

Controls

| | |
|--------------------------|--|
| <code>gemhead</code> | the start of rendering chain |
| <code>gemwin</code> | the window manager |
| <code>gemmouse</code> | outputs the mouse position and buttons in the GEM window |
| <code>gemkeyboard</code> | outputs the keycode of a key pressed when you are in the GEM window (there might be different keycodes in Windows/Linux) |
| <code>gemkeyname</code> | outputs a symbolic description of a key pressed when you are in the GEM window (there might be different symbols in Windows/Linux) |

Ref :
<http://gem.iem.at/documentation/manual/manual/List-of-gem-objects>

Manipulators

| | |
|-----------------------------|---|
| <code>accumrotate</code> | accumulate a rotation |
| <code>alpha</code> | enable/disable alpha blending |
| <code>ambient</code> | set the ambient color with a vector |
| <code>ambientRGB</code> | set the ambient color with 3 discrete values |
| <code>camera</code> | camera |
| <code>color</code> | set the color with a vector |
| <code>colorRGB</code> | set the color with 3 discrete values |
| <code>depth</code> | enable/disable depth testing |
| <code>diffuse</code> | set the diffuse color with a vector |
| <code>diffuseRGB</code> | set the diffuse color with 3 discrete values |
| <code>emission</code> | set the emissive color with a vector |
| <code>emissionRGB</code> | set the emissive color with 3 discrete values |
| <code>linear_path</code> | generate a path from an array of points |
| <code>ortho</code> | change the view to orthogonal, with the viewport the size of the window |
| <code>polygon_smooth</code> | turn on anti-aliasing for the objects below |
| <code>rotate</code> | rotate with an angle and vector |
| <code>rotateXYZ</code> | rotate with 3 discrete values |
| <code>scale</code> | scale with a vector |
| <code>scaleXYZ</code> | scale with 3 discrete values |
| <code>separator</code> | push the OpenGL state for the rest of the chain and pop when done |
| <code>shininess</code> | set the shininess of an object |
| <code>specular</code> | set the specular color with a vector |
| <code>specularRGB</code> | set the specular color with 3 discrete values |
| <code>spline_path</code> | generate a spline from an array of knots |
| <code>translate</code> | translate with a vector |
| <code>translateXYZ</code> | translate with 3 discrete values |

Geos

| | |
|---------------------------|---|
| <code>circle</code> | render a circle |
| <code>colorSquare</code> | render a colored square (evtl. with color gradients) |
| <code>cone</code> | render a cone |
| <code>cube</code> | render a cube |
| <code>cuboid</code> | render a box |
| <code>curve</code> | render a Bezier curve |
| <code>curve3d</code> | render a surface |
| <code>cylinder</code> | render a cylinder |
| <code>disk</code> | render a disk |
| <code>imageVert</code> | make pixel colors to a height field map |
| <code>model</code> | render an Alias Wavefront model |
| <code>multimodel</code> | render a series of Alias Wavefront models, render by number |
| <code>newWave</code> | render a wave (that is evolving over time) |
| <code>polygon</code> | render a polygon |
| <code>primTri</code> | a triangle primitive |
| <code>rectangle</code> | render a rectangle |
| <code>ripple</code> | a rectangle with distorted (over time) texture-coordinates |
| <code>rubber</code> | a grid where you can move one of the grid-points |
| <code>slideSquare</code> | render a number of sliding squares |
| <code>sphere</code> | render a sphere |
| <code>square</code> | render a square |
| <code>teapot</code> | render a teapot |
| <code>text2d</code> | render 2-D text (a bitmap) |
| <code>text3d</code> | render 3-D text (polygonal) |
| <code>textextruded</code> | render an extruded 3D-text |
| <code>textoutline</code> | render outlined text (polygonal) |
| <code>triangle</code> | render a triangle |

Geos

| | |
|---------------------------|---|
| <code>circle</code> | render a circle |
| <code>colorSquare</code> | render a colored square (evtl. with color gradients) |
| <code>cone</code> | render a cone |
| <code>cube</code> | render a cube |
| <code>cuboid</code> | render a box |
| <code>curve</code> | render a Bezier curve |
| <code>curve3d</code> | render a surface |
| <code>cylinder</code> | render a cylinder |
| <code>disk</code> | render a disk |
| <code>imageVert</code> | make pixel colors to a height field map |
| <code>model</code> | render an Alias Wavefront model |
| <code>multimodel</code> | render a series of Alias Wavefront models, render by number |
| <code>newWave</code> | render a wave (that is evolving over time) |
| <code>polygon</code> | render a polygon |
| <code>primTri</code> | a triangle primitive |
| <code>rectangle</code> | render a rectangle |
| <code>ripple</code> | a rectangle with distorted (over time) texture-coordinates |
| <code>rubber</code> | a grid where you can move one of the grid-points |
| <code>slideSquare</code> | render a number of sliding squares |
| <code>sphere</code> | render a sphere |
| <code>square</code> | render a square |
| <code>teapot</code> | render a teapot |
| <code>text2d</code> | render 2-D text (a bitmap) |
| <code>text3d</code> | render 3-D text (polygonal) |
| <code>textextruded</code> | render an extruded 3D-text |
| <code>textoutline</code> | render outlined text (polygonal) |
| <code>triangle</code> | render a triangle |

Particles

| | |
|-------------------------------|--|
| <code>part_head</code> | The start of a particle group |
| <code>part_color</code> | Set the range of colors for the new particles |
| <code>part_damp</code> | set the damping for particles |
| <code>part_draw</code> | Apply the actions and render the particles. Accepts a message "draw line" or "draw point" to change the drawing style. |
| <code>part_follow</code> | Particles will follow each other like a snake |
| <code>part_gravity</code> | Have the particles accelerate in a direction |
| <code>part_info</code> | get the information (position, color, size, ...) of each particle |
| <code>part_killold</code> | Remove particles past a certain age |
| <code>part_killslow</code> | Remove particles below a certain speed |
| <code>part_orbitpoint</code> | Orbit the particles around a specified point |
| <code>part_render</code> | render the remaining gem-tree as particles. |
| <code>part_size</code> | Set the size of new particles |
| <code>part_source</code> | Generate particles |
| <code>part_targetcolor</code> | Change color of the particles toward the specified color |
| <code>part_targetsize</code> | Change size of the particles toward the specified size |
| <code>part_velocity</code> | Set the velocity domain (distribution like CONE and the appropriate arguments) |
| <code>part_vertex</code> | emit a single particle |
| <code>light</code> | make a point light |
| <code>world_light</code> | make a light at infinity |

Nongeos

Pixes

| | |
|-------------------------------|--|
| <code>pix_2grey</code> | pix_2grey |
| <code>pix_a_2grey</code> | convert rgb pixels to grey based on alpha channel |
| <code>pix_add</code> | add two pixes together |
| <code>pix_aging</code> | super8-like aging effect |
| <code>pix_alpha</code> | set the alpha value of a pix |
| <code>pix_background</code> | let through only pixels that differ from a static "background" image |
| <code>pix_backlight</code> | a backlight photo effect |
| <code>pix_biquad</code> | 2p2z-filter for subsequent images |
| <code>pix_bitmask</code> | apply a bitmask to a pix |
| <code>pix_blob</code> | get center of gravity |
| <code>pix_buf</code> | buffer a pix |
| <code>pix_buffer</code> | storage room for pixes (like [table] for floats) |
| <code>pix_buffer_read</code> | put/get pixes into/from a pix_buffer |
| <code>pix_buffer_write</code> | |
| <code>pix_chroma_key</code> | color keying (like "blue-box") |
| <code>pix_coloralpha</code> | set the alpha-channel of a pix as a mean-value of the color-components |
| <code>pix_colormatrix</code> | recombine the RGBA-channels with matrix-operation |
| <code>pix_color</code> | |
| <code>pix_colorreduce</code> | reduce the number of colors (statistically) |
| <code>pix_composite</code> | composite two pixes together |
| <code>pix_convolve</code> | convolve a pix with a kernel comment |
| <code>pix_coordinate</code> | set the texture coordinates |
| <code>pix_crop</code> | get a sub-image of a pix |

Pixes_2

| | |
|-------------------------------|--|
| <code>pix_curve</code> | apply color-curves onto a pix |
| <code>pix_data</code> | get pixel data information |
| <code>pix_delay</code> | frame-wise delay |
| <code>pix_diff</code> | get absolute difference of two pixes |
| <code>pix_dot</code> | rasterize a pix with big dots |
| <code>pix_draw</code> | draw a pix |
| <code>pix_dump</code> | dump the pixel-data as a long list of floats |
| <code>pix_duotone</code> | reduce the number of colors by thresholding |
| <code>pix_filmm</code> | use a movie file as a pix source for image-processing |
| <code>pix_flip</code> | flip the pixels of a pix |
| <code>pix_grey</code> | convert any pix into greyscale colorspace |
| <code>pix_halftone</code> | rasterize a pix like it was printed in a newspaper |
| <code>pix_histo</code> | get the histogram of a pix |
| <code>pix_hsv2rgb</code> | transform a pix from HSV-colorspace into RGB-colorspace |
| <code>pix_image</code> | load in an image file |
| <code>pix_imageInPlace</code> | load a series of image files directly into texture-buffer, display by number |
| <code>pix_info</code> | get information about the pix (like dimension, colorspace, ...) |
| <code>pix_invert</code> | invert a pix |
| <code>pix_kaleidoscope</code> | as if you were looking at the pix through a kaleidoscope |
| <code>pix_levels</code> | level adjustment |
| <code>pix_lumaoffset</code> | y-offset pixels depending on their luminance |
| <code>pix_mask</code> | mask a pix based on another pix |
| <code>pix_metainage</code> | recompose an image out of smaller versions of itself |
| <code>pix_mix</code> | mix to pixes together |

Pixes_3

| | |
|-----------------------------|--|
| <code>pix_motionblur</code> | motionblur an image |
| <code>pix_movie</code> | use a movie file as a pix source and load it immediately into the texture-buffer |
| <code>pix_movement</code> | set the alpha-channel with respect to the change between two frames |
| <code>pix_multiply</code> | multiply two pixes |
| <code>pix_multiimage</code> | load in a series of image files, display by number |
| <code>pix_normalize</code> | normalize a pix |
| <code>pix_offset</code> | add an offset to a pix (wrapping instead of clipping) |
| <code>pix_pix2sig~</code> | interpret a pix as 4 (RGBA) audio-signals |
| <code>pix_puzzle</code> | shuffle an image |
| <code>pix_rds</code> | generate a Random Dot Stereogram out of the image (aka: Magic Eye (tm)) |
| <code>pix_rectangle</code> | generate a rectangle in a pix buffer |
| <code>pix_refraction</code> | break up an image into coloured "glass-bricks" |
| <code>pix_resize</code> | resize a pix to next power of 2 |
| <code>pix_rgb2hsv</code> | transform a pix from RGB-colorspace into HSV-colorspace |
| <code>pix_rgba</code> | transform a pix of any format into RGBA |
| <code>pix_roll</code> | (sc)roll through an image (wrapping) |
| <code>pix_rtx</code> | swap time-axis and x-axis |
| <code>pix_scanline</code> | take every nth line of the original image |
| <code>pix_set</code> | set the pixel-data with a long list of floats |
| <code>pix_sig2pix~</code> | interpret 4 audio-signals as (RGBA) image-data |
| <code>pix_snap</code> | capture the render window into a pix |
| <code>pix_snap2tex</code> | capture the render window directly as a texture |
| <code>pix_subtract</code> | subtract two pixes |

Pixes_4

| | |
|----------------------------|--|
| <code>pix_tIIR</code> | time-base Infinite-Impulse-Response filter (for motion-bluring, ...) with settable number of poles/zeros |
| <code>pix_takealpha</code> | take the alpha channel of one pix and put it into another pix |
| <code>pix_texture</code> | use a pix as a texture map |
| <code>pix_threshold</code> | apply a threshold to a pix |
| <code>pix_video</code> | use a video camera as a pix source |
| <code>pix_write</code> | capture the render window to disk |
| <code>pix_zoom</code> | zoom into a pix (using OpenGL) |