Open Command-oriented Geometric Graphics Generator

OpenCG^3 Spec Version 0.2.8

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 17, 2017

Perspective Projection

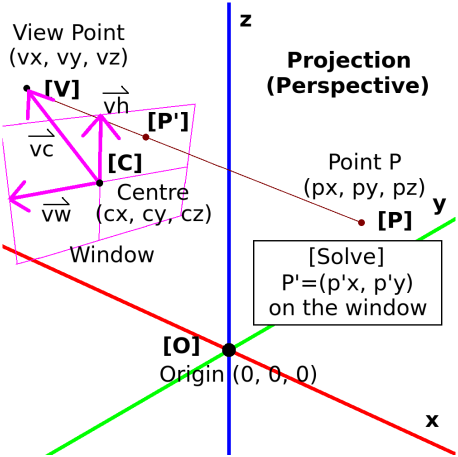

Figure: Projection in Euclidean $\mathbb{R}^3$ Space

Command Tokens

Regular Expressions

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

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

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

$\mathbb{W} := \{ \alpha \mid \alpha \in$ `[ \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

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

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

$$\mathbf{T}(\Pi) \equiv \Pi : n) \to \texttt{(}\ \Sigma(\Pi, n)\ \texttt{)}$$

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

$$\mathbf{S}(\Pi) \equiv \Pi : n\} \to \texttt{\{}\ \Sigma(\Pi, n)\ \texttt{\}}$$

$$\Sigma(\Pi, n) \to \overbrace{\Pi\ \cdots\ \Pi}^{n\ \text{items}} \qquad \text{(identical)}$$

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

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

Descriptions

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

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.

Class and Object System

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.

Create a Window

Command

$$\texttt{create window } \underline{\mathbb{S}\ \underline{\text{label}^{\text{w}}}} \qquad (1)$$

Parametres

- $\underline{\text{label}^{\text{w}}}$: the object name of the class window

Examples

```
create window main
```

Delete a Window

Command

delete window $\underline{\mathbb{S}\ \text{label}^{\text{w}}}$ $\underline{\mathbb{S}\ \text{string}}$ (2)

Parametres

- $\underline{\text{label}^{\text{w}}}$: the object name of the class window
- $\underline{\text{string}}$: the text printed right after exiting the session

Examples

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

Create a Camera

Command

$$\texttt{create camera}\ \underline{\mathbb{S}\ \text{label}^{\text{c}}}\ \underline{\mathbb{R}:3)\ \text{centre}}\ \underline{\mathbb{R}:3\rangle:2)\ \text{plane}}\ \underline{\mathbb{R}:3\rangle\ \text{sight}} \qquad (3)$$

Parametres

- $\underline{\text{label}^{\text{c}}}$: the object name of the class camera
- $\underline{\text{centre}}$: the coordinate (c_x, c_y, c_z) of the centre of the viewport
- $\underline{\text{plane}}$: the horizontal and the vertical vertors $(\vec{v_w}, \vec{v_h})$ of the viewport
- $\underline{\text{sight}}$: the reverse line of sight $\vec{v_c}$, which is from $\underline{\text{centre}}$ to the camera

Examples

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

Attach a Camera

Command

$\texttt{attach camera}\ \underline{\mathbb{S}\ \underline{\text{label}^{c}}}\ \underline{\mathbb{S}\ \underline{\text{label}^{w}}}$ (4)

Parametres

- $\underline{\text{label}^{c}}$: the object name of the class camera
- $\underline{\text{label}^{w}}$: the object name of the class window

Examples

```
attach camera z-top main
```

Create Points

Command

$$\texttt{create point}\quad \underline{\mathbb{S}\ \ \underline{\mathsf{label}^{\mathsf{p}}}\ :\ \}}\qquad \underline{\mathbb{R} : 3\,)\ \underline{\mathsf{coord}}} \tag{5}$$

$$\texttt{create point}\quad \underline{\mathbb{S}\ \ \underline{\mathsf{label}^{\mathsf{p}}} : \geqslant n\,)}\ \ \underline{\mathbb{R} : 3\,)\ \underline{\mathsf{coord}} : n\,)} \tag{6}$$

Parametres

- $\underline{\mathsf{label}^{\mathsf{p}}}$: the object name of the class point
- $\underline{\mathsf{coord}}$: 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` $\underline{\mathbb{S}\ \underline{\text{label}^{p}} : \}}$ (7)

Parametres

- $\underline{\text{label}^{p}}$: the object name of the class point

Examples

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

Create Attributes

Command

$$\texttt{create attrib}\ \underline{\mathbb{S}\ \underline{\text{attrib}} : \}}\ \ \underline{\mathbb{L}\,[\,\mathbb{L}\,[\,\mathbb{S}\ \underline{\text{class}^{t}}\ \mathbb{S}\ \underline{\text{property}}\ \mathbf{A}\ \underline{\text{value}}\,] : \,]} \tag{8}$$

$$\texttt{create attrib}\ \underline{\mathbb{S}\ \underline{\text{attrib}} :)}\ \ \underline{\mathbb{L}\,[\,\mathbb{L}\,[\,\mathbb{S}\ \underline{\text{class}^{t}}\ \mathbb{S}\ \underline{\text{property}}\ \mathbf{A}\ \underline{\text{value}}\,] : \,]} \tag{9}$$

Parametres

- $\underline{\text{attrib}}$: the object name of the class attrib
- $\underline{\text{class}^{t}}$: the name of one of the top classes
- $\underline{\text{property}}$: the property of the object of the class
- $\underline{\text{value}}$: the appropriate form of the property

Examples

```
create attrib (magenta dashed-and-traslucent-green) \
[[point fill-hsv (300 1.0 1.0)] \
 [line style dashed] [line fill-rgba [(0 255 0) .5]]]
```

Attach Attributes

Command

$$\texttt{attach attrib}\ \ \underline{\mathbb{S}\ \ \underline{\text{attrib}} :)}\ \ \underline{\mathbb{S}\ \ \underline{\text{label}} : \}} \tag{10}$$

$$\texttt{attach attrib}\ \ \underline{\mathbb{S}\ \ \underline{\text{attrib}} :)}\ \ \underline{\mathbb{S}\ \ \underline{\text{label}} :)} \tag{11}$$

Parametres

- <u>attrib</u> : the object name of the class attrib
- <u>label</u> : the name of the object instantiated from 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 trianle-0)
```

Assign an Operation Name

Command

`assign opname` $\mathbb{S}$ action $\mathbb{S}$ class $\mathbb{N}$ repeat $[= \infty]$ (12)

Parametres

- action : the name of the action
- class : the name of one of the classes
- repeat : the amount of the commands emitting operation names

Examples

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