Colliding ScrollObjects with Orx Objects

ScrollObject classes provides a lot of functionality and organisation for projects by wrapping the existing Orx object.

For testing collisions between one ScrollObject and another, the class provides the `OnCollide` function eg:

```
orxBOOL Truck::OnCollide(ScrollObject *_poCollider,
                         orxBODY_PART *_pstPart,
                         orxBODY_PART *_pstColliderPart,
                         const orxVECTOR &_rvPosition,
                         const orxVECTOR &_rvNormal)
{
    if (_poCollider == orxNULL) {
        return orxTRUE;
    }

}
```

The `OnCollide` in this `Truck` ScrollObject class will detect collisions with another ScrollObject. This expects that any Orx object in your game be wrapped in a ScrollObject class. However, this is not necessary.

You may have many simple Orx objects in your game like Rocks, or Ice, bullets that you wish to collide with your Truck. But it would be very heavy handed to have to create a ScrollObject class for each, especially if the classes are basically empty.

Fortunately, Orx allows you to declare an object in your config to be a ScrollObject. If you had a Rock object defined as:

```
[Rock]
Graphic = RockGraphic
```

You can convert it with: `@ScrollObject`.

```
[Rock@ScrollObject]
Graphic = RockGraphic
```

From here, `_poCollider` in the `OnCollide` function will be able to contain the `Rock` ScrollObject if it collides with the Truck.

Brilliant.

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

Permanent link:
https://www.orx-project.org/wiki/en/tutorials/orxscroll/colliding_scrollobjects_with_orxobjects

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