

# Editor de Escenarios

**Asignatura:** Diseño Asistido por Computador

**Curso:** 4º Ingeniería Informática

**Alumno:** Javier Trasierra Jiménez

## Descripción del Problema

Para la realización de esta práctica vamos a considerar la creación de un pequeño editor de escenarios, en el que habrá diversos objetos disponibles (casas, coches, árboles, vallas, farolas), visibles mediante un panel, y seleccionables mediante tal. Dichos objetos podrán ser colocados en un escenario, inicialmente vacío. No se permitirán superposiciones de objetos, es decir, un objeto no podrá ser colocado si hay otro previamente colocado en dicha posición. Una vez colocados los objetos estos podrán ser seleccionados, y entonces ser borrados, o ser cambiados de color. Además se tendrá la posibilidad de guardar el escenario en un archivo, y posteriormente recuperarlo.

## Estructuras de Datos Utilizadas

Las estructuras de datos principales que se van a utilizar van a ser 2.

1. La primera será un vector con una longitud = número de objetos de cada tipo disponibles en el editor de escenarios. Cada casilla de estas contendrá el número de objetos de ese tipo ya puestos sobre el escenario:

```
int NOobjetos[SUBVENTANAS]; // Contadores para cada tipo de objeto
```

La Constante subventanas, será el número de objetos distintos de los que consta el editor, en este caso 5. Esta constante se llama así porque cada objeto visible en el panel, tendrá su propia subventana de visualización. La asociación de número-objeto será la siguiente:

```
0 → Casa
1 → Coche
2 → Árbol
3 → Valla
4 → Farola
```

2. La segunda realmente se divide en 5, y será una estructura distinta (con distinto nombre, número y tipo de variables) para cada tipo de objeto.

Antes hay que decir que para la realización de estas estructuras, de ha definido una enumeración de colores, que definirán la posición de una vector que contendrán las componentes RGBA para cada color:

```
// definicion de colores
typedef enum {
    rojo, verde, azul, negro, blanco, amarillo, gris_oscuro,
    gris_claro, marron, naranja, rosa, purpura, celeste
}mis_colores;
```

### 2.1. Estructura para el objeto Casa

La estructura para la Casa, tendrá seis variables. Las tres primeras, definirán la posición del espacio en que está situada. Una que definirá la rotación que tiene el objeto sobre su eje vertical (definirá la orientación del objeto). Por último dos más que definirá el color del tejado y de la fachada de la casa.

```
typedef struct{
    float x, y, z, rot_y;
    mis_colores color_techo, color_fachada;
}TCa;
```

### 2.2. Estructura para el objeto Coche

La estructura para el Coche tendrá cinco variables. Las tres primeras, definirán la posición del espacio en que está situado. Una que definirá la rotación que tiene el objeto sobre su eje vertical (definirá la orientación del objeto). Por último una más que definirá el color de la carrocería del coche.

```
typedef struct{
    float x, y, z, rot_y;
    mis_colores color;
}TCo;
```

### 2.3. Estructura para el objeto Árbol

La estructura para el árbol tendrá sólo tres variables, las cuales definirán la posición del espacio en que está situado.

```
typedef struct{
    float x, y, z;
}TAr;
```

### 2.4. Estructura para el objeto Valla

La estructura para la Valla tendrá cinco variables. Las tres primeras, definirán la posición del espacio en que está situada. Una que definirá la rotación que tiene el objeto sobre su eje vertical (definirá la orientación del objeto). Por último una más que definirá el color de la valla.

```
typedef struct{
    float x, y, z, rot_y;
    mis_colores color;
}TVa;
```

## 2.5. Estructura para el objeto Farola

La estructura para la Farola, tendrá seis variables. Las tres primeras, definirán la posición del espacio en que está situada. Una que definirá la rotación que tiene el objeto sobre su eje vertical (definirá la orientación del objeto). Por último dos más que definirá el color del tubo y la del foco de luz.

```
typedef struct{
    float x, y, z, rot_y;
    mis_colores color_tubo, color_luz;
}TFa;
```

Una vez definidas las estructuras, definiremos unos vectores de dichas estructuras, para tener información de las instancias de cada objeto:

```
TCa TipoCasa[MAX_CASAS];
TCo TipoCoche[MAX_COCHES];
TAr TipoArbol[MAX_ARBOLES];
TVa TipoValla[MAX_VALLAS];
TFa TipoFarola[MAX_FAROLAS];
```

## Manual de usuario

Inicialmente se abrirán dos ventanas, una a la izquierda que será la del escenario y otra a la derecha que será un panel con cinco subventanas. En cada una se dibujará un objeto, que permanecerá fijo, y que simularán botones.

Habrán dos modos principales, modo edición y modo visor. En el modo editor se podrán colocar objetos, y en el modo visor, se podrán ver desde diferentes ángulos. Dichos modos serán seleccionables con un menú que se abrirá cuando se pulse con el botón derecho del ratón (y no haya ningún “botón” del panel pulsado). También mediante dicho menú se tendrá la opción de guardar el escenario actual, o cargar un escenario previamente guardado. Para estas opciones, se pedirá por terminal el nombre del archivo. Cuando se guarde un archivo, se le pondrá automáticamente la extensión .ed (no hay que ponérsela), y cuando queramos cargar un archivo, habrá que poner el nombre del archivo, terminado en .ed. También se tendrá la posibilidad de limpiar el escenario de objetos, es decir cuando se seleccione esta opción el escenario quedará libre de objetos, quedará vacío.

- Modo editor:

Cuando se pinche con el botón izquierdo del ratón sobre alguno de los “botones” se cambiará el color de fondo de dicho botón y entonces ese será el objeto activo, (además se borrará el menú asociado al botón derecho del ratón). Ahora cuando la ventana activa sea la del editor, si pulsamos con el botón izquierdo del ratón en la ventana del editor, se pondrá el objeto activo en la posición marcada, aparecerá otro y así sucesivamente, hasta que se pulse otro “botón” o se pulse el botón derecho del ratón (entonces volverá a estar disponible el menú asociado al botón derecho), en cuyo caso

se pondrá el “botón” con su color de fondo original, y el objeto seleccionado, quedará deseleccionado.

Cuando haya un objeto activo, se podrá variar su orientación (girará sobre si mismo) si pulsamos los cursores, y en cuyo caso podrá ser colocado en el editor con dicha orientación.

Con las teclas A y D se podrá mover el escenario a la izquierda y derecha respectivamente.

Con las teclas W y S se podrá mover el escenario arriba y abajo respectivamente..

Con las teclas + y – se podrá hacer un zoom.

Cuando tengamos un objeto activo (“botón” del panel pulsado) y haya objetos colocados en el editor, si el objeto activo está sobre otro ya colocado, el objeto activo cambiará de color y esto significará que no es posible colocarlo en dicha posición.

- Modo Visor

Con las teclas A y D se podrá mover el escenario a la izquierda y derecha respectivamente.

Con las teclas W y S se podrá mover el escenario arriba y abajo respectivamente..

Con las teclas + y – se podrá hacer un zoom.

Con las teclas Z y X se hará una rotación sobre el eje Z.

Con los cursores se harán rotaciones sobre los ejes X e Y.

- Ambos modos (da igual estar en modo editor o modo visor)

Existe la posibilidad de seleccionar un objeto del editor (siempre que no haya ningún “botón” del panel pulsado), pinchando con el botón izquierdo del ratón sobre dicho objeto, entonces el objeto sobre el que se ha pinchado, cambiará de color (significa que ha sido seleccionado). Ahora si pulsamos el botón central, aparecerá un menú, (distinto para cada objeto), con una parte común, que es la posibilidad de borrar el objeto, y una parte distinta, que será la posibilidad de cambiar el color del objeto o partes del objeto, según.

Cuando esté un objeto seleccionado, este también podrá borrarse al pulsar la tecla B.