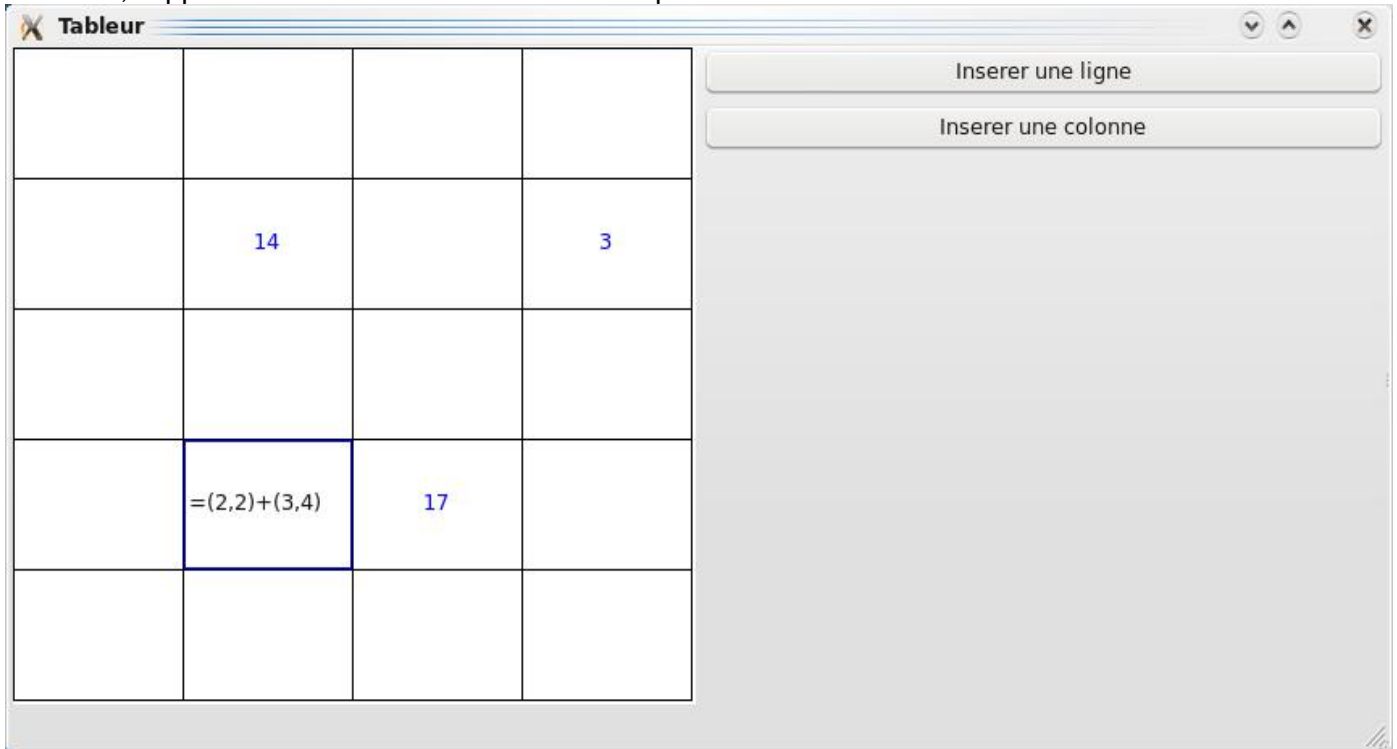


TP 5 *Création d'un composant de type « tableur »*

Dans ce TP, nous allons réaliser un widget « tableur » que nous allons intégrer dans une application de test.

Au final, l'application devrait ressembler à l'exemple suivant :



Notre application est composée d'un widget « tableur » que nous allons créer ainsi que de deux boutons.

Exercice 1 : Ébauche du widget « tableur »

Graphiquement, le widget « tableur » ressemble à une grille composée de cellules. Les cellules peuvent contenir des éléments qui sont affichés au centre.

La première étape est de créer un widget qui réagit aux événements de base de Qt (redimensionnement et rafraîchissement) de manière à conserver l'affichage de la grille. Les dimensions de la grille (nombre de colonnes et de lignes) sont données (en argument) au constructeur du widget.

La deuxième étape est de pouvoir sélectionner une case à la souris. Pour cela il faut récupérer les coordonnées de la souris lorsque l'utilisateur a cliqué sur une case du tableur. Il est préférable de convertir ces coordonnées (pixels) vers des coordonnées plus aisément manipulables (n°ligne, n°colonne). Enfin, il faut dessiner un cadre de couleur sur la case sélectionnée pour la mettre en évidence.

Exercice 2 : Éditer une case

Une des fonctionnalités de base d'un tableur est de pouvoir éditer une cellule. Pour cela, nous allons utiliser un widget « QLineEdit » que nous allons adapter à notre utilisation. Lorsque le widget tableur est créé, il crée un widget QLineEdit caché et désactivé par défaut. Lorsque l'utilisateur sélectionne une

case, le QLineEdit est déplacé, redimensionné et montré sur la case sélectionnée. Ainsi nous avons à notre disposition un composant d'édition de texte que nous déplaçons et montrons quand nous en avons besoin.

Exercice 3 : Création du moteur

Le moteur est en fait ici une liste de cellules que nous manipulons. Notons tout d'abord qu'il n'est pas judicieux de créer en mémoire une matrice de la taille de notre tableur. Cela prendrait beaucoup de place inutilement. Il est plus intéressant d'ajouter les cellules dans une liste au fur et à mesure de leur création. Une cellule est dite « créée » à partir du moment où elle contient des informations. On ne stocke donc que ce qui est nécessaire.

Dans notre tableur nous allons manipuler deux types de cellules : les cellules dites « entier », et les cellules dites « formule ». Au sein de notre composant graphique, nous gérerons une liste de cellules. Il faut donc créer la hiérarchie « objet » adéquat pour répondre à nos besoins. Une cellule est bien entendue caractérisée par son abscisse et son ordonnée ainsi que par la donnée qu'elle contient.

Les cellules « entier » stockent un entier résultant d'une saisie de l'utilisateur (grâce au composant d'édition du point précédent), cet entier est affiché au centre de la cellule lorsque celle-ci est affichée.

Les cellules « formule » stockent une chaîne de caractères représentant une séquence d'opérations (voir la capture d'écran). On se limitera uniquement à l'addition et la soustraction de cellules entre elles. Une formule se présente donc sous la forme : $(x1,y1)+(x2,y2)-(x3,y3)$ etc. **Attention**, les cellules de type formule stockent une formule. Lors de l'édition de la cellule, c'est la formule qu'on édite, mais lors de l'affichage de la cellule, c'est le résultat de la formule qui est affiché.

Exercice 4 : Fonctionnalité supplémentaire

Comme fonctionnalité supplémentaire nous allons implémenter l'insertion d'une ligne ou d'une colonne. Pour cela nous allons créer deux slots dans notre tableur qui permettront d'insérer une ligne (resp. une colonne) juste avant la case actuellement sélectionnée dans notre widget tableur. Ces deux slots seront connectés aux signaux des boutons vus sur la capture d'écran.

Lors de l'insertion d'une ligne (resp. d'une colonne), toutes les cellules qui se trouvent en dessous (respectivement à droite) de la ligne (resp. la colonne) seront décalées d'une ligne vers le bas (resp. d'une colonne vers la droite). Dans un premier temps, il n'est pas demandé de modifier les formules des cellules « formule » pour les mettre à jour.