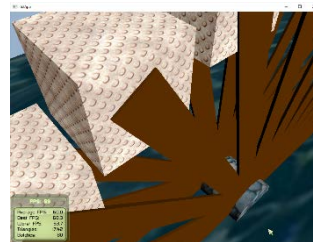
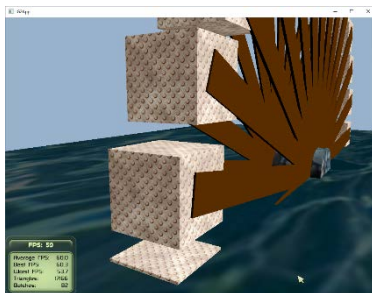
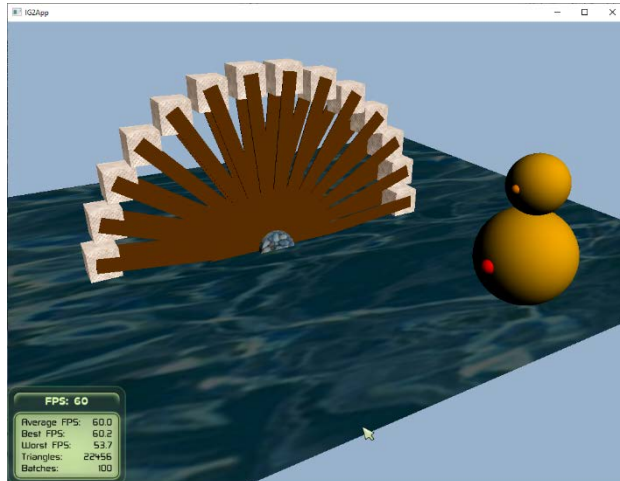


**INFORMÁTICA GRÁFICA 2**  
**Grado en desarrollo de videojuegos**  
**Curso 2022-23**  
**Práctica 1**

**(Entrega 2, apartados 20-33)**

**20.** Añade material a la escena mediante un script **Practica1.material**. En él se debe precisar lo siguiente. Para el muñeco, la nariz es de color naranja y el ombligo de color rojo. La cabeza y el cuerpo tienen el color naranja que se muestra. El plano tiene una textura de agua y el agua se mueve de derecha a izquierda. Para la noria, el cangilón y el rodillo central tienen la textura que se muestra en las capturas mientras que los tableros de las aspas son de color marrón.

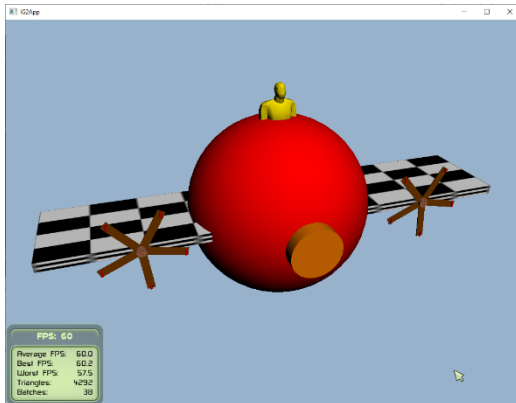
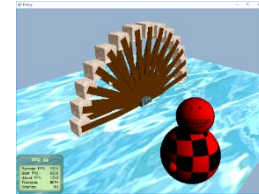


**21.** Cambia el script del material de la siguiente manera. Para el muñeco, elimina la nariz y haz que la cabeza y el cuerpo tengan las texturas que se ven en las capturas adjuntas. El agua del plano se muestra azul claro.



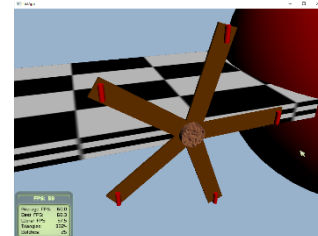
**22.** Programa **keyPressed()** de la clase **Muñeco** de manera que deje de actuar su **frameRendered()** y el muñeco se mueva hacia adelante/atrás con las teclas **UP/DOWN**. A su vez, las teclas **LEFT/RIGHT** permiten hacerle una pequeña rotación que se hace visible porque el ombligo gira alrededor del eje Y.

23. Programa el paso de mensajes que se describe a continuación. Al pulsar la tecla **r**, el plano manda un mensaje que hace que el agua del plano se pare, la noria se detenga y el muñeco se ponga de color rojo, según se muestra en la captura adjunta.

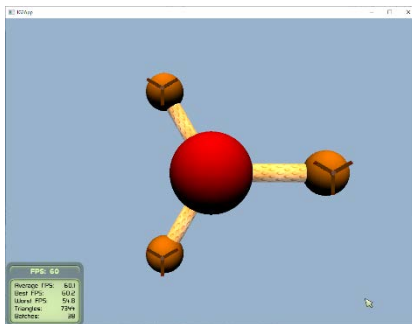


24. **[Avión caza drones]** Define la clase **Avion** cuyos objetos se renderizan como el que se muestra en la captura. Un avión está compuesto por una esfera roja, dos alas que son sendos cubos escalados apropiadamente, un morro cilíndrico naranja, un piloto ninja amarillo, y dos hélices que son elementos de la clase **AspasNave** y que están situadas en la mitad del borde de las alas. Un detalle de las alas se muestra en la captura adjunta.

Los objetos de la clase **AspasNave** están formados por un cilindro central rodeado por cinco objetos de la clase **Aspa**. Un objeto de esta clase a su vez está formado por un cilindro rojo y un tablero marrón.



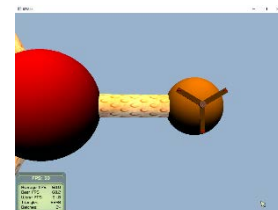
Las texturas deben ser las que se muestran y los colores deben aproximarse lo más posible a los que aparecen. El cilindro rojo de las aspas debe mantenerse vertical durante el giro de estas. El número de aspas de un objeto de **AspasNave** es parámetro de la constructora. Las hélices del avión giran con el movimiento de este. El morro del avión marca la dirección del movimiento del avión.



25. **[Dron nodriza]** Define la clase **Dron** cuyos objetos se renderizan como en la captura. Un dron está formado por una esfera roja rodeada de tres brazos. Los brazos de un dron son objetos de la clase **BrazoDron** que se renderizan como en el detalle de la captura.

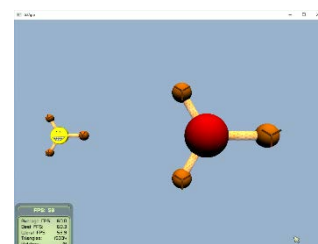
El brazo de un dron está formado por un cilindro terminado en una esfera naranja que lleva encima un

objeto de **AspasNave** con tres aspas.

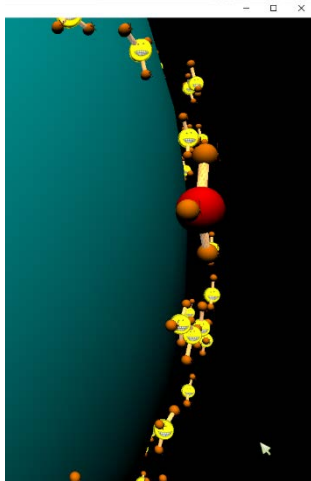


De nuevo, las texturas deben ser las que se muestran y los colores deben aproximarse lo más posible a los que aparecen. El número de brazos de un dron es un parámetro de la constructora. Las aspas de un dron giran con el movimiento de este. Uno de los tres brazos del dron es ligeramente más largo que los otros dos y marca la dirección del movimiento del dron.

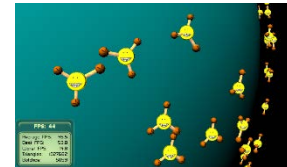
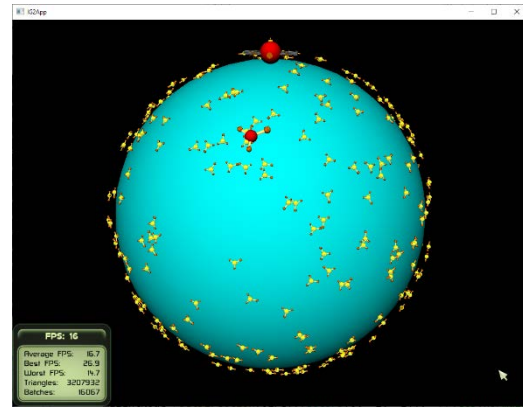
26. **[Drones avispa]** Los drones avispa son drones que tienen la carita feliz con efecto billboard adosada a la esfera roja, tal como se muestra en la captura. El tamaño relativo de un dron nodriza y un dron avispa también se muestra en la captura.



27. Crea una escena con fondo negro que contenga un planeta, que es una esfera de color cian, un avión caza drones situado en el polo norte de la esfera, un dron nodriza y un enjambre de 400 drones avispa.



La orientación de los drones (avispa y nodriza) debe ser consistente con el movimiento que estos tienen sobre la superficie de la esfera, es decir, los drones se encuentran, en todo momento, en un plano tangente al planeta. Esto es lo que se intenta mostrar en los dos detalles adjuntos.



Prueba dos formas de posicionamiento inicial para los drones y elige la que prefieras. Una sitúa el dron nodriza y todos los drones avispa en un punto fijo de la cara visible del planeta (por supuesto, este punto ha de ser distinto al polo norte, que es donde se encuentra inicialmente el avión). La otra forma sitúa todos los drones de la escena en puntos aleatorios del planeta (de nuevo, distintos al polo norte).

28. Añade dos eventos de teclado a la clase **Avion**. Con la tecla **h**, el avión describe un círculo máximo sobre la esfera, en la dirección que indica el morro del avión. Con la tecla **j**, el avión gira sobre sí mismo. Para ambos movimientos puedes usar cualquiera de las tres técnicas explicadas en clase (reposicionamiento, nodo ficticio o truco).

*Queridos sergio y laura del futuro: las helices del avion giran cuando se mueve, acordaros par de frikis*

29. Añade el método **frameRendered()** a la clase **Dron** de forma que el dron se mueva de manera autónoma alrededor del planeta. Para ello, por una parte, el dron describe círculos máximos sobre la esfera, en la dirección de su brazo largo. Por otra, y cada dos segundos, el dron se para un breve lapso de tiempo de menos de dos segundos y gira sobre sí mismo, para cambiar de dirección. Antes de iniciar este giro, decide aleatoriamente si lo va a hacer de forma horaria o anti-horaria. Una vez realizado el giro sobre sí mismo, el dron continúa su giro alrededor del planeta. Cada dron dispone de un temporizador **Ogre::Timer\* myTimer** y usa la función que los temporizadores tienen **getMilliseconds()** para tomar tiempos. Incluye **OgreTimer.h** para poder disponer de temporizadores. Como en el caso del avión, para los movimientos de rotación de los drones sobre sí mismos y alrededor del planeta puedes usar cualquiera de las tres técnicas explicadas en clase.

30. Añade el evento de teclado **h** a la clase **IG2App** de forma que, cuando movamos el avión alrededor del planeta mediante la tecla **h**, se vayan eliminando del enjambre aquellos drones avispa con los que el avión colisione. Para averiguar si hay colisión, usa la operación que posee todo **SceneNode\* node** y que devuelve la caja (alineada con los ejes) que lo contiene: **AxisAlignedBox aab = node->\_getWorldAABB()**. La caja intersección de dos de estas cajas **aab1** y **aab2** se calcula mediante la operación

**aab1.intersection(aab2).** Haz que cuando el avión colisione con un dron avispa, el dron desaparezca de la escena. La manera sencilla de hacerlo es que el dron pase a ser invisible.

31. **[Opcional]** Programa la colisión de forma que se elimine de la escena el nodo que contiene el dron avispa que ha colisionado. Averigua si proceder de esta manera permite aumentar el tamaño del enjambre sin que el tiempo de respuesta de los elementos de la escena se ralentice demasiado.
32. **[Opcional]** Usa el **TrayManager** de **IG2App** para mostrar un tray en la esquina inferior derecha de la ventana que contenga el número de drones avispa que quedan (visibles) en el grafo de la escena.
33. Añade un control al “juego” de forma que, cuando no queden drones avispa (visibles), la esfera central del dron nodriza pase de color rojo a amarillo.