

## Proyecto final

### Información general

El proyecto de esta asignatura consistirá en realizar un diseño e implementación de una aplicación sencilla utilizando la librería **OpenSceneGraph**, así como la implementación de parte de la aplicación a más bajo nivel utilizando **OpenGL**.

### Objetivos

Esta aplicación deberá mostrar una ventana gráfica en la que se visualicen dos relojes en 3D:

- Uno analógico.
- Uno digital.

Estos relojes se mostrarán uno al lado del otro. La aplicación deberá permitir la exploración mediante el ratón de la escena.

En el analógico deberán aparecer los números del 1 al 12 dispuestos de la forma habitual. Las agujas del reloj deben indicar la hora del sistema. Para ello podéis utilizar el método **GetSystemTime**. La manecilla del segundero es opcional. La hora posición se actualizará en cada actualización del grafo. La forma y materiales, tanto del reloj como de las manecillas y números es libre.

En el caso del reloj digital, los números deberán ser del tipo 7 segmentos, aproximadamente, como se muestra en la siguiente figura. Deberá dibujarse un reloj digital donde se muestre hora con dos cifras y los minutos separados por dos puntos. Las cifras no deben ser planas y deberán tener un grosor a gusto y definido por el diseñador, así como su tamaño. El que se muestren los segundos es opcional. La forma exacta de cada segmento queda de nuevo a gusto del diseñador y programador de la aplicación. Será necesario iluminar la escena, así como la asignación de distintos materiales a los elementos que aparecerán. El diseño queda en manos del programador. El texturizado de los modelos es opcional.

### Implementación

Para implementar la aplicación se utilizará la clase `osgViewer::viewer` para gestionar todo lo relacionado con las ventanas y los eventos de ratón y teclado.

Aunque es posible utilizar Geodes en OSG que dibujan texto, se le pide al alumno que implemente el dibujo de éstos elementos mediante la especialización de clases que hereden de la clase **`osg::Drawable`**. De este modo podrán definir sus propios métodos **`drawImplementation`** en los que incluirán sus propias instrucciones OpenGL que dibujen los números de forma correcta. El método a reimplementar en la clase especializada es:

### **void Drawable::drawImplementation(osg::RenderInfo& renderInfo) const**

Se recuerda a los alumnos que será necesario describir tanto la geometría como las normales y las coordenadas de texturas (si se va a texturizar) de cada punto del modelo.

En el caso de texturizar (opcional) se aconseja realizarlo desde OSG.

El siguiente código muestra un ejemplo de cómo se debe heredar de la clase `osg::Drawable`.

```
Class CifraDrawable:public osg::Drawable
{
    public:
        CifraDrawable(int v) {...}

        CifraDrawable(const CifraDrawable & cifra, const osg::CopyOp&
copyop=osg::CopyOp::SHALLOW_COPY): osg::Drawable(cifra,copyop) {}

        META_Object(RelojApp, CifraDrawable)

        void Drawable::drawImplementation(osg::RenderInfo& renderInfo) const
    }
}
```

Se aconseja el uso de nodos **osg::Switch** para gestionar la visualización de cada nodo número en el reloj digital.

### **Entregables**

Se deberá entregar en el aula virtual un paquete con:

- El código de la aplicación. Únicamente el fichero de código `main.cpp` donde deberán estar todas las clases que se hayan implementado
- El fichero ejecutable.
- Memoria del proyecto donde, entre otros, se explique:
  - Qué problemas se han encontrado en la realización y como se han resuelto, si se han podido llegar a resolver y, si no ha sido así, qué pruebas se han realizado y no han dado buen resultado.
  - El gráfico de escena que se ha implementado para cada uno de los relojes
  - Qué funciones de Callback se han implementado y a qué nodos se aplican.