

GÉOMÉTRIE ALGORITHMIQUE

TEXTURES ET SHADER

TP SHADER PHONG – VERTEX SHADER

- Ouvrez le template à l'URL : <https://yourls.pascal-lab.ch/g>
- L'objectif de ce TP est d'implémenter l'éclairage Phong dans un Shader
- Le vertex Shader transmet deux variables interpolées au Fragment Shader
- Dans un fichier texte :
 - Trouver sur Internet comment est construite la normalMatrix (appelée aussi gl_NormalMatrix) et expliquer à quoi elle correspond
 - Expliquez à quoi correspond l'attribut varying
 - Expliquez la différence entre la **normal** dans le vertex shader et la **fNormal** dans le fragment shader
 - Question bonus : expliquez pourquoi il faut normaliser fNormal pour obtenir un rendu de meilleur qualité

```
precision highp float;
attribute vec3 position;
attribute vec3 normal;
uniform mat3 normalMatrix;
uniform mat4 modelViewMatrix;
uniform mat4 projectionMatrix;
varying vec3 fNormal;
varying vec3 fPosition;

void main()
{
    fNormal = normalize(normalMatrix * normal);
    vec4 pos = modelViewMatrix * vec4(position, 1.0);
    fPosition = pos.xyz;
    gl_Position = projectionMatrix * pos;
}
```

TP SHADER PHONG – FRAGMENT SHADER

- Utilisez les variables fNormal et fPosition ainsi que les constantes déjà présentes dans le templates pour recréer un éclairage Phong:
 - lightPos: la position de la source lumineuse dans le repaire monde
 - shininessVal : la brillance de la spéculaire
 - Ka : facteur d'amplification de la lumière ambiante
 - Kd : facteur d'amplification de la lumière diffuse
 - Ks : facteur d'amplification de la lumière spéculaire
 - ambientColor : couleur de la lumière ambiante
 - diffuseColor: couleur de la lumière diffuse
 - specularColor: couleur de la lumière spéculaire
- Créer une variable mode telle que :
 - Si mode = 1 afficher les 3 composantes de la lumière
 - Si mode = 2 n'afficher que la lumière ambiante
 - Si mode = 3 n'afficher que la lumière diffuse
 - Si mode = 4 n'afficher que la lumière spéculaire
- Reformulez avec vos mots à quoi correspondent les concepts de lumière ambiante, lumière diffuse et lumière spéculaire.

```
void main()
{
    vec3 lightPos = vec3(20.0, 20.0, 20.0);
    float shininessVal = 100.0;
    float Ka = 1.0;
    float Kd = 0.8;
    float Ks = 1.0;
    vec3 ambientColor = vec3(0.1, 0.1, 0.1);
    vec3 diffuseColor = vec3(1.0, 1.0, 1.0);
    vec3 specularColor = vec3(1.0, 1.0, 1.0);

    gl_FragColor = vec4(1.0, 1.0, 1.0, 0.0);
}
```