

Open Command-oriented Geometric Graphics Generator

OpenCG³ Specification Version 0.2.11

Dong Nai-Jia ¹ Lin Yong-Hsiang ²

¹National Chiao Tung University
Department of Computer Science

²National Taiwan University
Department of Agricultural Chemistry

August 18, 2017

Regular Expressions

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

$\mathbb{R} := \{ \alpha \mid \alpha \in [+ \backslash -]^? ([0-9]^* [.])^? [0-9]^+ \}$

$\Rightarrow \mathbb{R} \supset \mathbb{N}$

$\mathbb{S} := \{ \alpha \mid \alpha \in '(. * ?)' \mid [. 0-9 A-Z a-z + \backslash -]^+ \}$

$\Rightarrow \mathbb{S} \supset \mathbb{R}$

$\mathbb{W} := \{ \alpha \mid \alpha \in [\backslash t]^* \}$

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 (').

Context-Free Expansions

$C \rightarrow AC \mid ; \mid \text{EOL}$

$A \rightarrow T(A) \mid V(A) \mid S(A) \mid L(A) \mid L(A, A, \dots, A) \mid N \mid R \mid S$

$$\begin{array}{l} T(\Pi) \equiv \Pi : n \rangle \rightarrow (\Sigma(\Pi, n)) \\ V(\Pi) \equiv \Pi : n \rangle \rightarrow < \Sigma(\Pi, n) > \\ S(\Pi) \equiv \Pi : n \} \rightarrow \{ \Sigma(\Pi, n) \} \\ L(\Pi_1, \Pi_2, \dots, \Pi_{n-1}, \Pi_n) \equiv \mathbb{L} [\Pi_1 \Pi_2 \dots \Pi_{n-1} \Pi_n] \rightarrow [\Pi_1 \dots \Pi_n] \end{array} \quad \left\| \quad \begin{array}{l} \Sigma(\Pi, n) \rightarrow \overbrace{\Pi \dots \Pi}^{n \text{ items}} \quad (\text{identical}) \\ L(\Pi) \equiv \mathbb{L} [\Pi : n] \rightarrow [\Sigma(\Pi, n)] \end{array} \right.$$

Descriptions

- Each command starts from C and ends with a $;$ or an EOL.
- Non-terminal symbol expansions are prior than function expansions except that symbols are used for describing arguments of a command.

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.

Classes

- Classes are split into two categories, top and bottom.
- Top classes are class window, class camera, and data classes.
- Bottom classes are class attrib and class group.
- Data classes are split into primitive classes and compound classes.
- Primitive classes are class point, etc.
- Compound classes are class line, class polygon, etc.

Objects

- An object is instantiated from a class aforementioned.
- An object has an unique name throughout the category of its class.

Relations

- References are bidirectional and can be created or deleted via commands.

Perspective Projection

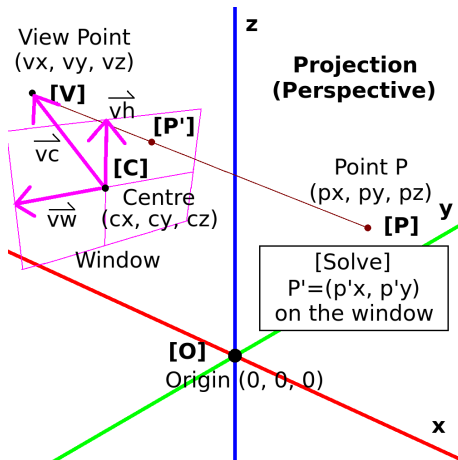


Figure: Projection in Euclidean \mathbb{R}^3 Space

Command

`create window § labelw` (1)

Parametres

- label^w : the name of the object instantiated from the class window

Examples

`create window main`

Command

```
delete window $ labelw $ string (2)
```

Parametres

- label^w : the name of the object instantiated from the class window
- string : the text printed right after exiting the session

Examples

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


Command

`create camera S labelc $\mathbb{R} : 3$) centre $\mathbb{R} : 3$: 2) plane $\mathbb{R} : 3$) sight` (3)

Parametres

- label^c : the name of the object instantiated from the class camera
- centre : the world coordinate (c_x, c_y, c_z) of the centre of the viewport
- plane : the horizontal and the vertical vectors (\vec{v}_w, \vec{v}_h) of the viewport
- sight : the reverse line of sight \vec{v}_c from centre to the camera

Examples

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

Command

```
select camera § labelc § labelw (4)
```

Parametres

- label^c : the name of the object instantiated from the class camera
- label^w : the name of the object instantiated from the class window

Examples

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

Command

`create point` \S $\underline{\text{label}^P} : \}$ ^{*} $\mathbb{R} : 3$) coord (5)

`create point` \S $\underline{\text{label}^P} : \geq n$)^{*} $\mathbb{R} : 3$) $\text{coord} : n$)^{*} (6)

Parametres

- label^P : the name of the object instantiated from the class point
- coord : the world coordinate (p_x, p_y, p_z) of the object named label^P

Examples

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

Command

`delete point § labelP : }` (7)

Parametres

- label^P : the name of the object instantiated from the class point

Examples

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

Command

`create attrib S attrib : }* L [L [S classt L [S prop S value :]] :]*` (8)

`create attrib S attrib :)* L [L [S classt L [S prop S value :]] :]*` (9)

Parametres

- attrib : the name of the object instantiated from the class attrib
- class^t : the name of one of the top classes
- prop : the property of the object of class^t
- value : the value of prop in 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]']]]
```

Command

`attach attrib` \S attrib :)^{*} \S label : }^{*} (10)

`attach attrib` \S attrib :)^{*} \S label :)^{*} (11)

Parameters

- attrib : the name of the object instantiated from the class `attrib`
- label : the name of the object instantiated from one of the top classes

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)
```

Command

`assign operat S action S class N repeat [= ∞]` (12)

Parametres

- action : the name of the corresponding action of class
- class : the name of one of the classes
- 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
```