

Entrega V Apartados del 70 al 77
Fecha de entrega: 8 de mayo de 2025

Instrucciones de la práctica

Para llevar a cabo este trabajo se tendrá en cuenta lo siguiente:

- Esta práctica se compone de una escena y la modificación de las luces de las escenas anteriores.
- Para realizar cada escena es necesario implementar una serie de pasos, en forma de apartados. No hay apartados opcionales.
- El código que se desarrolle deberá ser reutilizable para el examen final.
- Se tendrá especial atención a la calidad del código: inexistencia de fugas de memoria, variables con tipo correctamente definido, código bien organizado, comentado y legible, siguiendo los principios de la programación orientada a objetos.

Rúbrica de evaluación

Criterio	Pesos
Escena 10 (Materiales)	2.5
Luz direccional	1
Luz posicional	1
Luz focal	1
Luces en la nave espacial	2.5
Calidad del código	1
Ausencia de fugas	1
Total	10.0
Defensa (individual)	-1.5

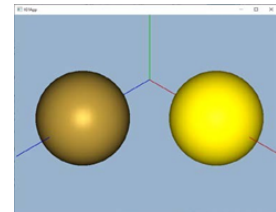
Escena 10: Materiales

Apartado 70

Incorpora a tu proyecto la clase `Material` para poder definir el material de los objetos de la escena.

Apartado 71

Define la clase `EntityWithMaterial` que hereda de `Abs_Entity` y que tiene un atributo de tipo `Material` como se ha visto en clase. Haz que la clase `ColorMaterialEntity` sea una subclase trivial de `EntityWithMaterial` que tenga como material uno con todas sus componentes a un color fijo.



Crea dos planetas Tatooine en la escena, uno con color amarillo estándar y otro con material dorado. No verás el resultado hasta acabar el apartado 74.

Esta escena se mostrará con la tecla `0`.

Apartado 72

Incorpora a tu proyecto la clase abstracta de las luces `Light`, las clases que heredan de ella `DirLight` y `PosLight`, para definir luces direccionales y posicionales, respectivamente, y la clase para definir focos `SpotLight`, que hereda de la clase de luces posicionales.

Apartado 73

Añade a la clase `Scene` un nuevo atributo `std::vector<Light*> gLights` para las luces de la escena. Extiende su método `render` para que llame a un nuevo método `uploadLights` que cargue las luces antes de renderizar las entidades. Actualiza también el resto de métodos de la clase que hagan falta.

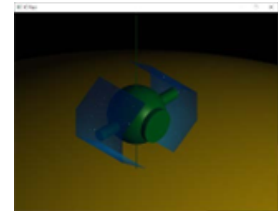
Apartado 74

Define una luz direccional `DirLight dirLight` con las características que aparecen en el código del shader de fragmentos de `simple_light`. Esta luz se activará/desactivará con la tecla `r` permitiendo pasar de una escena a oscuras a otra iluminada por esta luz.

La luz direccional estará disponible en todas las escenas, así que añádela al array `gLights` en `Scene::init`. Vacía el método `Camera::uploadVM`, que hacía referencia al shader `simple_light`. Las escenas de la práctica anterior se deberían renderizar como en aquella.

Apartado 75

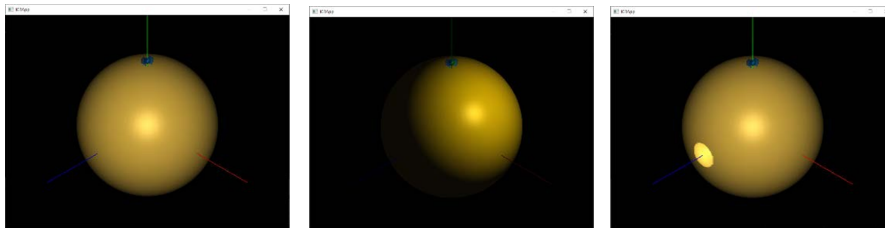
Define una luz posicional `PosLight` `posLight` en algún punto de la parte positiva del plano XY de forma que, cuando actúe solamente esta luz, con la tecla `t`, se ilumine/apague la escena tal como se muestra en la captura de abajo. Las componentes especular y ambiente de esta luz son como las de la direccional. La componente difusa es $(1.0, 1.0, 0.0)$ y el detalle adjunto muestra el resultado de iluminar el añil $(0, 65, 106)$ del TIE con esta luz.



La luz posicional (y las de los apartados siguientes) solo tienen que funcionar en la escena del planeta y el TIE (Escena 7), así que se crearán específicamente en esa escena y el manejo de las teclas del apagado/encendido se implementará en ella.

Apartado 76

Define un foco `SpotLight` `spotLight` y sitúalo en algún punto de la parte positiva del plano YZ de forma que, con las teclas `y`, se ilumine/apague. Las componentes de esta luz son como las de la direccional. En la captura de abajo se muestra encendida esta luz junto con la direccional.



Apartado 77

Añade un foco al vientre del `AdvancedTIE` de forma que, cuando el foco esté encendido, ilumine la parte de la esfera que está debajo del caza, tal como se muestra en las capturas adjuntas. Haz que con las teclas `h` se encienda/apague este foco. Por supuesto, el foco se debe desplazar junto con el TIE, según se mueva este con las teclas `f` y `g`.

