Open Command-oriented Geometric Graphics Generator

OpenCG^3 Specification Version 0.3.3

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 20, 2017

Command Tokens

Regular Expressions

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

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

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

$\mathbb{W} := \{\ \alpha\ |\ \alpha \in \texttt{[\textbackslash t]}\ \}$ 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

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

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

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

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

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

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

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.

Camera
(vx, vy, vz)
[V]
$\vec{vh}$
[P']
$\vec{vc}$
[C]
Center
(cx, cy, cz)
$\vec{vw}$
Viewport

z

**Projection
(Perspective)**

Point P
(px, py, pz)
[P]

y

[Solve]
P'=(p'x, p'y)
on the viewport

[O]
Origin (0, 0, 0)

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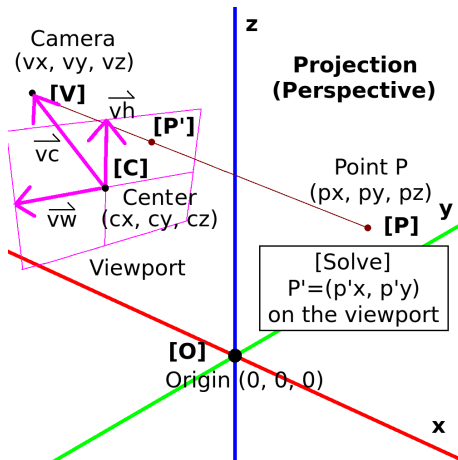
Figure: Projection in Euclidean $\mathbb{R}^3$ Space

Create a Window

Command

```
create window  𝕊  + label^e
```
(1)

Parameters

- label^e   : the name of the object from the class window

Examples

```
create window main
```

Delete a Window

Command

`delete window` $\underline{\mathbb{S} - \text{label}^e}$ $\mathbb{S}$ $\underline{\text{string}}$ (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.'
```

Properties of a Window

Property	Value Format	Example
dots-per-cm	$\mathbb{R}$	128
background-rgb	$\underline{\mathbb{N} \in [0, 255]} : 3\,)$	(0 0 0)

Create a Camera

Command

```
create camera  𝕊 +label^e  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>
```

Select a Camera

Command

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

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

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

Remove a Camera

Command

```
remove camera  𝕊 & label₁ᵉ  𝕊 & label₂ᵉ
```
(5)

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

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

Delete a Camera

Command

```
delete camera S – label^e
```
(6)

Parameters

- label^e   : the name of the object from the class camera

Examples

```
delete camera z-top
```

Properties of a Camera

Property	Value Format	Example
axis-enable	$\underline{\mathbb{S} \in \{\texttt{true}, \texttt{false}\}}$	`true`
axis-xyz	$\underline{\mathbb{S} \in \{\texttt{x}, \texttt{y}, \texttt{z}, \texttt{xy}, \texttt{xz}, \texttt{yz}, \texttt{xyz}\}}$	`xyz`
axis-xyz-rgb	$\underline{\mathbb{N} \in [0, 255] : 3) : 3)}$	`((255 0 0)(0 255 0)(0 0 255))`
axis-width	$\mathbb{L}\,[\,\underline{\mathbb{R} > 0}\ \ \underline{\mathbb{S} \in \{\texttt{px}, \texttt{cm}, \texttt{in}\}}\,]$	`[2 px]`
grid-enable	$\underline{\mathbb{S} \in \{\texttt{true}, \texttt{false}\}}$	`true`
grid-xyz	$\underline{\mathbb{S} \in \{\texttt{x}, \texttt{y}, \texttt{z}, \texttt{xy}, \texttt{xz}, \texttt{yz}, \texttt{xyz}\}}$	`xy`
grid-xyz-rgb	$\underline{\mathbb{N} \in [0, 255] : 3) : 3)}$	`((127 0 0)(0 127 0)(0 0 127))`
grid-width	$\mathbb{L}\,[\,\underline{\mathbb{R} > 0}\ \ \underline{\mathbb{S} \in \{\texttt{px}, \texttt{cm}, \texttt{in}\}}\,]$	`[1 px]`
grid-interval	$\underline{\mathbb{R} > 0}$	`10`

Create Points

Command

`create point`	$\underline{\mathbb{S}}$	$+\underline{\mathsf{label}^{\mathsf{p}}}:\}^{\star}$	$\underline{\mathbb{R}:3)}\ \underline{\mathsf{coord}}$	(7)
`create point`	$\underline{\mathbb{S}}$	$+\underline{\mathsf{label}^{\mathsf{p}}}:\geqslant n)^{\star}$	$\underline{\mathbb{R}:3)}\ \underline{\mathsf{coord}}:n)^{\star}$	(8)

Parameters

- $\underline{\mathsf{label}^{\mathsf{p}}}$: 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 $\underline{\mathsf{label}^{\mathsf{p}}}$

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 − label^p : }^*                                        (9)
```

Parameters

- label^p   : the name of the object from the class point

Examples

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

Properties of a Point

Property	Value Format	Example
shape	$\underline{\mathbb{S}} \in \{\mathtt{none}, \mathtt{dot}, \mathtt{square}\}$	`dot`
radius	$\underline{\mathbb{L}} \, [\, \underline{\mathbb{R} > 0} \quad \underline{\mathbb{S}} \in \{\mathtt{px}, \mathtt{cm}, \mathtt{in}\} \,]$	`[3 px]`
fill-rgb	$\underline{\mathbb{N}} \in [0, 255] : 3\,)$	`(255 255 255)`
border-type	$\underline{\mathbb{S}} \in \{\mathtt{none}, \mathtt{inner}, \mathtt{middle}, \mathtt{outer}\}$	`outer`
border-width	$\underline{\mathbb{L}} \, [\, \underline{\mathbb{R} > 0} \quad \underline{\mathbb{S}} \in \{\mathtt{px}, \mathtt{cm}, \mathtt{in}\} \,]$	`[2 px]`
border-rgb	$\underline{\mathbb{N}} \in [0, 255] : 3\,)$	`(127 127 127)`

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)
```

Parameters

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

Properties of a Line

Property	Value Format	Example
style	$\underline{\mathbb{S} \in \{\texttt{none}, \texttt{solid}, \texttt{dotted}, \texttt{dashed}\}}$	`solid`
width	$\mathbb{L}\,[\,\underline{\mathbb{R} > 0}\ \ \underline{\mathbb{S} \in \{\texttt{px}, \texttt{cm}, \texttt{in}\}}\,]$	`[2 px]`
fill-rgb	$\underline{\mathbb{N} \in [0, 255]} : 3\,)$	`(191 191 191)`
pattern	$\mathbb{L}\,[\,\underline{\mathbb{R} > 0} : \,)\ \underline{\mathbb{S} \in \{\texttt{px}, \texttt{cm}, \texttt{in}\}}\,]$	`[(8 8) px]`

Create Attributes

Command

```
create attrib  𝕊 + label^a : } ]*  𝕃 [ 𝕃 [ 𝕊 class^pc  𝕃 [ 𝕊 prop 𝕊 value : ] ] : ]*  (13)
create attrib  𝕊 + label^a : )*  𝕃 [ 𝕃 [ 𝕊 class^pc  𝕃 [ 𝕊 prop 𝕊 value : ] ] : ]*  (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-rgb '(0 255 0)'] [pattern '[(16 8) px]']]]
```

Attach Attributes

Command

attach attrib $\underline{\mathbb{S} \ \& \ \text{label}^a} :)^\star \quad \underline{\mathbb{S} \ \& \ \text{label}^{pc}} : \}^\star$ (15)

attach attrib $\underline{\mathbb{S} \ \& \ \text{label}^a} :)^\star \quad \underline{\mathbb{S} \ \& \ \text{label}^{pc}} :)^\star$ (16)

Parameters

- $\underline{\text{label}^a}$: the name of the object from the class attrib
- $\underline{\text{label}^{pc}}$: 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)
```

Detach Attributes

Command

```
detach attrib  S & label^a : }*   S & label^pc : }*
detach attrib  S & label^a : )*   S & label^pc : )*
```
(17)
(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  – label^a : }*                                    (19)
```

Parameters

- label^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

- <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
```