

# Open Command-oriented Geometric Graphics Generator

OpenCG<sup>3</sup> Spec Version 0.2.2

Dong Nai-Jia <sup>1</sup>    Lin Yong-Siang <sup>2</sup>

<sup>1</sup>National Chiao Tung University  
Department of Computer Science

<sup>2</sup>National Taiwan University  
Department of Agricultural Chemistry

August 13, 2017

# Perspective Projection

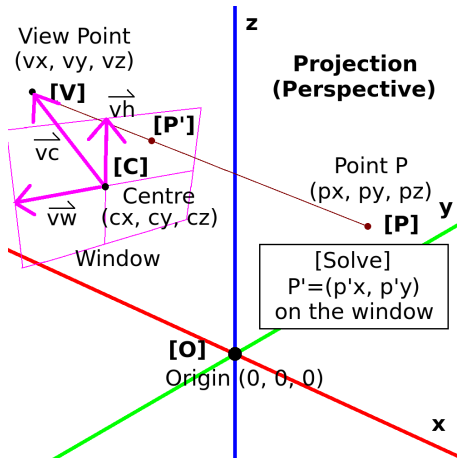


Figure: Projection in Euclidean  $\mathbb{R}^3$  Space

# Command Tokens

## Regular Expressions

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

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

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

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

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

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

$$\text{whitespace}$$

## 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

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

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

$$\mathbf{T(\Pi)} \equiv \Pi : n \rightarrow ( \Sigma(\Pi, n) ) \quad \left\| \quad \Sigma(\Pi, n) \rightarrow \underbrace{\Pi \cdots \Pi}_{n \text{ items}} \quad (\text{identical})$$

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

$$\mathbf{S(\Pi)} \equiv \Pi : n \} \rightarrow \{ \Sigma(\Pi, n) \} \quad \left\| \quad \mathbf{L(\Pi)} \equiv \mathbb{L} [ \Pi : n ] \rightarrow [ \Sigma(\Pi, n) ]$$

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

## Descriptions

- Each command starts from  $\mathbf{C}$  and ends with a  $;$  or an EOL.
- Non-terminal symbol expansions are prior than function expansions except that it is used in the form of describing types of a command.

# Create a Window

## Command

```
create window S label  $\mathbb{R} : 3$  coord  $\mathbb{R} : 3$  : 3 dirct (0)
```

## Parametres

- label: the unique name of the window
- coord: the coordinate  $(c_x, c_y, c_z)$  of the window centre
- dirct: the width  $v_w$ , height  $v_h$ , and the view point  $v_c$

## Examples

```
create window main (0 0 1) (<1 0 0> <0 1 0> <0 0 1>)
```

# Delete a Window

## Command

`delete window $ message` (1)

## Parametres

- message: the text string printed right after exit

## Examples

`delete window`

`delete window 'Have a nice day.'`

# Create Points

## Command

`create point`     $\$$  label : }     $\mathbb{R}$  : 3) coord    (2)

`create point`     $\$$  label :  $n$ )     $\mathbb{R}$  : 3) coord :  $n$ )    (3)

## Parametres

- label:    the name of the point
- coord:    the coordinate  $(p_x, p_y, p_z)$  of the point

## 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))`

# Delete Points

## Command

```
delete point  $ label : } (4)
```

## Parametres

- label: the name of the point

## Examples

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



# Create Attributes

## Command

`create attrib $ label : } L [ $ type $ key A value ]` (5)

`create attrib $ label : n ) L [ L [ $ type $ key A value ] : n ]` (6)

## Parametres

- type: the type of the object
- key: the property of the object
- value: the value of the property

## Examples

```
create attrib {surface} [surface translucency 0.85]
create attrib (magenta auxiliary) \
[[point fill-hsv (300 1.0 1.0)] [line style dashed]]
```

# Assign an Operation

## Command

`assign opratn $ action $ type N repeat [=  $\infty$ ]` (7)

## Parametres

- action: the name of the action
- type: the type of the object applying the action
- repeat: the amount of the commands emitting operations

## Examples

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

This page is intentionally left blank.