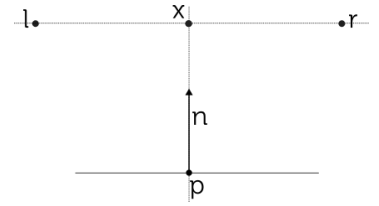


**Ejercicio 1 (15pts)** Describa brevemente en qué consisten los algoritmos de renderizado basados en *rasterización*, y basados en *raytracing*. Compare ámbos métodos señalando ventajas y desventajas

**Ejercicio 2 (10pts)** Defina y señale la diferencia entre “punto”, “vértice”, “fragmento” y “píxel”.

**Ejercicio 3 (15pts)** Para el cálculo de iluminación del modelo original de Phong se necesita calcular el rayo reflejado. Siendo  $l$  la posición de la luz,  $p$  el punto sobre la superficie a iluminar, y  $n$  la normal en ese punto, calcule el rayo reflejado de la siguiente forma: **a)** primero calcule el punto  $x$ ; **b)** luego defina los pesos para los puntos  $l$ ,  $p$  y  $x$  que permitan obtener  $r$  (explique/ilustre cómo los obtiene); y finalmente **c)** defina la dirección del rayo reflejado (como vector unitario).



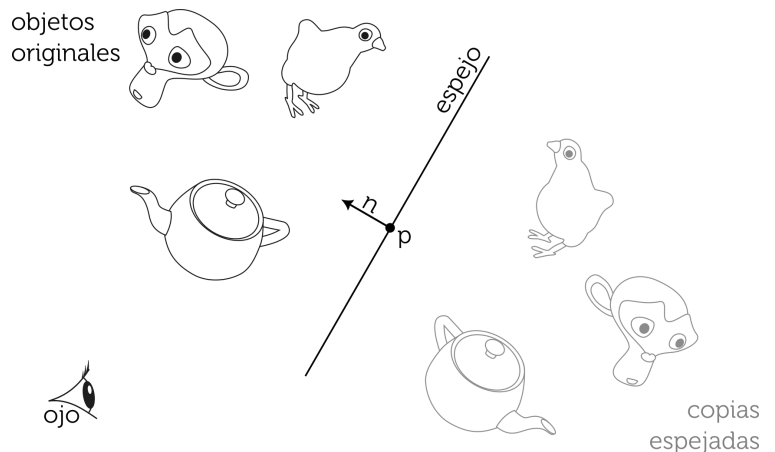
**Ejercicio 4 (15pts)** **a)** ¿Cuál es el efecto visual del exponente  $q$  del modelo de Phong? **b)** ¿Si tengo un material correctamente configurado, con un brillo especular intenso y bien definido, que será alto o bajo? **c)** Si en el material de la pregunta b) modificó sustancialmente el valor de  $q$  (si era muy alto, lo hago muy bajo, si era muy bajo lo hago muy alto), ¿el resultado tendría sentido? ¿o debería modificar algo más? Explique. **d)** ¿Tiene sentido  $q=0$ ? ¿Cómo se vería?

**Ejercicio 5 (20pts)** Describa un algoritmo para rasterizar un triángulo, dadas las coordenadas (x,y reales, pero ya en el espacio de la pantalla) de sus 3 vértices: **a)** ¿cómo generaría los fragmentos? o ¿cómo determina si un píxel debe pintarse? **b)** Explique qué más debería hacer/tener en cuenta para poder asociar a cada fragmento generado un color, una normal y un valor de profundidad (coord. z) interpolados. **c)** Discuta las ventajas/desventajas del algoritmo propuesto.

**Nota 1:** Se considerará correcta cualquier respuesta que haga que el triángulo se “vea” correctamente, sin importar el costo del algoritmo (es decir, podría no estar “optimizado” mientras sea correcto). Pero si es consciente de que el algoritmo podría no ser óptimo en ciertos casos, incluya estos casos en la discusión de ventajas/desventajas.

**Nota 2:** En a) debe describir detalladamente todos los algoritmos utilizados; en b) puede solo mencionar los métodos que utilizaría sin desarrollar todas las ecuaciones.

**Ejercicio 6 (25pts)** Un truco para simular un espejo plano perfecto cuando se usa un modelo de iluminación local, consiste en renderizar 2 veces todos los objetos, una vez en su posición real, y otra en la posición virtual del reflejo.



**a)** ¿Por qué un modelo de iluminación local necesita de estos trucos?

**b)** Si conoce un punto y la normal del plano del espejo, ¿cómo construiría la matriz de transformación que debe aplicar para llevar los objetos de su posición real a su imagen virtual en el espejo?

**c)** Si transforma las normales como habitualmente se lo hace (con la transpuesta de la inversa de la matriz del punto b), los objetos reflejados podrían verse completamente negros o directamente descartarse (dependiendo de la configuración de la etapa de clipping y culling del pipeline). ¿Por qué ocurriría esto? ¿Cómo lo resolvería en este caso? En un caso general, ¿cómo podría detectar cuando una matriz genera este problema?

**Ayuda:** Notar que si se traza una recta perpendicular al plano que pase por un punto/vértice original, su reflejo se encuentra sobre la misma recta, y a la misma distancia del plano. Comience por encontrar una función que permita obtener el reflejo de un punto.