Tutorial de Vista y Cámara

Sumario

Ver previamente tutoriales básicos para más información sobre creación básica de objetos, manejando el reloj, jerarquía de fotogramas y animaciones.

Este tutorial muestra como usar múltiples vistas con múltiples cámaras.
Cuatro vistas son creadas aquí.

La esquina superior izquierda una (`Viewport1`) y la esquina inferior derecha una (`Viewport4`) comparten la misma cámara.
Para archivar este, necesitamos usar el mismo nombre en el fichero de configuración para la cámara.

Furthermore, when manipulating this camera using left & right mouse buttons to rotate it, arrow keys to move it and left control & left shift to zoom it, the two viewports associated with this camera will both be affected.

The top right viewport (`Viewport2`) is based on another camera (`Camera2`) which frustum is narrower than the first one, resulting in a display twice as big. You can't affect this viewport at runtime in this tutorial.

The last viewport (`Viewport3`) is based on another camera (`Camera3`) which has the exact same settings than the first one.
This viewport will display what you originally had in the `Viewport1` & `Viewport4` before modifying their camera.
You can also interact directly with the first viewport properties, using WASD keys to move it and 'Q' & 'E' to resize it.

NB: When two viewports overlap, the oldest one (ie. the one created before the other) will be displayed on top.

Lastly, we have a box that doesn't move at all, and a little soldier whose world position will be determined by the current mouse on-screen position.
In other words, no matter which viewport your mouse is on, and no matter how the camera for this viewport is set, the soldier will always have his feet at the same position than your mouse on screen (provided it's in a viewport).

Viewports and objects are created with random colors and sizes using the character '~' in config file.

NB: Cameras store their position/zoom/rotation in an orxFRAME structure, thus allowing them to be part of the orxFRAME hierarchy as seen in the frame tutorial.
As a result, object auto-following can be achieved by setting the object as the camera's parent.
On the other hand, having a camera as parent of an object will insure that the object will always be displayed at the same place in the corresponding viewport [1].

Detalles

Recursos

1)

very useful for making HUD & UI, for example

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

Permanent link:
https://www.orx-project.org/wiki/es/orx/tutorials/viewport?rev=1330547052

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