Examen OpenGL de IDI 2015-2016, Q1. Grupo 20. Tiempo: 1h 30'

Leed detenidamente las instrucciones y el enunciado antes de empezar a hacer nada!

Instrucciones

1. Podéis usar el código que habéis elaborado en las clases de laboratorio y que tengáis en vuestra cuenta, pero **sólo el código que hayais generado vosotros**; no podéis usar código que otros estudiantes hayan compartido con vosotros (ni que hayais compartido con otros estudiantes). De lo contrario se considerará copia.

2. Partiremos del código que tenéis en `examen.tgz` (adjunto a esta práctica). Tenéis que descomprimir este archivo en un directorio vuestro. Os creará un subdirectorio `examen` donde tendréis todos los ficheros con los que tenéis que trabajar. **No tenéis que modificar los ficheros** `examen.pro` **ni el** `main.cpp`. Los ejercicios que os pedimos sólo requieren cambios en la classe `MyGLWidget` y en los shaders.

3. **Si el código que entreguéis no compila o da error de ejecución, la evaluación será un 0**, sin excepciones.

4. Para hacer la entrega tenéis que generar un archivo tar que incluya todo el código de vuestro examen que se llame `<nombre-usuari>_GL.tgz`, donde sustituiréis `<nombre-usuari>` por vuestro nombre de usuario. Por ejemplo, el estudiante Pompeu Fabra (desde una terminal en la que se ha colocado dentro del directorio `examen`):

   ```
   make distclean
   tar zcvf pompeu.fabra_GL.tgz *
   ```

   Es importante el '`make distclean`' para borrar los archivos binarios generados; que el nombre de usuario sea el correcto (el vuestro); y que tenga la raya baja '_' separando el nombre de usuario del sufijo `GL.tgz`

5. Una vez hecho esto, en vuestro directorio examen tendréis el archivo `<nombre-usuario>_GL.tgz` que es lo que tenéis que entregar. **Haced la comprobación**, descomprimiendo este archivo **en un directorio completamente vacío**, que el código que entreguéis compila (haciendo `qmake; make`) y ejecuta correctamente.

6. Finalmente, entregad el fichero en https://examens.fib.upc.edu

Nota: Recordad que si abrís el fichero `/assig/idi/man_3.3/index.html` desde el Firefox o el Konqueror tendréis acceso a las páginas del manual de OpenGL 3.3, y con el fichero `/usr/include/glm/doc/api/index.html` tendréis acceso a las páginas del manual de la librería gml. También tenéis, como ya sabéis, el `assistant` para dudas de Qt.

Enunciado

El código que os pasamos tal y como está pinta un suelo de 20x20 centrado en el origen y un Patricio sin escalar situado con el centro de su base en el origen de coordenadas, y con una cámara inicializada de forma arbitraria (ver una imagen en el fichero `EscIni-20.png`). Tiene inicializados todos los datos de materiales y normales necesarios para poder implementar el cálculo de la iluminación. También os pasamos los métodos `Lambert` y `Phong` que se encuentran en el Vertex Shader.

1. *(3 puntos)* Modifica esta escena para que en lugar de un Patricio sin escalar y con la base centrada en el origen, haya un pilar de 3 Patricios. El Patricio de la base del pilar debe ser de altura 3, mirar en la dirección del eje Z+ y tener la base de su caja centrada en el punto (0, 0, -8). El Patricio del piso del medio debe ser de altura 2.5 y mirar en la dirección del eje Z-. Y finalmente el Patricio de arriba del todo debe ser de altura 2 y mirar en la dirección del eje Z+ (igual que el de la base).

   Esta escena se debe ver centrada y sin recortar, y aprovechando el máximo del viewport (vista), con una cámara perspectiva. En caso de redimensionamiento de la ventana (resize) la escena no debe recortarse ni deformarse.

   Una imagen posible de la solución a este ejercicio la podéis ver en el archivo `EscSol1-20.png`.

2. *(2 puntos)* Añade a la escena el cálculo de la iluminación **en el Vertex Shader** usando el modelo de Phong y con un foco de luz blanca en la posición (-5, 5, 5) de la escena. Modifica también las propiedades del material del suelo para que sea de un material amarillo brillante.

3. *(2 puntos)* Haz que con las teclas 'W' y 'S' el pilar entero se mueva sobre un eje paralelo al eje Z, de forma que con 'W' el pilar se mueve hacia las Z+ y con 'S' el pilar se mueve hacia las Z-. El incremento/decremento de distancia a mover debe ser en cada momento de 0.5.

4. *(1 punto)* Haz que al pulsar la tecla 'L' el foco de luz cambie de color y pase a ser un foco magenta. Al pulsar la tecla 'L' de nuevo, volverá al color del foco inicial.

5. *(2 puntos)* Haz que al pulsar la tecla 'X' el Fragment Shader pinte toda la escena haciendo franjas blancas o negras dependiendo de la altura (coordenada Y) del fragmento. Las franjas deben ser de 10 píxeles de grosor. Si se vuelve a pulsar la tecla 'X' se vuelve a la iluminación original (la del ejercicio 2). Para hacer este ejercicio, recordad que la variable `gl_FragCoord` os da las coordenadas del fragmento y es de tipo `vec4`. El Vertex Shader no es necesario que lo toqueis en este ejercicio, es decir dejadlo igual que en el ejercicio 2.