

# Resumen prácticas PG

Jose Gómez Gadea

## Práctica 5a

---

En esta práctica, se pretende modificar el fichero `.frag` utilizado por GLSL para que la esfera con una imagen de la tierra se vea igual que si la estuviésemos viendo desde un televisor antiguo. Para ello se ha modificado el color de cada fila de píxeles de modo que, por cada bloque de 4 filas contiguas de píxeles, se ha reducido un 20% la intensidad del color. Para la segunda y tercera fila se ha reducido el color un 80%, mientras que la última fila de píxeles se ha dejado al color base.

## Práctica 5b

---

Para esta parte, se pedía modelar un orbe mágico. Para simular dicho orbe, se han generado más esferas a partir de la que se tenía en la práctica 5a. A estas esferas, se le han aplicado un par de texturas distintas en el `main.cpp`, así como se ha modificado el movimiento rotatorio que tienen inicialmente para que cada una se mueva en un sentido diferente. A nivel del fichero `.frag`, se ha hecho que solo se representen los píxeles de la textura que tengan una intensidad mayor a cierto valor, de este modo una parte de la imagen es transparente y permite ver las esferas interiores, dando la buscada sensación de que parezca un orbe mágico.

## Práctica 6a

---

Para esta primera parte, se ha completado la implementación de las fuentes en el cálculo de iluminación modificando el `.vert`. Para ello se han utilizado las fórmulas mostradas en las diapositivas. Se ha calculado la distancia entre el vértice y la fuente, que se utiliza para calcular el factor de atenuación. Además, se calcula el efecto foco, que se utilizará en función a si la fuente es focal o no. Por último, al sumar el color final, se multiplica dicho número por el factor de atenuación y el efecto foco.

## Práctica 6b

---

Partiendo de la práctica 6a, se añade un slider (parte opcional) para calcular la luz utilizando Gouraud o Phong (se puede elegir en tiempo de ejecución). Además, se ha añadido dicho algoritmo de Phong a nivel de `.frag` (shader de fragmento) para que proporcione un efecto más realista a partir de los colores recibidos por el shader de vértice.

## Práctica 7

---

En esta práctica se busca eliminar todos los triángulos del conejo que se encuentren completamente dentro del cilindro que se mueve a lo largo del plano. Además, los triángulos que tengan un vértice dentro de dicho cilindro se pintarán de rojo, y los que tengan dos vértices dentro se pintarán de verde. Para ello, se ha añadido un shader de geometría y dentro se ha comprobado por cada uno de los vértices del triángulo si se encuentra o no dentro del cilindro. Para dicha comprobación, se ha utilizado el algoritmo descrito en el siguiente link:

<https://stackoverflow.com/questions/47932955/how-to-check-if-a-3d-point-is-inside-a-cylinder/47933302>