

Documentation Technique des Barres d'Outils de l'Application de Visualisation 3D

Thibault Bouchard

12 août 2016

Table des matières

1	Introduction	3
2	Description des barres d'outils dans l'application	4
2.1	Vue d'ensemble de l'interface de l'application	4
2.1.1	Zone contenant les barres d'outils agissant sur toute l'application	5
2.1.2	Zone contenant les barres d'outils agissant sur une vue	7
3	Description technique des barres d'outils et des outils en eux-même	12
3.1	Introduction	12
3.2	Description technique	12
3.2.1	Barre d'outils	12
3.2.2	Outils abstraits (ElementTool et InputTool)	14
3.2.3	Outils de saisie de texte (TextTool)	14
3.2.4	Outils de création d'onglet (TabTool et TabContainerTool)	15
3.2.5	Outils bouton ON/OFF (SwitchTool)	16
3.2.6	Outils de jauge (RangeTool)	17
3.2.7	Outils d'affichage de nombre (PrintTool)	18
3.2.8	Outils de menu déroulant (ListTool)	19
3.2.9	Conteneur d'outils (FolderTool)	20
3.2.10	Outils de sélection de couleur (ColorTool)	21
3.2.11	Outils de sélection de données (CaseTool)	21
3.2.12	Outils bouton (ButtonTool)	22
4	A Savoir	24

Chapitre 1

Introduction

Au début du développement de l'application, on utilisait les barres d'outils `dat.GUI`. Or, ce type de barre d'outils qui peut être suffisant pour des applications de faible envergure ne s'est pas avéré suffisant pour l'application de visualisation 3D. En effet, l'application nécessite un grand nombre d'outils ainsi qu'une bonne ergonomie (c'est-à-dire une meilleure organisation des outils) afin de rendre l'utilisation de l'application assez simple. De plus, `Dat.GUI` ne répondait pas aux besoins techniques de l'application telle que de modifier l'état d'un bouton depuis le code (par exemple, cocher ou décocher une checkbox), en effet, `dat.GUI` ne permet que de modifier l'état d'un bouton par l'utilisateur.

Ce document décrira d'une part les fonctionnalités des outils présents dans l'application et d'autre part une description technique des outils afin de savoir par exemple comment ajouter un outil dans une barre d'outils.

Chapitre 2

Description des barres d'outils dans l'application

2.1 Vue d'ensemble de l'interface de l'application

Voici une vue d'ensemble de l'application :

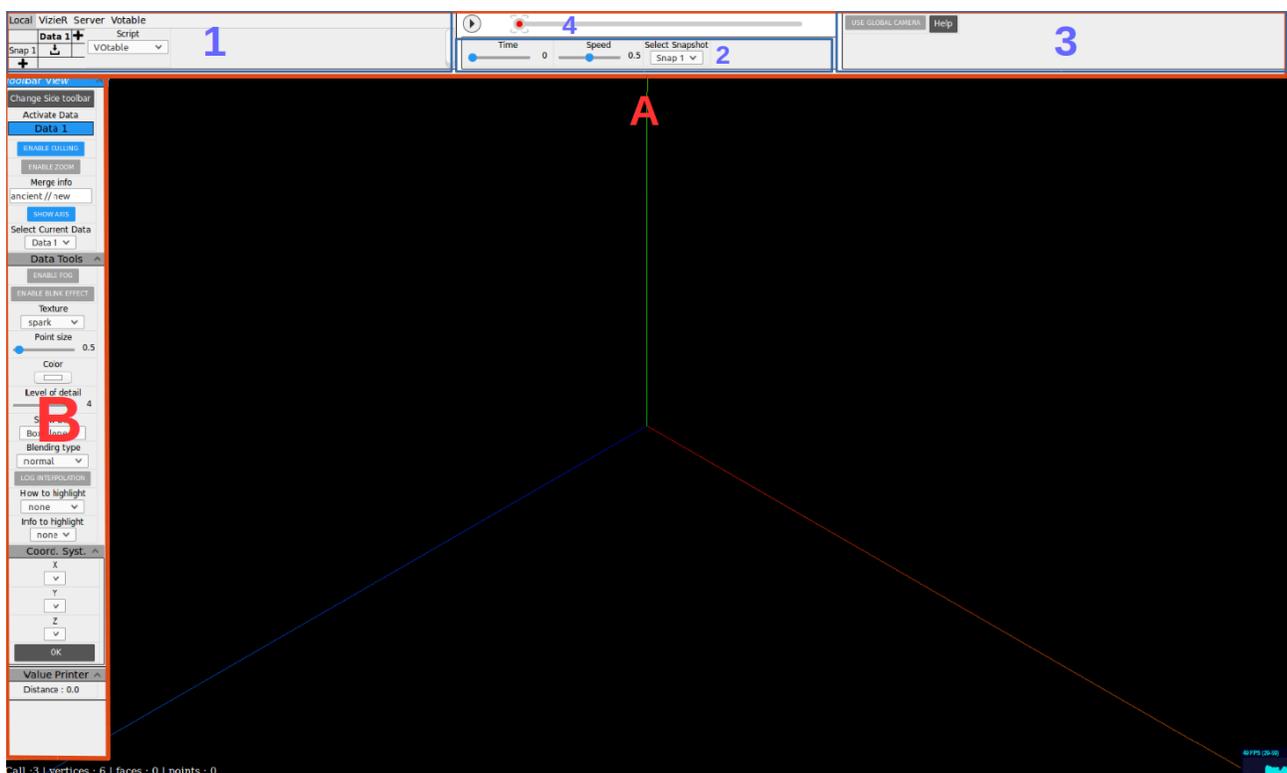


FIGURE 2.1 – Vue d'ensemble de l'interface de l'application

Voici à quoi correspond chaque zone de l'interface :

- A : Zone contenant les barres d'outils agissant sur toute l'application.
- B : Zone contenant les barres d'outils agissant sur une vue.
- 1 : Barre d'outils s'occupant du chargement des données dans l'application.
- 2 : Barre d'outils modifiant les paramètres d'animation des données.

3 : Barre d'outils contenant les outils généraux de l'application.

4 : Outil s'occupant de l'animation des données.

2.1.1 Zone contenant les barres d'outils agissant sur toute l'application

Barre d'outils s'occupant du chargement des données dans l'application

Cette barre d'outils est découpée en 4 onglets, voici une description de ces onglets :

Onglet "Local" : L'onglet local contient les outils de chargement des données depuis le disque dur, il contient deux outils :

- L'outil de chargement des données qui permet de charger une donnée dans un slot *data* et *snapshot* particulier. Il est possible d'ajouter un slot *data* en cliquant sur le signe plus de la ligne où il y a écrit *Data* et il est possible d'ajouter un slot *snapshot* en cliquant sur le signe plus de la colonne où il y a écrit *snapshot*. Pour allouer une donnée dans le slot $[Data\ i; Snapshot\ j]$, il suffit de cliquer sur le symbole dans la colonne *Data* i et la ligne *Snapshot* j . Une fois la donnée chargée, il est possible d'afficher le nom du fichier chargé en laissant le curseur de la souris sur le slot de la donnée.
- L'outil de sélection de script permet de sélectionner le script qui va lire les données que l'on va charger en local après le sélection de ce script.



FIGURE 2.2 – Vue de l'onglet "Local"

Onglet "VizieR" : L'onglet VizieR contient les outils de chargement des données depuis VizieR, il contient deux outils :

- L'outil **ADQL Query** permettant d'écrire une requête VizieR.
- L'outil **Submit** permettant d'envoyer la requête VizieR.



FIGURE 2.3 – Vue de l'onglet "VizieR"

Onglet "Server" : L'onglet Server contient les outils de chargement des données depuis un serveur, il contient trois outils :

- L'outil **IP** permet d'écrire l'IP du serveur (*ATTENTION* : Pour le moment (Aout 2016), ce champ n'est pas utilisé).
- L'outil **Port** permet d'envoyer le port du serveur (*ATTENTION* : Pour le moment (Aout 2016), ce champ n'est pas utilisé).
- L'outil **Load** permet de charger les données depuis le serveur.
- L'outil **Load on Move** permet de dire si oui ou non on veut charger depuis le serveur les données quand la caméra bouge.

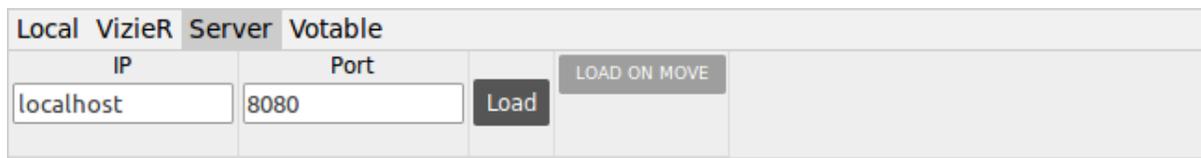


FIGURE 2.4 – Vue de l'onglet "Server"

Onglet "VoTable" : L'onglet VoTable contient les outils de chargement des données VoTable, il contient cinq outils :

- L'outil **Field RAJ2000** permettant de modifier la dénomination du champ *RAJ2000* pour le chargement des données VoTable.
- L'outil **Field DEJ2000** permettant de modifier la dénomination du champ *DEJ2000* pour le chargement des données VoTable.
- L'outil **Field zsp** permettant de modifier la dénomination du champ *zsp* pour le chargement des données VoTable.
- L'outil **Field umag** permettant de modifier la dénomination du champ *umag* pour le chargement des données VoTable.
- L'outil **Field gmag** permettant de modifier la dénomination du champ *gmag* pour le chargement des données VoTable.



FIGURE 2.5 – Vue de l'onglet "VoTable"

Barre d'outils modifiant les paramètres d'animation des données

Cette barre d'outils permet de modifier les paramètres d'animation des données. Elle contient trois outils :

- L'outil **Time** qui permet de fixer l'état du temps de l'animation.
- L'outil **Speed** qui permet de fixer la vitesse de l'animation.
- L'outil **Select Snapshot** qui permet de fixer quelle snapshot afficher.

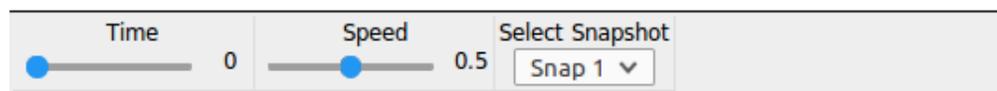


FIGURE 2.6 – Vue de la barre d'outils d'animation

Barre d'outils contenant les outils généraux de l'application

Cette barre d'outils contient les outils généraux de l'application. Elle contient trois outils :

- L'outil **Use Global Camera** qui permet de pouvoir contrôler les caméras de toutes les vues en mode de vue Zoom ou Multivue en même temps.
- L'outil **Help** qui permet d'afficher une aide. Cette aide s'affiche dans une fenêtre au centre de l'application.
- L'outil **Credits** qui permet d'afficher les noms des contributeurs de l'application.



FIGURE 2.7 – Vue de la barre d'outils général

Outil s'occupant de l'animation des données

Cet outil permet de lancer une animation des données quand celle-ci comporte plusieurs snapshots. Le bouton **Play** permet de lancer une animation. Cliquer sur la barre grise permet d'aller à un temps de l'animation précis. Pour plus de détail au sujet de cette barre, il faut se référer au rapport de stage de *Pierre Lespingal* et *Nicolas Buecher*.



FIGURE 2.8 – Vue de l'outil d'animation

2.1.2 Zone contenant les barres d'outils agissant sur une vue

Cette barre d'outils agit sur une vue. Si l'application utilise le zoom ou si elle est en mode multivue, il y aura une barre d'outils pour chaque vue. Il est possible de cacher ces barres d'outils en cliquant en haut de celle-ci sur

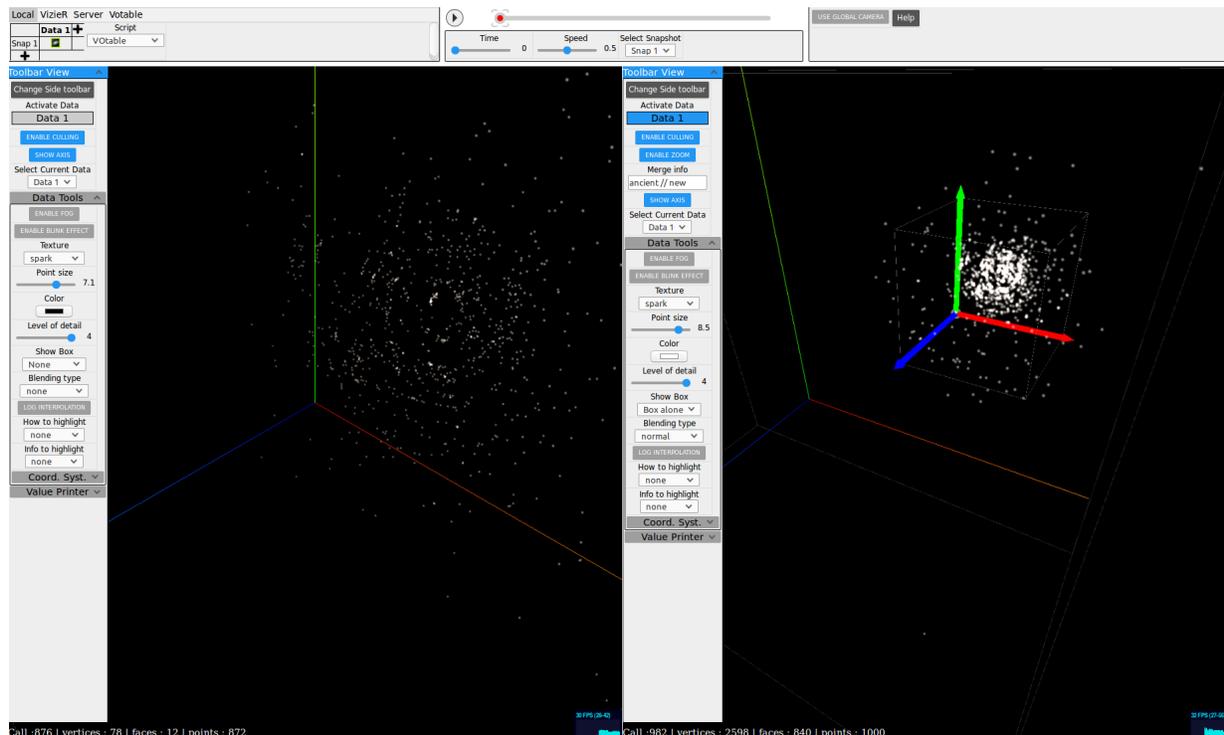


FIGURE 2.9 – L'application en mode de zoom (l'interface est pratiquement la mode en mode multivue)

Toolbar View. Une fois caché, un nouveau clic sur celle-ci permet de les ré-afficher.

Ces barres d'outils sont découpées en plusieurs parties :

Outils généraux de la vue

Ces outils se situent en haut de la barre d'outils. Ils sont toujours affichés dans la barre d'outils. Voici une description de ces outils :

- L'outil **Change Side Toolbar** permet de changer la position de la barre d'outils, quand la barre d'outils est à gauche, un clic permet de mettre la barre d'outils à droite et si la barre d'outils est à droite, un clic permet de mettre la barre d'outils à gauche.
- L'outil **Activate Data** permet de choisir les datas à afficher dans la vue. Toutes les datas sont affichées dans l'outil, pour afficher une data, il suffit que la couleur de cette data dans l'outil soit bleue. Si la data n'est pas affichée, alors sa couleur dans l'outil sera grise.
- L'outil **Enable Culling** sert à utiliser le frustum culling
- L'outil **Show Axis** permet d'afficher ou non les axes dans la vue.
- L'outil **Select Current Data** permet de sélectionner la data courante de la vue, c'est-à-dire la data sur laquelle les outils spécifique à une data de la barre d'outils agit.

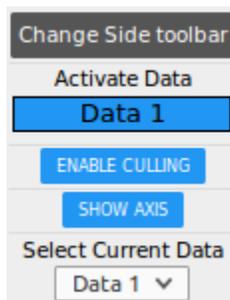


FIGURE 2.10 – Outils généraux d'une vue

Outils spécifiques à une Data

Ces outils se situent dans une section nommée **Data Tools**. Il est possible de ne pas afficher ces outils en cliquant sur le titre de la section et de les ré-afficher en effectuant la même action. Ces outils agissent spécifiquement sur une data. Il est possible de choisir la data courante en la choisissant dans l'outil **Select Current Data**. Quand on modifie la data courante, les outils de cette section se mettent à jour (en effet, on garde en mémoire l'état de ces outils pour chaque data). Voici les outils qui composent cette section :

- L'outil **Enable Fog** permet d'activer le brouillard (Plus une particule est loin, moins elle sera visible).
- L'outil **Enable Blink Effect** permet de faire scintiller les particules.
- L'outil **Texture** permet de choisir la texture des particules.
- L'outil **Point Size** permet de modifier la taille des particules.
- L'outil **Color** permet de modifier la couleur des particules.
- L'outil **Level of Detail** permet de modifier le niveau de détail affiché dans la vue.
- L'outil **Show Box** permet de choisir le type de boîte englobante de la data. Il est possible de ne pas afficher de boîte, d'afficher une boîte englobante simple, d'afficher une boîte englobante avec 10 graduations sur chaque arête et d'afficher une boîte englobante avec une grille 10 x 10 sur les faces de la boîte.
- L'outil **Blending Type** permet de choisir le type de blending de la data.
- L'outil **Log Interpolation** permet d'avoir une interpolation logarithmique des couleurs quand on met en valeur un attribut d'une donnée.
- L'outil **How to Highlight** permet de choisir comment mettre en valeur un attribut de la data courante.
- L'outil **Info to Highlight** permet de choisir quel attribut de la data courante mettre en valeur.
- La section **Coord. Syst.** permet d'agir sur le système de coordonnées de la data courante. On peut donc afficher la data le long d'un axe en fonction d'un attribut de la data (d'où les champs **X**, **Y** et **Z**. Le bouton **OK** permet d'appliquer les modifications du système de coordonnées sur la data courante. Les champs **Log Inter. X**, **Log Inter. Y** et **Log Inter. Z** permettent de choisir si on veut avoir des coordonnées sur une échelle logarithmique ou non selon les axes X, Y et Z. On appelle *échelle logarithmique* l'application d'une

fonction tel que :

$$f(x) = \begin{cases} \log(x) & \text{si } x > 0 \\ 0 & \text{si } x = 0 \\ -\log(-x) & \text{si } x < 0 \end{cases} \quad (2.1)$$

- La section **Value Data** permet de voir des données appartenant à la vue. On affiche les échelles des données le long des axes X,Y,Z d'un graduation de la boîte englobante de la donnée. Ces échelles sont calculées en fonction des données réelles (des données chargés en mémoire) et non des données affichées dans la vue (qui ont été normalisées afin d'être contenues dans un cube de 1 x 1 x 1).

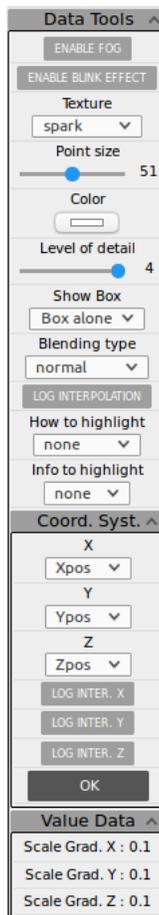


FIGURE 2.11 – Outils spécifique à une data

Outils de changement de nom d'attribut dans les données

Les outils présents dans cette section servent à modifier le nom d'un attribut pour données VoTable. En effet, il est possible qu'un même attribut ait un nom différent dans deux données VoTable. Cette section d'outils sert donc à convertir le nom d'un attribut en un nouveau nom afin que si on utilise deux données VoTable, ces deux attributs aient le même nom (et donc par exemple, on peut utiliser l'outil *Info to Highlight* sur les deux données et non sur qu'une seule donnée) (pour plus d'info, cf Rapport de stage de Jérôme Desroziers p27-28-29). (*ATTENTION : Si la vue est elle-même la vue de l'outil de zoom, cet outil ne sera pas affiché*).

Voici une description des trois outils de cet section :

- L'outil **Ancien Merge Info** est le nom de l'attribut à remplacer.
- L'outil **New Merge Info** est le nouveau nom de l'attribut.
- L'outil **OK** exécute les modifications de nom d'attribut.

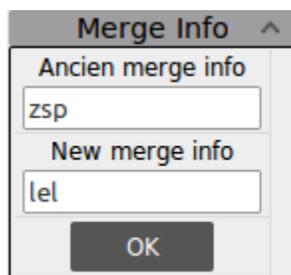


FIGURE 2.12 – Outils de changement de nom d'attribut dans les données

Outils d'affichage

Cette section contient les outils d'affichage de valeur de l'application. Voici les outils associés à cette section :

- L'afficheur **Distance** qui affiche la distance entre deux particules sélectionnés dans la vue. Lors de la sélection (avec la souris) des particules, il est possible en appuyant sur le clavier sur **SHIFT** d'ajouter des particules à la sélection. Dans ce cas, la valeur affichée et la distance de chemin entre tous les points sélectionnés. La distance affichée est calculée en fonction des données réelles (des données chargées en mémoire) et non des données affichées dans la vue (qui ont été normalisées afin d'être contenues dans un cube de 1 x 1 x 1). (*ATTENTION : La valeur affichée n'est pas très précise (avec des tests, elle n'est précise qu'à 5 ou 6 décimales), il ne faut donc pas utiliser cet outil pour avoir des distances exactes, par contre, cet outil peut être utilisé pour avoir des ordres de grandeur ou pour comparer des distances.*)

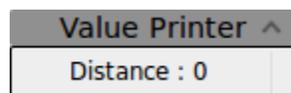


FIGURE 2.13 – Outils d'affichage de valeur d'une vue

Outils de zoom

Cette section contient les paramètres de l'outil de zoom. (*ATTENTION : Si la vue est elle-même la vue de l'outil de zoom, ces outils ne seront pas affichés.*)

- L'outil **Enable Zoom** permet d'utiliser l'outil de zoom
- Les outils **Zoom X**, **Zoom Y** et **Zoom Z** permettent de régler la position du centre de l'outil de zoom.
- L'outil **Zoom Size** permet de régler la taille de l'outil de zoom.

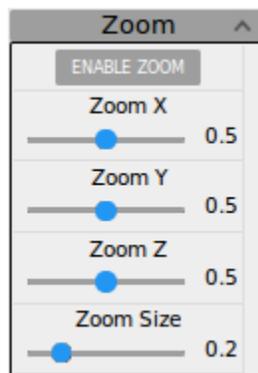


FIGURE 2.14 – Outils de zoom d'une vue

Outils d'affichage des infos de debug

Cette section contient un affichage des données de debug, c'est-à-dire le nombre de primitives affiché dans la scène THREE.js.

- L'afficheur **Calls** correspond au nombre d'appel de la fonction THREE.js *drawCalls*.
- L'afficheur **Faces** correspond au nombre de polygones affichés dans la scène THREE.js.
- L'afficheur **Points** correspond au nombre de particules affichées dans la scène THREE.js.
- L'afficheur **Vertices** correspond au nombre d'appel de vertices contenus dans la scène THREE.js.

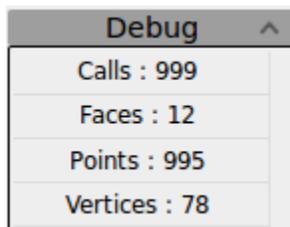


FIGURE 2.15 – Outils d'affichage des infos de debug

Outils d'affichage des infos de la caméra

- Les afficheurs **Position X**, **Position Y** et **Position Z** correspondent à la position de la caméra dans la scène THREE.js.
- Les afficheurs **Angle X**, **Angle Y** et **Angle Z** correspondent aux angles en degré de la caméra autour des axes X,Y et Z dans la scène THREE.js.
- Le bouton *Save Camera Pos.* sauvegarde la position courante de la caméra.
- L'outil **Go to Save** permet de revenir à une position de la caméra précédemment sauvegarder.

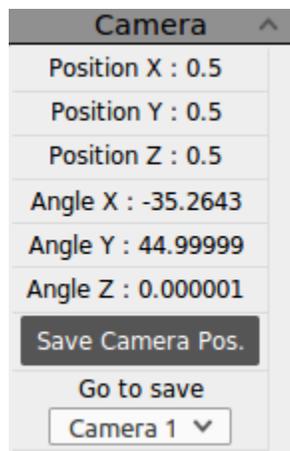


FIGURE 2.16 – Outils d'affichage des infos de la caméra

Chapitre 3

Description technique des barres d'outils et des outils en eux-même

3.1 Introduction

La barre d'outils et les outils associés sont écrits comme le reste de l'application en Javascript sans bibliothèque extérieure. Ce code Javascript, présent dans le dossier *js/Toolbar* génère une fois utilisé dans l'application le code HTML et CSS des barres d'outils et des outils en eux-mêmes. Cela permet, grâce à des appels de fonctions d'ajouter des outils aux barres d'outils facilement.

Voici une succincte description des fichiers présents dans le dossier *js/Toolbar* :

buttonTool.js : Fichier contenant l'outil bouton.

caseTool.js : Fichier contenant l'outil de sélection de données dans les vues.

colorTool.js : Fichier contenant l'outil de choix de couleur.

elementTool.js : (*Classe Abstraite*) Fichier contenant l'outil de base (Classe mère de tous les outils).

folderTool.js : Fichier contenant l'outil qui contient des outils dans la barre d'outils de la vue.

inputTool.js : (*Classe Abstraite*) Fichier contenant l'outil basé sur la balise HTML `<input>` (Classe mère des outils contenant la balise `<input>`).

listTool.js : Fichier contenant l'outil menu déroulant.

printTool.js : Fichier contenant l'outil d'affichage de nombre.

rangeTool.js : Fichier contenant l'outil de saisie de nombre par le biais d'une jauge.

switchTool.js : Fichier contenant l'outil bouton ON/OFF.

tabContainer.js : Fichier contenant l'outil contenant les onglets.

tabTool.js : Fichier contenant l'outil de création d'onglet.

textTool.js : Fichier contenant l'outil de saisie de texte.

toolbar.js : Fichier contenant la barre d'outils et contenant aussi le CSS de tous les outils et de la barre d'outils.

3.2 Description technique

Pour chaque outil et élément de la barre d'outils seront décrits un plan de la structure du DOM HTML de ces outils ainsi qu'un exemple d'utilisation.

3.2.1 Barre d'outils

La barre d'outils est contenue dans le fichier *toolbar.js*. Elle est faite de telle sorte qu'elle puisse contenir beaucoup d'outils tout en gardant une dimension fixe. Si il n'est pas possible d'afficher tous les outils en même temps dans la barre d'outils, un ascenseur se met en place (à droite pour les barres d'outils verticales et en bas pour les barres d'outils horizontales). La classe à utiliser pour intégrer une barre d'outils est la classe *Toolbar*.

Description de l'HTML

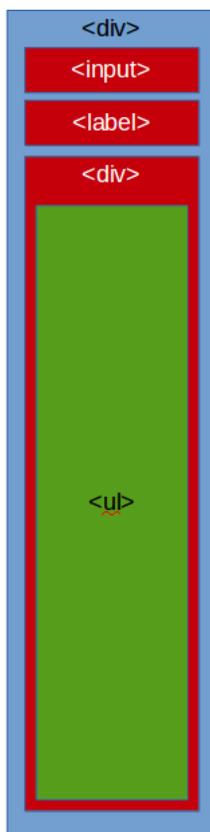


FIGURE 3.1 – Schéma HTML de la Toolbar

- La balise `<div>` (nommé dans la classe `Toolbar` `this.toolbar_volet`) en bleu contient tous les éléments de la barre d'outils.
- La balise `<input>` (nommé dans la classe `Toolbar` `this.toolbar_button`) est le bouton pour ouvrir/fermer la barre d'outils.
- La balise `<label>` (nommé dans la classe `Toolbar` `this.toolbar_button_label`) est le label associé au bouton précédent.
- La balise `<div>` (nommé dans la classe `Toolbar` `this.toolbar_main`) en rouge est un container de barre d'outils.
- La balise `` (nommé dans la classe `Toolbar` `this.toolbar_contener`) contient tous les outils de la barre d'outils.

Utilisation

Pour ajouter une barre d'outils dans une application, il suffit d'appeler le constructeur de la classe `Toolbar` et ensuite de paramétrer l'objet grâce aux méthodes présentes dans la classe `Toolbar` (Pour plus de détails au sujet de ces méthodes, il faut se référer à la Javadoc de l'application).

Créons et affichons une barre d'outils et ensuite la paramétrer :

```

1 var toolbar = new Toolbar('tdata',parentHTML,200,90); //Creation de la barre d'outil avec l'id "
   tdata", comme parent "parentHTML" et comme largeur 200 pixel et hauteur 90 pixel
2 toolbar.addButtonClose(false); //On enleve le bouton qui permet d'ouvrir et fermer la barre d'
   outils
3 toolbar.largueurInPourcent(35); //On definit la largeur de la barre d'outil a 35% de la fenetre
4 toolbar.setVertical(false); //La barre d'outil est horizontale
5 toolbar.noPadding(true); //On ne met pas de marge interieur a la barre d'outils
6 toolbar.noScroll(true); //On n'affiche pas le scroll de la barre d'outils

```

3.2.2 Outils abstraits (ElementTool et InputTool)

ElementTool

L'outil abstrait *ElementTool* est la base à tous les outils, c'est-à-dire que tous les outils héritent de cet outil abstrait. Cet outil est contenue dans une balise ``, en effet, lors de l'ajout d'un outil dans la barre d'outils, le container de la barre d'outils étant une balise ``, la balise `` est parfaite pour contenir l'outil.

Voici un schéma HTML de l'outil *ElementTool* en lui-même :

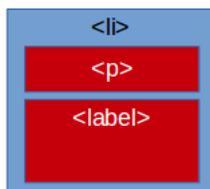


FIGURE 3.2 – Schéma de ElementTool

- La balise `` (nommé dans la classe *ElementTool* *this.main*) contient l'outil en lui-même.
- La balise `<p>` (nommé dans la classe *ElementTool* *this.label*) contient le nom de l'outil.
- La balise `<label>` (nommé dans la classe *ElementTool* *this.element*) contient l'intérieur de l'outil.

InputTool

L'outil abstrait *InputTool* est un outil qui hérite directement de l'outil abstrait *ElementTool*. Cet outil contient une balise `<input>` en plus de l'HTML de l'outil *ElementTool* (en effet, la balise `<input>` est utilisée dans bon nombre d'outil).

Voici un schéma HTML de l'outil *InputTool* en lui-même :

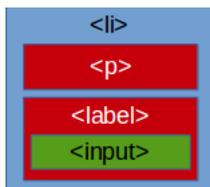


FIGURE 3.3 – Schéma de InputTool

- La balise `` (nommé dans la classe *InputTool* *this.main*) contient l'outil en lui-même.
- La balise `<p>` (nommé dans la classe *InputTool* *this.label*) contient le nom de l'outil.
- La balise `<label>` (nommé dans la classe *InputTool* *this.element*) contient l'intérieur de l'outil.
- La balise `<input>` (nommé dans la classe *InputTool* *this.input*) contient la zone de texte.

3.2.3 Outils de saisie de texte (TextTool)

L'outil *TextTool* sert à saisir du texte. Cet outil est créée par la classe *TextTool* (qui hérite de *InputTool*).



FIGURE 3.4 – L'outil TextTool

Description de l'HTML

L'HTML est le même que l'outil InputTool.

Utilisation

Pour ajouter cet outil dans une barre d'outils, il suffit d'appeler le constructeur de la classe *TextTool* et d'ajouter l'outil à une barre d'outils. De plus, dans notre exemple, on va ajouter un événement qui permettra d'interagir avec l'application. (Pour plus de détails au sujet de ces méthodes, il faut se référer à la Javadoc de l'application).

```
1 var texttool = new TextTool("NOM", "Id", Valeur_Initiale);
2 texttool.addEventListener("change", function (e){/*Evenement*/});
3 toolbar.add(texttool);
```

3.2.4 Outils de création d'onglet (TabTool et TabContainerTool)

L'outil de création d'onglet permet de créer des containers stockant des outils. Cet outil utilise deux classes (*TabContainerTool* et *TabTool*). En effet, la classe *TabContainerTool* sert à contenir les onglets en eux-mêmes et la classe *TabTool* représente les onglets en eux-même.



FIGURE 3.5 – L'outil onglet

Description de l'HTML

Classe TabContainerTool : L'outil *TabContainerTool* sert à contenir les onglets. Voici un schéma du conteneur sans onglet :

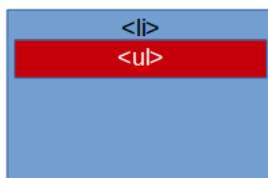


FIGURE 3.6 – Schéma de TabContainerTool sans onglet

- La balise `` (nommé dans la classe *TabContainerTool* `this.main`) contient l'outil en lui-même.
- La balise `` (nommé dans la classe *TabContainerTool* `this.contener_label`) contient la liste des labels associés aux onglets.

Classe TabTool : L'outil *TabContainerTool* est l'onglet en lui-même. Voici un schéma de l'onglet :



FIGURE 3.7 – Schéma de TabTool seul

- La balise `` (nommé dans la classe *TabTool* `this.label`) contient le nom et le lien de l'onglet.

- La balise `<a>` contient lien de l'onglet (c'est grâce à ce lien que l'on peut ouvrir un onglet, en effet, ce lien est attaché à un événement de type `onClick` qui permet d'ouvrir l'onglet et de fermer tous les autres onglets du conteneur d'onglet père).
- La balise `` (nommé dans la classe `TabTool` `this.content`) contient la liste de outils inclus dans l'onglet.

Ajout d'un onglet : Lors de l'ajout d'un onglet dans le conteneur d'onglet, deux actions se produisent :

- La balise `` (qui est le label de l'onglet) de l'objet `TabTool` est inclus dans la balise `` de l'objet `TabContainerTool` (qui contient tous les labels des onglets).
- La balise `` (qui contient les outils de l'onglet) de l'objet `TabTool` est inclus dans la balise `` de l'objet `TabContainerTool`.

Voici donc la schéma complet HTML de l'outil `TabContainerTool` avec un onglet à l'intérieur :

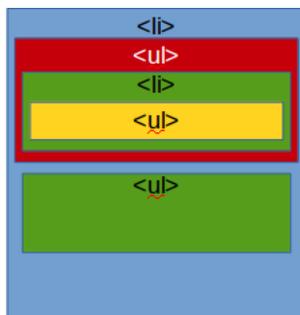


FIGURE 3.8 – Schéma d'outil `TabContainerTool` avec un onglet à l'intérieur

Utilisation

Pour ajouter cet outil dans une barre d'outils, il suffit d'appeler le constructeur de la classe `TabContainerTool` et d'ajouter l'outil à une barre d'outils. De plus, dans notre exemple, on va ajouter deux onglets au conteneur d'onglet ainsi qu'un outil dans le premier onglet (Pour plus de détails au sujet de ces méthodes, il faut se référer à la Javadoc de l'application).

```

1 var onglet_conteneur = new TabContainerTool('Id');
2 toolbar.add(onglet_conteneur);
3
4 var onglet1 = new TabTool("Nom onglet 1", 'id onglet 1', 65); //Ici, 65 est la hauteur en pixel du
   conteneur d'outils de l'onglet
5 onglet_conteneur.addTab(onglet1);
6
7 var onglet2 = new TabTool("Nom onglet 2", 'id onglet 2', 65); //Ici, 65 est la hauteur en pixel du
   conteneur d'outils de l'onglet
8 onglet_conteneur.addTab(onglet2);
9
10 var outils = new ButtonTool("Bouton", "id_bouton");
11 onglet1.add(outils); //On ajoute l'outils au 1er onglet

```

3.2.5 Outils bouton ON/OFF (SwitchTool)

L'outil `SwitchTool` sert à activer ou désactiver un paramètre. Cet outil est créé par la classe `SwitchTool` (qui hérite de `ElementTool`).



FIGURE 3.9 – L'outil `SwitchTool` en mode activé

Description de l'HTML

L'outil *SwitchTool* est basé sur l'outil *ElementTool*. Même si l'outil contient une balise `<input>`, il n'est pas basé sur *InputTool* car la construction de cette balise est assez spéciale (elle se fait dans la classe *SwitchTool* avec un appel de la fonction Javascript *innerHTML*).

Voici un schéma HTML de l'outil *SwitchTool* en lui-même :

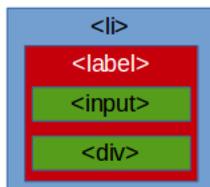


FIGURE 3.10 – Schéma de SwitchTool

- La balise `` (nommé dans la classe *SwitchTool* *this.main*) contient l'outil en lui-même.
- La balise `<label>` (nommé dans la classe *SwitchTool* *this.element*) contient l'intérieur de l'outil.
- La balise `<input>` contient une checkbox (qui n'est pas affiché en temps que tel dans l'application).
- La balise `<div>` (nommé dans la classe *SwitchTool* *this.div*) contient le nom de l'outil.

Utilisation

Pour ajouter cet outil dans une barre d'outils, il suffit d'appeler le constructeur de la classe *SwitchTool* et d'ajouter l'outil à une barre d'outils. De plus, dans notre exemple, on va ajouter un événement qui permettra d'interagir avec l'application. (Pour plus de détails au sujet de ces méthodes, il faut se référer à la Javadoc de l'application).

```
1 var switch = new SwitchTool("NOM", 'Id');
2 switch.addEventListener('click', function(e) { /* Evenement */ });
3 toolbar.add(switch);
```

3.2.6 Outils de jauge (RangeTool)

L'outil *RangeTool* sert à sélectionner un nombre entre une valeur maximum et minimum. Cet outil est créé par la classe *RangeTool* (qui hérite de *InputTool*). Il affiche à droite du sélecteur la valeur sélectionnée.

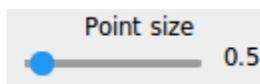


FIGURE 3.11 – L'outil RangeTool

Description de l'HTML

L'outil *RangeTool* est basé sur l'outil *InputTool*.

Voici un schéma HTML de l'outil *RangeTool* en lui-même :

- La balise `` (nommé dans la classe *RangeTool* *this.main*) contient l'outil en lui-même.
- La balise `<p>` (nommé dans la classe *RangeTool* *this.label*) contient le nom de l'outil.
- La balise `<label>` (nommé dans la classe *RangeTool* *this.element*) contient l'intérieur de l'outil.
- La balise `<input>` (nommé dans la classe *RangeTool* *this.input*) l'outil range.
- La balise `<p>` (nommé dans la classe *RangeTool* *this.print_value*) contient l'affichage de la valeur sélectionnée.

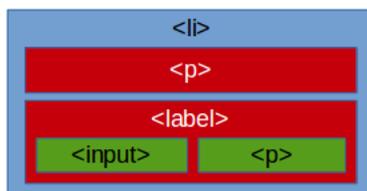


FIGURE 3.12 – Schéma de RangeTool

Utilisation

Pour ajouter cet outil dans une barre d'outils, il suffit d'appeler le constructeur de la classe *RangeTool* et d'ajouter l'outil à une barre d'outils. De plus, dans notre exemple, on va ajouter un événement qui permettra d'interagir avec l'application. (Pour plus de détails au sujet de ces méthodes, il faut se référer à la Javadoc de l'application).

```

1 var range = new RangeTool("NOM", 'Id', 0, 1, 0.01, 0.5); //ici, 0 represente la borne min, 1 la borne
  max, 0.01 le pas et 0.5 la valeur initiale du range
2 range.addEventListener('click', function(e) { /* Evenement */ });
3 toolbar.add(range);

```

3.2.7 Outils d'affichage de nombre (PrintTool)

L'outil *PrintTool* sert à afficher un nombre. Cet outil est créé par la classe *PrintTool* (qui hérite de *ElementToolbar*).

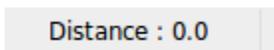


FIGURE 3.13 – L'outil PrintTool

Description de l'HTML

L'outil *PrintTool* est basé sur l'outil *ElementTool*.

Voici un schéma HTML de l'outil *PrintTool* en lui-même :

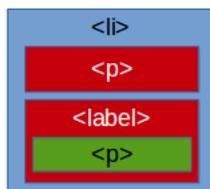


FIGURE 3.14 – Schéma de PrintTool

- La balise **** (nommé dans la classe *PrintTool* *this.main*) contient l'outil en lui-même.
- La balise **<p>** en rouge (nommé dans la classe *PrintTool* *this.label*) contient le nom de l'outil.
- La balise **<label>** (nommé dans la classe *PrintTool* *this.element*) contient l'intérieur de l'outil.
- La balise **<p>** en vert (nommé dans la classe *PrintTool* *this.printer*) contient l'affichage de la valeur sélectionnée.

Utilisation

Pour ajouter cet outil dans une barre d'outils, il suffit d'appeler le constructeur de la classe *PrintTool* et d'ajouter l'outil à une barre d'outils. De plus, dans notre exemple, on va ajouter un événement qui permettra d'interagir avec l'application. (Pour plus de détails au sujet de ces méthodes, il faut se référer à la Javadoc de l'application).

```

1 var print = new PrintTool("Nom", 'Id', 12); //On affiche le nombre avec 12 caracteres
2 print.addPrint(1); //Ajouter 1 au compteur (on a donc 1 qui est affiche)
3 print.addPrint(2); //Ajouter 2 au compteur (on a donc 3 qui est affiche)
4 print.reset(); //Reset le compteur (on a donc 0 qui est affiche)
5 toolbar.add(print);

```

3.2.8 Outils de menu déroulant (ListTool)

L'outil *ListTool* sert à afficher un menu déroulant . Cet outil est crée par la classe *ListTool* (qui hérite de *ElementToolbar*).

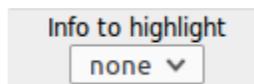


FIGURE 3.15 – L'outil ListTool

Description de l'HTML

L'outil *ListTool* est basé sur l'outil *ElementTool*.

Voici un schéma HTML de l'outil *ListTool* en lui-même :

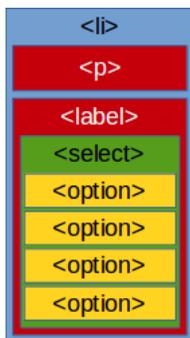


FIGURE 3.16 – Schéma de ListTool

- La balise `` (nommé dans la classe *ListTool* *this.main*) contient l'outil en lui-même.
- La balise `<p>` (nommé dans la classe *ListTool* *this.label*) contient le nom de l'outil.
- La balise `<label>` (nommé dans la classe *ListTool* *this.element*) contient l'intérieur de l'outil.
- La balise `<select>` (nommé dans la classe *ListTool* *this.list*) contient les options de la liste déroulante.
- Les balises `<option>` contiennent sont les options du menu.

Utilisation

Pour ajouter cet outil dans une barre d'outils, il suffit d'appeler le constructeur de la classe *ListTool* et d'ajouter l'outil à une barre d'outils. De plus, dans notre exemple, on va ajouter un événement qui permettra d'interagir avec l'application. (Pour plus de détails au sujet de ces méthodes, il faut se référer à la Javadoc de l'application).

```

1 var list = new ListTool("Nom", "Id");
2 list.addEventListener('change', function(e) { /* Evenement */ });
3 list.addList("Option 1", 0); //Option 1 est le nom affiche et 0 est l'id de l'element
4 list.addList("Option 2", 1);
5 list.addList("Option 3", 2);
6 list.addList("Option 4", 3);
7 toolbar.add(list);

```

3.2.9 Conteneur d'outils (FolderTool)

L'outil *FolderTool* sert à contenir des outils dans un même espace. Il est possible, en cliquant sur le libellé de l'outil de caché le contenu d'un *FolderTool* (et de le ré-afficher par la même action). Cet outil est crée par la classe *FolderTool*.

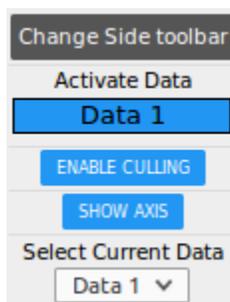


FIGURE 3.17 – L'outil FolderTool

Description de l'HTML

Voici un schéma HTML de l'outil *FolderTool* en lui-même :

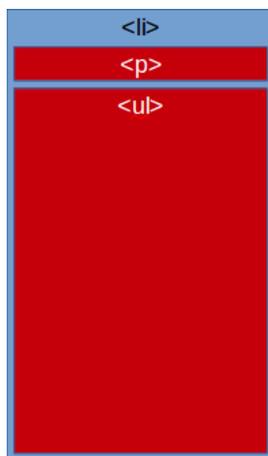


FIGURE 3.18 – Schéma de FolderTool

- La balise `` (nommé dans la classe *FolderTool* `this.main`) contient l'outil en lui-même.
- La balise `<p>` (nommé dans la classe *FolderTool* `this.label`) contient le nom de l'outil.
- La balise `` (nommé dans la classe *FolderTool* `this.element`) est le conteneur qui va contenir les outils qui seront à l'intérieur.

Utilisation

Pour ajouter cet outil dans une barre d'outil, il suffit d'appeler le constructeur de la classe *FolderTool* et d'ajouter l'outil à une barre d'outils. De plus, dans notre exemple, on va ajouter un outil dans le conteneur. (Pour plus de détails au sujet de ces méthodes, il faut se référer à la Javadoc de l'application).

```

1 var folder = new FolderTool("NOM", "Id");
2 toolbar.add(folder);
3
4 var color = new ColorTool("NOM", "Id");
5 folder.add(color); //On ajoute l'outil color dans le conteneur

```

3.2.10 Outils de sélection de couleur (ColorTool)

L'outil *ColorTool* sert à saisir une couleur. Cet outil est créé par la classe *ColorTool* (qui hérite de *InputTool*).

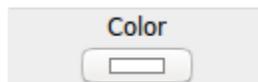


FIGURE 3.19 – L'outil ColorTool

Description de l'HTML

L'HTML est le même que l'outil *InputTool*.

Utilisation

Pour ajouter cet outil dans une barre d'outil, il suffit d'appeler le constructeur de la classe *ColorTool* et d'ajouter l'outil à une barre d'outils. De plus, dans notre exemple, on va ajouter un événement qui permettra d'interagir avec l'application. (Pour plus de détails au sujet de ces méthodes, il faut se référer à la Javadoc de l'application).

```
1 var color = new ColorTool("NOM","Id");
2 color.addEventListener("change",function (e){/*Evenement*/});
3 toolbar.add(color);
```

3.2.11 Outils de sélection de données (CaseTool)

L'outil *CaseTool* permet d'activer ou désactiver des données dans une vue. Cet outil est créé par la classe *CaseTool* (qui hérite de *ElementTool*).

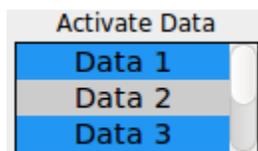


FIGURE 3.20 – L'outil CaseTool

Description de l'HTML

L'outil *CaseTool* est basé sur l'outil *ElementTool*.

Voici un schéma HTML de l'outil *CaseTool* en lui-même :

- La balise `` (nommé dans la classe *CaseTool* *this.main*) contient l'outil en lui-même.
- La balise `<p>` (nommé dans la classe *CaseTool* *this.label*) contient le nom de l'outil.
- La balise `<label>` (nommé dans la classe *CaseTool* *this.element*) contient l'intérieur de l'outil.
- Les balises `<label>` en vert contiennent les cases à sélectionner ou désélectionner.
- Les balises `<input>` contiennent les checkboxes sélectionnés ou désélectionnés.
- Les balises `<div>` contiennent le nom des cases.

Utilisation

Pour ajouter cet outil dans une barre d'outil, il suffit d'appeler le constructeur de la classe *CaseTool* et d'ajouter l'outil à une barre d'outils. De plus, dans notre exemple, on va ajouter un événement qui permettra d'interagir avec l'application. (Pour plus de détails au sujet de ces méthodes, il faut se référer à la Javadoc de l'application).

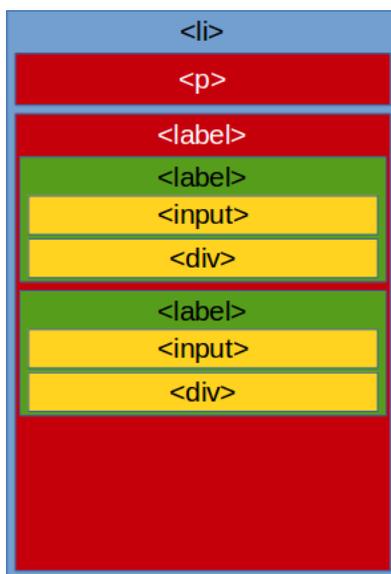


FIGURE 3.21 – Schéma de CaseTool

```

1 var caseTool = new CaseTool("NOM", "Id");
2 caseTool.addCase("Case 1", 1, false); //Case 1 est ici le nom, 1 l'id et false signifie que la case
   n'est pas selectionnee.
3 caseTool.addCase("Case 2", 2, true); //Case 2 est ici le nom, 2 l'id et true signifie que la case
   est selectionnee.
4 caseTool.addEventListener("change", function (e){/*Evenement*/});
5 toolbar.add(caseTool);

```

3.2.12 Outils bouton (ButtonTool)

L'outil *ButtonTool* est un simple bouton. Cet outil est créé par la classe *CaseTool* (qui hérite de *Input*).

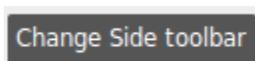


FIGURE 3.22 – L'outil ButtonTool

Description de l'HTML

L'outil *ButtonTool* est basé sur l'outil *InputTool*.

Voici un schéma HTML de l'outil *ButtonTool* en lui-même :

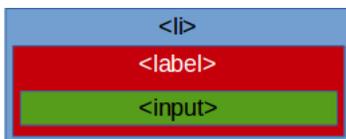


FIGURE 3.23 – Schéma de ButtonTool

- La balise `` (nommé dans la classe *ButtonTool* *this.main*) contient l'outil en lui-même.
- La balise `<label>` (nommé dans la classe *ButtonTool* *this.label*) contient le nom de l'outil.
- La balise `<input>` (nommé dans la classe *ButtonTool* *this.folder*) contient le bouton en lui-même.

Utilisation

Pour ajouter cet outil dans une barre d'outil, il suffit d'appeler le constructeur de la classe *ButtonTool* et d'ajouter l'outil à une barre d'outils. De plus, dans notre exemple, on va ajouter un événement qui permettra d'interagir avec l'application. (Pour plus de détails au sujet de ces méthodes, il faut se référer à la Javadoc de l'application).

```
1 var button = new ButtonTool("NOM", "Id");
2 button.addEventListener("click", function (e){/*Evenement*/});
3 toolbar.add(button);
```

Chapitre 4

A Savoir

- Pour avoir plus de détail au sujet des outils décrit dans la première partie ainsi qu’au fonctionnement de l’application, merci de se référer au rapport de stage de *Arnaud Steinmetz, Pierre Lespignal, Nicolas Buecher, Jérôme Desroziers, Nicolas Adam et Thibault Bouchard*.
- Pour le choix de couleur (ColorTool), cet outil ne marche pas sur les navigateurs Internet Explorer et Microsoft Edge. Sur ces navigateur, une zone de saisie de texte est affichée.
- L’application a été testée majoritairement sur Mozilla Firefox. L’application fonctionne aussi sur Safari.
- Sur les navigateurs Internet Explorer et Microsoft Edge, des erreurs Three.js sont lancées. De plus, il est impossible de charger des données VoTable (Erreur dans le fichier *votable.js*). Pour le choix de couleur (ColorTool), cet outil ne marche pas sur ces navigateurs, une zone de saisie de texte est affichée où il faut écrire la couleur sous forme hexadécimal (par exemple *#ffffff*).