Tutorial Objeto

Resumen

Dado que orx es una plataforma impulsada por datos, necesitamos de solo dos líneas de código para crear una vista y un objeto. Todas sus propiedades son definidas en el fichero de configuracióon (01_Object.ini).

La vista está asociada con una camara la cual es creada implicitamente a travéz de la información dada en el fichero de configuración. Además en el fichero de configuración, puedes también fijar su tamaño o posición o alguna otra propiedad del objeto como su color, escala, ángulo de rotación, animaciones, propiedades físicas, y más. También puedes definir valores aleatorios para todas estas propiedades sin escribir una sola línea de código.

En futuros tutoriales veremos cómo generar un objeto complejo con jerarquías o incluso una escena completa (todos los fondos y paisajes por ejemplo) con una sola línea de código.

Por ahora puedes descomentar varias de las líneas de 01_Object.ini, cambiar sus valores y luego relanzarlo. Para una lista de opciones exhaustiva, por favor échele un vistazo a CreationTemplate.ini.

Detalles

Creating an object is really simple. However, we first need to make sure that we have loaded the config file where all the object's properties are defined. We also want to display the created object, through a viewport/camera set.

Don't panic! All this is really easy. 

In this tutorial, we are loading a config file which is in our parent directory. As you may see, in this case, all the executables are in their own child directory, depending on their build type (mingw, msvs2005, msvs2008, etc...) and we didn't want to duplicate the corresponding config files everywhere. [1)]

In our case, loading the config file is done like with this line:

```
orxConfig_Load("../01_Object.ini");
```

We then create our viewport. Note that the camera creation is automatically done given the config information stored for this viewport.

```
orxViewport_CreateFromConfig("Viewport");
```

We're almost done. We only need to create our object now!

```
orxObject_CreateFromConfig("Object");
```

That's it! The object has been created and will be displayed as long as it is in the frustum of our camera.

Now, as we're using Orx's default launcher, we need to declare our plugin entry point (here our `Init` function). This is easily done with a macro.

```
orxSTATUS Init(){...}

orxPLUGIN_DECLARE_ENTRY_POINT(Init);
```

As orx is data driven, we don't need to manually load any data, such as a sprites. Everything is handeld for us, with a data manager that will make sure sprites won't be duplicated in memory and freed when not used anymore.

If you look at the config file, in the [Object] section, you'll see that you can specify all the object's properties, such as: graphic (sprite), pivot, color, alpha, physics properties, position, rotation, scale, tiling (repeat), animation, visual FX, etc...

Don't worry, all this will be covered in further tutorials.

Now we have an object, we need to learn how to interact with it. This brings us to our second tutorial: Clock.

Recursos

Código fuente: 01_Object.c

Fichero de configuración: 01_Object.ini

1)
However, if your config file name matches your executable and is in the same folder, it'll be loaded automatically.

From:
https://www.orx-project.org/wiki/ - **Orx Learning**

Permanent link:
https://www.orx-project.org/wiki/es/orx/tutorials/object?rev=1249405561

Last update: **2025/09/30 17:26 (14 months ago)**