

This is the initial release of 'NP station' package. It's purpose is to join all the NP operators made over time.

The details of the addon are not polished, the code is yet to be cleaned. I think i managed to put the structure up and make all the operators work as they did individually. Also, there is a couple of new, untested operators. The addon settings should also be working.

The package is made in 2.75a and tested briefly in 2.75a and 2.78b. There was a lot of work, while not much time to test it in production. Please use with caution and avoid production use.

This package is meant to be free and open-source, GNU GPL software.

DESCRIPTION

NP operators are an experiment to see whether it is possible to emulate precision modeling tools and workflow in Blender, using it's existing technology. Operators are made to work mostly in object mode, intended for quick CAD-like sketching and modification, with real time user interaction and snapping procedures. The operators are mainly macros, manipulating with multiple Blender's modal operators.

NP station is at it's best working with cubic and flat-surfaced shapes. It is primarily intended for use in architecture design but it can help in other similar design and engineering areas. It does not feature organic shape or sculpting tools.

INSTALLATION

Unzip and copy the NP_020 folder to addons_contrib folder at your Blender app's location. Start Blender, go to User Preferences / Addons, turn on Testing filter and type 'NP' or '020' into the search box. Check the box next to the NP_020 name. If you have a previous version, first deactivate that one and then check the new one. A new tab with the name NP 020 will appear in the tool shelf (T) and will be active in object mode, addon's operative context.

LICENSING

The NP collection is not for sale and it is absolutely free.

I immensely appreciate the fact mighty Blender is a free and open source application, as well as numerous user-made addons, produced and donated to the community in good spirit. However, as the open source principles allow it, i am using Blender professionally and obtaining commercial gain. Therefore - i am in debt. I never donated money to Blender. The NP as well as the participation in the Blender upgrade discussions is the way i am paying back. This addon and all it's operators are free and will remain as such.

It's license, as with majority of Blender's addons is GPL2, and it is included in the bundle as a text file. You may copy, redistribute, change and use the addon as you find suitable as long as you redistribute the same license with it.

The only copyrighted aspect of it is the name. You can not use 'NP' or 'NP station' in your modifications.

ADDON SETTINGS

When you activate the addon in the User Preferences a panel segment will appear with the addon settings. The settings are divided into two parts:

Main addon settings:

Main color scheme: Color system that will be used for operator widgets and OpenGL graphics and text drawing. This should correspond to your main Blender theme since it will interact with it's colors. So far only two color schemes are built into this version. One for the default Blender theme and other for architectural sketching theme i made for my own workflow. Others will probably follow in future.

Size of the numerics: Large portion of operators deal with geometry and displays dimensions of elements such as length, width, height or distance. These operators share instructions on how to display these dimensions and this is where you define the size of the numerics. The default is 18.

Distance unit scale: NP operators can display dimensions independently from Blender unit system. This is where you define the scale of NP units in relation to Blender units. For example, if you consider Blender units to be meters (or you designated them to be) you can work in centimeters in NP by setting the scale of the units to 100, or millimeters by setting this number to 1000. The default is 100, for use in architecture sketching.

Distance unit suffix: If you would like to have units' suffix displayed on screen during NP operations, choose suffix from the drop-down list according to the unit scale you have chosen previously. If you have chosen centimeters, set the suffix to 'cm' instead of 'None'. If you wish for suffix not to be displayed, choose 'None'. The default is cm.

Mouse badge: In order to make instructions and info more visible i included an option for a little graphical element to be displayed alongside mouse cursor with basic notifications. It differs from operator to operator and this is where you enable it and define it's size. The default is 2.

Individual operator settings:

Based on their specialty, some operators have their own dedicated settings which appear below the drop-down selection box. They will be described individually, in relation to their operators.

OPERATORS

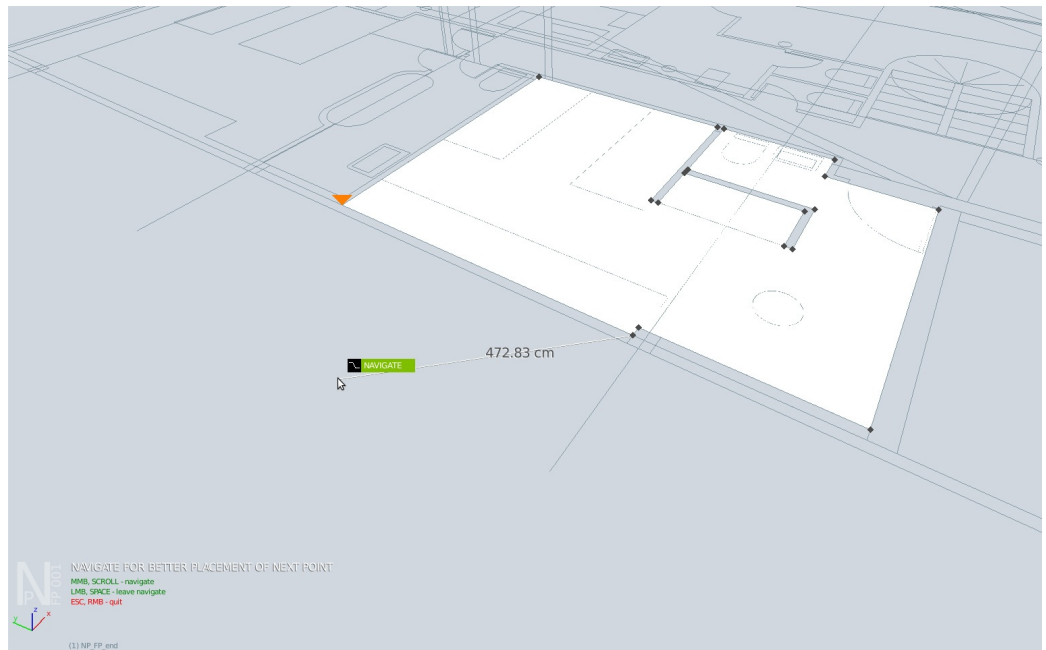
The following is a list of the operators that are included in this version:

- `NP_float_poly` Draws mesh lines and closed polygons using snap points;
- `NP_float_rectangle` Draws a mesh rectangle using snap points;
- `NP_float_box` Draws a mesh box using snap points;
- `NP_point_move` Moves selected objects using "take" and "place" snap points;
- `NP_point_copy` Duplicates selected objects using "take" and "place" snap points;
- `NP_point_instance` Duplicates selected objects as linked instances using "take" and "place" snap points;
- `NP_point_array` Makes a directional array of selected objects using "take" and "place" snap points;
- `NP_roto_move` Rotates selected objects using "start angle" and "end angle" snap points;
- `NP_point_scale` Scales selected objects using bounding box with handles;
- `NP_point_align` Aligns selected objects using 1, 2 or 3 pairs of snap points;
- `NP_shader_brush` Transfers materials between objects in 3D view;
- `NP_point_distance` Measures distance between two snapped points;

INDIVIDUAL OPERATOR DESCRIPTIONS

NOTE: Some information in this section is repeated from operator to operator, to enable modular reading.

NP_float_poly



IDEA:

This is an experiment to see if a basic 'draw' function is possible in Blender, upgrading the Blender's creation paradigm "create a preset shape and modify it to your needs" to "create what you need, where you need it". It emulates the functionality of the standard polyline command in CAD applications. The user sets the start of the line with snapping enabled and goes on designating points of the polyline. Afterwards, the operator optionally extrudes and bevels the newly created shape. Escaping the tool during extrusion sends the shape back to 2D, making the floors and other surfaces possible. Very useful for generating 3D objects on top of CAD blueprints, whether architectural or other technical nature. Good for making walls, floors or slabs. You can also use it without the blueprint and make arbitrary shapes by applying the numerical input.

RUN:

- float_poly button on NP tab (displayed in object mode)
- spacebar search - NP Float Poly
- keystroke if you assigned it

USAGE:

Select a point anywhere in the scene (holding CTRL enables snapping). This will be your start point. Move your mouse and click to a point anywhere in the scene with the left mouse button (LMB), in relation to the start point (again CTRL - snap). The addon will make a line segment from the first to the second point. You can continue adding other points in the same way. Middle mouse button (MMB) enables axis constraint during snapping, while numpad keys enable numerical input of the poly segment length. **ENTER** key changes snap target from *vertex* to *edge* to *face*. In case ENTER is pressed again the cycle starts over from *vertex*. RMB and ESC keys interrupt the operation. When you want to finish the poly, press ESC or if you want to close the polyline and make a surface, press right mouse button (RMB).

If at any point you lose sight of the next point you want to snap to, you can press SPACE and go to **NAVIGATION** mode in which you can change the point of view using standard MMB drag for orbiting and SCROLL for zoom. When your next point is clearly in your field of view, you return to normal mode by pressing SPACE or LMB again.

After the closure of the poly, the command will automatically start the extrusion of the surface into 3D. You can confirm this with the LMB or avoid the extrusion with ESC which will restrict the poly to 2D surface.

After the extrusion, if you enabled the bevel function in the addon settings, the script will start the bevel operation which you control as usual - LMB for the amount and MMB scroll for the number of segments.

NOTES:

Depending of the direction in which the user defined the base shape (clock / counter clock), the extrusion values in the 'up' direction could go in positive / negative direction. The addon's dimension display disregards the sign of the dimensions so pay attention to the header values in the viewport, they will display what you need to type in.

While in 'define_extrusion_height' phase, the operator still doesn't display the height on-screen, a legacy from the old operator. Probably will be improved over time.

The UNDO command after the use of the operator requires cycling through certain phases of the command, so be careful to click undo enough times to delete all the helper vertexes that were created during the operation. This is a limitation of the macro nature of the addon and hopefully will be upgraded in the future.

SETTINGS:

Common settings:

The operator inherits unit scale, unit suffix, line color and shadow from main addon settings. It will share these settings with other operators that have line and numerical distance, no individual overrides exist yet.

Individual settings:

Point markers: An option to use a graphical representation of points added to polyline for better overview

Default point color: Define the color you would like the point markers to be

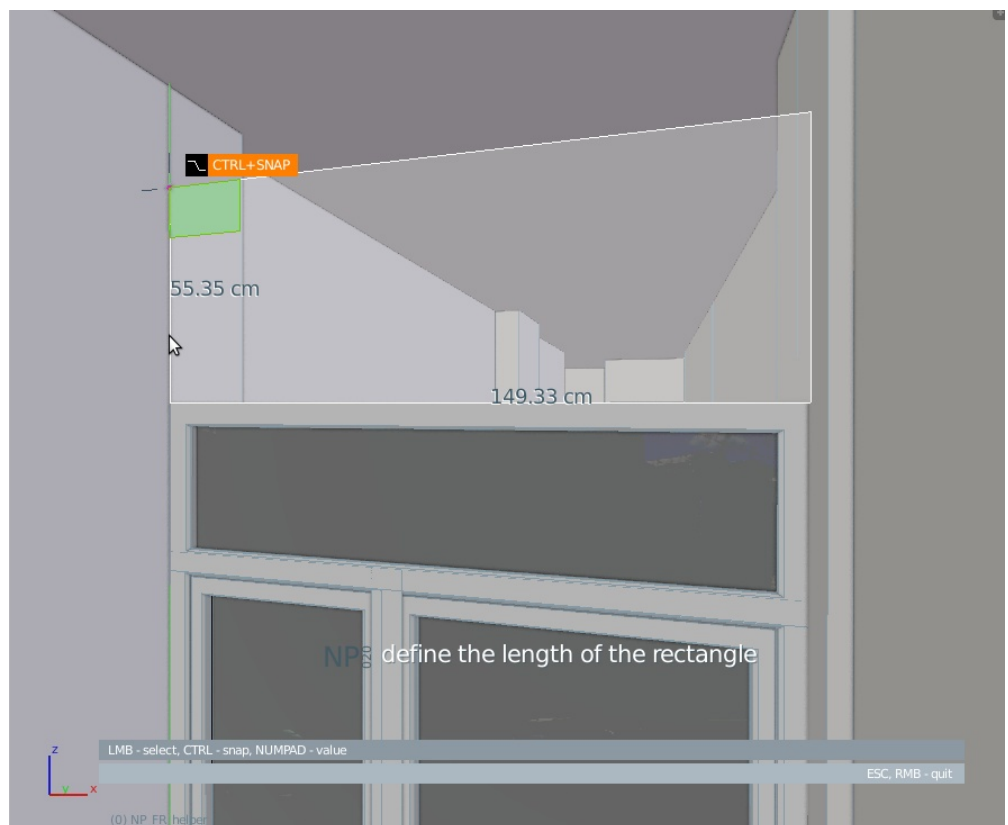
Bevel: An option to include interactive bevel procedure after the extrusion of the poly shape

Base material: An option to automatically assign a new material to the newly created object

Smooth shading: An option to automatically turn on smooth shading for this object. Useful for objects made with bevel option turned on

Wire contour: An option to show wireframe on newly created object, emulating sketchy style of display, enabling better geometry overview

NP_float_rectangle



IDEA:

Float rectangle is the first in a series of **Geo tools** operators planned for this addon. They will experiment with in-situ, modal parametric creation of the basic geometrical shapes and bodies (what you need / where you need). These will include tools for creating rectangular and circular shapes, n-gons, prisms, cylinders etc. The set will be structured in a way that will use common mini-engines for creation and display, so any upgrade to those engines will reflect all Geo operators. When the basis of this set functionality is fully defined and polished, extending the set to new types of geometry will be pretty straight forward. In this release, the mini engines are located in sub-modules named `utils_geometry.py` and `utils_display.py`.

As for the UI procedures, the float_rectangle operator has a somewhat different approach to drawing plane selection in the roto_move operator. The plane widget dances around *while* the user searches for the starting point, and by pressing ENTER user locks the plane for the rectangle generation (grab operator stops / exists / locks plane / resumes). I did this intentionally in order to test different UI approaches.

RUN:

- a. float_rectangle button on NP tab (displayed in object mode)
- b. spacebar search - NP Float Rectangle
- c. keystroke if you assigned it

USAGE:

After starting the operator you will be required to designate the starting point of the rectangle. The operator will test whether there is geometry underneath the cursor and track the surface waiting to pin the first point to it. You will notice a dancing Geo widget in shape of the chosen primitive, with base plane of the shape highlighted. In this case, being a 2D shape, the whole rectangle of the geo widget will be highlighted in a transparent color. This color will depend on the orientation of the surface underneath the cursor. If the surface beneath the cursor is looking straight up, the widget will be blue. If the surface normal is pointing in the +x / -x or +y / -y (left / right / forward / back) direction, the widget will go red / green respectively. If the surface is pointing in an arbitrary direction the geo widget will go gray, and if there is no surface, the color will be blue / green / red depending of the view direction. Pressing ENTER locks the drawing plane in current state and the future rectangle will have this orientation.

Select a point anywhere in the scene (holding CTRL enables snapping). This will be your start point and one of the corners of the future rectangle. If you did not lock the drawing plane prior to this, it will be locked in the orientation shown immediately before the click. Move your mouse along the designated direction and click to a point anywhere in the scene with the left mouse button (LMB), again with CTRL - snap. The addon will make a line from the first to the second point, defining the first side of the rectangle. The geowidget will slide along and show the selected drawing plane. The third point you designate will define the last dimension and after that the operator finishes and creates the rectangle.

You can interrupt the operator and exit the command at any time by pressing ESC.

NOTES:

This operator is still in experimental phase, NAVIGATION mode is not applied yet.

The widget size engine is not polished yet, when you zoom in too close to drawing plane the widget gradually disappears and OpenGL display stops functioning while the operator is still active and able to finish the procedure. This will hopefully be upgraded in the future.

The draw_plane engine is doing special calculations when there is no surface under the cursor and produces the plane for the shape to be drawn on based on the view coordinates. You can see that when you look up from the viewport base plane and try to draw on empty space. This feature is still experimental and may not work as intended in all circumstances.

Try not to change axis while in second and third phase (hitting MMB while defining width and height), the operator expects dimensions in the prepared axes of the chosen plane.

The UNDO command after the use of the operator requires cycling through certain phases of the command, so be careful to click undo enough times to delete all the helper vertexes that were created during the operation. This is a limitation of the macro nature of the addon and hopefully will be upgraded in the future.

SETTINGS:

Common settings:

The operator inherits unit scale, unit suffix, and color scheme from main addon settings. It will share these settings with other operators that have lines and numerical distance displayed, no individual overrides exist yet.

Individual settings:

No individual settings for Geo operators exist yet.

NP_float_box



IDEA:

This is the second in a line of Geo tools operators and first one to go into 3D. It is based heavily on the float_rectangle operator and shares a lot of functionality with it. This will probably be the case with most of the

future geo operators. They will experiment with in-situ, modal parametric creation of the basic geometrical shapes and bodies (what you need / where you need). These will include tools for creating rectangular and circular shapes, n-gons, prisms, cylinders etc. The set will be structured in a way that will use common mini-engines for creation and display, so any upgrade to those engines will reflect all Geo operators. When the basis of this set functionality is fully defined and polished, extending the set to new types of geometry will be pretty straight forward. In this release, the mini engines are located in sub-modules named `utils_geometry.py` and `utils_display.py`.

As for the UI procedures, the `float_box`, as well as the `float_rectangle` operator, has a somewhat different approach to drawing plane selection in the `roto_move` operator. The plane widget dances around *while* the user searches for the starting point, and by pressing ENTER user locks the plane for the box generation.

RUN:

- a. `float_box` button on NP tab (displayed in object mode)
- b. spacebar search - NP Float Box
- c. keystroke if you assigned it

USAGE:

After starting the operator you will be required to designate the starting point of the box. The operator will test whether there is geometry underneath the cursor and track the surface waiting to pin the first point to it. You will notice a dancing Geo widget in shape of the chosen primitive, with base plane of the shape highlighted. In this case, the base rectangle of the box will be highlighted with a transparent color. This color will depend on the orientation of the surface underneath the cursor. If the surface beneath the cursor is looking straight up, the widget will be blue. If the surface normal is pointing in the $+x$ / $-x$ or $+y$ / $-y$ (left / right / forward / back) direction, the widget will go red / green respectively. If the surface is pointing in an arbitrary direction the geo widget will go gray, and if there is no surface the color will be blue / green / red depending of the view direction. Pressing ENTER locks the drawing plane in current state and the future rectangle will have this orientation.

Select a point anywhere in the scene (holding CTRL enables snapping). This will be your start point and one of the corners of the future box. If you did not lock the drawing plane prior to this, it will be locked in the orientation shown immediately before the click. Move your mouse along the designated direction and click to a point anywhere in the scene with the left mouse button (LMB), again with CTRL - snap. The addon will make a line from the first to the second point, defining the first side of the box. The geowidget will slide along and show the selected drawing plane. The third point you designate will define the depth dimension of the base. After that the operator requires one more snap point for the height of the box, upon which it finishes and creates the box.

You can interrupt the operator and exit the command at any time by pressing ESC.

NOTES:

The geo operators are still in experimental phase, **NAVIGATION** mode is not applied yet.

The widget size engine is not polished yet, when you zoom in too close to drawing plane the widget gradually disappears and OpenGL display stops functioning while the operator is still active and able to finish the procedure. This will hopefully be upgraded in the future.

The draw_plane engine is doing special calculations when there is no surface under the cursor and produces the plane for the shape to be drawn on based on the view coordinates. You can see that when you look up from the viewport base plane and try to draw on empty space. This feature is still experimental and may not work as intended in all circumstances.

Try not to change axis while in second and third phase (hitting MMB while defining width and height), the operator expects dimensions in the prepared axes of the chosen plane.

The UNDO command after the use of the operator requires cycling through certain phases of the command, so be careful to click undo enough times to delete all the helper vertexes that were created during the operation. This is a limitation of the macro nature of the addon and hopefully will be upgraded in the future.

SETTINGS:

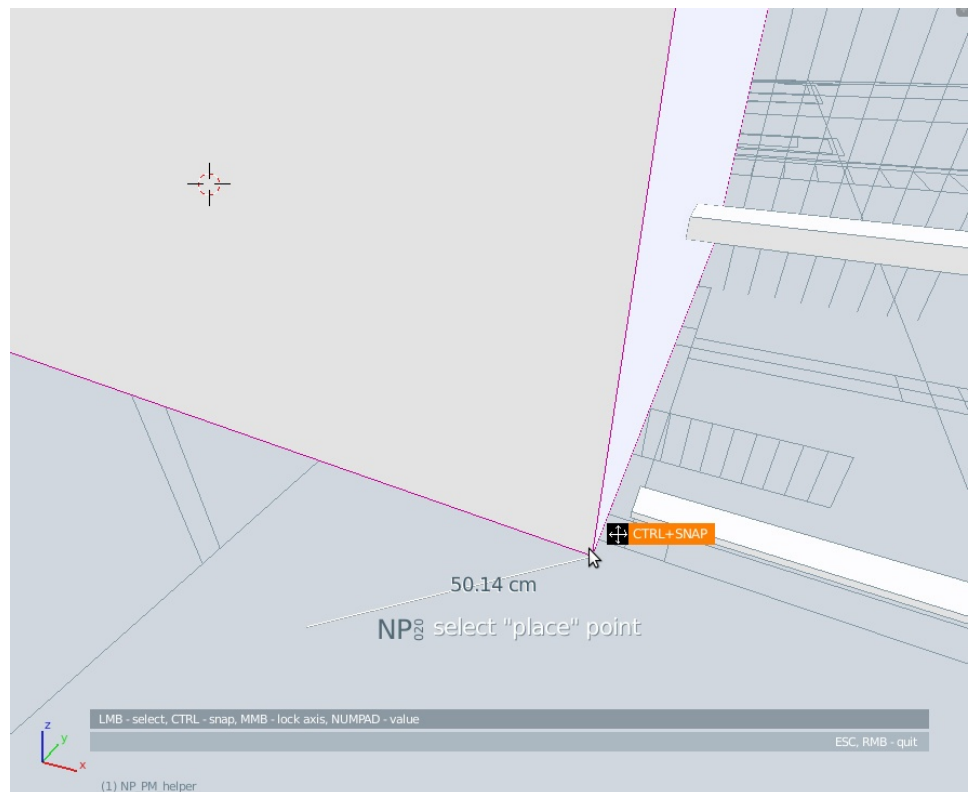
Common settings:

The operator inherits unit scale, unit suffix, and color scheme from main addon settings. It will share these settings with other operators that have lines and numerical distance displayed, no individual overrides exist yet.

Individual settings:

No individual settings for geo operators exist yet.

NP_point_move



IDEA:

This was the first real NP operator written, using basepoint snapping paradigm. It emphasized the missing object transformation features needed by the technical members of the Blender community. My workflow was pretty hampered without it and upon its completion a couple years ago it proved to be essential for architectural precision sketching and it received positive comments from friends and colleagues using it. It basically translates objects using 'take' and 'place' snap points. It emulates the functionality of the standard 'move' command in CAD applications, with start and end points. This way, it does pretty much what the basic 'grab' function does, only it locks the snap ability to one designated point in selected group, giving more control and precision to the user. It also eliminates the need for post - operation tweaking of object position in various ortho views. It finishes the translation precisely, fast and without the need for view change.

RUN:

- a. point_move button on NP tab (displayed in object mode)
- b. spacebar search - NP Point Move
- c. keystroke if you assigned it

USAGE:

Make sure that you are in object mode. Select one or more objects that need to be moved. Select a point anywhere in the scene (holding CTRL enables snapping). This will be your 'take' point. Place objects anywhere in the scene, in relation to the 'take' point (again CTRL - snap). The operator will follow the movement with a straight line trajectory and numerical distance according to your chosen unit scale and suffix. Middle mouse button (MMB) enables axis constraint, numpad keys enable numerical input of distance, and RMB and ESC key interrupt the operation.

NOTES:

This operator is freshly introduced to the NP package from an older anchor_translate addon i made years ago and unlike its close younger relatives point_copy and point_instance, it does not feature **NAVIGATION** mode yet. It should be implemented in the future.

The UNDO command after the use of the operator requires cycling through certain phases of the command, so be careful to click undo enough times to delete all the helper vertexes that were created during the operation. This is a limitation of the macro nature of the addon and hopefully will be upgraded in the future.

SETTINGS:

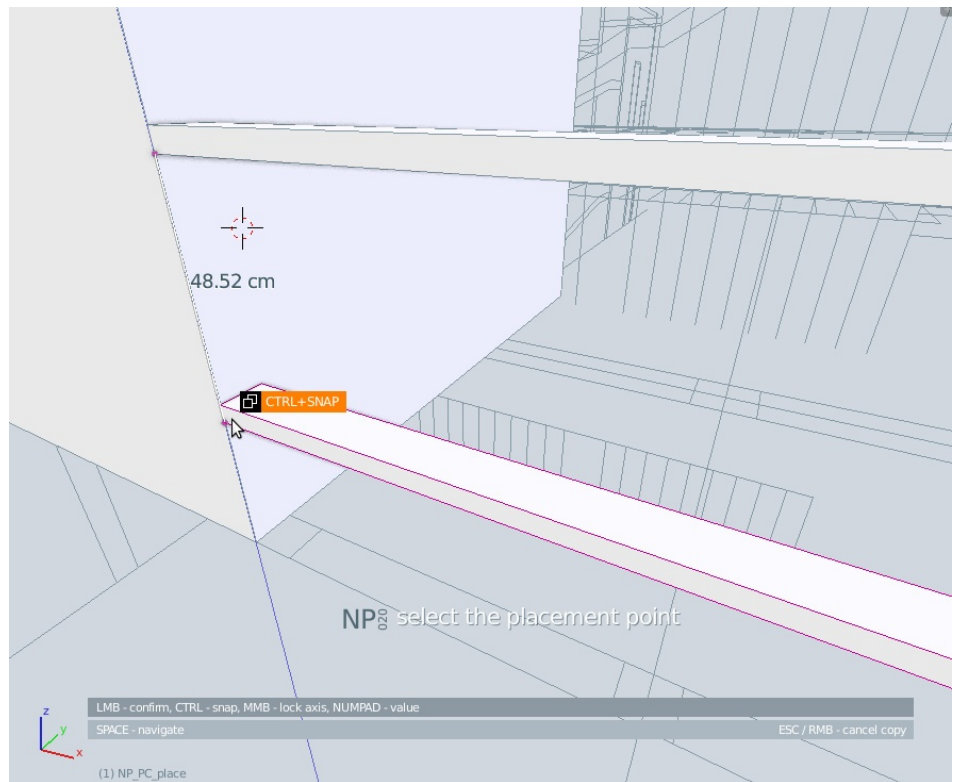
Common settings:

The operator inherits unit scale, unit suffix, and color scheme from main addon settings. It will share these settings with other operators that have lines and numerical distance displayed, no individual overrides exist yet.

Individual settings:

No individual settings for this operator exist yet.

NP_point_copy



IDEA:

Long time planned NP operator, benefits heavily on the technology introduced by point_move. It basically duplicates objects using 'take' and 'place' snap points. It emulates the functionality of the standard 'copy' command in CAD applications, with start and end points. This way, it does pretty much what the basic 'duplicate' function does, only it locks the snap ability to one designated point in selected group, giving more control and precision to the user. It also eliminates the need for post - operation tweaking of object position in various ortho views. It finishes the duplication precisely, fast and without the need for view change. Also, has the ability to repeat the command until the user stops it.

RUN:

- point_copy button on NP tab (displayed in object mode)
- spacebar search - NP Point Copy
- keystroke if you assigned it

USAGE:

Make sure that you are in object mode. Select one or more objects that need to be copied. Select a point anywhere in the scene (holding CTRL enables snapping). This will be your 'take' point. Move your mouse and click to a point anywhere in the scene with the left mouse button (LMB), in relation to the 'take' point and the operator will duplicate the selected objects at that position (again CTRL - snap enables snapping to objects around the scene). You can continue duplicating objects in the same way. The operator will follow the movement with a straight line trajectory and numerical distance according to your chosen unit scale and suffix.

If at any point you lose sight of the next point you want to snap to, you can press SPACE and go to **NAVIGATION** mode in which you can change the point of view using standard MMB drag for orbiting and SCROLL for zoom. When your next point is clearly in your field of view, you return to normal mode by pressing SPACE or LMB again.

Middle mouse button (MMB) enables axis constraint, numpad keys enable numerical input of distance, and RMB and ESC key interrupt the operation.

NOTES:

The UNDO command after the use of the operator requires cycling through certain phases of the command, so be careful to click undo enough times to delete all the helper vertexes that were created during the operation. This is a limitation of the macro nature of the addon and hopefully will be upgraded in the future.

SETTINGS:

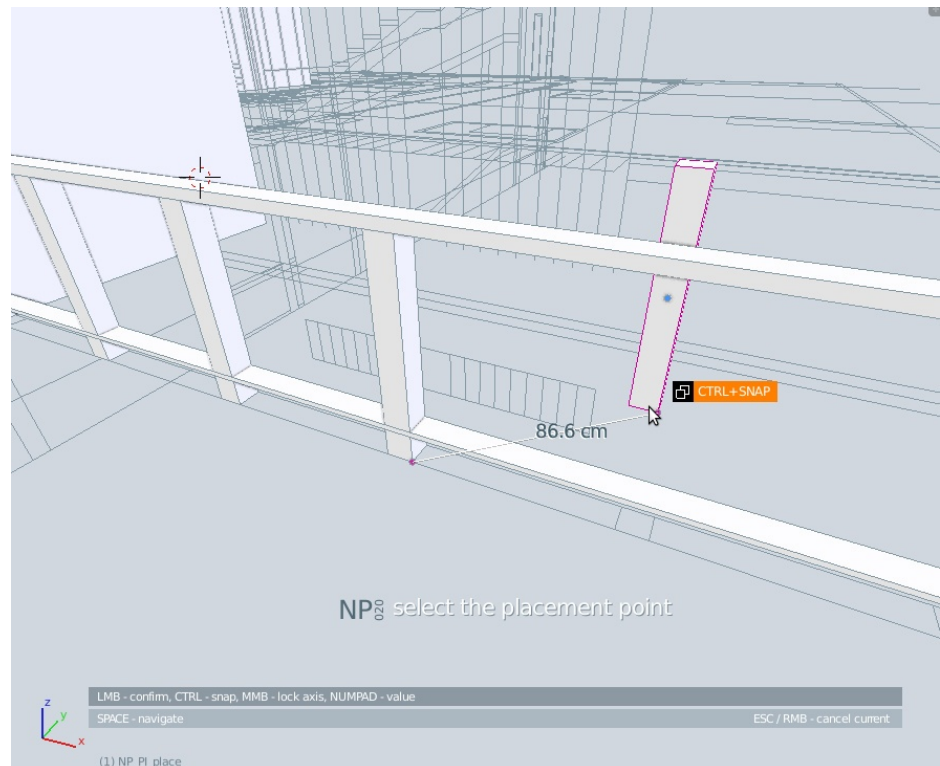
Common settings:

The operator inherits unit scale, unit suffix, and color scheme from main addon settings. It will share these settings with other operators that have lines and numerical distance displayed, no individual overrides exist yet.

Individual settings:

No individual settings for this operator exist yet.

NP_point_instance



IDEA:

Alongside point_copy, a long time planned NP operator, benefits heavily on the technology introduced by point_move. It basically duplicates objects with linked mesh data, using 'take' and 'place' snap points. Emulates the functionality of the standard 'copy' command in CAD applications, with the difference of duplicated objects acting the same during modifications. This way, it does what the basic 'duplicate linked' function does, only it locks the snap to one designated point in selected group, giving more control and precision to the user. It also eliminates the need for post - operation tweaking of object position. It finishes the duplication precisely, fast and without the need for view change. Also, has the ability to repeat the command until the user stops it.

RUN:

- point_instance button on NP tab (displayed in object mode)
- spacebar search - NP Point Instance
- keystroke if you assigned it

USAGE:

Make sure that you are in object mode. Select one or more objects that need to be instanced. Select a point anywhere in the scene (holding CTRL enables snapping). This will be your 'take' point. Move your mouse and click to a point anywhere in the scene with the left mouse button (LMB), in relation to the 'take' point and the operator will link - duplicate the selected objects at that position (again CTRL - snap enables snapping to objects around the scene). You can continue instancing objects in the same way. The operator will follow the movement with a straight line trajectory and numerical distance according to your chosen unit scale and suffix.

If at any point you lose sight of the next point you want to snap to, you can press SPACE and go to **NAVIGATION** mode in which you can change the point of view using standard MMB drag for orbiting and SCROLL for zoom. When your next point is clearly in your field of view, you return to normal mode by pressing SPACE or LMB again.

Middle mouse button (MMB) enables axis constraint, numpad keys enable numerical input of distance, and RMB and ESC key interrupt the operation.

NOTES:

The UNDO command after the use of the operator requires cycling through certain phases of the command, so be careful to click undo enough times to delete all the helper vertexes that were created during the operation. This is a limitation of the macro nature of the addon and hopefully will be upgraded in the future.

SETTINGS:

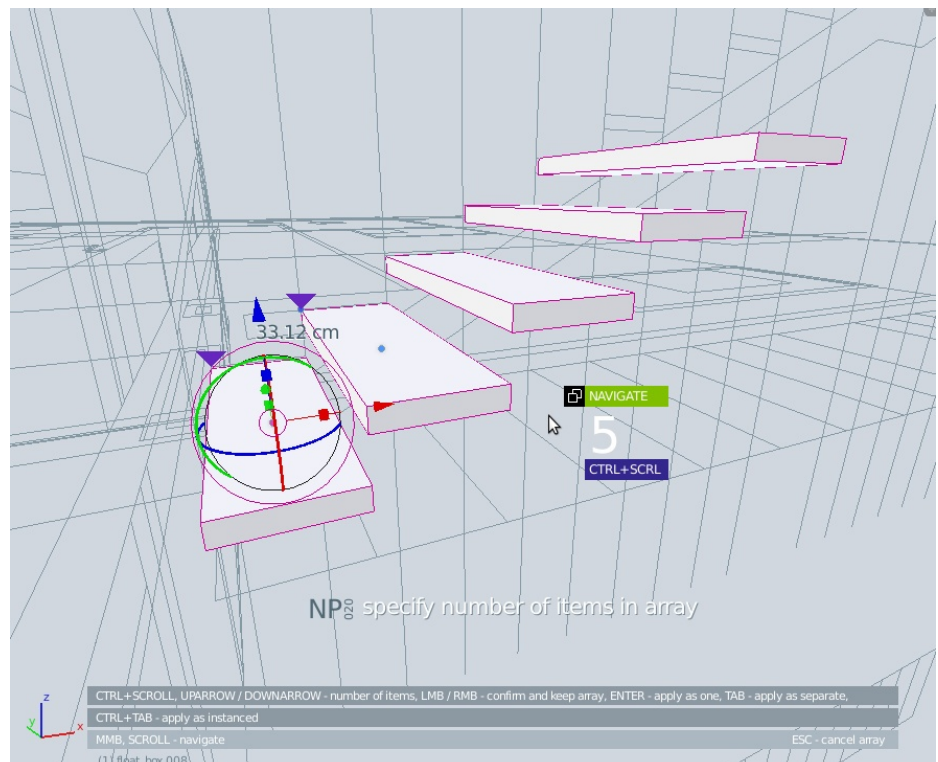
Common settings:

The operator inherits unit scale, unit suffix, and color scheme from main addon settings. It will share these settings with other operators that have lines and numerical distance displayed, no individual overrides exist yet.

Individual settings:

No individual settings for this operator exist yet.

NP_point_array



IDEA:

After the `point_copy` command a logical step further, a freshly introduced NP operator. Also benefits heavily on the technology introduced by `point_move`. It makes an array of selected objects using 'take' and 'place' snap points. It emulates the functionality of the standard 'array' command in CAD applications. This way, it does pretty much what the basic 'add modifier / array' function does, only it does it in an interactive way while acting similar to `point_copy` operator.

RUN:

- a. `point_array` button on NP tab (displayed in object mode)
- b. spacebar search - NP Point Array
- c. keystroke if you assigned it

USAGE:

Make sure that you are in object mode. Select one or more objects that need to be arrayed. Select a point anywhere in the scene (holding CTRL enables snapping). This will be your 'take' point. Move your mouse and click to a point anywhere in the scene with the left mouse button (LMB), in relation to the 'take' point. The command will automatically read the direction and the distance between the designated points, make an initial 5 - member array and present an interface to modify the number of arrayed copies. You specify the number with CTRL + mouse scroll, with the possibility to go below the amount of 2 which changes the mode of array to division. You confirm the array with RMB / ENTER / TAB key or cancel it with ESC. Pressing RMB at the end will confirm the array and keep it as a modifier in the modifier stack, ENTER will apply the modifier as a single object and remove the modifier, while TAB would apply the modifier as an array of separate individual objects and remove the modifier from the modifier stack.

During the designation of the first two instances of the array the operator will follow the movement with a straight line trajectory and numerical distance according to your chosen unit scale and suffix.

If at any point you lose sight of the next point you want to snap to, you can press SPACE and go to **NAVIGATION** mode in which you can change the point of view using standard MMB drag for orbiting and SCROLL for zoom. When your next point is clearly in your field of view, you return to normal mode by pressing SPACE or LMB again.

Middle mouse button (MMB) enables axis constraint, numpad keys enable numerical input of distance, and RMB and ESC key interrupt the operation.

NOTES:

The ARRAY command engine relies on complex calculations in order to take current object transformations to produce the array vector and it may not work as expected in all circumstances.

The UNDO command after the use of the operator requires cycling through certain phases of the command, so be careful to click undo enough times to delete all the helper vertexes that were created during the operation. This is a limitation of the macro nature of the addon and hopefully will be upgraded in the future.

SETTINGS:

Common settings:

The operator inherits unit scale, unit suffix, and color scheme from main addon settings. It will share these settings with other operators that have lines and numerical distance displayed, no individual overrides exist yet.

Individual settings:

No individual settings for this operator exist yet.

NP_roto_move



IDEA:

This operator is basically a standard Blender's rotate command enriched with snapping ability for 'start' and 'end' angles as well as fast transformation plane selection. It emulates the functionality of the standard 'rotate' command in CAD applications, with 'start' and 'end' angle. More complex aspect here is the introduction of 3rd dimension and the need for rotation plane selection (rotation axis emerging from it), which was resolved similar to Geo tools drawing plane selection - with the use of a 'dancing' widget.

Although made possible with the introduction of movable / snappable functions and helper objects that serve as snapping designators (similar to most NP operators), this operator introduces another fairly complex transformation aspect - rotation. That in return put on the table a whole new set of mathematical problems.

Although i managed to put Blender's rotate operator under control and integrate it into the command, the elusive arbitrary nature of object rotation intended to serve as a mediator for info display turned out to be a tough opponent. The mini-engine that was supposed to display the angle value that appears in the header during rotation did not end up in it's intended quality. Although the user will be able to precisely use the operator, input the rotation angle and get the desired results, the widget and the numerics display will not always correspond to the exact angle value. This remains to be corrected in future releases. When the operator reaches it's mature state it should serve as a basis for other, more complex rotation tools.

RUN:

- a. roto_move button on NP tab (displayed in object mode)
- b. spacebar search - NP Roto Move
- c. keystroke if you assigned it

USAGE:

Make sure that you are in object mode. Select one or more objects that need to be rotated. Upon activation a roto widget will appear. Unlike the float_rectangle and other geo tools which handle first point and drawing plane selection at once, roto move handles it separately, requiring the user to designate the *rotation center* first with LMB (holding CTRL enables snapping). Moving the mouse afterwards while roto widget adjusts it's orientation to the surfaces under the cursor, and clicking to a surface or empty point anywhere in the scene defines the rotation plane (together with the rotation axis). Next is the starting angle - by LMB snapping the user designates the starting handle for the rotation and with the last snap designates the final rotation angle. The operator will follow the user's actions with the corresponding widget behavior.

As with most of the NP operators, RMB and ESC keys interrupt the operation.

NOTES:

The angle mini-engine that was supposed to display the same angular value that appears in the header during rotation did not end up in it's intended quality. Although the user will be able to precisely use the operator, input the rotation angle and get the desired results, the widget and the numerics display will not always correspond to the exact angle value. This remains to be corrected in future releases.

This operator is still in experimental phase, **NAVIGATION** mode is not applied yet.

The widget size engine is not polished yet, when you zoom in too close to drawing plane the widget gradually disappears and OpenGL display stops functioning while the operator is still active and able to finish the procedure. This will hopefully be upgraded in the future.

The UNDO command after the use of the operator requires cycling through certain phases of the command, so be careful to click undo enough times to delete all the helper vertexes that were created during the operation. This is a limitation of the macro nature of the addon.

SETTINGS:

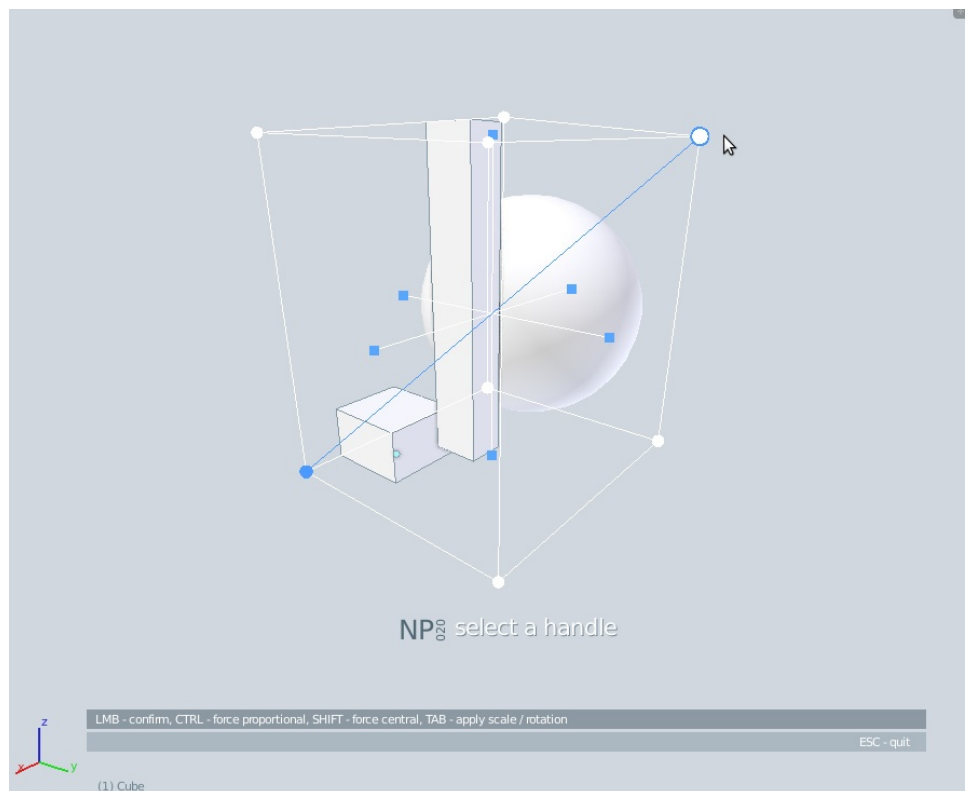
Common settings:

Being in the experimental phase, this operator is still not connected to main addon settings and does not feature adjustable graphical and functional aspects.

Individual settings:

No individual settings for this operator exist yet.

NP_point_scale



IDEA:

This is probably the best illustrator of 3D cursor obsolescence. It is an old reconstructed NP cage_scale operator, now with the Blender's resize operator at its core. It relies primarily on OpenGL on-screen drawing and fast OpenGL selection. It basically shortens the procedures for scale pivot point and scale axis selection through the use of on-screen handles drawn around the selected objects. It emulates the functionality of the cage scaling commands in popular graphics applications. It does pretty much what the basic 'scale' function does, only it enables automate and intuitive pivot point and direction selection, giving more control and speed to the user. It also eliminates the need for post - scale tweaking of object position. Useful in resizing a tree, a car, a glass, a boat, a man, an arm, a pipe, a flower, basically everything that is touching another object on either one of its sides and where the user needs to designate the pivots and constraints fast.

For example: in order to choose a pivot point to be on one end of (for example) the tube in order to scale it in the opposite end, usually would mean: entering edit mode / selecting a vertex / pressing shift S / cursor to selected / TAB / pivot point to 3D cursor / S / Y / LMB and optional, reverting pivot point to median. In point_scale, after selection is made, you only need S and LMB drag. This way you have automatic viewport choice on scale axis as well as pivot point and you avoid later correction of the object to fit the space it occupied. Right now, if you scaled a chair and it went through the floor and wall, you have to reposition it back to the floor, or go through a process of placing 3D cursor for pivot / changing transform pivot / transforming / changing pivot back. Or applying a lattice modifier.

If this operator reaches a mature state it will be an all-in-one, pretty fast and simple tool for such tasks. It also features the standard scale procedure if the user opts for it. Right now, the operator is in an alpha state and serves as a tech demonstrator.

RUN:

- a. point_scale button on NP tab (displayed in object mode)
- b. spacebar search - NP Point Scale
- c. keystroke if you assigned it

USAGE:

Make sure that you are in object mode. Select one or more objects that need to be scaled and activate the operator. A cage manipulator would appear around the selection. Choose the scaling direction (together with the position of the scale pivot) by clicking on one of the 14 handles. Drag the handle and snap to outside geometry*. The opposite point will become the pivot and remain static. Holding CTRL reverts the pivot point to central position and SHIFT keeps the aspect ratio while scaling no matter which handle was selected. Pressing TAB will enable a 'force apply' mode and will apply transformations to all of the selected objects when scaling begins. This is a pretty powerful feature if the scaling result turns out to be not as expected because of the rotations of the objects.

Being a fully OpenGL operator, **NAVIGATION** is possible at all time, using standard MMB drag for orbiting and SCROLL for zoom.

RMB and ESC key interrupt the operation.

NOTES:

Some problems occur when the operator is interrupted by a program warning (error). The context state before the operator is not restored afterwards. I tried to avoid that possibility but still, if an error arises you might find your pivot switched to cursor, snap target to vertex etc.

*Snap features of this operator still don't work as intended, limited by the capabilities of the inbuilt 'resize' operator.

SETTINGS:

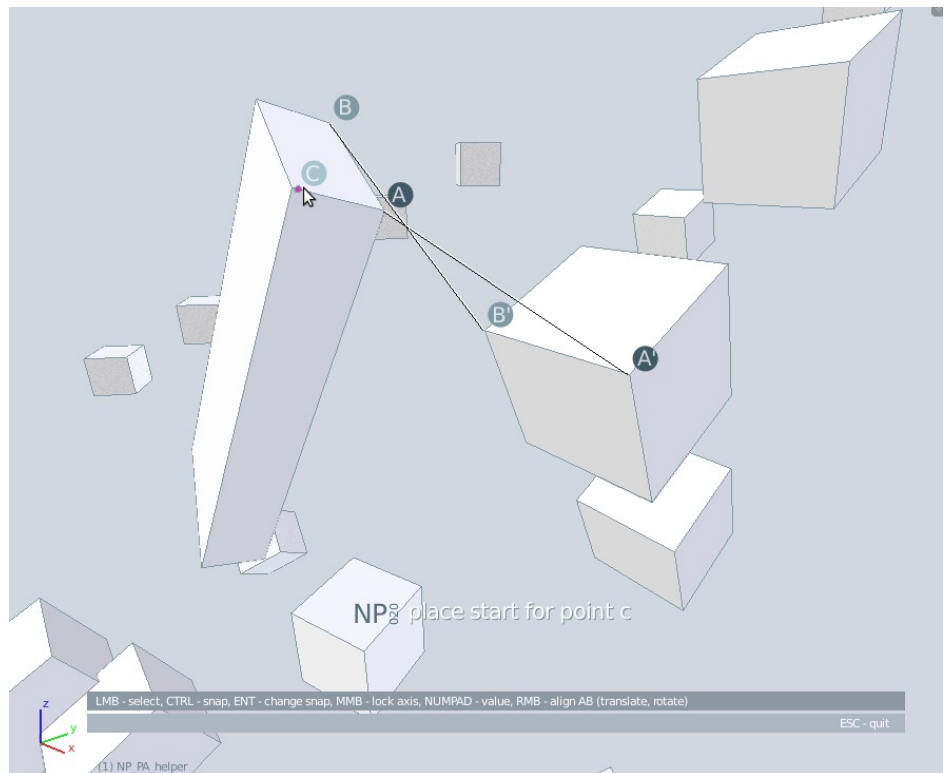
Common settings:

Being in the experimental phase, this operator is still not connected to main addon settings and does not feature adjustable graphical and functional aspects.

Individual settings:

No individual settings for this operator exist yet.

NP_point_align



IDEA:

As a standard part of the CAD tool-sets, align is starting to emerge more often in recent Blender addons. NP point_align seeks to complete the NP set and tries to give a special interactive flavor to this tool. The concept is simple - select three points on an object and connect them to the three corresponding points elsewhere in the scene. The operator will use these point pairs as guides for aligning selected objects trying it's best to make the pairs fit together. It emulates a standard 'align' command from the CAD world.

Unlike to what you might expect, the operator doesn't need the whole three pairs to be designated in order to make an alignment. The objects can be aligned using only the first pair (simple translation), first two pairs (translation and planar rotation around the first pair) or complete three pairs of points (translation, planar rotation, rotation around first axis).

RUN:

- a. point_align button on NP tab (displayed in object mode)
- b. spacebar search - NP Point Align
- c. keystroke if you assigned it

USAGE:

Make sure that you are in object mode. Select one or more objects that need to be aligned to a certain geometry in the scene. Activate the operator and when you see a circular marker with an A letter in it, select a point on an object that you wish to align (holding CTRL enables snapping). This will be your 'A start' point. To designate the target for the A point move your mouse and click to a point anywhere in the scene with the left mouse button (LMB), in relation to the 'take' point and the operator will display a connection line between 'A start' and A 'target' points. In the same way you can designate other two pairs of points (B and C) needed for the alignment. The objects can be aligned using only the first pair (simple translation), first two pairs (translation and planar rotation) or complete three pairs of points (translation, planar rotation, second planar rotation). The more pairs you designate the more precise the alignment will be. If there is at least one designated pair of points, the user can activate alignment at any time by pressing RMB.

Middle mouse button (MMB) enables axis constraint, numpad keys enable numerical input of distance, and ESC key interrupts the operation.

NOTES:

Being an operator still in experimental phase, **NAVIGATION** mode is not applied yet.

The UNDO command after the use of the operator requires cycling through certain phases of the command, so be careful to click undo enough times to delete all the helper vertexes that were created during the operation.

The point_align operator is especially vulnerable to this, since it uses six helper objects and the user needs to cycle undo through all of them. This is a limitation of the macro nature of the addon and hopefully will be upgraded in the future.

SETTINGS:

Common settings:

Being in the experimental phase, this operator is still not connected to main addon settings and does not feature adjustable graphical and functional aspects.

Individual settings:

No individual settings for this operator exist yet.

NP_shader_brush



IDEA:

Given the lack of standard material tools in Blender, the technical users might find this operator useful. It enables fast and intuitive transfer of materials between objects in 3D viewport. I often found myself making a lot of color variations on an object's facade or a certain design item. Transferring shaders from one object to another, painting certain parts of an object, swapping places of materials on an object or between them directly in the viewport, without looking down to keyboard or Blender menus. The operator main procedures are picking materials, adding them to selection and / or painting only particular faces of meshes. The operator is fully OpenGL so navigation around the scene is possible at any time.

RUN:

- a. shader_brush button on NP tab (displayed in object mode)
- b. spacebar search - NP Shader Brush
- c. keystroke if you assigned it

USAGE:

Make sure that you are in object mode. Activate the operator and you will see a paint bucket marker with the name of the active material next to it. This is the material you will paint with. If you wish to change the current material press CTRL to change the marker to pipette and click with LMB on a mesh face containing the new material. If the pick selection is made on an empty space, the bucket marker will change to 'empty' and the material name will change to None. Painting with this type of material clears the material slots from the painted objects or object's faces.

Painting is made with simple LMB click on an object. The painted object will change all of its material slots to the material that was in the bucket. If the user holds ALT while painting, the material will be assigned only to the face under the cursor and a new slot will be added to the object's material list. If the user wishes to paint more than one object at a time, a selection of objects has to be made before the activation of the operator. LMB click on one of the selected objects will paint all of the selected objects to specified material.

Being a fully OpenGL operator, **NAVIGATION** is possible at all time, using standard MMB drag for orbiting and SCROLL for zoom.

The user can continue painting and changing active materials as long as the operator is active. Action is interrupted by pressing RMB or ESC key.

NOTES:

Due to the limitations of the raycast engine that this operator uses for selecting, the shader brush does not work with non - mesh objects. Hopefully this can be upgraded in later versions.

Be careful when painting through objects that have bound-box or wire display. Although you can't see their surfaces, `shader_brush` will apply the paint on them if they stand between you and the desired object. In order to override this, move your point of view beyond the invisible surfaces.

Painting on objects with an array modifier gives a non - standard behavior when paint is applied to arrayed instances. Instead, apply paint to the original object that contains the modifier.

In case of this operator, the UNDO command behaves in a non - standard way. Undo function does not recognize the painting action for some reason and together with the whole paint session undoes the last operation before the painting. It is advisable to be careful when using undo with `shader_brush`.

SETTINGS:

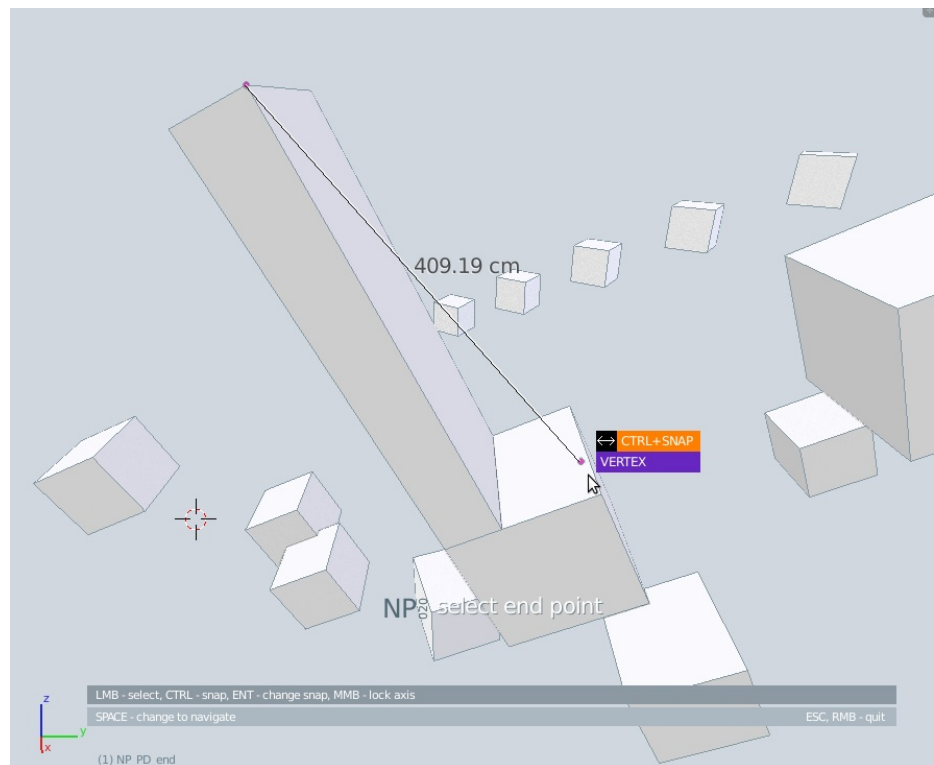
Common settings:

Being in the experimental phase, this operator is still not connected to main addon settings and does not feature adjustable graphical and functional aspects.

Individual settings:

No individual settings for this operator exist yet.

NP_point_distance



IDEA:

Probably the first totally complete and robust operator of the NP collection. Also, the operator with the highest download count from its individual page. It measures distance using snappable 'start' and 'end' points. It emulates the functionality of the standard 'distance' command in CAD applications and acts as a replacement for the unreliable ruler / protractor tool in Blender. User is now able to change snap target between vertex / edge / face while operating. Among newly added features are x / y / z distance components and optional stereo cage for better spatial comprehension.

RUN:

- point_distance button on NP tab (displayed in object mode)
- spacebar search - NP Point Distance
- keystroke if you assigned it

USAGE:

After activating the operator a graphical badge will appear in the 3D view letting the user know the command has started. Select a point anywhere in the scene (holding CTRL enables snapping). This will be your start point. Move your mouse in relation to the start point (again CTRL - snap). The addon will display a straight line and a numerical distance between your start and end points, scaled to the values designated in the addon settings. In case the addon setting *Step* is set to *continuous*, the second LMB click would re-initiate the operator and user can continue to make measurements.

If at any point you lose sight of the next point you want to snap to, you can press SPACE and go to **NAVIGATION** mode in which you can change the point of view using standard MMB drag for orbiting and SCROLL for zoom. When your next point is clearly in your field of view, you return to normal mode by pressing SPACE or LMB again.

Middle mouse button (MMB) enables axis constraint while **ENTER** key changes snap target from *vertex* to *edge* to *face*. In case ENTER is pressed again the cycle starts over from *vertex*. RMB and ESC keys interrupt the operation.

NOTES:

The UNDO command after the use of the operator requires cycling through certain phases of the command, so be careful to click undo enough times to delete all the helper vertexes that were created during the operation. This is a limitation of the macro nature of the addon and hopefully will be upgraded in the future.

SETTINGS:

Common settings:

Being an important and reliable command, this operator is - for now - affected only by it's individual settings.

Individual settings:

Scale: Distance multiplier for various measuring scenarios. Default is 100;

Suffix: Unit abbreviation after the numerical distance. Default is cm;

Step: Mode of command - whether the command operates in one-step or continuous fashion. Default is simple;

Mouse badge: Option to display a small cursor info. Default is ON;

Hold result: Option to have an extra step to display the last measured distance in the viewport. Default is OFF;

Value to header info: Option to display the last measured distance in the header. Default is ON;

Value to clipboard: Copy the last measured distance to clipboard for later reuse. Default is ON;

Custom colors: Default or custom colors for graphical elements. Default is Default;

Stereo cage: Display bounding box that contains the dimension, for better graphical overview. Default is ON;

XYZ lines: Display axial distance lines. Default is ON;

XYZ distances: Display axial distances. Default is ON;

XYZ backdrop: Display backdrop field for xyz distances. Default is OFF.

USING OPERATORS VIA SHORTCUTS, MENUS, PIES OR OTHER ADDONS

You don't have to use the addon in it's panel/icon form with all it's tools. You can use available operators of your choice. You can activate them by searching for them in spacebar menu, integrating them into your pie-menus, custom-made key shortcuts and custom menus or panels. Following is a table of operators that become available in your system when you activate the addon. Columns show the names of the operators for spacebar search, their identifiers for shortcut definitions in the User Preferences, custom menu and custom pie python designations.

	spacebar search	assigning shortcut keys	recommended key	python designation for custom menus and pies
	press spacebar / type any part of the	User Preferences / Input Object Mode / Add New		col.operator('object.np_020_point_move', icon='MAN_TRANS', text='point_move')
NP_float_poly	NP 020 Float Poly	object.np_020_float_poly	P	object.np_020_float_poly
NP_float_rectangle	NP 020 Float Rectangle	object.np_020_float_rectangle		object.np_020_float_rectangle
NP_float_box	NP 020 Float Box	object.np_020_float_box		object.np_020_float_box
NP_point_move	NP 020 Point Move	object.np_020_point_move	M	object.np_020_point_move
NP_point_copy	NP 020 Point Copy	object.np_020_point_copy	C	object.np_020_point_copy
NP_point_instance	NP 020 Point Instance	object.np_020_point_instance		object.np_020_point_instance
NP_point_array	NP 020 Point Array	object.np_020_point_array		object.np_020_point_array
NP_roto_move	NP 020 Roto Move	object.np_020_roto_move		object.np_020_roto_move
NP_point_scale	NP 020 Point Scale	object.np_020_point_scale	S	object.np_020_point_scale
NP_point_align	NP 020 Point Align	object.np_020_point_align	N	object.np_020_point_align
NP_shader_brush	NP 020 Shader Brush	object.np_020_shader_brush	B	object.np_020_shader_brush
NP_point_distance	NP 020 Point Distance	object.np_020_point_distance	T	object.np_020_point_distance

FEEDBACK AND UPGRADES

There are a lot of ideas, but not much time, knowledge or manpower. For this reason there are no roadmaps or wishlists in this document. There is a natural and obvious way these operators should evolve, although any suggestions are welcome as sometimes an obvious missing feature is overseen by one person.

Because of the lack of time (the operators are made in spare time), this package is not planned for continuous upgrade so that aspect should not be expected. However, all user feedback is archived, evaluated and eventually incorporated in the next version. Please have patience on this matter.

Aside from suggestions, the important thing to report are bugs. Some of these operators have seen heavy production environment and have proven themselves, but most of them are new or freshly upgraded and have not seen much real production testing.

BACKGROUND

Being someone who is immersed in real life architecture design in Belgrade's urban context, outputting a lot of real - world architectural solutions and variants on daily basis, i found myself searching for an capable app that can follow my pace of creative thinking, make fast variations with simple cubic architectural shapes and a robust renderer that can output fast, believable images for use with the everyday clients. If it was open-source and can run in Linux it would have been a great bonus. You can imagine what a huge discovery Blender was.

BUT! The user base and development quirkiness made this transition complicated. It is like somebody gave you a custom made streamlined supersonic jet for present, with robust aerodynamic wings, powerful turbofan engines, perfect aerodynamic line, put a steering wheel in the cockpit and started convincing you it's best usage is for racetrack competitions. And you have just landed from the skies flying dogfights with the various aircraft and you still have your helmet on and bruises on your hands from struggling with unoptimized commands and rugged airframes of various planes you have flown, remembering the good old days when you were showcased a similar super airplane, now in hands of the opposite side. And the only one who knows how to change the steering wheel to joystick and connect it to the flaps and engines is a quirky genial guy who is in love with track racing and made the plane dreaming of beating all the bad guys on the race track. Flying? Who wants to fly anyway? You need an airplane for that.

Waiting for the Blender development or user base response to engage in some enhancements turned out to be in vane and i had to change the approach. Although the time budget is tight i resorted to learning some Python basics and writing some extensions for my own use. At first they were quite basic and resolved simple problems but expanding them appeared to be quite easy and inviting.

This in return made me study a bit more of Python scripting and gradually producing a string of small Blender tools for everyday use. Most notably the basepoint tools (move, copy, etc) that we as a profession got used to in the business. To me it was obvious that the existing Blender technology is capable of satisfying all the architectural sketching needs but there was not much interest and no serious engagement in that direction from the existing architectural user base.

Although in complicated and non intuitive ways, with existing Blender tools, we already DO have the means to achieve the required level of precision in modeling, we just need to make some quick new tools with more intuitive procedures. The name of the game is *speed*.

PRIMITIVE CODE

Good will aside, i have to mention i am not a coder. I am in architecture design business, constantly working on projects, i have very little time to thoroughly work on these tools. The code is primitive, lacking superior functions that i see (and don't understand) in other scripts, the operators are clumsy and slow and too unreliable for production standards. The operators use the existing modal 'grab' and 'rotate' operators, constantly entering / exiting different edit modes, manipulating / hiding helper objects and what not. These tools should ideally be made in Blender's programming language and hard coded to Blender. The addons were made with stick and stone - grabbing stuff i saw was working in other addons without really knowing what or how it does. I have no coding skill upgrade planned and i am making this set primarily to help me in my own workflow and as an operator mockup, so please use with care.

NOT FOR PRODUCTION USE

This package was released as a gratitude gesture to the Blender community, not for professional or commercial gain. Professional level of code efficiency or robustness is not an imperative. At least for the time being, with one person is working on it.

There is also a slight difference in quality between the operators when they were managed individually and now when they are integrated into one system. This approach has it's advantages and should eventually yield easier evolution of the package but in this prototype phase there is a lot of underlying mess as a result of complex transitions. A couple of operators were recently introduced and some of the existing extracted and separated for better set consistency. There is a lot of structure recycling, code exchanging, variable renaming, settings and graphics syncing and some of functionalities may have been lost or corrupted.

There was not enough time to check all the scenarios of usage for all the operators, so i will be testing them in production in parallel with you. If someone tries this collection, please report bugs to the BA thread.

99% OBJECT MODE

I am really not a fan of edit mode. Manual editing of objects, at least for simple cubic shapes and meddling with internal structure as it is, for technical sketching should be brought down to minimum. The workflow for simple shapes and modification such as scaling must be faster, more precise and more automated. The needed operations must be handled and controlled from outside and with just a few logical geometry inputs. You will notice that most, if not all NP operators work in object mode, or change the context to it for functioning.

DISREGARDING 3D CURSOR

I know many of you find Blender's 3D cursor an original and helpful feature. I even think some of you consider it a cutting edge feature enabling precise manipulation. However, coming from 15 years of digital architectural design i have somewhat different view upon the matter.

3D cursor is in fact something user should never meet with. I know it is an original feature, but in my years of dealing with computer graphics in architecture modeling and design i have never failed to complete my tasks without it. Not only that, the fastest and most intuitive 3D apps i know today, in terms of architectural visualizations finish the goals i'm after in quarter of time compared to Blender, with precision and without such a feature. It seems 3D cursor is a relic feature inherited from the first days of Blender when the industry was still struggling with pivot points and origin designation. It is outdated today. Apps have resolved the problems of precise user input for pivots and origins in a fluent and intuitive way a long time ago. The way it is now in Blender, it is also not helping in terms of fast variant modification of structures - fast moving, rotating, scaling with precision. Too many menus, too many clicks, too many look diverting, getting creative thoughts lost. Maybe this sounds harsh, but outputting up to 200 trial renders per design, i count my time in seconds, and the sooner 3D cursor is replaced with more intuitive automated procedures, the sooner will Blender take the place it deserves.

I even got some comments like 'relax, it will come to you when you do this or that'. I am a Blender user for six years now, other app user for fifteen, i know what i am talking about. It seems to me like a lot of people's opinions on this is based of not really venturing much outside of Blender. You can be sure that architects do exactly that. If there is one thing you can rely on architects is that we have seen and used a lot of graphical apps of various purposes and continue to do daily. Although we are an introvert group of users, BF should use our experience in

dynamic and versatile work environments. For the very same reason of fast workflow and ferocious effectiveness criteria, you won't find a lot of architects using Blender. I was forced here by somewhat strange and strict personal principles and i was nagging in confusion and frustration the whole way.

The operators i made are often using the functionality of 3D cursor and you can notice it in some operations. However, you will never be in a situation requiring to manually place the 3D cursor for any purpose or even think about it. If i knew how to do it, i would hide it altogether. This is meant to show that 3D cursor is an design surplus and is slowing things down. It's functionality needs to be intuitively integrated into the commands.

OLDER BLENDER VERSION

Operators were produced in, tested by, and intended for Blender version 2.75a which is my work surroundings.

The reason i am using an older version of Blender and did not upgrade is because of the nature of work i do, complexity of modifications i made to original release and Blender features that were introduced in the meantime. For the work i do, 2.75 features satisfy all the needs in my daily work. New versions haven't brought any dramatic enhancements that would benefit my workflow and justify addon and shortcut migration and occasional re-coding. The 2.8 version is a possible candidate for migration as it will introduce some substantially new tech (layer system, widgets, asset management, viewport and api enhancements etc). Although, it could happen that i will wait for the new tech to settle in first, so be careful before opting for this collection as a post 2.79 tool.

NEW OPERATOR MOCK-UPS

The bundled operators are not meant to be real operative tools. A real, efficient, robust tools would have to be coded in C and implemented directly to Blender's core. These are best considered as an illustrative show of proposed functionalities to the users and developers. These addons are not 100 % reliable, often lack undo and are slow in comparison to C commands. In short, consider this nothing more than new operator testing ground. Also, a good way to illustrate my proposals at sites such as Blender Artists and Right-Click Select.

USING AVAILABLE BLENDER TECHNOLOGY

Addon is made using available Blender tech, integrating existing Blender's operators in alternative procedures.

For example, snap functions are not yet available in python (eg: 'detect the nearest vertex / edge / midpoint') and i was forced to use available modal operators available in Blender, that have snap. Each time you are asked to select something with snap, it actually invokes the standard 'grab' function, moving a one-vertex mesh in the background, with snap enabled and graphics overlay following it, in order to communicate with user. This operator, in return, does not react to any user input other than few keys already assigned to the operator in Blender, and no other keys can interact in that moment. An addon maker can only add subsequent actions over modified result upon the pressing of that key.

That is why you have SPACEBAR for navigate mode or ENTER for changing snap - those are the keys that grab actually reacts to. The keypress stops the 'grab' operator, the macro reads that, changes the operator settings and starts the 'grab' function again, trying to reconstruct the context it was in when it got stopped. If my code was any good, you as a user won't notice that.

Not until 'detect snap', 'select object under the mouse' and a couple of other functions are made available in python will these operators become more efficient. So far it's just improvisation and 'jazz' homemaking.

CONTINUITY

Sorry for often changing the names, shortcuts and colors - the whole system is still in rethinking and baking, through trial and production. The system identity will probably change further in the future.

NAMING

NP is a generic abbreviation - N for 'new' or 'novi' in Serbian, P for 'prototype' or 'proba' in the same language. I was supposed to change it to just 'N' when it reaches maturity, but since i am never satisfied it is probably prescribed to NP. I was waiting to come up with a better name but it kind of stuck around and it seems kind of cool now. I have some ideas for a new name but i have no plans to rush with the change.

Station was the one term that had all the essential characteristics i was going for. It is a one word / technical nature / combining platform - like term. The other options were 'package', 'set', 'collection', 'terminal', etc.

Float in float_rectangle and other shape creation operators is referring to their drawing plain constraining feature. Once the user designates the 3D plane to draw the shape on (sometimes refereed to as the reference plane), it is bound to it. For purpose of precision and speed, the whole shape or 3D basis will be set exclusively on this plane, the user point designations will 'float' on it.

Point in point_copy and other modification operators refers to precision and snapping capabilities. Although i will try to expand the snapping to edge and face modes, it will probably keep this designation for sake of continuity.

The **020** comes from the fact the biggest number in the individual operator's life was 018, namely in NP_point_distance. In order to avoid numbering and operator capability confusion i resorted to starting the NP_station with the number 020. This should show that any operator in this package is more advanced than the previous ones with smaller version numbers. For example, the previous version for NP_shader_brush was 002 so the new 020 designation should avoid any confusion as to which version is more advanced.

Also, in order to emphasize the transition to the package system, the position of the version number has changed from NP_point_distance_000 to NP_000_point_distance. This shows the package nature of the NP and it will change the version number in all featured operators with the next release.

Because of the need for testing releases i will probably resort to odd / even version numbering. In consequence, every odd numbered version (e.g. 021) will be for testing purposes and even numbered for usage (e.g. 022). This release (020) will be a long term one, i need to cool off a little bit.

GRAPHICS

The ideal graphic system is still being looked for. You can detect a transition in graphics between the individual operators and this package. I will continue to experiment with graphics further so please have understanding for this. The graphic system is divided into three main parts:

Geometry / Widget / Numerics

Icon / Mouse badge

Instructions

Each one of these has a separate graphics logic, has to fit nicely with the whole graphic system and is likely to become subject of further experimenting. The inspiration is drawn from operator's use in production, graphic design in general, my experience with other software as well as other Blender addons. New versions will probably see slight changes in the overall graphics as well as new color scheme templates.

RECOMMENDED SHORTCUTS

The best way to use the NP operators is via keyboard shortcuts. This has been confirmed many times in my daily work flow. It is faster and does not require additional interface space in the tool shelf. When the users get familiar with the interface and procedures, there will probably be an option to turn the tool-shelf tab off completely. The following is a list of recommended keyboard keys or, to be more precise the ones i use. This keymap makes most sense in architectural design work flow, which does not require commands used in sculpting, animation, game design or video editing. Right now, there are not much NP's i transferred to keystrokes in my personal work flow because of their experimental nature. That should change with the operators becoming more capable and robust.

Procedure: File / User Preferences / Input tab / 3D View / Object Mode / + Add New / change 'none' to item in Operator Table, third column / 'A' to key of your choice

Recommended keys:

P - float_poly (freed by the fact of not using the game engine and it's 'P' command)

M - point_move (freed by the usage of Layer manager addon for transferring between layers)

C - point_copy (freed by the fact of not using the circle selection)

S - point_scale (replacing the standard 'resize' operator, which is being incorporated in the new command)*

N - point_align (freed by redirecting properties panel to 'X' for more convenient finger distribution and 'A' and 'L' being reserved for possible future operators)

B - shader_brush (freed by reassigning 'border select' to LMB tweak - much faster and convenient)

T - point_distance (as in 'tape measuring', freed by redirecting tool panel to 'Z' for more convenient finger distribution and 'D' being reserved for possible dimensioning operators)

*The point_scale command is not yet ready for this transition as it fails in some circumstances, with me having to revert to original 'resize' operator.

CONTRIBUTORS

In producing NP station i owe great thanks to:

mont29, together with **meta - androcto** and **Ryker** - for help in overcoming macro - bundling sound barrier. A problem with packaging macros stopped me in my tracks when trying to release a unified addon. I was struggling with finding a way to join the NP's into single package because of undocumented nature of macros, which most of the operators are. Those guys gave me tips on overcoming this hurdle, with **mont29** tackling the problem with a concrete solution. If we go hypersonic with this it will be mostly because of them;

mano-wii, **meta-androcto**, **nBurn**, **matali23** and **pyc0d3r** for testing and support on developer.org as well as extensive help in making these tools usable and more efficient;

CoDEmanX, **lukas_t**, **matali23** - for modal operator integration to NP's and basic macro routines which are in the core of almost all operators. Huge thanks!

lijenstina - for PEP8 conversion and code support on float_poly operator;

Ton Roosendaal - aside from Blender in general - for motivational support and approach advice;

hackerman-, **ideasman42** & **the developers** from the #blendercoders - for logistics support and technical tips;

batFINGER - for support on Online python reference and certain code details;

Pixivore and **pitiwazou** from the 'Carver' addon project - for inspiration on 3D view interface design details;

Vitor Balbio from the 'Bool Tool' project - for the inspiration on tool shelf button design;

Antonioya - for addon packaging and panel creation procedures as well as huge CAD overall effort;

Pixivore - for code solutions on rotating bgl shapes according to normal vector under the mouse;

Elegant Artz and **mano-wii** - for making the tutorials and sharing tips that laid the foundations of the on-screen geometry drawing as well as the inspiration for the original float_box operator;

Martin Buerbaum, **TNae**, **Benjamin Lauritzen**, **Alessandro Sala** for making the 2.60 'Measure panel' addon, that influenced the point_distance operator a lot;

mauge and **mano-wii** - for keeping the wind in the CAD community's sales with 'Mechanical Blender' and 'CAD snap utilities' projects;

Fweeb - for keeping an ear to CAD community members' outcries on Right-Click Select;

Numerous BA members with their support and suggestions that often made it to the next release - **JuhaW**, **matali**, **Spirou4D**, **ejnaren**, **vkliidu**, **Alain**, **bluecd**, **mkbreurer**, **bliblubli**, **dphantom**, **nBurn**, **1D_inc**, **mbbmbbmm**, **tynka topi**, **eppo**, **floo**, **kakachiex2**, **danilius**, **raziel_henzo**, **burnin**, **Craig Jones**, **urkokul**, **erdinc**, **zcaalock**, **bekic.bojan**, **Albertfox**, **AlmaTalp**, **saltorio**, **whitek**, **bls**, **wanilson_m**, **Asmund**, **poly2poly2** and many others;

Professional grade software developers - for their devotion to excellence and inspiration.

JOINING THE EFFORT

If you like this addon, have coding skills and want to help the Blender CAD community by enhancing these tools, you have an option to join me in voluntary development of the NP set, under it's name (or other that i come up with) and, for the reasons of keeping the standards i envisioned, under my strict control. This probably does not sound like much but i could surely use help, there are a lot of ideas on the paper. What would be your motifs? I really could not say, probably total irrationality.

If you don't like that option, you can freely hijack the addon, make your own version and make a new life and evolution for it. The only thing that you should do is change the name and start a separate Blender Artists thread to avoid confusion. Absolutely no credit required, this addon is free in all aspects.

