# Open Command-oriented Geometric Graphics Generator

OpenCG[3] Specification Version 0.3.4

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

$\mathbb{N} := \{ \; \alpha \mid \alpha \in$ `[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]` $\}$ $\qquad\qquad\qquad$ 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

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

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

$\mathbf{T}(\Pi) \equiv \Pi : n\,) \to (\ \Sigma(\Pi, n)\ ) \ \left\| \ \overbrace{\Sigma(\Pi, n) \to \Pi\ \cdots\ \Pi}^{n \text{ items}} \right. \quad \text{(identical)}$

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

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

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

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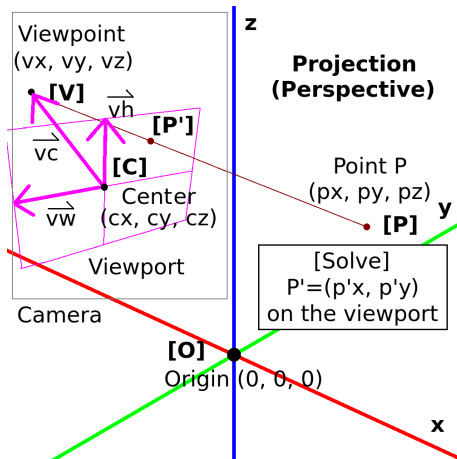
Figure: Projection in Euclidean $\mathbb{R}^3$ Space

# 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  S – label^e  S  string                                    (2)
```

### Parameters

- label^e   : the name of the object from the class window
- 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-color | $\underline{\mathbb{N} \in [0, 255] : 3}\,)$ | (0 0 0) |

## Create a Camera

### Command

```
create camera  S +label^e  R : 3) center  R : 3⟩ : 2) plane  R : 3⟩ sight        (3)
```

### Parameters

- $\underline{label^e}$ : the name of the object from the class camera
- $\underline{center}$ : the world coordinate $(c_x, c_y, c_z)$ of the center of the viewport
- $\underline{plane}$ : the horizontal and the vertical vertors $(\vec{v_w}, \vec{v_h})$ of the viewport
- $\underline{sight}$ : the reverse line of sight $\vec{v_c}$ from $\underline{center}$ 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` $\underline{\mathbb{S}}$ $\underline{\text{\& label}_1^e}$ $\underline{\mathbb{S}}$ $\underline{\text{\& label}_2^e}$ (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

- $\underline{label_1^e}$ : the name of the object from the class camera
- $\underline{label_2^e}$ : the name of the object from the class window

### Examples

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

# Delete a Camera

## Command

```
delete camera  $\mathbb{S}$  – label$^e$                                                    (6)
```

## Parameters

- <u>label</u>$^e$   : 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-show | $\underline{\mathbb{S} \in \{\texttt{x}, \texttt{y}, \texttt{z}, \texttt{xy}, \texttt{xz}, \texttt{yz}, \texttt{xyz}\}}$ | `xyz` |
| axis-color | $\underline{\mathbb{N} \in [0, 255] : 3) : 3)}$ | `((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-show | $\underline{\mathbb{S} \in \{\texttt{x}, \texttt{y}, \texttt{z}, \texttt{xy}, \texttt{xz}, \texttt{yz}, \texttt{xyz}\}}$ | `xy` |
| grid-color | $\underline{\mathbb{N} \in [0, 255] : 3) : 3)}$ | `((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  𝕊 + labelᵖ : }*       ℝ : 3) coord                    (7)
create point  𝕊 + labelᵖ : ⩾ n )*   ℝ : 3) coord : n )*             (8)
```

### Parameters

- <u>labelᵖ</u>   : the name of the object from the class point
- <u>coord</u>   : the world coordinate $(p_x, p_y, p_z)$ of the object named <u>labelᵖ</u>

### 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

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

# Create Line Segments

## Command

```
create line    𝕊 + labelᶜ : }*        𝕊 & labelᵖ : 2}                    (10)
create line    𝕊 + labelᶜ : ⩾ n )*    𝕊 & labelᵖ : 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 | $\mathbb{L}\,[\,\mathbb{R} > 0 \quad \mathbb{S} \in \{\texttt{px}, \texttt{cm}, \texttt{in}\}\,]$ | `[2 px]` |
| color | $\underline{\mathbb{N} \in [0, 255] : 3}\,)$ | `(191 191 191)` |
| cap | $\mathbb{S} \in \{\texttt{butt}, \texttt{round}, \texttt{square}\}$ | `butt` |
| 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 (purple dashed-and-translucent-line) \
[[point color '(128 0 128)'] \
 [line [style dashed] [color '(0 255 0)'] [pattern '[(16 8) px]']]]
```

## 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

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