Spacedraw User Manual

Version 1.3

Welcome Spacedraw's user manual!

This documentation completely covers Spacedraw's functions, and can be used both

- as guide to learning the program, reading it in sequence,
- and as reference.

It precisely describes the operation of the program, but assumes basic knowledge in 3d computer graphics.

Conventions in this manual

- technical terms, both ones common from other 3d graphics software and ones specific to Spacedraw, are displayed in *italics;* where they are explained in ***bold***; Terms not explained here should be to find in Wikipedia and are used as usual
- menu-items are displayed green, in the same font in which they appear in the program
- preferences are displayed in blue serif
- procedures are underlined, the steps are numbered

User interface

Spacedraw is meant for a variety of tasks:

- freely drawing lines in 3d space
- CAD-like construction
- polygonal modeling
- patch-modeling
- texturing
- painting on 3d-objects,

for which varying user interfaces are appropriate. For desktop computers, 3d-modeling applications, CAD systems and digital painting software have established several concepts.

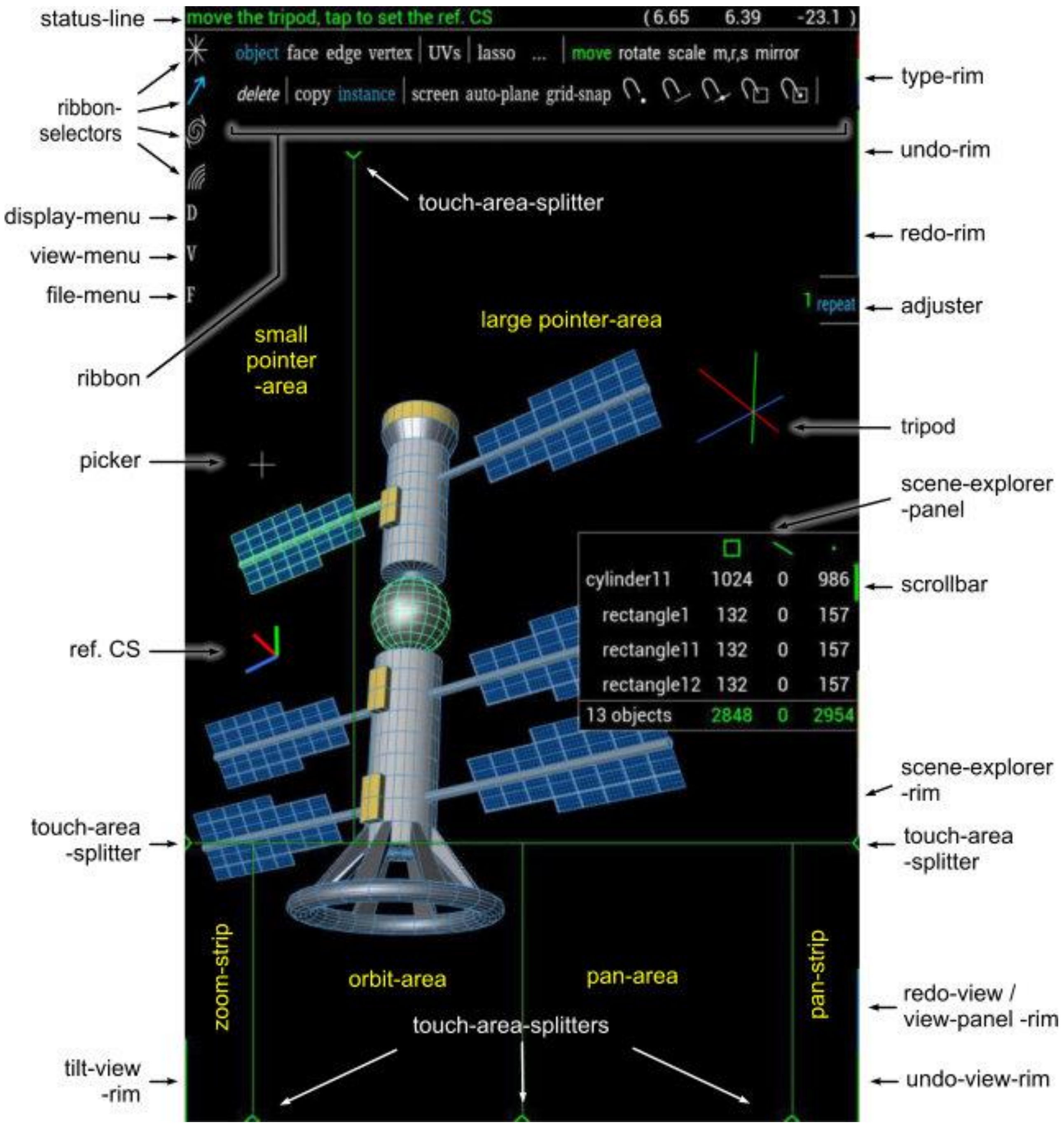
For one thing, Spacedraw's interface builds on these concepts and conventions; for another, it is specifically contrived for tablet computers and larger smartphones, it

- takes great advance of multi-touch input and motion sensors,
- copes with the lack of mouse and keyboard
- deals with the limited screen size of about 5'' to 10'', or even down to 3.2'',

- is adaptable to the screen-size, -type and -resolution of any Android device

The interface is designed to be fast to learn for novices, and fast to relearn for experts. However, Spacedraw is a large program and introduces a multitude of novel concepts, so learning it may take some effort.

Interface elements



The Spacedraw window consists of one large 3d-viewpot and various overlaid control elements:

- the ***status-line*** at the top
  - in the left part, hints for using the active tool, prompts for inputs, or messages are shown
  - in the right part, the coordinates of the tripod, or info about the selection are displayed
- the ***ribbon*** below the status-line provides access to the tools, commands and settings

- the ***ribbon-selectors*** and ***menus***, displayed as symbols resp. letters along the top left border, for selecting the *ribbon,* resp. accessing additional *menus* containing commands and settings
- ***rims***, parts of the screen borders marked by colored lines, where sliding into the screen has specific effects
- ***adjusters*** for numeric settings, stacked along the right border when needed
- the ***text-input***, shown when needed together with a keyboard
- ***touch-area dividers*** mark the boundaries between the touch-areas and can be dragged to shift the boundaries
- ***panels*** docked to the right screen's border for specific tasks; only one panel can be open at a time

The screen is divided in ***touch-areas***, where touching has different effects:

- the lower part of the screen, called ***view-zone***, is for changing the view
- the upper part, called ***tool-zone***, is for operating the active tool, mostly by moving the *2d-* or *tripod,* or tapping geometry *elements* to select them

By using multiple fingers, the view can be changed while operating a tool.

Tools, commands and settings

Spacedraw's functionality is exposed via ***tools***, ***commands*** and ***settings***, organized in four ***ribbons*** and three ***menus***.

- there is always one *ribbon* displayed along the top of the screen; it is selected by tapping one of the four symbols on the top left border, the symbol of the shown *ribbon* is highlighted
- the *tools* are used by touching the screen within the *tool-zone.* They are displayed in upright sans-serif font in the *ribbons*; there always is one ***active tool***, indicated in green
- the *commands* are immediately executed on the selection when tapping them. They are displayed in *oblique sans-serif* font
- the settings influence how the tools and commands work; some settings belong to specific tools and are only shown when the respective tool is active. There are two types of settings:
    - "on/off"-settings are displayed in serif font in the ribbons or menus and toggled by tapping them; enabled options are indicated in blue
    - numeric settings are made available via *adjusters*, displayed along the right screen's border; to change a value either
        - touch the screen on the adjuster's name and slide up/down
        - tap the adjuster's name; the text-input appears; type the value and tap the Enter-key on the keyboard (it may read "Next" or "Go")

      to reset a value to the default, touch the screen on the adjuster's name and slide rightwards

Panels

For specific tasks, ***panels*** are shown along the screen's right border, containing control-elements and lists:

- *scene-explorer:* for inspecting and selecting objects
- *file-browser:* for loading and saving files
- *material-panel:* for editing and organizing materials
- *brush-panel:* for selecting brushes for painting
- *preferences-panel:* for changing advanced settings
- *history-list:* for accessing the construction-history

The panels can freely be resized, and lists can be zoomed using multi-touch:

to resize a panel, place one finger within the panel and another outside of it, then slide both fingers left/right to shift the left border, and up/down do shift the bottom or top border; start sliding above the panel's center to shift the top border, in the lower half to shift the bottom border

The lists can be scrolled by *dragging* or *flinging* like in any Android-application, or by dragging the scrollbar like on a desktop computer if the list is longer; in this case, the scrollbar is displayed bolder, and you can drag it by touching the screen close left besides it.

to zoom a list, place two fingers within it and *pinch*

Text-input

For typing coordinates and names of objects, material or files, a text-input box and the default Android keyboard are shown. The text box is used as follows:

to scroll the text if it doesn't fit into the box, drag or fling sideways within the box

to move the cursor, tap the desired position, or slide sideways beneath the box somewhere *not* directly below the cursor

to select text, either

- move the cursor to one end of the selection, then touch the screen below the cursor and slide sideways
- to select a word, double-tap it

to cut the selected text to the clipboard, touch the screen directly on the cursor and drag downwards

to copy the selected text to the clipboard, touch the screen directly on the cursor and drag upwards

to paste the text from the clipboard, touch the screen under the cursor and drag upwards

to switch the keyboard, touch and hold within the box; a list of the keyboards installed on your device is shown

When the text-box appears, it is possibly filled with the current coordinates resp. the current name, but the text is selected, and when you start typing it is deleted; to modify the text instead, first tap in the box at the desired insertion point.

Hiding the Interface

To get an unobstructed view (e.g. to take a screenshot), all interface elements can be hidden by tapping the menu-button or via the F-menu

Customizing the interface

The sizes of all screen elements, the effects of touch input and of tilting the device can be adjusted. Also, you can freely choose the colors of the user interface and the geometry elements in the viewport; a "home grid" and "sky" can be displayed to aid the sense of direction. Preferences can be saved and loaded via simple text-files.

The preferences-panel

The settings are managed with the *preferences-panel;* NEW it can be shown or hidden by sliding leftwards into the screen over the preferences-rim, the yellow line at the right screen's border, at any time.

There are 3 types of items:

- numeric values: NEW slide horizontally on the entry (you can use the full width of the panel) to adjust the value, or tap the value to type
- colors: tap the color-box to show the color-picker and adjust the value, see Selecting colors
- on/off values; tap the checkbox to enable/disable the option

All changes are applied immediately; to see the effect, make sure affected elements are visible.

"Reset" sets the selected entry to the factory setting; "Reset all" restores all entries in the preferences-panel, and also the touch-areas and panel-sizes.

"OK" (or sliding the rim again) retains the changes and closes the *preferences-panel*, "Cancel" discard them.

The preferences can be saved and loaded via plain text files as simple colon-separated key-value pairs:

- Tap "Save" to save the current preference to a file on your device; it gets the ".spse" extension and can be edited with any text-editor
- Tap "Load" to load a preferences-file; entries defined in the file are loaded, the others are not changed
- To load only certain settings, e.g. colors, first delete the other entries in the file with a text-editor

Adapting and adjusting the interface

For convenient operation, it is essential to adjust the interface to your device, and to your personal preferences and working style. The most important settings are described in the following.

The "Sizes" settings determine the dimensions of the screen elements:

- NEW All sizes changes all size values (incl. Selection radius, Snap radius and Grid distance) at one go
- Tap size determines the minimal dimensions of the "tappable" areas, e.g. for menu-items and buttons
- Text size sets the font- and symbol-size for menu-items, the status line, and in panels
- Pointer grip length and Pointer grip width set the grip-area below the *2d*- and *tripod*, used for drawing freehand-lines and painting; when the entries are selected in the preferences-panel, these areas are outlined in the viewport
- tripod size, picker size and Pointer line width set the pointer-sizes
- Vertex size, Selected vertex size, Active vertex size and Preview vertex size set the dot-size for the vertices

    - if Antialiasing is disabled, the values are rounded to integers
- the Edge width entries set the line-width for edges and lines and curves
    - Some devices don’t support widths greater than 1, and devices that don’t support antialiasing **must have** Antialiasing disabled for the line-width to work
    - if Antialiasing is disabled, the values are rounded to integers

The “Input” settings determine the effects on touching and tilting:

- Drag speed sensitivity, together with Normal drag speed, sets the “***touch-acceleration***” for pointer-movement and view-change;
    - a sensitivity of 0 means no acceleration, Normal drag speed has no impact
    - if the sensitivity is greater 0, motions faster than a certain speed, set with Normal drag speed, are accelerated, while slower motions are decelerated
- Touch move factor can make the *2d-* and *tripod,* and the view, move faster or slower than you slide
- Tilt rotate factor can make the view rotate faster or slower than you rotate the device

The “Viewport display” settings determine the rendering of the geometry and can impact the performance:

- curve approximation steps sets the number of divisions for splines and arcs
- X-ray opacity determines the faces’ transparency when x-ray is enabled
- Antialiasing makes edges appear smoother; not supported on devices with "Adreno" GPU (most newer devices)
- Real-time transparency sort provides a permanently correct transparency representation while changing the view and while moving elements, but heavily impacts the performance; if disabled, transparent surfaces lying on top of each other are displayed correctly only after releasing the touch
    - when changing the view by tilting, tap within the *view-zone* to update the transparency

Starting, saving and loading scenes and models

Starting, running and exiting Spacedraw

Once you've installed Spacedraw, you can start it like any other Android application from the apps-list, or by opening a .spa-, .spah- or .obj-file (see below) from a file browser.

There always runs only one instance of Spacedraw on a device; if you open a file with Spacedraw and it is already running, it is brought to front, and asks you to save unsaved changes to the currently loaded scene if necessary.

To send Spacedraw to the background and return to the previous application, use the back-key; Spacedraw will be kept in memory and brought to front again if you tap its icon in the apps-list or task-switcher.

To exit Spacedraw and free the used memory, invoke *exit* from the F-menu; the current state is saved to the disk and restored when you restart Spacedraw.

- when a large scene is loaded, this can take some time

Using files

The current state of a scene can be saved to a file, with the paid version of Spacedraw optionally including the complete construction history. It includes all geometry in the scene, the material definitions, and the current selection, settings and active tool, so you can continue working on a project exactly where you left off when saving. In the free version of Spacedraw, only scenes with no more than 1000 vertices can be saved.

These ***scene-files*** are saved with a .spa-extension, or .spah-extension if construction history is included.

Models can be saved and loaded in the Wavefront .obj format that can be read and written by nearly every 3d-graphics program; texture coordinates, *normal* definitions, *smoothing groups* and materials are imported and exported. In the free version of Spacedraw, .obj-files can only be exported from scenes with no more than 1000 vertices.

- If the .obj file contains *normals*, these are used, if it contains *smoothing groups* but no *normals*, the *normals* are assigned according to the *smoothing groups*

All commands for managing scenes and loading and saving files are found in the F-menu.

to start a new scene, invoke *new scene*; an empty scene containing only one light is created, and all settings are reset

to open a scene, or load a .obj model in a new scene, invoke *open*, browse to the .spa-, .spah- or .obj-file and tap it; loading large files can take some time

to merge a saved scene with the current one, or add an .obj model to the current scene, invoke *merge*

- scene-files with history can't be merged; save a scene without history if it shall be merged to another one
- all objects form a .spa-file, including lights, are inserted as they are defined in that file; all materials defined in the .spa file except the *standard-material* are also added, elements with the *standard-material* assigned get the *standard-material* of the current scene
- .obj models are inserted with their coordinate-origin at the *tripod* position; the imported objects are selected automatically, so you can use the transformation tools to place, scale and rotate the models as desired immediately

to save the current state of the scene, invoke *save* or *save as*

- if the scene was not previously saved, or if *save as* is used, the file-browser and text-input are shown; navigate to the folder where the scene shall be saved and type the name, then use the enter-key or "OK" button to approve
    - to include the undo-information, enable "include history"
    - the .spa- or .spah-extension is automatically appended to the typed filename
    - the generated *paint-texture-images* are saved in the same folder as the *scene-file;* to avoid clutter on your device's file-system, save the scene to a new folder if you plan painting (use *new folder* from the ribbon)
- if you use *save* and the scene was previously saved as .spa or .spah file, that file will be overwritten
- saving big scenes with a large history can take some time; when the file is saved successfully, a message is shown in the *status-line*

to save models in the .obj-format, invoke *export as .obj*; the file-browser and text-input are shown; navigate to the folder where the file shall be saved and type the name, then use the enter-key or "OK" button to approve. The material definitions are stored in an equally named .mtl-file in the same folder, and all used texture image files are also copied to that folder if necessary.

- to save all objects in the scene, disable "only selected"
- to save specific objects, select them beforehand, and enable "only selected"
    - if only one object is selected, it is placed at the coordinate-origin in the .obj-file

The file-browser

For opening and saving files, the file-browser is displayed in a panel and lets you browse your device's local file system

- see Panels for resizing the panel, and scrolling and zooming the file list

- the full path of the shown folder is displayed in the header
- the subfolders are displayed in blue, sorted before or behind the files
- when zoomed in enough, the size and the time modified is shown for each file

to navigate to the parent folder, tap the header

to open a subfolder, tap it

to navigate back, use the *back* command from the ribbon; the back-key closes the file-browser

to change the sorting, use the options from the ribbon

to create a new folder invoke *new folder* in the ribbon; the text-input appears; type the name and approve with the *enter-key*

to delete a file or empty folder, slide rightwards on it; a confirmation dialog appears

- non-empty folders can't be deleted

Undo and history

In Spacedraw, every construction step can be undone, even after exiting and restarting the application, and with the paid version of Spacedraw, the construction-history of a scene can be saved and loaded in a file. The steps can be displayed in a list, and you can directly load any state from the list.

In principle, any number of steps can be saved; they are first saved to the RAM, and the older ones then moved to the disk (the internal storage); when your device is out of RAM or disk-space, errors will occur. You can enable and disable the recording of actions at any time via enable undos in the F-menu.

Selections are treated as individual steps; the view is also saved for every state and can be loaded together with the state; therefor, enable load view with undo in the preferences.

to undo the last action, slide into the screen from right over the *undo-rim*, the green line at the top right border, or tap close besides the line

to redo an action, slide into the screen from right over the *redo-rim,* the blue line below the green one, or tap close besides it

to show the *history-list*, slide leftwards into the screen over the *undo-rim* further, until the list is displayed; the currently loaded state is indicated in green; you can now scroll, zoom or resize the list, see Panels

to load a state from the *history-list*, tap it; you can "scrub" through the history by touching the active (green) state, and then dragging slowly up/down; if a state takes longer to load than you linger over it, it is skipped

- when you perform any edit-operation with a state from the history loaded, the subsequent history is cleared; Spacedraw therefore displays a warning message in this case and opens the *history-list* - to cancel the operation and load the last history state again, tap it in the list; to continue, tap within the *tool-zone*

to clear the history, invoke *clear undos* from the F-menu; all undo information, and the historic painting texture image-files are deleted

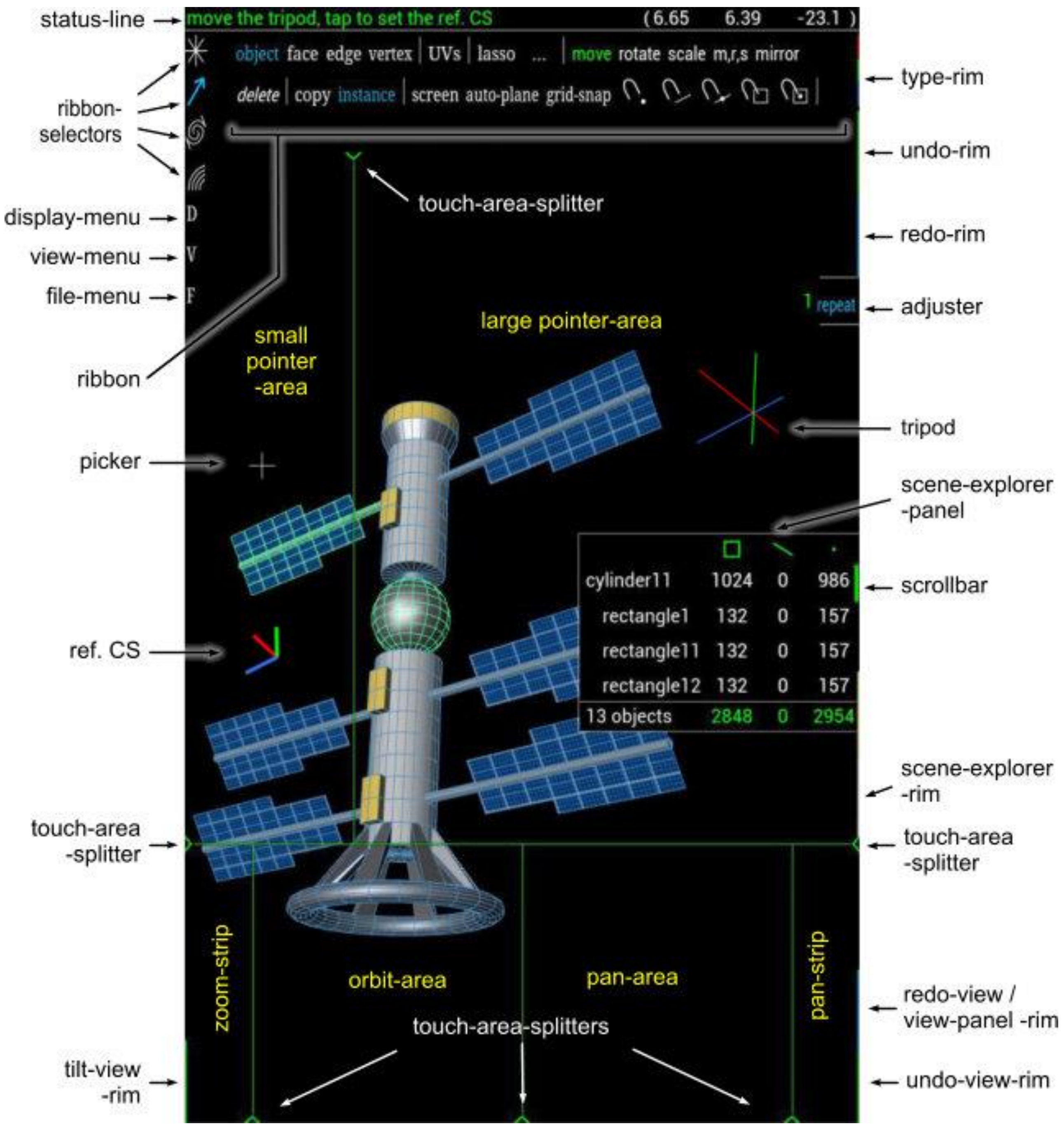
- if the scene is saved to a .spah-file, that file is not changed immediately, but when you save the next time, only the states after invoking *clear undos* are included

Changing the view

By default, Spacedraw uses a perspective projection. NEW Via the V-menu, you can switch to an *orthographic projection*.

The view can be changed with single- or multi-touch, and by tilting the device via motion-sensors. Using different fingers, or by simply moving the device, you can change the view while moving the pointer, drawing or modeling. That way you get a 3d-impression of the model, and how you're changing it in real time.

The lower part of the screen, called *view-zone,* is destined for the view-operations; its top boundary can be shifted by dragging the *touch-area-splitter* on the screen's left or right border. For single-touch, there are distinct *touch-areas* for translating, rotating, and zooming. Using two fingers, you can *pan* and *zoom,* or *pan* and *dolly* concurrently. Simultaneously, you can *orbit* or *look around* by rotating the device.



move the tripod, tap to set the ref. CS (6.65 6.39 -23.1)

	□	⟋	·
cylinder11	1024	0	986
rectangle1	132	0	157
rectangle11	132	0	157
rectangle12	132	0	157
13 objects	2848	0	2954

Most options and commands are available from the V-menu, the others in the "Input" section of the *preference-panel*.

Rotation

- You can *orbit* the view around the *view-center* (determined by the zoom-level) or around the *tripod*, or *look around* from the viewpoint (to choose with orbit / look around and orbit: center / pointer).
- To rotate the view either
  - slide within the *orbit-area*
    - You can snap to *face- edge-* or NEW exactly *isometric* views by activating angle snap
    - The Touch orbit velocity, together with the *touch-acceleration*, determines the impact, see here
  - activate *tilt-view* by sliding into the screen from left over the *tilt-view-rim*, marked by a green line, then tilt your device
    - with the Tilt rotate factor the effect can be amplified or mitigated;
    - if effects on tilting left-right / up-down are wrong (reversed), enable Switch gyroscope orientation

Translation

- For single-touch, there are two modes, using distinct touch-areas for translating (to switch with pan / walk)
  - parallel to the *screen-plane* (*pan*) / back and forth (*dolly*)
  - in the ground-plane (*walk*) / up and down

  The Touch move factor, together with the *touch-acceleration*, determines the impact
- Using two fingers, the average motion determines the movement parallel to the *screen-plane*, and *pinching* either zooms (in orbit-mode), or moves back and forth (in look-around-mode)
  - The Touch move factor determines the impact of the average motion, the Touch pinch velocity the impact of *pinching*

The movement speed depends on the zoom-level.

Zoom

Zooming magnifies the view by moving the virtual camera towards a fixed point, the *view-center*. It affects orbiting around the *view-center* and also changes the speed of translating operations.

- Slide up/down within the *zoom-strip*. In look around-mode, zooming leaves the camera fixed and moves the view-center, thus having no immediately apparent effect
- If orbit-mode is enabled, *pinch* with two fingers within the view-zone (the average motion simultaneously translates the view)

While zooming, the zoom-level is displayed in the *status-line*.

to approach a specific point, first invoke *look at pointer* from the V-menu, then zoom in

NEW Perspective distortion

For perspective projection, always a certain *distortion* occurs; the larger the *angle of view*, the more. It can be adjusted by holding one finger on the *zoom-strip* and moving another one on the pan-strip (or the other way round). This changes the *angle of view* and zooms at the same time to compensate the effect, such that the *field of view* stays approximately the same .

Specific views

to reset all view settings, invoke *default view* to move the view-center to the tripod-position, invoke *look at pointer*

NEW to frame the selection / all, tap the view-zone with 2 / 3 fingers (lights are taken into account if they are shown)

NEW to switch to the standard *isometric* / top / front / left / ... views, use the buttons in the *view-panel*

To comprehend the construction process, you can enable load view with undo in the preferences.

NEW Undo / redo view changes

This works analogously to normal undo; use the *undo-view-rim*, the green line at the bottom right border. There's also a *view-history-list* similar to the *history-list*, just "upside down" (accessed by sliding over the *undo-view-rim* farther). If redoing the view is possible, there is a blue line above the green one for that; else, there is a white one.

NEW Saving and loading views

The ***view-panel*** lets you save and load custom views (to delete a view, swipe right on it), or switch to standard views. To open the panel, slide into the screen over the white/blue line above the green *undo-view-rim* until it appears.

Moving and snapping the *tripod*

For drawing, placing and dimensioning primitives, and for transforming elements, a ***tripod*** is used, shown as an "*axis-tripod*". It can be moved freely in space, or snapped to the coordinate-grid, vertices, lines or surfaces. It also has an orientation that is used for transformations.

While you move the *tripod,* and when it snaps to positions, its apparent size changes with its distance from the viewpoint like an ordinary object, so you can estimate its movement; when you release the touch, it returns to its initial size.

Moving

The upper part of the screen, called *tool-zone,* is used for moving the *tripod*.

There are various methods to move the *tripod*, using single- or multi-touch:

- in one coordinate-plane / the perpendicular axis, by sliding in the large pointer-area / small pointer-area, possibly concurrently (with screen disabled)
    - with auto plane disabled, the plane always is the x-z-plane of the pointer
    - with auto plane enabled, the plane is automatically chosen based on the view (the one most perpendicular to the view-direction) and indicated on the pointer with a line connecting the corresponding tripod-axes
- parallel to the *screen-plane* / "back and forth", by sliding in the large pointer-area / small pointer-area, possibly concurrently (with screen enabled)
    - with *tilt-view* active, the current *screen-plane* is used, allowing to move in any direction even using just one finger
- freely in space using two fingers, where the average motion determines the movement parallel to the *screen-plane*, and *pinching* moves back and forth; therefor, start sliding with both fingers within the *large pointer area* (the screen and auto plane options have no bearing)

To improve efficiency and precision, the *tripod* by default moves faster (in *screen-space*) than you slide when you slide fast, and slower when you slide slowly. To make it follow the finger-movement directly, start sliding within the *grip-area* below the *tripod*.

Entering coordinates

You can also position the *tripod* by typing absolute or relative cartesian coordinates:

- Slide into the screen from right within the *type-rim*, the topmost zone marked by a red-green-blue line; the *text-input* appears.
    - to enter the coordinates relative to the current *tripod's* position, slide straight leftwards, so that left of the text-input "x +" is displayed
    - to enter absolute coordinates, slide downwards slantwise, so that left of the text-input "x =" is displayed
- Enter the relative or absolute x-value, tap the enter-key on the keyboard (it may read "Next" or "Go"), then likewise the y- and z-value; after entering the z-value, the pointer is positioned, and the *text-input* and keyboard hidden
    - to abort the input, use the *back-key* at any time
    - if the input is invalid, an *message* appears and you can enter the value again

Snapping and aligning

Snapping is essential both for positioning the *tripod* precisely, and for reaching points in space fast. Also, it allows for creating lines with shared vertices, and faces with shared vertices and edges, and previously separate parts can be welded together. You can snap to

- an adaptive 3d-coordinate-grid (grid-snap);
  it adjusts its spacing automatically to the zoom and the *tripod* -position, such that the distances in *screen-space* always stay appropriate – when zoomed out, or when the *tripod* is farther away it is coarser, and when zoomed in, or when the *tripod* is nearer, it is finer. Hence, for finer coordinates you just have to zoom in.
  - the spacing at the default view and pointer-position is 1 unit, when zooming out, it turns to 2.5, then 5, then 10, then 25 ..., and the other way round.
  - the target-distance in *screen-space* can be adjusted with the grid-distance preference
- existing geometry:
  - vertices
  - edges / lines / curves (the point nearest in *screen-space* to the *tripod*-position)
  - midpoints of edges / lines / curves (for splines, the point with the mean parameter is used)
  - faces (the point nearest in *screen-space* to the *tripod*-position)
  - face-centers

  for faces, the *tripod* always snaps to the nearest face under its screen-position, if any;
  for vertices and edges, it snaps to the target nearest in *screen-space*, if any:
  - only targets within a specific radius around *tripods* screen-position are considered, adjustable with the snap-radius preference in the Input-section
  - with x-ray disabled, only targets not covered by other faces are considered (even if the material of the faces is transparent, or if shading is disabled, so that you see the targets);
    with x-ray enabled, all targets under the *tripods* screen-position are considered

Several snaps can be active at once; if so, the priority is as follows: vertices, midpoints of edges, face-centers, edges, faces, grid-points. However, if grid-snap is enabled and the *tripod* would snap to a position not on a grid-point, it snaps to the nearest grid-point instead.

For drawing lines and creating primitives, the pointer-movement can be restricted to the coordinate-directions with ortho snap. If another snap is active additionally, and the *tripod* would snap to a position away from a snap-axis, it is projected to the nearest axis.

The pointer-*tripod*, and possibly the elements being moved, immediately skip to the target, while a dot moves unaffected according to your touch-motion.

The tripod can also align itself, and possibly the objects you are moving, to the faces or edges it snaps to; thereto, activate orient, available when a face- or edge-snap is active. To use “complex” snaps, it is necessary to zoom in appropriately and set the snap-radius large enough.

- with a face-snap enabled, the y-axis of the *tripod* is oriented perpendicular to the face
- with an edge-snap enabled, the y-axis is aligned along the edge
- with a face- and edge-snap enabled, and when the *pointer's* dot is near an edge as well as over a face, the y-axis is oriented perpendicular to the face and the x-axis along the edge
- if vertex-snap is active additionally, and a vertex is within reach, the *pointer* is positioned to the vertex, and aligned to the nearest edge and the face the dot hovers.
  - This is especially useful to establish a coord.-system suitable for modifying existing geometry

<u>To set the *tripod* to the *view-center*,</u> use *center pointer* from the V-menu.

The *reference coordinate-system*

The ***reference coordinate-system*** determines the axes and planes for moving the *tripod*, defines the *snap-grid,* and is used for entering and displaying coordinates*;* it is always aligned with the *tripod*, and it's origin is indicated by a small symmetric axes-cross.

To change the *reference coord.-system,*

1. activate the the ref. CS tool, available from the ✳- and ↗-ribbon; the *reference coord.-system* is positioned and aligned with the *tripod*
2. if required. either
    - move the *tripod* and tap to finish
        - enable an *edge*- or *face-snap* and orient to align the pointer; with face- edge- and vertex-snap enabled together, it can conveniently be aligned to a face
    - use the appearing commands or tools from the ribbon, see Transform-center and -orientation

Controlling object-display

You can control how the scene is displayed in many respects. Via the D-menu, you can at any time choose

- wireframe: weather the edges of objects are drawn
- shade: weather the faces are filled at all
- texture: weather textures are used
- x-ray: to draw all faces semi-transparent, and to make obstructed edges and vertices selectable
- NEW backface culling: weather only faces with the *normals* facing the camera are rendered
- lighting:
    - if enabled, the lights are used together with the viewing direction and the material properties to determine the surface-color
    - if disabled, the surfaces are displayed exactly in the assigned colors, resp. the texture-images are used directly
- lights: default / scene:
    - with default, a mild *ambient light* and one *directional light* is used, always shining towards the view-center from a little above, and slightly left, of the virtual camera
    - with scene, the lights placed in the scene are used; by default, there is one *point-light* at position (0,15,0)
- show lights: to display the lights (and light-*targets*)
- show normals: to display the *normals* as lines emerging from the vertices
- show UVs: to display the UVs as dots at (resp. beside) the vertex positions
- show texture corners: to draw dots at the texture images' corners
- home grid: to display a grid in the x-z-plane around the *origin*

Hiding and freezing

When working on specific objects, or details of a larger object, it can be of great use, or even necessary, to hide other objects or parts of an object.

to temporarily hide objects or faces from view, select them, then invoke *hide selected* from the V-menu to hide the unselected objects or the unselected faces of an object, use *invert* from the *selection-rollout* beforehand

to unhide all objects or all faces, switch to object-*selection-mode* or face-*selection-mode* then invoke *unhide all* from the V-menu

NEW For objects, you can alternatively freeze them; this way, they stay visible, but are not selectable

Drawing lines and curves

With Spacedraw, you can draw freehand lines, CAD-like polylines and splines (known as "paths" in 2d-vector-graphics) directly in 3D. They can be combined to arbitrary complex networks.

The lines can then be used to create surfaces and solids:

- they may be extruded, i.e. to create the walls of a building,
- “inflated” to arbitrary branched pipes, possibly of varying width, or
- networks can be used to define the contours of surfaces

All the tools for drawing are found in the ✳-ribbon. It is organized as follows

- First come the *tools*
    - freeh., for drawing freehand-lines
    - line, for straight lines (and polylines)
    - spline, for Bezier-curves
    - arc, for circular arcs
    - helix, for helices and spirals
    - rect., for rectangles and rectangular grids
    - n-gon, for regular polygons
    - circle, for approximate circles and polar grids
    - sphere, see Creating primitives
- next come the options for the active tool, if any; they are described in the specific sections below
- last are the options for pointer-movement and snap

For adjusting the lines, you can switch to the ↗-ribbon at any time with one tap, which is esp. important for splines.

to draw, in general

1. activate a *tool* and adjust the options
2. move the *tripod* to the start-point and tap
    - to continue an existing line, or to branch off from any point of an existing object, use point- or an edge-snap with combine enabled
        - in case of an edge-snap, the existing segment is split automatically
        - the start-vertex is shared with the existing segment, and the new line will be part of the existing object
3. move the *tripod* repeatedly to draw
    - with quick enabled, you never have to tap in-between; every time you release the touch, the current pointer-position specifies the next point resp. *control-point;* with quick disabled, tap for each point resp. *control-point*
    - you can undo every step
    - you can change the tool-, pointer-movement- and snap-settings between each step

        - to change the *drawing-plane*, it suffices to rotate the view with screen or auto-plane enabled
    - concurrently to moving the *tripod,* you can change the view, using a second or third finger, or by tilting the device with *tilt-move* enabled
        - with screen enabled, this also rotates continuously the drawing-plane
        - with auto-plane enabled, you can switch between the x-y-, x-z- and y-z-plane
    - you can switch between line, spline and arc by tapping the resp. ribbon-items
    - when you snap to an existing vertex of the same object and combine is enabled, it is shared with the adjoining lines; when snapping to a line, it is split automatically and a shared vertex is created
4. tap (with quick enabled) or double-tap (with quick disabled) within the *tool-zone* to stop drawing; now you can move the *tripod* to a next start-position
    - activating another *tool* or *ribbon-selector* also finishes drawing

NEW lines, splines and arcs can be inflated at creation, see Inflating lines

Freehand lines

This tool lets you easily draw arbitrary shaped curves in space by either

- moving the *tripod* freely in space using two fingers, or
- rotating the *screen-plane* while you draw, using multi-touch or by tilting the device, see Changing the view

Drawing freehand lines in 3d naturally is somewhat more involved than in 2d; for each stroke, you first move the *tripod* to the start-position, then move it to draw. There are two ways to indicate when the *tripod* should draw and when it should just move:

- tap to switch the mode
    - here, you can move the *tripod* freely in space to draw, using two fingers
- start dragging within the *grip-area* below the *tripod* to draw, somewhere else within the tool-zone to just move
    - here, the pointer moves directly with the motion of your touch in a plane parallel to the screen
    - you can't use the two-finger-methods to move while drawing; however, you can still move in any spatial direction by simultaneously rotating the view with another finger, or by tilting the device;
      thereto, set orbit / look around to orbit and orbit: center / pointer to pointer in the V-menu. The view then always rotates around the current *tripod*-position
    - you can also tap outside the *grip-area* to position the *tripod*

Freehand lines are created as a chain of straight line segments. As you move the pointer, vertices are inserted evenly spread in *screen-space*, adjustable with distance. The actual length of the segments therefor depends on the zoom and pointer position, and you can draw more precisely by zooming in.

You can *snap* a stroke's start-point to any *snap-target*; while drawing, all *snaps* are automatically disabled.

to simplify a freehand-line, use the auto-weld-command, see Automatic stitching and simplifying

to smooth a freehand-line, use *to splines* from the 🌀-ribbon

Straight lines

line lets you draw single straight lines or polylines. Every time you release the touch (with quick enabled) resp. tap (with quick disabled), a segment is added and a vertex inserted.

Splines

Splines (chains of Bezier-curves) are created and adjusted in Spacedraw similarly to “paths” in vector-drawing programs, using *anchor-points, control-points* and *handles*, except that they can form arbitrary complex networks, where *anchor-points* are shared between multiple splines.

Control-points are the end-points of the ***handles***. ***Anchor-points*** are vertices through which the curve passes. Transitions between segments are categorized in three types:

- ***corners***, where the splines meet at an angle
- ***smooth transitions***, where both splines have the same direction; the handles are parallel
- ***symmetric transitions***, where the handles additionally have the same length

Vertices where multiple splines meet are called coplanar if all splines join in a plane there.

Splines may be used to define the contours of complex organic surfaces, which can then be deformed by adjusting the splines.

Drawing

To draw splines, you alternately specify *anchor-points* and *control-points*:

1. move the *tripod* to the *start-point* and tap
2. move the *pointer* to the first *control-point* to define the start-direction (with quick disabled, tap)
    - the distance of the control-point influences the curvature
3. repeatedly
    1. move the *pointer* to the end-point of the segment (with quick disabled, tap)
    2. either
        - for a *smooth (symmetric) transition*, move the *pointer* to define the tangent and curvature (with quick disabled, tap)
        - for a *corner*,
            1. move the *pointer* to define the end-direction of the segment (with quick disabled, tap)
            2. activate "new dir.", then move the *pointer* to specify the start-direction for the next segment (with quick disabled, tap)
4. tap (with quick enabled) or double-tap (with quick disabled) within the *tool-zone* to stop drawing, or activate the line or arc-tool to continue with a line or arc

Editing

To insert an anchor-point without changing the shape

1. with the spline-tool active and an *edge-snap* and combine enabled, snap the *tripod* to the spline at the desired position
    - to split the segment at the middle parameter value, use midpoint-snap
2. tap twice

To remove anchor-points, switch to the ↗-ribbon, select them and invoke *delete;* a new spline-segment will be inserted joining the neighboring anchor-points

To continue a spline smoothly from an endpoint, snap to it using point-snap, then tap

To convert straight line segments to splines, invoke *to splines* from the -ribbon

To convert spline segments to polylines (individual straight lines), invoke *to poly* from the -ribbon

Adjusting

Splines are adjusted using the selection- and transform-tools from the -ribbon.

For shaping splines and *spline-surfaces*, the m,r,s-tool is especially well suited: it can simultaneously move an anchor-point and rotate and scale the related handles, see Moving, rotating and scaling concurrently. With screen enabled, you can concurrently rotate the transformation-plane by changing the view.

to adjust an individual *handle*, first select the corresponding vertex NEW or an adjacent edge to make it appear, then select the endpoint of the handle

- the move-tool is activated automatically
- you can switch to another *handle* by tapping it's endpoint; multiple handles can't be selected simultaneously
- for vertices where two splines meet, link handles links the unselected handle such that it is rotated and scaled accordingly, so smooth and symmetric transitions are preserved
- to see the effects on the spline, it is essential to concurrently change the view

to adjust all handles of a control-point together, select the corresponding control-point and use the m,r,s-, rotate- or scale-tool

- when selecting a single vertex where one or more splines end, the m,r,s-tool is activated automatically

to transform the handles of all selected vertices or edges alone, disable sel. vertices

to transform the selected vertices without the handles, or the vertices of all selected edges alone, disable sel. handles

to move, rotate or scale whole segments, select them in *edge-select* mode and use the transform tools

to make the transition between two segments *smooth* or *symmetric*, invoke *smooth* or *symm.*

- if the segments meet in a vertex where no additional spline segments branch off, you may select the vertex; otherwise, select the two segments

to make segments straight, invoke straight

to make all segments leading away from a vertex start in direction of the opposite vertices, invoke straight with the vertex selected

to make segments meeting at a vertex join in a plane there, invoke coplanar

- to affect all spline segments meeting at vertex, select the vertex
- to affect only certain segments, select them

Arcs

Arcs can be constructed in ways familiar from CAD-*Polylines*, but in 3d:

1. move the *tripod* to the *start-point* and tap
2. either
   - to specify a point on the arc, select through, move the *pointer* to the *through-point*, then to the *end-point*
   - to specify the center, select center, move the *pointer* to the *center*, then to the *end-point*

        - for the *end-point,* the *pointer*-position is projected on the sphere defined by *start point* and *center*.
        - to change the *winding* (CW or CCW), move the *pointer* first again towards the *start point*, then away from it in the other direction
    - to specify the *start-direction*, select direction, move the *pointer* to define the *start-direction*, then move it to the end point
    - if the *start-point* is the *end-point* of a single straight line or arc, select cont. dir. to continue the direction; then you only have to specify the *end-point*
3. continue with the next arc, activate the line or spline-tool to continue with a line or spline, or tap to stop drawing

Editing networks

For editing line- and curve-networks, the move-tool from the -ribbon, and commands from the -ribbon are used:

to weld vertices to their center, invoke *weld* from the -ribbon

- the vertices may belong to different objects; if so, the objects are combined automatically

to weld a vertex on another vertex or edge (*target-weld*), move it to the destination with *vertex*-, or an *edge-snap* enabled

- in case of an edge-snap, the segment is split automatically
- Only elements belonging to the same object are welded this way; to weld to a distinct object, first use *combine* from the -ribbon

to automatically weld lines together at proximal vertices, use auto-weld, see Automatic stitching and simplifying

to split a line at a vertex, invoke *split*

- for each connected line, a new vertex is created, and you can now successively select and move them apart

to split off a part from a line or network, select the respective segments and invoke *split*

Helices and spirals

The *helix*-tool can create circular and conical helices and planar (Archimedean) spirals. They are defined via three points and the number of *full turns*:

To draw a helix or spiral

1. move the *tripod* to the center of the “*base-circle*” and tap
    - to draw the helix around to a line, or perpendicular to a face, use an edge- or face-snap with orient here
2. move the *tripod* to the start-point; it defines the “start-radius” and “start-angle”
    - it stays always in the plane of the *base-circle*; when you move the *tripod* out of the plane, it is project back
    - to start the helix or spiral exactly at the center, use point-snap here to snap “back” to the first point you specified
3. move the *tripod* to the end-point; it can lie anywhere in space and defines the height and the “end-radius” and “end-angle”
    - for a planar spiral, only slide in the *large pointer-area*
    - for a circular (“straight”) helix, only slide in the *small pointer-area*

4. adjust the turns and divisions
    - turns is the number of *full turns* and may be 0, added to this is a *fractional turn*, defined by the angle between start- and end-point
    - when changing the turns, the divisions are adjusted automatically
5. tap to finish

After the first tap, screen and auto-plane is disabled automatically, so sliding in the *large pointer-area* always moves in the plane perpendicular to the helix’s axis, while sliding in the *small pointer-area* moves parallel to it

Rectangles, regular polygons, circles and grids

These are created with the same tools as the primitives, see the next chapter. Use

- rect. for rectangles and rectangular grids
- n-gon for regular polygons and polar grids
- circle for approximate circles and polar grids

and make sure fill is disabled.

Creating primitives

Spacedraw can create plenty primitive objects with various parameters. The surface may be filled, or a wireframe may be created, the sides can be divided and a (possibly textured) material can be assigned at creation-time.

The primitives can be positioned anywhere in 3d-space, or placed on the surface of other objects, aligned to a face (known as "dynamic UCS", "AutoGrid" or “live objects”).

Types and tools

For creating primitives, the last four tools of the ✳-ribbon are used; 3d-objects are created by "***extruding***" a planar shape perpendicular to the plane, to a point, or radial:

- rect. creates
    -
    - **rectangles** and **rectangular grids** between diagonally opposite corners
    - **cuboids** (**boxes**) between diagonally opposite corners
    - **rectangular-based pyramids** by specifying the base and the height
- n-gon creates
    - **regular polygons** by specifying the center and a corner
    - **prisms** by specifying the *base polygon* and the height
    - **pyramids** by specifying the *base polygon* and the height
    - “**cornered rings**” by specifying the center, a point on the *mid-circle* of the *tube* and the *tube’s* radius
- circle creates
    - **discs** by specifying the center and a point on the circumference
    - **cylinders** by specifying the *base circle* and the height
    - **cones** by specifying the *base circle* and the height
    - **rings** (**tori**) by specifying the center, a point on the *mid-circle* of the *tube* and the *tube’s* radius

- sphere creates
    - **spheres**, divided along *circles of latitude and longitude*
    - “**icospheres**”, possibly tessellated icosahedra

  by specifying the center and a point on the surface

Circles are approximated by regular polygons, and the only differences between the circle- and n-gon-based objects are

- the larger default number of sides for circle (here called divisions)
- for circle, the base is aligned with the coordinate-system, while for n-gon the position of a corner is defined by the *pointer-position*
- the faces of circle-based objects are marked and rendered as curved around the girth, see Curved and flat faces, smooth and hard edges

Creation process

All primitives are created by specifying two points; the orientation is defined by the *reference coord.-system*:

1. activate the suitable *tool*
2. move the *tripod* to the first point and tap
    - the *tripod’s* position and orientation define the *construction-plane*
    - to place the primitive on the surface of another object, enable faces- or centers-snap, then activate orient; the *tripod* will align itself with faces, see Snapping and aligning
3. move the *tripod* to define the dimensions
    - screen and auto-plane are disabled automatically, so sliding in the *large pointer-area* always moves in the *construction-plane* and dimensions the *base*, while sliding in the *small pointer-area* moves “up and down” and changes the height
    - how the pointer-position is interpreted depends on the object-type:
        - for a sphere, specify any point on the surface
        - for a planar object, only move in the plane
            - if you accidentally move off the plane, just disable the box-, prism- or cylin.-option again
        - for an extruded object, raise the *tripod* off the plane
            - the shape is extruded perpendicular to the construction plane, and the ribbon-item for the respective object-type (box, prism, cylin.) turns active
            - if required, select another type now
    - the dimensions are displayed in the *status-line*
    - simultaneously, the view may be changed to get an 3d-impression of the object
4. adjust the parameters
    - the parameters and the object-type can be changed prior to defining the first point, or before or after the dimensions, or alternately
    - to assign a material,
        1. tap material to open the *material-panel*
        2. tap the image of the material to assign in the list
5. tap to finish

Parameters

There are various parameters to influence the character of the object created:

- fill: whether the surface is filled, or a wireframe is created
- material, if fill is enabled: lets you assign a (possibly textured) material; mapping-coordinates are created in any case, so you can also easily assign a textured material later
    -
    - when a scene-material is selected in the material-panel (even if it's closed), it is used for the new object; else, the *standard-material* is used, see Creating, changing and assigning materials
- smooth, for round objects: whether edges are marked and rendered smooth, see Curved and flat faces, smooth and hard edges
    -
    - disable this for a facetted object
- various **division**-numbers
    -
    - for grids, first activate divide option of the rect.-tool to make the x- and y-divisions available
    - for prisms, pyramids, cylinders and cones, set cap division to -1 to create no *caps*

Additionally for rings and tori:

- roll: the rotation of the *tube*
- twist: the twist of the *tube*

Additionally for spheres

- icosphere: to create a icosahedron, possibly spherical *tessellated*

Selecting

For all modeling operations, you first select the related *elements*. There are four types of selectable ***elements*** and corresponding ***selection-modes***:

- ***objects***, collections of lines, curves and surfaces
    - individual *instances* can be selected and transformed independently, see Organizing, multiplying and arraying objects
- ***faces***, usually planar, surface components
- ***edges***,
    - the borders of faces, possibly shared by two or more faces, and also
    - lines, the segments of *freehand-lines* and of splines, and arcs
- ***vertices***,
    - the corners of faces, possibly shared by many faces, and also
    - the junctures and end-points of lines, spline-segments and arcs

The modeling tools and commands vary with the *selection-modes;* some only work in certain modes, while others adapt their behavior to the type of *elements* selected. The -, and -ribbons therefor start with the *selection-modes*, and the available tools and commands depend on the *mode* selected.

You can select any number of *elements*, and then apply every meaningful operation to them at once. For faces, edges or vertices, they may belong to different objects; where needed, these are combined automatically (i.e. bridge or weld).

For sculpting objects, elements can also be selected partially so that the effect of transformations gradually decreases with the distance from the explicitly selected point or region, called *soft selection*.

If multiple elements are selected, there always is one ***active element***, which is treated special by some operations. When adding an individual element to the selection, it becomes *active*; when adding multiple elements, and there is not already an *active element,* it is chosen somewhat arbitrarily. To make a specific selected element *active*, first deselect it, then select it again.

How the selected elements and the *active element* are indicated, and how preview-highlight is performed, depends on the *element-type* and active tool, and can be changes in the preferences:

- for vertices, dots with a specific color or size are displayed
- edges are displayed in a specific color
- faces may be filled in a specific color, or their boundary-edges may be displayed in a specific color, to choose with the fill selected faces preference; however, for color- or material-assignment, always the second method is used
- for objects, the wireframe is displayed in a specific color

The *ribbon-items* object, face, edge and vertex are special; they serve both as tools and as indicators for the *selection-mode*:

- When another tool is active, the *selection-mode* is indicated in blue (the selected option-color); tapping a *selection-ribbon-item* finishes the current tool and activates the *selection-tool*
- When a *selection-tool* is active, the ribbon-item shows in green (the active tool-color)

There are many tools and commands for selecting, which can be used successively to select all the desired *elements*. They are available from the -, - and -ribbon; by default, only the lasso-tool is shown, the other tools are grouped in the ***selection-rollout**,* which can be expanded or collapsed by tapping the ...-symbol right from the lasso-tool.

to clear the selection, tap the *tool-zone* somewhere away from any object, or tap the view-zone. (tapping the view-zone always clears the selection, and does not select elements at the tap-position)

Selecting *elements* individually

By simply tapping, or by using the *picker,* aided by preview-highlight, *elements* can be selected very fast, and with any needed precision.

to select or deselect a single object, face, edge or vertex

1. activate the respective *selection-tool*
2. either
   - tap the element, or
   - slide with one finger within the *tool-zone* to position the *picker* until the element is highlighted, then release the touch

Lasso- and paint-selection

Using two fingers, you can intuitively shape, and simultaneously move, a rectangular “lasso” or a circular “brush” to select multiple elements NEW or objects at once:

to select all elements within a rectangular region (known as *marquee-select* or *rectangle-select*)

1. switch to the desired *selection-mode* and activate the lasso-tool
2. slide with two fingers within the tool-zone to shape a rectangle; the average motion moves the rectangle, the horizontal distance determines its with, the vertical distance its height
3. when the desired elements are framed, release the touch

to select elements by “painting” over them (known as *paint-select* or *drag-select*)

1. switch to the desired *selection-mode* and activate the paint-tool from the *selection-rollout*
2. slide with one finger to move the *picker* to the start-position

3. slide with two fingers within the *tool-zone* to shape a circle; the average motion moves the circle, the horizontal distance determines the radius; all *elements* touched by the circle get selected

NEW analogously, you can deselect objects or elements with the uns. lasso- and uns. paint-tools

Paths, loops and rings

to select all faces, edges or vertices along the shortest path between two *elements*, select the start-*element*, activate the path-tool from the *selection-rollout*, then move the *picker* to the end-*element*; while you move the pointer, the path is highlighted

to select irregular paths, use the path-tool repeatedly; the path always starts from the *active element*, which is the end-element of the previous path-segment

Loops and ***rings*** are "straight" paths of faces, edges or vertices across a surface, defined for topology with four-sided faces; they may or may not be closed and can have gaps. The *loop* and *ring* commands let you easily create complex repeating gap-patterns.

- a ***face-loop*** is a path of faces, connected in sequence by their shared edges
- an ***edge-loop*** is a string of edges, connected in sequence by their shared vertices; it only propagates through four-way junctions
- an ***edge-ring*** is a path of "parallel" edges, connected in sequence by their shared faces
- a ***vertex-loop*** is a path of vertices, connected in sequence by their shared edges

Which of the previously selected elements are considered by the *loop* and *ring* commands depends on the element-type and the gap-settings:

- for faces and vertices, a *loop* is defined by at least two elements within it, and the *loop*-command expands the selection from the *active face / active vertex* in the directions where faces / vertices are already selected
- for edges, a gapless *loop* or *ring* is defined by one edge within it, and the *loop*-command selects every *loop* / *ring* with at least one edge selected

to select a gapless *face-* or *vertex-loop*, select two adjacent faces / vertices within it, then invoke *loop*

to select gapless *edge-loops or -rings*, select one edge within every loop / ring, then invoke *loop* / *ring*

to selected multiple *face-* or *vertex-loops* at once, first select the corresponding *edge-rings* / *loops*, then convert the selection

to select a loop or ring with a repeating pattern, select one "instance" of the pattern, invoke *loop,* then set the gap-setting to the pattern-length

Special selections

to select all objects, clear the selection, then invoke *invert*

to select all instances of the active object, invoke *instances*

to select all faces, edges or vertices of an object, select it in object-mode, then tap face, edge, or vertex twice.

to select connected faces, edges, or vertices, invoke *connected*; all elements connected to any previously selected element will be selected

to select UV-connected faces, edges, or vertices, invoke *UV-conn.*; all elements that are via between faces shared UVs connected to any previously selected element will be selected

to select border-edges, -faces or -vertices, invoke *border*; the elements along every border with at least one selected element will be selected

Modifying selections

to invert the selection, invoke *invert*; for faces, edges or vertices, all *meshes* with some selected elements are considered; lights are not considered

to select the “outline” resp. “border” of the current selection, invoke *outline*; for faces and vertices, the meaning is clear; for edges, the command selects the border-edges of the faces adjacent to the previously selected edges, creating interesting patterns, changing with repeated application of the command

to grow or shrink a selection, activate the grow/shrink tool, slide up/down within the tool-zone until the desired elements are highlighted, then release the touch

Switching selection-modes, converting selections

When you switch the *selection-mode*, the current selection is remembered and restored when switching back.

Selections can be converted a to another type; in some cases this is ambiguous,

- for the maximal number of elements, tap object, face, edge, or vertex twice
- for the minimal number, long-tap (touch and hold for at least half a second)

When converting to objects,

- with twice-tap, all instances of the meshes with selections are selected
- with long-tap, for every mesh, only the instance on which the last selection was done is selected

Soft selection

Soft selection automatically selects elements surrounding the explicit selection “partially”, gradually decreasing with the distance, measured either as the

- spatial distance in 3d-space
    - therefor, enable spatial, and set spatial reach to the value up to which the soft selection shall extend
- *edge-distance*, the number of edges of the shortest path to the explicit selection
    - therefor, enable edge-dist. and set edge reach to the value up to which the soft selection shall extend

The distance may also be spatially measured, but the selection limited by a maximal *edge-distance;* this is e.g. useful when you don't want to influence nearby unconnected parts. Therefor, enable both spatial and edge-dist. and set the desired values.

How the selection strength decreases with the distance can be adjusted with falloff speed.

The selection strength falloff is represented as a color gradient on the vertex-dots, ranging from the selected vertex color to the normal vertex dot color.

To create a soft selection, enable soft in the ↗-ribbon, then

- select vertices, edges or faces using any *selection-tools*; surrounding vertices are partially selected automatically, according to the settings
- adjusted the settings; the vertices surrounding the explicit selection are partially selected accordingly

Transforming

Spacedraw's multi-touch *transform-tools* let you move, rotate and scale *elements* in various intuitive and efficient ways; use

- move to translate along an axis, in a plane, or simultaneously in all spatial directions,
- rotate to rotate around an axis, or freely around all 3 axes,
- scale to scale along an axis, in a plane or uniformly
- m,r,s to move and rotate in a plane, and uniform scale concurrently
    - when using the *screen-plane*, it may be rotated simultaneously by changing the view
- mirror to mirror through a plane

You can also quickly switch between selecting and moving without activating a *transform-tool,* using different touch-actions.

The transform-axes may be chosen automatically according to the selection or the view, or aligned with existing geometry.

Simultaneously to transforming, you can change the view to get a 3d-impression of the effects, using multi-touch or by tilting the device.

Various *accompanying operations* can be performed in the process of the transformations; objects may be *copied* or *instantiated*, possibly multiple times, creating arrays; faces may be detached or *copied*; faces, edges and vertices may be *extruded*.

Any number of elements can be transformed at once,

- objects may be transformed about a common *transform-center* or individually
- faces, edges or vertices of different objects may be transformed about a common *transform-center*

Using the tools

All the tools for selecting, transforming and the *accompanying operations* are found in the ↗-ribbon. It is organized in sections as follows:

- the four *selection-modes / -tools* object, face, edge and vertex
    - tapping one switches to the respective *mode* and also finishes any active *transform-tool* and activates the *selection-tool*
    - the *selection-mode* determines the available *selection-options* and *-commands* and the *operations*
- the special *selection-options* and *commands*
- the *transform-tools* move, rotate, scale, m,r,s, mirror, and the *delete*-command
- the applicable ones of the *accompanying operations* copy, instance, extrude, detach
- the options for pointer-movement and snap, depending on the *active tool*
- the *reference-coord.-system-tools* and *-commands*

to transform, in most general

- activate a *selection-tool* and select the *elements* to transform
    - to rotate or scale the adjoining elements of a vertex about it, you only have to select the vertex
- select an *accompanying operation*
- specify the *transform-center and -orientation*
- activate a *transform-tool* and adjust the move- and snap-options, resp. select the *transform-axis* or *-plane*
- perform the transformation, see the respective sections

- either
    - to finish, tap within the *tool-zone*, activate a *selection-tool* or another *menu*
        - with a tap, you can simultaneously select an element for the next transformation
    - continue with 2. to start a new *operation* with the selected elements
    - continue with 3. or 4. to further transform

Transform-center and -orientation completely revised

The displayed and typed coordinates relate to the *reference coord.-system*; it is also used to specify the *transform-center*, *-axis* resp. *-plane* for rotate- and scale-transformations. The *base point* and snap position for moving is determined by the *tripod*. Prior to a transformation, you can use the ref. CS and ref. pos. tools to change the *reference coord.-system* resp. the tripod position:

Activate ref. CS, then either

- move the *tripod*, then either tap the view to set both the *reference coord.-system* and the *tripod*, or activate the ref. pos. tool to set another *tripod* position (the tripod is reset to the position where it was before invoking ref. CS in this case)
    - enable an edge- or face-snap and orient to align the *ref. CS* along an edge or with a face
    - activate rotate (the one besides ref. pos.) to rotate the *reference coord.-system*; the procedure is the same as for rotating elements
- use the appearing commands:
    - *selection* (only for vertex-, edge- and face-selections): the the y-axis of the coord.-system is aligned with the average normal of the affected faces, the origin is set to the geometric center of all affected vertices

      (the position on the *instance* used last for selecting is used)
    - *object*: use the coord.-system of the active object
    - *each* (only for objects): for every selected object, its own coord.-system is used
    - *parent* (only for objects): use the coord.-system of the "common" parent of the selection (or the world CS if there's none)
    - *world*: the world coord.-system is used

Activate ref. CS, then either

- move the *tripod*, tap to finish
- use the appearing commands:
    - *active* (only for vertex-, edge- and face-selections): the center of the active (last selected) face, edge, or vertex is used
    - *selection*: the center of the selection is used (for objects the pivots are averaged)
    - *center* (only for objects): the center of the respective vertices is used
    - *object*: use the coord.-system origin of the active object
    - *each* (only for objects): for every selected object, its own coord.-system origin is used
    - *parent* (only for objects): use the coord.-system origin of the "common" parent of the selection (or the world CS if there's none)

The used commands are remembered and applied at each subsequent selection automatically.

If ref. CS is set to *each*, ref. pos. also becomes *each*, but not the other way round.

The coord.-system of objects (the "*pivot*") can be set to the *tripod* with the *set obj. CS* command (available when ref. CS or ref. pos. is active)

Automatic positioning and orienting

When selecting elements, the *tripod* is

- automatically positioned at the selection-center according to the *pos. sel.*-command
- in face- edge- or vertex-selection-mode, with auto-ori. enabled, it also orients itself according to the *ori. sel.*-command

and updated when modifying the selection. With the show tripod while selecting preference disabled, the pointer is hidden until you activate a transform-tool or perform a move-operation.

Transforming multiple objects at once

When multiple objects are selected, they may be rotated around / scaled from either

- a common center
    - therefor, use the ref. CS-tool or invoke *pos. sel.* to position the *tripod*
- their own pivots
    - therefor, invoke *each*
    - for each selected object, a tripod is displayed at its pivot, aligned with the *transform-axes* / *-planes* that will be used

In case of independent transform-centers, the *transform-axes* / *-planes* may be either

- the same for all objects
    - therefor, use the *ori. obj.*-, *ori. world*- or *ori. view*-commands, or the rotate-tool (the ref. CS-tool will also set the transform-center to the *tripod*s position for all objects)
- for each aligned with its coord.-system
    - therefor, invoke *each*

Selecting and moving with a *selection-tool*

With a *selection-tool* active, you can select or move, using different touch-actions:

- tapping always selects
- sliding with one finger within the *tool-zone* moves the picker and selects, with one exception:
    - with elements selected, start sliding up / down within the *small pointer-area* to move
- when sliding with two fingers within the *tool-zone*
    - with lasso or paint enabled, the “lasso” resp. “brush” is moved, see Lasso- and paint-selection
    - else, the *tripod* is moved together with the selected elements, see Moving the *tripod*

to quickly move parts of a surface perpendicular or parallel to it,

1. activate the face-, edge- or vertex-tool
2. in the *selection-rollout*, activate auto-orient; deactivate both screen and auto-plane
3. select the elements to be moved; the *tripod* will align its y-axis with the average of the affected faces *normals,* and update its orientation when modifying the selection
4. either
    - to move perpendicular to the surface, slide up / down with one finger within the *small pointer-area*
    - to move along the surface, hold down one finger within the *small pointer-area,* and simultaneously slide another within the *large pointer-area*

        - sliding up / down within the *small pointer-area* simultaneously moves perpendicular to the surface

Moving

to move freely through space,

1. either
    - with a *selection-tool* active, make sure the lasso- and paint-options are disabled, or
    - activate the move-tool
2. if desired,
    - select an *accompanying operation*
    - adjust the snap-options and move the *tripod* to the *snap-position*, using the ref. CS-tool
        - with the move-tool active, all snaps are disabled when using 2-finger-move
3. slide with two fingers within the *large pointer-area* to move the selected *elements* together with the *tripod*, see the 3. method in Moving the *tripod*
4. to move another element, select it; the transformed elements are deselected automatically

to move in a specific plane,

1. activate the move-tool
2. if required, orient the *tripod* appropriately, see Transform-center and -orientation
    - to move parallel to a face or perpendicular to an edge, use the ref. CS-tool with a face- resp. edge-snap and orient enabled to align the *tripod*, then turn the snap off again
3. disable screen, enable auto-plane and orient the view roughly parallel to the desired plane
    - the plane that will be used is indicated on the *tripod*
4. slide with one finger within the *large pointer-area*
    - sliding within the small pointer-area moves along the perpendicular axis
    - sliding with two fingers moves freely

to move along a specific axis,

1. activate the move-tool
2. if required, orient the *tripod* appropriately, see Transform-center and -orientation
    - to move perpendicular to a face or parallel to an edge, use the ref. CS-tool with a face- resp. edge-snap and orient enabled to align the *tripod*, then turn the snap off again
3. either
    - tap the respective axis-arrow on the *tripod*, then slide with one finger within the *tool-zone,* roughly parallel to the axis
        - the other two arrows will disappear
    - disable screen, enable auto-plane and orient the view roughly parallel to the perpendicular plane, then slide up / down within the *small pointer-area* with one finger
        - this way, you always move roughly in / out the screen, which makes it difficult to estimate the movement

to enter the absolute or relative target coordinates, follow the steps described in Entering coordinates

Rotating

to rotate freely in space,

1. activate the rotate-tool
2. if required, move the *tripod* to the (common) rotation-center, or invoke *each* to rotate each selected object around its own *pivot*
3. slide two fingers within the *tool-zone;* twisting rotates in the screen-plane, the average motion rotates around the axis perpendicular to the motion (the distance has no influence); imagine turning a freely around its center rotatable ball.
    - sliding with one finger also rotates around the axis perpendicular to the motion

to rotate around a specific axis,

1. activate the rotate-tool
2. if required, define the rotation-axis; either
    - move the *tripod* to a point on the (common) rotation-axis, or
    - invoke *each* to rotate each selected object around an axis through its own *pivot*
        - now, each object will rotate about its own coordinate-axes;
        - to use a common axis-direction, use the *ori. obj.*-, *ori. world*- or *ori. view*-commands, or the rotate-tool after the *each*-command
3. tap the respective axis-arrow on the *tripod;* the other two arrows will disappear
4. slide one finger within the *tool-zone,* if the axis is roughly perpendicular to the screen, slide "around" the transform-center, else roughly perpendicular to the axis

    .

    - sliding with two fingers freely rotates

to type the rotation-angle,

1. follow the steps 1.-3. above to specify the rotation-axis
2. slide into the screen from right within the *type-rim*, like for entering coordinates; the text-input appears
3. enter the rotation-angle in degrees, according to the *right-hand rule*

Scaling

to uniform scale,

1. activate the scale-tool
2. if required, move the *tripod* to the scale-center, or invoke *each* to scale each selected object out from, resp. in toward, its own pivot
3. either
    - slide one finger within the *tool-zone* roughly away from, resp. towards, the *tripod*
    - *pinch* with two fingers within the *tool-zone*

to scale along an axis or in a plane

1. activate the scale-tool
2. if required, define the scale-center and the scale-axis resp. -plane; either
    - move the *tripod* to the (common) scale-center and orient it appropriately
    - invoke *each* to scale each selected object out from, resp. in toward, its own pivot
        - now, each object will scale in alignment with its own coordinate-system;
        - to use a common axis resp. plane, use the *ori. obj.*-, *ori. world*- or *ori. view*-commands, or the rotate-tool after the *each*-command
3. tap the respective part on the *tripod,*

    - to scale along an axis, tap the respective axis-arrow; the other two arrows will disappear
    - to scale in a plane, tap the line connecting the respective axes; the arrow on the perpendicular axis will disappear
4. slide one finger within the *tool-zone* roughly away from, resp. towards, the *tripod*
    - pinching with two fingers scales uniformly

to type the scale-factor,

1. follow the steps above to specify the scale-center, and possibly scale-axis resp. -plane
2. slide into the screen from right within the *type-rim*, like for entering coordinates; the text-input appears
3. enter the scale-factor

Moving, rotating and scaling concurrently

Using two or three fingers, elements can intuitively be moved, rotated and uniformly scaled simultaneously:

1. activate the m,r,s-tool
2. if required, move the *tripod* to the (common) transform-center, or invoke *each* to transform each selected object about its own pivot
3. slide two fingers within the tool-zone; the average motion moves in the *screen-plane*, twisting rotates in the *screen-plane*, *pinching* scales uniformly.
    - Simultaneously, you can orbit the view around the transform-center to rotate the *screen-plane;* thereto, set orbit / look around to orbit and orbit: center / pointer to pointer in the V-menu
4. to move another element, select it by either tapping or moving the *picker* with one finger; the transformed elements are deselected automatically and the m,r,s-tool stays active

Mirroring

to mirror through a plane

1. activate the mirror-tool
2. if required, define the *mirror-plane*; either
    - move the *tripod* to a point in the (common) mirror-plane, or
    - invoke *each* to mirror each selected object through a plane through its own *pivot*
        - now, each object will be mirrored through its own coordinate-planes;
        - to use a common plane-orientation, use the *ori. obj.*-, *ori. world*- or *ori. view*-commands, or the rotate-tool after the *each*-command
3. tap the *tripod*'s arrow on the axis orthogonal to the *mirror-plane*

Accompanying operations

The *accompanying operations* specify what happens with the selected (and affected unselected) *elements* in the process of the transformations. The possible *operations* depend on the *type* of the *elements* selected:

- objects:
    - by default: the original objects are transformed
    - copy / instance: copies / instances of the objects are created,

        - with repeat set to greater than 1, multiple additional copies / instances are created at once, which are transformed multiple times, producing arrays

- faces:
    - by default: they are moved together with their edges and vertices, deforming the adjacent faces
    - extrude: new faces are created, connecting the transformed with the adjacent faces
        - NEW using rep. extr. instead, with every transformation, a new extrusion is performed
    - detach: the transformed faces are detached, leaving behind a gap
        - with new obj. enabled, a new object is created from the detached faces; otherwise, they stay part of the object they come from
    - copy: copies of the selected faces are transformed
        - with new obj. enabled, a new object is created from the copied faces; otherwise, they become part of the object they come from
- edges:
    - by default: they are moved, deforming the connected faces
    - extrude: new faces are created between the old and new positions
    - detach: the transformed edges (of lines or curves) are detached if they are not connected to faces
    - copy: copies of the selected edges are transformed, for edges of faces, this yields a wireframe-model
- vertices:
    - by default: they are moved, deforming the connected faces
    - extrude: lines are created between the old and new positions

The *transform-tool*, the *transform-axis* resp. *-plane* and the *reference-coord.-system* may be changed during an *operation*; that way, you can create complex extrusions or arrays, where the elements are scaled, rotated and translated in various ways.

Snapping, aligning and welding

When elements are moved and the respective snaps are enabled, they snap to other elements or the coordinate-grid, with the following exceptions

- when transforming with two fingers in distinct *touch-areas*, all object-snaps are disabled
- when transforming with two fingers in the *large pointer-area*, grid-snap is disabled
- with the move-tool active, or when another than the ↗ -ribbon is loaded, all object-snaps are disabled when transforming with two fingers somehow

When objects are moved and orient is enabled in the snap-options, they are also aligned to the faces or edges they snap to; the *tripods* x-z-plane will be aligned with the face, resp. its y-axis with the edge. To align a specific face or edge on the moved object with another face or edge, orient the *tripod* accordingly before moving; see Transform-center and -orientation.

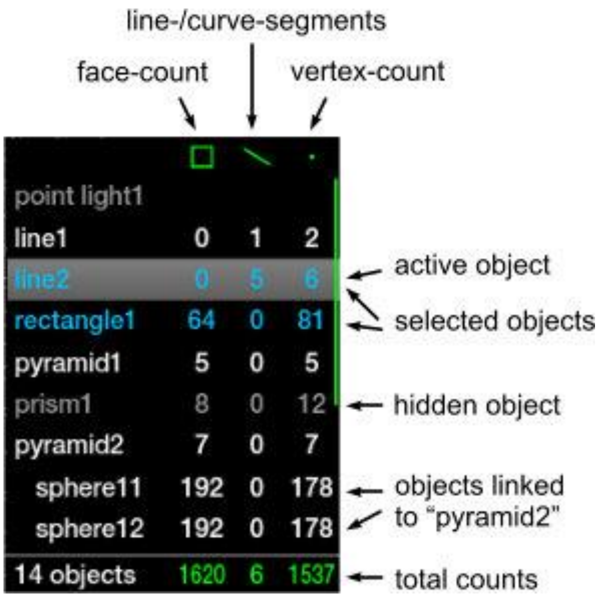
When moving and snapping vertices or edges to other vertices or edges on the same object, they are automatically welded to the targets, possibly collapsing shared edges and faces, see also Removing and combining elements.

Organizing, multiplying and arraying objects

Surfaces and lines are grouped to ***aggregates***, which can be treated as a unit. In principle, arbitrary collections can be grouped; they are often connected, but needn't be. The representation of an aggregate at a specific location in the scene, with specific scale and rotation, is called ***object***.

The scene-explorer

The scene-explorer-panel lists all objects in the scene, shows statistics and lets you select and rename objects; it can be shown or hidden by sliding leftwards into the screen over the scene-explorer-rim, the white line at the right screen's border, at any time.



	☐	⟍	·
point light1			
line1	0	1	2
line2	0	5	6
rectangle1	64	0	81
pyramid1	5	0	5
prism1	8	0	12
pyramid2	7	0	7
sphere11	192	0	178
sphere12	192	0	178
14 objects	1620	6	1537

It is structured and used as follows:

- the selected objects are indicated in blue, the active object is highlighted
- hidden objects are displayed in gray
- *linked* objects are indented and placed below their *superior*
- besides the objects the number of their faces, line-/curve segments, and vertices is shown, the total counts are displayed below the list
- tapping an object selects it, deselects all others, and possibly switches to object-selection mode
    - when the create-ribbon is loaded, tapping has no effect
- tapping the active (highlighted) object shows the text-input for renaming it; use the *enter-key* after typing the new name to apply it and close the text-input, or use the *back-key* to cancel

Splitting and combining *aggregates*

When drawing lines and curves or creating primitives, *aggregates* and *objects* are created automatically, but they can arbitrarily be split or combined at any time.

<u>to split an aggregate</u>,

1. select the faces or lines to separate
2. enable new obj. and then detach in the *-ribbon* , then move the detached part away; a new object is created from the selected part.
    - if selected and unselected elements share vertices or edges, they are split

<u>to combine aggregates</u> into a single one

1. select the objects to combine

2. invoke combine from the -ribbon
    - if a selected object has multiple instances, it is first made unique automatically; see the next section

Creating *copies* and *instances*

One *aggregate* can be placed in a scene multiple times at distinct locations, with different rotation and scale, yielding individual *objects*; they are called *instances*. If an *aggregate* only has one *instance*, the terms *object* and *aggregate* are interchangeable.

All *instances* of an *aggregate* are treated equal; to modify an *aggregate*, you can use any *instance*, and the other *instances* will change accordingly.

When duplicating *objects*, they can either be

- *copied*, creating individual copies of the *aggregate* for each new *object*, or
    - therefor, use copy with transformations
- *instantiated*, using the original *aggregate* for each new *object*
    - therefor, use instance

To modify an *instance* individually, you can convert it to a unique *object*; therefor, invoke *make unique* from the -ribbon

- for every selected *object* that has multiple instances, a copy of its *aggregate* is created and assigned to the object

Relating objects

Objects can be ***linked*** hierarchically, such that all ***subordinate objects*** are selected automatically together with a ***superior object***; this makes it easy to transform related objects together. The hierarchies may have any depth, i.e. an object with subordinates can be linked to another object.

to link objects, invoke *link* from the -ribbon; the *active object* becomes the *superior* to the other selected objects

Sometimes, it is necessary to select an object without its subordinates (i.e. to make a further object subordinate); therefor, deactivate sel. linked in the *selection-rollout*.

to unlink objects, invoke *unlink* from the -ribbon; all selected objects are unlinked from their superior (if any)

Arraying objects

to create arrays

1. select the objects to be used
2. enable copy or instance, then set the repeat-adjuster to the number of copies to be created
3. transform the copies by either moving them with two fingers or using a transform-tool
    - the first copy is transformed normally, the second one two-times, the third one three times, etc.
    - you can change the *transform-tool*, the *transform-axis* resp. -*plane*, and the reference-coord.-system, if required; that way, you can easily create complex arrangements, e.g. spiral arrays
4. if desired, continue with 2. to create copies / instances of all the copies and the originals once again
    - that way, you can easily create multi-dimensional arrays; rectangular, polar, or other very complex ones

Duplicating objects to vertices or faces

Objects can be duplicated to the locations of vertices or faces of other objects.

to copy an object to vertices or faces,

1. first select the vertices resp. faces to be used as the target locations,
2. last select a vertex resp. face on the object to be duplicated so it is the *active element*
3. invoke *replic.* from the -ribbon; instances are created
    - for vertices, they are placed with the *active vertex* to the locations of the other selected vertices, with the orientation and scale of the original object
    - for faces, they are placed with the center of the *active face* to the centers of the other selected faces and aligned to them; the scale is not changed

    the duplicates are linked to the objects on whose vertices resp. faces they are placed automatically. Moreover, they are selected so they can be further transformed at once

    - to rotate or scale the duplicates without changing their positions, enable each

Editing polygon meshes

Spacedraw provides plenty useful and established tools for polygonal modeling and retopology.

- they work with arbitrary complex topology: manifold and non-manifold meshes, faces with any number of sides, edges shared by multiple faces.
- the operations can be applied to any number of elements at once, possibly from different objects.
- *normals, vertex-colors* and texture-coordinates are retained, resp. reasonable modified or generated

Most editing-tools and -commands are found in the -ribbon, depending on the *selection-mode*. For *manual tessellation* and for *slicing*, the line-tool from the -ribbonn is used, *extrusions* are created by moving (or scaling or rotating) faces or edged with extrude as *accompanying operation*, *target-weld* is done by *moving* with vertex- or an edge-snap enabled.

Using the commands

The available commands in the -ribbon and their effects depend on the selection. They are used as follows:

1. switch to the appropriate *selection-mode* and select *elements*
    - the applicable commands are now available
2. invoke the command
    - it is executed immediately on the current selection, and you can continue with another task if you are content with the result, or undo the changes
3. change the settings, if any
    - the effect is updated immediately
    - some options are available depending on other options
    - the chosen settings are used when the command is executed the next time
4. to continue, either
    - select *elements* for the next task; the previous selection is cleared
    - tap a selection-mode to use the “output-selection” of the command

Some commands are applied to every ***contiguous selection*** individually; *contiguous* means

- vertices connected by edges
- edges connected by vertices
- faces connected by edges

Removing and combining elements, simplifying meshes

To remove or combine *elements*, the *delete*-, *collapse*- and *weld*-commands can be used; their effect depends on the selection:

- *delete*:
    - *faces* are deleted
    - *edges* are removed when shared by 2 faces
    - *vertices* are removed and the connected edges removed or merged
- *collapse*: for every contiguous selection, all contained vertices are merged to the center
- *weld*:
    - *vertices* are all merged to one at the center, possibly removing connecting edges and collapsing or dividing faces
    - *edges*: when two edges of a face are selected, they are welded together, possibly collapsing or dividing the face; when two distinct edge-paths are selected, these are welded, see Welding and bridging edges and faces
    - *faces*: when two distinct *face-clusters* are selected, these are welded, see Welding and bridging edges and faces

There are many operations that simplify meshes, e.g.

- removing edges, merging the adjacent faces
- removing vertices, merging all adjacent faces
- collapsing faces, accordingly enlarging the surrounding faces
- collapsing edges, reshaping or collapsing the surrounding faces
- welding vertices that share an adjoining edge or face, thereby collapsing or merging edges, and reshaping, collapsing or dividing faces

- welding edges that share an adjoining face, thereby reshaping or collapsing faces

Vertices and edges may either be welded to the center, or one vertex resp. edge may be moved and welded to another, called ***target-weld***.

The same results can often be achieved in several ways, by applying different commands to different selections; in the following, preferably easy ways are described:

to merge adjacent faces, select the separating edges or shared vertices, then invoke *delete*

to weld adjacent vertices to their center, select the connecting edges or faces, then invoke *collapse*

to weld a vertex on another vertex or edge (*target-weld*), move it to the destination with vertex-, edge- or midpoint-snap enabled

to weld vicinal edges to their center, select them, then invoke *weld*

to weld an edge on another edge (*target-weld*), move it to the destination with an edge-snap enabled

Automatic stitching and simplifying

The auto-weld-tool automatically welds proximal vertices, thereby collapsing small faces, or welding unconnected elements.

It can be used to automatically

- merge separate but coincident vertices
  - therefor, make sure unconn. is enabled
- simplify meshes,
- stitch together surfaces where they are closest, either along borders, or at arbitrary positions

Which vertices are welded depends on the settings:

- connected: vertices connected by edges or faces may be welded
  - disable this option to avoid collapsing boundary elements when used to stitch together surfaces
- unconn.: vertices not connected by edges or faces may be welded
  - disable this option to avoid nearby unconnected parts from being welded when used to simplify meshes
- unsel.: nearby unselected vertices may be welded
  - using this option, you have to select less for simplifying, or you possibly only have to select one side of a seam
- real-time: the effect is continually updated while adjusting the distance
  - for large selections, this may be slow
- distance: vertices being nearer may be welded

Only vertices belonging to the same object are welded; to weld distinct object together, first use *combine*.

Welding and bridging edges and faces

The weld-tool welds distinct edge-paths or surface-areas; bride creates “bridges” between them. They work quite analogously.

to weld together two edge-paths, or create a “bridge” between them, select two distinct edge-paths, then invoke weld or bridge

- you may use border- and non-border edges and separate lines or curves

        - splines and arcs are converted to polylines; use *cover* to retain them and create *patches*
- the paths may belong to different objects; if so, the objects are combined automatically
- multiple pairs of edge-paths can be welded / bridged at once; if more than two distinct edge-paths are selected, Spacedraw welds / bridges the nearest pairs
- the paths may have a different edge-count; if so, vertices are inserted in the shorter paths edges
- which ends of the edge-paths are connected is determined automatically, but can be changed with reverse
- the edge-paths may be open or closed but not branched; if one or both paths are closed, the edges are connected the straightest possible way automatically, but the twist can also be adjusted

to weld together two surface-areas, or create a “bridge” between them, select two distinct *face-clusters*, then invoke weld or bridge

- the *face-clusters* may belong to different objects; if so, the objects are combined automatically
- multiple pairs of *face-clusters* can be welded / bridged at once; if more than two distinct *face-clusters* are selected, Spacedraw welds / bridges the nearest pairs
- the “twist” is determined automatically to minimize distortion, but can be adjusted with twist
- if the *face-clusters* are oriented very slant to each other, the “winding” of the bridge may not be clear; it can be changed with reverse
- the faces of the selected *clusters* are deleted

Adding elements, refining meshes

to extend a surface from border-edges, or to append faces to interior edges, enable extrude in the ↗-- ribbon and then move (or scale or rotate) the duplicated edges, see Accompanying operations

to extrude faces or *face-clusters*, enable extrude and then move (or scale or rotate) the faces

to fill gaps with a faces, select the border-edges and invoke *cover*

- for a single gap, it is enough to select two adjoining border-edges

to regularly subdivide (tessellate) faces, use the ◈, ⊞and ⊠-commands; they work on each selected face individually:

- ◈, called ***diamond tessellation***, connects the midpoints of adjacent edges
- ⊞, called ***quad tessellation***, inserts a vertex in the face-center and connects it with all edge-midpoints
- ⊠, called ***tri tessellation***, inserts a vertex in the face-center and connects it with all vertices

to insert “edge loops”, select “paths” of faces (using e.g. *loop* or path), and invoke *edge-loop*

- the paths may be arbitrary shaped and branched and can contain faces with any number of sides, but must contain at least two faces
- the edges are inserted at center by default, but can be shifted to one side using shift

to surround edges or vertices with edges, use surround with chamfer and inflate disabled, see Enclosing and chamfering edges and vertices

Detaching and dissecting

to dissect a surface along edge paths, invoke *split* from the -ribbon; you can now move the elements on either side independently

- the mesh is only split at vertices where at least two selected edges meet, except for border-vertices
- the paths may be arbitrary shaped and branched
- if the path is closed or self-intersects, the isolated parts can now be selected with *connected* from the *selection-rollout* and moved apart

to split a shared vertex, invoke *split*; for each connected face, a new vertex is created, and you can now successively select and move them apart

- for each selected vertex, one of the new vertices is selected automatically

to detach a part from a mesh, select the respective faces, then either

- invoke *split* from the -ribbon; you can now move away the selected faces
- use detach from the -ribbon as *accompanying operation* for a transformation
    - enable new obj. beforehand to create a new object from the detached faces

Subdividing and cutting by drawing edges

To subdivide or cut meshes, you can “draw” edges on a surface with the line-tool, using snaps; in one step, you can

- split a single face with an edge, connecting a vertex or edge with another vertex or edge, or
- connect two arbitrary distant points on a surface with an “edge-path”; it may either
    - straightly “slice” through faces and other edges, or
    - snap to vertices or edge-midpoints, possibly following existing edges
- “slice” straight through an entire connected part of a mesh with edges.

The line-tool only inserts edges; to split the mesh along the drawn path,

1. switch to the -ribbon thereupon and enable edge-selection mode; the edges of the path are selected automatically
2. invoke *split*

to freely “slice” faces with edges,

1. activate the line-tool from the -ribbon and enable the suitable snaps, at least one out of , and , then enable combine, disable snap path
    - to use grid-snap or ortho-snap, orient the *reference coord.-system* to the face to slice by using ref. CS with a face-snap enabled, see Snapping and aligning
2. snap the *tripod* to a vertex or edge, then tap
3. move the *tripod* to another vertex or edge or a point within a face (with a *face-snap* enabled); the path is highlighted
4. release the touch; the edges are created
5. either
    - continue with 3. to continue the path from the current position, or
        - you may change *snaps*, or enable snap path
    - tap and move the *tripod* to a new start point

to add edges, following a “path”,

1. activate the line-tool from the -ribbon and enable - or -snap or both, then enable combine, then snap path
2. snap the *tripod* to a vertex or edge-midpoint, then tap
3. move the *tripod* so it snaps to another vertex or edge-midpoint; the path is highlighted
    - if only -snap is enabled, the path only uses edge-midpoints; if only -snap is enabled, it uses only vertices
4. release the touch; the edges are created
5. either
    - continue with 3. to continue the path from the current position, or
        - you may change *snaps*, or disable snap path
    - tap and move the *tripod* to a new start point

to slice a connected part of a mesh along a plane,

1. activate the line-tool from the -ribbon and enable the suitable snaps, at least one out of , and , then enable combine, then across
2. orient the view perpendicular to the "*slice plane*",
3. snap the *tripod* to a vertex or edge in the *slice plane,* then tap
4. move the *tripod* to another point in the *slice plane* within the surface; a line indicates the slice-direction
    - a face-snap must be enable to use a point within a face
5. release the touch; the part connected to the first point is sliced

Generating and modifying surfaces and solids

Spacedraw comes with sophisticated tools for deriving surfaces and solids from lines or other surfaces or solids.

The inflate-, diverge- and surround-tools introduce novel ways for quickly generating contiguous manifold surfaces from existing geometry.

NEW The inflate- and diverge-tools create *modifiers*

- they update when the source changes, e.g. when the underlying splines are transformed
- the parameters can be changed at any time
    - to adjust the parameters at individual places, simply select the respective vertices, edges or faces of the source with the -ribbon open, and the applicable adjusters will be available
        - if there are different modifiers at an element, you can switch the one to adjust by tapping the respective tool (which is light blue in this case)
        - on selection, the values of the adjusters are set to ones of the *active element*, sliding the adjusters changes the values for all selected elements proportionally; to give all selected elements the same value, tap the adjuster to bring up the text-input and tap Go / Enter.
    - to adjust the overall parameters, select an element of the source, then activate the modifier-tool (it is highlighted in blue in the -ribbon)
- they can be disabled temporarily, *collapsed*, removed, and resized. The commands are available when a modifier-tool is active; if not, first activate the modifier by selecting an element of the source and then tapping the respective modifier-tool
    - to enlarge or shrink the range of a modifier,

1. activate it; the current edges (for inflate) resp. vertices (for diverge) will be selected
2. change the selection, than tap *resize* (if the new elements already have a modifier of the same type applied, the parameters are retained)

- to re-enable a modifier, select an element of the source and tap the respective modifier-tool (it is highlighted in light blue in the -ribbon)
- the elements generated by a modifier are editable only after if is collapsed

Inflating lines

The inflate-tool from the -ribbon creates surfaces around lines and curves, or the edges of meshes. “Pipes” with any number of sides, or flat “stripes” may be generated, and their thickness, NEW "squash", and rotation can be adjusted overall and in places. At branchings, the simplest possible joints are created, yielding contiguous manifold surfaces from complex networks.

The tool may be used e.g. to create pipes, cables, hoses, frames, beams, trunks and branches of trees, culms, blades of grass, snakes…

1. select the lines / edges to inflate
    - whole objects or arbitrary parts may be used
2. invoke inflate; the surface is created (according to the previous settings)
    - for large selections, this may take some time
3. adjust the overall parameters
    - joint: whether the pipes are assembled at branchings to create a single surface
        - if enabled, all branches starting at a joint have the same thickness there
        - at very complex 3-dimensional branchings, the tool may fail to create sensible junctions; in this case, disable joint
    - NEW cap: whether the ends are closed
    - smooth girth: whether the surface is *smoothed* crossways
        - enable for round pipes, disable for edged pipes
    - NEW use tris: whether triangles are used at transitions between different numbers of sides
    - NEW turn: the pipes are rotated 45°; this results in a different quashing, and creates another type of branchings with joint enabled
        - with joint only applied at branchings
    - NEW optimize: (only for joint disabled): whether the continuous "strings" are chosen such that the straightest ones are continuous at branchings
    - sides: number of sides; 2 gives flat stripes
        - with joint enabled, only even numbers are possible
    - thickness
    - NEW squash
    - rotation (only for joint disabled): relative to the first turn of each string
    - smooth angle: maximal kinking-angle for which the surface is *smoothed* lengthways
        - only available with smooth girth disabled; to smooth everything, *disable* smooth girth and set the smooth angle to 180°

4. (optional) adjust the parameters at specific places: activate the vertex- or edge-selection tool and select elements; the applicable adjusters and options will be available NEW also rotation and sides can be changed individually

to create branches with different parameters (e.g. thickness) at a joint, apply distinct modifiers to them; then you can use edge selection to adjust the values for specific branches.

NEW Lines, splines and arcs can be inflated at creation by activating inflate before starting to draw.

- if you snap to an inflated line with combine enabled, append if available to continue an inflate
    - if you wish to create a branch starting out with a different thickness, make sure append is *disabled*
- adjusters are available to change the parameters at each added vertex
- the inflate can be temporarily disabled

Diverging surfaces

The diverge-tool from the -ribbon gives thickness to existing surfaces by splitting them in two, either as a whole or in places. The displacement to each side can be adjusted and varied area by area. At borders, flat or beveled rims can be created. Closed, contiguous surfaces are generated even from complex ramified constructs.

1. select whole objects, or the faces, edges, or vertices around which the surface shall be diverged
2. invoke diverge; the surface is created (according to the previous settings), and *vertex-selection-mode* is activated automatically
    - for complex objects, this may take some time
3. adjust the overall parameters
    - cap, bevel: whether to create flat or beveled rims
    - out dist., in dist., NEW out&in: displacements of the created surfaces; use out&in to adjust both displacements at once
4. (optional) adjust the parameters at specific places: activate the vertex-, edge-, or face-selection tool and select elements; the applicable adjusters and options will be available

Enclosing and chamfering edges and vertices

The surround-tool from the -ribbon creates edges or faces around selected vertices or edges of a surface, possibly removing the enclosed surface-parts. It can be used to

- insert *edge-loops* around edge-paths or single vertices
- chamfer or round corners and edges of objects
- create veins or bulges on surfaces, or frames around them

It is used as follows:

1. select vertices or edges
2. invoke surround; the operation is executed according to the previous settings
    - the new edges are inserted at an equal distance from the selected edges or vertices, such that they are centered between other edges at the narrowest locations
3. adjust the options
    - chamfer: deletes the enclosed surface-parts and inserts plain faces
    - inflate: deletes the enclosed surface-parts and inserts four-sided “pipes” for edges, resp. octahedral for vertices, connected with the enclosing surface

   - weld: automatically welds colliding vertices
     - this occurs for large distance values
   - smooth border: whether to smooth the transitions
   - distance: overall distance of the inserted edges from the selected edges or vertices, equal to each side
   - distance selected: adjusts this distance at the selected vertices
   - out front, out back: position of the central edges / vertices for inflate
   - out front selected, out back selected: adjusts these positions at the selected vertices
     - the vertices of the original surface are "x-rayed" and can be selected using any selection-tools; the surround-tool stays active
   - smooth angle: maximal kinking-angle for which the new surface-parts are *smoothed* lengthways
4. tap another *selection-mode* or *ribbon-selector* to finish

Spline surfaces and patches

Spline surfaces are curved surfaces that are defined by spline-networks, constituting their contours. They can be formed by adjusting *control points* and *handles*, providing excellent, intuitive control about their shape:

- the surface passes exactly through the *control points*
- the surface spreads out parallel to the *handles* from the *control points*
- the *handle's* length determine the curvature at the *control points*

They are composed of two-, three- or four-sided ***patches*** (coons patches in the four- and three-sided case), bounded by splines. Also, patches bounded by any combination of splines, straight lines and arcs can be created. The *patches* consist of many planar four- or three-sided ***facets***.

Usual *polygon faces* and all kinds of *patches* of varying *facet*-count can arbitrarily be combined to contiguous surfaces. Any number of patches can meet at a vertex and at an edge, and they may converge there smoothly or angled. Four-sided *patches* can be assembled among themselves, and with three-sided *patches* and planar faces with good tangential- or curvature-continuity (for practical purposes).

Thus, *Spline surfaces* are ideally suited to create complex organic objects with smooth surfaces that also have creases and cusps, with varying richness of detail, and arbitrary topology. Patches bounded by straight lines and arcs let you easily and precisely define surfaces for architectural and mechanical applications.

Simultaneously moving control-points and rotating and scaling the adjoining handles with the m,r,s-tool, while concurrently changing the view, lets you easily create any surfaces according to your imaginings.

Spline surfaces can be created by

- drawing the curve-networks and then "covering" them with a surface,
- extruding cross-sectional curves,
- deriving them from existing polygon meshes, taking into account the *normals* to determine the curvature and locations of creases and cusps

They can arbitrarily be extended by extruding border- or interior edges to new patches.

For adding detail, *spline surfaces* or single *patches* can be converted to polygon meshes to make their *facets* editable individually, and parts of polygon meshes can be converted to *spline surfaces*.

Creation and conversion

to create *spline surfaces* form curve-networks,

1. select the lines to cover
    - whole objects or parts may be used
    - to cover a single stitch, it suffices to select two adjoining sides
2. invoke *cover* from the -ribbon; the surface is created
    - for large networks, this may take some time
    - for all circuits only containing straight lines, usual polygon-faces are created
        - they may have any number of sides
        - if they are not planar, the shape of the face is not defined
    - a patch is created for every two-, three- or four-sided circuit containing at least one spline or arc
    - to use networks of straight lines to create *Spline surfaces*, first invoke *to splines* on the lines
3. adjust the divisions

to create or extend *spline surfaces* by extruding,

1. select the lines to extrude
    - arbitrary networks of straight lines, splines and arcs may be used; straight lines are extruded to polygon faces, splines and arcs to patches
    - solitary lines, or the boundaries of existing *patches* or polygonal faces can be extruded; border-edges as well as interior edges may be used
2. enable extrude in the -ribbon, then move (or scale or rotate) the duplicated edges
    - the lateral edges are created as straight lines; they can be converted with *to splines* from the -ribbon
3. adjust the divisions

to derive *patch surfaces* form polygon meshes,

1. select the objects or faces to convert
2. invoke *to patches* from the -ribbon and adjust the number of divisions
    - the number of *facets* to be created is displayed
3. tap to generate the patch surface
    - for a large facet-count, this may take some time
    - all affected edges are converted to splines
    - faces with more than four sides are not affected
    - where patches and usual faces meet, there is a spline bounding the patch and the straight edge of the face, so that gaps may arise; to close them, either straighten the spline or create a two-sided patch in-between
    - where patches of different division-count meet, gaps occur if the interjacent spline is not straight; they cannot be closed until the patches are converted to meshes
    - faces sharing a normal at a vertex are assembled with tangential continuity there

to convert patches to editable polygon meshes, select the objects or patches to convert and invoke *to poly* from the -ribbon

- at the boundaries between *patches* and thusly converted meshes, vertices are shared among a *patch* and faces so that a gapless transition can easily be retained; these vertices can be moved manually, but move back to their position at the patch-boundary when the patch is transformed

Selecting, transforming and editing *elements*

To shape *spline surfaces,* you select patches, edges, vertices or handles with the ↗-ribbon loaded, and then either transform them, or invoke appearing commands that shape the surface in certain ways.

Selecting and transforming edges, vertices and *handles* on patches works analogously to solitary spline-networks.

- *patches*
    - are selected in face-selection-mode and can be transformed or deleted as a whole analogously to faces
    - the facets are shown but can’t be modified individually
    - to change the *facet*-count, invoke *to patches* again with the patches to change selected
- *edges*
    - are selected and transformed like the segments of solitary splines
    - additionally, loops can be selected and patch-selections can be converted to edge-selections
    - the interior edges of patches are shown but can’t be adjusted individually
    - they can be converted between straight lines and splines with *to splines* resp. *to lines*
- *vertices* and *handles*
    - work like those of solitary spline-networks
    - at vertices where four splines and patches meet, link handles links the opposite handles, and *smooth* and *symmetric* affect the transitions between opposite splines

Shaping the surface

Transitions between patches at vertices and along splines can be categorized in three types:

- *creases* / *cusps*, where the patches meet at an angle
- *smooth transitions*, where all patches share the tangent plane
    - the surface is *smooth* at *coplanar vertices*, where all handles are in the same plane
    - the transition between two patches along the connecting spline is *smooth* if the “framing” splines join *smooth*
- *symmetric transitions*, where the relevant handles additionally have the same length

<u>to make the transition between adjoining patches smooth or symmetric</u>, select them and invoke *smooth* or *symmetric*

- the transitions between the “framing” splines are made smooth or symmetric

<u>to make the surface smooth around vertices</u>, invoke *coplanar*

- this doesn’t necessarily make the transitions along intermediate splines smooth

<u>to make the surface smooth along edge-paths</u>, invoke *smooth* or *symmetric*

- you may use *loop* to select loops

<u>to make patches planar and all their edges straight</u>, invoke *straight*

<u>to create cusps at vertices</u>, adjust individual handles, or invoke *straight* to make the splines start in direction of the opposite vertices there

<u>to create creases along splines</u>, adjust the handles of the “framing” splines

- for “soft creases”, make the handles parallel by invoking *smooth* or *symmetric* with the framing splines select, then scale them to

to convert edges between straight lines and splines, invoke *to splines* resp. *to lines*

- while straightened splines yield the same surfaces as straight lines, the splines crook again automatically when the adjoining vertices are moved

Light and color

Spacedraw can produce realistic images of a scene, taking into account positions and properties of lights, material properties of surfaces as well as the relative viewing position and angle.

Together with the *perspective projection*, this makes it possible to implicitly create visual art. Rather than considering how an image should look from a specific position like a (traditional or digital) painter, you define the material colors and properties and the light sources according to your ideas, let the software calculate the image, and then possibly adjust the definitions.

Also, realistic lighting is highly important for modeling 3d-objects, since only lighting effects like specular highlights create a real 3-dimensional look.

This realistic lighting can be enabled or disabled at any time via lighting in the D-menu.

Using the tools

All the tools for lighting, shading, and painting are organized in the -ribbon as follows:

It starts with three special modes for specific tasks,

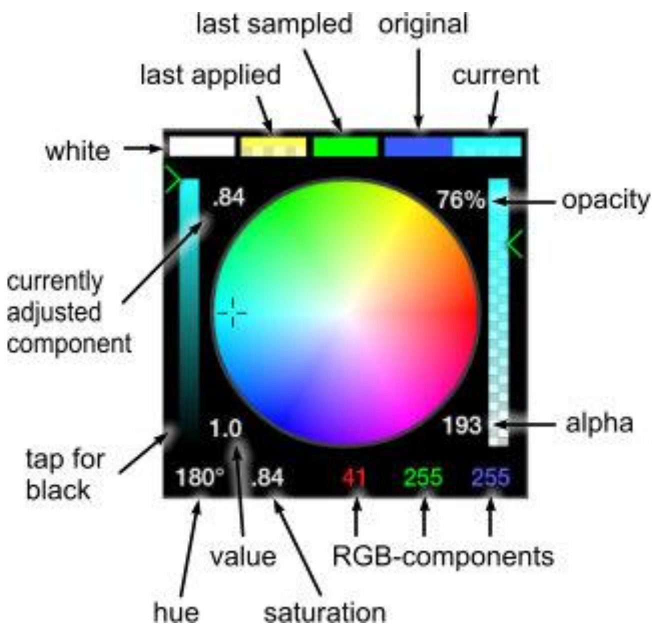
- color, for assigning vertex-colors
- material, for defining, organizing and assigning materials
- paint, for painting.

When activated, only the tools and commands for the respective task are available. For creating lights, working with normals and *UVs*, make sure color, material and paint are deactivated.

Selecting colors

With Spacedraw's novel multi-touch color picker you can move in the 3-dimensional color-space by sliding with two fingers, and that way intuitively choose any color with one grip. The average motion moves a pointer on a color-field to define the hue and saturation, *pinching* varies the brightness.

The sophisticated, yet compact interface lets you also precisely adjust colors according to the HSV- or RGB-color model, chose a transparency, and access previously used colors. It can freely be resized and moved on the screen.



Choosing colors from the spectrum

The color-picker shows two linked color-fields, where colors are arranged according to the HSV color model

- the strip on the left shows the colors of the current hue and saturation
- the central wheel shows the colors of the current value, arranged in polar coordinates

to freely move in the color-space, slide two fingers within the *tool-zone*, *outside* the color-picker; the average motion moves the crosshairs in the color-wheel , pinching moves the slider on the left color-strip.

to adjust the value (brightness), slide up/down within the left color-strip

to adjust hue and value together, slide or tap within the color-wheel to move the crosshairs

to adjust the hue, place one finger on the hue-display at the bottom-left corner and slide up/down ; the crosshairs move on a circle, and top-left the current hue is displayed

to adjust the saturation, place one finger on the saturation-display and slide up/down; the crosshairs move in/out the wheel, and top-left the current saturation is displayed

to adjust the red, green or blue component according to the RGB color model, place one finger on the respective component-display and slide up/down; the crosshairs and the slider on the left move accordingly, and top-left the current component-value is displayed

NEW to type the RGBA or RGB values, slide into the screen from right within the *type-rim*

to choose pure red, green or blue, tap the respective component-display

to choose black, tap the bottom end of the left color-strip

to choose white, tap the white color-box at the top-left corner

to fully saturate the current color, tap the hue-display

to fully desaturate the current color, tap the saturation-display

to choose a pure shade of gray, tap the saturation-display, then slide within the left color-strip

to choose the brightest color with the current hue and saturation, tap the top end of the left color-strip

Transparency

The transparency of surfaces is handled together with their diffuse color; you can use texture images with an alpha-channel, paint with transparent colors, or assign them to elements.

When transparency is applicable, the color-picker shows a transparency-slider on the right, and the value is displayed as opacity-percentage at the top and as alpha-component at the bottom.

In the color-boxes at the top, the colors are displayed without transparency in the upper halves, and overlaid a gray-white pattern in the lower halves to portray the transparency.

to adjust the transparency, slide up/down within the right strip

to make the color fully opaque, tap the top end of the right strip

to choose total transparency, tap the bottom end of the right strip

Sampling and reapplying colors

The color-boxes at the top let you compare the color you adjust with the original color, and access the previously applied and sampled colors. To choose any of these, tap the respective box.

- the “last sampled” box shows the original color of the previously selected element or color-input
- if elements with dissimilar colors are selected, the current color is set to white, the “original” box to the color of the first selected element
- if you increase the size of the color-picker, or reduce the tap size, additional color-boxes with the previously applied colors appear left of the “last applied” box

to copy the color from an element or color-box, select it so that the color-picker switches to its color; when you then select another element or color-input, the “last sampled” box shows that color.

Moving and resizing the color-picker

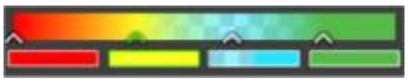
to move the color-picker, place two fingers somewhere within the color-picker and slide

to resize it, place one finger somewhere within the color-picker and another outside, somewhere within the tool-zone, then *pinch*

Color gradients

Color gradients are used to create linear or radial color ramps, define the colors for *procedural materials*, and for mapping colors of texture images.

They are defined by a series of ***stops*** between which the color is linearly interpolated. Any number of *stops* can be added and placed at arbitrary locations along the gradient.



To create and edit gradients, a neat but full-featured gradient-editor is shown in the upper part of the *material-panel*, see Creating, changing and assigning materials. It shows the gradient as a horizontal color-strip, overlaid a gray-white checker-pattern to portray the transparency. Arrows indicate the positions of the stops, beneath are color-boxes, showing the respective colors. The boxes are spaced equally, independent of the stop-positions, to make the stops accessible even if the arrows are very close.

- to edit a complex gradient, increase the width of the *material-panel*

to change the color of a stop, tap the respective box; the color-picker appears

to adjust the location of a stop, drag downward, and then sideways from the center of the respective stop-box; the related arrow will move.

- you can drag a stop over other ones to change their order

To insert a new stop between two existing stops, drag down from the gradient-stripe between the respective stop-boxes; a stop is added with the average color of the surrounding stops centrally between them.

- after the stop appears, you can drag sideways to shift it

to remove a stop, drag upward from the stop-box

to create an abrupt transition, move two stops directly together

Lighting

For realistically illuminating the scene, there are two options:

- lights with specific properties can be placed at fixed locations
    - up to eight lights can be used
    - by default, there is one point-light at position (0,15,0)
- a *directional light,* always following the virtual camera, together with a mild ambient light can be used
    - the *directional light* shines towards the view-center from a little above, and slightly left, of the viewpoint

Whether the lights placed in the scene or the camera-lights are used can be changed with lights: default / scene in the D-menu. The lights are depicted as yellow wireframe objects, formed according to their type; these can be shown or hidden with show lights in the D-menu.

For placing, orienting and adjusting lights, so that all objects look as you imagine from different positions, Spacedraw lets you simultaneously move, rotate and regulate lights, and concurrently change the view.

Types and properties

Spacedraw has three kinds of lights:

- ***directional lights***, casting parallel light with the same direction and intensity everywhere, as the sun does (for all practical purposes) at the surface of the earth
- ***point lights***, radiating light in all directions from a point, attenuating with the distance, as free light bulbs, candles, fires or stars do in a sufficient distance
- ***spot lights***, casting focused, cone shaped beams of light, attenuating with the distance and with the angle relative to the spot's target vector, like a flashlights or headlights
    - their direction can be changed by rotating the light, as well as by moving a special *target* object towards which the light orients itself

Lights have the following properties

- color: the color of the emission
- intensity: the intensity of the radiation coming directly from the light source when scattered on a surface
    - for *point*- and *spot-lights*, attenuates with the distance according to the linear decay and quadratic decay
- ambient intensity: the intensity of the radiation from the light that has been randomly scattered by the environment so that its comes evenly from all directions
    - for *point*- and *spot-lights*, attenuates with the distance according to the linear decay and quadratic decay
- linear decay, quadratic decay: determine how fast the intensity attenuates with the distance, resp. the square of the distance, from the source

    - real light from point sources attenuates with the square of the distance; therefor, set linear decay to zero
- cone angle: the opening angle of the light-cone, also depicted on the light-object
- cone falloff: determines how fast the intensity attenuates from the center of the cone towards its border
    - zero means a uniform distribution

Creating and adjusting lights

to create a light,

1. load the -ribbon, make sure color, material and paint are deactivated, and enable lights, then select directional, point or spot
    - show lights and lighting is enabled, and lights: default / scene is set to scene automatically
2. move the *tripod* to the location of the light an tap; the light is created
    - for directional lights, the location has no influence to the lighting
3. depending on the type, a transform-tool is activated automatically
    - point: the move-tool is activated and you can adjust the position
    - directional: the rotate-tool is activated and you can orient the radiation
    - spot: the *target*, initially positioned at the lights location, is selected and the move-tool activated; you can now define the beam-direction by moving the target
4. adjust the properties, tap to finish
    - the properties can be adjusted before or after the orientation, or alternately; concurrently, you can change the view using multi-touch or by tilting the device with *tilt-move* enabled

to adjust the position, direction and intensity of lights,

1. enable *object-selection-mode* and select the light
    - multiple lights, or lights together with objects can be transformed
2. move the light with two fingers, or use the move-, rotate- or m,r,s-tool from the -ribbon
    - scaling lights only scales the objects representing them has no influence on the lighting effect
    - with one light selected, the m,r,s-tool changes its intensity instead of scaling
    - *spot lights* move without their target and automatically orient their direction towards the target; to preserve their direction, and move the target together with the light, select both
    - to orient a *spot light*, either
        - move the target; the light orients itself automatically
        - rotate the light; the target moves accordingly
    - to see the effects, it is essential to change the view concurrently

to view and adjust the properties of a light, load the -ribbon, enable *object-selection-mode* and select the light; the properties are displayed and can be adjusted

- the properties always belong to the light which is the *active element*

Materials

The visual characteristics of surfaces, i.e. the colors, shininess and transparency, are specified by assigning *materials* to them. Together with the *normals,* they determine how light is scattered.

If lighting is enabled, the appearance of a surface at a specific point is influenced by the material properties at that point, the relative position, orientation and properties of lights, and the view-point and -direction.

To define the *diffuse* (usual) color, you can

- assign colors to faces, edges or vertices, possibly creating gradients
- map any 2d-images onto the surface
- directly paint on the surface.

Surface properties

Materials have the following properties:

- ***diffuse color**,* determines the light that is scattered on the surface evenly in all directions
    - if lighting is disabled, this color is used unaltered
    - a single color can be used, or it can vary over the surface where the material is applied:
        - different colors can be assigned to individual faces, edges or vertices
        - a texture image can be projected on the surface
        - the surface can be painted
    - the effect of this color on the surface-appearance depends on the relative angle, intensity and color of the incident light, but not on the view-direction; however, if the camera-light is enable it changes when you change the view, because the light is rotated together with the view
- ***specular color*** determines the mirror-like reflected light; it is the color of highlights on a shiny surface
    - the brightness of this color defines the intensity of the highlights at their center; use white for a high-gloss surface, black for a matte or rough one
    - the effect of this color on the surface-appearance depends on the lighting as well as on the view-direction
- ***shininess***, defines how focused the mirror-like reflection is; it influences the size of the specular highlights
    - if this value is too high, you will barely see highlights, if it is too low, the highlights overspread the whole surface
- ***emission***, light emitted from the surface itself, independently from lights
    - the emission is not used to light other objects
- ***transparency***
    - it is assigned together with the *diffuse color,* and can equally vary over the surface; textures images with an alpha channel can be used, and you can “paint” transparency, see Color and transparency

Creating, changing and assigning materials

Every object has a ***base-material***, which can be overridden by another material for individual faces. All instances of an object share the *materials* and *vertex-colors*.

When a new scene is created, there is one ***standard-material***, used for new objects by default; It can’t be renamed or deleted, but any properties can be changed (by default it is gray and shiny). When you assign a material from the library or a procedural one, a new instance is created that can be changed and added to the scene-materials.

To create, modify, assign and organize materials, the *material-panel* is used; it is structured and used as follows:

- to show or hide the *material-panel,* tap material in the -ribbon
    - when the *material-panel* is opened the first time after starting or restarting Spacedraw, the *material-library* is loaded, which may take a few seconds
    - the lasso tool is activated automatically (if paint-select is not active)
- in the top part, the properties of the selected material are displayed and can be adjusted
    - **Diffuse**, the basic *diffuse color* and *transparency* of the material; used wherever no other color is assigned to elements, if the material has no texture
        - in the right half of the color-strip, the pure *diffuse color* is displayed, in the left it is overlaid a gray-white pattern with the materials transparency
        - to change it, tap; the color-picker is shown
    - **Spec.**, the *specular color* and *shininess*, constant over the surface where the material is applied
        - shown as a strip of the *specular color* whose with depicts the shininess; a small strip means small highlights, a broad one large highlights
        - to change it, tap; the color-picker and an adjuster for the shininess is shown
    - **Emiss.**, the *emission*, constant over the surface where the material is applied
        - to change it, tap; the color-picker is shown
    - **Texture-image...**, see Using images
- in the bottom part, all materials are listed and can be assigned
    - the list is structured in groups that you can expand or collapse by tapping the title-bars, as follows:
        - first come the ***scene-materials***, that are, or were, used in the scene
        - then the *procedural* ones
        - last is the *material-library,* organized in expandable categories
    - an image and the name is displayed for each entry; for materials without a texture, the diffuse color is used
    - tap the image to assign the material to the current selection; tap right thereof to select the material and show its properties
    - if objects or faces are selected and all have the same material applied, its name is displayed in blue in the list
- the commands *new*, *rename*, *delete*, *save*, and *using* refer to the selected material, but are located in the *ribbon* rather than the *material-panel,* and only shown if applicable to the selected material

to create a material,

- activate material in the -ribbon; the *material-panel* opens
- select the material whose properties shall be used for the new one, than invoke *new*
- adjust the properties;
    - invoke *rename* to name it; if you assign a texture image, the material is automatically named accordingly

to delete a material, select it and invoke *delete*; only unused materials can be deleted

to delete all unused materials, invoke *del. unused*

to change a material, select is and change the properties; invoke *rename* to rename it

to assign a material,

- activate material in the -ribbon; the *material-panel* opens
    - to use a *procedural-* or *library-material*, expand the respective group (and category) in the list by tapping its title-bar
- select the objects or faces to apply the material to
    - faces on multiple objects can be selected together
- in the list, tap the image of the desired material
    - in case of a *procedural-* or *library-material*, a new "instance" is created and placed and selected in the upper selection of the list
    - if you assign a textured material, and the selection doesn't has a single mapped material applied, the planar *projection tool* is activated automatically, but you can change the mapping method by activating another *projection tool*

to select all objects to which a material is applied as *base-material*, invoke *using* with the material selected in *object-selection-mode*

to select all faces to which a material is assigned, invoke *using* with the material selected in *face-selection-mode*

- if no face is previously selected, the faces on all objects to which the material is assigned are selected; else, only the faces on the object of the *active* face are selected

Managing the material-library

Spacedraw comes with a material-library of over 100 mostly textured materials for you to use, sorted in 10 categories for various applications. You can simply assign these to your objects, or add your own materials and textures to the library.

When installing Spacedraw, the library is stored on your device in “/spacedraw/materials”. The material definitions are stored in standard-compliant Wavefront .mtl-files, as they are used for .obj-files, one for each category. The associated texture images are stored in equally-named subfolders, named according to the materials. You may create or edit the .mtl-files with a text editor or use the following methods.

to add a material to the library, either

- create the material in the *material-panel* using a *library-material* from the respective category as template (i.e., select that material and invoke *new)*, adjust the properties, then invoke *save* in the *ribbon*
    - you may assign a texture-image from any folder; when you save the material to the library, the image is copied to the respective library-folder
- copy the image to use as texture in the respective subfolder of the “/spacedraw/materials”-folder, using any file-manager; when you start or restart Spacedraw the next time and open the *material-panel,* an equally-named material will appear in the respective category and can be customized

to remove a material from the library, either

- select it in the material-panel and invoke *delete*
- open the respective .mtl-file with a text-editor and remove the corresponding part

to create a category, create an accordingly named subfolder in the “/spacedraw/materials”-folder; when you start or restart Spacedraw the next time and open the *material-panel,* the new category will show up

to delete a category, delete the respective folder and .mtl-file

to change a library-material, either

- create an instance by assigning it to a selection, change the properties of the instance, then invoke *save*

- select it in the library-group of the list and change the properties; the definition in the library is updated immediately

to restore all materials that are bundled with Spacedraw, delete the “/materials”-subfolder in the “/spacedraw” folder, then start or restart Spacedraw; the library will be created as when the program is installed

Curved and flat faces, smooth and hard edges

To specify

- where a surface shall appear flat and where curved, even though it is approximated by flat faces,
- along which edges and at which vertices creases resp. cusps shall appear, and which shall appear smooth,

normals are assigned to the faces at the vertices; if the transition between two faces shall be smooth at a vertex, the same normal is assigned to both faces at that vertex, else a separate one is assigned to each face.

For lighting calculations, the *normals* are then oriented automatically and interpolated across the faces; *normals* belonging to a single face are oriented perpendicular to that face, *normals* shared between faces are oriented in the average normal direction.

The *normals* are also used to determine the curvature and the locations of creases and cusps when polygon meshes are converted to *spline-surfaces*.

The *normals* can be displayed as lines emerging from the vertices; therefor, enable show normals in the D-menu.

When primitives or derivative surfaces are created, Spacedraw automatically generates reasonable normals, and many .obj-files also contain normal definitions (or *smoothing groups*, from which *normals* are derived), but you can change them at any time. They are not assigned manually, but simply by marking edges or vertices as ***smooth*** or ***hard***; the faces will appear curved between smooth elements and flat between hard elements. Whole objects, face selections, or individual edges and vertices can be marked.

To edit *normals*, switch to the -ribbon, make sure color, material and paint are deactivated, then enable normals

- to mark an edge or vertex as hard, invoke *harden*
- to mark an edge or vertex as smooth, invoke *smooth*
- to automatically mark transitions between faces as hard or smooth, depending on their angle,
    1. select the objects or faces to be affected
    2. invoke *auto-smooth*
    3. adjust the threshold angle
- to unify the direction of the *normals* on connected parts, invoke *conform*
- to flip *normals*, invoke *flip*

Assigning colors and transparency to *elements*

Colors can be assigned to whole objects, faces, edges or vertices, possibly creating gradients. This way, objects can be quickly and precisely colored matching the geometry, using the same selection-tools as for modeling. This method is especially suitable for technical applications, and for coloring highly detailed models.

The thusly assigned colors are called ***vertex-colors***. They override the basic *diffuse color* and *transparency* of the material, or are blended with the texture for textured materials.

to assign colors to *elements*,

1. activate color in the -ribbon; the *color-picker* is shown
2. enable the desired *selection-mode* and select *elements* using any selection-methods
    - if all selected elements have the same color assigned, the color-picker switches to that color, else to white
    - when using *lasso- or paint-selection*, the color-picker is hidden and deactivated; tap color again to show it
    - elements on multiple objects can be selected and colored together
3. adjust the color
    - for objects, all faces are colored
    - selected faces obtain the color overall, unselected faces are not affected, so sharp transitions arise
    - for edges and vertices, gradients are created between differently colored elements; all adjacent faces are affected
        - for faces with more than three edges, the exact color-distribution of the gradients depends on the triangulation and is not predictable

to clear assigned colors, select the elements to clear and invoke *clear*

Textures

With textures, the detail and realism of surfaces can easily and efficiently be enhanced without increasing their geometric complexity. Using transparency, you can also add arbitrary 2-dimensional geometry, and effectively approximate flat-shaped objects like grilles, fences, blades of grass or leaves.

Spacedraw lets you easily and precisely map your own photos or other images to surfaces, or you can use any of the over 100 provided textures in the *material-library*, or quickly generate regular or random patterns procedurally.

Using images

To assign an image file to a material to define the diffuse ("usual") color of surfaces, select the material in the *material-panel*, then tap "Texture image…" in the upper part of the panel; the *file-browser* opens and you can select an image file stored on your device

- image files in various formats like .jpg, .png or .gif of theoretically any dimension and aspect ratio can be used; however, internally they are converted to textures (bitmaps) with both width and height a power of 2

Use the *size-slider* to define the size of the texture in 3d-space-units; the image is scaled wherever the material is assigned, and the size will be used when creating mappings.

Texture images can be repeated any number of times to cover a large surface with a small image; e.g. the photo of a small part of a brick wall may be used to create a realistic, detailed looking wall with thousands of bricks. Therefor, the texture image has to be cropped and possibly manipulated such that the seams are not or barely recognizable, called seamless or tileable. Almost all of the images in the provided material-library are well tileable, and Spacedraw includes a tool to quickly crop your own images and preview the tiled appearance, see below.

Defining the color of surfaces

Image files can be used unaltered to define the diffuse color of a material, colorized with the single color set as the materials diffuse color, or with the assigned vertex-colors. To retain the original color, make sure the diffuse color is set to pure white.

Adjusting and mapping colors

Image files can be used unaltered to define the diffuse color of a material, or the colors may be adjusted or mapped to other colors, including transparency. The original file is not changed, and the same file can be used for many materials with different adjustments or mappings.

to adjust or map the colors of a texture image for a material, select the material, then activate adj. color or grad. map in the ribbon

- with adj. color, the contrast, saturation and brightness can be adjusted, and the hue can be shifted according to the HSV-color model with rotate hue
- grad. map maps the brightness of an images pixels to the colors of a gradient. When grad. map is enabled, the gradient-editor is shown in the materials properties and lets you define an arbitrary gradient, possibly with full or partial transparent colors. contrast adjusts the mapping curve.

Cropping and resizing

Spacedraw includes a tool to crop and resize images, specifically designed to quickly create small and well-tileable texture images:

to crop or resize images, select a material with the respective texture image in the *material-panel* and invoke crop; a window divided in three columns is shown; it is used as follows:

- the right column lists the files and subfolders of the image's directory
    - it can be zoomed like any panel by pinching within the column
    - when zoomed in, the size, dimensions and the time last modified are shown
    - at the top, the path is shown; tap there to switch to the parent folder
    - the subfolders are displayed in blue at the top of the list; tap one to open it
    - tap an image file to load it in the editor; the changes on the currently loaded image are *not* saved automatically; tap the same image again to undo all changes and reset the zoom of the central column
- the central column shows the original image and is used to crop it
    - slide one finger up/down in the upper resp. lower part, or left/right in the left resp. right quarter of the column to crop the image
    - pinch to zoom
- the left panel shows a tiled preview of the cropped and resized image and is used resize it (change the resolution)
    - the image its repeated three times in both directions to let you preview the resulting appearance
    - slide one finger diagonal within the column to resize the image
- place two fingers in different (adjacent) columns, then slide sideways, to change their distribution
- at the top, the original and new dimensions (in pixels), and the scale factor are shown
- tap 100% to reset the scale, or reset to also undo the cropping
- save saves the adjusted images, overwriting the original; save copy saves the adjusted image with a new name (appending a numbering to the file name)

Generating procedural textures

The tools for generating procedural textures are found in the "Procedural" group of the *material-panel's* list.

to generate a procedural texture,

1. open the *material-panel* and expand the "Procedural"-group in the material-list
2. select objects or faces to initially apply the material to, then tap the image of the respective entry in the list; a new texture is created with the standard-settings and placed in the upper part of the list.
3. adjust the parameters and colors
    - the texture is updated continuously in the viewport

to change a procedural texture, select the respective material and adjust the parameters

Gradient

gradient generates repeating linear or radial (circular or elliptical) gradients, creating striped or spotted patterns. Using full transparency for the inner (first) part of the gradient, grids with circular holes can be created.

- radial: whether to create a linear or radial gradient
- mirror: to alternate the gradient with its mirrored version
- width: the width of one iteration
- height: for radial gradients; use the same value as for with for circular, a different for elliptical gradients

Tiles

tiles generates various two- or three-colored tile- brickwork- or checker-patterns. Using full transparency for the tiles, grids can be created.

- shift: to shift every other row; if enabled, the shift can be set with the equally named adjuster
- checker: to alternate the tile-colors
    - with shift disabled, creates a checkerboard-pattern
    - with shift enabled, alternates the colors of the rows
- with, height: the distances between columns and rows
- horiz. gap, vert. gap: the interstices between the tiles; use 0 for a simple checkerboard-pattern

Clouds and turbulence

clouds and turbulence generate random cloudy, turbulent, or veined marble-like fractal patterns, using the Perlin-noise algorithm. By tuning the various parameters and using color-gradients with multiple stops, diverse, singular structures with an amazing richness of forms can be created. Using partially transparent colors in the gradient, smoke and clouds may be produced.

- seed: creates a new random, singular pattern using the current settings
- veins (only for turbulence): whether to create swirled veins or undirected turbulence
- size: the overall scale of the fractal components
- frequency: the number of fractal iterations; influences the level of detail
- turbulence: how much turbulence is added; with veins enabled, controls how much they are swirled
- contrast: adjusts the mapping of the color-gradient

Texture mapping

To specify how the flat texture images shall be mapped onto surfaces, ***texture-coordinates*** (points on the images), also called ***UVs***, are assigned to the surfaces' vertices. For every face, the corresponding part of the image is then spanned between the vertices accordingly.

Texture images can be repeated multiple times to cover a surface, and the seams may arbitrarily cross the faces. Therefor, *texture coordinates* outside the range of the images are assigned.

When primitives or derivative surfaces are created, Spacedraw automatically generates reasonable UVs, and many .obj-files also contain them. If you assign a textured material to an object or face-selection that already has UVs, and where a single or no material is previously assigned, the existing UVs are used and only scaled to match the new materials scale. Otherwise, the planar- (for a face-selection), or the box-projection tool (for an object-selection) is activated automatically.

Creating texture-maps

One reasonable and easy way to map textures is to project them onto the surface, either straight from a plane (called *planar mapping*), cylindrical or spherical. Since *planar mapping* roughly reverses the process of photographing, it can produce good results for photo-textures.

Spacedraw provides the following tools for established projection methods

- planar: projects the image from a single plane straight against the surface
- box: projects the image from the six sides of a box; each face is mapped from the box side whose normal parallels its own the most
- cylindr.: projects the image from a cylinder, wrapping it around the object
- spheric.: projects the image "from a sphere", by associating the spherical coordinates of the surfaces points with the Cartesian image-coordinates

These tools show appropriate shaped ***gizmos*** depicting the surfaces from which the image is projected; they can freely be moved and rotated using the standard transformation tools or positioned and aligned automatically, relative to the selection, the view or the global coordinates.

to create a projection map,

1. switch to the -ribbon and activate material; the *material-panel* opens
2. select the objects or faces to be mapped and select a textured material to be used
    - multiple objects, or faces on multiple objects can be mapped together
    - if the selection already has a textured material applied and it shall be used, make sure it is selected in the material-panels list
    - if the selection doesn't have a single mapped material applied and you assign a textured mapping, the planar projection tool is activated automatically, but you can change the mapping method by activating the respective tool
3. activate a mapping-tool; a gizmo, fitted to the selection, is shown and the projection performed accordingly
4. adjust the projection, using the appeared tools, commands and adjusters
    - move, rotate: the standard transformation-tools
    - *pos. sel.*, *ori. sel.*, *ori. obj.*, *ori. world*, *ori. view* work like the reference-coord.-system-commands , see Transform-center and -orientation
    - scale, scale width, scale height: let you adjust the scale uniformly, or in the x- and y-direction of the image individually
        - use the *size-slider* in the *material-panel* to scale the mapping wherever the material is and will be assigned
        - for the box projection, the gizmo displays three dots on its edges instead; tap one, and then slide within the tool-zone to scale the projection along the respective axis; to uniform scale the projection, use the white dot on the corner of the gizmo.
    - fit: stretches the image to fit on the selection, resulting in the "best" unique mapping (with exception of box; here, the image is used 6 times)
    - turn, flip, mirror: rotate the image by 90°, flip it upside-down, resp. mirror it

    - use snapping to snap the gizmos to elements and possibly align them
5. tap to finish

For uneven surfaces, the projections always result in distorted and unequally scaled mappings from the images' sections to the faces. To avoid this, the image can be mapped to each face individually aligned, or the map can be started at one face, and then straightly wrapped to neighboring faces until the surface is covered, using the following tools:

- per face: maps the image to each face individually, equally scaled, aligned with one edge, and undistorted for planar faces; seams arise along every edge
- wrap: for every connected part of the selection, maps the image to one face, then wraps it equally scaled and (for planar faces) undistorted to neighboring faces until all selected faces are covered, such that seams arise only where necessary

They are used like the *projection-tools*; the shift, scale and rotation can be changed using the respective adjusters.

Transforming and tweaking UVs

Spacedraw lets you easily and intuitively adjust texture maps by transforming the images on the surfaces, directly in the 3d-viewpot. Aided by preview-highlight, individual UVs can easily be selected and then moved to tweak the mapping, or entire UV-connected parts can be selected and transformed

to transform or tweak UVs, switch to the -ribbon and enable UVs; the applicable selection- and transform-tools are now available in the ribbon, and the UVs are displayed as dots on the respective vertex positions; for UVs not shared by all adjoining faces of a vertex, the dots are shifted towards the respective faces. The tools are used as follows:

- single UVs can be selected in vertex-selection-mode
- in edge-selection-mode, the UVs of the adjoining faces for the edges' endpoints are selected; if they're not shared between the faces, only the UVs of the face under the *picker* are selected
- in face-selection-mode, all UVs of a face are selected
- *connected* selects the geometrically connected parts, *UV-conn.* selects via between faces shared UVs connected parts
- *border* selects UVs along borders of UV-connected parts
- *move, rotate* and *scale* transform the mapping along the surface
    - for a single selected UV, rotate and scale transform the edge-connected UVs
- snapping can be used to exactly position specific points on the image at elements of the mesh
    - use pivot to position the tripod on the image
    - texture in the snapping-group of the ribbon snaps to the corners of the images, displayed as yellow dots

Splitting and welding UVs

to split or weld UVs, switch to the -ribbon, make sure color, material and paint are deactivated, then enable texture. Enable show UVs in the D-menu to see the UVs, and show texture corners if desired. The available commands and their effects depend on the *selection-mode*:

- *weld UVs*: merges the UVs of the selected vertices (or of all vertices of the selected edges or faces) to their average (if the respective faces have the same material applied)
- *split UVs*: assigns unique UVs to the faces at shared vertices, so that the texture image can be moved in them independently
    - for vertices, assigns a unique UV to each adjoining face
    - for edges-paths, splits the UVs such that faces on different sides of the path don't share UVs; requires at least two adjoining edges to be selected

    - for faces, splits the UVs for all selected *face-clusters* along their border-edges
- *auto-weld UVs*: automatically welds UVs that are very close together in *UV-space* (if the respective faces have the same material applied)

Painting

For coloring surfaces, Spacedraw provides a wide range of sophisticated digital painting tools, rivaling and partially exceeding those found in leading 3d graphics software:

- highly customizable brushes with custom tip images
- many *blending modes*
- painting with transparency
- applying filters/effects to selections, or by painting
- stamping decals / patterns
- projecting textures by painting
- smudge tool
- clone tool
- erasing to history
- magic-wand- and face-selection

Preparing to paint

Before a surface can be painted, a (possibly blank) texture image must be mapped to store the color information. The possible precision of the paintings depends on the resolution of the image.

You can paint on a surface wherever an image is mapped; however, if the image is *tiled*, or the same part used on multiple locations of an object, your painting will appear repeatedly as well. To avoid this, every part of the surface must be mapped to a unique part of an image.

Spacedraw provides a tool to create unique texture images and maps, possibly from existing tiled and repeated mappings, retaining the surface's appearance. This is especially useful when part of a large surface, covered with a tiled small texture image, shall be customized by painting. It works as follows:

For every *UV-connected* part of the selection, a new image will be created and mapped, using the existing mapping-coordinates. The dimensions are chosen such that an adjustable minimum precision in pixels per unit (of the 3d-space) is attained everywhere, and that the resulting images have with an height of a power of 2; if the mapping is much distorted, far higher resolutions may occur on some parts.
Where the surface's materials are textured, the original, possibly repeated images are then copied to the new images; where they're not, the new images are filled with the material's diffuse color (and the *new* material's diffuse color is set to white). A new material, textured with the generated image is then created and assigned for every part.
The generated images are stored in a subfolder named "textures" of the folder where the scene is saved as .spa or .spah file; if the scene was not saved before, an error message will appear.

<u>to create a unique texture mapping</u>,

1. switch to the -ribbon, make sure color, material and paint are deactivated, then enable texture
2. select the objects or faces to be mapped and invoke *unique texture*
    - multiple objects, or faces on multiple objects can be selected together
3. the dimensions (in pixels) of the images to create, and the total pixel-count (in megapixels) are shown; adjust the desired minimal resolution with pixel/unit, then tap to create the images and materials
    - you may now delete the old material(s) manually if they're not used anywhere else

to start painting,

1. switch to the -ribbon and make sure paint is disabled
2. select the objects or faces you want to paint on, then tap paint
    - select as little as possible; all texture images assigned to the selected surfaces have to be loaded into the RAM multiple times to make the painting effects possible; if error, select less
    - faces on different objects can be used
3. you can now use the painting tools on the selected surfaces; they are ***paintable***
    - only surfaces where a textured material is assigned are *paintable*
    - when you move the *picker* over a *paintable* face, it shows a circle indicating the brush's size and a preview of the *dab*, resp. the start of the stroke the brush would create when starting to paint; else, the pointer shows in red
4. tap paint, or another *tool* or *ribbon-selector* to finish;
    - to continue painting on another part, select it and invoke paint again

Using brushes

Brushes are used to apply, modify, or erase color or transparency; they can create single ***dabs*** (called ***stamping***) or continuous strokes (called ***stroking***). Strokes are composed of individual dabs defined by the tip image; for smooth, continuous strokes, the dabs must overlap so that they are not (or barely) recognizable anymore. Distributed, separate dabs can be created by stroking with a large spacing or scatter.

The brush to use is selected from the *brushes-panel*, where all available brushes are listed grouped in expandable / collapsible categories; to open or close the *panel*, slide leftwards into the screen over the *panel-rim*, the lower long blue line at the right border.

If undoable is enabled, every stroke can be undone; therefor, after each stroke, a new version of the affected texture image(s) is saved to the "textures" subfolder of the folder where the scene is saved; this may lead to lags when painting quickly and require a lot of disk space.

There are two modes for painting with brushes:

- grip disabled: the brush paints whenever you touch the screen over a paintable surface, exactly under the touch-position; slide to *stroke*, tap to *stamp*
    - this is more intuitive and makes natural, quickly repeated strokes possible
- grip enabled: the brush is moved like the *picker;* touch the screen within the *grip-area* below the brush to paint, somewhere else within the *tool-zone* to move the brush; tap within the *grip-area* to apply a *dab*, somewhere else to move the brush to the tapped position
    - this lets you precisely define and preview the position where the *stroke* shall start, resp. where a *dab* shall be applied

Color and transparency

The ***current color*** (and transparency), used for painting and filling, is displayed in the *color-box* at the screen's top right border; its lower halve shows the color without transparency, in the upper halve it is overlaid a gray-white pattern to portray the transparency.

to set the *current color* and *-transparency*, either

- tap the *color-box* to display the *color-picker* and use it as described in Selecting colors
    - tap the *color-box* again to close the *color-picker*, or leave it open and use two fingers to size and place it conveniently
- place two fingers within the *tool-zone*; the *color-picker* is shown temporarily, and the color can be changed by moving the fingers, see to freely move in the color-space

to sample the color from a point on a texture,

1. slide leftwards into the screen over the *color-box* until "eyedropper active" appears in the *status-line,*
2. move the pointer over the texture; the color (and transparency) under the pointer is displayed in the color-box; release the touch to select the color
    - open the *color-picker* beforehand to see the component values

You can paint with (partially) transparent colors to modify the transparency of surfaces: Where the brush works with full *total strength* (with the normal *blending mode*), the resulting transparency will be exactly the chosen one; where it applies the color gradually; the transparency is also changed gradually towards the chosen value.

Brush properties and paint settings

The painting behavior of a brush can be varied in a wide range by adjusting the settings.

The adjusters for scale, strength and mode are always shown when applicable. The respective values are not linked to a brush, but retained when changing it. To show or hide the other adjusters, slide leftwards into the screen over the *adjuster-rim*, the upper long blue line at the right border. While you adjust a value, a preview of the stroke is shown temporarily.

The adjustments made to brushes are not saved automatically in the library, and are reset to the original values when you restart Spacedraw; invoke *save* to make the adjustments permanent.

- tip: defines the brush’s “tip” via an image, according to colorful tip
    - tapping tip opens the file-browser and lets you choose an image file
- rotate along: whether the tip is rotated in the stroke’s direction
- colorful tip: determines how the tip image is interpreted:
    - enabled: the alpha values of the image’s pixels defines the painting strength and the original colors are used; only possible for .png and .tif images
    - disabled: the darkness of the image’s pixels defines the painting strength; where the image is white, no paint is applied, where it is black, the maximal effect results
- scale: the scaling of the tip, not linked to the brush
- size: the brush’s tip diameter, measured in 3d-space-units scaled with the scale-factor
- strength: the maximal strength of a stroke’s effect; e.g. with a value of 0.5, the original color is not altered more than 50% no matter how many dabs overlap and how often you move the brush over the same spot
- mode: the *blending mode*; defines how the brush’s color and transparency is combined with the original color and transparency
    - normal: averages both color and transparency according to the *strength*
    - behind: paints “behind” the existing color; the stroke has no effect on fully opaque areas, on fully transparent areas the effect is the same as with normal; the transparency is never increased
    - atop: paints only “atop” of existing color; the stroke has no effect on fully transparent areas, on opaque areas, the strength depends on the brush’s transparency; the original transparency is not altered
    - darken: only changes colors to make them darker; else works like normal
    - multiply: multiplies both the color- and alpha-values; the surface becomes darker and possibly more transparent with each stroke
    - lighten: only changes colors to make them lighter; else works like normal
    - screen: has the inverse effect of multiply; the surface becomes lighter and possibly more opaque with each stroke

- flow: the strength of each dab; the resulting strength of the stroke depends on how many dabs overlap at a spot and the strength value
- spacing: the distance between the individual dabs
- spacing noise: randomness applied to the spacing
- scatter: how much the dabs are scattered perpendicular to the stroke path
- rotation: orientation of the tip
- jitter: randomness applied to the rotation
- squish: squashing of the tip shape in x- resp. y-direction for valued >0 resp. <0

Organizing brushes

Spacedraw comes with a varied collection of brushes, sorted in 5 categories for different purposes; they demonstrate some of the effects achievable by adjusting a brush's properties:

- basic: brushes with round, homogeneous tips for creating smooth, contiguous strokes with different "hardness", and a calligraphic brush, creating lines whose width depends on the stroke direction
- covering: brushes to cover areas with various structures
    - "colorful fur" demonstrates what is possible by using multicolored tip images with transparency
- scatter: brushes to scatter the tip images
- trace: brushes to create various kinds of "traces" along the stroke, using rotate along
- decals: larger, colorful tip images to be "stamped" on surfaces

The brush definitions are stored in equally named plain text files with ".spbr"-extension as simple colon-separated key-value pairs. They are located together with the *tip images* in subfolders of the "/spacedraw/brushes"-folder.

to add a brush to the library, place the image to be used for the tip in the respective subfolder of the "/spacedraw/brushes"-folder; when you start or restart Spacedraw the next time and open the *brush-panel,* an equally-named brush will appear in the respective category and can be customized

to remove a brush from the library, delete the respective .spbr and image-file

to create or delete a category, create, resp. delete, an equally named subfolder in the "/spacedraw/brushes"-folder

Selecting areas

Areas on the texture can be selected to fill them with color or transparency, apply a filter, or erase them. Contiguous areas with similar colors can quickly and reliably be selected with the magic wand tool.

to select texture areas,

1. enable select
2. either
    - to select faces, enable face, then either tap the faces to select, or move the picker over them and release the touch when the desired face is framed
    - to select contiguous areas with similar colors, enable magic wand and
        - either tap within the area to select, or place the picker at a point within the area
            - while moving the pointer, the area that will be selected according to the current selection tolerance is outlined; however, if this area is very large, only the part of the border nearest to the pointer is marked so

the pointer-movement is not slowed down to much by the calculations.

    - adjust the selection tolerance until the selection has the desired size
        - the area to be selected is updated continuously

3. if desired, modify the selection:
    - enable add, substr. or inters. and continue with 2. to add another area to the selection, subtract it from the selection or intersect another area with the selection
    - invoke *invert* to invert the selection within the paintable texture(s)
4. use *fill*, *erase*, or a *filter*
    - fill floods the selection with the current color according to strength and mode, then clears the selection
    - to apply a filter (see Modifying textures),
        - enable filter, then activate the filter; it is applied to the selection according to its current settings
        - adjust the settings; the effect is updated continuously
        - tap to finish; the selection is cleared, or a new selection created according to the selection settings and tap position

to clear the selection, tap somewhere outside any paintable surface

to select the entire paintable texture(s), clear the selection, the invoke *invert*

Modifying textures

You can mutate parts of a texture in several ways by applying a ***filter***, either by “painting” the effect using a brush, or by first selecting the desired areas and then applying it.

To mutate by painting,

1. enable filter, select a *filter* and adjust its options
2. use a brush to “paint” the effect
    - adjust strength and flow to apply a filter gradually

To mutate by selecting, see Selecting

The following *filters* are available:

- opacity: changes the opacity of the surface towards a target opacity without affecting the color; with maximal *total strength,* the resulting opacity is the exact value of the opacity adjuster
- adj. color: changes the contrast, brightness, saturation and hue according to the values set with the adjusters; changing the brightness results in the effect also known as “***dodge***” and “***burn***”
- blur: applies a Gaussian blur
- sharpen: enhances the apparent sharpness by increasing the contrast along color edges
- emboss: creates an emboss effect

Smudging

to smudge the color on a texture, enable smudge and move a brush as for painting

- The brush's properties and the strength value determine how the color is moved
    - spacing has no influence, but by choosing the tip and adjusting properties like rotate along and jitter, varied effects can be achieved
- while moving the brush, it is shown in the current color (which has no influence on smudging)

Erasing

to erase to a specific color (and transparency), simply paint with it, or select the desired area and invoke *fill*

to erase to the original texture (to the state when it was made *paintable*), either enable erase and paint, or select the desired area and invoke *erase*

- the original texture is composed with the current state according to the *total strength* and mode

Cloning areas

The clone tool copies the texture from one location to another

1. enable clone; source is activated automatically
2. move the pointer to the source position and tap; source is deactivated
    - if the source shall be stamped, adjust the size, squish and rotation of the brush so that it encompasses the desired region
3. move the pointer to the target position; a preview is shown
    - you can clone to any *paintable* surface area
    - adjust the brush's size, squish and rotation to scale and rotate the source
4. either
    - tap within the grip-area below the brush to stamp
    - move the brush to continuously clone from the area around the source-position
5. either
    - to clone from the same source to another area, move the brush to the new target position with aligned disabled, then *stamp* or *stroke*
    - to continue cloning from the same source continuously, proceed painting with aligned enabled
    - to clone from a new source, enable source again and continue with 2.

Projecting images by painting

Images can be projected onto parts of surfaces by painting, based on the view

1. tap project; the file-browser opens; selected the image to be projected
2. place two fingers on the screen; the image is overlaid the viewport as it will be projected; move, rotate and pinch to adjust the position, orientation and scale
3. *stroke* or *stamp* to apply the projection with a brush
    - the *strength* and mode determine how the color and transparency of the projected image is combined with the original color and transparency
4. change the view or image placement to change the projection, or tap project twice to load another image