

WebGL TP5

LP DAWIN

2015-2016

1 Un Carré

WebGL ne dispose pas de primitive plus complexe qu'un triangle. Quelle que soit la forme à dessiner, il faut donc la diviser en triangles. Pour obtenir un carré, on dessine donc deux triangles qui partagent une arête.

1. Faire un programme qui dessine un carré en utilisant la primitive `TRIANGLES`.
2. Vous pouvez remarquer que deux des sommets sont dupliqués. Remédiez à ce problème en utilisant la primitive `TRIANGLE_STRIP`.

2 Textures

Les textures sont des images chargées dans la mémoire de la carte graphique, que l'on peut ensuite appliquer sur une primitive. Pour appliquer une texture à une primitive, on utilise un nouvel attribut (donc des données définies pour chaque sommet) : mes coordonnées de texture. Cette nouvelle information permet de décider quel pixel de la texture (a.k.a. texel) sera utilisé comme pixel à l'écran. Ils possèdent deux dimensions, qui varient de 0 à 1 quelle que soit la taille effective de l'image.

2.1 Charger une texture

Le chargement d'une texture se fait en deux étapes : d'abord, charger l'image, puis la transférer sur la carte graphique.

1. Voici le code pour charger l'image :

```
var texture = gl.createTexture()
var image = new Image()
image.onload = function() {
    gl.bindTexture(gl.TEXTURE_2D, texture);
    gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image)
    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MAG_FILTER, gl.LINEAR);
    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR);
    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_WRAP_S, gl.CLAMP_TO_EDGE);
    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_WRAP_T, gl.CLAMP_TO_EDGE);
    gl.bindTexture(gl.TEXTURE_2D, null)
};
image.src = urlOfMyImage;
```

2.2 Dessiner avec une texture

Le but de cette partie est d'appliquer une texture sur le carré dessiné au premier exercice.

2.2.1 Les coordonnées

1. Ajoutez un attribut et son buffer pour contenir les coordonnées de texture.
2. Passez les coordonnées au fragment shader. Pour vérifier que les coordonnées sont correctes, modifiez la couleur en fonction de ces coordonnées (par exemple `vec4(texcoord.x, texcoord.y, 0.0, 1.0)`)

2.2.2 La texture

Dans la fonction de dessin :

1. Il faut binder la texture sur un 'texture unit', un emplacement prédéfini pour que le shader puisse accéder à la texture. Pour cela, activer la texture unit 0 avec `gl.activeTexture(gl.TEXTURE0)`, puis binder votre texture.
2. Le fragment shader lit la texture à partir d'un uniform d'un type particulier : `sampler2D`. En ajouter un pour votre texture. La valeur à donner à cet uniform est 0. Utilisez pour cela la fonction `gl.uniform1i`.
3. Dans le shader, la fonction `texture2D(sampler, coords)` vous permet d'obtenir la couleur à appliquer.

3 Grille texturée

1. Dessinez une grille carré de 10*10 et plaquez y une texture. Attention : les coordonnées de texture doivent être définis pour tous les sommets.
2. Sur la même grille, faites en sorte que la texture se répète (vous devrez changer les paramètres 'TEXTURE_WRAP').
3. Appliquez une animation sur les sommets de la grille.
4. Faites défiler la texture de droite à gauche (sans changer le contenu des buffers).