

## GiVD - curs 2022-2023 - Pràctica 2 – Grup B

NOM: \_\_\_\_\_ **Noah Márquez Vara** \_\_\_\_\_ NIUB: \_\_\_\_\_ **20392536** \_\_\_\_\_

Obre el projecte que has lliurat i fes els següents tests.

**Per a cada pregunta**, se't demana que:

- (1) modifiquis el teu codi per a fer el test, afegint les línies que calgui i comentant els canvis tant en el codi com incloent les línies de codi canviades en aquest document.
- (2) afegeixis una breu explicació en aquest mateix document raonant quines classes has canviat i les línies que has canviat.
- (3) facis una captura del resultat

**Què has de lliurar?**

- Aquest document amb totes les modificacions que hagi fet
- El codi del projecte modificat i comprimit en un .zip

**Segueix els següents passos i contesta cadascun dels punts:**

**1.** Executa la teva pràctica tal i com l'has lliurada i carrega el fitxer f16.obj amb la textura F16.bmp. Situa la càmera per a que es vegin l'avió completament. Fes tres captures de pantalla mostrant els resultats usant matisant **la textura en un 0.25** i la resta sigui el color calculat amb Blinn-Phong-Gouraud, Blinn-Phong-Phong i Toon shading. Inclou les tres captures en aquest fitxer.

- L'avió té com a material base els valors  $K_a = (0.1, 0.1, 0.1)$ ,  $K_d = (0.6, 0.6, 0.7)$ ,  $K_s = (0.7, 0.7, 0.7)$  i  $\beta = 250$ .

Explica breument com els hi assignes diferents materials a les esferes i a quina part del codi. Si et cal canviar alguna línia de codi, comenta-la amb la paraula **INICI**.

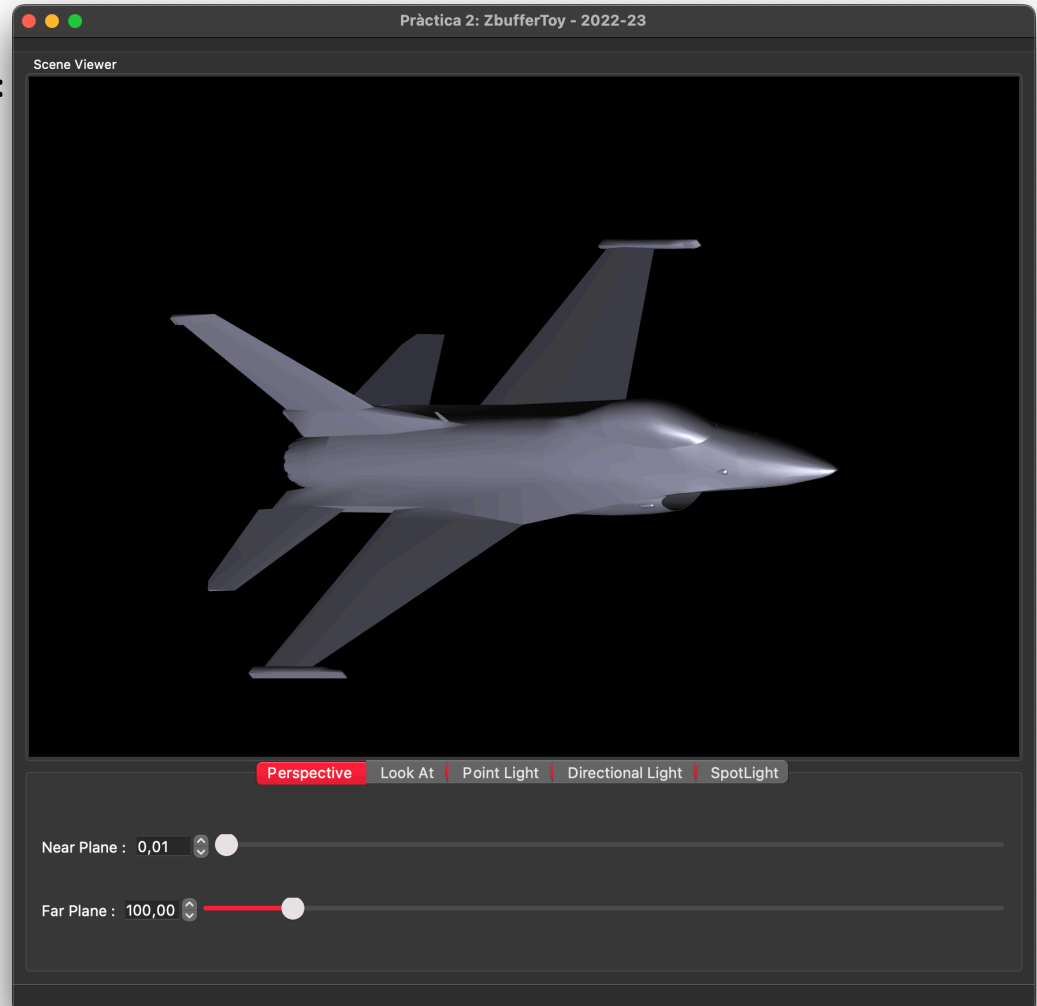
**[RESPOSTA]:** Tenim diverses formes per tal d'assignar diferents materials a cada una

de les esferes o .obj:

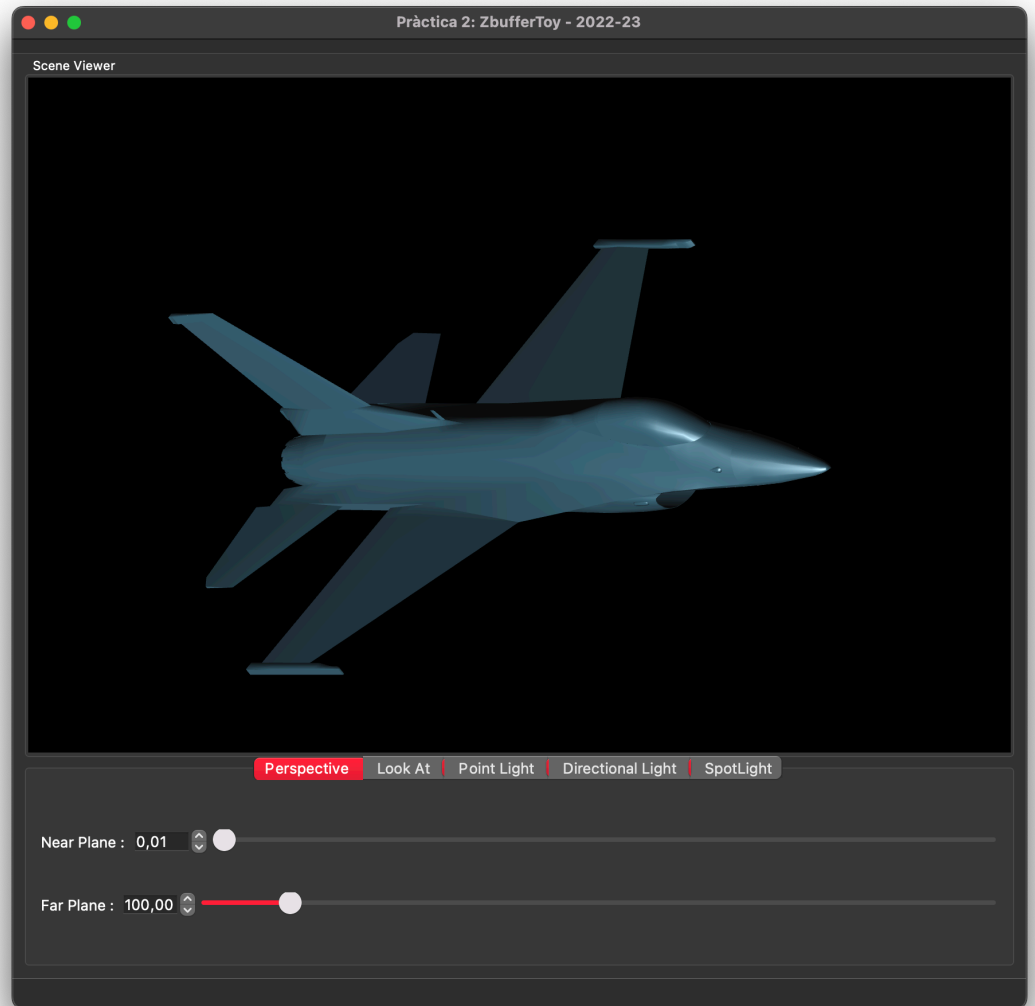
- Creant un fitxer .mtl corresponent per cert objecte .obj, formava part d'un dels opcionals de la pràctica i el vam implementar, així que una de les opcions seria aquesta.
- Un altre seria crear un fitxer d'escena virtual (un .json) afegint l'objecte / els objectes que volguéssim, cadascun amb els seus materials.
- I un altre possibilitat seria *hardcodejant* els valors del material quan s'assigna el material al llegir un .obj, des de la classe GPUMesh (l'opció que he escollit per *setejar* els valors del material del f16, i que he marcat amb la paraula INICI en el codi).

**[NOTA]:** Com es comenta al GitHub, no vam poder acabar d'aconseguir fer servir les textures de manera correcta, sembla que canvia el color segons la textura, però no s'acaba de mostrar la imatge.

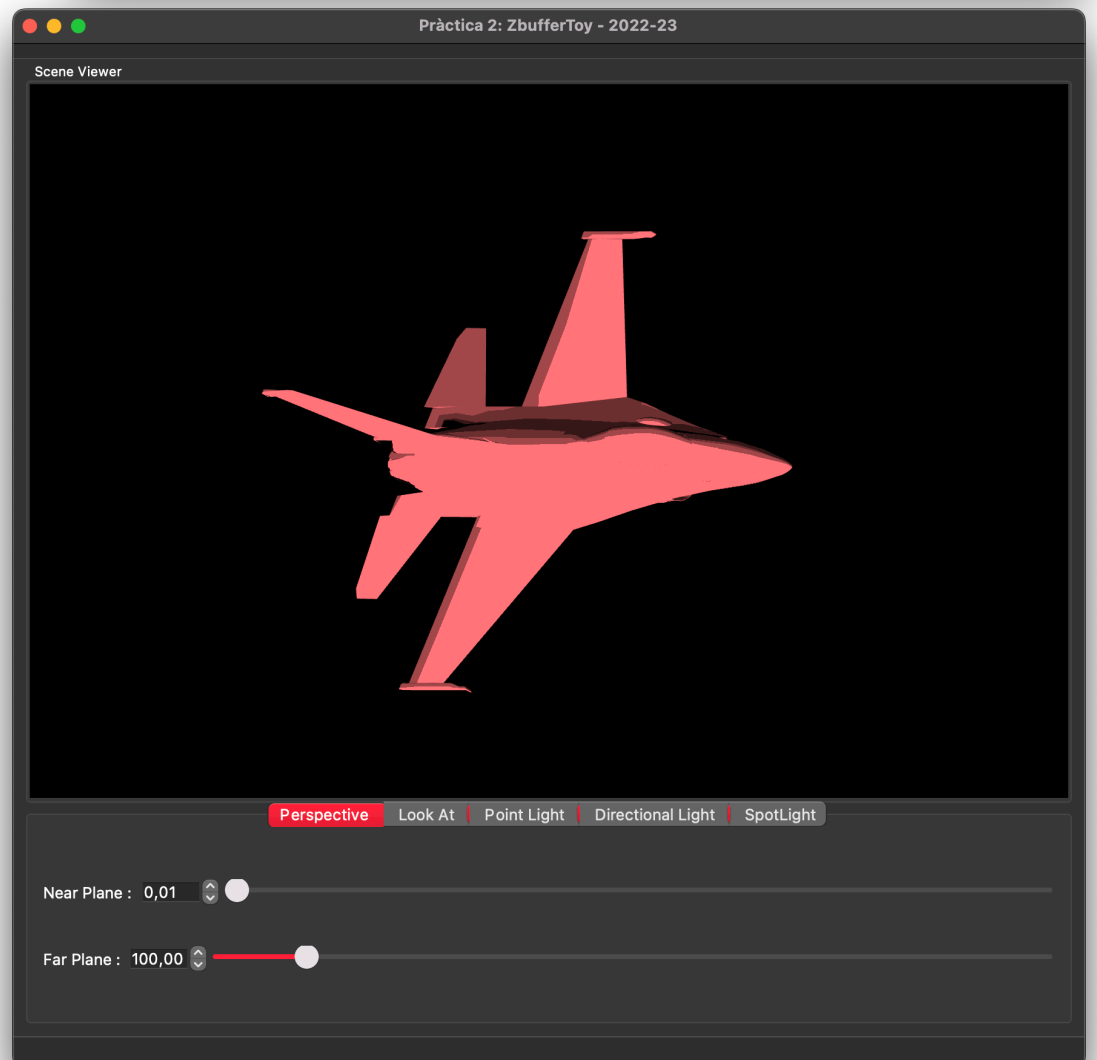
**[Blinn-Phong-Gouraud]:**



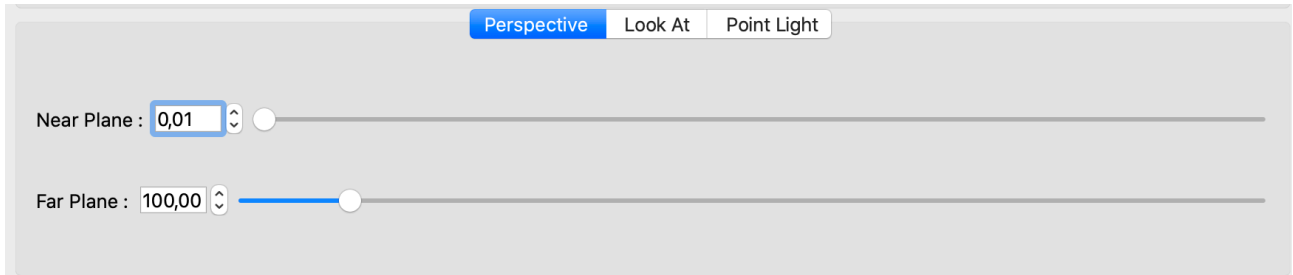
[Blinn-Phong-Phong]:



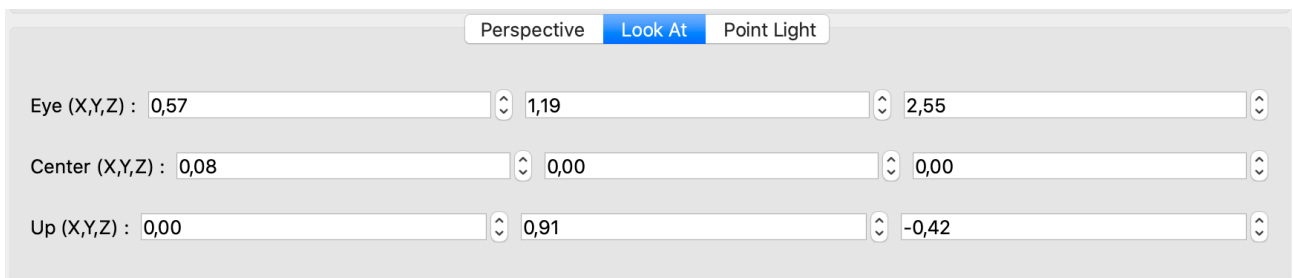
[Toon shading]:



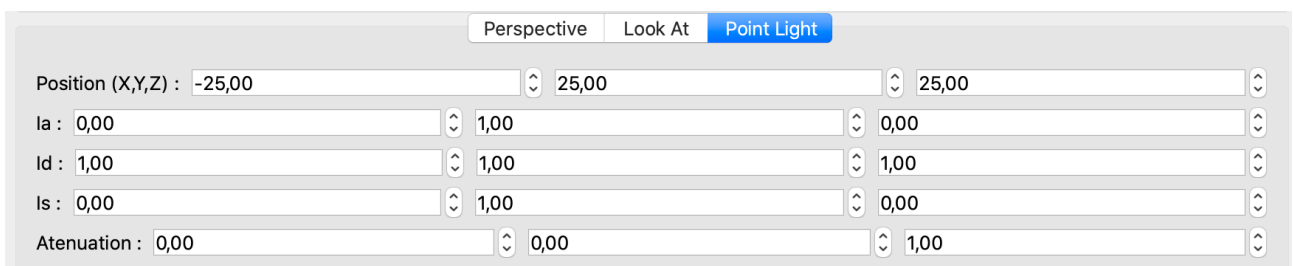
**2.** Modifica els paràmetres de la càmera segons les captures següents. Posa només una llum amb les característiques indicades a la figura. Si has de modificar alguna línia de codi, comenta-la amb la paraula **SETTINGS**. Adjunta aquí les modificacions fetes en els diferents fitxers del teu codi, si és que et calen, i una captura de la imatge obtinguda.



Interface showing camera settings. The 'Perspective' tab is selected. The 'Near Plane' is set to 0,01 and the 'Far Plane' is set to 100,00. Both are represented by sliders.



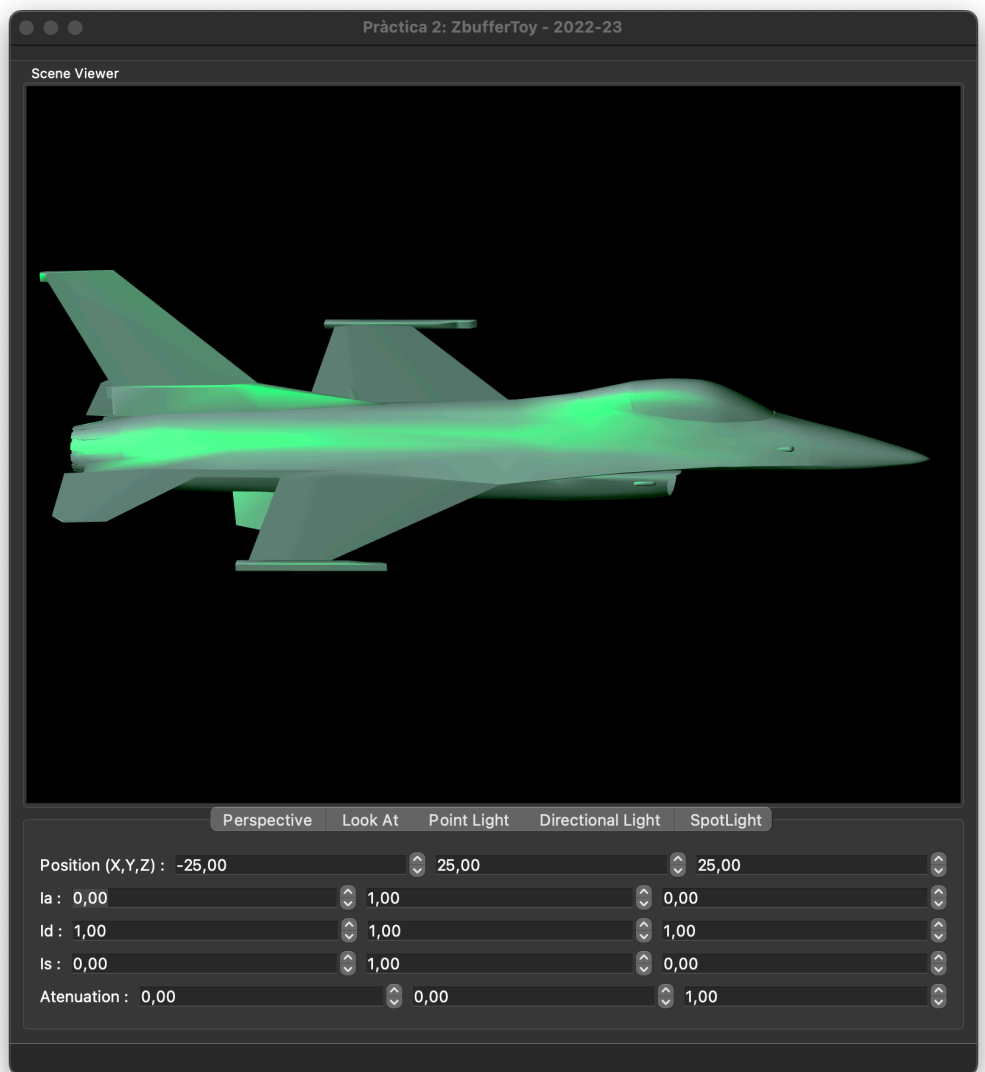
Interface showing camera settings. The 'Look At' tab is selected. The 'Eye (X,Y,Z)' is set to 0,57, 1,19, 2,55. The 'Center (X,Y,Z)' is set to 0,08, 0,00, 0,00. The 'Up (X,Y,Z)' is set to 0,00, 0,91, -0,42.



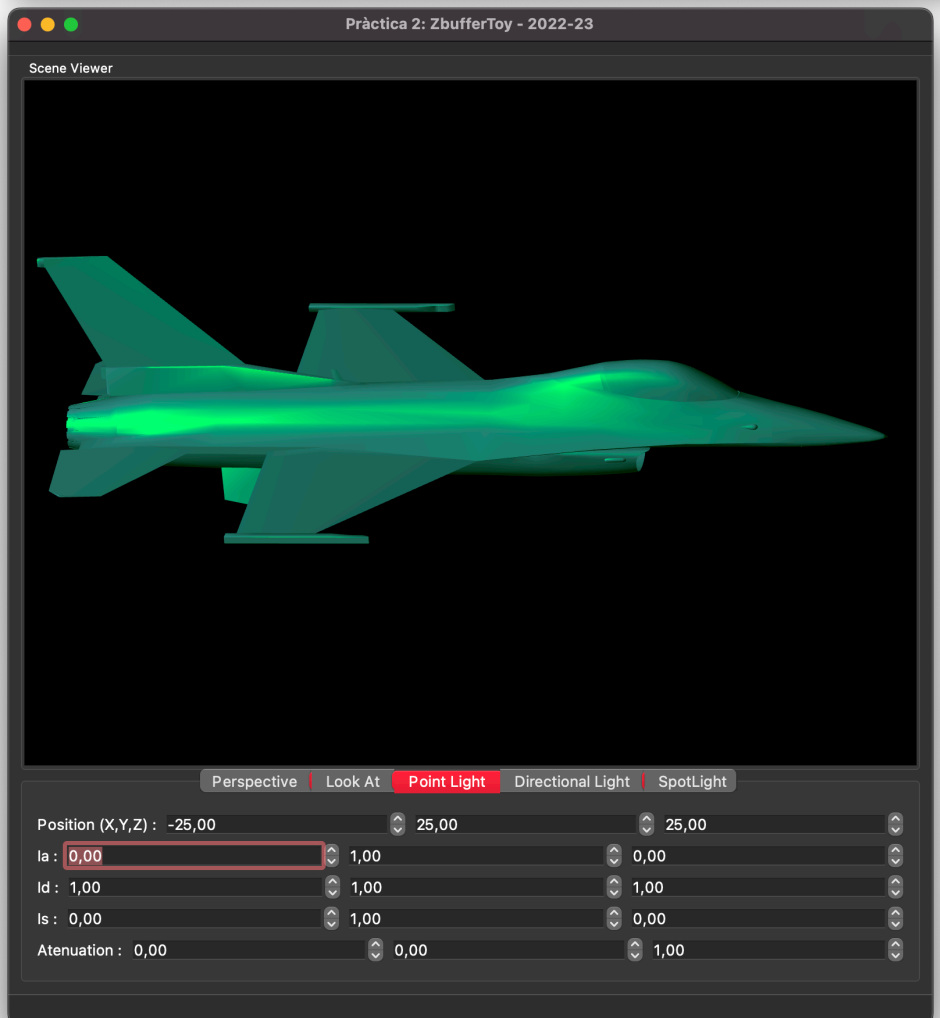
Interface showing light settings. The 'Point Light' tab is selected. The 'Position (X,Y,Z)' is set to -25,00, 25,00, 25,00. The 'Ia' is set to 0,00, 1,00, 0,00. The 'Id' is set to 1,00, 1,00, 1,00. The 'Is' is set to 0,00, 1,00, 0,00. The 'Attenuation' is set to 0,00, 0,00, 1,00.

[RESPOSTA]:

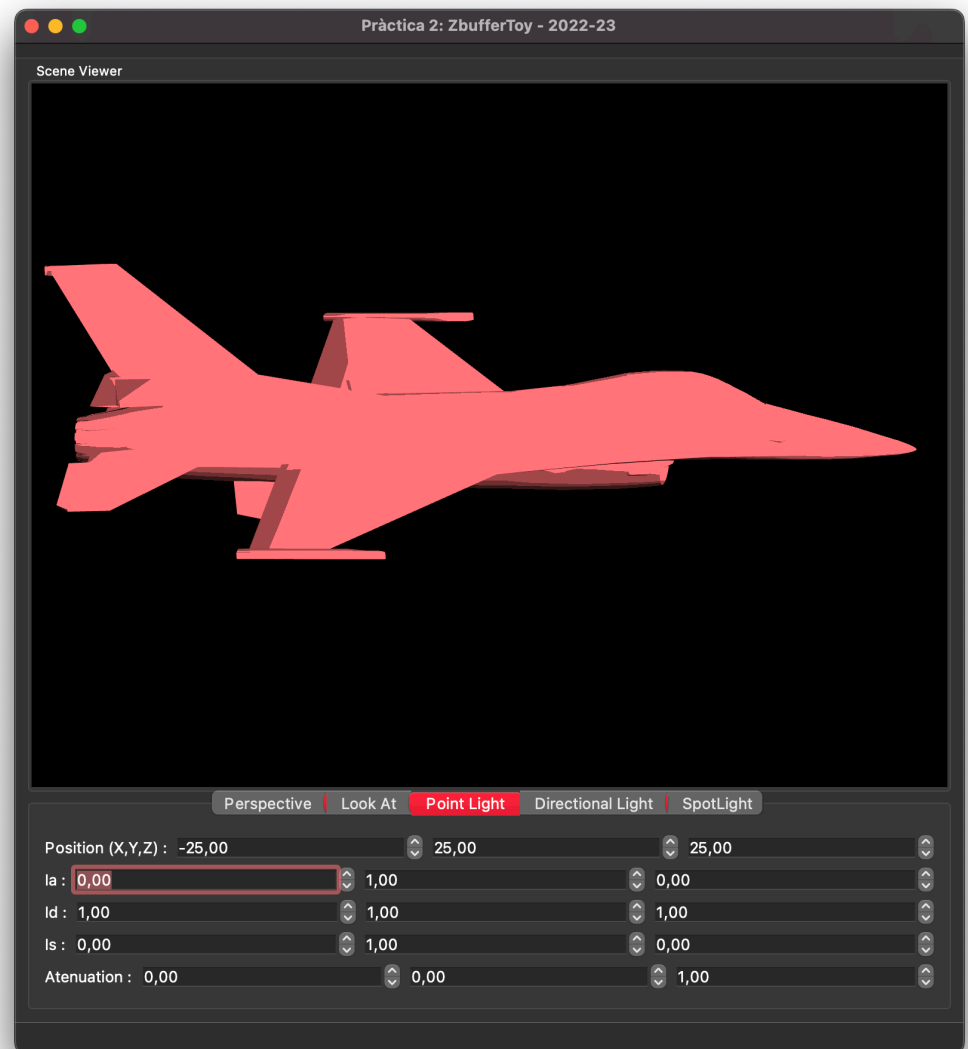
[Blinn-Phong-Gouraud]:



[Blinn-Phong-Phong]:



[Toon shading]:



3. Crea els parells de vèrtexs-fragment shaders que consideris per a poder tenir una visualització sèpia de l'escena de forma que els objectes que estiguin dins d'un cercle centrat en la finestra de visualització, es pintin **en tons de sèpia** matisant els colors de l'avió obtinguts a l'apartat 1 amb Blinn-Phong Gouraud i **els que estan fora del cercle es pintin en el color del material base de l'avió sense la textura** també usant Blinn-Phong Gouraud. Es vol tenir transició **no** suau entre els píxels que estan dins i fora. On ho hauràs de comprovar?

Per a calcular el color sèpia cal que facis servir la següent fórmula. Donat un color rgb:

```
r = r * 0.393 + b * 0.769 + g * 0.189;  
g = r * 0.349 + b * 0.686 + g * 0.168;  
b = r * 0.272 + b * 0.534 + g * 0.131;
```

Recorda que els valors dels colors han d'estar entre 0 i 1.

Modifica el teu codi per a que per defecte s'activi/n aquest/s shader/s.

- Abans de modificar codi, respon a les preguntes:

**a. On has de calcular el color?**

El color de les parts de l'avió que estan fora del cercle es calcula en el vèrtex *shader*, tal i com ho fèiem amb les diferents variants de *Gouraud*. A l'hora de calcular el color final, amb les parts de dins i fora, enviem el color resultant al fragment shader, on modifiquem el color segons si els objectes estan dins o fora del cercle.

**b. On has de comprovar que el punt està dins del cercle?**

Ho comprovo al fragment shader, primer transformo les coordenades 3D a 2D de viewport. Calculo el centre del cercle en l'espai projectiu, calculo la distància des del centre del fragment al centre del cercle en l'espai projectiu, i llavors segons un paràmetre de distància a l'esfera, pinto el color amb una tonalitat sèpia, o bé deixo el càlcul de Blinn-Phong Gouraud.

**c. Cal que passis paràmetres addicionals a la GPU? Si és així, serà UNIFORM? O seran variables IN? A quin shader els has d'enviar? Quan?**

No m'ha fet falta enviar cap més paràmetre addicional a la GPU, almenys cap paràmetre extra dels que enviava per tal de fer Gouraud.

**d. Quan has d'inicialitzar aquests nous shaders? Quan has d'activar aquests shaders?**

Aquests nous shaders (anomenats *vProva.glsl* i *fProva.glsl*) els inicialitzo a l'iniciar el codi, en el mètode *initializeGL()* de la classe *GLWidget*, cridant al mètode *initShadersGPU()*. Posteriorment, com es demana

iniciar aquests nous shaders per defecte, l'activo també dins del mètode *initializeGL()* de la classe *GLWidget*, cridant al mètode *useShader()*.

- **Soluciona el codi, comentant-lo amb comentaris que comencin per la paraula **SÈPIA**.**

S'han comentat totes les parts modificades del codi afegint la paraula "SÈPIA" com a comentari. Les classes modificades han estat:

- *GLShader* per tal d'afegir a l'enum el nou tipus de shader.
  - *GLWidget* per tal d'inicialitzar els nous shaders (*vProva.glsl* i *fProva.glsl*) i utilitzar-los.
- **Adjunta les captures de pantalla que obtinguis.**

