Autosized Displays and Objects

With the Borderless Window support and Fullscreen support in Orx, it is possible in config to tell the Camera's Frustum to be the same size as the Display.

To get Borderless Window mode working, you don't specify any `ScreenHeight` or `ScreenWidth`.

So how do you tell the frustum where to get the specified screen size? By using commands in your properties to get the current real screen dimensions.

Here is an example:

```
[Display]
Title           = Display Test
Decoration      = false
VSync           = true

[MainCamera]
FrustumWidth    = % get Display ScreenWidth
FrustumHeight   = % get Display ScreenHeight
FrustumFar      = 2
FrustumNear     = 0
Position        = (0, 0, -1)
```

See the magic? For the `FrustumWidth`, `% get Display ScreenWidth` is used to assign the current screenwidth.

So any time the application opens onto a different screen resolution with Borderless Window mode, the camera's frustum will be set to whatever the current screen dimensions are.

Objects

Objects can use the same feature, but this time, we'll use a different command to set the Object's Size property. Imagine you wanted a large object to act as a background that needs to automatically size to the same dimensions as the screen:

```
[SceneObject]
Size = % > texture.find screen, texture.getsize <
Alpha = 0.3
Graphic = @
Texture = pixel
Pivot = center
```

The command here locates the screen's texture used for overall rendering, gets the vector, and assigns it to the `Size` property for the `SceneObject`.

And that's it. Two ways to dynamically get and use the current screen size for objects and camera frustums rather than having to specify an exact screen resolution.

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

Permanent link:
https://www.orx-project.org/wiki/en/tutorials/display/autosized_displays_and_objects

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