

Année universitaire 2016-2017
Université de Caen Basse-Normandie

Rapport sur la numérotation des lignes sur l'éditeur

Thomas Lécluse
L2 Informatique

Table des matières

1	Insertion dans l'interface graphique	2
2	Comment faire ?	3
2.1	Les signaux	3
2.2	Surcharger les méthodes originales	3
3	Pour finir	6

1 Insertion dans l'interface graphique

L'objet que nous utilisons pour afficher et éditer le code est un QTextEdit de QT. Il ne possède pas de méthode pour afficher les numéros de lignes, et il n'y a pas d'autres alternatives à celui-ci pour faire cela non plus.

Il nous fallait donc un autre élément, afin d'afficher les numéros de lignes qui serait placé sur le côté, nous avons choisi le côté droit.

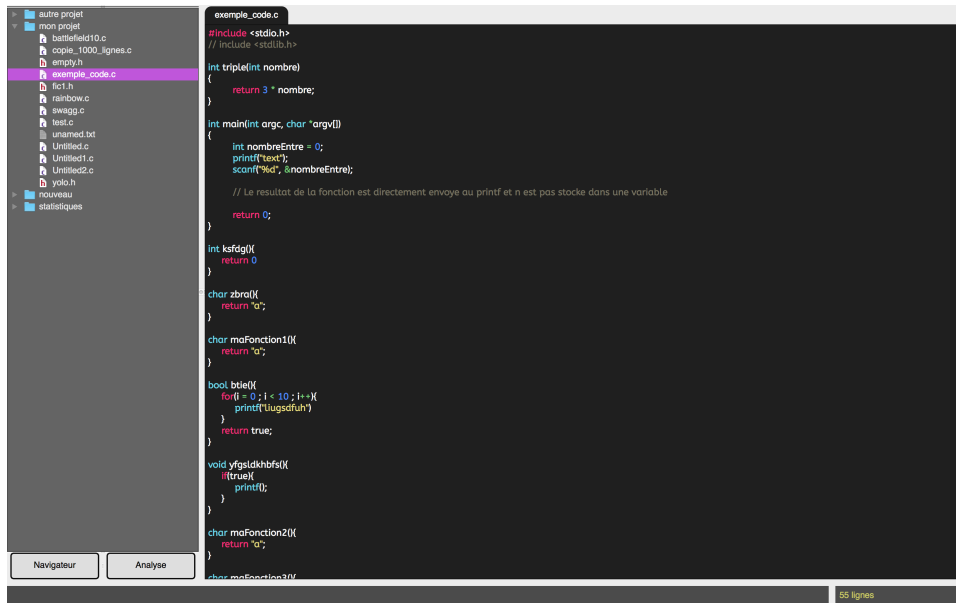


FIGURE 1 – Sans la barre de numérotation des lignes

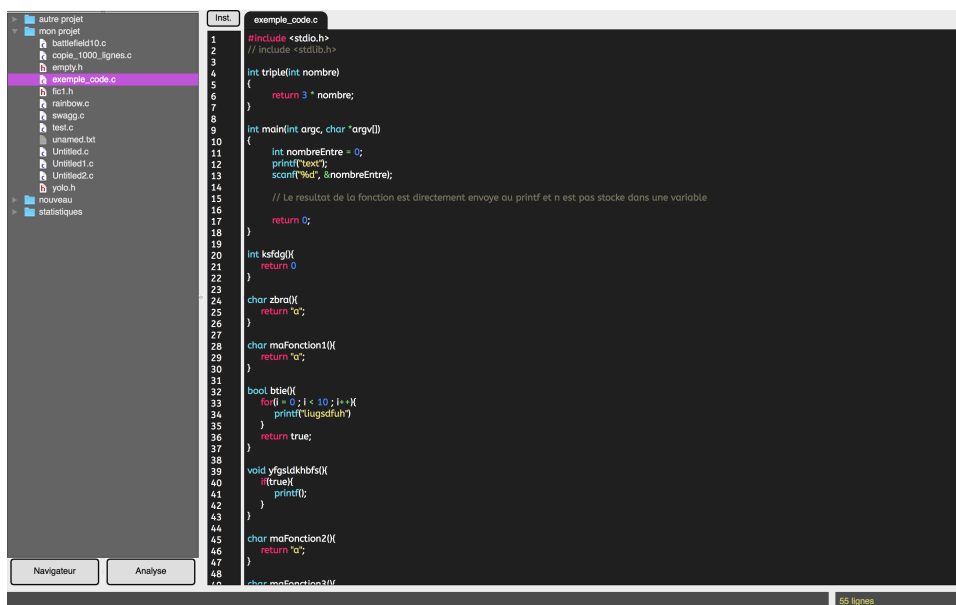


FIGURE 2 – Avec la barre de numérotation des lignes

L'utilisateur peut choisir d'afficher ou non cette barre.

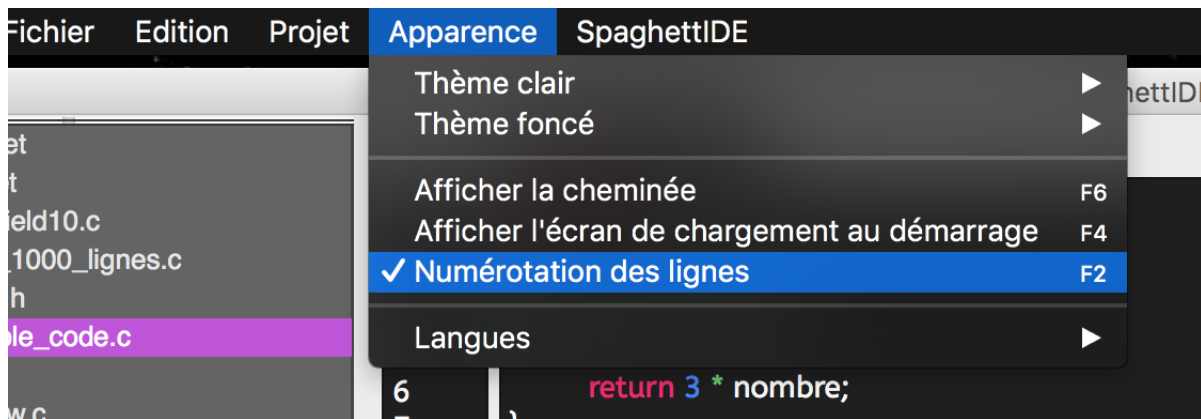


FIGURE 3 – Dans le menu apparence

2 Comment faire ?

La difficulté n'était pas d'afficher une barre avec des numéros dedans, ni de récupérer le nombre de lignes, car nous avons une méthode pour cela. La difficulté était de synchroniser le défilement des deux éléments (widgets).

2.1 Les signaux

Une première idée était d'intercepter les signaux émis par les éléments respectifs, afin d'envoyer à l'autre élément le décalage à effectuer. Le problème était qu'il n'y a pas de moyens de connaître la première ligne affichée sur le QTextEdit du code. On ne peut pas donc savoir de combien on décale d'autre façon d'incrémenter et de décrémenter "manuellement" une variable en fonction du défilement que l'on effectue.

Le soucis est que c'est très compliqué (pas réussi) de récupérer le nombre de lignes que l'on a décalé lors du défilement. La seule chose qu'on arrive à récupérer est un delta qui correspond à la vitesse de défilement que l'utilisateur a fait avec la molette de sa souris ou de son pad. Cela nous donne juste le sens du défilement.

Il a fallu chercher une autre solution, plus propre et plus efficace.

2.2 Surcharger les méthodes originales

Lorsque l'on fait un défilement avec la molette de souris ou le pad **uniquement**, la méthode appelée sur le QTextEdit est le "wheelEvent(e)" où "e" est l'événement utilisé par QT pour effectuer le défilement.

Le principe consiste à appeler la fonction d'origine ainsi que celle de l'autre objet avec le même argument "e" lorsque la méthode "wheelEvent()" est appelée.

```
def wheelEvent(self, e, syncr=False):
    """
    Événement appelé lors du scroll via la souris

    :param e: evenemnt
    :type e: object
    """
    QTextEdit.wheelEvent(self, e)
    if not syncr:
        self.master.codes[self.master.get_idx()].wheelEvent(e, True)
```

FIGURE 4 – Méthode de l'objet contenant la numérotation des lignes

Ici, "self.master.codes" désigne la liste des onglets de codes ouverts, et "self.master.get_idx()" retourne l'indice de l'onglet courant. On appelle donc la méthode "wheelEvent()" de l'onglet courant lorsque l'on fait défiler la liste de numérotation des lignes. L'argument booléen sert à dire qu'il ne faut pas rappeler la méthode "wheelEvent()" car sinon on rentrerait dans une boucle infinie.

```
def wheelEvent(self, e, syncr=False):
    """
    Événement appelé lors du scroll via la souris

    On rappelle ici la même fonction sur l'objet pour afficher les lignes afin de les synchroniser.

    :param e: evenemnt
    :type e: object
    """
    QTextEdit.wheelEvent(self, e)
    if not syncr:
        self.parent.nb_lignes.wheelEvent(e, True)
```

FIGURE 5 – Méthode de l'objet contenant le code

"self.parent.nb_lignes" désigne l'objet contenant la numérotation des lignes, on y appelle donc la méthode "wheelEvent()" avec les mêmes arguments. Et toujours l'argument empêchant la boucle infinie.

Nous arrivons ainsi à synchroniser les deux éléments lors du défilement de l'un comme de l'autre avec la souris ou le pad.

Le bon fonctionnement de cette fonctionnalité est difficile à montrer dans un rapport, mais voici tout de même un exemple :

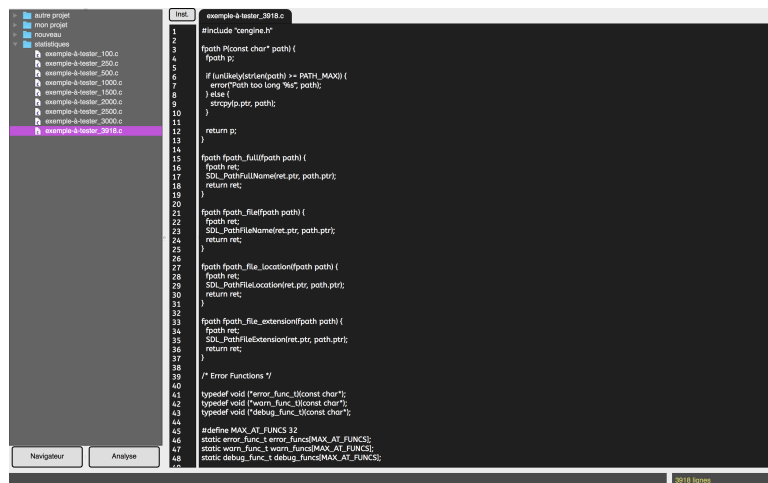


FIGURE 6 – Premier exemple : haut du code, bien en face des numéros

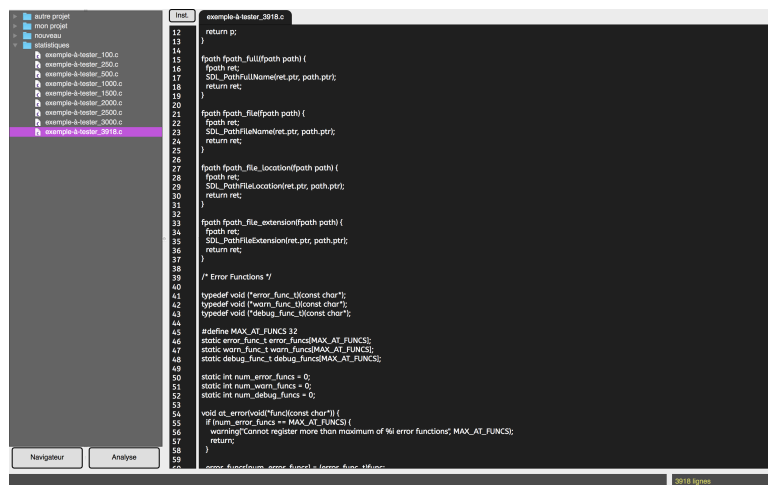


FIGURE 7 – Deuxième exemple : les lignes sont toujours en face du code correspondant

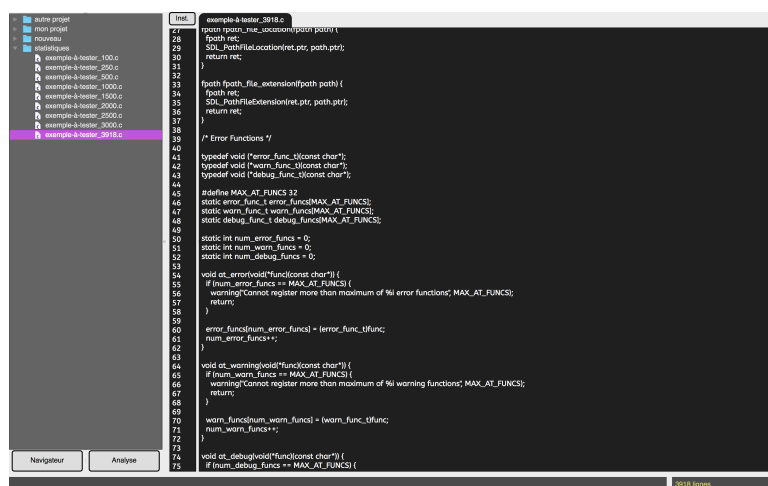


FIGURE 8 – Dernier exemple : même si la moitié d’une ligne est affichée (en haut), la numérotation est toujours alignée

3 Pour finir

Néanmoins, lorsque le texte affiché est défilé à l'aide des flèches, la méthode "wheelEvent()" n'est pas appelée. Il va donc falloir trouver un moyen de synchroniser tout de même ces éléments.