

Deferred Shading

En el mètode tradicional, conegut com a **Forward Rendering**, cada objecte de l'escena es processa individualment. Per a cada objecte, es calculen les interaccions amb totes les llums de l'escena, independentment de si la llum afecta realment l'objecte. Això pot resultar en un gran nombre de càlculs innecessaris, especialment quan hi ha moltes llums a l'escena.

El **Deferred Shading** proposa una solució elegant a aquest problema. En lloc de calcular la il·luminació per a cada objecte, la informació necessària per a la il·luminació es guarda en unes textures temporals, l'agrupació d'aquestes textures es coneix com a G-Buffer. Aquestes textures contenen informació com la posició, la normal i el color de cada píxel de l'escena.

En una segona passada, s'utilitzen les textures temporals per calcular la il·luminació. Com que la informació de la geometria ja està disponible, el càlcul de la il·luminació es fa només una vegada per cada píxel, independentment del nombre d'objectes que hi hagi a l'escena. Això permet il·luminar l'escena amb un gran nombre de llums sense un impacte significatiu en el rendiment.

En aquest exercici, volem fer una implementació senzilla de **Deferred Shading**. El programa haurà de:

- Utilitzarem un shader diferent per generar cada textura temporal.
 - *gbuffer_colorAndSpecular*: aquest shader escriu el color de l'objecte (matDiffuse) en els tres primers components del color buffer; també escriu el matShininess de l'objecte en la quarta component del color buffer.
 - *gbuffer_normal*: escriu la normal del fragment en world space en els tres primers components del color buffer. Com que estem escrivint al color buffer, haurem d'estandarditzar la normal perquè els seus valors estiguin entre 0 i 1.
 - *gbuffer_position*: escriu la posició del fragment en world space en els tres primers components del color buffer. En aquest cas també haurem d'estandarditzar la posició, ho podem fer amb la capsula englobant de l'escena.
- En la funció `paintGL()`, executem cada shader individualment, i per cada shader:
 - Fem un `glClear()` del color i depth buffers.
 - Activem el shader corresponent, enviem els uniforms necessaris, i dibuixem l'escena.
 - Copiem el contingut del color buffer a la textura corresponent.

- Finalment, amb totes les textures calculades activem el shader *deferredshading*, enviem els uniforms necessaris i dibuixem un rectangle que ocupi tota la pantalla. Aquest shader serà l'encarregat de fer els càlculs d'il·luminació amb el model de Phong.

Us proporcionem un codi esquelet on només cal afegir/canviar codi on s'indica amb comentaris de l'estil **// TODO: ...**

Heu de completar codi als fragment shaders i al .cpp del plugin.

Amb la tecla 'E' podeu canviar (cíclicament) entre diferents modes de rendering. Els tres primers modes pinten el color buffer dels shaders utilitzats per generar les textures temporals; l'últim mode pinta el resultat final del deferred shading.

