

Devoir individuel

Tableur - Elements de style

ZHOU Peikun
Arpil 1, 2016

Elements de style

J'ai choisi la fonctionnalité "Elements de style",

On souhaite ajouter la possibilité de mettre en forme les cases et leur contenu. L'interface doit présenter 5 boutons : gras, italique, souligné, choix de couleur du texte, choix de couleur du fond de case. Sur pression de l'un de ces boutons, le style de la case sélectionnée est modifié de manière adéquate.

J'ai fini la fonctionnalité basique, pas de premium. J'ai essayé quelques composants mais il ne fonctionne pas bien.

MVC

Le projet utilise MVC model, Model - View - Controller, pour implémenter la fonctionnalité, on a besoin de modifier les fichiers petit à petit, en premier, le view.

- Le view **tableView.js**

J'ai ajouté 5 boutons, gras, italique, souligné, choix de couleur du texte, choix de couleur du fond de case.

par exemple le bouton gras.

```
this.button_gras= document.createElement("button");  
this.button_gras.innerHTML = "gras";  
this.div.appendChild(this.button_gras);
```

- Le model **tableController.js**

Dans ce fichier, on ajoute les fonctions de bouton. Sélectionner une fois une cellule, puis cliquer sur le bouton, nous obtenons la cellule modifiée. Pour sélectionner une cellule, on doit sélectionner son objet *td*, puis faire un changement sur elle. Par exemple la fonction gras,

```
function gras(e) {  
    var td = this._selection;  
    if(!td) return;  
    //form.style.fontWeight = bold;  
    if(td.style.fontWeight == "normal")  
        td.style.fontWeight = "bold";  
    else  
        td.style.fontWeight = "normal";  
};  
  
view.button_gras.addEventListener("click", gras);
```

- Le model **tableModel.js**

Dans ce fichier, on peut ajoute pour implement la fonction de sélection rectangulaire multiple.

Détail

- **Gras, italique et souligné,**

Les fonctions **gras**, *italique*, et souligné, ce sont similaires. En premier, on ajoute le button sur tableView.js ou je utilise description du texte. Puis ajoute la fonction sur tableController.js. Les différents sont comment changer son style, ils sont,

```
td.style.fontWeight = "bold";
td.style.fontStyle = "italic";
td.style.textDecoration = "underline";
```

J'ai ajoute la logique if / else pour récupération.

enfin, on ajoute le clickevent,

```
view.button_gras.addEventListener("click", gras);
```

- Couleur

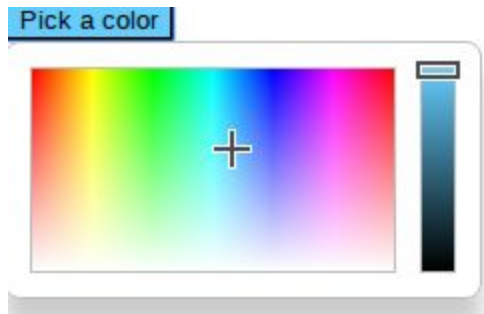
Pour la fonction couleur de texte et du fond de base, j'ai implemente couleur fixe en premier fois, changer le style de object td,

```
td.style.color="#ab2567";
td.style.backgroundColor = "red";
```

apredosnecy page.html

	italique	souligne	choix de couleur du texte	choix du couleur du fond de case					
	D	E	F	G	H	I	J		
		OUI							

évidemment, c'est pas optionnel. On a besoin de plus de couleur.



J'ai choisi un couleur picker composant [jscolor](#) d'abord, c'est plus facile. On change la propriété de button comme,

```
this.button_textCouleur = document.createElement("button");
this.button_textCouleur.innerHTML = "choix de couleur du texte";
this.button_textCouleur.classList.add('jscolor {valueElement:null,value:'000000'}');
this.div.appendChild(this.button_textCouleur);
```

puis on a la couleur valeur dans la class de button, passe ce propriété a tableController.js, ça marche. Il y a problem de changement de class de button, donc j'ai essayé l'autre method, on utilise **input couleur**

```
this.textColor = document.createElement("input");
this.textColor.type = "color";
this.textColor.id = "textColor";
this.textColor.value = "#000000";
this.div.appendChild(this.textColor);

this.bgColor = document.createElement("input");
this.bgColor.type = "color";
this.bgColor.id = "bgColor";
this.bgColor.value = "#CC0000";
this.div.appendChild(this.bgColor);
```

tableController.js

```
// text's color
function textCouleur(e) {
  var td = this._selection;
  if(!td) return;
  //td.style.color="#ab2567";
  var couleur;
  couleur = (document.getElementById("textColor")).value;
  td.style.color = couleur;
};

// case's color
function caseCouleur(e) {
  var td = this._selection;
  if(!td) return;
  //td.style.backgroundColor = "red";
  var couleur;
```

```

couleur = (document.getElementById("bgColor")).value;
td.style.backgroundColor = couleur;
};

```

```

view.textColor.addEventListener("change", textCouleur);
view.bgColor.addEventListener("change", caseCouleur);

```

Test

J'ai amélioré les boutons.

<input type="text" value="=OUI"/> <input type="button" value="✓"/> <input type="button" value="B"/> <input type="button" value="/"/> <input type="button" value="U"/> <input type="button" value="■"/> <input type="button" value="■"/>					
	A	B	C	D	E
1					
2			OUI		
3					

Il y a quelques exemples de gras, italique, souligné, choix de couleur du texte, choix de couleur du fond de case.

<input type="text"/> <input type="button" value="✓"/> <input type="button" value="B"/> <input type="button" value="/"/> <input type="button" value="U"/> <input type="button" value="■"/> <input type="button" value="■"/>					
	A	B	C	D	E
1		GRAS	ITALIC	UNDER	
2		TEXT	OUI		
3					

La sélection de couleur,

The screenshot shows a spreadsheet application with a color selection dialog box open. The spreadsheet has columns A-K and rows 1-10. Cell B1 contains 'GRAS', C1 contains 'ITALIC', and D1 contains 'UNDER'. Cell B2 contains 'TEXT' in red. Cell D2 is highlighted in green. The color selection dialog box is open, showing a color gradient and a color bar. The selected color is #14c242.