

Smorynes „WinForms“

The Stride game engine is used to integrate the Smorynes simulator into "WinForms". Stride currently includes a number of modules that simplify development.

For example:

<https://github.com/Doprez/Awesome-Stride?tab=readme-ov-file>

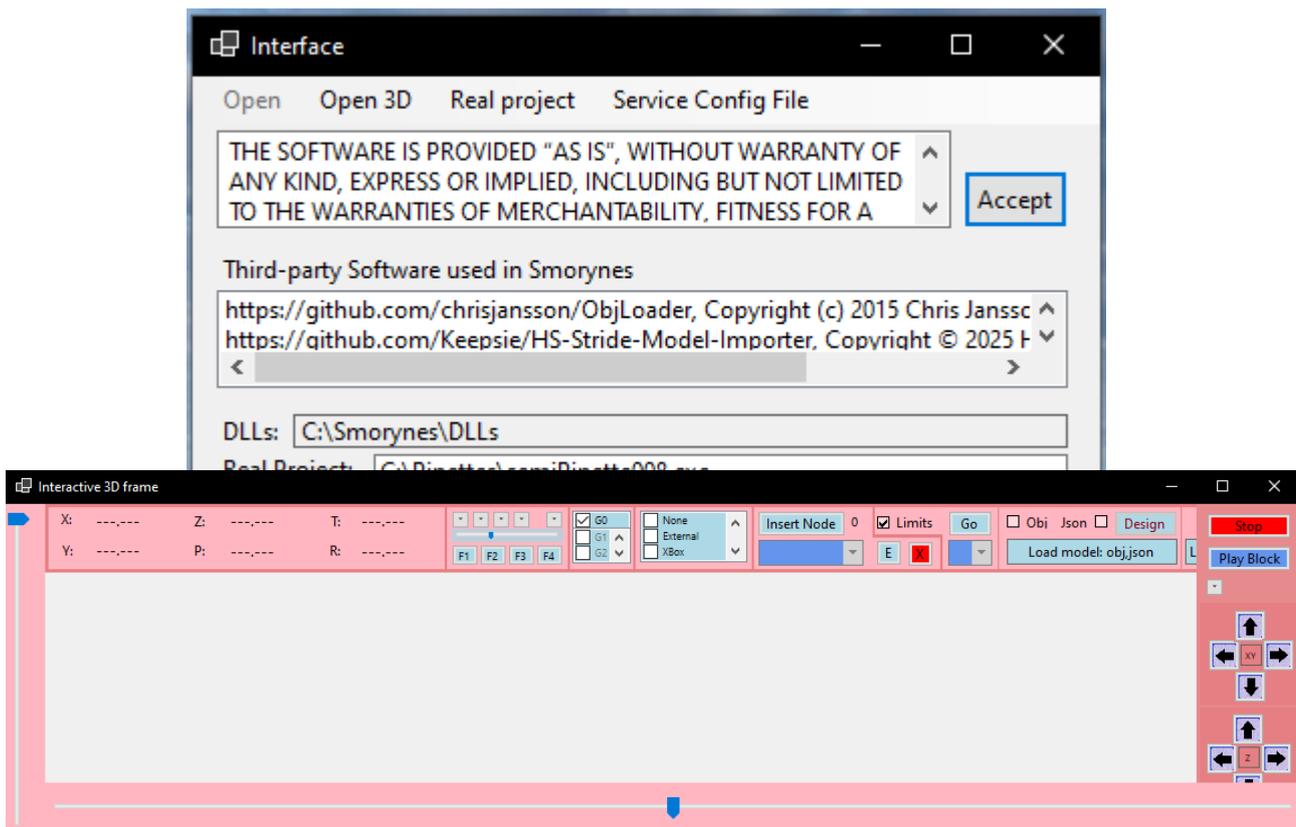
Embedded Stride section

<https://github.com/Doprez/avalonia-hosted-stride/tree/main>

<https://gist.github.com/EricEzaM/5797be1f4b28f15e9be53287a02d3d67>

Seems to me the latest

<https://github.com/luca-domenichini/StrideEmbedding>



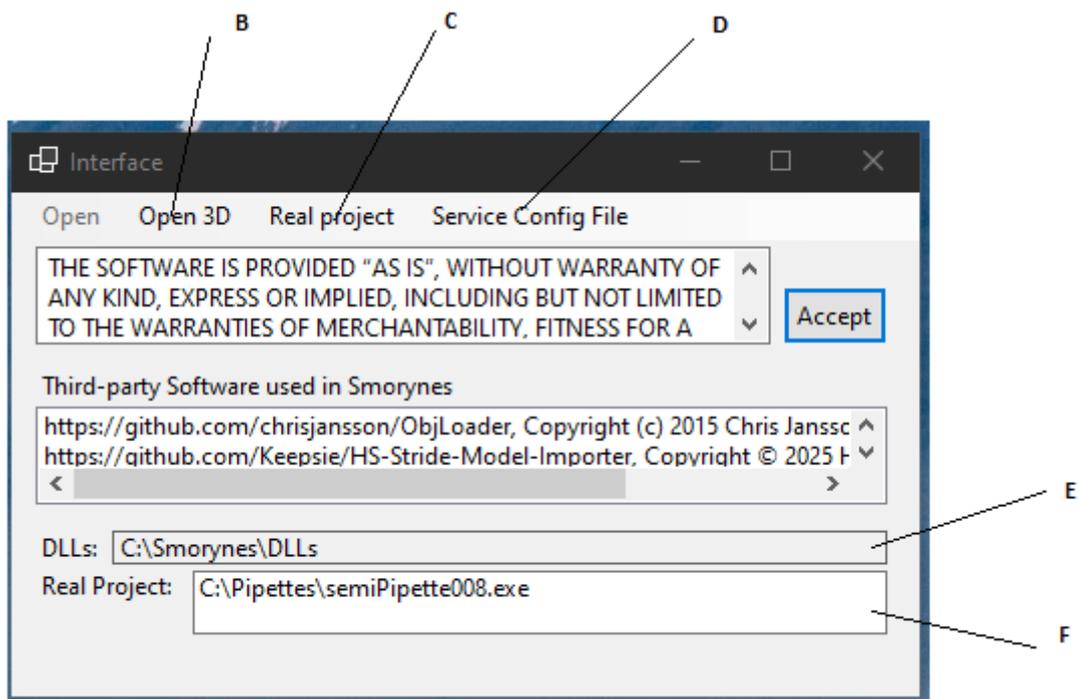
Inspiration from the above links enabled us to develop interactive 3D content in the form of WinForm.

Installation

Simply extract the downloaded file to a folder of your choice.
Run the WinForm3D.exe file. Set the folder for your user .dll files, if you have them ready.

Set the name of the real project if you are going to run it in parallel with the simulation

- B open simulator
- C open real project
- D set .DLL folder, set file of real project
- E folder of .dll files
- F start file of real project



The "WinForm" version is based on the WebGL version. The 3D scene is part of the "Form" window. The application is controlled using standard control components (buttons, list boxes, text boxes, etc.).

What is this program good for?

1. It allows you to supplement the documentation of previously developed machines with interactive controls, write application programs in the form of logical nodes, and quickly familiarize customers with the implemented functions.
2. It allows you to start and even complete the development of a new machine in close cooperation with the customer long before the first hardware parts are ready.
3. User DLLs allow you to integrate the control of hardware components as part of the simulation. This mainly involves running PLCs, sensor readings, etc. during simulated sequences.

The simulator contains several internal schematic models, which are sufficient for simple simulations in many cases. The range of encoders for the actual device can be set in the simulator in the <name>.smrenc file. This way, the positions of the sequences correspond to the actual device. For more information, see "Learn More" or the Documentation.

If a more accurate model is required, the OBJ format is supported. The Design item opens the parameter form. The parameters are saved in the <name>.json file.

User DLL

Simulator users can write their own DLLs. The link between the simulator and the DLL defines several methods:

```
// DLL instances
public string Init(string s)
{
}

// Play blocks
public string Run(string s)
{
}
```

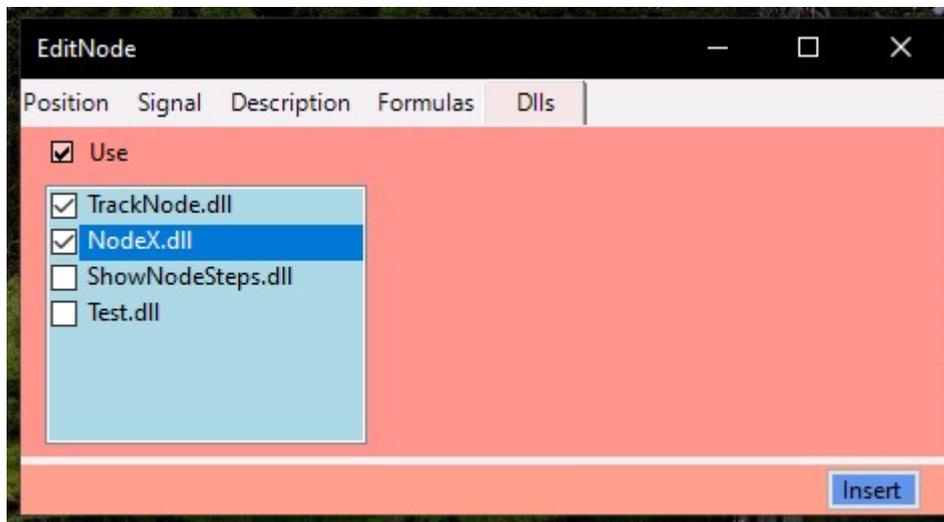
```

// Start move
public string MoveStart(string s)
{
    // where
    // string s = positionNode.titleNode.ToString() + ";" + index.ToString() + ";" +
    // delay.ToString() + ";" + time.ToString() + ";" + value.ToString();

    // index X = 0, Y = 1 ...
}

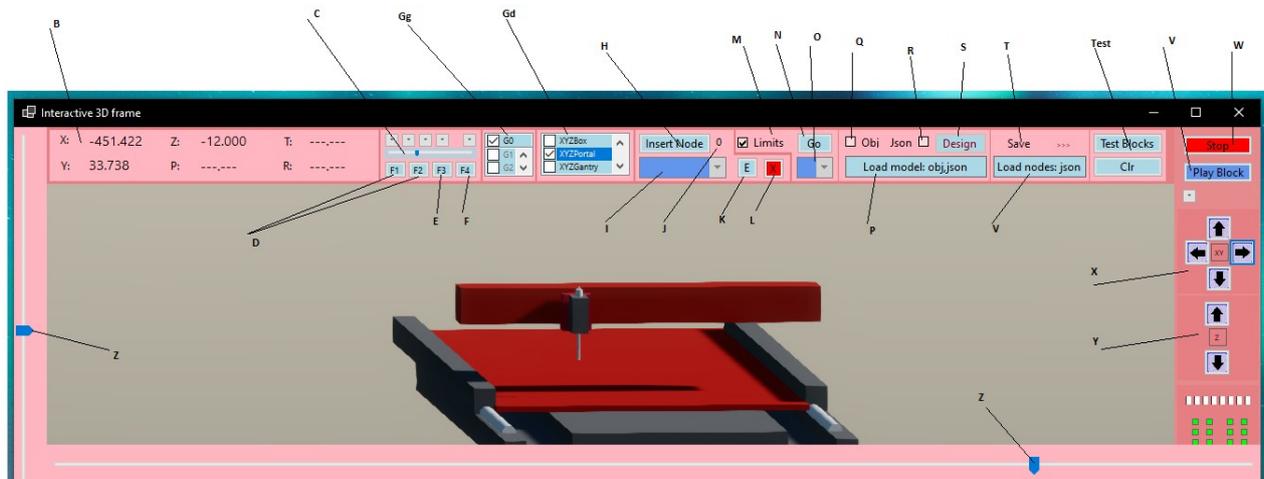
// Done move
public string MoveDone(string s)
{
    // where
    // string s = positionNode.titleNode.ToString() + ";" + index.ToString()
}

```



The user places their DLL in the folder. In the relevant nodes, they perform a DLL selection.

The following figure shows the internal model of the portal system with controlled XYZ axes.

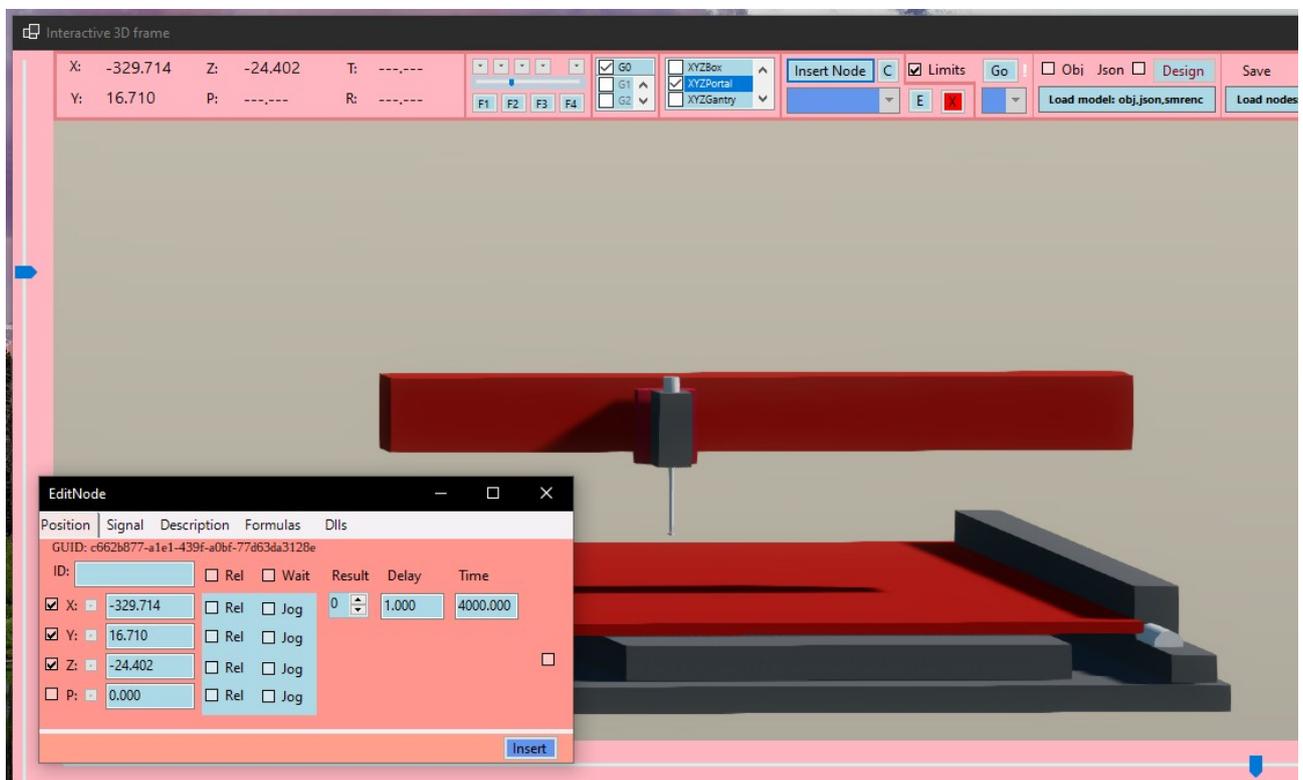


Function of the controls

- | | | | |
|----|--|---|------------------------------|
| B | Measuring system | D | Multifunction buttons F1, F2 |
| C | Speed setting in jog | E | Front camera |
| | | E | Top camera |
| Gg | Select gallery of devices
G0 open
G1, ... not yet open | | |
| Gd | List of internal or external models | | |
| H | Inserting a node into a positioning sequence | | |
| I | List of nodes in the sequence | | |
| J | Number of nodes in the sequence | | |
| K | Edit the selected node | | |
| L | Deleting a selected node | | |
| M | Enabling/disabling limits | | |
| N | Continue the sequence / after wait state / | | |

- O Sorting value
- P Paste obj, json files of model
- Q Checks if the obj file is inserted
- R Checks if the json file is inserted
- S Opens the external model parameters form
- Test Test sequences
- W Stop play
- V Play
- X X, Y movement
- Y Z, P movement
- Z Pan and tilt of a laboratory table or floor

Node Editor



The user programs/sets sequences in the node editor. Sometimes, a fairly extensive application can be described using only a "small" number of steps without the need to use a complex graphical node editor. The "Documentation" or "Learn More" on <https://industry40.online/> contains or will contain examples of Pick-Place, Dispensing, and Sorting operations. In "Jog" mode, move to the requested position. The "Insert Node" button saves the node to the working buffer and displays it in the list / I /. All parameters of the operations to be performed at this position must be set in the node. Re-editing of node parameters is invoked by the "E" button, deletion of the selected node by the red X button. The sequence is started by the Play button and stopped by the Stop button.

Positioning in Jog mode

The "X, Y movement" and "Z, P movement" controls allow you to position the axes of the device.

For "X, Y movement" left mouse button
For "Z, P movement" right mouse button

External machine model

If a user model is used, the simulation is more accurate. However, the model must be prepared. Parameter settings are described in the documentation and also in the "Learn more" section on <https://industry40.online/>. The parameter setting forms are similar to the Smorynes program version in "Unity."

However, for the supported .OBJ format, it is necessary to write down the structure of the model elements. In general, for simpler models, it is faster to fill in the .json file "manually." For one model, you can also have multiple .json files, and there, a minor "manual" modification is the easiest.

When using the form, the machine model <name>.OBJ must be inserted into the "Load model: obj, json, smrenc" field. The "Design" button opens the form. By selecting the appropriate

kinematics, predefined keywords are displayed in the right list box. Physical objects displayed in the left list box must be assigned to these words. After creating the structure and setting the parameters, click the Use button to refresh the model. You can then verify its functionality in jog mode or write several nodes and verify their functionality.

The OBJ format contains a material definition, but this is not used. The simulator contains several internal materials that can be assigned to individual objects in the form.

A .smrenc file can also be used to set the material at any time while the simulation is running.

If they are not assigned, the default materials are used as configured

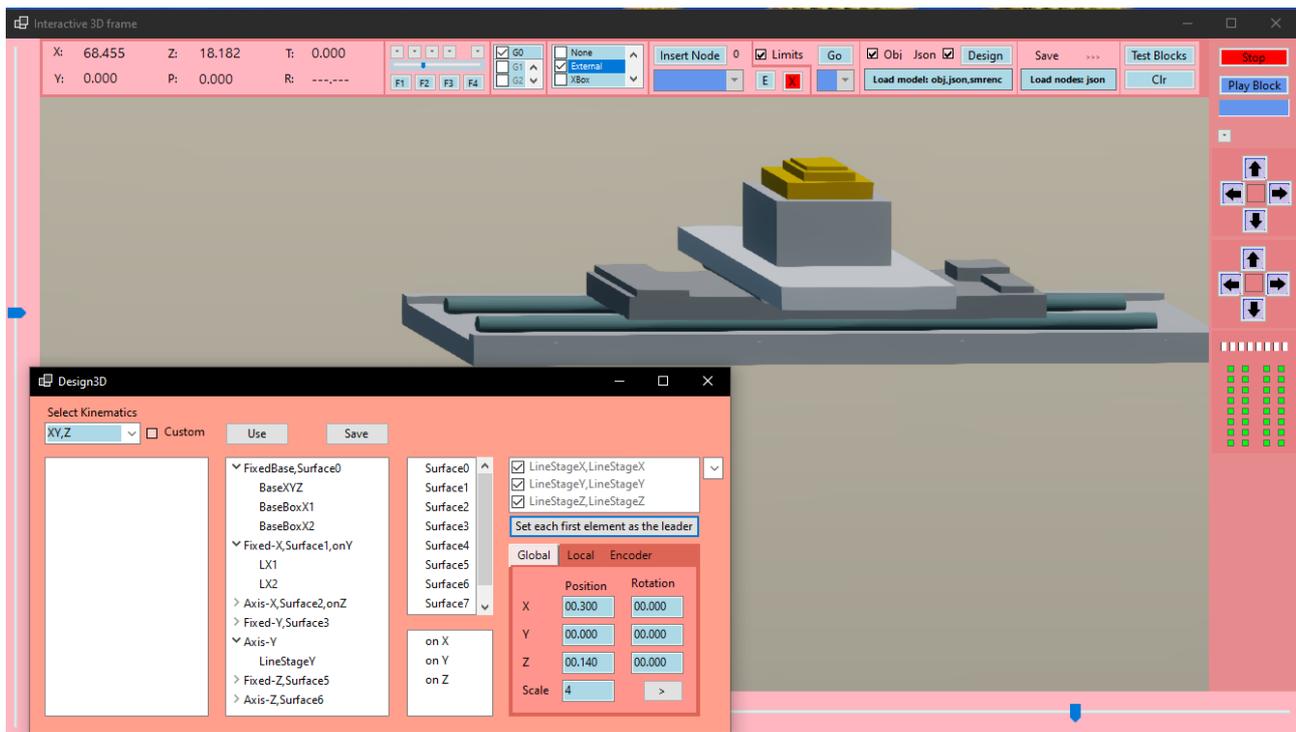
Examples of user models

The following sections will gradually introduce teaching models that are geared towards actual designs. They present specific design issues and can serve as basic frameworks.

1. Demo OBJ model
2. Dispensing of liquid or solid samples
3. Standard pick-and-place system

Demo OBJ model

The figure below shows a simple schematic model of XYZ. OBJ and Json files are contained in a distributed .zip file in the CAD folder. The structure of the links between the axes assumes that the axis has a fixed part and a movable part. The fixed part may be missing. The kinematic chain then defines the link between the axes. Details can be found in the documentation and in the Learn more section.



The example given is intended only to demonstrate the integration of an external model into the simulator.

Dispensing of liquid or solid samples

The practical application of the model requires connection to specific simulated technology. The model must therefore be supplemented with components that are commonly used in the

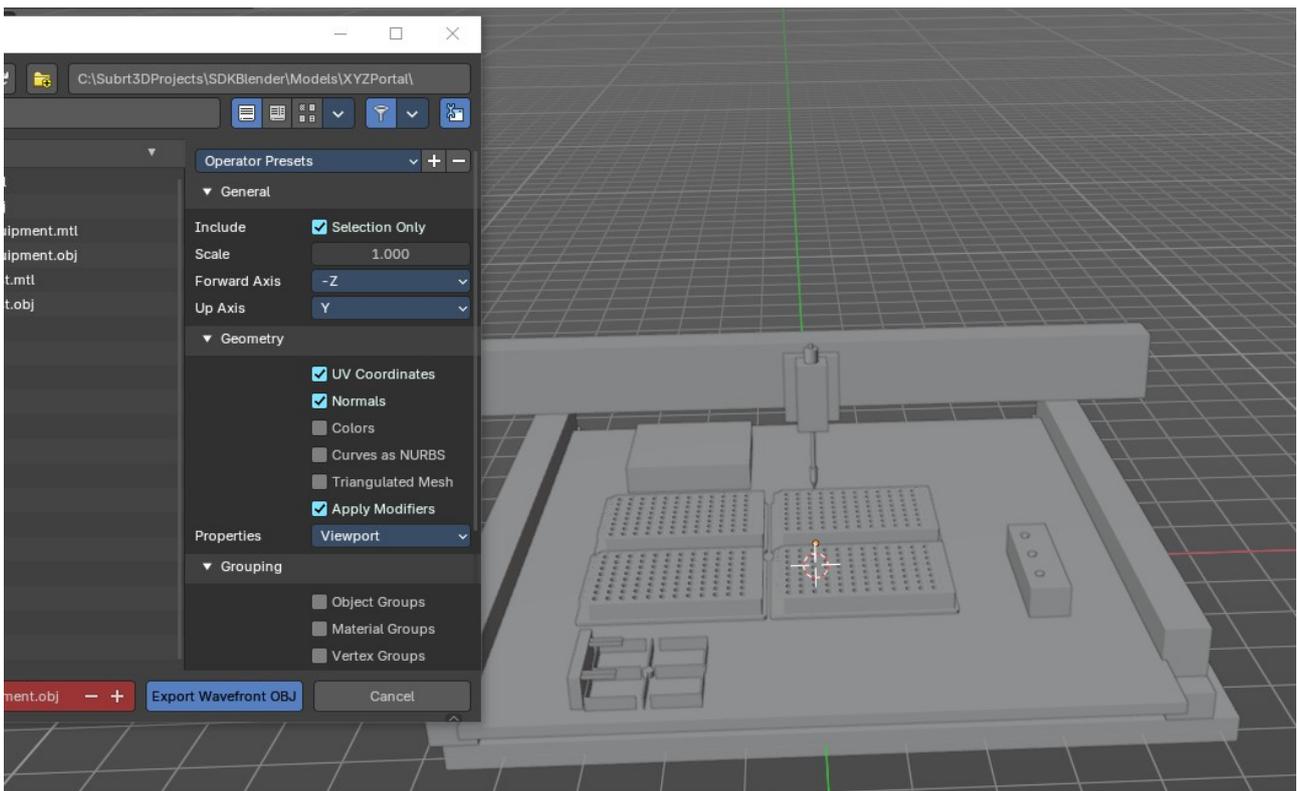
given technology.

For this reason, in this example, the model is supplemented with components used in the handling of liquids.

The model therefore includes a plates with wells, a container with pipettes, a trash bin, etc.

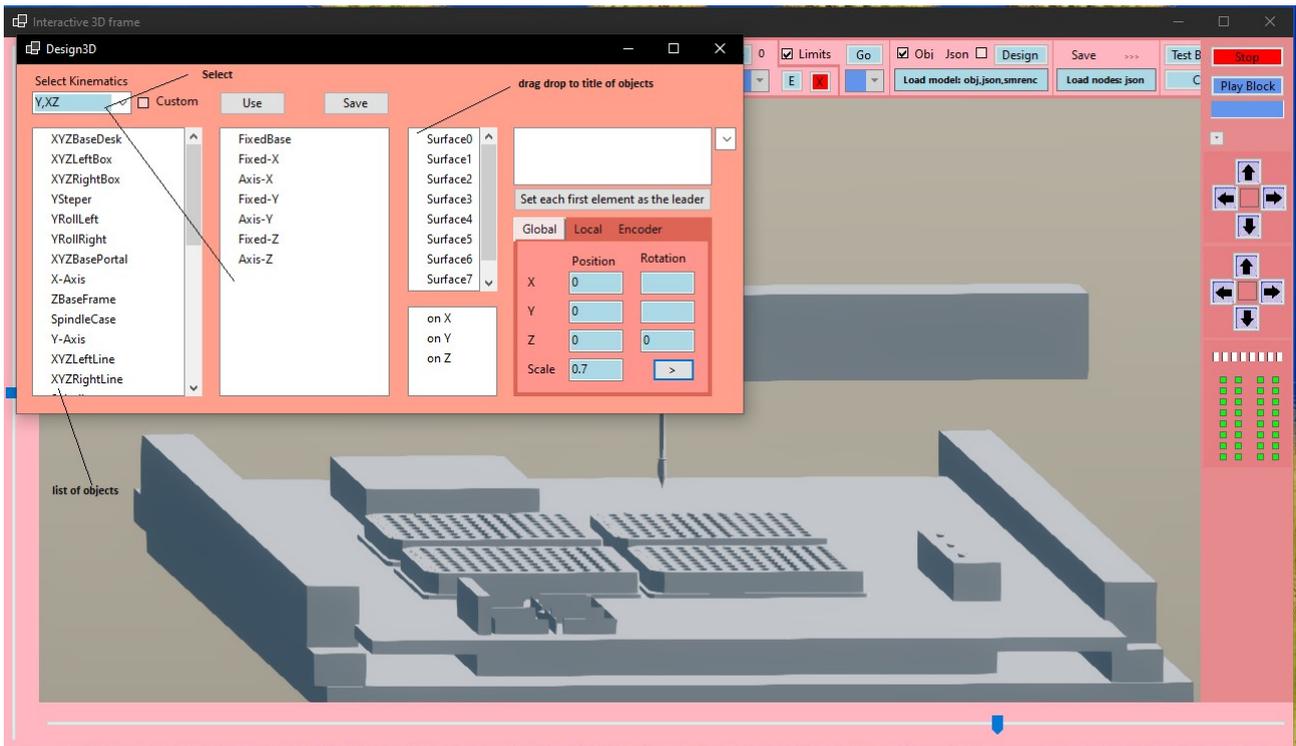
Preparing the model requires the following steps:

1. Design a model in the Blender editor and export it to OBJ format.
2. Load the OBJ file into the Smorynes simulator and open the design form. The Scale parameter optimizes the display size.

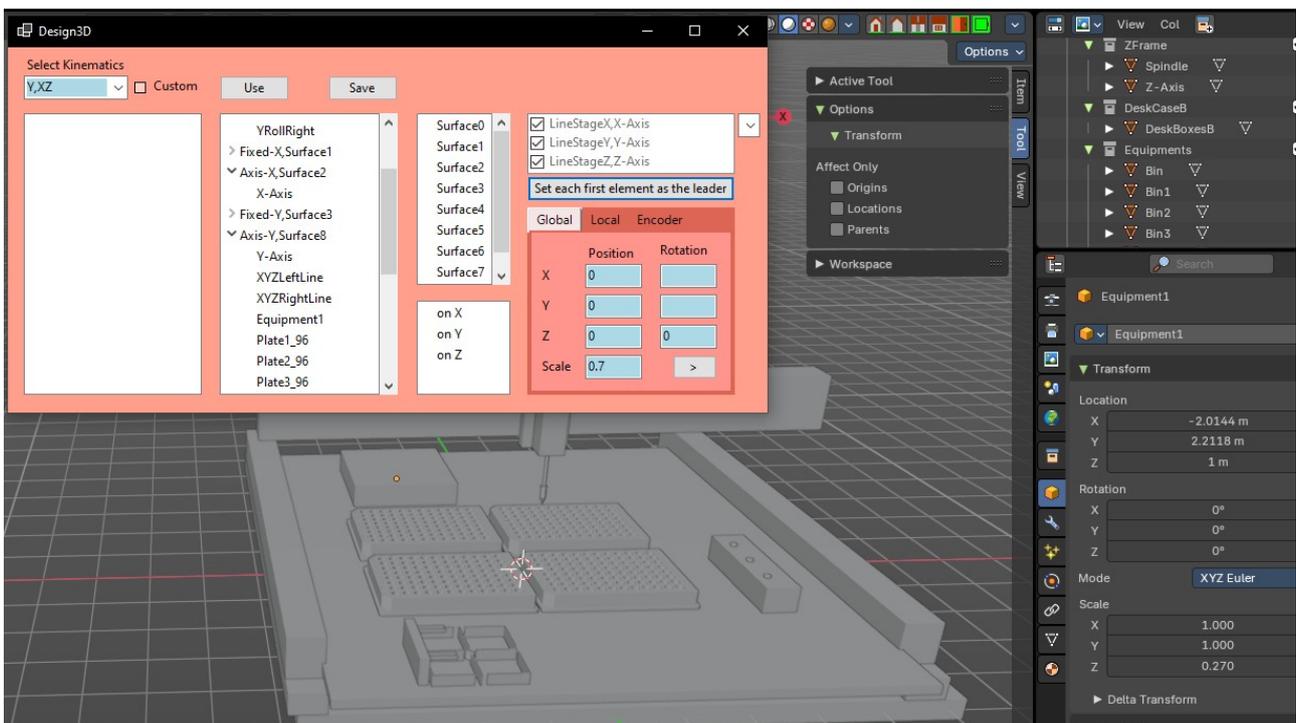


Select External request

Drag drop OBJ file



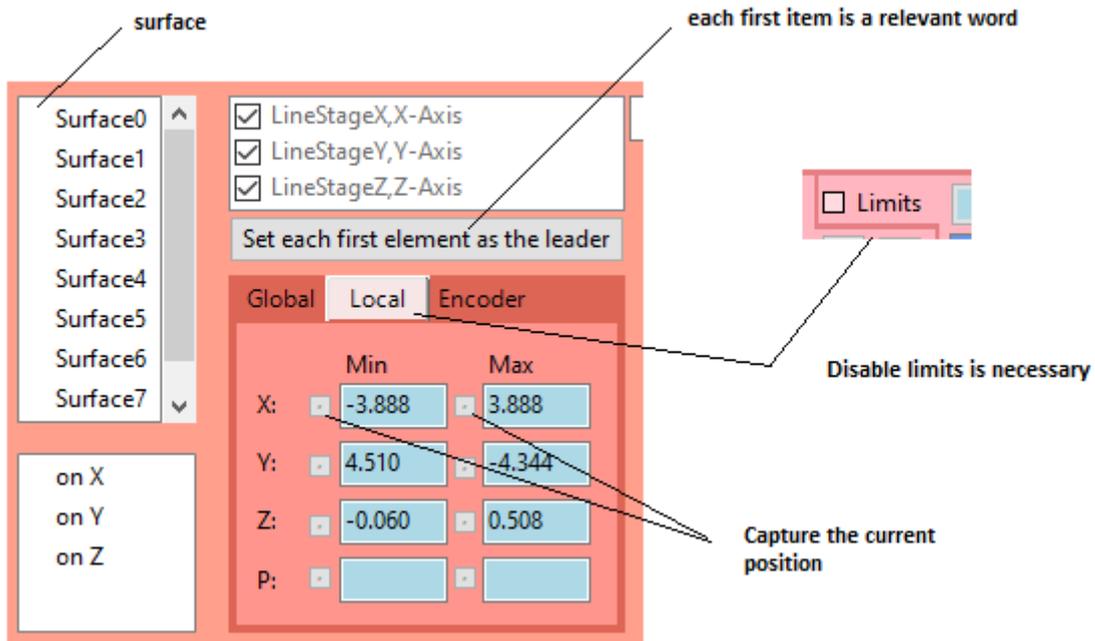
3. Select the kinematic chain and assign individual objects to fixed and moving parts.



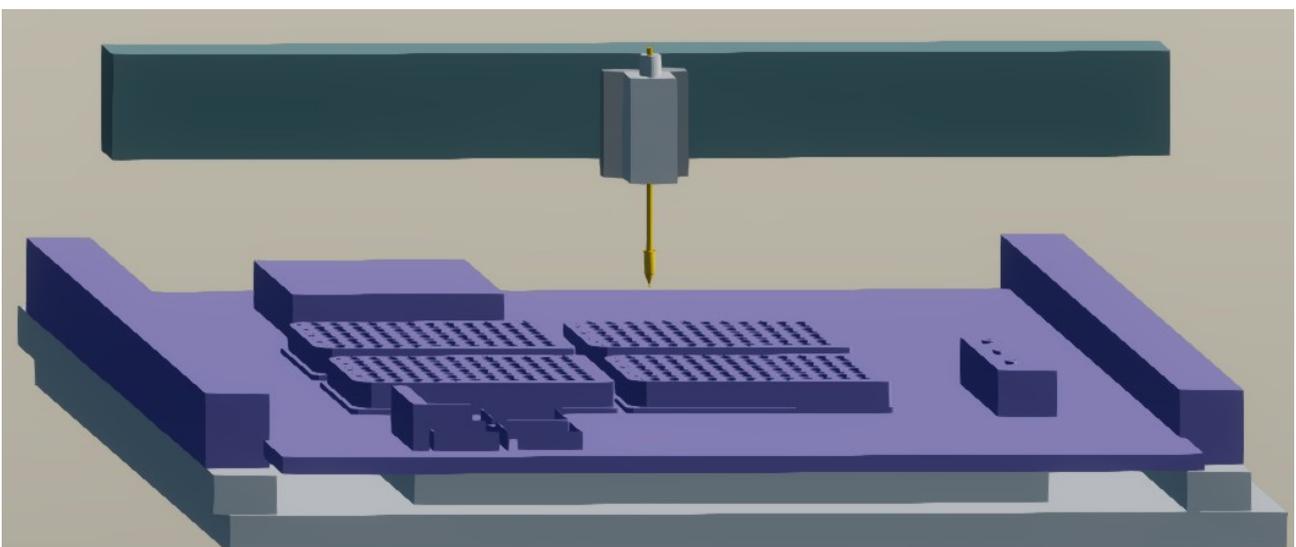
For better orientation in the object list, it is advisable to open the Blender editor at the same time.

4. Follow the rules for keywords or assign relevant words.

5. Use the Use button to generate a model update and set the size of the axis travel.



6. Assign color assignments from the Surface table.



7. Save the parameters to the <file>.JSON file.

Keywords

As described in the "Documentation" section, the user model axis in the "Unity" version must be marked with the following keywords.

"LineStageX"
"LineStageY"
"LineStageZ"
"JointPan"
"JointTilt"
"JointRoll"
"SpindleEffector"

This is very impractical because it leads to the need to rewrite an existing model in CAD, e.g., Blender. Therefore, a mechanism of alternative words has been implemented. For each keyword, an alternative word can be defined that marks the axis in the existing model.

Alternative words can be set in <file>.smrenc or written directly for the model in a .json file.

Standard pick-and-place systém

Study : <https://industry40.online/pdf/BonderXYZP.pdf>