

Manual

WARNING:

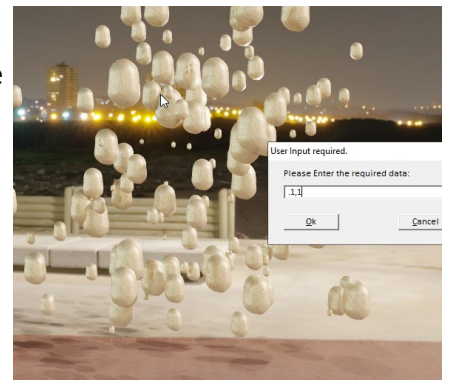
*This Software-Product **will only work with MS-Windows**. It will NOT work on LINUX, MAC etc. [Windows-Only] does NOT work with other OS!*

The External Blender Button-Bar (EBB) is a Sort of "Macro-Button Bar". The idea behind is, that you can get „anything you need with just a button-press“.



Instead of clicking through the Blender UI, or using hundreds of Keyboard-

Shortcuts. You can define these Buttons with whatever features you need for your productions.



IMPORTANT:

Please see the further Installation-Instructions on Page 3.

Blenderbar MAY NOT WORK otherwise.

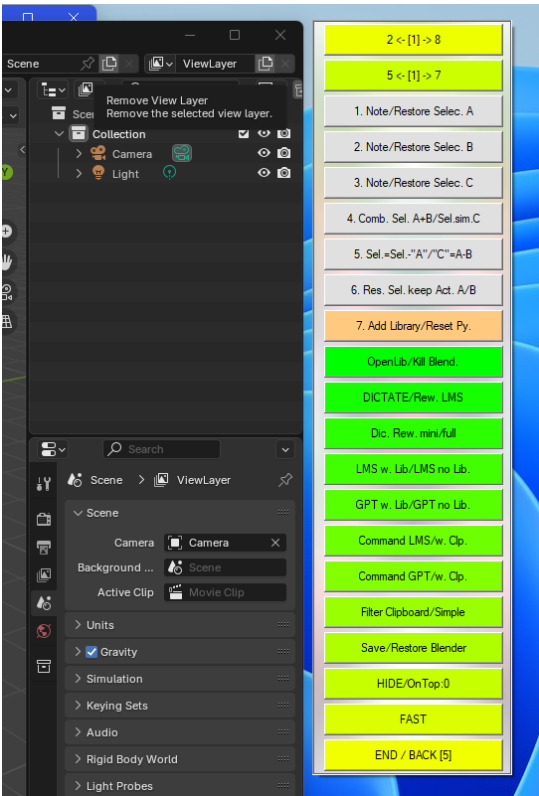


****Important note about the 2025 Go4 update of the Blender button bar****

Below you will find two screenshots.

The first screenshot shows the button bar ****without an active OpenAI API key****. As you can see, several buttons are missing. This is because they only work when a valid OpenAI API key is stored in the `library.py` file.

Therefore, **please obtain an API key from OpenAI and insert it into the `library.py`** file first. Once that’s done, you’ll also get access to the new features that the updated button bar offers. Further details can be found in the video we will upload here shortly.



How to Get an API Key from OpenAI – A Beginner’s Guide

Getting started with the OpenAI API is surprisingly simple. Below you’ll find a step-by-step walk-through, complete with direct links, so you can focus on building your app instead of getting lost in the details.

1. Create an OpenAI Account

- ****Visit****: <https://platform.openai.com/signup>
- Fill in your name, email and a password (or sign up via Google/GitHub).
- Verify your email address by clicking the link sent to your inbox.

> ****Tip**** If you’re part of an organization or want to use a business email, choose “Create account” instead of “Sign up for free.”

2 Choose a Pricing Plan

After logging in, you’ll be prompted to pick a plan:

Plan	Monthly Cost (USD)	Free Tier	Notes
-----	-----	-----	-----

| ****Free**** | \$0 | 3 \$10k tokens per month | Ideal for experimenting; includes GPT-4o mini and some older models. |
| ****Pay As You Go**** | Variable | None | Pay only for the usage you consume. |

- ****Click**** the plan that suits your needs (the free tier is enough for what we do here).
- If you choose a paid plan, you'll be asked to add a credit card or billing method.

3 Verify Your Identity (If Required)

For higher tier plans or certain regions, OpenAI may ask for:

1. ****Phone number**** – for two factor authentication.
2. ****Identity verification**** – upload a photo ID or passport.

Follow the on screen instructions; it usually takes only a few minutes.

4 Generate Your API Key

Once you're in the dashboard, follow these steps:

1. ****Go to the "API keys" section****
- <https://platform.openai.com/account/api-keys>
2. ****Click "Create new secret key."****
3. ****Copy the key immediately**** – it will look like `sk - XXXXXXXXXXXXXXXXXXXXXXXX`.
- The key is shown only once for security reasons, so store it safely (e.g., in a password manager or an environment variable).

****Security best practice:**** Do not commit your API key to public repositories. Use `.env` files or secret management tools.

5 Manage Your Keys & Usage

- ****Dashboard****: <<https://platform.openai.com/account>> – view usage stats, billing history, and project limits.
- ****Rotate or Revoke****: If you suspect a key was exposed, go back to "API keys" and either delete the old key or create a new one.

6 In case your API-Key does not work as expected, you can test it using a commandline or batchfile:

Test the Key with a Quick API Call to confirm everything works:

```
```batch
curl https://api.openai.com/v1/chat/completions \
 -H "Content-Type: application/json" \
 -H "Authorization: Bearer YOUR_API_KEY" \
 -d '{
 "model": "gpt-4o-mini",
 "messages": [{"role": "user", "content": "Hello!"}]
 }'
```

``` Replace **YOUR\_API\_KEY** with the key you just copied.

If you receive a JSON response containing a message from the model, you're set!

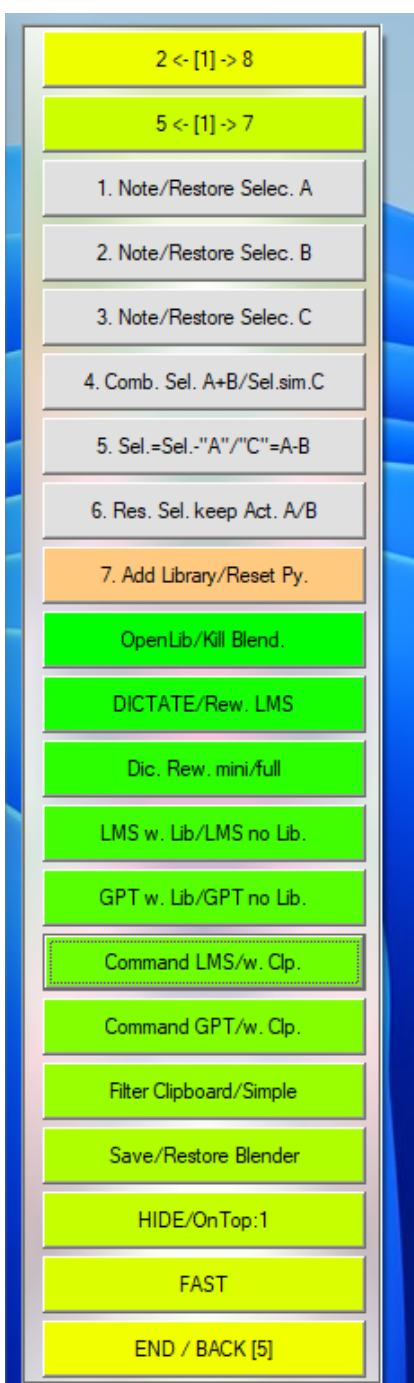
Congratulations!

You now have an API key and the knowledge to start integrating OpenAI into your projects. Happy coding!

Here you can see where you have to enter your Open-AI API-Key.

```
41 g_grok_response_log = "grok_response.log" .. # Global: Response
42 #####
43 # Insert your API-Key from OpenAI here.
44 g_Open_ai_key = "INSERT HERE YOUR OPEN AI-API-Key"
45 g_Open_ai_model = ""
46 #####
47 # Global Variables for Hunyuan API Configuration (Persistent,
48 # Enhanced: JSON-configurable load from 'hunyuan_config.json'.
49 g_hunyuan_api_key = "" .. # Global: Holds the API key (lazy-load
50 g_hunyuan_endpoint = "http://localhost:7860" .. # Wait, no-for: F
```

Now simply restart Quickbuttons and it should look like this:



To the already available features you got new KI-Buttons.

Some of these Buttons will work with LM-Studio, some will work with Open AI API.

On the next Page you will find more Informations about LM-Studio and where to download it.

Here follows some short Info about the new KI-Button:

The first new button is the **Dictate, Rework, LMS button**. You can use the left mouse button to dictate text, as I am doing right now.

If you want this text to be corrected later—errors fixed and sentences improved—you should press the right mouse button instead of the left; then you can dictate, and the text will automatically be reworked.

All output from the KI are always available in the clipboard.

This is a general rule of thumb.

The next button „**Do Rework: Mini/Full**“ will not work with LMStudio, but it is intended for users who wish to access **OpenAI's Cloud AI**. When you click the left mouse button, it will utilize GPT-4.1 Mini; clicking the right mouse button will employ the more advanced GPT-4.1 model.

Next we introduce two buttons dedicated for use with Blender.

The first Button „**LMS with Library/LMS no Library**“ will call LM-Studio to generate whatever Geometry you want.

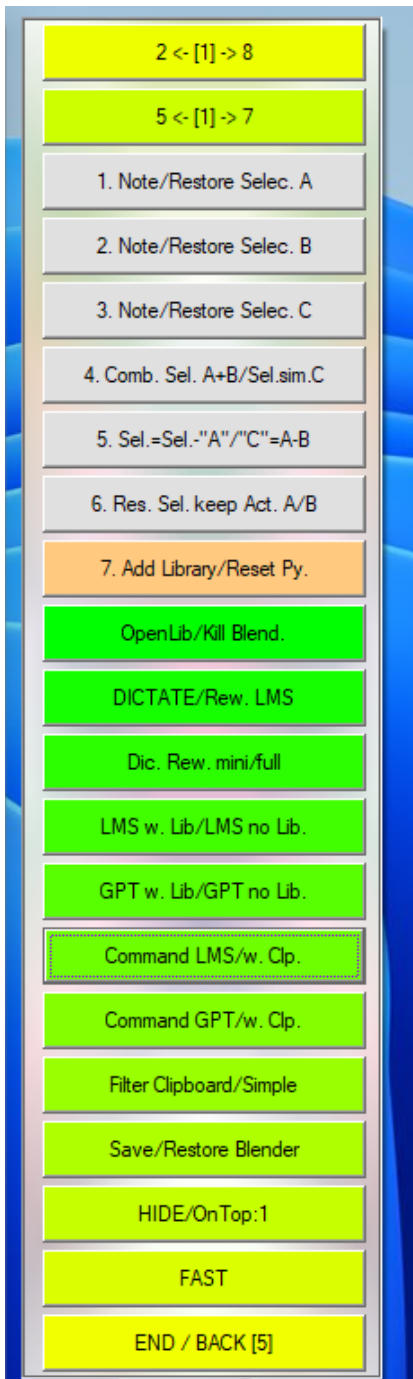
The next Button gives you the choice between GPT 4.1-Mini and GPT-4.1 to do the same.

These are likely the primary controls for Blender integration. You can activate a button using either the left or right mouse button.

After clicking, simply dictate into the microphone the model or geometry you desire.

Then press **stop**; the AI—either LM Studio or GPT-4.1—will generate the corresponding script for you. Paste this script into Blender „Text Editor Window“, run it, and obtain the desired outcome.

There are two options: “Generate script with user library” and “Generate script without library.” The first option is recommended as it provides the AI with additional context



and access to our library, which is imported directly into the script. This enables the AI to create more refined geometries and functionalities. As demonstrated in the video, you can enhance the user library with your own subprograms. In such cases, remember to update the library description file; this file is supplied to the AI as context so it can effectively utilize your customized library.

This approach greatly simplifies using AI within Blender-Bar.

The next two buttons are the command buttons. They have a general use and a very special use.

The general use is that you can simply dictate a command, press stop, and **the AI will execute your command and return the result to the clipboard** as before.

The output from this button is not filtered; it contains all commands and **whatever the AI outputs**. If you want to filter it, you have the buttons below: **"Filter Clipboard"** and **"Filter Clipboard Simple."** These are the new buttons that apply a certain degree of filtering.

The command buttons also have a special use.

Assume you tell the geometry buttons to create a script with certain features; you receive the script in the clipboard. You paste the script into Blender, but it contains errors or unnecessary commands.

It's still in the clipboard. What you do is press the command button—either LMStudio or GPT, depending on which you prefer—and instruct it to "change the script below." Use the right mouse button for this because it sends the clipboard content along with your command. The AI receives both: your instruction and the clipboard text, which appears as the prompt's "text below." This is crucial to understand.

So, if you have a script in the clipboard that contains errors, press the command button (right mouse) and say, "Please correct the script below, remove all errors, and return only the ready-to-run script for Blender." The AI will output the corrected script. You can then click "Filter Clipboard Simple" to remove any residual unnecessary text. Finally, paste it into Blender and run it.

If you're not satisfied, repeat the process: use the command button, specify what you want changed with your mouse, and the AI will modify it again, returning the result in the clipboard. You can continue this cycle until the script meets your needs. This is the purpose of the command buttons. Remember, there are two options: "Command" (left mouse) which only adds your instruction to the prompt, and "Command with Clipboard" (right mouse), which appends the current clipboard content to the prompt. These are the two modes available.

****LM Studio – A Lightweight Local LLM IDE****

****LM Studio**** is a desktop application that makes it easy to run and experiment with large language models (LLMs) directly on your own computer.

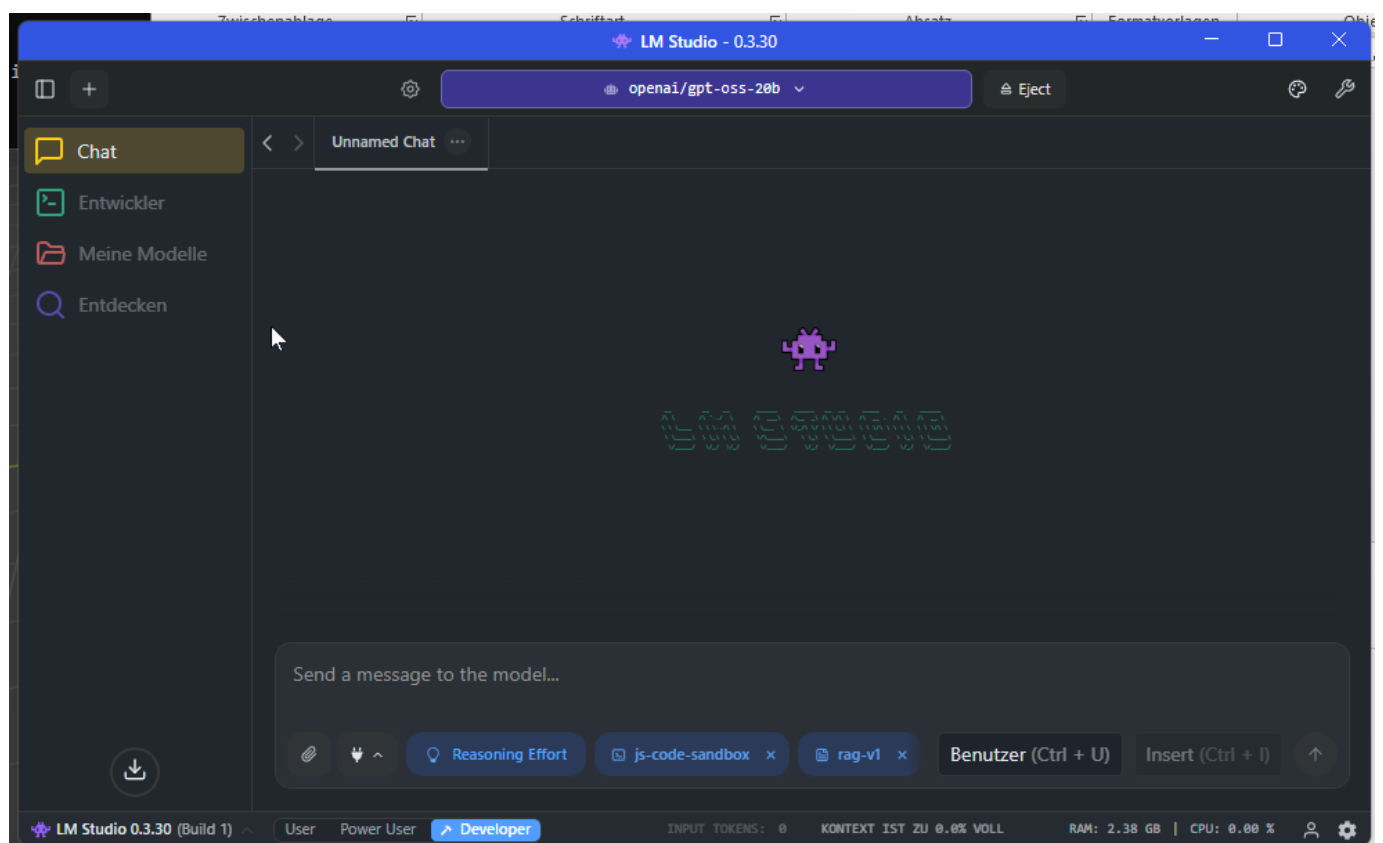
With LM Studio, you can download and interact with a variety of open-source LLMs without complex setup, whether you work on Windows, macOS, or Linux. This tool is ideal for developers, researchers, and anyone curious about LLMs who prefers local, offline access to AI capabilities—no cloud required.

Feature highlights include a user-friendly interface for prompt writing, model selection, conversation management, and more. LM Studio handles model quantization and optimization, letting you browse and download supported models from repositories like Hugging Face.

You can download LM Studio and learn more from the ****official website:****

<https://lmstudio.ai>(<https://lmstudio.ai>)

Always download from the official site to ensure authenticity and security.



Further changes:

There are many changes in the Python library „Library.txt“ due to changes in Blender, including a new user library „user_library.py“.

I cannot guarantee that all legacy buttons will continue to work as they did before.

This release primarily focuses on two aspects: **first, restoring compatibility with the latest Blender version; second, integrating an AI function that allows you to interact with Blender and automate tasks.** Additionally, we have added another layer to the existing functionality, increasing the total from seven to eight layers. The other functions have only been roughly tested, so you may need to adjust some library calls accordingly, which is feasible.



Is your Blender-Interface crowded from so many Plug-Ins and AddOns?

Get an external Button-Bar that will start all your favorite Python-Scripts for Blender. It comes with over 50 built-in Functions like "Select Similar Size".

Generally the External AddOn is independent from the Blender Version. Just the changes in the Python-Library that is provided with the Button-Bar makes it a bit Blender Version dependent.

Get up to 360 Buttons at the right Side of Blender in up to 9 "Button-Layers". You can put on each Button your own Python-Code, even separate Code for the Left and the right Mouse-Button.

Please note that the Buttonbar was improved since the Video was recorded. Many buttons have now two functions (left- and right-Click), which can not be seen in the Video.

You can easily redefine Buttons with your own Python Code, or add new Buttons. Of course you can also delete Buttons if you do not need them.

And you can switch Buttons (move them around) and group them together in the way like you want.

To make this easier, there is a File „Switchbutton.exe“ included in the Archive.

INSTALLATION:

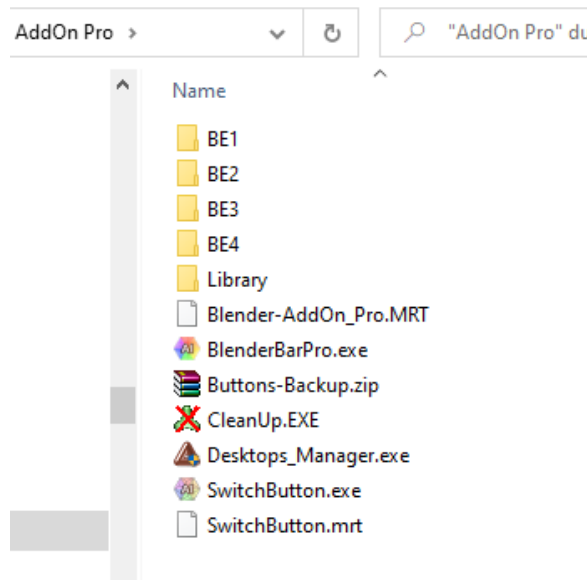
It's only 3 Steps until you can use the Blenderbar.

1. UNPACK the ZIP.File

Just unpack the ZIP-File and copy the Folder with the Files wherever you like.

From there you can start Blenderbar.

Technically behind the Buttons are Blender-Python Scripts which get executed when you press the Button.



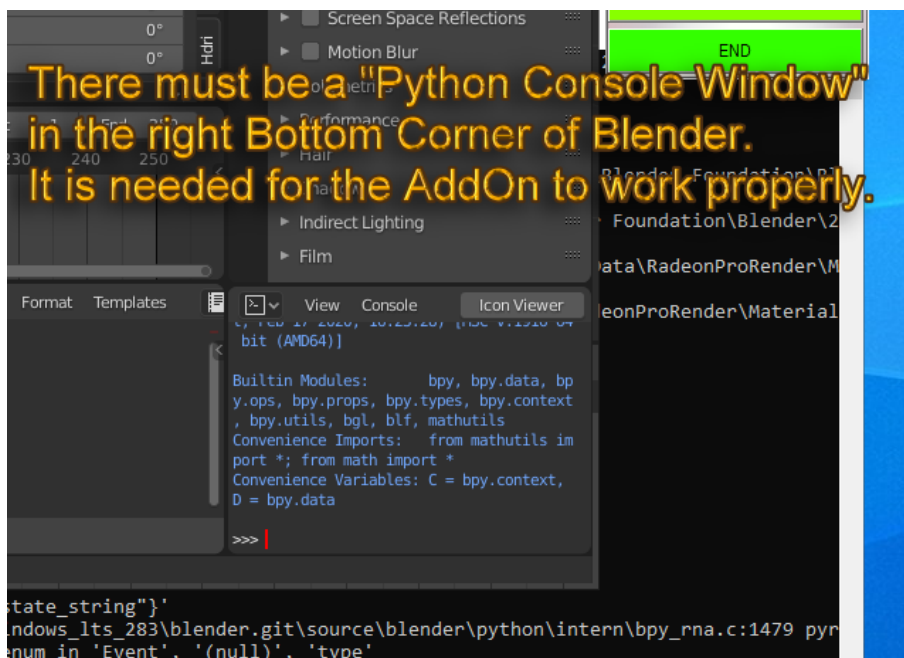
2. Make sure the PYTHON-CONSOLE is in Place!

2. Make sure that there is a Python-Console Window in the lower right corner of Blender. I recommend that you make this your „Startup-Configuration“.

A „Python-Console-Window“ **MUST BE** in the lower right corner of Blender.

3. Run Blenderbar and Press the „Add Library“-Button.

For Blenderbar to work, it must register its Python-Library with Blender.

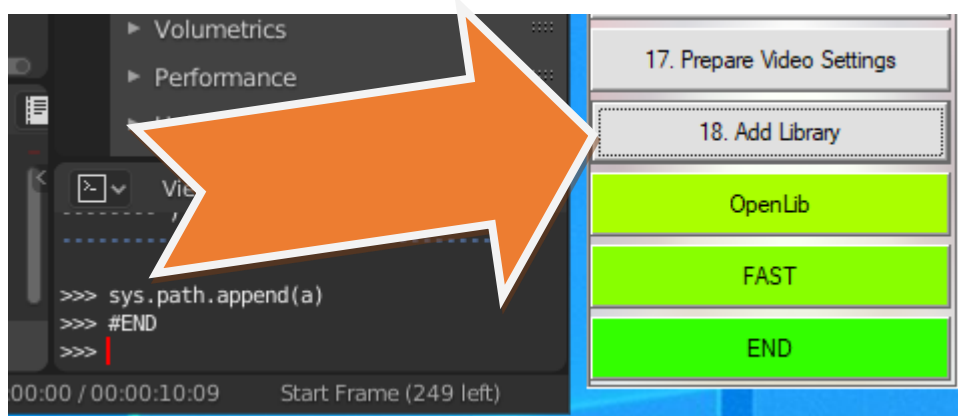


This is done with a Single Click on „Add Library“.

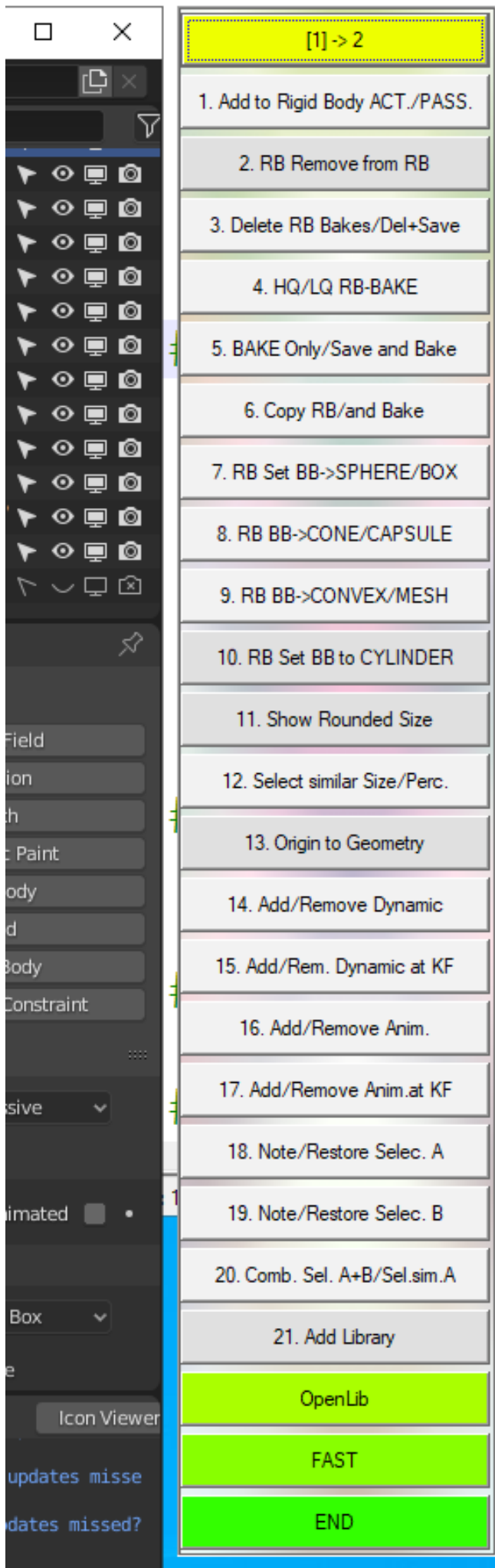
Therefore this Button is always the first Button you click.

If you forget to do this, you will get error-messages in the Python Console-Window for every Button you click.

Then just press „Add Library“ - its never too late. And if you are not sure if have already clicked it?



Don't worry. Its never too much: you can click „Add Library“ as often as you like.



Technical Details:

The The Blenderbar PRO-Version can have up to 40 Buttons times up to 9 Layers times 2 for Left and right Mouse-Button.

So you have much Space for all your Python Scripts. And then just call them with a Button-Click.

Note that the Calculation includes the reserved green System-Buttons. These green Buttons do not have a number and can therefore not be changed by the User.

The difference to most other AddOns is, that this is a separately running "Standalone" Program that will "hang on" to the Blender Window on the right side.

So it will not use any Space inside of Blender and just hang outside of it. The advantage is that most functions can just be called with a single click on a Button.

Sometimes a Popup appears that will ask you to enter a number that is then used in the Script.

I did the Blenderbar AddOn originally when i made a lot of Rigid-Body stuff. It was annoying for me, that I had always to go to the right side choose Rigid-Body "click here and there" to change something or to BAKE. ***It should had been done with just 1 Click!***

So thats the short-story behind the Blenderbar and that is why I made it primarily for myself.

You may also have some tasks that are special?

What are your task that you can automate and put on a button?

When you are inside a Blender Project, you will find them. Ist everything where you think „Oh again I have to press this and that and this ... „. These are you Candidates for your buttons.

You also have functions that you need often and would like to have them on a own Button?

The Python Code that is on the Buttons can be changed, which is as easy as editing a Text-File. All the Button-Functions are inside normal Text Files, with the Number of the Button.

The „Library“

Included is a Library Folder with a "**Library.py**" file which contains the Code behind the Buttons and a lot of Code-Samples about doing a lot of things with (selected) Objects in Blender. You can also put your own Library Routines into this file and then easily call them from the AddOn and from Blender.

The AddOn-library must be registered into Blender one time when Blender starts. Blender will remember the Library as long as its open.

This "Library-Register" is done with just a Single-Click on the "Add Library"-Button. You need to do this always as the first Click to "get Blender and the AddON a Team".

General Description of the current Button-Layers

So if you also have favorite Blender-Actions which you would like to have on a Single Button Click. And you have the Python code for these. Then you can put these tasks (as Python Scripts) on a button and call them **with just one Click**.

If these are multiple sequential tasks (for example "Save and Bake"), its even more interesting. Then you click one time, and things happen like they should.

Using the Layers of the PRO-Version, you can sort several types of tasks in several "Button Layers". Below is a **complete Functions-List** for all (currently 4) Button-Layers.

1. Currently we have **Rigid-Body** mostly on **Layer 1**,
2. **Object-Generation** mostly on **Layer 2**
3. **Move and Rotate Objects** mostly on **Layer 3**
4. **Manta-Flow**, most Keyframe stuff, Sample-Scenes and "Plank-Towers" on **Layer 4**

Below you will also find a Description of the additional Programs/Tools that are in the Package.

HINT: To switch between the (currently 4) Button-Layers, you can press the Yellow Button on Top of the Buttonbar. If you press it with the *left Mouse-Button*, it will go *one Layer forward* e.g. from Layer 1 to Layer 2. If you press the *right Mousebutton* instead, it will go *one Layer backwards*. e.g. from layer 1 to Layer 4 or from Layer 2 to Layer 1. This way you can quickly switch to the Layer where you want to go. If you make a *Double Click* onto the Button, it will quickly *switch +3 Layers*. This will be interesting if you have more then 4 Button Layers.

Clicking Left or Right?

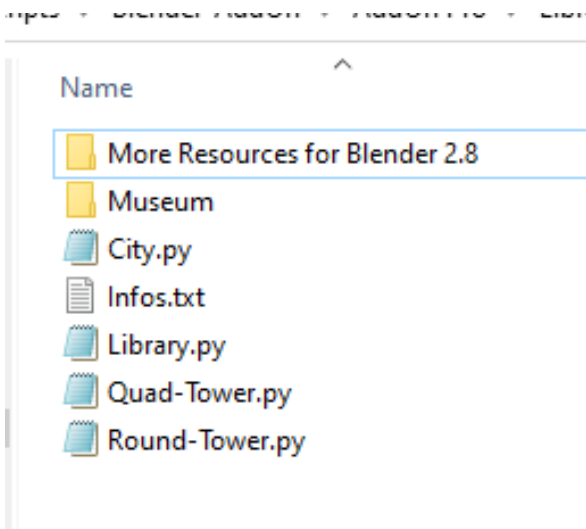
The Buttonbar has now been enhanced to also detect Right Clicks. This made it possible, to put 2 Functions on each button.

One for the Left Click, one if you Click with the Right Mouse-Button.

This is useful for Example when you rotate Objects with a Button.

So clicking Left can rotate by "90 Degree", and a "RightClick" in the opposite direction.

Buttons that have got 2 Functions and can therefore be right-clicked are painted in a bit more light color, compared to the "normal Buttons" that can only be clicked with the Left Mouse-Button.



If you open a Button-Textfile, it will look like this (Example):

As you can see, the file contains both Parts. The Part for the Left-Click as well as the Part for the Right-Click. In the Textfile they are separated with the "[OnRightMouseButton!]"-Directive.

If you do not understand what a Button is doing, you can simply open his "Textfile" and read inside what is going on.

Below is a commented Example.

This way ist easy for you to customize any Button the way you like it.

You press a Button, the EBB will write a Python Script (that you can define) into the Python Console at that place. Making Blender do whatever you have placed onto that Button.

This way you can add your own Python-Scripts on a Button for use in Blender. Without crowding Blender's AddOn Area more.

Just a Comment.

Text on the Button

Reference to the Library.py file.

Subprogram from the Library.py file.

Button10.txt - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
#Move Left/Right
from Library import *
Geo_Move_Left_by_own_SizeX()
#END
[OnRightMouseButton!]
from Library import *
Geo_Move_Right_by_own_SizeX()
#END
```

Button13.txt - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
#Move sel. L/R /2
from Library import *
Geo_Move_Left_by_half_SizeX()
#END
[OnRightMouseButton!]
from Library import *
Geo_Move_Right_by_half_SizeX()
#END
```

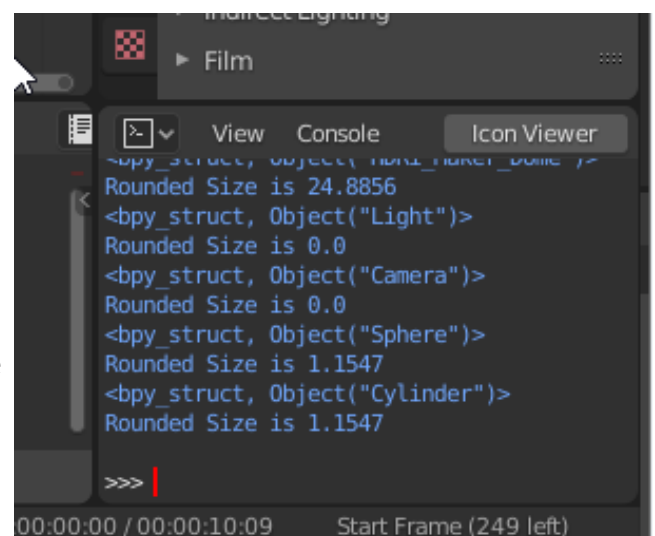
Button10.txt - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
#Move Left/Right
from Library import *
Geo_Move_Left_by_own_SizeX()
#END
[OnRightMouseButton!]
from Library import *
Geo_Move_Right_by_own_SizeX()
#END
```

Functions-List Layer 1/4:

1. **"Add to Rigid Body (ACTIVE)"** and **"Add to Rigid Body (PASSIVE)"**. This will add all selected Objects to the "Rigid-Body Simulation System". Use "ACTIVE" (left Mouse Button) for all Objects that shall move. And use "PASSIVE" (right Mouse Button) for "floor-like" Objects that should not move, but still be able to react to other (falling or moving) Objects.
2. **"Remove from Rigid-Body"**. Using this Button you can remove *all selected Objects* from RB-Simulation.
3. **„Delete RB Bake“** and **„Delete RB Bake and Save“**. These Functions will delete the last RB-Bake. The RMB will also SAVE your Scene. Ist like your press „SAVE“ in the Filemenu.
4. **"Bake-Hi Quality"** and **"Bake-Low Quality"**
These 2 Buttons are meant to speed up working with Rigid Body. No matter where in Blender you are or what is selected. Just press the Button and it will BAKE with the preferred settings. The HQ-Settings are 240,80 the LQ-Settings are 40,20.
5. **„BAKE“** (with last settings) and **„SAVE and Bake“**. These two functions do NOT change the Baking-Settings and just bake with the last used settings. Because Baking may take very long, ist useful to SAVE just before. Thats what happens if you click the right Mousebutton here (SAVE and BAKE).
6. **„Copy Rigid Body“** and **„Copy Rigid Body and BAKE“**. If you have multiple Objects selected and you change some of the RB-Properties manually, For example you change the weight or the Bouncyness. Then Blender will change that only for the actual Object. The other selected Objects will keep their values. Until you press this Button. This will make that the RB-Properties from this Object will be duplicated on all the other selected Objects.
Often after you did so, the next step is to BAKE—to see how things work out. That's what you get with the Right Mouse-Button (Copy RB and Save).
7. -10. **"Set Rigid-Body Bouncing Box"** to: SPHERE, BOX, MESH, CONVEX, CONE, CAPSULE or CYLINDER (one click for all selected Objects). With these Buttons you can set all selected Objects to another Bounding box. Doing so in Blender would require several steps (Do it with one Object, select the other Objects, go to "Copy RB Settings"), here its just one Button Click for all selected Objects.
11. **"Show Rounded Size"**. This is for you to better understand the size of Objects with Button 12. It will write the calculated size of the selected Objects into the Python-Console Window.
12. **"Select similar Size"** and **„Select similar Size with Percentage"**. this is one of my favorite functions. It will just select all Objects that have a similar size then the one that is right now selected. For doing so it will calculate the size of the Bounding Box of the Object and compare it with all the other Objects. This will select all Objects that have a similar size then the one that is right now selected. Using the RMB - PLUS/MINUS a Tolerance that you have to enter in the Input-Box in Percent. It will select all Objects with a similar size +/- the Percentage that

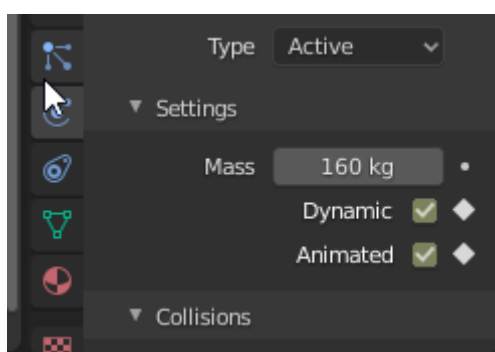


```
#Select similar Size Perc.
from Library import *
[INP]
Select_Similar_Size_Percent([INP1])
```

you entered. For doing so, Blender will calculate the size of the Bounding Box of the Object and compare it with all the other Objects. Now if the selected Object calculates to a size of "1" and you enter "10" into the Input-Box. Then all Objects with sizes between 0.9 and 1.1 will be selected. (See old Video below at: **12:25**)

13. **"Origin to Geometry"**. This Buttons is just a Shortcuts for Blender Functions that are often used while doing Rigid-Body stuff. Maybe it will be removed in future versions, because ist integrated already where ist needed..

14. **„Add & Remove Dynamic“**. for all selected Objects. Ist just that you can set this small Check-



mark in one strike for all selected Objects. RB-Objects have these two checkmarks that help you to coordinate the interaction between RB-Simulation and Keyframe-Animation. As this is an important Topic, i have put these on own buttons. This Button will just set or remove the Checkmark for all selected Objects. (No need to copy RB-Settings).

15. **„Add & Remove Dynamic KF“**. This will set or unset the same Checkmark like Button 14. And then it will record it as a Keyframe on the current Animation-Slider Position. LMB—Add

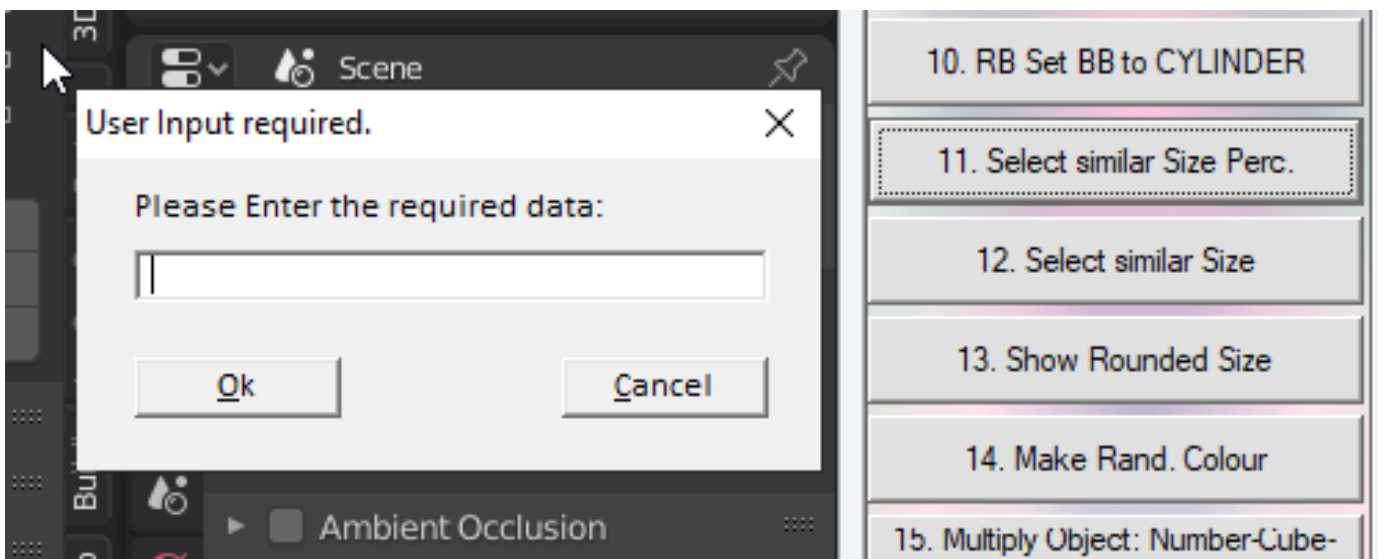
Checkmark, RMB-Uncheck Checkmark.

16. **„Add & Remove Animation Checkmark“**. This is the second important Checkmark, that you need to intermix Keyframe Animation and Rigid-Body Animation. So we have an extra But-
tons for it. This Button will just set or remove the Checkmark for all selected Objects. (No
need to copy RB-Settings).
17. **„Add & Remove Animation KF“**. for all selected Objects. This will set the Checkmark and then
add a Keyfile at the Frame/Position where the Animation-Slider right now is. This will set or
unset the same Checkmark like Button 16. And then it will record it as a Keyframe on the cur-
rent Animation-Slider Position. LMB—Add Checkmark, RMB-Uncheck Checkmark. “
18. **„Note/Restore Selection A“**. Clicking left Button here, will SAVE the current Object-
Selection. If you then Click RIGHT Mousebutton here, your Selection is restored.
19. **„Note/Restore Selection B“**. Clicking left Button here, will SAVE the current Object-
Selection. If you then Click RIGHT Mousebutton here, your Selection is restored.
20. **„Combine Selection A+B“** (Left Mouse Click) combines the stored Selection of Butons 18.
and 19. into one Selection and select all these Objects. **„Select similar to Selection A“** (Right
Mouse Click). There needs to be a stored Selection with Button 18. Now the first Object of
this selection is taken, and a „Select similar Items“ is done with this. This is especially useful, if

you want to do later a „Layer 2 / Button 11: **Make Point at**“. Then all the found Objects will rotate themselves direction of the Object that was selected and stored with Button 18 (A).

21. „**Add Library**“. This is possibly the most important Button. It will make the connection between the Blenderbar and Blender. So this Button must be pressed One-time before you can use the Blender-Bar. If you forget it, don't worry—it's never too late. You can also press it multiple times, this is completely Ok (but not needed). Once pressed Blender will keep the Library-Path until it's closed.
22. "**OpenLib**" - will open the Library-Folder in the Windows Explorer. Note that this is a green-System Button that can not be changed. It will however change its functionality in the other Layers. Other Buttons are available in Layers 2+3 instead of „**OpenLib**“. "**Save Blender-Size**", "**Restore Blender**" will save and restore the Position and size of the Blender-Main Window. These Buttons are in **Layers 2 and 3** and replace the "**OpenLib**"-Button from **Layer 1**. There may be a new function in Layer 4 in further Updates.
23. "**FAST**"/"**SLOW**"/"**END**" are three green System-Buttons that are available on all Layers and can not be changed from the User. "**FAST**" will change the writing speed of the Blenderbar from "**faster then you can see**" to "**Normal human typing**". (See Video 2:44)
25. "**END**" will end the Blenderbar AddOn. If you close Blender, the Buttonbar will close itself automatically.

You can have multiple Instances of Blender with multiple Buttonbars running at the same time. If you minimize Blender the Buttonbar will minimize itself together with Blender.



Functions-List Layer 2/4:

Layer 2 has mostly Functions to multiply Objects along the X,Y and Z-Axis. Generally these objects are "linked Duplicates". The advantage is that you can easily convert them into real Objects using Button 3. If needed. Because „Linked Duplicates“ are faster in CAI-calculation and need much less memory.

If you use Button 12 with the Array-Modifier you will get "real copied Objects".

You can use Button 2 to decide if you prefer „Linked Duplicates“ or real Objects for nearly all functions.

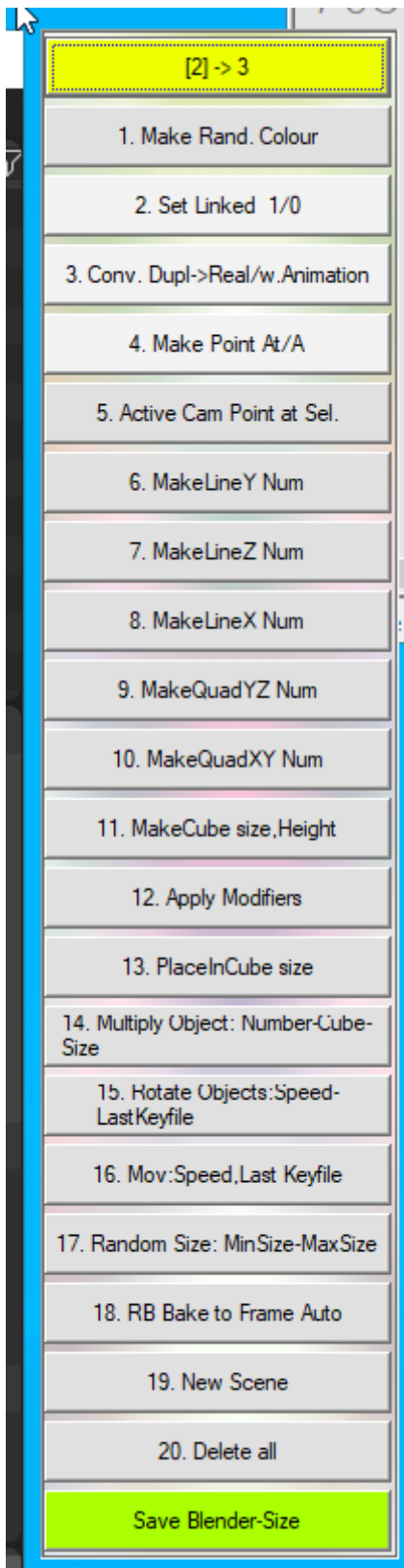
Using Button 3 you can convert „Linked Duplicates into real Objects“. There currently is no way in the other direction.

1. "Make Random Color" - this will give each selected Object another random Color. If that does not work, please check:

Is it a „Real Object Copy“ or a „Linked Duplicate“?

Only real Objects can have an own color. "Duplicate Linked" Objects share the same geometrical and texturing Data, so they will most often (unless using special location dependent shaders?) all have the same color. So you can even use this function to quickly test if Objects are „Linked Duplicates“ or not. Because i used this often, the Button is now on Top at Position 1.

However due to the sharing of the Object data, „linked Duplicates can more effectively used in „Rigid-Body“ especially Plank-Tower-Simulations. If you change the color of one linked Duplicate, ALL of them will immediately get the new color.



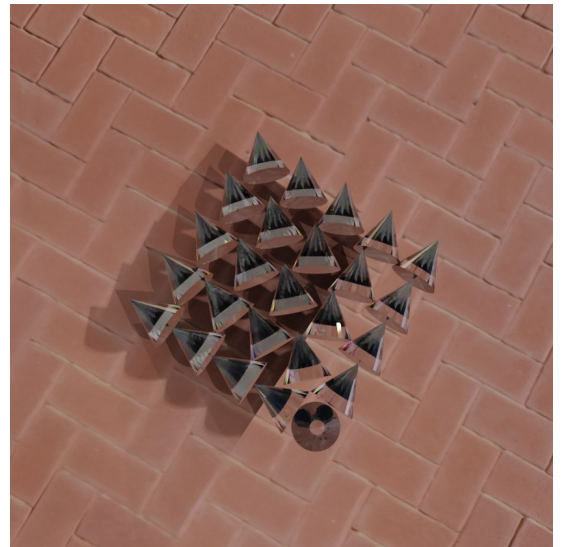
```
> View Console Icon Viewer
>>> LIF=1
>>> print("Will produce REAL Objects")
Will produce REAL Objects
>>> #END
>>>
```

```
> View Console Icon Viewer
>>> LIF=0
>>> print("Will produce linked Duplicates")
Will produce linked Duplicates
>>> #END
>>>
```

2. "Set Linked 1/0". Press the Left Mousebutton to switch to „Real Objects“. Press the right Mouse Button, to get „Linked Duplicate Objects“. See the blue Text „Will produce REAL Objects“. That appears in the Python-Console Window. See picture.



3. „**Convert linked Duplicates to Real Objects**“. This Button will convert all selected Objects that are „Linked Duplicates“, to real objects. If you use the RMB, the „Animation Data“ will also be unlinked, otehrwise not.
4. "**Make Point at**". This will make the **selected Objects** all point at the **first selected Object**. It can handle multiple Objects in one run. Now if you use the RMB, then all selected Objects will point at the Object that was stored in „Selection A“ (Layer 1, Button 18). Using



this Button can make nice looking Object patterns that look like magnetic field-lines, see Picture.

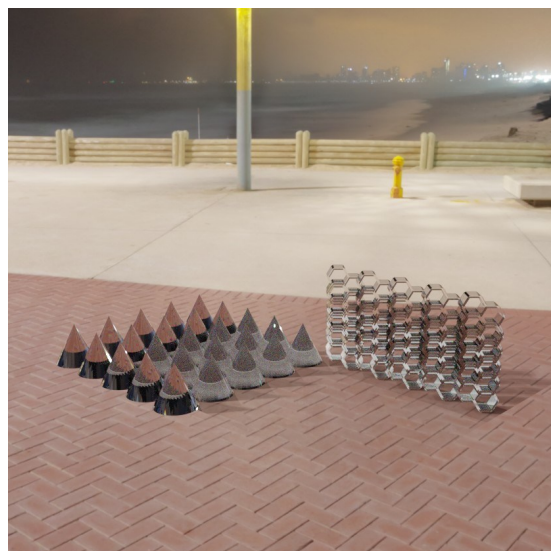
5. "**Active Camera Point at Selected Object**". Press this Button to make the active Camera look at the selected Object. This is **not** identical with the Blender Function "**Align Active Camera to selected**" (orange Arrow in the picture below) as it will not change the Zoom of the Camera and instead just turn the Camera into the right direction (blue Arrow in the picture above).



- 6.-8. "**MakeLine X/Y/Z**". This functions duplicates the selected Object(s) and line them up in the selected direction (X/Y/ or Z). Depending on the settings in Button 2, normally "Linked Duplicates" are used. This way the Objects can be faster animated using the Rigid-Body-System. It will work with 1 selected Object or with multiple selected Objects, in such case the number of generated Objects will exponentially rise!

- 9.-10. "**MakeQuad XY/YZ**". This is the Combination of two times "MakeLine" while all Objects stay selected. It will produce a field of Objects. As most functions in Layer 2 this will produce "Linked Duplicates" of the original Object. This saves Memory and speeds up the Rigid-Body Simulations.

11. "**Make Cube (Size,Height)**". This will duplicate "linked" the selected Objects in 3 Dimensions so that you get a resulting Cube





with Size*Height duplicated Objects. Using the Buttons in Layer 1 you can immediately add them to Rigid-Body (if the original Object was not), BAKE it with another Click and voila, your Animation is ready.

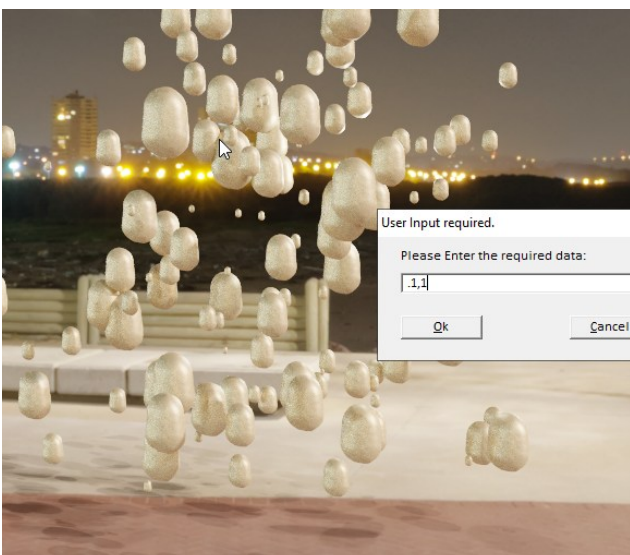
12. **"Apply Modifiers"**. This is a special **backend for the Array Modifier**. Use the Array Modifier first, then click this Button. You will get **perfectly separated Objects**, also the Centerpoint will already be **centered in the Geometry** for each new Object. Unlike the other Functions, here you do get real Objects not "Linked Duplicates".(See Video 13:56)

13. **"PlaceInCube"**. This function will distribute all the selected Objects randomly in a (assumed) Cube of a chosen size at and around Position 0,0,0. See picture left.

So you can use this function, if you have „lost“ a bunch of Objects, and you don't know where they are. Select these Objects in the Outliner, press this button. Enter the size of the imaginary Cube. Press ENTER. Thats it. This will collect your stuff together for you.

14. **„Multiply Objects, number, Cube-Size“** . You get an Input Box to enter 2 numbers. First is the number of objects (per side). The second ist the Size of the Cube in which the Objects will be distributed randomly. This will multiply the objects in a quadratic way. So if you enter 5 you may get 25 Objects in result.

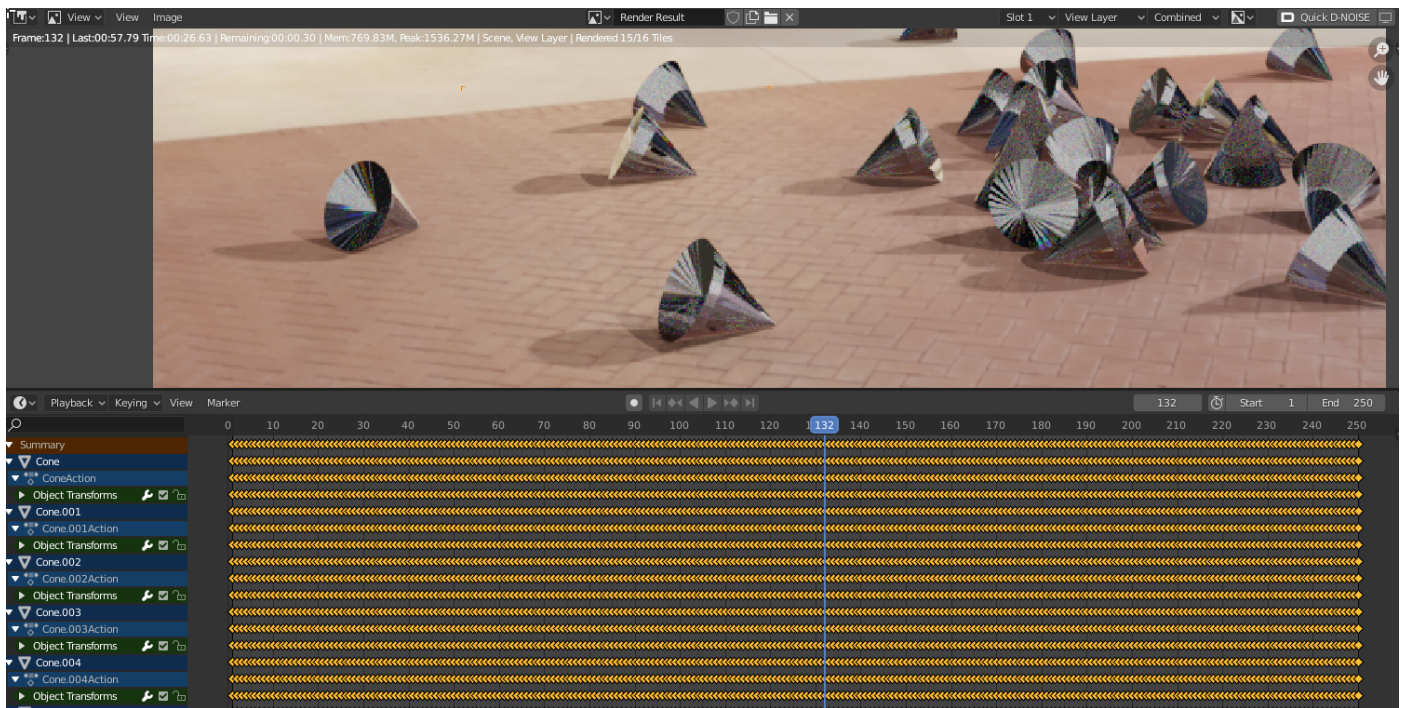
15. **"Rotate Objects (Speed, Last Keyframe)"**. This Function will create the Keyframes for an animation. It will rotate all selected Objects around themselves with a random speed (below given Speed) in a way that they return to the original position at the last Keyframe. This way you can make Looping animations with no end.



User Input required.
Please Enter the required data:

16. **"Move (Speed, last Keyframe)"**. This Function will create the Keyframes for an animation. It will move all selected Objects around on a random path, in a way that they return to the original position at the last Keyframe. This way you can make Looping animations with no end.

17. **„Random Size“**. You get an Input Box to enter 2 numbers. First is the minimum Size, second ist the maximum size. If you enter „0.5,1“ then the Objects will be scaled to a random value between these two number.



18. **"Bake to Frame Auto"**. This will Bake a Rigid-Body Simulation into Animation Frames. For all selected Objects(!). Just click the Button - you get it.
19. **"New Scene"** - A Click on this Button will delete all Items in your Scene and create a new Scene as a starting point. Of course you can modify the details.
IMPORTANT: Due to the concept of this "Just one Click"-AddOn there is NO Safety and no Undo available for this Button. Click it and whatever was there in your Scene, is gone forever.
20. **"Delete all"** Objects, leave all Collections. This will delete all Objects but leave all Collections untouched. So you can restart whichever Project you just did.
IMPORTANT: Due to the concept of this "Just one Click"-AddOn there is NO Safety and no Undo available for this Button. Click it and whatever was there in your Scene, is gone.

Datei Bearbeiten Format Ansicht Hilfe

```
#Add Plane + Ball
from Library import *
AddPlane(siz=100)
AddSphere()
```

Add Plane and Ball: This function was removed.

[3] -> 4

1. Rotate X+/-45

2. Rotate Y+/-45

3. Rotate Z+/-45

4. Rotate X+/-90

5. Rotate Y+/-90

6. Rotate Z+/-90

7. Random Rotater Sel.

8. Move Object to 0,0,0/5,5,0

9. Set Rotation to 0,0,0

10. Move Left/Right

11. Move Up/Dn

12. Move Near/Far

13. Move sel. L/R /2

14. Move sel. U/D /2

15. Move sel. N/F /2

16. Move sel. by X

17. Move sel. by Y

18. Move sel. by Z

19. EDIT:Align act. Faces

20. RB: Bullet Tar/Src

Restore Blender-Size

Functions-List Layer 3/4:

Buttons in Layer 3 are mostly around moving and Rotating Objects quickly. „Button 19“ and „Button 20“ are special Buttons, therefore please read the description for Button 19 and 20 carefully.

1.-6. **“Rotate X+/-“**. These Buttons rotate the selected Objects round their local Axis by either 45 or 90 Degree.

The direction depends on whether you use the left or the right Mouseclick.

So these Buttons can be clicked with left or right Mousebutton.

7. **„Random Rotate Selected“**. Randomly rotate all selected Items to a angle between 0 and 360 degree.

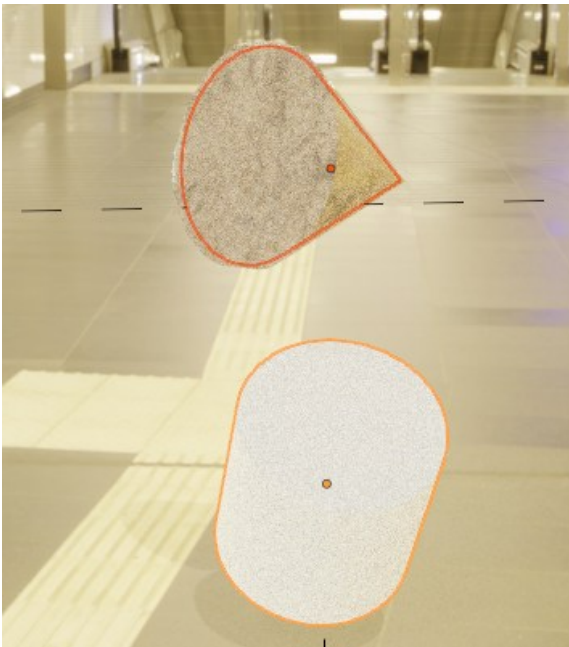
8. **“Move Objects to 0,0,0“**. Move all selected Objects to location 0,0,0.

9. **„Set Rotation to 0,0,0“**. Clicking this Button will reset the Rotation of all selected Objects.

10. – 15. **“Move Objects“** These Buttons will Move all selected Objects whether by its own full size (Button 10-12) or by its half size (Button 13.-15.). In any direction. Generally you can click with either the left or the right Mousebutton to get this or the opposite direction.

16.-18. **“Move by X“**: These Buttons will move the Object on their Axis for the amount that you enter in the Inputbox.





19. „**EDIT: Align Faces**“. This is a special Function, as it works (unlike other Buttons) not in Object-Mode, but only in EDIT MODE. So here is what you do. Step by Step.

Step 1.

Here are two Objects that you want to Align to each other. Make sure BOTH Objects are selected.

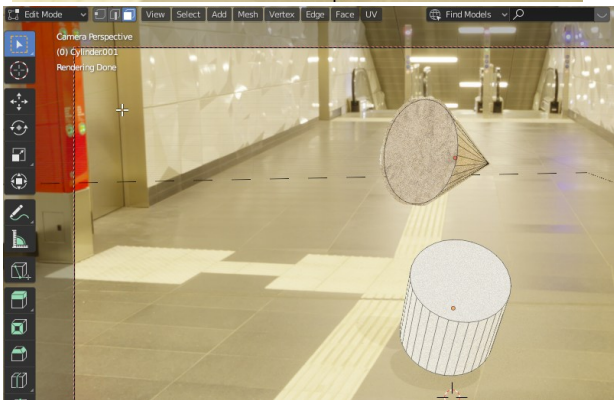
Step 2.

Get into
EDIT MODE.



Step 3.

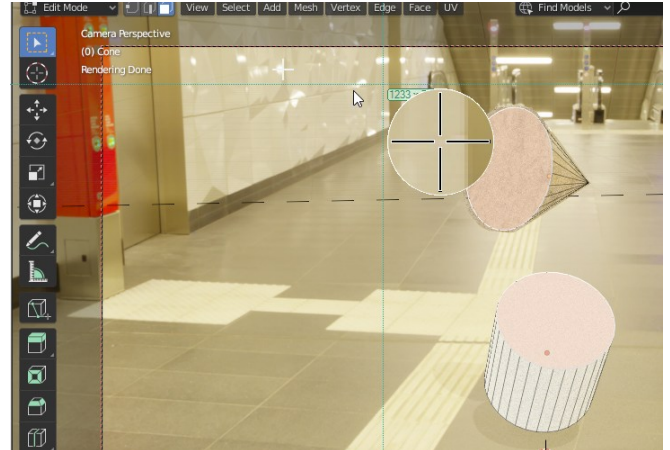
Make sure
that nothing is selected.



Step 4.

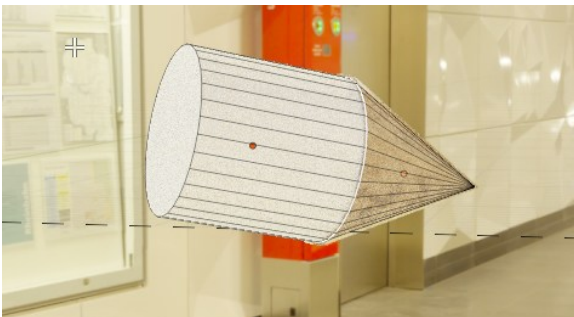
You must be in
EDIT-MODE in

FACE-SELECT MODE.



Step 5.

Now select the two faces that shall be aligned to each other. Make sure ONLY these two faces are selected.



Step 6. Press Button 19.

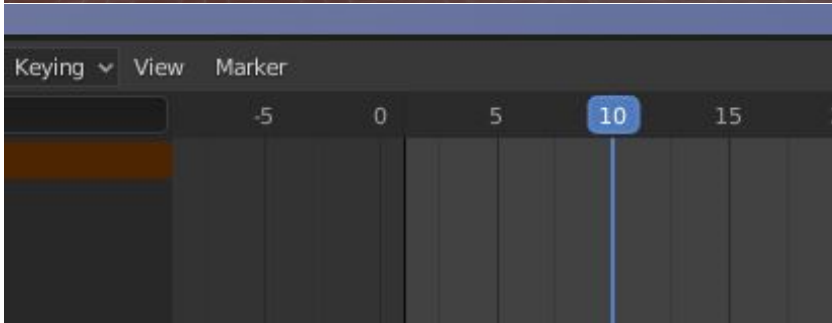


Step 7. Switch back to OBJECT MODE. This is teh Result.

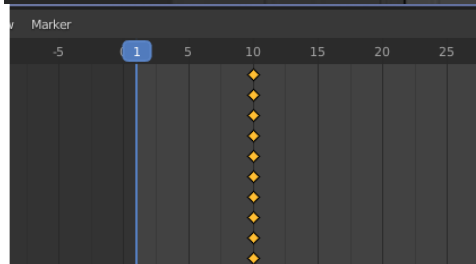
20. „**Bullet Tar**“ and „**Bullet Src**“. The Bullet-Function is a special Function that will help you to make a „Bullet-Type“ Animation in 2 somehow easy steps! Just Follow exactly the steps below.



1. Select the Stones, or Bullets or whatever you want to shoot.
Move them to the Target-Position.
Move the animation-Slider also to the target Position (maybe Frame 10).
2. Press „Bullet Tar“ (LMB).
3. Now move the Animation-Slider to the start-Position (maybe Frame 1).
4. Move the „Bullets“ to the starting Position.
5. Press „Bullet Src.“ (RMB)
6. Switch to Layer 1 and press the „BAKE“-Button.



Generally thats all. The Bullet-



Animation is READY.

Now to make it work, you will have to do some additional checks:

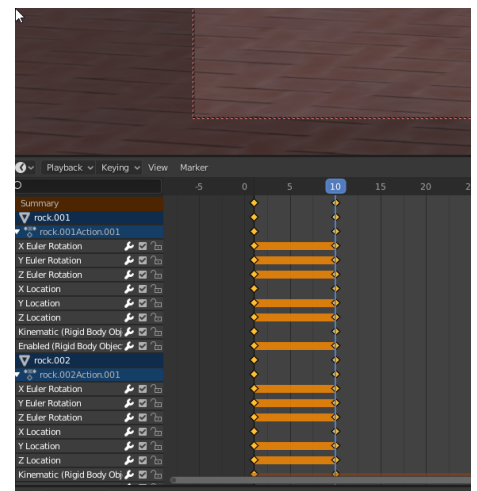
- Check that your Target is in the RB-System.
- Verify that the Weight of the „Bullets“ is much higher then the weight of the target.
- Then finally you get something like this:

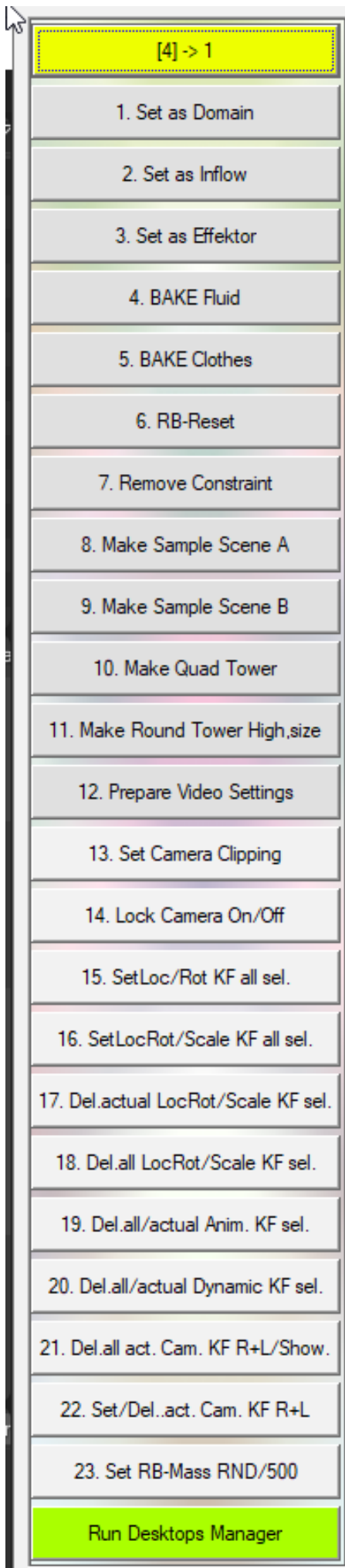
Make sure that you always press

„Bullet Tar“ first and then later „Bullet Src“.

Otherwise you will have to do some Cleanup (Delete all KF, Reset RB-World).

And do it again.





Functions-List Layer 4/4:

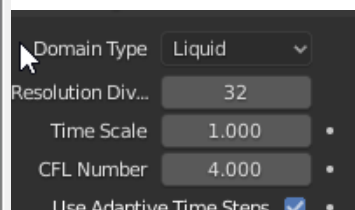
Buttons 1. to 4. are for Mantaflow-Simulations. There are some details you need to know. First, to BAKE with Button 4, the Domain Object should be selected.

Second, the BAKING will do very silent, there is no movement and no text-Output. Only the Mousepointer will rotate and show „Busy“. And third, Mantaflow-Baking can produce Gigabytes of temporary data. So i recommend taht you care to delete these „Caches“ after the Simulation is done.

Generally you can do this very easy in few steps:

->. Make a CUBE and press Button 1.

1. „**Set as Domain**“. This will declare the selected Object as Domain. The Domain is the surrounding spave in which the simulation takes place. To my current knowledge there can be only 1 Domain at a time.



2. „**Set as Inflow**“. This will declare a small Object—for example a CUBE or CYLINDER as the „Water“ or „Smake“ INFLOW. Note that the size must be larger then the „Domain-Size divided by the resulation“. If no water comes,

wether increase the INFLOW, or increase the resolution from 32 to 96 or more. The higher this number the more detailed is the simulation. But will also produce much more data and take much longer.

3. „**Set as Effector**“. An Effector is an Obstacle that will lead the water to flow somewhere. After you have defined these 3 Objects, with three Buttonclicks, you can just set the Domain-Material to „Water“ (or whatever you like to flow) and then press:

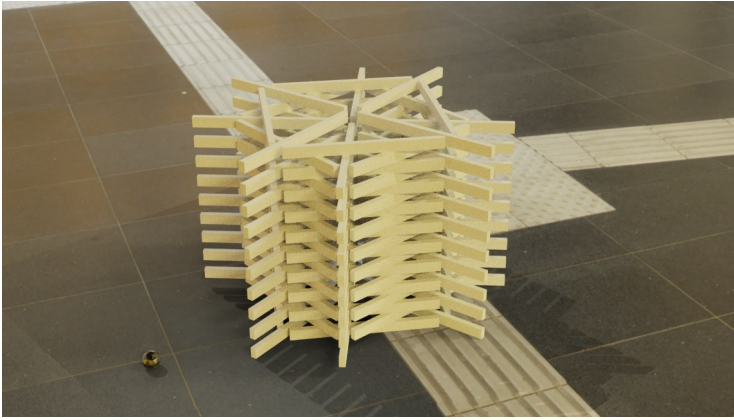
4. „**BAKE Fluid**“. Then this will bake your simulation. After that you can play it.

5. „**BAKE Clothes**“. I did not test this much, it should bake Cloth-Simulations.

6. „**Reset Rigid-Body World**“. This will destroy the current RB-World, and delete all Bakings from RB. It will then create a new, empty RB-World.

7. „**Remove Constraints**“. Untested Function.

8. „**Make Sample Scene A**“. This will generate a complete Rigid-Body Scenen, with a Chain and some Cubes. Ist just a starting Point for you. And it will take time to build!



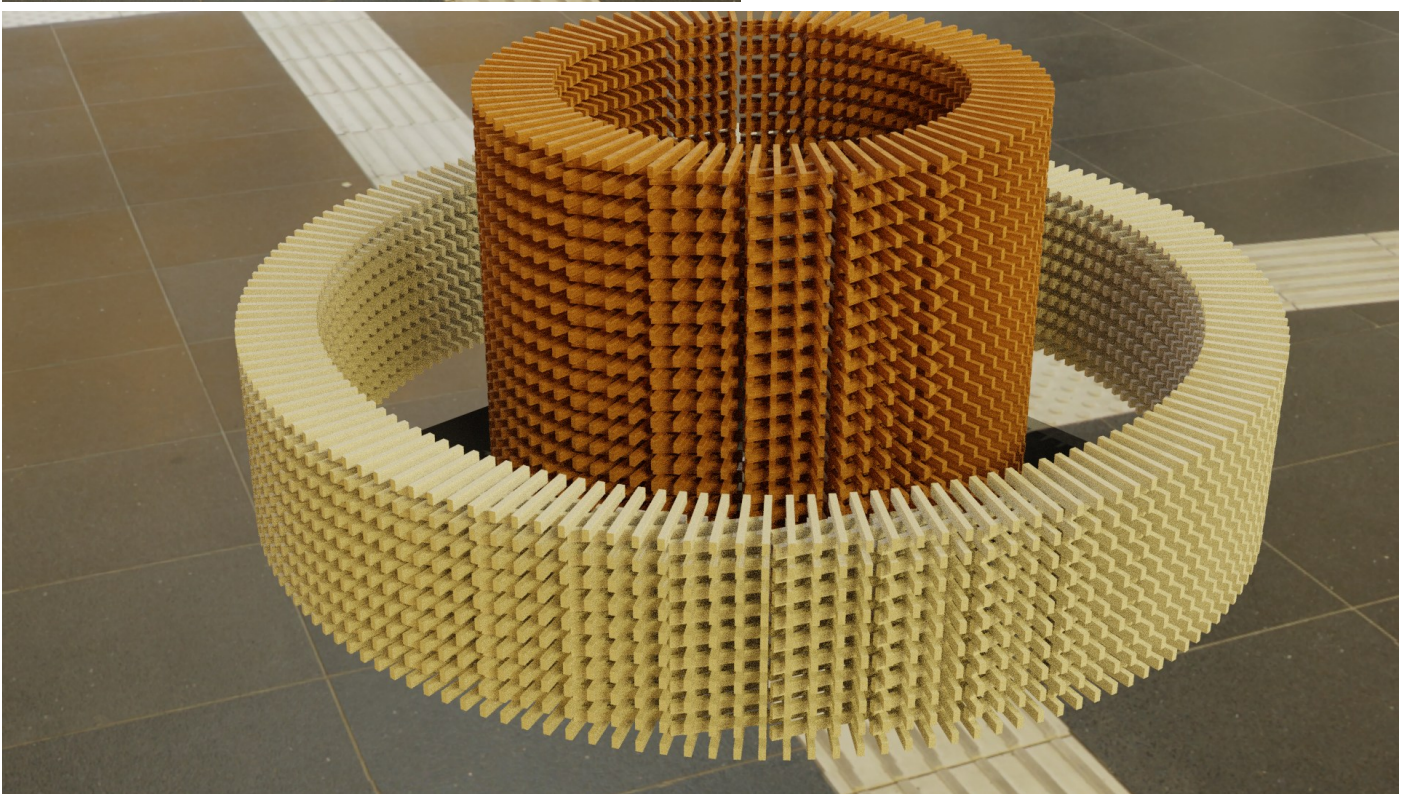
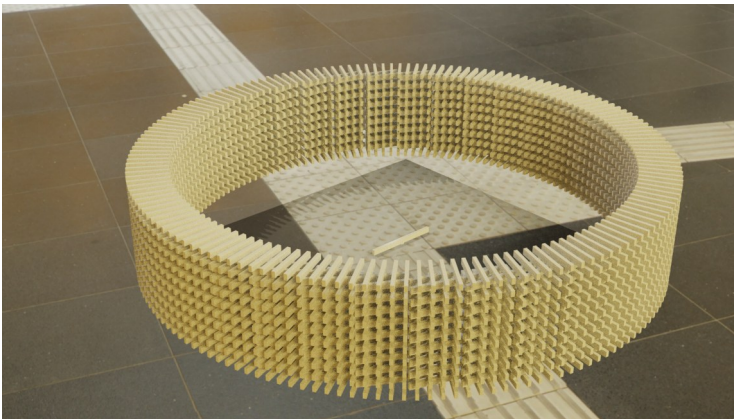
9. „**Make Sample Scene B**“. This will generate a complete Rigid-Body Scenen, with a lot of Plank Tower. **IMPORTANT:** If you press this Button, this may **RUN OVER NIGHT** forseveral Hours. Its just a starting Point for you.

10. „**Make Quad Tower**“. This will ask you for one number that is the „Hight“ of the Tower. This Tower on the Picture has a height of 20.

11. „**Make Round Tower.**“. This one takes 2 Parameters: Size and Height. Try 40x20 Tower for example.

This is a „20,40“ Round-Tower.

Below are 2 Tower, one 40,20 (in the Middle), the other is the „20,40“ Tower.



12. "**Prepare Video settings**". If you choose to produce a video instead of a Single Picture, you will need to set multiple Parameters each time. Mostly its the same. Also if you need to render a Preview picture Blender will often reset the values to "PNG". So instead of doing this stuff over and over again, here is one Button that does it for you. Of course you can modify these Parameters to fit your needs in the Textfile "BE1\Button17.txt".

13. „**Set Camera Clipping Off/On**“ to 1000.000 or to 100. I found that i need every time increase the Camera Clipping. In the Viewport as well as with the actual Camera. Now a simple Buttonclick will do that.

14. „**Lock Camera to Viewport**“. This will set or clear the checkmark for you.

Keyframe-Buttons

Having some stones flying on a Castle, i realized that Blender will record Keyframes only for the active Object. That is understandable, because in the UI you can also see only the data from one Object. However using this Button, it does not matter if there is one or 100 Objects selected: They will all be Keyframed at the current Animation-Position. Handling Keyframes in large number can be difficult and time-consuming as there is not even a vertical Slider in the Timeline. So this was a good reason to pack it on some buttons.

These Buttons have all the Left Mousebutton (LMB) and the right Mousebutton (RMB) used. Generally there are several Groups:

- A) Keyframes from all selected Objects that are at the „current Animation Position“ (Slider)
- B) All Keyframes from the selected Objects
- C) Keyframes from the actual Camera that are at the „current Animation Position“ (Slider)
- D) All Keyframes from the actual Camera
- E) We handle currently „Rotation“, „Location“, „Scale“ and the special „Rigid-Body.Animation“, „Rigid-Body.Dynamic“-Checkmarks

Now all these Groups come together in hopefully useful combinations of „Left- and RightClick“ on the following Buttons.

Please give Blender a little bit time to work after pressing any of these Keyframe-Button!

Do NOT move the mouse after you have pressed the Blenderbar-Button, give it some seconds to handle the requests first.

15.-16. „**Set Location (LMB) / Rotation (RMB) Keyframe**“ at actual Position“. The Button shows, wether ist only a „Location“, a „Rotation“ or both Keyframe. And you can also Keyframe the Scale.

Button 15 gives you the choce wether to set „Location“ (LMB) or „Rotation“ (RMB).

Button 16 will set both (LMB) and a Scale-Keyframe with the RMB

17. „**Delete actual Keyframes from all selected Objects with Location+Rotation / Scale**“

This Button will delete these Keyframes at the actual animation-Position, with one Click. Left-Button -> remove Location and Rotation, right Button: Remove Scaling.

18. „**Delete all Keyframes from all selected Objects with Location+Rotation / Scale**“

This Button will delete all such Keyframes with one Click. Left-Button -> remove Location and Rotation, right Button: Remove Scaling.

19. LMB: Delete RB-Animation-Checkmark Keyframes at actual Position from all selected Objects.
















RMB: Delete RB-Kinematic-Checkmark Keyframes at actual Position from all selected Objects.

20. Like Button 19, but will delete all such Keyframes from selected Objects

21. LMB: Delete all Location and Rotation KF from actual Camera. RMB: Show (Print) all Keyframes in Console Window

22. „**Set/Delete actual Camera Loc. And Rot.-KF at current Animation Pos.**“

23. „**Set Rigid-Body Mass to Random**“ Values between 100 and 1000, (LMB). For all selected Objects. And the RMB is: „**Set RB-Mass to 500**“.

| Name | Änderungsdatum | Typ |
|---|------------------|----------------|
|  BE1 | 16.08.2020 15:06 | Dateiordner |
|  BE2 | 14.08.2020 13:05 | Dateiordner |
|  BE3 | 16.08.2020 16:33 | Dateiordner |
|  BE4 | 16.08.2020 14:25 | Dateiordner |
|  Library | 16.08.2020 23:20 | Dateiordner |
|  Blender-AddOn_Pro.MRT | 17.08.2020 08:24 | MrtFile |
|  Blenderbar-Manual(en).pdf | 16.08.2020 22:39 | PDF Document |
|  BlenderBarPro.exe | 16.08.2020 09:16 | Anwendung |
|  Blendersize.txt | 17.08.2020 07:11 | Textdokument |
|  Buttonbackup.zip | 17.08.2020 09:22 | WinRAR-ZIP-Arc |
|  CleanUp.EXE | 30.06.2020 10:54 | Anwendung |
|  Desktops_Manager.exe | 30.06.2020 19:33 | Anwendung |
|  Download Update.exe | 17.08.2020 08:02 | Anwendung |
|  SwitchButton.exe | 17.08.2020 08:28 | Anwendung |
|  SwitchButton.mrt | 17.08.2020 08:27 | MrtFile |

Download Update.exe



This programm will Download the current Version in a zip-file. It will NOT install anything and NOT change your Buttons.

After running the „Download Update.exe“, nothing will happen for about a Minute.
Don't worry, the APP just does not talk a lot.

Ist downloading a 20 MB File and then doing some checking and Decryption.

If all works fine, you will find the newest version: „BBarAddon_Pro.zip“ aside of the executable.

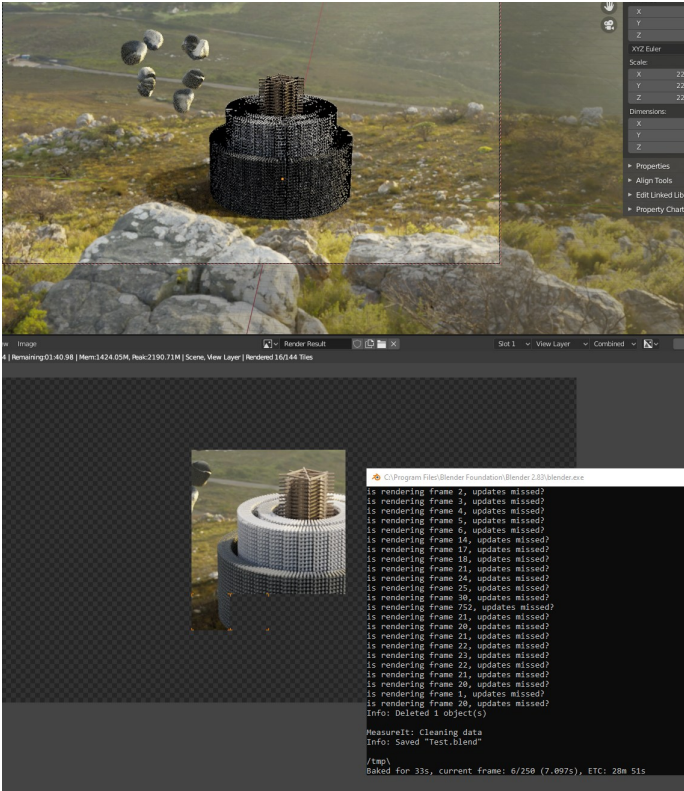
Copy it to another place and unpack it there, ist completely up to you, to use the newest version, or only some features etc.

| | | | |
|---|------------------|-------------------|-----------|
|  BBarAddon_Pro.zip | 17.08.2020 09:22 | WinRAR-ZIP-Archiv | 27.003 KB |
|  Download Update.exe | 17.08.2020 08:02 | Anwendung | 1.372 KB |

HINT: Using the Blenderbar you can start the BAKING while Cycles is still rendering.

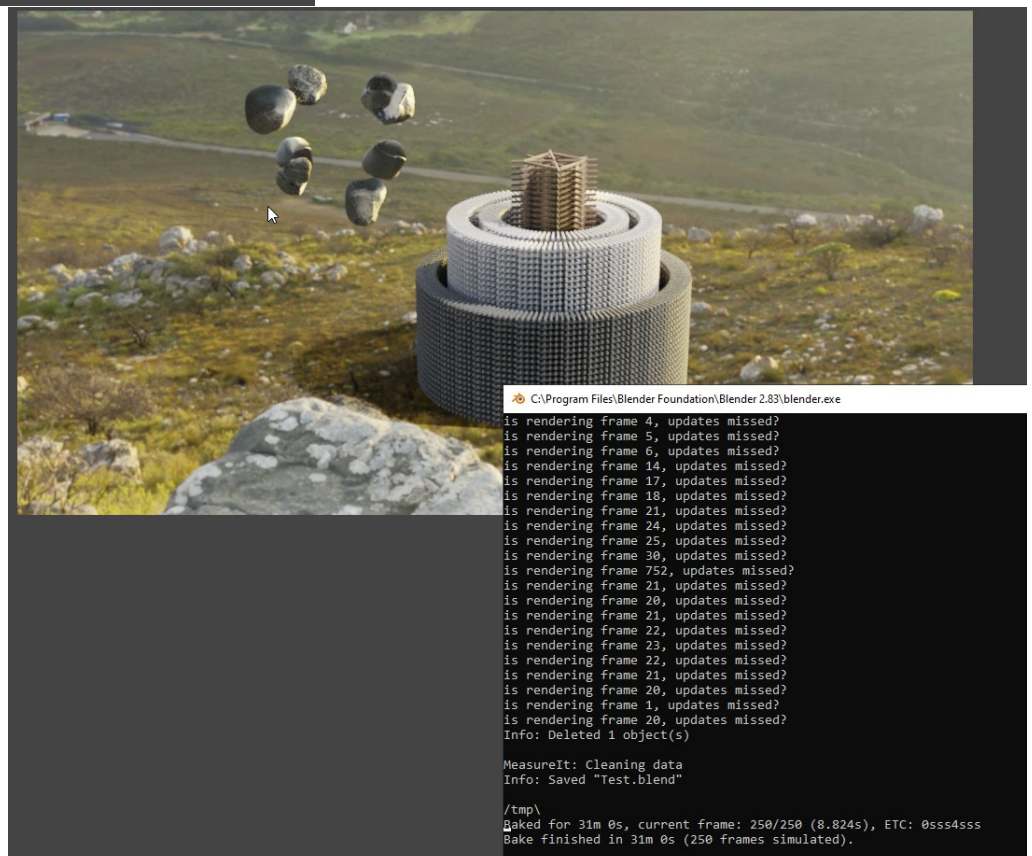
For this let Cycles start Render in the Blender-Main Window. Now click on BAKE in the Blender-bar. Cycles will continue to render, even though the Window is NOT been updated. Once the Baking is ready you will see wether the Render is also finished or not yet.

You have to start Cycles Rendering first then Baking via teh Blender-Bar. Once the Baking is running, Blender will not be responsive until ist done.



Picture 1: Cycles is rendering, but you can start „BAKING“ anyway using the Blenderbar. Cycles will NOT continue to update the Screen but will continue to render!

Picture 2: With some Luck, immediately after the BAKING, your Rendering will also be ready.



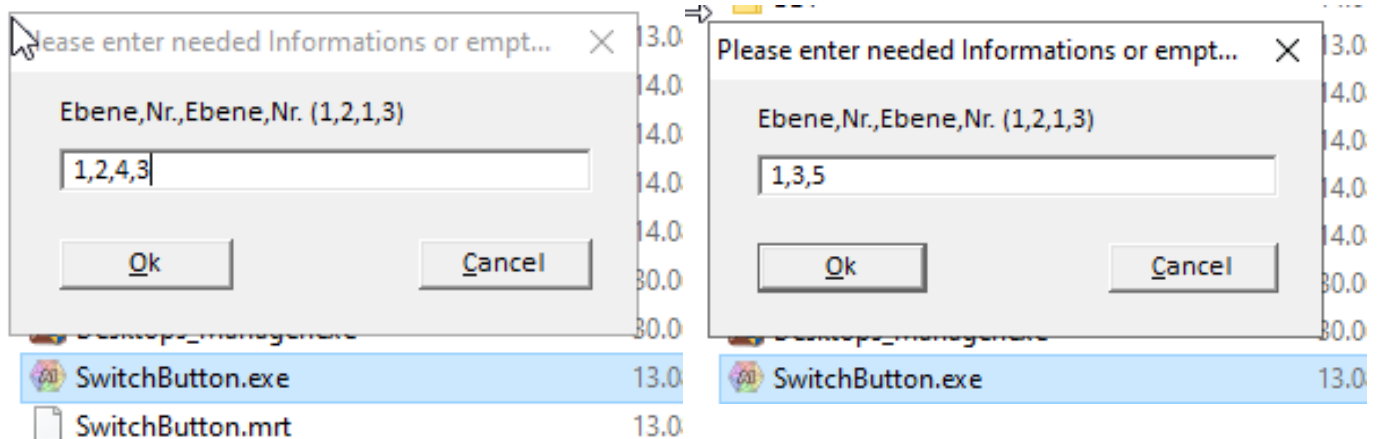
How to change or redefine Buttons?

In these Textfiles (picture) the Buttons—their Text—and their functions are described.

You can redefine all Buttons that have a number on the left side by editing the Textfiles.

You can also add or delete Buttons by deleting or adding Textfiles.

You can also exchange Buttons just by renaming their Textfiles. To make this easier, there is a tool „Buttonswitcher“ included. If you start it, you will get a INPUTBOX.



You can enter whether 4 Numbers like:

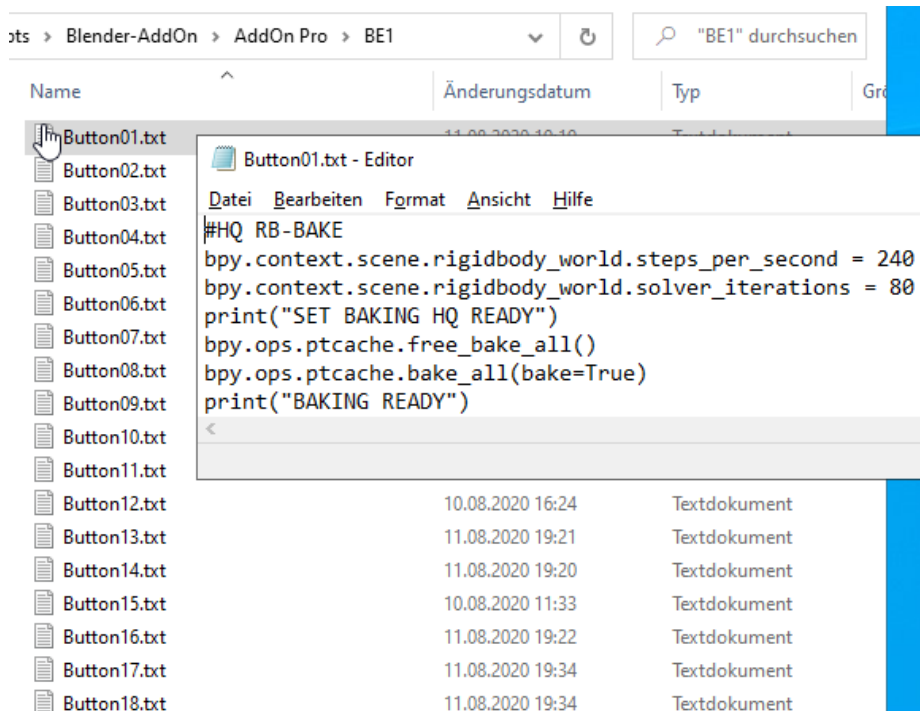
(Button-Layer A),(Button-Number A),(Button-Layer B),(Button-Number B)

Especially if you want switch Buttons that are not on the same Layer.

If both Buttons are on the same Layer, enter:

(Button-Layer A),(Button-Number A),(Button-Number B)

See pictures.



After pressing ENTER, the Buttons are exchanged, this will be visible once the Buttons get loaded new from the AddOn. Switching the Buttonlayer with the yellow button will make the changes visible.

Besides that you can also just change, delete and rename the Textfiles.

Just a Comment.

Text on the Button

Reference to the Library.py file.

Subprogram from the Library.py file.

Button10.txt - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
#Move Left/Right
from Library import *
Geo_Move_Left_by_own_SizeX()
#END
[OnRightMouseButton!]
from Library import *
Geo_Move_Right_by_own_SizeX()
#END
```

ts > Blender-AddOn > AddOn Pro > BE1

"BE1" durchsuchen

| Name | Änderungsdatum | Typ | Größe |
|--------------|------------------|--------------|--------|
| Button01.txt | 11.08.2020 19:19 | Textdokument | 1.0 KB |
| Button02.txt | | | |
| Button03.txt | | | |
| Button04.txt | | | |
| Button05.txt | | | |
| Button06.txt | | | |
| Button07.txt | | | |
| Button08.txt | | | |
| Button09.txt | | | |
| Button10.txt | | | |
| Button11.txt | | | |
| Button12.txt | 10.08.2020 16:24 | Textdokument | 1.0 KB |
| Button13.txt | 11.08.2020 19:21 | Textdokument | 1.0 KB |
| Button14.txt | 11.08.2020 19:20 | Textdokument | 1.0 KB |
| Button15.txt | 10.08.2020 11:33 | Textdokument | 1.0 KB |
| Button16.txt | 11.08.2020 19:22 | Textdokument | 1.0 KB |
| Button17.txt | 11.08.2020 19:34 | Textdokument | 1.0 KB |
| Button18.txt | 11.08.2020 19:34 | Textdokument | 1.0 KB |

Button01.txt - Editor

Datei Bearbeiten Format Ansicht Hilfe

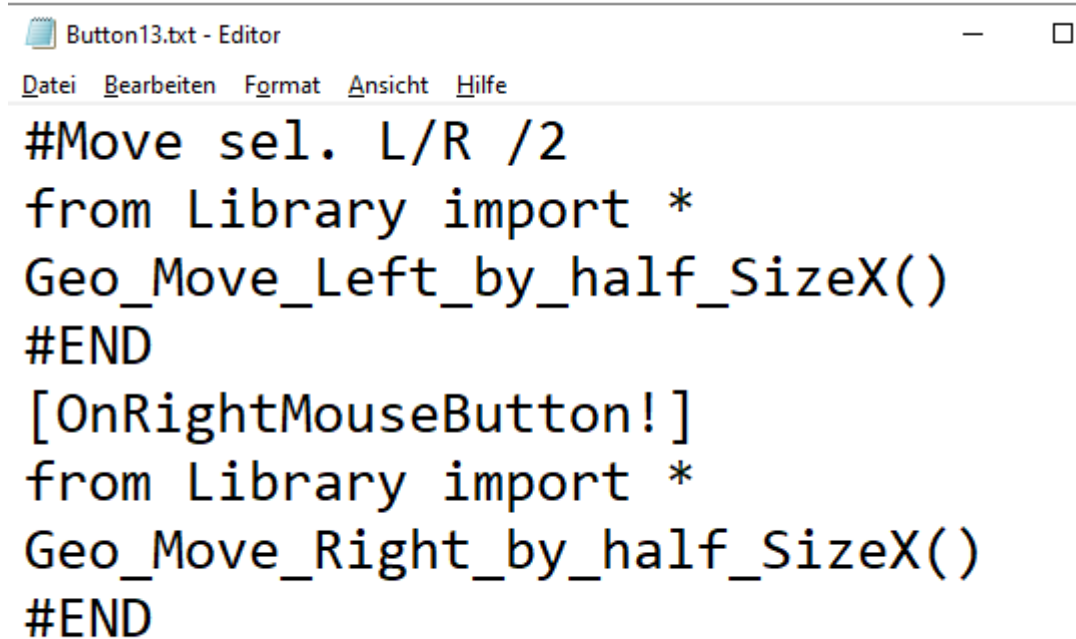
```
#HQ RB-BAKE
bpy.context.scene.rigidbody_world.steps_per_second = 240
bpy.context.scene.rigidbody_world.solver_iterations = 80
print("SET BAKING HQ READY")
bpy.ops.ptcache.free_bake_all()
bpy.ops.ptcache.bake_all(bake=True)
print("BAKING READY")
```

Rules:

1. Make sure that all Buttonfiles succeed each other, so don't leave holes!
2. All Buttons in one Layer together should not exceed 40 Buttons.
3. You can have up to 9 Layers. If someone needs more write me. There is no real technical limitation from having 99 Layers :-).

Left and Right Mousebutton-Feature.

To use both Mousebuttons on a Button, use the **[OnRightMouseButton!]** Feature.



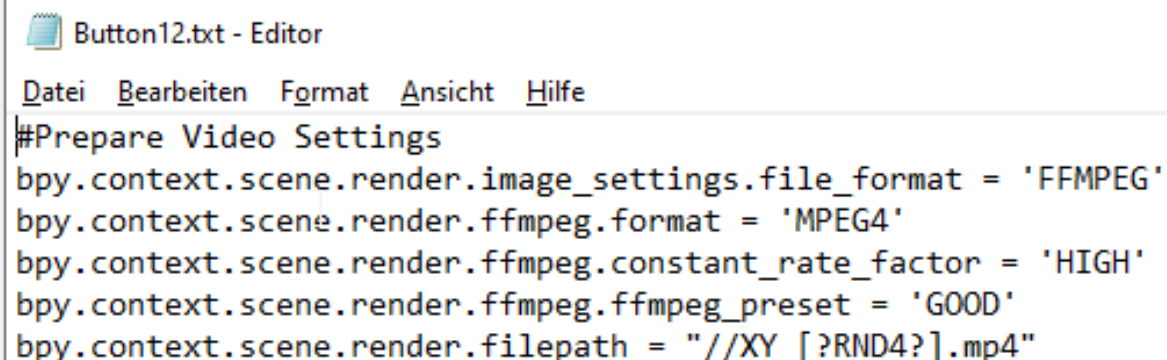
```
Button13.txt - Editor
Datei Bearbeiten Format Ansicht Hilfe
#Move sel. L/R /2
from Library import *
Geo_Move_Left_by_half_SizeX()
#END
[OnRightMouseButton!]
from Library import *
Geo_Move_Right_by_half_SizeX()
#END
```

The Random-Number Feature

If you use **[?RND4?]** in the Textfiles, this will be replaced with a 4 digit random number.

This way you can prevent already saved files from being overwritten for example when specifying filenames for SAVEing or File-Caches.

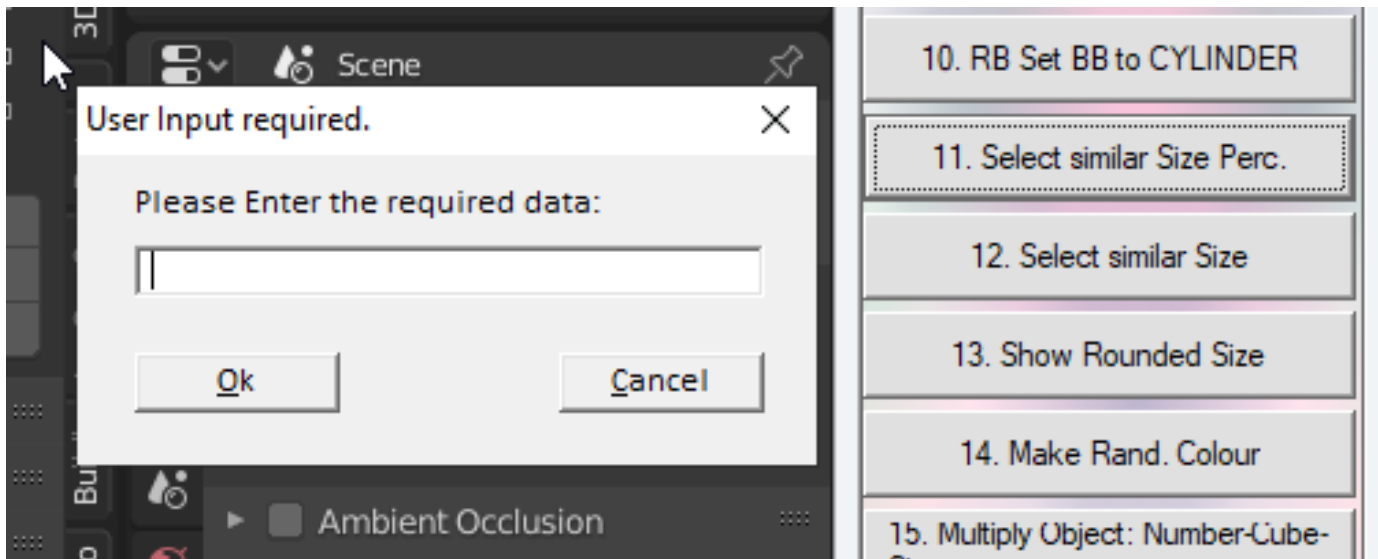
For example „//Myfile_[?RND4?].mp4“ could become „//Myfile_0369.mp4“ just a random 4 Digit number.



```
Button12.txt - Editor
Datei Bearbeiten Format Ansicht Hilfe
#Prepare Video Settings
bpy.context.scene.render.image_settings.file_format = 'FFMPEG'
bpy.context.scene.render.ffmpeg.format = 'MPEG4'
bpy.context.scene.render.ffmpeg.constant_rate_factor = 'HIGH'
bpy.context.scene.render.ffmpeg.ffmpeg_preset = 'GOOD'
bpy.context.scene.render.filepath = "//XY_[?RND4?].mp4"
```

SPECIAL FEATURES:

INPUTBOX-Feature.



In the Picture you can see a button that uses the „Input Box“ feature.

You can enter two comma-separate numbers and then press ENTER.

To use the Input-Box feature, use the [INP] directive.

The results will be filled in the Python-Code, where you place the

[INP1] ...[INP9] directives.



The Museum-Script in the Library.

The Museum-Script will generate a Museum-Gallery like Animation in Blender (see picture).

The pictures will be generated from the pictures in the "Pics"-Folder.

The number of pictures can be changed in the Script (NumPics-Variable).

You have to:

1. Add the Library to the Blender Library Path using the Addon-Button "Add Library"
2. You have to provide the pictures with the proper Names in one folder (use picture-renamer?)
3. You have to change the script that it knows your picture folder (see below)
4. run this Script from within the Blender-Text-Editor.

```
#####
```

```
# Pictures must be named "PXC_1.jpg","PXC_2.jpg" etc.
```

```
#####
```

```
# Change these variables to your needs
```

```
#####
```

NumPics=23

PicPath="F:\\00_MR\\PB_MiniRobots-Editoren\\SPRE 02\\Source\\Sample Scripts\\Blender-AddOn\\AddOn

```
Button13.txt - Editor
Datei Bearbeiten Format Ansicht Hilfe
#Move sel. L/R /2
from Library import *
Geo_Move_Left_by_half_SizeX()
#END
[OnRightMouseButton!]
from Library import *
Geo_Move_Right_by_half_SizeX()
#END
```

Pro\\Library\\Museum\\Pictures\\"

print(PicPath)

You need to change this that it fits to your system where the Folder with the pictures for the museum is.

All pictures that shall be used must be in the Picture-Folder and must be named "PXC_(number).jpg"

For example: "PXC_1.jpg", "PXC_2.jpg" etc.*

Once you start the Script the Script runs for a while.

At then end you get the ready Scene, you will just need to add some material to the Walls and the Floor.

#####

*see here in the script:

You have to correct this path to your system

Pictures must be named "PXC_1.pg", "PXC_2.pg" etc.

pat=PicPath+"PXC_"+str(picn)+".jpg"



The Museum-Script in the Library.

The Museum-Script will generate a Museum-Gallery like Animation in Blender (see picture).

The pictures will be generated from the pictures in the "Pics"-Folder.

The number of pictures can be changed in the Script (NumPics-Variable).

You have to:

1. Add the Library to the Blender Library Path using the Addon-Button "Add Library"
2. You have to provide the pictures with the proper Names in one folder (use picture-renamer?)
3. You have to change the script that it knows your picture folder (see below)
4. run this Script from within the Blender-Text-Editor.

```
#####
```

```
# Pictures must be named "PXC_1.jpg","PXC_2.jpg" etc.
```

```
#####
```

```
# Change these variables to your needs
```

```
#####
```

```
NumPics=23
```

```
PicPath="F:\\00_MR\\PB_MiniRobots-Editoren\\SPRE 02\\Source\\Sample Scripts\\Blender-AddOn\\AddOn Pro\\Library\\Museum\\Pictures\\"
```

```
print(PicPath)
```

You need to change this that it fits to your system where the Folder with the pictures for the museum is.

All pictures that shall be used must be in the Picture-Folder and must be named "PXC_(number).jpg"

For example: "PXC_1.jpg", "PXC_2.jpg" etc.*

Once you start the Script the Script runs for a while.

At then end you get the ready Scene, you will just need to add some material to the Walls and the Floor.

```
#####
```

*see here in the script:

```
# You have to correct this path to your system
```

```
# Pictures must be named "PXC_1.pg", "PXC_2.pg" etc.
```

```
pat=PicPath+"PXC_"+str(picn)+".jpg"
```

Video 1 Blender-Bar: This is an more actual Video that shows the current version of the Blender-bar.

<https://youtu.be/Z2O6r5L-xR0>

See this Video about the first version of the Buttonbar. Note that the video is not perfectly "Up to date" because in the time between the recording and now there have been several Improvements.

Video 2 Blenderbar

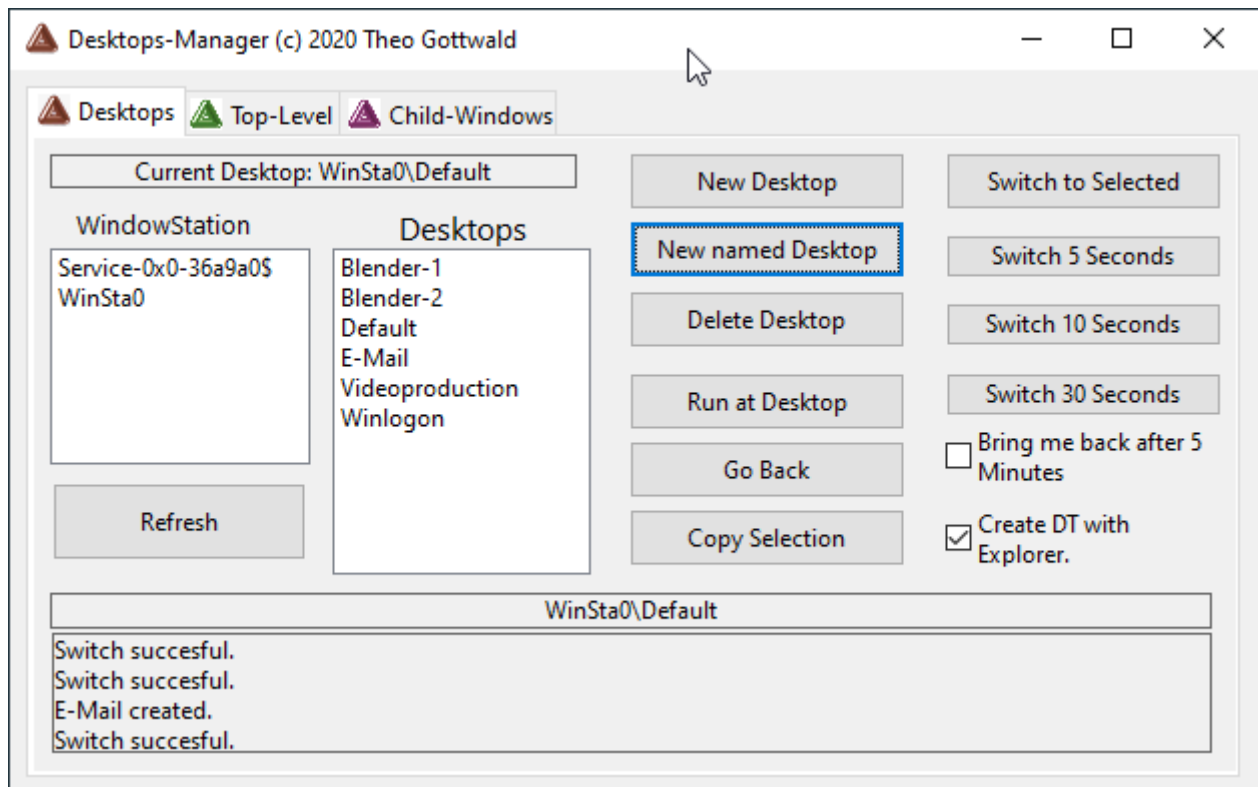
<https://youtu.be/A9Ogl0gWS2A>

Yet the Video shows the concepts and most of the buttons are still where they are in the video.

Included is our "**Desktop-Manager**" that will enable you to run multiple Instances of Blender on **multiple real Desktops**. and of course **with multiple Buttonbars**.

Is one of my favourite tools. You can organize your work on several desktops, and just switch to see how far a rendering currently is. Or work on one desktop, render on another.

Please note that through technical limitation, the windows explorer does not support all features on „Private Desktops“. The Startmenu may not open there, and also VM-Ware and few other programmes may not work as expected on „Private Desktops“. This is due to limitations of those programmes. You will see, which programmes work and which not. Blender works perfect on „Private Desktops“.



Using real Desktops in our Tests, Blender renders more stable if you have multiple Blender running on their own "Private Desktop".

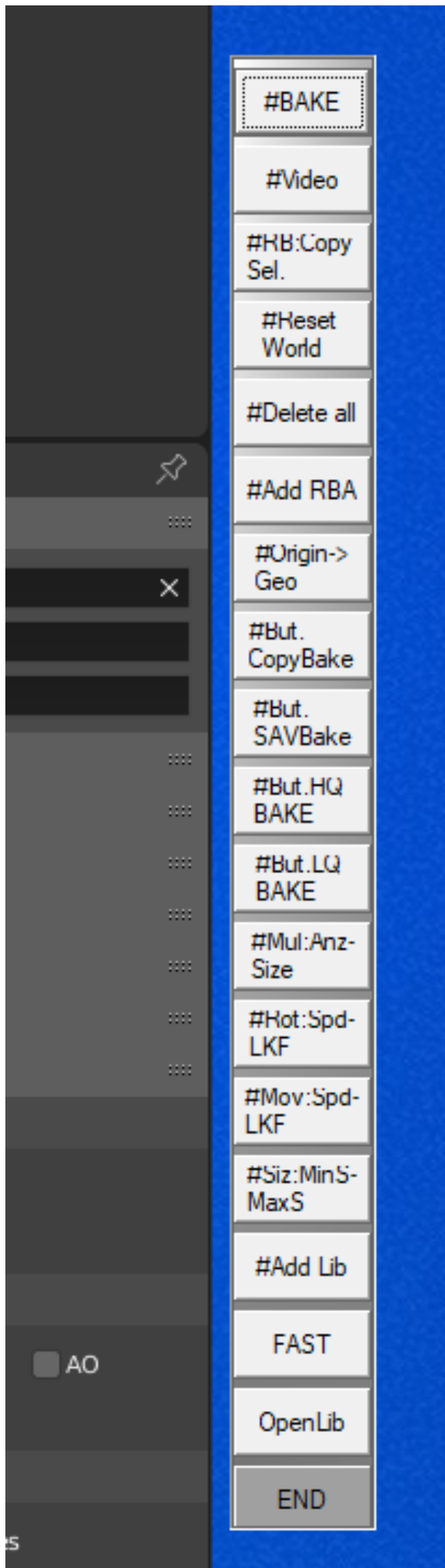
Note that our Desktops are REAL Desktops, not the "virtual Desktops" that come with Windows 10. Real Desktops have their own Window structure. Therefore moving Windows between „Private Desktops“ is impossible. They are isolated from each other (which improves stability).

Delete only Desktops that are newly (accidentally?) created. If you delete „older“ Desktops, you may end up with a „Black hole“ Desktop. Without Explorer and anything.

If you accidentally get stuck on such an desktop, the only way out is starting the Task-Manager (ctrl-ALT-ESC) and then running the Desktops-Manager from there.

Take a bit of time to explore this tool, it may bring a new dimension into your Workflow.

Note that Screen-Recorders do NOT record „real Desktops“, and mostly only the Default Desktop. Unless they get started there explicitly.



You want to recommend Blenderbar to a friend?

You want to try first if it works on your System?

Here is „Blenderbar-FREE“.

It has 20 Buttons that you can define with your own Pythoncode to test the system.

It's not a „downgraded“ or „time-limited“ Product. Ist perfectly usable, in fact, this was my first attempt to make a Blenderbar.

Also ist smaller and preferable for people with lower resolution-Screens or Laptops.

#####

If you want to test Blenderbar first, there is a free version available with up to 20 Buttons.

Look at my [Patreons Page](#) to download it.

Once you have used all of the 20 Buttons in the Free Version and **you want much more - 40 Buttons plus Multiple Layers** with each 40 Buttons) then this **PRO-Version** is the AddOn for you.

The Addon is external to Blender so its technically version independent. The Library has been updated for Blender 2.80 up to 2.90.

It is not as optimized as the new PRO-Blenderbar.

But it has all important features that you need to test the System.

And its free!

#####



Rendering „The Museum“.

Author:

Theo Gottwald
Herrenstr.11
76706 Dettenheim / Germany
Tel. 07247-9851112

E-Mail: info@smart-package.com
www.smart-package.com

Version-Info:

15.08.2020 First Release

+++++

16.8.2020 Second Release

Added Buttons Layer 4:

13 LMB: Set Camera Clipping Off / RMB:On

14.LMB: Lock Camera to Viewport On / RMB: Off

15. LMB: Set Location KF / RMB: Set Rotation Keyframe for all Selected Objects

16.LMB: Set Location and Rotation KF RMB: Set Scale KF for all selected Objects

17. Like Button 18, but will only delete the KF at the actual Animation Pointer Pos.

18.LMB: Delete all Location and Rotation Keyframes from selected Objects. RMB: LMB: Delete all Scale-Keyframes from selected Objects

19.LMB: Delete RB-Animation-Checkmark Keyframes at actual Position from all selected Objects.

RMB: Delete RB-Kinematic-Checkmark Keyframes at actual Position from all selected Objects.

20. Like Button 19, but will delete all such Keyframes from selected Objects

21. LMB: Delete all Location and Rotation KF from actual Camera. RMB: Show (Print) all Keyframes in Console Window

22. Set/Delete actual Camera Loc. And Rot.-KF at current Animation Pos.

Changed Algo for Blender Window:

1. When Blender.exe was found, SPR checks if its on this current Desktop and if so, if ist minimized.
2. When ist not minimized and on the current Desktop, the BB will attach to the existing Bledner-Window.
3. If Blender is minimized, or not on the current Desktop, then the SPR will assume that it is a „busy version“ and not touch it. Instead in such case, the SPR will start himself a new Blender. And attach to this.

Added Libchecker.exe

After many cases of Python Indentation errors, i have now made a primitive chcker that will only check the Indentation of a „.py“-File. For this just drop it onto the Icon. Give it some time.

If there are errors found, a Console Window will open and show the errors.

In any case a trimmed and corrected „Corrected.py“-File is written aside of the Program.

+++++