VisualCube1e3

Andreas Rentschler, Michael Rentschler

VisualCube Inc. 2009

Package
processing.visualcube1e3

processing.visualcube1e3
Interface AbstractVisualCube

All Known Implementing Classes:
VisualCube

public interface **AbstractVisualCube**
extends

Interface to an abstract VisualCube device. Abstract in the means that this interface is overly simplified from hardware interface.

Nested Class Summary

class	AbstractVisualCube.Color AbstractVisualCube.Color

Field Summary

public static final	colors Maximum number of color tones per r/g/b portion Value: **0**
public static final	depth Maximum number of voxels along z axis Value: **0**
public static final	height Maximum number of voxels along y axis Value: **0**
public static final	width Maximum number of voxels along x axis Value: **0**

Method Summary

void	clear() Set all voxels to be set at next update to black (0, 0, 0).
boolean	close() End connection to device.
String	error() Get error state.
void	fill(AbstractVisualCube.Color c) Set all voxels to be set at next update to specified color value.
void	fill(int r, int g, int b) Set all voxels to be set at next update to specified color portions.
void	fill(int r, int g, int b, float a) Set all voxels to be set at next update to specified color portions.

AbstractVisualCube.Color	get(int x, int y, int z) Get color of specified voxel to be set at next update.
int	getBlue(int x, int y, int z) Get blue color portion of specified voxel to be set at next update.
int	getGreen(int x, int y, int z) Get green color portion of specified voxel to be set at next update.
int	getRed(int x, int y, int z) Get red color portion of specified voxel to be set at next update.
boolean	open() Start connection to device.
boolean	set(int x, int y, int z, AbstractVisualCube.Color c) Set color value of specified voxel to be set at next update.
boolean	set(int x, int y, int z, int r, int g, int b) Set color portions of specified voxel to be set at next update.
boolean	set(int x, int y, int z, int r, int g, int b, float a) Set color portions of specified voxel to be set at next update.
boolean	update() Show current state of voxels on device.

Fields

width

```
public static final int width
```

Maximum number of voxels along x axis
Constant value: **0**

height

```
public static final int height
```

Maximum number of voxels along y axis
Constant value: **0**

depth

```
public static final int depth
```

Maximum number of voxels along z axis
Constant value: **0**

colors

```
public static final int colors
```

Maximum number of color tones per r/g/b portion
Constant value: **0**

(continued on next page)

(continued from last page)

Methods

open

```
public boolean open()
```

Start connection to device.

Returns:
true iff successful

---

close

```
public boolean close()
```

End connection to device.

Returns:
true iff successful

---

update

```
public boolean update()
```

Show current state of voxels on device.

Returns:
true iff successful

---

clear

```
public void clear()
```

Set all voxels to be set at next update to black (0, 0, 0).

---

fill

```
public void fill(AbstractVisualCube.Color c)
```

Set all voxels to be set at next update to specified color value.

Parameters:
`c` - color type defining color portions

---

fill

```
public void fill(int r,
         int g,
         int b)
```

Set all voxels to be set at next update to specified color portions.

Parameters:
`r` - red color portion within interval [0, color - 1]
`g` - green color portion within interval [0, color - 1]
`b` - blue color portion within interval [0, color - 1]

(continued on next page)

(continued from last page)

fill

```
public void fill(int r,
         int g,
         int b,
         float a)
```

Set all voxels to be set at next update to specified color portions.

Parameters:

`r` - red color portion within interval [0, color - 1]
`g` - green color portion within interval [0, color - 1]
`b` - blue color portion within interval [0, color - 1]
`a` - covering degree, 0 being none, 1 being full

---

set

```
public boolean set(int x,
         int y,
         int z,
         AbstractVisualCube.Color c)
```

Set color value of specified voxel to be set at next update.

Parameters:

`x` - voxel position along the x axis within interval [0, width - 1]
`y` - voxel position along the y axis within interval [0, height - 1]
`z` - voxel position along the z axis within interval [0, depth - 1]
`c` - color type defining color portions

Returns:

true iff successful

---

set

```
public boolean set(int x,
         int y,
         int z,
         int r,
         int g,
         int b)
```

Set color portions of specified voxel to be set at next update.

Parameters:

`x` - voxel position along the x axis within interval [0, width - 1]
`y` - voxel position along the y axis within interval [0, height - 1]
`z` - voxel position along the z axis within interval [0, depth - 1]
`r` - red color portion within interval [0, color - 1]
`g` - green color portion within interval [0, color - 1]
`b` - blue color portion within interval [0, color - 1]

Returns:

true iff successful

(continued on next page)

(continued from last page)

set

```
public boolean set(int x,
         int y,
         int z,
         int r,
         int g,
         int b,
         float a)
```

Set color portions of specified voxel to be set at next update.

Parameters:

`x` - voxel position along the x axis within interval [0, width - 1]
`y` - voxel position along the y axis within interval [0, height - 1]
`z` - voxel position along the z axis within interval [0, depth - 1]
`r` - red color portion within interval [0, color - 1]
`g` - green color portion within interval [0, color - 1]
`b` - blue color portion within interval [0, color - 1]
`a` - covering degree, 0 being none, 1 being full

Returns:

true iff successful

---

get

```
public AbstractVisualCube.Color get(int x,
         int y,
         int z)
```

Get color of specified voxel to be set at next update.

Parameters:

`x` - voxel position along the x axis within interval [0, width - 1]
`y` - voxel position along the y axis within interval [0, height - 1]
`z` - voxel position along the z axis within interval [0, depth - 1]

Returns:

a color type

---

getRed

```
public int getRed(int x,
         int y,
         int z)
```

Get red color portion of specified voxel to be set at next update.

Parameters:

`x` - voxel position along the x axis within interval [0, width - 1]
`y` - voxel position along the y axis within interval [0, height - 1]
`z` - voxel position along the z axis within interval [0, depth - 1]

Returns:

a color value ranging from 0 to (colors - 1)

(continued on next page)

(continued from last page)

getGreen

```
public int getGreen(int x,
                    int y,
                    int z)
```

Get green color portion of specified voxel to be set at next update.

Parameters:

`x` - voxel position along the x axis within interval [0, width - 1]
`y` - voxel position along the y axis within interval [0, height - 1]
`z` - voxel position along the z axis within interval [0, depth - 1]

Returns:

a color value ranging from 0 to (colors - 1)

---

getBlue

```
public int getBlue(int x,
                   int y,
                   int z)
```

Get blue color portion of specified voxel to be set at next update.

Parameters:

`x` - position along the x axis within interval [0, width - 1]
`y` - position along the y axis within interval [0, height - 1]
`z` - position along the z axis within interval [0, depth - 1]

Returns:

a color value ranging from 0 to (colors - 1)

---

error

```
public String error()
```

Get error state.

Returns:

descriptive text describing error

processing.visualcube1e3
Class AbstractVisualCube.Color

```
java.lang.Object
   |
   +-processing.visualcube1e3.AbstractVisualCube.Color
```

public static class **AbstractVisualCube.Color**
extends Object

Type represents color value of a voxel.

Field Summary

public	a
public	b
public	g
public	r

Constructor Summary

public	AbstractVisualCube.Color(int r, int g, int b) Initialize a specific color
public	AbstractVisualCube.Color(int r, int g, int b, float a) Initialize a specific color

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

r

public int **r**

g

public int **g**

(continued from last page)

b

```
public int b
```

---

a

```
public float a
```

Constructors

AbstractVisualCube.Color

```
public AbstractVisualCube.Color(int r,
                                int g,
                                int b)
```

Initialize a specific color

Parameters:

`r` - red color portion within interval [0, color - 1]
`g` - green color portion within interval [0, color - 1]
`b` - blue color portion within interval [0, color - 1]

AbstractVisualCube.Color

```
public AbstractVisualCube.Color(int r,
                                int g,
                                int b,
                                float a)
```

Initialize a specific color

Parameters:

`r` - red color portion within interval [0, color - 1]
`g` - green color portion within interval [0, color - 1]
`b` - blue color portion within interval [0, color - 1]
`a` - degree of coverage within interval [0, 1]

processing.visualcube1e3
Class VisualCube

```
java.lang.Object
   |
   +-processing.visualcube1e3.VisualCube
```

All Implemented Interfaces:
AbstractVisualCube

---

public class **VisualCube**
extends Object
implements AbstractVisualCube

Communication interface for the VisualCube. To be used from Processing or real java alike.

Field Summary

public static final	colors Number of colors per R/G/B value Value: **256**
public static final	depth Number of voxels along Z axis Value: **10**
public	frame Voxel matrix, last updated
public static final	frameRate Number of frames per second Value: **10**
public static final	height Number of voxels along Y axis Value: **10**
public	voxels Voxel matrix
public static final	width Number of voxels along X axis Value: **10**

Fields inherited from interface processing.visualcube1e3.AbstractVisualCube

colors, depth, height, width

Constructor Summary

public	VisualCube(PApplet sketch) Initialize cube with parameter stating the calling applet possibly like this: `cube = new VisualCube(this);`

Method Summary

void	clear()
boolean	close()
void	cuboid(int x0, int y0, int z0, int x1, int y1, int z1, AbstractVisualCube.Color c) Draw a cuboid filled with a color.
void	cuboid(int x0, int y0, int z0, int x1, int y1, int z1, AbstractVisualCube.Color c0, AbstractVisualCube.Color c1) Draw a cuboid filled with a specific color, frame colored differently.
void	cuboid(int x0, int y0, int z0, int x1, int y1, int z1, int r, int g, int b) Draw a cuboid filled with a color.
void	cuboid(int x0, int y0, int z0, int x1, int y1, int z1, int r, int g, int b, float a) Draw a cuboid filled with a color.
void	cuboid(int x0, int y0, int z0, int x1, int y1, int z1, int r0, int g0, int b0, float a0, int r1, int g1, int b1, float a1) Draw a cuboid filled with a specific color, frame colored differently.
void	cuboid(int x0, int y0, int z0, int x1, int y1, int z1, int r0, int g0, int b0, int r1, int g1, int b1) Draw a cuboid filled with a specific color, frame colored differently.
String	error()
void	fill(AbstractVisualCube.Color c)
void	fill(int r, int g, int b)
void	fill(int r, int g, int b, float a)
AbstractVisualCube.Color	get(int x, int y, int z)
int	getBlue(int x, int y, int z)
AbstractVisualCube.Color	getFromFrame(int x, int y, int z) Get voxel color from last updated frame made visible
int	getGreen(int x, int y, int z)
int	getRed(int x, int y, int z)
void	line(int x0, int y0, int z0, int x1, int y1, int z1, AbstractVisualCube.Color c) Draw a line between 2 points with a specific color.
void	line(int x0, int y0, int z0, int x1, int y1, int z1, int r, int g, int b) Draw a line between 2 points with a specific color.

void	line(int x0, int y0, int z0, int x1, int y1, int z1, int r, int g, int b, float a) Draw a line between 2 points with a specific color.
boolean	open()
boolean	open(String url) Open VisualCube device specified by URL
boolean	set(int x, int y, int z, AbstractVisualCube.Color c)
boolean	set(int x, int y, int z, int r, int g, int b)
boolean	set(int x, int y, int z, int r, int g, int b, float a)
void	simulate(int width, int height) Initialize a sketch's drawing canvas during start-up like this init() { simulate(640, 480); }
boolean	update()

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface processing.visualcube1e3.AbstractVisualCube

clear, close, error, fill, fill, fill, get, getBlue, getGreen, getRed, open, set, set, set, update

Fields

width

public static final int **width**

Number of voxels along X axis
Constant value: **10**

height

public static final int **height**

Number of voxels along Y axis
Constant value: **10**

depth

public static final int **depth**

Number of voxels along Z axis
Constant value: **10**

(continued from last page)

colors

```
public static final int colors
```

Number of colors per R/G/B value
Constant value: **256**

---

frameRate

```
public static final int frameRate
```

Number of frames per second
Constant value: **10**

---

voxels

```
public processing.visualcube1e3.device.VisualCubeConstants.Voxel voxels
```

Voxel matrix

---

frame

```
public processing.visualcube1e3.device.VisualCubeConstants.Voxel frame
```

Voxel matrix, last updated

Constructors

VisualCube

```
public VisualCube(PApplet sketch)
```

Initialize cube with parameter stating the calling applet possibly like this: `cube = new VisualCube(this);`

Parameters:

`sketch` - Calling sketch with a canvas to draw onto

Methods

open

```
public boolean open()
```

---

open

```
public boolean open(String url)
```

Open VisualCube device specified by URL

Parameters:

`url` - URL identifying cube on network, e.g. "192.168.0.123". If it is null, environment variable "visualcube1e3.url" or java property "visualcube1e3.url" is taken. If those are null too, /etc/hosts is asked for a name->IP mapping of "visualcube1e3" (compare to "localhost").

(continued from last page)

close

```
public boolean close()
```

---

simulate

```
public void simulate(int width,
         int height)
```

Initialize a sketch's drawing canvas during start-up like this `init() { simulate(640, 480); }`

Parameters:

`width` - Width of canvas
`height` - Height of canvas

---

update

```
public boolean update()
```

---

clear

```
public void clear()
```

---

fill

```
public void fill(AbstractVisualCube.Color c)
```

---

fill

```
public void fill(int r,
         int g,
         int b)
```

---

fill

```
public void fill(int r,
         int g,
         int b,
         float a)
```

---

set

```
public boolean set(int x,
         int y,
         int z,
         AbstractVisualCube.Color c)
```

set

```
public boolean set(int x,
         int y,
         int z,
         int r,
         int g,
         int b)
```

set

```
public boolean set(int x,
         int y,
         int z,
         int r,
         int g,
         int b,
         float a)
```

get

```
public AbstractVisualCube.Color get(int x,
         int y,
         int z)
```

getFromFrame

```
public AbstractVisualCube.Color getFromFrame(int x,
         int y,
         int z)
```

Get voxel color from last updated frame made visible

Parameters:

x - Position on x axis
y - Position on y axis
z - Position on z axis

Returns:

Color of specified voxel

getRed

```
public int getRed(int x,
         int y,
         int z)
```

getGreen

```
public int getGreen(int x,
         int y,
         int z)
```

(continued from last page)

getBlue

```
public int getBlue(int x,
         int y,
         int z)
```

error

```
public String error()
```

line

```
public void line(int x0,
         int y0,
         int z0,
         int x1,
         int y1,
         int z1,
         AbstractVisualCube.Color c)
```

Draw a line between 2 points with a specific color.

Parameters:

- `x0` - x-coordinate of starting point
- `y0` - y-coordinate of starting point
- `z0` - z-coordinate of starting point
- `x1` - x-coordinate of target point
- `y1` - y-coordinate of target point
- `z1` - z-coordinate of target point
- `c` - Line's color

line

```
public void line(int x0,
         int y0,
         int z0,
         int x1,
         int y1,
         int z1,
         int r,
         int g,
         int b)
```

Draw a line between 2 points with a specific color.

Parameters:

- `x0` - x-coordinate of starting point
- `y0` - y-coordinate of starting point
- `z0` - z-coordinate of starting point
- `x1` - x-coordinate of target point
- `y1` - y-coordinate of target point
- `z1` - z-coordinate of target point
- `r` - red color portion within interval [0, color - 1]
- `g` - green color portion within interval [0, color - 1]
- `b` - blue color portion within interval [0, color - 1]

line

```
public void line(int x0,
         int y0,
         int z0,
         int x1,
         int y1,
         int z1,
         int r,
         int g,
         int b,
         float a)
```

Draw a line between 2 points with a specific color.

Parameters:

- `x0` - x-coordinate of starting point
- `y0` - y-coordinate of starting point
- `z0` - z-coordinate of starting point
- `x1` - x-coordinate of target point
- `y1` - y-coordinate of target point
- `z1` - z-coordinate of target point
- `r` - red color portion within interval [0, color - 1]
- `g` - green color portion within interval [0, color - 1]
- `b` - blue color portion within interval [0, color - 1]
- `a` - covering degree, 0 being none, 1 being full

cuboid

```
public void cuboid(int x0,
         int y0,
         int z0,
         int x1,
         int y1,
         int z1,
         AbstractVisualCube.Color c0,
         AbstractVisualCube.Color c1)
```

Draw a cuboid filled with a specific color, frame colored differently.

Parameters:

- `x0` - x-coordinate of starting point
- `y0` - y-coordinate of starting point
- `z0` - z-coordinate of starting point
- `x1` - x-coordinate of target point
- `y1` - y-coordinate of target point
- `z1` - z-coordinate of target point
- `c0` - inner color
- `c1` - color to draw frame with

(continued from last page)

cuboid

```
public void cuboid(int x0,
         int y0,
         int z0,
         int x1,
         int y1,
         int z1,
         int r0,
         int g0,
         int b0,
         int r1,
         int g1,
         int b1)
```

Draw a cuboid filled with a specific color, frame colored differently.

Parameters:

- `x0` - x-coordinate of starting point
- `y0` - y-coordinate of starting point
- `z0` - z-coordinate of starting point
- `x1` - x-coordinate of target point
- `y1` - y-coordinate of target point
- `z1` - z-coordinate of target point
- `r0` - interior red color portion within interval [0, color - 1]
- `g0` - interior green color portion within interval [0, color - 1]
- `b0` - interior blue color portion within interval [0, color - 1]
- `r1` - frame's red color portion within interval [0, color - 1]
- `g1` - frame's green color portion within interval [0, color - 1]
- `b1` - frame's blue color portion within interval [0, color - 1]

cuboid

```
public void cuboid(int x0,
         int y0,
         int z0,
         int x1,
         int y1,
         int z1,
         int r0,
         int g0,
         int b0,
         float a0,
         int r1,
         int g1,
         int b1,
         float a1)
```

Draw a cuboid filled with a specific color, frame colored differently.

Parameters:

- `x0` - x-coordinate of starting point
- `y0` - y-coordinate of starting point
- `z0` - z-coordinate of starting point
- `x1` - x-coordinate of target point
- `y1` - y-coordinate of target point
- `z1` - z-coordinate of target point
- `r0` - interior red color portion within interval [0, color - 1]
- `g0` - interior green color portion within interval [0, color - 1]
- `b0` - interior blue color portion within interval [0, color - 1]
- `a0` - interior color's covering degree, 0 being none, 1 being full
- `r1` - frame's red color portion within interval [0, color - 1]
- `g1` - frame's green color portion within interval [0, color - 1]

(continued from last page)

`b1` - frame's blue color portion within interval [0, color - 1]
`a` - frame's color's covering degree, 0 being none, 1 being full

---

cuboid

```
public void cuboid(int x0,
         int y0,
         int z0,
         int x1,
         int y1,
         int z1,
         AbstractVisualCube.Color c)
```

Draw a cuboid filled with a color.

Parameters:

`x0` - x-coordinate of starting point
`y0` - y-coordinate of starting point
`z0` - z-coordinate of starting point
`x1` - x-coordinate of target point
`y1` - y-coordinate of target point
`z1` - z-coordinate of target point
`c` - frame's color

---

cuboid

```
public void cuboid(int x0,
         int y0,
         int z0,
         int x1,
         int y1,
         int z1,
         int r,
         int g,
         int b)
```

Draw a cuboid filled with a color.

Parameters:

`x0` - X-coordinate of starting point
`y0` - Y-coordinate of starting point
`z0` - Z-coordinate of starting point
`x1` - X-coordinate of target point
`y1` - Y-coordinate of target point
`z1` - Z-coordinate of target point
`r` - red color portion within interval [0, color - 1]
`g` - green color portion within interval [0, color - 1]
`b` - blue color portion within interval [0, color - 1]

---

cuboid

```
public void cuboid(int x0,
         int y0,
         int z0,
         int x1,
         int y1,
         int z1,
         int r,
         int g,
         int b,
         float a)
```

(continued from last page)

Draw a cuboid filled with a color.

Parameters:

- `x0` - X-coordinate of starting point
- `y0` - Y-coordinate of starting point
- `z0` - Z-coordinate of starting point
- `x1` - X-coordinate of target point
- `y1` - Y-coordinate of target point
- `z1` - Z-coordinate of target point
- `r` - red color portion within interval [0, color - 1]
- `g` - green color portion within interval [0, color - 1]
- `b` - blue color portion within interval [0, color - 1]
- `a` - covering degree, 0 being none, 1 being full

Index