
Le joueur devra effectuer les mouvements avec les mains pour réaliser les déplacements latéraux de la pièce.   
Lever le poing droit fera déplacer la pièce à droite. Lever le poing gauche fera déplacer la pièce à gauche.   
Rapprocher ses poings l’un de l’autre permettra la rotation de la pièce.  
Les faire descendre en bas de l’écran fera descendre la pièce d’un cran plus rapidement.  
Appuyer sur le bouton « n » en fin de partie permet de relancer une partie.

# Conception

Nous avons choisi de réaliser notre projet Tetris en nous inspirant du design pattern « Model View Controler »   
  
En effet, nous avons souhaité réaliser une classe se rapprochant d’un contrôleur « Jeu », une du modèle : MyGLwidget et une classe se rapprochant d’une vue la « MainWindow ». La différence avec ce modèle vient du fait que la « MainWindow » affiche aussi la caméra afin de pouvoir gérer les actions utilisateurs.

La classe grille permet de gérer les lignes lorsqu’elles sont complétées et de gérer le positionnement des pièces.  
La classe jeu gère l’ensemble des actions liées aux pièces et aux règles du jeu.  
La classe pièce dont hérite tous les types de pièces, permet d’obtenir les informations sur les positions de la pièce et de la déplacer.  
MyGLWidget gère l’affichage de toute la partie Tetris  
CameraWidget permet de faire l’affichage de la webcam et de récuperer les actions de l’utilisateur.

Les deux affichages sont réalisés dans dans le mainWindow.



# Etat lors du rendu

Les fonctionnalités demandées dans le cahier des charges sont toutes présentes. Il existe néanmoins des pistes d’amélioration pour certaines fonctions et pour l’affichage ainsi qu’il est possible d’implémenter de nouvelles fonctionnalités.

## Améliorations

Il est possible d’améliorer l’interface graphique en ajoutant un séparateur qui afficherait la pièce suivante au-dessus du cadre qui affiche le joueur. Cela permettrait au joueur de mieux préparer ses coups, bien que la dimension aléatoire des pièces (en version actuelle) soit aussi intéressante stratégiquement puisqu’elle fait travailler les réflexes du joueur ainsi que son adaptation.

Il est possible de réaliser des modifications sur la façon dont sont appelées les fonctions de rotation et déplacement des pièces de tetris. Actuellement, les fonctions sont appelées une unique fois lorsque l’action permettant de bouger la pièce a été effectuée. Il est ensuite nécessaire de revenir en position initiale pour refaire le mouvement pour la bouger à nouveau.  
Cela évite les déplacements trop rapides et/ou involontaires mais demande plus d’efforts au joueur.

## Ajout

Nous aussi pouvons améliorer la fonction de descente en permettant de descendre de plus d’un cran en accéléré. En accélérant la vitesse de descente de la pièce jusqu’en bas par exemple.

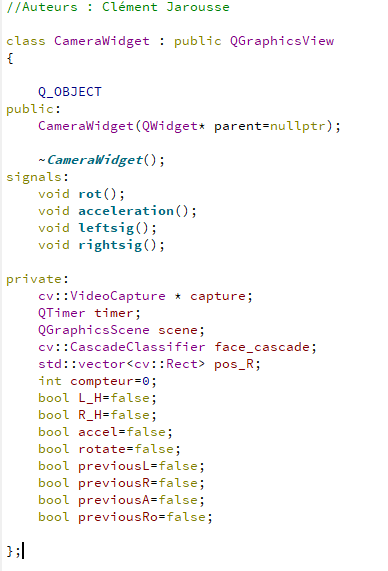
Nous pouvons aussi améliorer l’application en ajoutant un fichier de sauvegarde qui permettrait de conserver les meilleurs scores et pouvoir les afficher à chaque nouvelle partie.

## Bugs

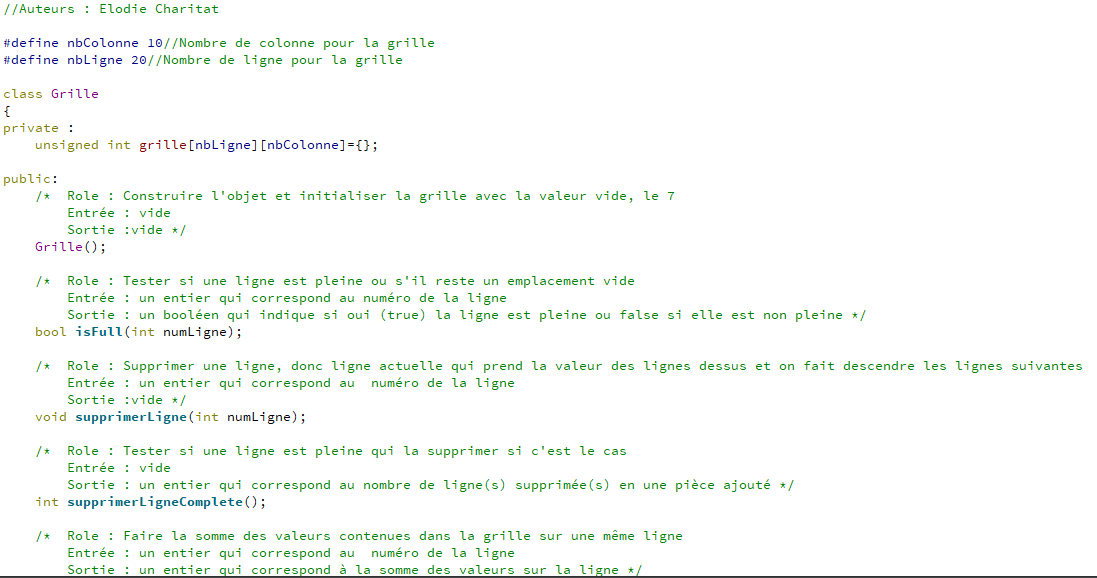
La détection des poings est peu efficace, elle a tendance à détecter les têtes/mentons des joueurs et parfois même des éléments de décor en arrière-plan. Il pourrait être intéressant de réaliser une amélioration des attributs et calculs permettant la détection des poings pour améliorer le confort du joueur tout en lui permettant de jouer plus efficacement en permettant certains mouvements (comme la descente rapide) qui peuvent être facilement perturbés par une fausse détection.

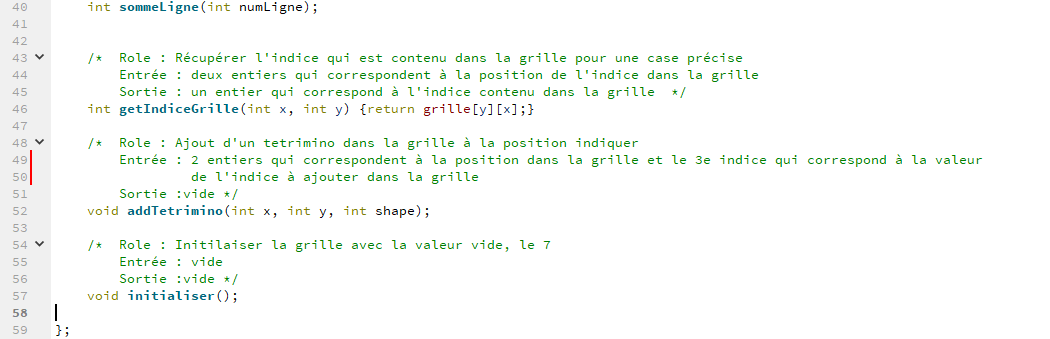
# Fichiers d’en-tête

## CameraWidget.h

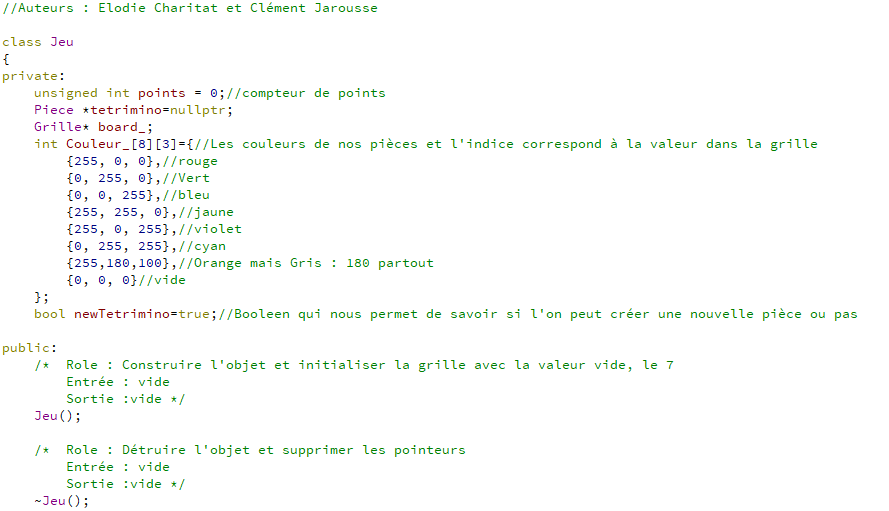


## Grille.h

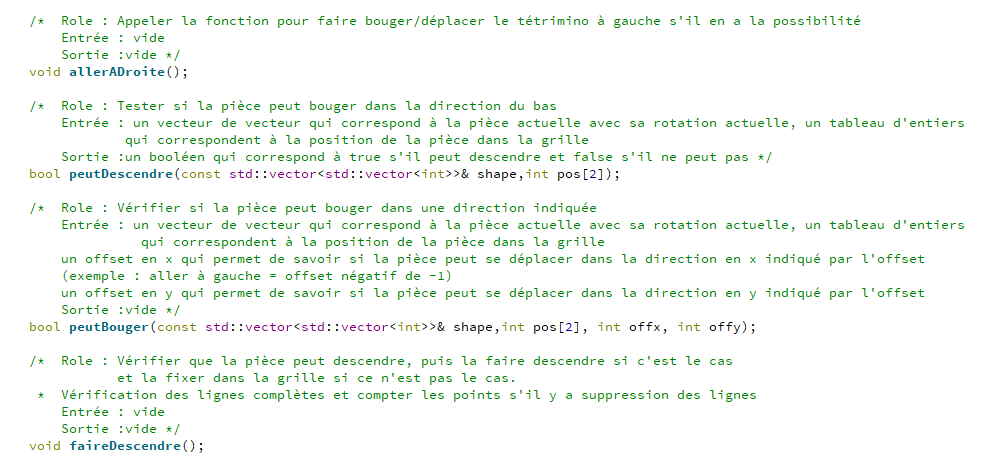


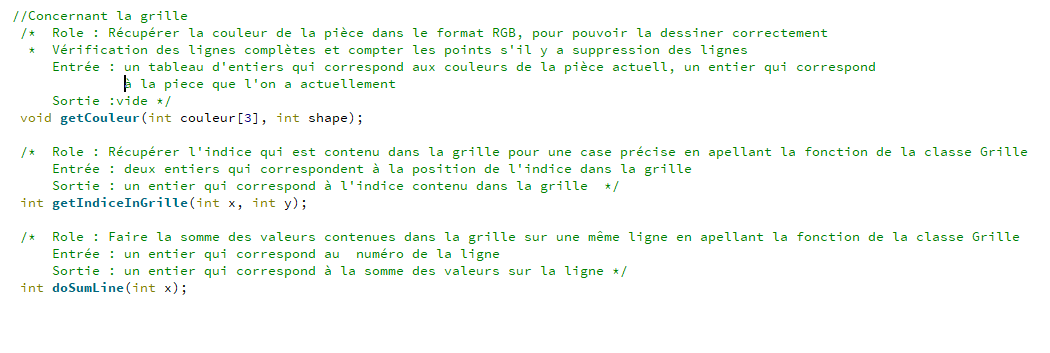


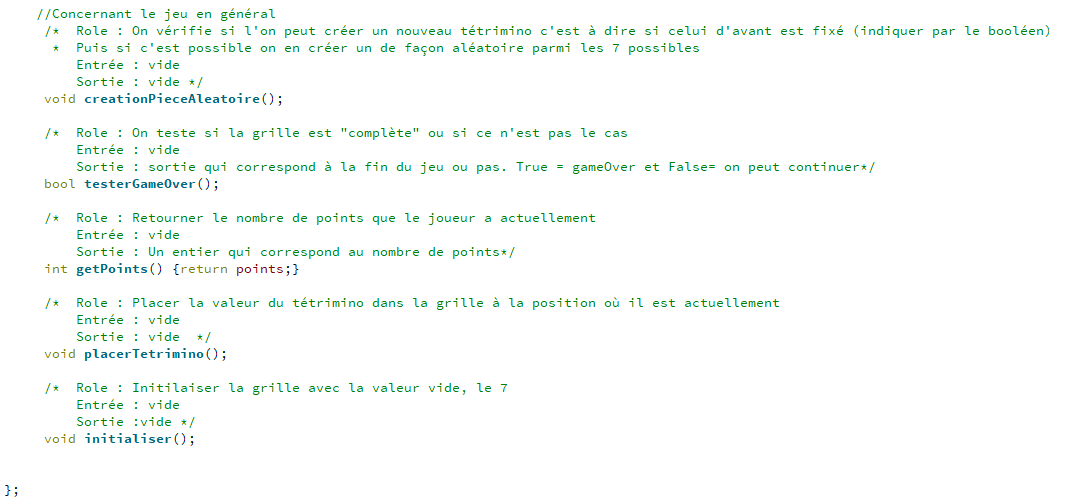
## Jeu.h



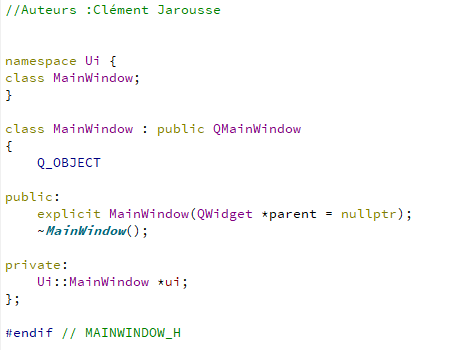




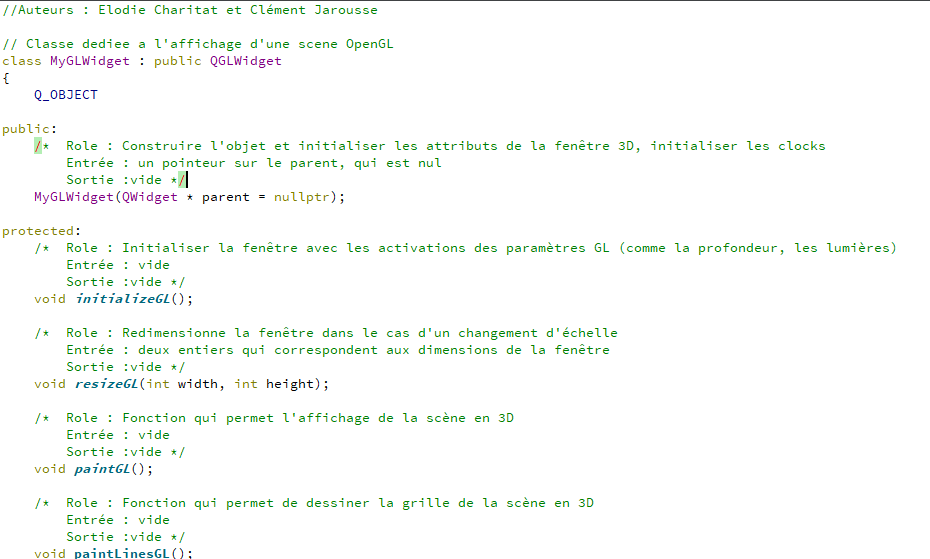


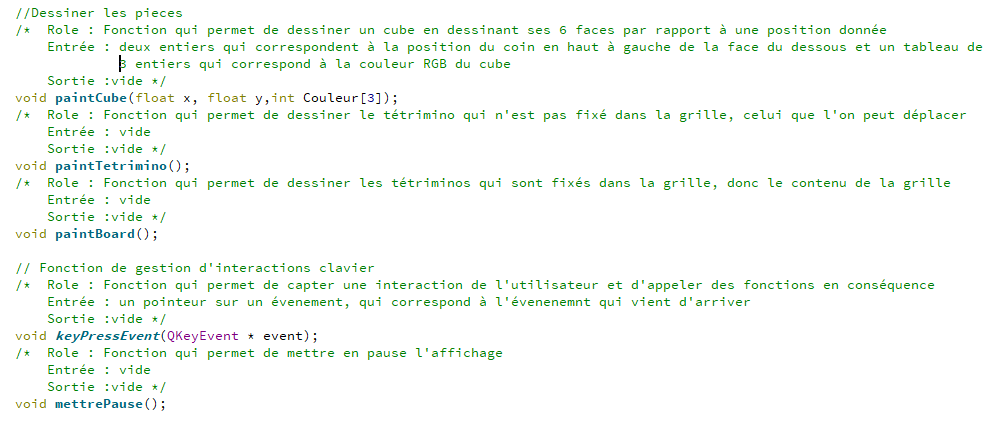


## Mainwindow.h



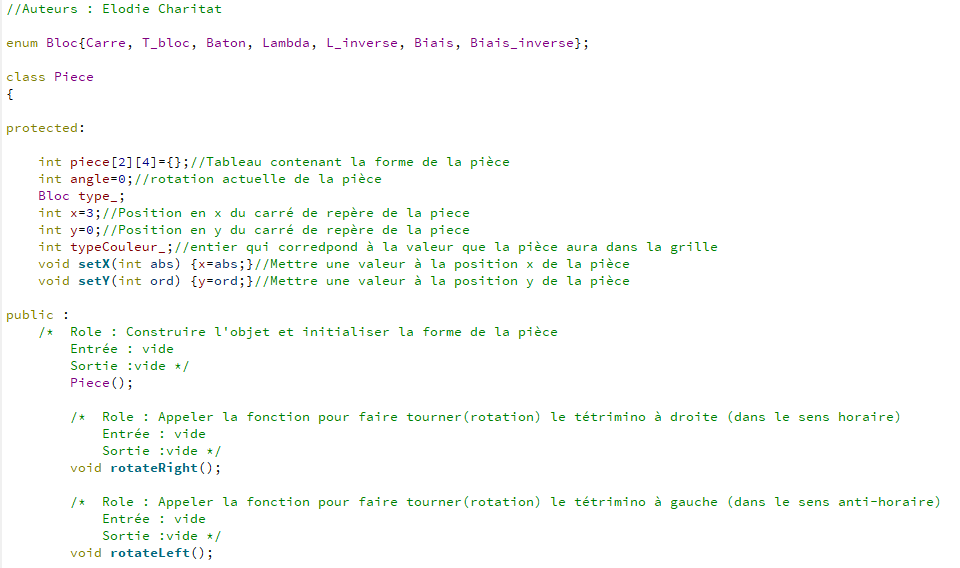
## MyGlWidget.h

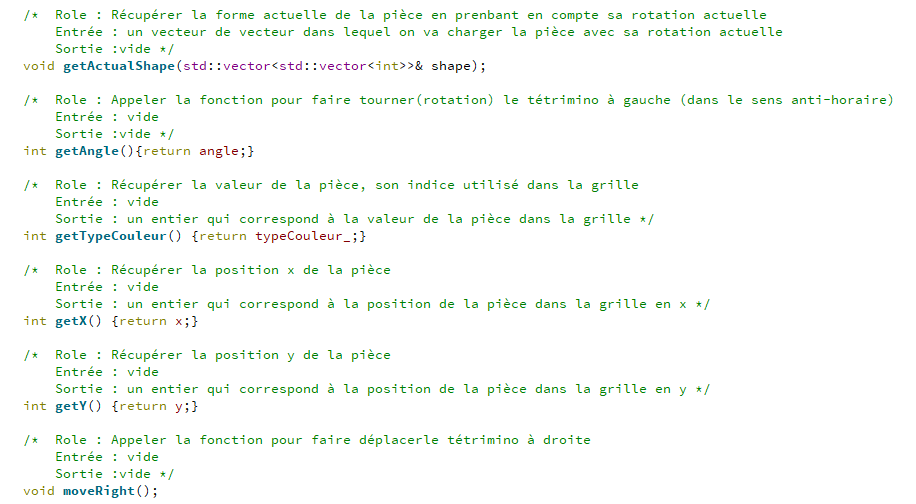


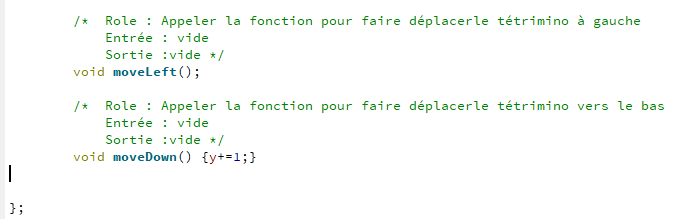




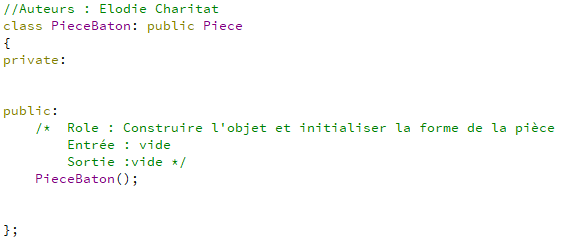
## Piece.h



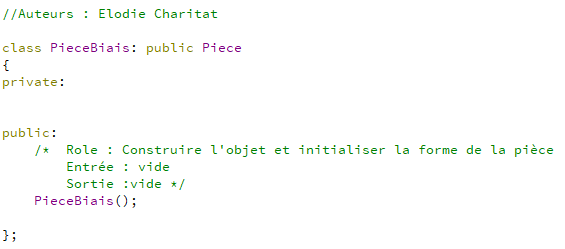




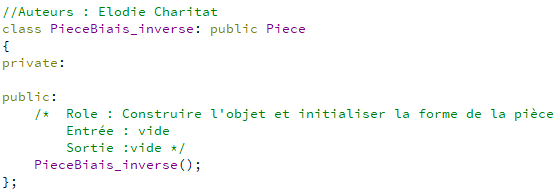
## PieceBaton.h



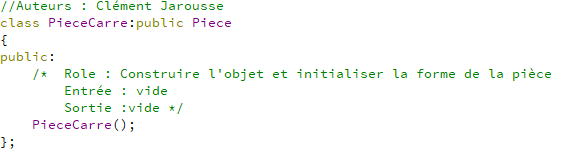
## PieceBiais.h



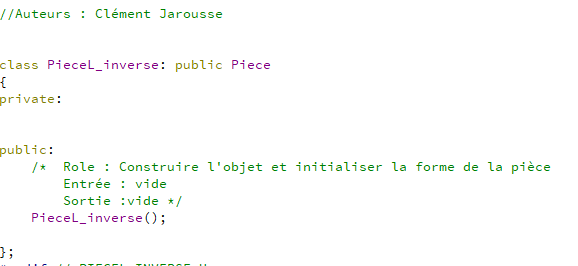
## PieceBiais\_inverse.h



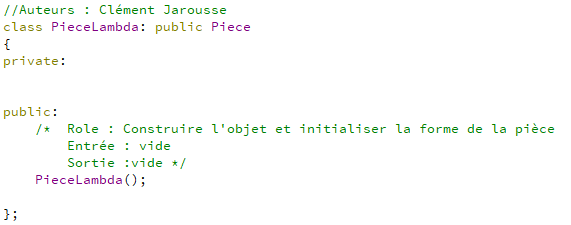
## PieceCarre.h



## PieceL\_inverse.h



## PieceLambda.h



## PieceT\_bloc.h

