Open Command-oriented Geometric Graphics Generator

OpenCG^3 Specification Version 0.2.12

Dong Nai-Jia [1]   Lin Yong-Hsiang [2]

[1]National Chiao Tung University
Department of Computer Science

[2]National Taiwan University
Department of Agricultural Chemistry

August 19, 2017

Command Tokens

Regular Expressions

$\mathbb{N} := \{\ \alpha \mid \alpha \in$ `[0-9]+` $\}$

$\mathbb{R} := \{\ \alpha \mid \alpha \in$ `[+\-]?([0-9]*[.])?[0-9]+` $\}$ $\quad \Rightarrow \mathbb{R} \supset \mathbb{N}$

$\mathbb{S} := \{\ \alpha \mid \alpha \in$ `'(.*?)'|[.0-9A-Za-z+\-]+` $\}$ $\quad \Rightarrow \mathbb{S} \supset \mathbb{R}$

$\mathbb{W} := \{\ \alpha \mid \alpha \in$ `[ \t]` $\}$ whitespace

Descriptions

- The matching mechanism abides by the maximal munch rule.
- Each command is whitespace-insensitive except being quoted by a pair of single quotation marks (`'`).

Command Grammars

Context-Free Expansions

$$\mathbf{C} \rightarrow \mathbf{A}\,\mathbf{C} \mid \texttt{;} \mid \texttt{EOL}$$

$$\mathbf{A} \rightarrow \mathbf{T}(\mathbf{A}) \mid \mathbf{V}(\mathbf{A}) \mid \mathbf{S}(\mathbf{A}) \mid \mathbf{L}(\mathbf{A}) \mid \mathbf{L}(\mathbf{A},\mathbf{A},\cdots,\mathbf{A}) \mid \mathbb{N} \mid \mathbb{R} \mid \mathbb{S}$$

$$\mathbf{T}(\Pi) \equiv \Pi : n\,) \rightarrow \texttt{(}\ \Sigma(\Pi, n)\ \texttt{)}$$

$$\mathbf{V}(\Pi) \equiv \Pi : n\,\rangle \rightarrow \texttt{<}\ \Sigma(\Pi, n)\ \texttt{>}$$

$$\mathbf{S}(\Pi) \equiv \Pi : n\,\} \rightarrow \texttt{\{}\ \Sigma(\Pi, n)\ \texttt{\}}$$

$$\Sigma(\Pi, n) \rightarrow \overbrace{\Pi\ \cdots\ \Pi}^{n \text{ items}} \quad \text{(identical)}$$

$$\mathbf{L}(\Pi) \equiv \mathbb{L}\,[\,\Pi : n\,] \rightarrow \texttt{[}\ \Sigma(\Pi, n)\ \texttt{]}$$

$$\mathbf{L}(\Pi_1, \Pi_2, \cdots, \Pi_{n-1}, \Pi_n) \equiv \mathbb{L}\,[\,\Pi_1\,\Pi_2 \cdots \Pi_{n-1}\,\Pi_n\,] \rightarrow \texttt{[}\ \Pi_1 \cdots \Pi_n\ \texttt{]}$$

Descriptions

- Each command starts from $\mathbf{C}$ and ends with a `;` or an `EOL`.
- Non-terminal symbol expansions are prior than function expansions.

Command Parsing

Escape Sequence

- `\x` is an escape sequence.
- If `x` is `\`, then it is treated as a single backslash.
- If `x` is `EOL` which may vary from platforms, then the sequence is omitted.
- Otherwise, the sequence is ignored and triggers a warning by default.

Error Handling

- Physical lines are separated by an `EOL`.
- Logical lines are separated by either a semicolon or an unescaped `EOL`.
- If the command cannot be parsed by the grammar, then all the characters on the same logical line will be discarded.

Fields, Classes, Objects and References

Definitions

- The whole system are divided into four fields and several classes:
  1. field e-(nviron.): includes class window and class camera.
  2. field p-(rimitive): includes class point, class circle, etc.
  3. field c-(ompound): includes class line, class triangle, class polygon, etc.
  4. field a-(uxiliary): includes class attrib and class group.

Notations

- class^{x} denotes the name of a class in the field `x`.
- label^{x} denotes the unique name of the object from a class in the field `x`.

Prototypes

- Argument prototypes are written in a mixture of types and names with underlines.
- Each type with an asterisk indicates that the brackets are used for cross-referencing.
- Cross-reference is a feature for manipulating multitple objects in a single command.
- Each name with a plus/minus/ampersand implies that the given name is used for creating new objects/deleting existed objects/cross-referencing among objects, etc.

Perspective Projection

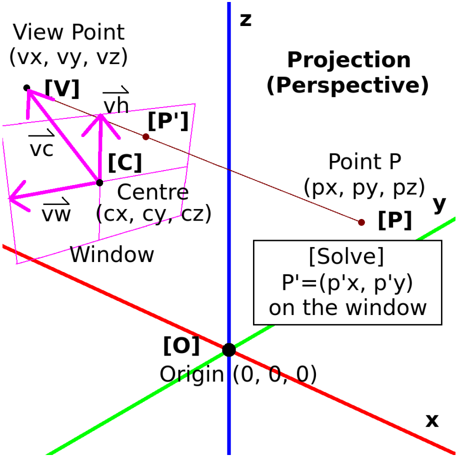

Figure: Projection in Euclidean $\mathbb{R}^3$ Space

Create a Window

Command

$$\texttt{create window}\ \underline{\mathbb{S}\ \ \underline{+\text{label}^{\text{e}}}} \tag{1}$$

Parameters

- $\underline{\text{label}^{\text{e}}}$: the name of the object from the class window

Examples

```
create window main
```

Delete a Window

Command

delete window $\underline{\mathbb{S}\ \text{-label}^{e}}$ $\underline{\mathbb{S}\ \text{string}}$ (2)

Parameters

- $\underline{\text{label}^{e}}$: the name of the object from the class window
- $\underline{\text{string}}$: the text printed right after exiting the current session

Examples

```
delete window main
delete window main 'Have a nice day.'
```

Create a Camera

Command

`create camera` $\underline{\mathbb{S}\ \ \underline{+\text{label}^{e}}}$ $\underline{\mathbb{R}:3)\ \underline{\text{center}}}$ $\underline{\mathbb{R}:3\rangle:2)\ \underline{\text{plane}}}$ $\underline{\mathbb{R}:3\rangle\ \underline{\text{sight}}}$ (3)

Parameters

- $\underline{\text{label}^{e}}$: the name of the object from the class camera
- $\underline{\text{center}}$: the world coordinate (c_x, c_y, c_z) of the center of the viewport
- $\underline{\text{plane}}$: the horizontal and the vertical vertors $(\vec{v_w}, \vec{v_h})$ of the viewport
- $\underline{\text{sight}}$: the reverse line of sight $\vec{v_c}$ from $\underline{\text{center}}$ to the camera

Examples

```
create camera z-top (0 0 1) (<1 0 0> <0 1 0>) <0 0 1>
```

Select a Camera

Command

select camera $\underline{\mathbb{S}\ \underline{\&\,\mathsf{label}_1^{\mathsf{e}}}}$ $\underline{\mathbb{S}\ \underline{\&\,\mathsf{label}_2^{\mathsf{e}}}}$ (4)

Parameters

- $\underline{\mathsf{label}_1^{\mathsf{e}}}$: the name of the object from the class camera
- $\underline{\mathsf{label}_2^{\mathsf{e}}}$: the name of the object from the class window

Examples

```
select camera z-top main
```

Create Points

Command

create point $\underline{\mathbb{S}\ \ \underline{+\,\mathsf{label}^{\mathrm{p}}}\ :\ \}^{\star}}$ $\underline{\mathbb{R} : 3\,)\ \underline{\mathsf{coord}}}$ (5)

create point $\underline{\mathbb{S}\ \ \underline{+\,\mathsf{label}^{\mathrm{p}}} : \geqslant n\,)^{\star}}$ $\underline{\mathbb{R} : 3\,)\ \underline{\mathsf{coord}} : n\,)^{\star}}$ (6)

Parameters

- $\underline{\mathsf{label}^{\mathrm{p}}}$: the name of the object from the class point
- $\underline{\mathsf{coord}}$: the world coordinate (p_x, p_y, p_z) of the object named $\underline{\mathsf{label}^{\mathrm{p}}}$

Examples

```
create point 'origin'   (0 0 0)
create point {X-1 X-2}  (1 0 0)
create point (Y-1 Y-2) ((0 1 0))
create point (Z D1 D2) ((0 1 0) (1 1 1))
```

Delete Points

Command

`delete point` $\underline{\mathbb{S}\ \ \underline{-\,\mathsf{label}^{\mathrm{p}}}\ :\ \}^{\star}}$ (7)

Parameters

- $\underline{\mathsf{label}^{\mathrm{p}}}$: the name of the object from the class point

Examples

```
delete point  origin
delete point {origin 'random-point'}
```

Create Line Segments

Command

$$\texttt{create line} \quad \underline{\mathbb{S}\ \underline{+\text{label}^{\text{c}}} : \}^{\star}} \quad \underline{\mathbb{S}\ \underline{\&\text{label}^{\text{p}}} : 2\}} \tag{8}$$

$$\texttt{create line} \quad \underline{\mathbb{S}\ \underline{+\text{label}^{\text{c}}} : \geqslant n\,)^{\star}} \quad \underline{\mathbb{S}\ \underline{\&\text{label}^{\text{p}}} : 2\} : n\,)^{\star}} \tag{9}$$

Parameters

- $\underline{\text{label}^{\text{c}}}$: the name of the object from the class line
- $\underline{\text{label}^{\text{p}}}$: the name of the object from the class point

Examples

```
create line  seg-1         {X-1 Y-1}
create line {seg-2 seg-3} {X-2 Y-2}
```

Create Attributes

Command

`create attrib` $\underline{\mathbb{S}\ \underline{+\text{label}^{a}} : \}^{\star}}\ \underline{\mathbb{L}[\ \mathbb{L}[\ \mathbb{S}\ \underline{\text{class}^{pc}}\ \mathbb{L}[\ \mathbb{S}\ \underline{\text{prop}}\ \mathbb{S}\ \underline{\text{value}} :]\ ] :]^{\star}}$ (10)

`create attrib` $\underline{\mathbb{S}\ \underline{+\text{label}^{a}} :)^{\star}}\ \underline{\mathbb{L}[\ \mathbb{L}[\ \mathbb{S}\ \underline{\text{class}^{pc}}\ \mathbb{L}[\ \mathbb{S}\ \underline{\text{prop}}\ \mathbb{S}\ \underline{\text{value}} :]\ ] :]^{\star}}$ (11)

Parameters

- $\underline{\text{label}^{a}}$: the name of the object from the class attrib
- $\underline{\text{class}^{pc}}$: the name of a class in the field primitive or compound
- $\underline{\text{prop}}$: the property of the object from $\underline{\text{class}^{pc}}$
- $\underline{\text{value}}$: the value of $\underline{\text{prop}}$ in the designated format

Examples

```
create attrib (magenta dashed-and-translucent-line) \
[[point fill-hsv '(300 1.0 1.0)'] \
 [line [style dashed] [fill-rbga '[(0 255 0) 0.5]']]]
```

Attach Attributes

Command

$$\texttt{attach attrib}\ \underline{\mathbb{S}\ \ \underline{\&\,\text{label}^{\text{a}}}\ :\ )^{\star}}\ \ \underline{\mathbb{S}\ \ \underline{\&\,\text{label}^{\text{pc}}}\ :\ \}^{\star}} \tag{12}$$

$$\texttt{attach attrib}\ \underline{\mathbb{S}\ \ \underline{\&\,\text{label}^{\text{a}}}\ :\ )^{\star}}\ \ \underline{\mathbb{S}\ \ \underline{\&\,\text{label}^{\text{pc}}}\ :\ )^{\star}} \tag{13}$$

Parameters

- $\underline{\text{label}^{\text{a}}}$: the name of the object from the class attrib
- $\underline{\text{label}^{\text{pc}}}$: the name of the object from a class in the field primitive or compound

Examples

```
attach attrib  red          point-0
attach attrib (red large)   point-1
attach attrib  blue        {point-2 rect-0}
attach attrib (5px black)  {point-3 circ-0}
attach attrib (red thick)  (point-4 line-0 triangle-0)
```

Assign Operations

Command

$$\texttt{assign operat}\ \underline{\mathbb{S}\ \text{action}}\ \underline{\mathbb{S}\ \text{class}}\ \underline{\mathbb{N}\ \text{repeat}\ [=\infty]} \tag{14}$$

Parameters

- <u>action</u> : the name of the corresponding action of <u>class</u>
- <u>class</u> : the name of a class
- <u>repeat</u> : the amount of the commands emitting operation names

Examples

```
assign operat create point 2
x-axis (1 0 0)
y-axis (0 1 0)
// Back To Normal
```