Open Command-oriented Geometric Graphics Generator

OpenCG^3 Spec Version 0.2.2

Dong Nai-Jia [1] Lin Yong-Siang [2]

[1] National Chiao Tung University
Department of Computer Science

[2] National Taiwan University
Department of Agricultural Chemistry

August 13, 2017

Perspective Projection

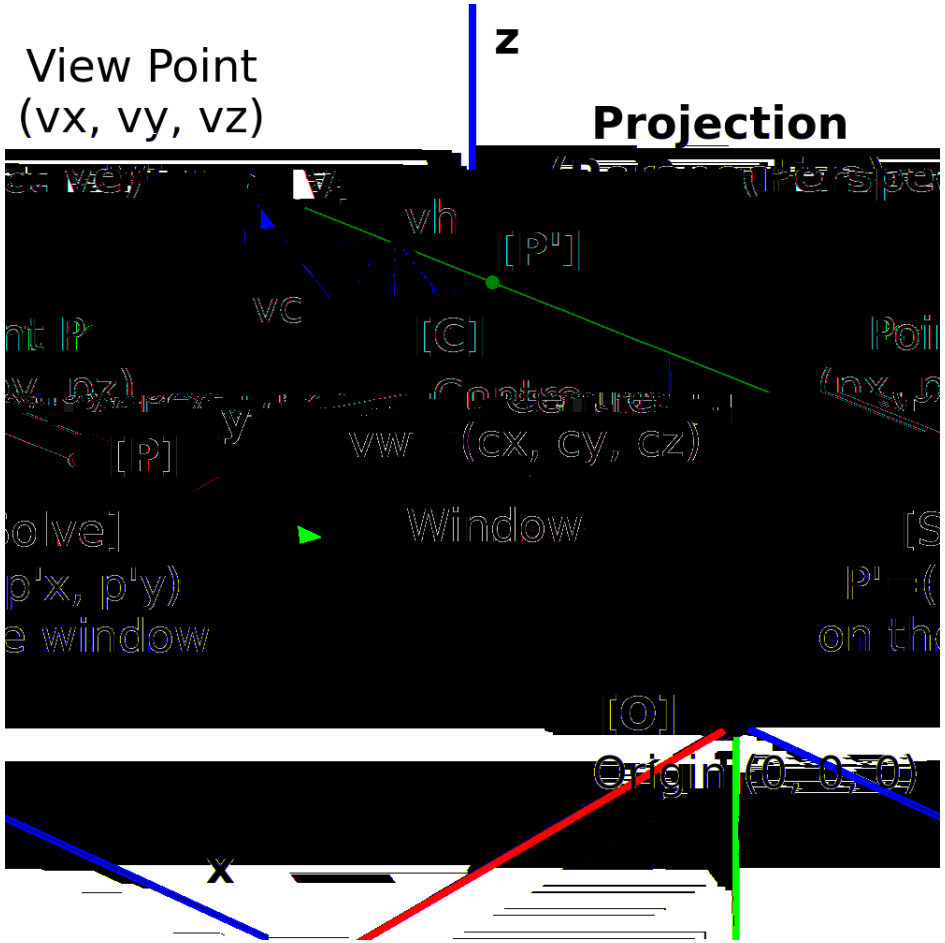

Figure: Projection in Euclidean $\mathbb{R}^3$ Space

Command Tokens

Regular Expressions

$\mathbb{N} \coloneqq \{\ \alpha \mid \alpha \in$ `[0-9]+` $\}$

$\mathbb{R} \coloneqq \{\ \alpha \mid \alpha \in$ `[+\-]?([0-9]*[.])?[0-9]+` $\}$ $\Rightarrow \mathbb{R} \supset \mathbb{N}$

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

$\mathbb{W} \coloneqq \{\ \alpha \mid \alpha \in$ `[ \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

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

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

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

$$\Sigma(\Pi, n) \rightarrow \underbrace{\Pi\ \cdots\ \Pi}_{n \text{ items}} \quad \text{(identical)}$$

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

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

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` $\underline{\mathbb{S}\ \text{label}}\ \underline{\mathbb{R}:3)\ \text{coord}}\ \underline{\mathbb{R}:3\rangle:3)\ \text{dirct}}$ (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 $\vec{v_w}$, height $\vec{v_h}$, and the view point $\vec{v_c}$

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 : S }  ( coord : R 3 )          (2)
create point  ( label : S n )  ( coord : R 3 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` $\mathbb{S}$ label : $\}$ (4)

Parametres

- label: the name of the point

Examples

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

Create Attributes

Command

```
create attrib
```

$$\underline{\mathbb{S}\ \text{label} : \}}\quad \underline{\mathbb{L}[\ \mathbb{S}\ \text{type}\ \mathbb{S}\ \text{key}\ \mathbf{A}\ \text{value}\ ]} \tag{5}$$

```
create attrib
```

$$\underline{\mathbb{S}\ \text{label} : n)}\quad \underline{\mathbb{L}[\ \mathbb{L}[\ \mathbb{S}\ \text{type}\ \mathbb{S}\ \text{key}\ \mathbf{A}\ \text{value}\ ] : n\ ]} \tag{6}$$

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` $\mathbb{S}$ action $\mathbb{S}$ type $\mathbb{N}$ 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.