Open Command-oriented Geometric Graphics Generator

OpenCG[3] Specification Version 0.3.0

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]+` $\}$ $\qquad \Rightarrow \mathbb{R} \supset \mathbb{N}$

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

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

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

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

$\mathbf{T}(\Pi) \equiv \Pi : n\,) \to (\; \Sigma(\Pi, n) \;) \;\Big\|\;$

$\mathbf{V}(\Pi) \equiv \Pi : n\rangle \to \texttt{<} \; \Sigma(\Pi, n) \; \texttt{>} \;\Big\|\; \Sigma(\Pi, n) \to \overbrace{\Pi \; \cdots \; \Pi}^{n \text{ items}} \quad \text{(identical)}$

$\mathbf{S}(\Pi) \equiv \Pi : n\} \to \texttt{\{} \; \Sigma(\Pi, n) \; \texttt{\}} \;\Big\|\; \mathbf{L}(\Pi) \equiv \mathbb{L}\,[\,\Pi : n\,] \to \texttt{[} \; \Sigma(\Pi, n) \; \texttt{]}$

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

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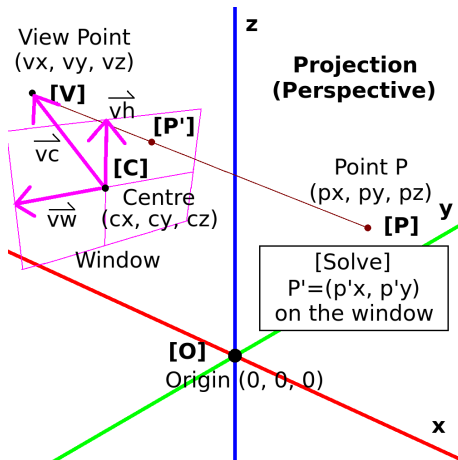
Figure: Projection in Euclidean $\mathbb{R}^3$ Space

Create a Window

Command

```
create window  𝕊  + labelᵉ
```
(1)

Parameters

- <u>label</u>ᵉ : the name of the object from the class window

Examples

```
create window main
```

Delete a Window

Command

`delete window` <u>$\mathbb{S}$ – label^e</u> $\mathbb{S}$ <u>string</u> (2)

Parameters

- <u>label</u>^e : the name of the object from the class window
- <u>string</u> : 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  𝕊 +labelᵉ  ℝ:3) center  ℝ:3⟩:2) plane  ℝ:3⟩ sight        (3)
```

Parameters

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

Examples

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

Command

```
select camera  𝕊 & label₁ᵉ  𝕊 & label₂ᵉ
```
(4)

Parameters

- $label_1^e$: the name of the object from the class camera
- $label_2^e$: the name of the object from the class window

Examples

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

Remove a Camera

Command

`remove camera` $\underline{\mathbb{S} \ \& \ \text{label}_1^e} \quad \underline{\mathbb{S} \ \& \ \text{label}_2^e}$ (5)

Parameters

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

Examples

```
remove camera z-top main
```

Delete a Camera

Command

```
delete camera S – labelᵉ                                          (6)
```

Parameters

- <u>label</u>ᵉ   : the name of the object from the class camera

Examples

```
delete camera z-top
```

Properties of a Camera

Group	Property	Value Format	Example
BG-Color	background-rgb*	$\mathbb{N} \leqslant 255 : 3$)	(0 0 0)
Axis	axis-enable*	$\mathbb{S} \in \{\texttt{true}, \texttt{false}\}$	true
	axis-x-rgb*	$\mathbb{N} \leqslant 255 : 3$)	(255 0 0)
Axis-Color	axis-y-rgb*	$\mathbb{N} \leqslant 255 : 3$)	(0 255 0)
	axis-z-rgb*	$\mathbb{N} \leqslant 255 : 3$)	(0 0 255)
Axis-Width	axis-width*	$\mathbb{L} \, [\, \mathbb{R} \in [0, \infty) \;\; \mathbb{S} \in \{\texttt{px}, \texttt{cm}\} \,]$	[2 px]
Grid	grid-enable*	$\mathbb{S} \in \{\texttt{true}, \texttt{false}\}$	true
Grid-Color	grid-rgb*	$\mathbb{N} \leqslant 255 : 3$)	(191 191 191)
Grid-Width	grid-width*	$\mathbb{L} \, [\, \mathbb{R} \in [0, \infty) \;\; \mathbb{S} \in \{\texttt{px}, \texttt{cm}\} \,]$	[1 px]
Grid-Intvl	grid-interval*	$\mathbb{R} \in [0, \infty)$	10

Create Points

Command

```
create point   S + labelᵖ : }*     R : 3 ) coord                    (7)
create point   S + labelᵖ : ⩾ n )*  R : 3 ) coord : n )*            (8)
```

Parameters

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

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    𝕊 – labelᵖ : }*
```
(9)

Parameters

- labelᵖ   : the name of the object from the class point

Examples

```
delete point 'origin'
delete point {Z D1 D2}
```

Properties of a Point

Group	Property	Value Format	Example
In-Type	type*	$\mathbb{S} \in \{\texttt{dot}, \texttt{square}\}$	`dot`
In-Size	radius*	$\mathbb{L}\,[\,\mathbb{R} \in [0, \infty)\quad \mathbb{S} \in \{\texttt{px}, \texttt{cm}\}\,]$	`[3 px]`
In-Color	fill-rgb*	$\mathbb{N} \leqslant 255 : 3\,)$	`(0 0 0)`
	fill-rgba	$\mathbb{L}\,[\,\mathbb{N} \in [0, 255] : 3\,)\quad \mathbb{R} \in [0, 1]\,]$	`[(31 0 63) 0.5]`
	fill-hsv	$\mathbb{L}\,[\,\mathbb{R} \in [0, 360)\quad \mathbb{R} \in [0, 1] : 2\,)\,]$	`[240 (0.8 1.0)]`
	fill-hsl	$\mathbb{L}\,[\,\mathbb{R} \in [0, 360)\quad \mathbb{R} \in [0, 1] : 2\,)\,]$	`[240 (1.0 0.6)]`
Ex-Type	border-type*	$\mathbb{S} \in \{\texttt{none}, \texttt{mid}, \texttt{in}, \texttt{out}\}$	`none`
Ex-Width	border-width*	$\mathbb{L}\,[\,\mathbb{R} \in [0, \infty)\quad \mathbb{S} \in \{\texttt{px}, \texttt{cm}\}\,]$	`[1 px]`
Ex-Color	border-rgb*	same as fill-rgb	`(127 127 127)`
	border-(...)	same as fill-rgba/hsv/hsl	`(skipped)`

Create Line Segments

Command

```
create line    𝕊 + label^c : }*      𝕊 & label^p : 2}                    (10)
create line    𝕊 + label^c : ⩾ n )*   𝕊 & label^p : 2} : n )*          (11)
```

$$\texttt{create line} \quad \underline{\mathbb{S} \ +\mathsf{label}^c : \}}^{\star} \quad \underline{\mathbb{S} \ \&\,\mathsf{label}^p : 2\}} \tag{10}$$

$$\texttt{create line} \quad \underline{\mathbb{S} \ +\mathsf{label}^c : \geqslant n\,)}^{\star} \quad \underline{\mathbb{S} \ \&\,\mathsf{label}^p : 2\} : n\,)}^{\star} \tag{11}$$

Parameters

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

Delete Line Segments

Command

```
delete line    𝕊 − label^c : }*                                    (12)
```

Parameters

- label^c   : the name of the object from the class line

Examples

```
delete line  seg-1
delete line {seg-2 seg-3}
```

Create Attributes

Command

create attrib $\underline{\mathbb{S} + \underline{label}^a} : \}^\star \ \mathbb{L}\,[\,\mathbb{L}\,[\,\underline{\mathbb{S}\ \underline{class}^{pc}}\ \mathbb{L}\,[\,\underline{\mathbb{S}\ \underline{prop}}\ \underline{\mathbb{S}\ \underline{value}} : \,]\,] : \,]^\star$ (13)

create attrib $\underline{\mathbb{S} + \underline{label}^a} :)^\star \ \mathbb{L}\,[\,\mathbb{L}\,[\,\underline{\mathbb{S}\ \underline{class}^{pc}}\ \mathbb{L}\,[\,\underline{\mathbb{S}\ \underline{prop}}\ \underline{\mathbb{S}\ \underline{value}} : \,]\,] : \,]^\star$ (14)

Parameters

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

```
attach attrib  S & labelᵃ : )*   S & labelᵖᶜ : }*                    (15)
attach attrib  S & labelᵃ : )*   S & labelᵖᶜ : )*                    (16)
```

Parameters

- <u>labelᵃ</u>   : the name of the object from the class attrib
- <u>labelᵖᶜ</u> : 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)
```

Detach Attributes

Command

```
detach attrib  S & label^a : }*   S & label^pc : }*          (17)
detach attrib  S & label^a : )*   S & label^pc : )*          (18)
```

Parameters

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

Examples

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

Delete Attributes

Command

```
delete attrib  𝕊 – labelᵃ : }*
```
(19)

Parameters

- <u>label</u>ᵃ   : the name of the object from the class attrib

Examples

```
delete attrib  red
delete attrib {5px large}
```

Assign Operations

Command

```
assign operat  𝕊 action  𝕊 class  ℕ repeat [= ∞]                    (20)
```

Parameters

- **action** : the name of the corresponding action of class
- **class** : the name of a class
- **repeat** : 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
```