



# TOB : Projet Long

## Rapport itération 3

Groupe MN-4

Le projet de notre groupe est de développer un tableur gratuit, permettant de faire des calculs et des opérations simples sur les cellules. Il est destiné à un public assez vaste, de l'utilisateur peu expérimenté grâce à sa simplicité d'utilisation à l'utilisateur plus expérimenté avec ses traitements mathématiques et calculs sur cellules.

### 1 Principales fonctionnalités :

Fonctionnalité	Etat d'avancement	Itération
Interface graphique	Opérationnel	1
Ecriture et enregistrement de données	Opérationnel	1
Parseur de formules mathématiques	Opérationnel	1
Menu des fonctionnalités	Opérationnel	1
Sauvegarde de tableur dans un fichier	Opérationnel	2
Ouverture d'un fichier tableur utilisable	Opérationnel	2
Interprétation de formule avec un "="	Opérationnel	2
Historique d'actions	En cours	1
Créer nouveau tableur vierge	Opérationnel	3
Sélection d'une cellule et mis en surbrillance	Opérationnel	3
Création de graphiques 2D	En cours	3
Traitements mathématiques de données	En cours	1
Calculs mathématiques simples	En cours	1
Tests de lois de probabilités	A faire	n/a
Traitements statistiques	A faire	n/a
Copier / Coller	En cours	2
Personnalisation des cellules	A faire	n/a
Navigation par souris/clavier/recherche	En cours	3

## 2 Découpage en paquetages de l'application :

- Paquetage tableur :
  - Tableur
  - TableurControleur
  - TableurSwing
  - TableurVisuel
- Paquetage tableur.action :
  - Action (Interface)
- Paquetage tableur.action.elementaire
  - ActionCopier
  - ActionCouper
  - ActionDefinirCellule
  - ActionSelectionnerCellule
- Paquetage tableur.composant
  - Cellule
  - Historique
  - Position
  - CelluleFormule
- Paquetage tableur.exception
  - FormatFormuleException
  - NoFormuleException
  - NoMeasurableException
  - NoValuableException
- Paquetage tableur.graph
  - FenetreGraphique
- tableur.math
  - Formule
  - Lecteur
  - Traitement (Interface)
- Paquetage tableur.math.formule
  - FormuleAddition
  - FormuleCellule
  - FormuleCelluleRelative
  - FormuleConstante

- FormuleDivision
- FormuleMultiplication
- FormuleParenthese
- FormulePuissance
- FormuleSomme
- FormuleSoustraction
- FormuleVide
- Paquetage tableur.math.traitement
  - TraitementProduit
  - TraitementPuissance
  - TraitementSomme

### 3 Diagramme de classe :

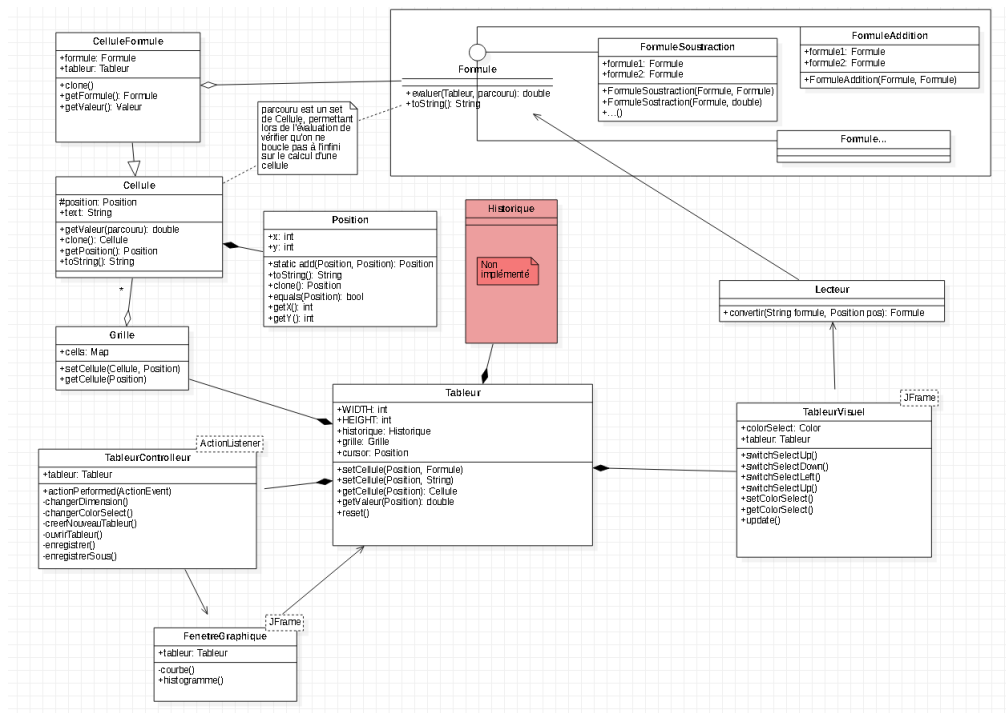


FIGURE 1 – Diagramme de classes du tableur

### 4 Réalisation :

Voici l'interface graphique actuelle du tableur.

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										

- Choix de conceptions : Le tableur graphique a ses propres cellules indépendantes du tableur objet, il doit donc être mis à jour en permanence pour savoir quoi afficher.
  - Difficultés : La difficulté vient du fait que le visuel doit afficher différents états de la cellule en fonction de la sélection de l'utilisateur  
 Lors du déplacement avec les flèches ce qui captait la prise du "focus" de la cellule n'est plus sélectionné / accessible en direct  
 Les cellules du tableur étant stockées dans une HashMap de Position/Cellule, 2 instances de Position représentant la même Cellule auront en fait des clés différentes pour la HashMap, on avait donc une erreur pour accéder aux cellules
  - Solutions : Chaque Position est représentée par un String ce qui permet de l'identifier même pour 2 instances de la même position, par exemple Position(1,3) donne le String "1 :3", la HashMap fonctionne alors avec les Cellules qui deviennent accessibles.
- Choix de conception 2 : Nous avons décidé de coder un système de sauvegarde du tableur afin de pouvoir reprendre une tâche entamée plus tard.
  - Difficultés : Tous les objets ne sont pas "Serializable" ce qui signifie que le tableur dans son intégralité ne peut pas être sauvegarder correctement. De plus, le système de sauvegarde doit être aussi intuitif/pratique que pour un tableur commercial.
  - Solution : Nous avons décidé de n'enregistrer que les Cellules du tableur grâce à l'HashMap, en effet les Cellules et Positions étaient compatibles avec l'interface Serializable qui permet de convertir un objet Java en fichier binaire, ce qui n'était pas le cas de tous les éléments du tableur. On reconstruit ensuite le tableur à partir de l'HashMap, ce qui permet également de réduire l'espace de stockage nécessaire. Pour le confort d'utilisation, le même principe que les logiciels commerciaux a été adopté : si un fichier a été "enregistré sous" ou "ouvert" alors le bouton "enregistrer" enregistre automatiquement le tableur dans le bon fichier, sinon il demande à l'utilisateur de créer un nouveau

fichier et lui affiche une interface de navigation prévue à cet effet.

- Choix de conception 3 : Il est possible de changer de cellule avec les flèches du clavier pour ainsi naviguer dans le tableur graphique.
  - Difficultés : La difficulté vient du fait qu'il fallait, en plus des actions de bases de chaque cellule créer un lien pour pouvoir changer de cellule. De plus les fonctions de base de `java.awt.event.*` permettent seulement de changer le focus sur la cellule précédente et la cellule suivant.
  - Solution : Il fallait un `KeyListener` par cellule et créer une fonction qui permet de changer de focus d'une cellule à partir de sa position (i,j)

## 5 Organisation de l'équipe / Mise en oeuvre des méthodes agiles :

L'équipe s'est organisé en se donnant un temps de travail collectif et un temps autonome, avec une réunion hebdomadaire le mercredi après-midi et un temps de travail à la discrétion de chacun.

Maël s'est occupé de l'architecture du code et de la majeure partie des décisions techniques, tout en participant massivement au développement des fonctions.

Steven et Johan ont principalement participé au développement FrontEnd de l'application, que ce soit dans l'aspect visuel ou fonctionnel.

Tom a travaillé sur le BackEnd du programme et la manière dont l'application interagit avec l'environnement de travail (Sauvegarde, ouverture de fichier, navigation), de plus il a développé plusieurs fonctions mathématiques en vue de les utiliser dans le tableur.

Léo a travaillé sur l'historique et le traitement de données de manière graphique.

Justin a travaillé sur la partie calcul statistiques et traitements mathématiques.

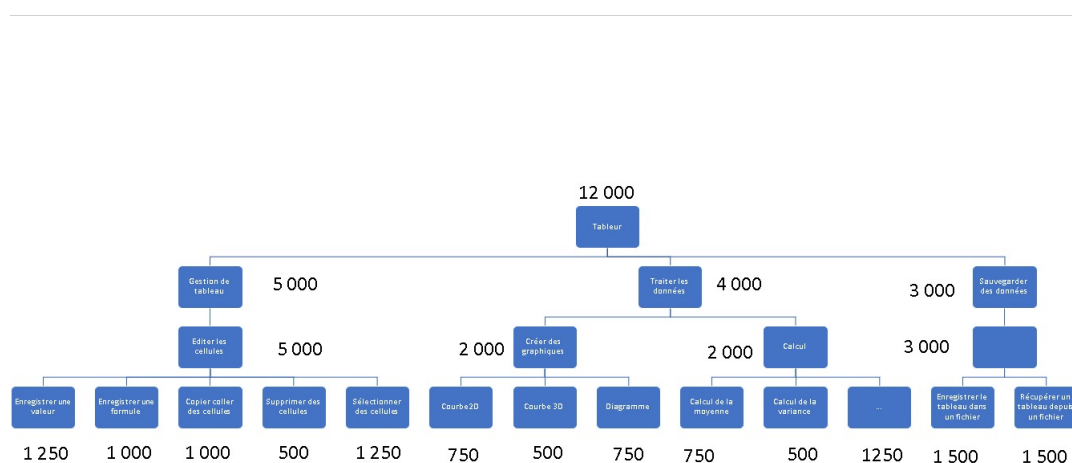


FIGURE 2 – Diagramme des valeurs métier

Vision produit
<p><u>POUR?</u> Les professionnels et initiés aux mathématiques</p> <p><u>QUI SOUHAITENT?</u> Utiliser un outil mathématique statistique</p> <p><u>NOTRE PRODUIT EST?</u> Un tableur graphique</p> <p><u>QUI?</u> Permet d'enregistrer et traiter des données et les soumettre à un traitement graphique de façon intuitive</p> <p><u>A LA DIFFERENCE DE?</u> Logiciels de ce type</p> <p><u>PERMET DE?</u> Utiliser des traitements mathématiques usuels de façon accessible</p>
Clients / Utilisateurs
<ul style="list-style-type: none"> <li>• Initiés aux mathématiques</li> <li>• Métiers de comptabilité</li> <li>• Lycéens</li> </ul>
Valeur métier
<ol style="list-style-type: none"> <li>1. Enregistrer / ajouter des données</li> <li>2. Sauvegarder des données (traitées ou non)</li> <li>3. Traiter / exploiter des données</li> <li>4. Afficher des graphiques à partir d'une série de données</li> <li>5. Personnalisation des cellules</li> <li>6. Navigation par souris / clavier</li> <li>7. Copier / coller</li> <li>8. Complétion / remplissage de cellules avec un traitement mathématique</li> </ol>
Jalons : dates ou fonctionnalités
<p>21 / 03 / 2020</p> <ul style="list-style-type: none"> <li>- Cellules utilisables (enregistrement de données)</li> <li>- Navigation dans les cellules (pas encore de manière graphique)</li> <li>- Application de traitements</li> <li>- Sauvegarde de données et chargement</li> </ul>

FIGURE 3 – Diagramme User Stories