3D_APPLET

Java-Applet for interactive representation of flippable polyhedron

Call

```
<applet archive="3d_app.jar"
 code="DreiDApplet.class" width=300 height=200 >
     optional parameter
     optional text
</applet>
```

Width and height specify the size of the applet – a minimum size of width=267 and height=200 is required, since otherwise the space for the control items would not be sufficient!
The size for the actual object display is defined by:

```
Width*  = width  – 162
Height* = height –  62
```

If the object display should be square, the applet-width must be 100 pixels larger than the height. So there is a minimum size with square object display of width=300 and height=200.

All parameters are optional! If no parameter is specified, then the following values are used:

```
fColor     = 150,150,255    ( lightblue)
bColor     = 255,255,255    (white)
lColor     = 255,  0,  0    (red)
shading    = 50
sort       = true
mode       = 3
showPoints = false
points     = (-1, 1, 1),(-1,-1, 1),( 1, 1, 1),( 1,-1, 1),
             (-1, 1,-1),(-1,-1,-1),( 1, 1,-1),( 1,-1,-1)
triangles  = (3,0,1),(3,2,0),(5,4,6),(5,6,7),(5,0,4),(5,1,0),
             (2,3,6),(6,3,7),(6,0,2),(6,4,0),(3,1,5),(3,5,7)
```

This parameters correspond to a cube in light blue color on white background and red lines, whereby the shading value is set to 50. The surfaces are sorted and the display mode is set to “flatshading mode 2”. The points are not displayed. If a parameter is not specified correctly, then it is ignored.

Example:

```
<applet archive="3d_app.jar"
code="DreiDApplet.class" width=501 height=401 >
     <param name="fColor" value="150,255,150">
     <param name="bColor" value="255,255,255">
     <param name="lColor" value="255,0,0">
     <param name="shading" value="50">
     <param name="sort" value="false">
     <param name="mode" value="3">
     <param name="showPoints" value="true">
     <param name="points" value="
          (-1,-1, 2),( 1,-1, 2),(-1,-1,-1),
          (-2,-2, 1),( 2,-2, 1),( 0, 2,-1)">
     <param name="triangles" value=”
          (0,1,2),(3,4,5)">
</applet>
```

Parameter:

- Foreground color of the object
  - *Name:* `fColor`
  - *Type:* String
  - *Value:* Red-value, Green-value, Blue-value; separated by comma;
    the values must be in the range from 0 to 255
  - *Example:* `<param name="fColor" value="150,255,150"`

- Background color of the object
  - *Name:* `bColor`
  - *Type:* String
  - *Value:* Red-value, Green-value, Blue-value; separated by comma;
    the values must be in the range from 0 to 255
  - *Example:* `<param name="bColor" value="255,255,255"`

- Line color of the object (for flat-shading Mode 2)
  - *Name:* `lColor`
  - *Type:* String
  - *Value:* Red-value, Green-value, Blue-value; separated by comma;
    the values must be in the range from 0 to 255
  - *Example:* `<param name="lColor" value="250,0,0"`

- Strength of shading
  - *Name:* `shading`
  - *Type:* String
  - *Value:* integer value in percent (0-100), whereby 0 is equivalent
    to no shade and 100 is equivalent to maximal shade
  - *Example:* `<param name="shading" value="50">`

- Turn off the surface sorting after loading *
  - *Name:* `sort`
  - *Type:* String
  - *Value:* `false` or `no` turns the sorting off – in other cases sorting remains
    turned on
  - *Example:* `<param name="sort" value="false">`

- Starting mode for graphical representation
  - *Name:* `mode`
  - *Type:* String
  - *Value:* integer value between 0 and 4
    0 = Point representation, 1 = Wireframe, 2 = Flat-Shading
    3 = Flat-Shading Mode 2 (with Borderlines), 4 = Flat-Shading + Wireframe
  - *Example:* `<param name="mode" value="3">`

- Points for graphical representation *
  - *Name:* `showPoints`
  - *Type:* String
  - *Value:* `true` or `yes` so that the points are displayed additionally;
    the points can be toggled off and on with the button Points
  - *Example:* `<param name="showPoints" value="true">`

* pay attention to annotations below

- Points of the Object *
    - *Name:* `points`
    - *Type:* String
    - *Value:* Points are separated by comma; each point consists of X-value, Y-value, Z-value which are separated by comma and packed in brackets; the values have to be integer
    - *Example:*
      ```
      <param name="points" value="
      (-1, 1, 1),
      (-1,-1, 1),
      ( 1, 1, 1),
      ( 1,-1, 1),
      (-1, 1,-1),
      (-1,-1,-1),
      ( 1, 1,-1),
      ( 1,-1,-1)">
      ```

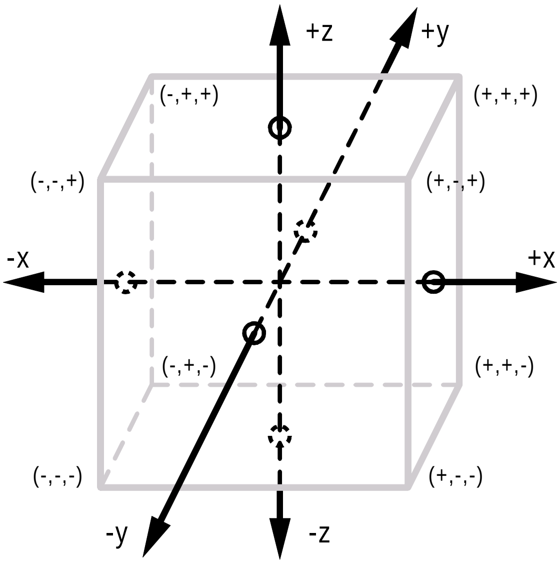
- Triangles of the object *
    - *Name:* `triangles`
    - *Type:* String
    - *Value:* Triangles are separated by comma; each triangle consists of 1st point, 2nd point, 3rd point which are separated by comma and packed in brackets; the values correspond to the sequence of the points; the first point has the **index 0**!
    - *Example:*
      ```
      <param name="triangles" value="
      (2,1,0),
      (1,2,3),
      (4,5,6),
      (5,6,7),
      (0,1,4),
      (1,4,5),
      (2,3,6),
      (3,6,7),
      (0,2,4),
      (2,4,6),
      (1,3,5),
      (5,3,7)">
      ```

* pay attention to annotations below

Annotations:

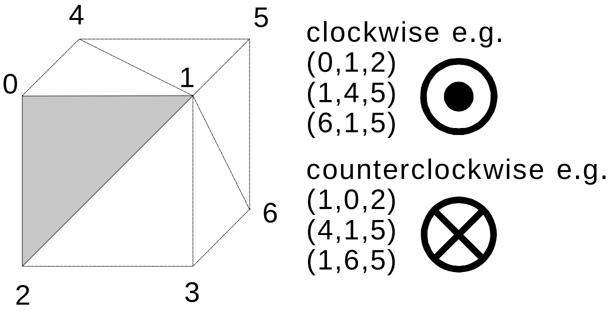
points

The points are given by their x, y and z-coordinates. Refer to the figure on the right for the axes orientation!
The viewer is looking along the y-axis towards plus infinity.
The values have to be in the range from –58 000 to +58 000.
If values outside of this range are used, the display may be incorrect.
The theoretical maximum number of points is 2 147 483 647 (only tested up to 1500)!
The point-coordinates are translated automatically in such a way that the center of the object is at the origin (0,0,0).

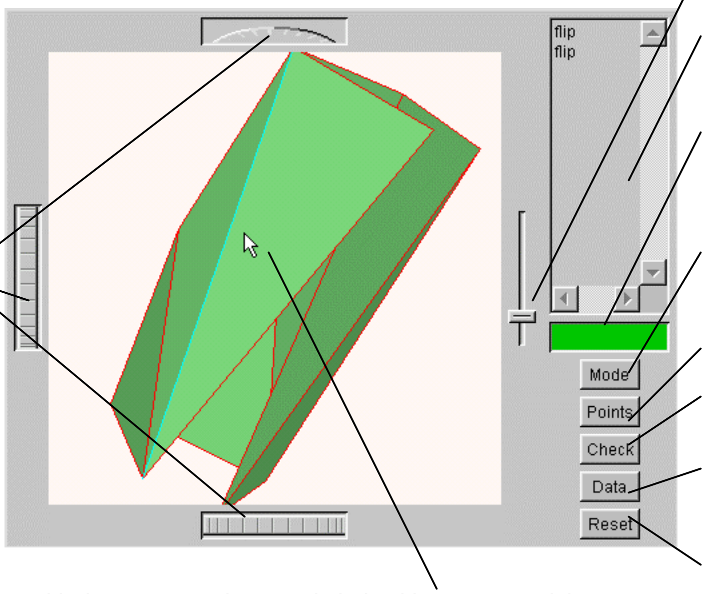


triangles

The sequence of the points, determine the direction of the triangle!
If the order is clockwise, then the surface is orientated towards the viewer. If the points are indicated counter clockwise, then the surface is orientated backwards the viewer.
Surfaces, which are orientated backwards, are not visible from the outside of a polyhedron and therefore they are not displayed.
So that one does not need to consider the orientation of the triangles, the applet sorts the order of the points in such a way that all surfaces look outward. This works only with a closed, contiguous object. If this is not desired or if one wants to represent a not closed object, then sorting must be turned off with the parameter **sort**!
The theoretical maximum number of triangles is 2 147 483 647 (only tested up to 3000)!



Graphical user interface



Scale

Text area for output

Yellow – calculation in progress
Red – selected edge is not flippable
Green – selected edge is flippable

Change graphical representation (Points representation, Wireframe...)

Show points additionally yes / no

Calculate and show all flippable edges

Output the current triangulation to the text area

Reset the triangulation und view

Rotate (automatic rotating by double click on Button Points)

Select an edge with the mouse pointer and flip it with a mouse click.
(works only in the representation mode Flat-Shading, Flat-Shading Mode 2 and Flat-Shading+Wireframe)