

Classe Vitrail:

- **Vitrail()** : Constructeur par défaut. Initie cols et vitresParColone à 0.
- **Vitrail (int n_cols, int n_vitres)** : Constructeur avec param. Alloue dynamiquement les colonnes du vitrail avec n_cols comme nombre de colonnes et n_vitres comme nombre de rangées. Pour chaque rangée, color1, color2 et h sont choisis au hasard et correspondent aux couleurs des vitraux et de la longueur des segments, définit par h.
- **int construireVitrail(std::vector<char> vitres, int colonne)** : traverse la colonne donnée en paramètre et si la valeur du char sur lequel l'itérateur est le même que l'élément actuel de la boucle, il le remplace par un 'x'. Sinon incrémente l'itérateur.
- **bool estComplete(int colonne)** : traverse la colonne donnée et si un char autre que 'x' est trouvé, retourne faux.
- **bool estEnConstruction(int colonne)** : traverse la colonne donnée et si un 'x' est trouvé, retourne vrai.
- **bool estCompleteFull()** : Vérifie si toutes les colonnes du vitrail sont complètes. Utilise pour la fonction main. Traverse chaque colonne du vitrail et utilise la fonction estComplete(colonne) afin de déterminer si toutes les colonnes sont complètes
- **char intToChar(int color)** : fonction qui aide à convertir les nombre choisis au hasard en char représentant les couleurs.

Getter et setters :

- **int getCols()** :getter pour cols
- **int getRows()** : getter pour rows
- **char **getDesc()** : getter pour le tableau

Classe Joueur

- **Joueur (nom_j)** : constructeur qui initie un tableau vide avec le nom de joueur
- **Joueur (nom_j, n_cols, n_vitres)** : constructeur qui initie un tableau de n_cols x n_vitres avec le nom de joueur
- **Getters et setters :**
 - int getPosition()
 - void setPosition(int x)
 - Vitrail *getVitrail()
 - int getPoints()
 - void updatePoints(int x)

classe Lots :

- **Lots()** : Constructeur par défaut
- **~Lots()** : destructeur. Loop a travers desc et efface les colonnes
- **std::vector<char> ramasseVitre(char couleur, int numeroLot);**
- **void reset()** : reset le lot. Choisis un nombre une couleur au hasard et place dans la colonne.
- **std::string toStringTableau()** : met le lot en string. Utilisé dans le l'opérateur <<
- **std::string toStringSurplus()** : met le surplus en string. Utilisé dans le l'opérateur <<
- **bool collsEmpty(int col)** : retourne vrai si la colonne est vide. Traverse la colonne donnée et retourne true si un 'x' est trouvé
- **bool lotsEmpty()** : retourne vrai si le lot est vide. Vérifie que toutes les colonnes sont vides en utilisant la fonction collsEmpty(int col).
- **bool surplusIsEmpty()** : retourne vrai si le surplus est vide. Si la taille du vecteur est 0, il retourne vrai.
- **char intToChar(int color)** : fonction qui aide à convertir les nombre choisis au hasard en char représentant les couleurs.

- **int getSizeCol(int x)** : retourne la taille de la colonne x. Si négatif, retourne la taille du surplus
- **Getters et Setters**
 - **char **getLots()**
 - **int getRows()**
 - **int getCols()**