RhinoTerrain
Terrain modeling & City builder software

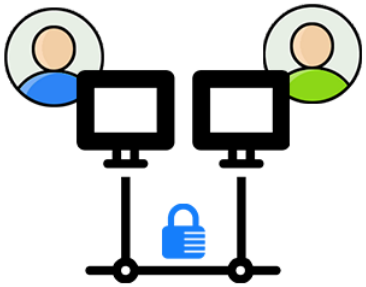
3D CURSOR DEMONSTRATOR

RhinoTerrain *Terrain modeling & City builder software*
SAS au capital de 129 630 € - RCS EPINAL 502 281 454

Siège social : 29 rue de l'Usine, 88 120 Le Syndicat – France
APE 6201Z - N° TVA intracommunautaire FR 08 502 281 454

Permanent licenses
Our software licenses never expire.
Updates within the same minor version are free.
You can take out a maintenance contract: this allows you to freely install major releases and access support.

Our software licenses can be bound to one computer.



Licenses can also be shared across your personnal network or by the use of **Cloud Zoo** from **McNeel**

Our software is available at a reduced price for students, teachers (Educational license) and universities (LAB licenses).

TABLE OF CONTENTS

1 CONTENT

This package contains the 3D cursor demonstrator developed by **RhinoTerrain SAS**.

It consists of:

- RT_3DCursor.rhp: the Rhinoceros V7 plug-in
- ftd2xx.dll: a DLL required by the plug-in
- RT_3DCursor_Sample_01.3dm: an example of a 3D scene

2 INSTALLATION

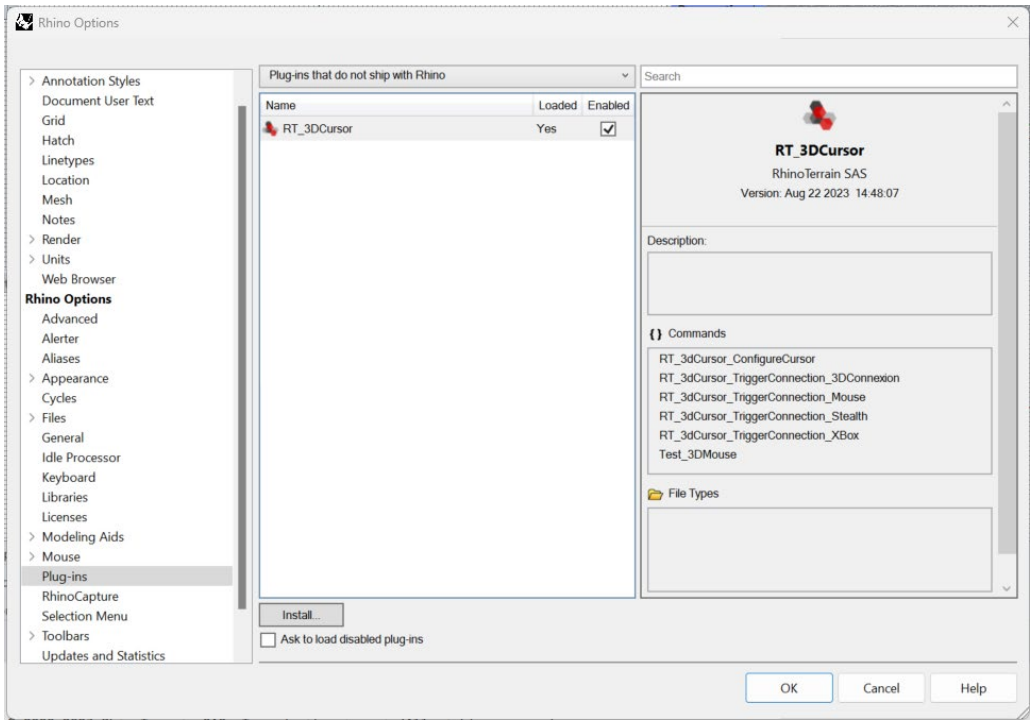
Copy the files `RT_3DCursor.rhp` and `ftd2xx.dll` wherever you like. Just make sure they are in the same directory.

In **Rhinoceros 7**, from the "*Tools*" menu, click on "*Options...*"

In the "*Plug-ins*" section, click on the "*Install*" button.

Browse to the directory where you copied the `RT_3DCursor.rhp` file, select it and click the "*Open*" button.

You should then see the "RT_3DCursor" plug-in appear in the list of installed plug-ins.



3 COMMANDS

There are 5 commands available:

- RT_3DCursor_ConfigureCursor
- RT_3DCursor_TriggerConnection_3DConnexion
- RT_3DCursor_TriggerConnection_Mouse
- RT_3DCursor_TriggerConnection_Stealth
- RT_3DCursor_TriggerConnection_XBox

The "RT_3DCursor_ConfigureCursor" command

This command is used to modify the values of various 3D cursor parameters.

Parameters

- **Reset**
  Moves the cursor to just in front of the camera
- **OnClosest**
- **OnPrev**
- **OnNext**
- **StayOnObjects**
  Default value=`false`
  If the value is `false`: the cursor is free.
  If `true`: the cursor tries to stay on a surface
- **CursorType**
  Default value=`Cross3d`

  Used to select the type of cursor.
  The possible values are: `Point`, `Cross2d`, `Cross3d`, `Circle`, `Sphere`, `Cube` and `Diamond`.

  Depending on the cursor type selected, other parameters appear.

  If *CursorType*=`Point`
  - **Diameter**
    Default value=`0.3` : size according to the unit of the current document

  If *CursorType*=`Cross2d`
  - **Branch**
    Default value=`1.5` : length according to the unit of the current document
  - **InnerGap**
    Default value=`0.1` : size according to the unit of the current document

  If *CursorType*=`Cross3d`
  - **Branch**
    Default value=`1.5` : length according to the unit of the current document
  - **InnerGap**
    Default value=`0.1` : size according to the unit of the current document

If *CursorType*=`Circle`

- **Radius**
  Default value=`0.5` : radius according to the unit of the current document
- **PointSize**
  Default value=`0.1` : size according to the unit of the current document

If *CursorType*=`Sphere`

- **Radius**
  Default value=`0.5` : radius according to the unit of the current document
- **PointSize**
  Default value=`0.1` : size according to the unit of the current document

If *CursorType*=`Cube`

- **Side**
  Default value=`1.0` : length according to the unit of the current document
- **PointSize**
  Default value=`0.1` : size according to the unit of the current document

If *CursorType*=`Diamond`

- **HalfDiagonal**
  Default value=`0.5` : length according to the unit of the current document
- **PointSize**
  Default value=`0.1` : size according to the unit of the current document

- **Done**
  Stop the command. Equivalent to using the [`ESC`] or [`Return`] key.

The "RT_3DCursor_TriggerConnection_3DConnexion" command

This command starts the 3D cursor and connects it to a 3Dconnexion mouse.

The joystick moves the cursor. Depth is on the front to back axis, while moving the cursor up and down is done by pressing/pulling the joystick.

Turning the joystick rotates the camera around the cursor.

The "RT_3DCursor_ TriggerConnection_Mouse" command

This command starts the 3D cursor and connects it to the basic 2D Windows mouse.

- The 3D cursor follows the 2D cursor. The wheel changes the depth.
- **Right-click** to move the camera around the cursor.
- **Right-click + `[Shift]`** moves the camera so that the cursor is always under the mouse.
- **Right click + `[Ctrl]`** to zoom in and move the camera towards the cursor.
- Turning the wheel with **`[Shift]`** or **`[Ctrl]`** pressed also zooms in.

The "RT_3DCursor_ TriggerConnection_Stealth" command

This command starts the 3D cursor and connects it to a Stealth 3D mouse.

The cursor follows the movement of the mouse, and the **scroll wheel** is the depth control.

The "RT_3DCursor_ TriggerConnection_XBox" command

This command starts the 3D cursor and connects it to a Xbox pad.

- **Left joystick** to move the on-screen cursor.
- **LT/RT triggers** to control depth.
- **Right joystick** to rotate the cursor.
- **Left joystick + RB/LB** to move the camera.
- **Right joystick + RB/LB** to zoom in.
- Press the **Left joystick** to reset the cursor in front of the camera.
- **Back** (or **Select** for older pad) to toggle between free movement and moving on surfaces.
- **Up** and **down arrows** to move to next/previous surface.
- **A** button to simulate a left mouse click.
- **B** button to simulate a right click (/!\ avoid it).
- **Left** and **right arrows** to change cursor type (axis, circle, sphere, etc..)
- **X** and **Y** keys to change the size of the cursor.
- **X** and **Y** keys with **RB/LB** pressed to change the internal size of the cursor.

4 WHERE TO START

In **Rhinoceros 7**, load the file RT_3DCursor_Sample_01.3dm

Start the `RT_3DCursor_ConfigureCursor` command.

Define the 2 options as follows:

- **StayOnObjects**=`True`
- **CursorType**=`Circle`

Start the `RT_3DCursor_TriggerConnection_Mouse.`

Mouve the mouse on the 3D objects, in the *Perspective* view.

You'll notice that the 3D cursor 'sticks' to the surfaces.

This is particularly noticeable where 2 objects meet, as in the image below: