

TP 1 WebGL : Hello CourbesParametriques!

Avec WebGL un simple navigateur Internet devient un environnement hors du commun pour créer et animer en temps réel des scènes 3D pouvant être à la fois complexes et réalistes.

Le but de WebGL est d'utiliser l'infographie comme outil aux multiples facettes pour le développement et la conception innovante de sites Web.

1 Pourquoi WebGL ?

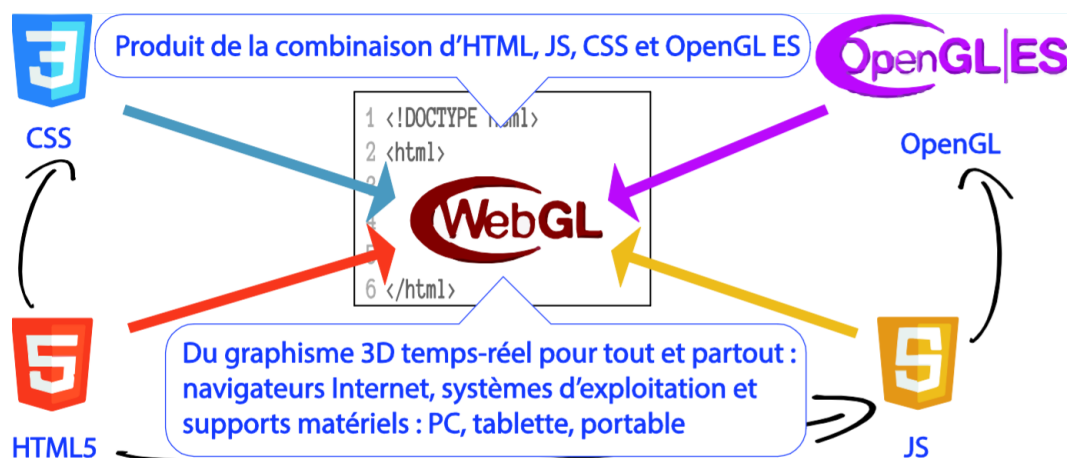
- Outil de visualisation 3D temps-réel
- Développement pur en Infographie 3D adapté au contexte Internet
- API basée sur la norme HTML5 et le langage JavaScript (JS)
- Jeux vidéo, visualisation d'environnements virtuels, simulation
- Gratuit
- Pas de plugin, ni compilateurs, ni bibliothèques annexes
- Multi-OS
- Navigateurs : Firefox 4, Chrome 9, Safari 5.1, Opera 12, Microsoft Internet Explorer (IE) 11.

Pratique, complet, portable, utile et à la pointe de la technologie

Contraintes :

- Carte graphique qui permet d'utiliser OpenGL Shading Language (GLSL) → Faux problème
- Changement de la façon de programmer (mélange : JS, GLSL, HTML, CSS)
- Certaines incompatibilités d' IE®

2 D'où vient WebGL ?



Avec WebGL rien n'est précompilé : les parties HTML et JS sont interprétées et le code GLSL est compilé à la volée (*at runtime*) par la carte graphique lors de l'exécution.

3 WebGL et infographie

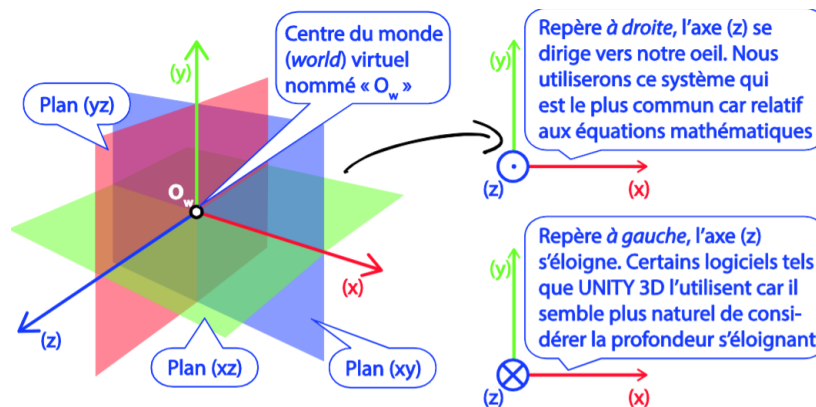
WebGL sans infographie, c'est un peu une coquille vide, sans forme, ni couleur.

Mais qu'est-ce que l'infographie ?

Le terme "infographie", déposé officiellement par la société Benson dès 1974, désigne tout graphisme produit par ordinateur. Il est formé par les mots "informatique" et "graphie". Ce dernier est la représentation écrite de tout caractère. L'infographiste, à l'origine, était ingénieur informaticien chargé d'implémenter les outils destinés aux graphistes. Ces derniers se sont emparés de l'appellation et, depuis, la définition du terme est un peu confuse.

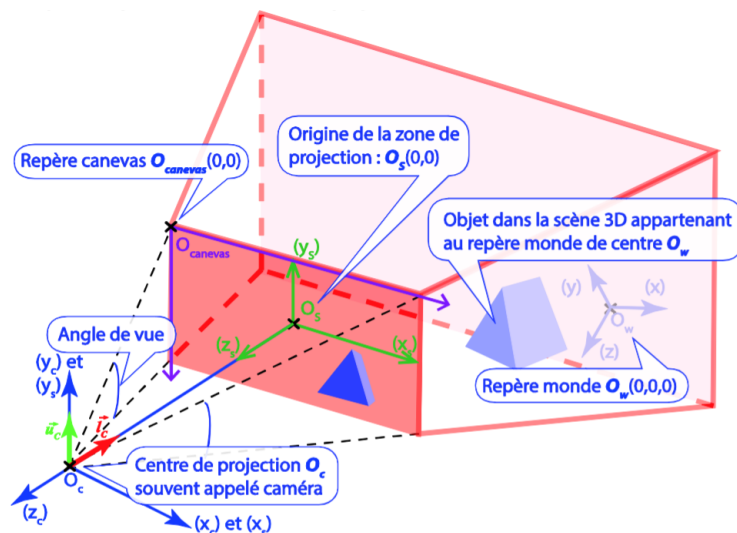
Notions fondamentales en infographie :

- La **scène** : caméra mobile et scène fixe ou objet mobile et caméra fixe

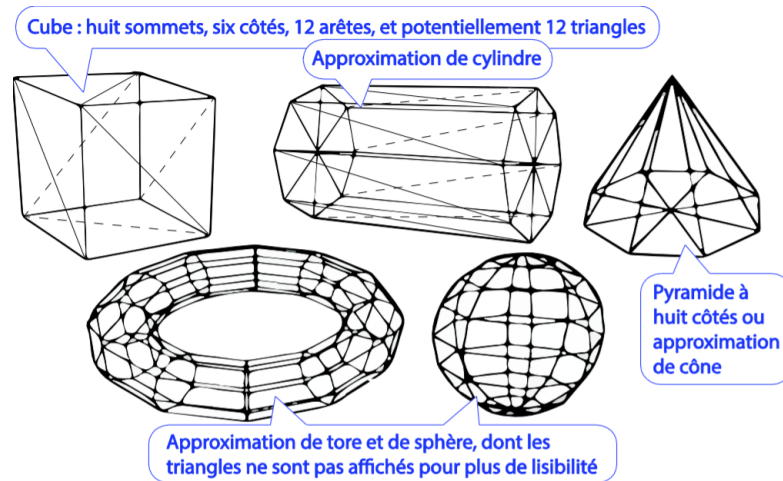


- La **caméra** : permet de définir où l'utilisateur est situé et de quelle manière il est orienté.

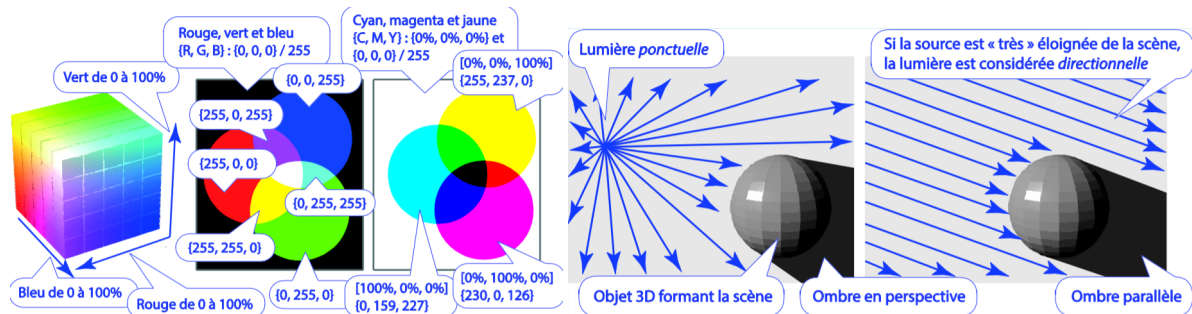
En WebGL, la caméra est représentée par deux matrices. La première définit la position et l'orientation de la caméra (matrice de vue). La deuxième (matrice de projection) définit la transformation géométrique permettant le passage d'une scène en 3D à un viewport en 2D.



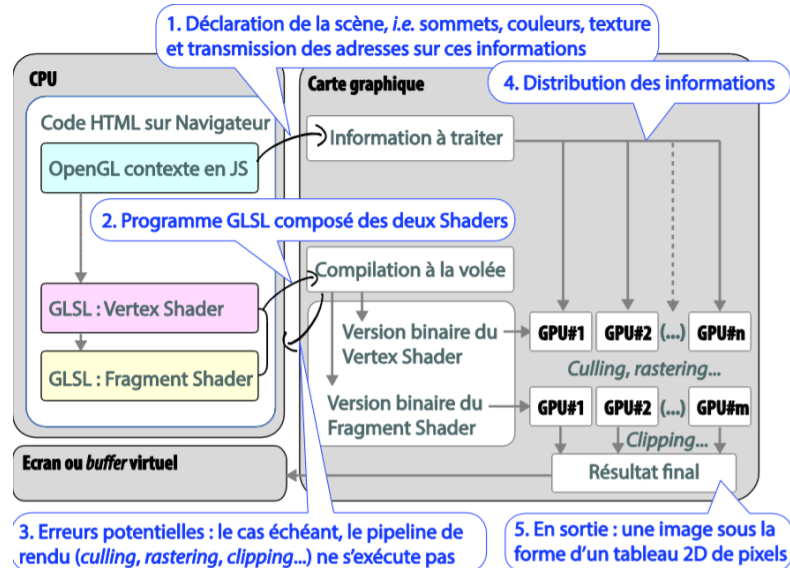
- **Modélisation géométrique** : outils permettant de définir un objet (sa surface), qui est une forme, avec des primitives (point, segment, cercle, polygone, cube, sphere, cone ...). En infographie "tout est triangle".



- **Rendu graphique** : couleurs et lumières (ambiante, ponctuelle et directionnelle)

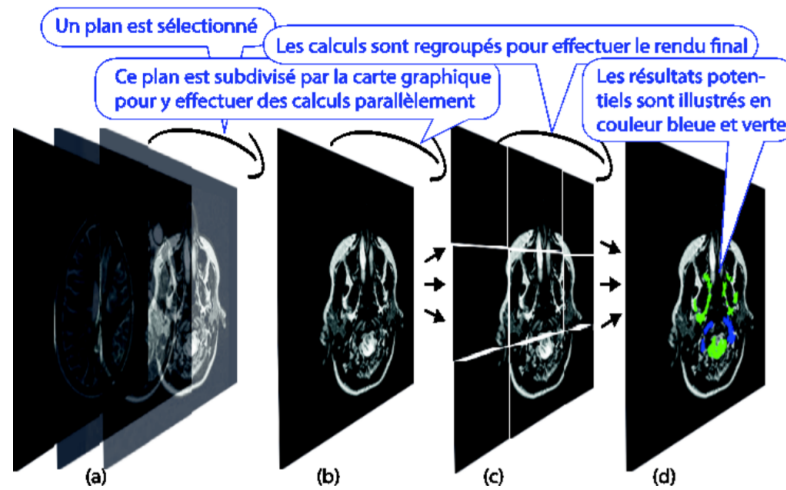


- **Animation et déformation** : matrices de projection et de transformations
- **Architecture**



Ci-dessous un exemple de traitement de l'information entre CPU et GPUs.

En (a) les données 3D d'un IRM, en (b) une tranche d'information (donc 2D) est sélectionnée et transmise à la carte graphique via GLSL. Ensuite (c) la carte graphique s'occupe de partager le travail entre GPUs et (d) regrouper l'ensemble des résultats.



Développer en WebGL "pur" (sans utiliser de bibliothèques de hauts niveaux permettant un nombre d'abstractions) implique d'implémenter un programme GLSL (OpenGL Shading Language) composé de deux *Shaders*, qui sont loin d'être des concepts de type *plug & play*: Vertex Shader qui place les points dans l'espace de la caméra et le Fragment Shader qui colorie les pixels de la scène projetée à l'écran. Cela permet de créer des effets réalistes (par exemple de lumière sur des matériaux composites) et des effets de type NPR (non-photo realistic rendering, comme les simulations de peintures artistiques ou de dessins animés).

Comme souvent, à grand pouvoir, grand devoir !

4 Hello WebGL!

4.1 Environnement d'édition

- **Notepad++** : sources HTML et JS avec code couleur + possibilité de tester le programme à la volée via 'Ctrl-Alt-Shift-X' pour Firefox
- **Sublime Text** : code couleur pour JS + plugins JSHint et GLSL Syntax Highlighting

4.2 Quelques liens

- Site officiel : <http://www.khronos.org/webgl/>
- Tutorial et informations générales :
 - <https://webglfundamentals.org/>
 - https://developer.mozilla.org/fr/docs/Web/API/WebGL_API/Tutorial/Getting_started_with_WebGL
- Démon (parmi d'autres) : <http://www.chromeexperiments.com/webgl>

4.3 Hello CourbesParametriques!

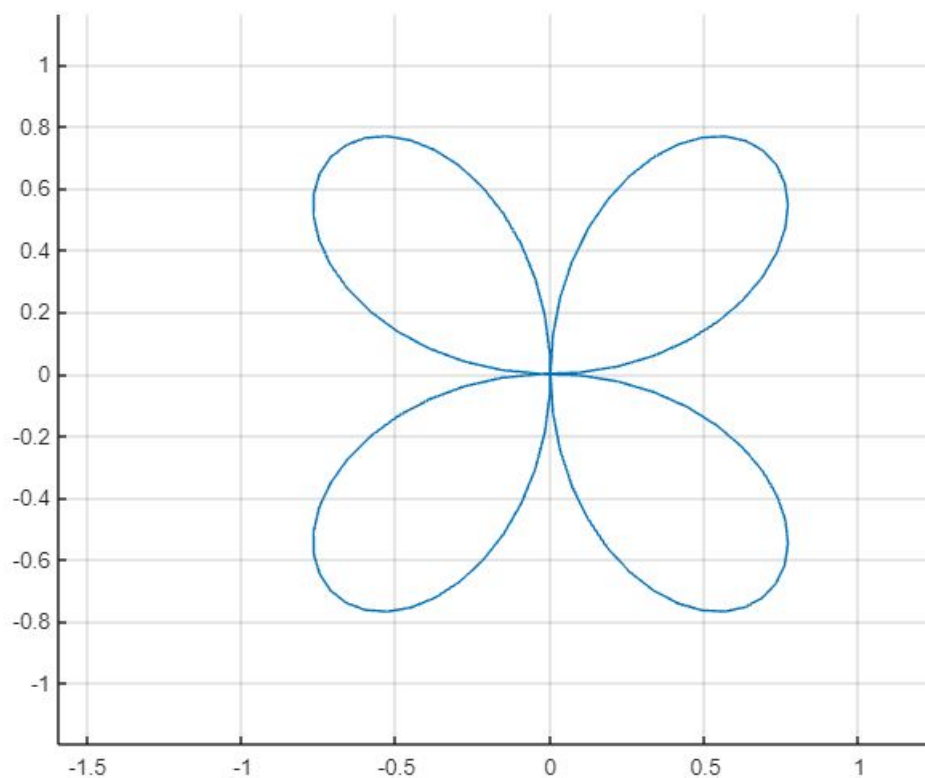
Définir votre premier programme en WebGL qui permet de tracer une courbe paramétrique.

Tracer le cercle

$$\begin{cases} x(t) = 4 \cos(t) \\ y(t) = 4 \sin(t) \end{cases}$$

et la courbe paramétrique suivante

$$\begin{cases} x(t) = 2 \sin^2(t) \cos(t) \\ y(t) = 2 \cos^2(t) \sin(t) \end{cases} \quad \text{avec } t \in [0, 2\pi]$$



A rendre :

- Code source bien commenté.
- Un mode d'emploi qui décrit le fonctionnement de votre programme.