3D Node Transformations

3D Node Manipulation Example

To illustrate basic node transformation operations, the *NodeManipulationWidget_3D* and the *DisplayPositionWidget3D* part of the Tutorial Widgets can be used. An usage example for *NodeManipulationWidget_3D* widget is presented in **NodeManipulationSolution_3D** solution provided in the content folder of *cgi_studio_player*.

Please consider:

- The widget property *Node* is derived from a base class and provides the node associated to the widget (e.g. the root node for content managed by the widget).
- Only if the widget is *enabled* (this property is also derived from a base class), the setter methods of the widget properties will take effect.
- If a widget property setter changes a node property of the associated node, the visual effect can be seen, but note that the static node properties maintained by SceneComposer are not changed accordingly!
- Widget dynamics are never reflected to static scene structure. It is recommended to use SceneComposer only for property configuration and dynamics only in the Player.

Translations

Once the NodeManipulationWidget_3D is linked to a node from the static scene tree and enabled, the position coordinates can be set in the Player for this property to change the position.

```
        // Set node position
        node->SetPosition(m_position);
        // end Set node position
```

Use TranslateX property to move the node on the x axis with a delta value only.

Candera::Transformable::Translate adds the given Vector to the actual position .

```
       // Translate node on x axis
        node->Translate(Vector3(m_translateX, 0.0F, 0.0F));
        // end Translate node on x axis
```

Rotations

Similar to translation, rotation can also be applied via Candera::Transformable::SetRotation() or Candera::Transformable::Rotate().

```
        // Set node rotation
        node->SetRotation(m_rotation);
        // end Set node rotation
```

Scaling

Again there is also a Candera::Transformable::SetScale and Candera::Transformable::Scale method.

```
        // set node scale
        node->SetScale(m_scale);
        // end node scale
```

Pivot Point

It is possible to change your pivot point with the Candera::Transformable::TranslatePivotPoint (or Candera::Transformable::SetPivotPoint) method. The pivot point can be used to define e.g. the rotation point that the node should rotate around. It affects the scaling too.

How to get the screen space coordinates of a node

With the DisplayPositionWidget3D you can retrieve the screen coordinates of an associated node. Please consider that a node and a camera have to be selected in the widget. In the solution the widget is disabled by default. To see the output of the node's display position, enabling the widget and log filter (via CgiAppLog -> Info) in SceneComposer is mandatory. To get the screen space coordinates of an associated node you need the function Candera::Math3D::TransformPointToScreen, which transforms a point in world space into a point in screen space.

- First you need the world position of your node. With Node::GetWorldPosition you get the screen coordinates as output.

  ```
          Vector3 worldPosition = node->GetWorldPosition();
  ```

- With the world position of your node and with your camera you can transform the point to screen space and you retrieve the display position as a Candera::Vector2.

  ```
          Vector2 displayPos;
          static_cast<void>(Math3D::TransformPointToScreen(worldPosition, camera, displayPos)
  ```

- displayPos now contains the X and Y screen space coordinates of the selected node.

Revision #3
Created 2023-02-21 06:17:51 UTC by Tsuyoshi.Kato
Updated 2024-07-31 23:18:52 UTC by Tsuyoshi.Kato