WebGL - TP1 Premiers pas

LP DAWIN

2015-2016

1 Page HTML de base

Pour utiliser WebGL, vous avez besoin d'une page HTML à ouvrir dans un navigateur. Le squelette de la page comprend simplement une balise `canvas` (avec un `id pour le retrouver plus tard`) en plus des balises `html` et `body`, ainsi qu'un script `javascript` externe pour le code :

```
<html>
    <body onload='main()'>
        <canvas id='dawin-webgl' width=800 height=800>
            Achete-toi un vrai navigateur avec WebGL !
        </canvas>
        <script src='tp1.js'></script>
    </body>
</html>
```

Dans le fichier `tp1.js`, pensez à définir la fonction `main` qui sera appelée lorsque la page sera chargée.

2 Création du contexte

La première étape pour utiliser WebGL est de récupérer le `contexte` sur lequel on va pouvoir agir. Pour cela, il faut tout d'abord récupérer le canvas, puis le contexte WebGL attaché au canvas et vérifier qu'il a été reçu correctement :

```
var canvas = document.getElementById('dawin-webgl');
var gl = canvas.getContext('webgl');
if (!gl) {
    console.log('ERREUR : Echec du chargement du contexte !');
    return;
}
```

3 Effacer la zone de dessin

Le programme WebGL le plus simple qui soit se contente d'effacer la zone de dessin. Cela se fait en deux étapes :

- Définir une couleur pour l'effacement avec `gl.clearColor` Cette fonction prend 4 arguments : les valeurs des canaux rouge, vert, bleu et alpha (transparence).
- L'effacement lui-même avec la fonction `gl.clear`. Son unique argument est le buffer à effacer. Ici, nous voulons effacer le buffer de couleur, l'argument est donc `gl.COLOR_BUFFER_BIT`.

A vous de jouer :

1. Effacez la zone de dessin avec du noir. Quelles sont les valeurs des differends canaux ?
2. Changez de couleur d'effacement. Testez-en plusieures. Trouvez notament le blanc, le jaune, le magenta et le cyan.
3. Changez la valeur du canal alpha. Quel effet cela a-t-il ? Expérimentez avec différentes couleurs de fond pour le `body` (attribut `bgcolor`). Vous pouvez aussi mettre du texte ou un `div` sous le canvas.

4 La transparence

Par défaut, WebGL attend que les valeurs RGB soient déjà multipliées par la valeur du canal alpha. Pour obtenir une gestion de la transparence plus instinctive, il faut ajouter une option lors de la création du contexte :

```
var gl = canvas.getContext('webgl', {'premultipliedAlpha': false});
```

1. Rééssayez de jouer sur la transparence et constatez la différence.