	Manual de Prácticas
Secretaría/División: División de Ingeniería Eléctrica	Área/Departamento: Ingeniería en Computación


Laboratorio de Computación Gráfica e Interacción Humano Computadora

Iluminación 2

N° de practica: 08

Nombre completo de los alumnos		Firma
Arroyo Llanes Miguel Alejandro		
N° de brigada: 2	Fecha de ejecución: 12/04/2025	Grupo: 03
Calificación:	Profesor: Ing. Jose Roque Roman Guadarrama	

Elaborado por:	Revisado por:	Autorizado por:	Vigente desde:
-----------------------	----------------------	------------------------	-----------------------

	Manual de Prácticas
Secretaría/División: División de Ingeniería Eléctrica	Área/Departamento: Ingeniería en Computación

Comentario:

En esta práctica se reforzaron los conocimientos adquiridos de la iluminación y se complementaron con nuevos conceptos como el que al avanzar o retroceder se enciendan las luces delanteras o traseras del auto, también se reforzaron los conceptos del texturizado en este caso haciendo un BMO.

Para el primer ejercicio relacionado al spotlight se copio lo realizado para el helicóptero ligando la luz al cofre para que este acompañe al modelo del cofre en su movimiento copiando también su rotación.

Para el segundo se modifíco lo que ya se tenía para que se encendiera cuando se presionara el botón de avanzar o retroceder (luz azul para avanzar y roja para retroceder).

Para el tercero solo se realizó lo mismo que en la lámpara donde por facilidad o costumbre primero se utilizó un spotlight para encontrar el punto clave de donde debería de estar la luz y así volverlo pointlight. Se tuvo que modificar lo que se tenía sobre la forma en la que este se encendía y apagaba ya que a como estaba antes no permitía probar la luz de BMO, una vez modificado esto se copió la misma lógica, pero ahora con las teclas "I" y "K" para la luz de BMO.


Bloque de código:

Controles de encendido y apagado luz de BMO en window.h y window.cpp.

```

44  GLfloat getLuzLamp() { return luzLamp; }
45  GLfloat getLuzLamp2() { return luzLamp2; }
46
47  ~Window();
48  private:
49  GLFWwindow *mainWindow;
50  GLint width, height;
51
52  GLfloat rotax, rotay, rotaz, articulacion1, articulacion2, articulacion3, articulacion4, articulacion5, articulacion6, articulacion7,
53  articulacion8, articulacion9, articulacion10,
54  articulacionLlanta1, articulacionLlanta2, articulacionLlanta3, articulacionLlanta4,
55  avanzar, avanzarH, luzLamp, luzLamp2;
56

```

	Manual de Prácticas
Secretaría/División: División de Ingeniería Eléctrica	Área/Departamento: Ingeniería en Computación

```

33
34     luzLamp = 0.0f;
35     luzLamp2 = 0.0f;
36

```

```

248
249 //Para la lampara de BMO
250 if (key == GLFW_KEY_K && action == GLFW_PRESS)
251 {
252     theWindow->luzLamp2 = 0;
253 }
254
255 if (key == GLFW_KEY_I && action == GLFW_PRESS)
256 {
257     theWindow->luzLamp2 = 1;
258 }
259
260

```

Importar BMO.

```


62
63     //Importar BMO
64     Model BMO;
65

```

```

309
310
311 //Importar BMO
312 BMO = Model();
313 BMO.LoadModel("Models/BMO.obj");
314
315 //Dado

```


	Manual de Prácticas
Secretaría/División: División de Ingeniería Eléctrica	Área/Departamento: Ingeniería en Computación

Luces de cofre, auto y BMO.

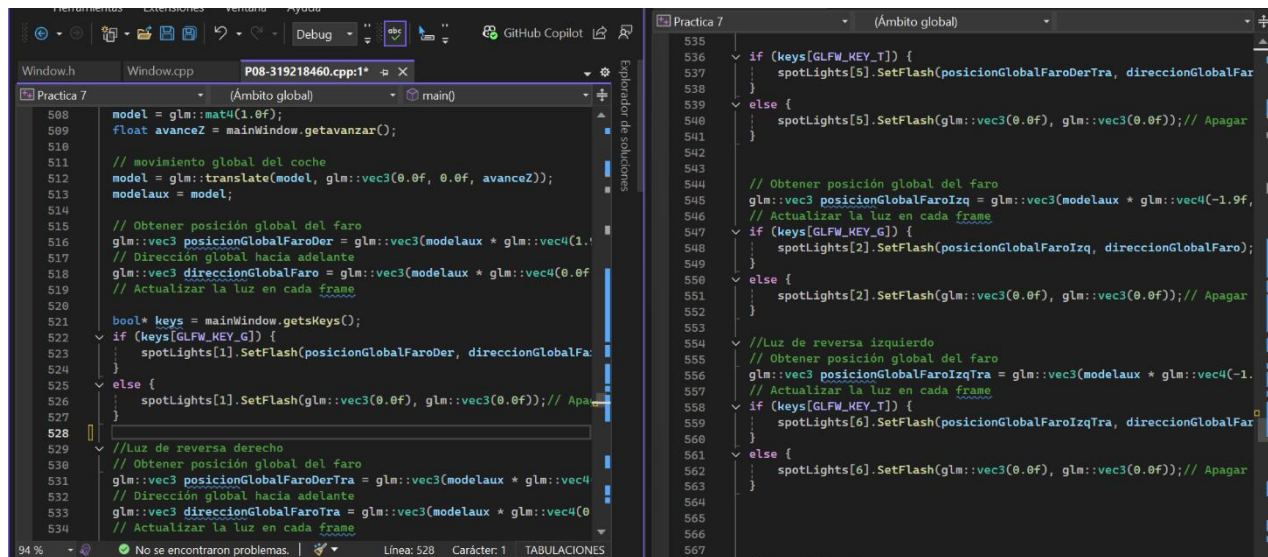
```

367      //luz fija faro izquierdo
368      spotLights[2] = SpotLight(0.0f, 0.0f, 1.0f, //Color
369          1.0f, 2.5f,          //ambient, diffuse intensity
370          -1.9f, 1.15f, 3.0f,  //posición inicial
371          0.0f, 0.0f, 1.0f,    //dirección
372          1.0f, 0.0f, 0.0f,    //atenuación
373          12.0f);              //ángulo de luz (borde del cono)
374      spotLightCount++;
375
376      //Luz del cofre
377      spotLights[3] = SpotLight(0.5f, 0.5f, 0.0f, //Color
378          1.0f, 2.5f,          //ambient, diffuse intensity
379          0.0f, 1.7f, 3.0f,    //posición inicial
380          0.0f, -1.0f, 0.0f,   //dirección
381          1.0f, 0.0f, 0.0f,    //atenuación
382          25.0f);              //ángulo de luz (borde del cono)
383      spotLightCount++;
384
385
386
403
404      //luz fija faro derecho trasero
405      spotLights[5] = SpotLight(1.0f, 0.0f, 0.0f, //Color
406          1.0f, 2.5f,          //ambient, diffuse intensity
407          1.9f, 1.7f, -6.0f,   //posición inicial
408          0.0f, 0.0f, -1.0f,   //dirección
409          1.0f, 0.0f, 0.0f,    //atenuación
410          12.0f);              //ángulo de luz (borde del cono)
411      spotLightCount++;
412
413      //luz fija faro izquierdo trasero
414      spotLights[6] = SpotLight(1.0f, 0.0f, 0.0f, //Color
415          1.0f, 2.5f,          //ambient, diffuse intensity
416          -1.9f, 1.7f, -6.0f,  //posición inicial
417          0.0f, 0.0f, -1.0f,   //dirección
418          1.0f, 0.0f, 0.0f,    //atenuación
419          12.0f);              //ángulo de luz (borde del cono)
420      spotLightCount++;
421
422      //luz fija BMO
423      pointLights[2] = PointLight(1.0f, 1.0f, 1.0f, //Color
424          0.5f, 0.4f,          //ambient, diffuse intensity
425          -10.0f, 0.7f, 6.0,   //posición inicial
426          0.6f, 0.2f, 0.1f);  //atenuación
427      pointLightCount++;
428
429      //se crean mas luces puntuales y spotlight

```

	Manual de Prácticas
Secretaría/División: División de Ingeniería Eléctrica	Área/Departamento: Ingeniería en Computación

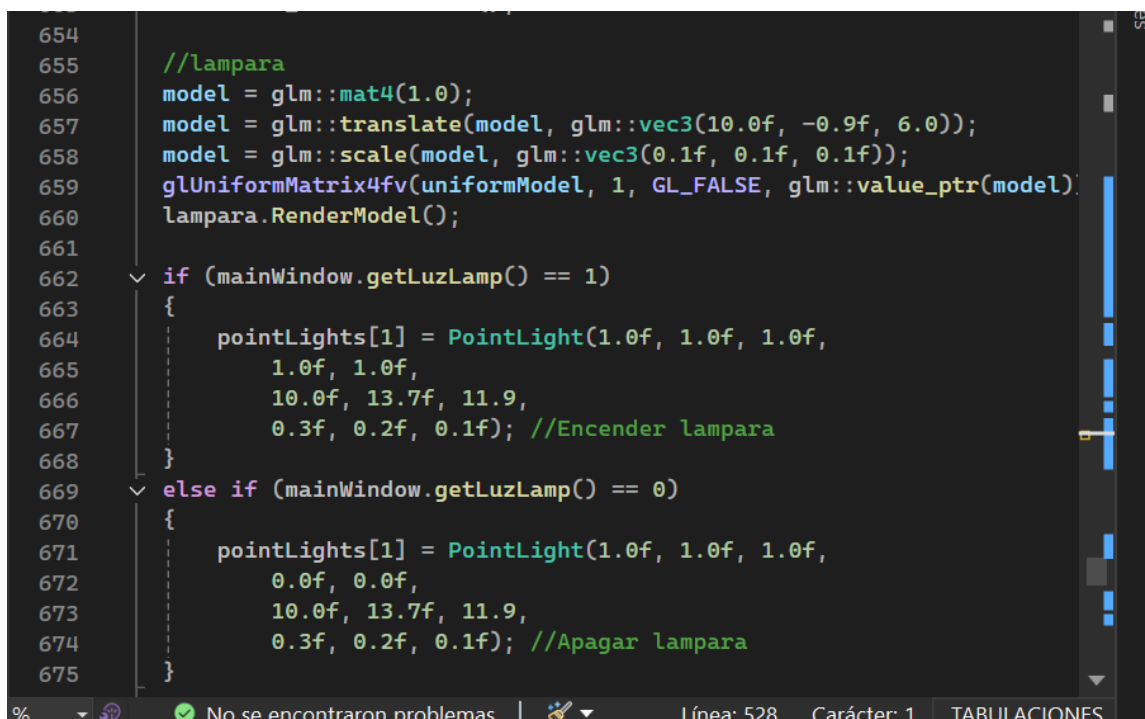
Lógica dentro del while.



```

508 model = glm::mat4(1.0f);
509 float avanceZ = mainWindow.getAvanzar();
510
511 // movimiento global del coche
512 model = glm::translate(model, glm::vec3(0.0f, 0.0f, avanceZ));
513 modelaux = model;
514
515 // Obtener posición global del faro
516 glm::vec3 posicionGlobalFaroDer = glm::vec3(modelaux * glm::vec4(1.0f, 0.0f, 0.0f, 1.0f));
517 // Dirección global hacia adelante
518 glm::vec3 direccionGlobalFaro = glm::vec3(modelaux * glm::vec4(0.0f, 0.0f, 1.0f, 0.0f));
519 // Actualizar la luz en cada frame
520
521 bool* keys = mainWindow.getKeys();
522 if (keys[GLFW_KEY_G]) {
523     spotLights[1].SetFlash(posicionGlobalFaroDer, direccionGlobalFaro);
524 }
525 else {
526     spotLights[1].SetFlash(glm::vec3(0.0f), glm::vec3(0.0f)); // Apagar
527 }
528
529 // Luz de reversa derecho
530 // Obtener posición global del faro
531 glm::vec3 posicionGlobalFaroDerTra = glm::vec3(modelaux * glm::vec4(1.0f, 0.0f, 0.0f, 1.0f));
532 // Dirección global hacia adelante
533 glm::vec3 direccionGlobalFaroTra = glm::vec3(modelaux * glm::vec4(0.0f, 0.0f, 1.0f, 0.0f));
534 // Actualizar la luz en cada frame
535
536 // Obtener posición global del faro
537 glm::vec3 posicionGlobalFaroIzq = glm::vec3(modelaux * glm::vec4(-1.9f, 0.0f, 0.0f, 1.0f));
538 // Actualizar la luz en cada frame
539 if (keys[GLFW_KEY_G]) {
540     spotLights[2].SetFlash(posicionGlobalFaroIzq, direccionGlobalFaro);
541 }
542 else {
543     spotLights[2].SetFlash(glm::vec3(0.0f), glm::vec3(0.0f)); // Apagar
544 }
545
546 // Luz de reversa izquierdo
547 // Obtener posición global del faro
548 glm::vec3 posicionGlobalFaroIzqTra = glm::vec3(modelaux * glm::vec4(-1.9f, 0.0f, 0.0f, 1.0f));
549 // Actualizar la luz en cada frame
550 if (keys[GLFW_KEY_T]) {
551     spotLights[6].SetFlash(posicionGlobalFaroIzqTra, direccionGlobalFaro);
552 }
553 else {
554     spotLights[6].SetFlash(glm::vec3(0.0f), glm::vec3(0.0f)); // Apagar
555 }
556
557


```



```

654
655 //lampara
656 model = glm::mat4(1.0);
657 model = glm::translate(model, glm::vec3(10.0f, -0.9f, 6.0));
658 model = glm::scale(model, glm::vec3(0.1f, 0.1f, 0.1f));
659 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
660 lampara.RenderModel();
661
662 if (mainWindow.getLuzLamp() == 1)
663 {
664     pointLights[1] = PointLight(1.0f, 1.0f, 1.0f,
665     1.0f, 1.0f,
666     10.0f, 13.7f, 11.9,
667     0.3f, 0.2f, 0.1f); //Encender lampara
668 }
669 else if (mainWindow.getLuzLamp() == 0)
670 {
671     pointLights[1] = PointLight(1.0f, 1.0f, 1.0f,
672     0.0f, 0.0f,
673     10.0f, 13.7f, 11.9,
674     0.3f, 0.2f, 0.1f); //Apagar lampara
675 }

```

	Manual de Prácticas
Secretaría/División: División de Ingeniería Eléctrica	Área/Departamento: Ingeniería en Computación




```


//BMO
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(-10.0f, -0.7f, 6.0));
model = glm::scale(model, glm::vec3(30.0f, 30.0f, 30.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
BMO.RenderModel();

if (mainWindow.getLuzLamp2() == 1)
{
    pointLights[2] = PointLight(1.0f, 1.0f, 1.0f, //Color
                                0.5f, 0.4f, //ambient, diffuse
                                -10.0f, 0.7f, 6.0, //posición inicial
                                0.6f, 0.2f, 0.1f); //Encender lampara
}
else if (mainWindow.getLuzLamp2() == 0)
{
    pointLights[2] = PointLight(1.0f, 1.0f, 1.0f, //Color
                                0.0f, 0.0f, //ambient, diffuse
                                -10.0f, 0.7f, 6.0, //posición inicial
                                0.6f, 0.2f, 0.1f); //Apagar lampara
}

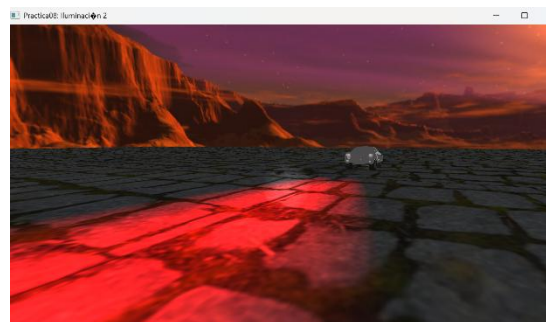
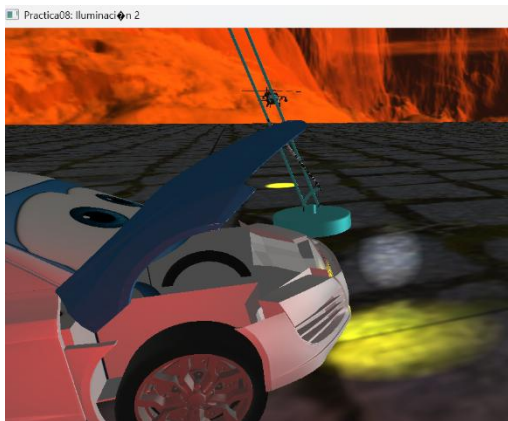
//Agave ¿qué sucede si lo renderizan antes del coche y el helicóptero?
//Desaparece


```


 No se encontraron problemas. |

Línea: 528 Carácter: 1 TABULACIONES

	Manual de Prácticas
Secretaría/División: División de Ingeniería Eléctrica	Área/Departamento: Ingeniería en Computación

Ejecución:



	Manual de Prácticas
Secretaría/División: División de Ingeniería Eléctrica	Área/Departamento: Ingeniería en Computación

Conclusión:

Se pusieron en practica todos los conocimientos adquiridos hasta el momento, esta práctica reforzando conceptos como la texturización y la iluminación aplicando este último de maneras más versátiles, solo tuve ligeras dificultades con cada ejercicio relacionado a la manipulación de la luz, pero todos estos pudieron ser resueltos eventualmente alcanzando una maestría en resolver los problemas.

Bibliografía:

1. Sketchfab. (s. f.). BMO - Adventure Time - Download Free 3D model by felipeferreira. <https://sketchfab.com/3d-models/bmo-adventure-time-74c64f6ede7f408dad57bc946eae2881>