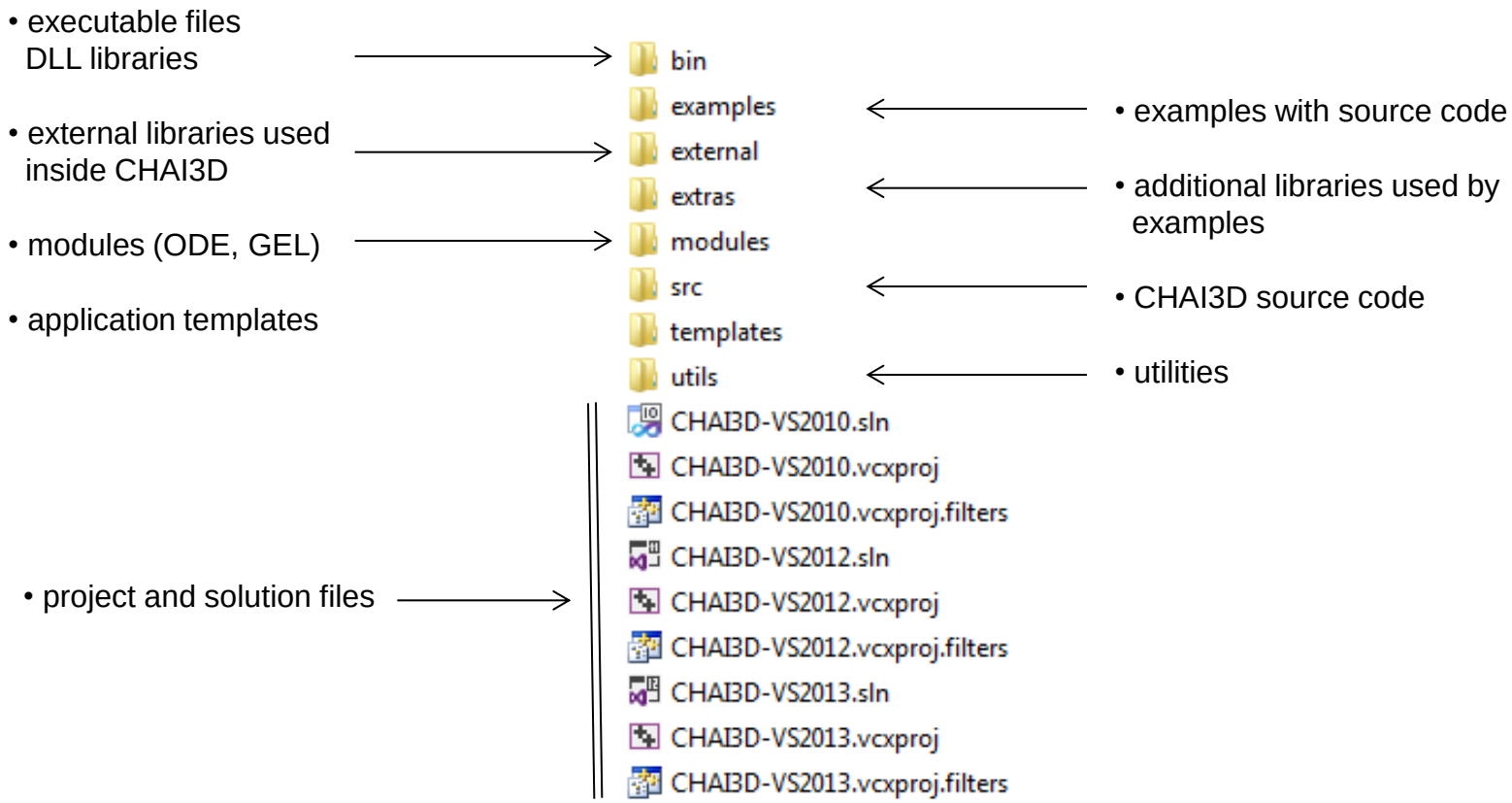
CS 277 - Experimental Haptics
Lecture 5

CHAI3D

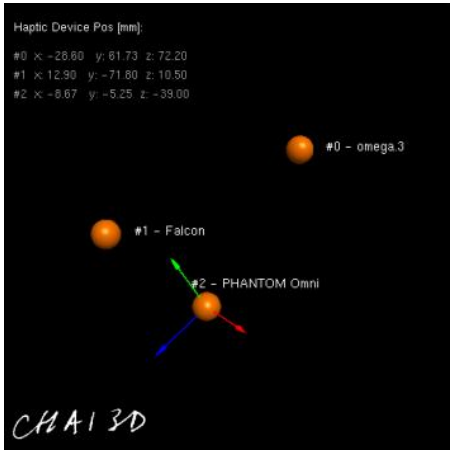
Overview

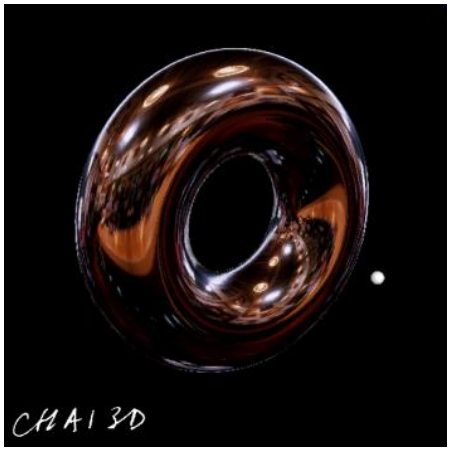
- CHAI3D framework organization
- Haptic interfaces
- Coordinates
- Building a world
- Scene graph
- Objects
- Tools
- Force algorithms
- Examples

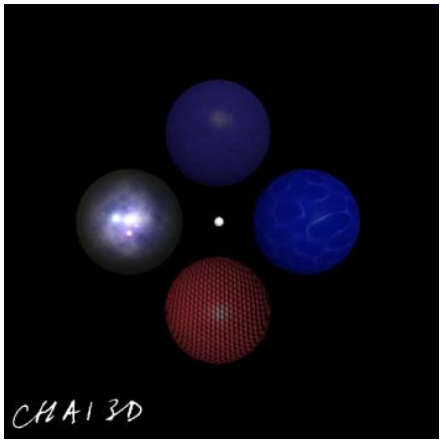
Organization



Examples

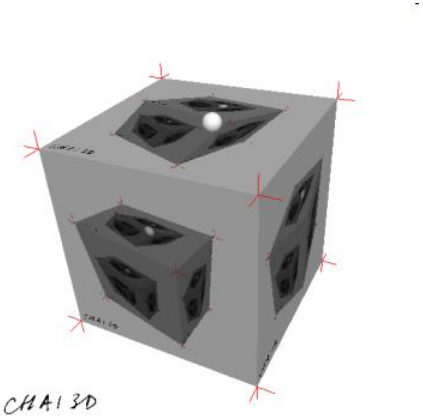

Haptic Device Pos [mm]:
#0 x: -28.60 y: 61.73 z: 72.20
#1 x: 12.90 y: -71.80 z: 10.50
#2 x: -8.67 y: -5.25 z: -39.00
#0 - omega.3
#1 - Falcon
#2 - PHANTOM Omni
CHAI 3D


CHAI 3D

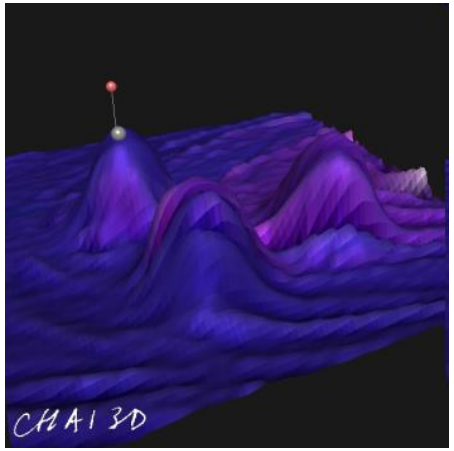

CHAI 3D

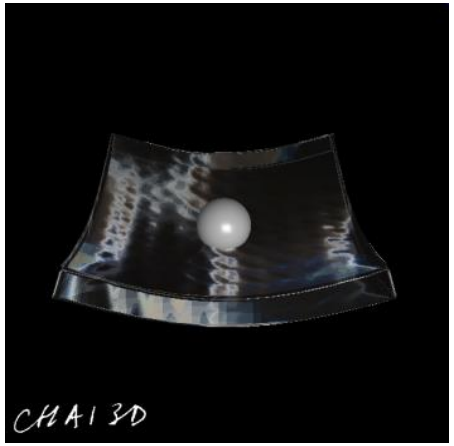

CHAI 3D

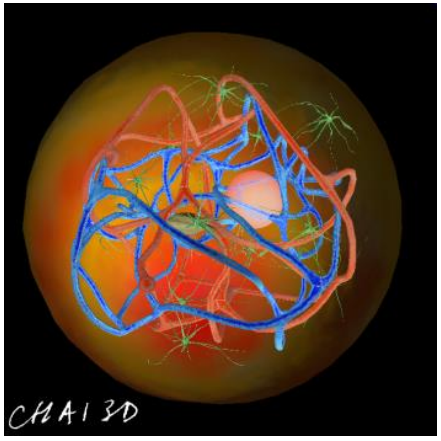

Nikon
CHAI 3D


CHAI 3D

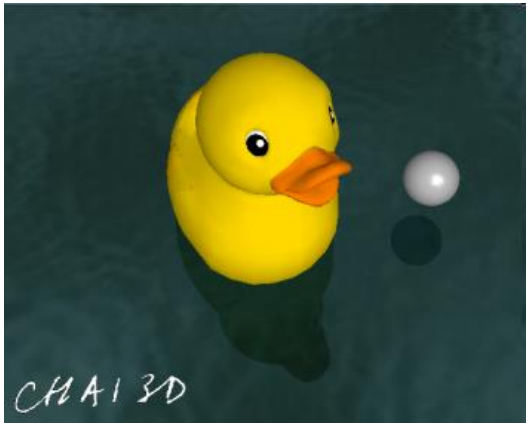
Examples

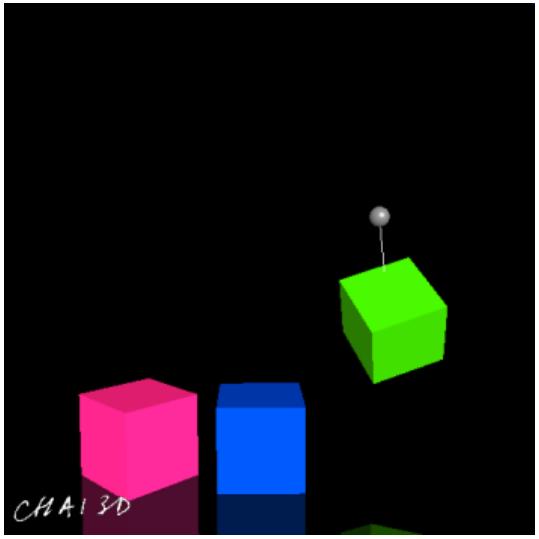


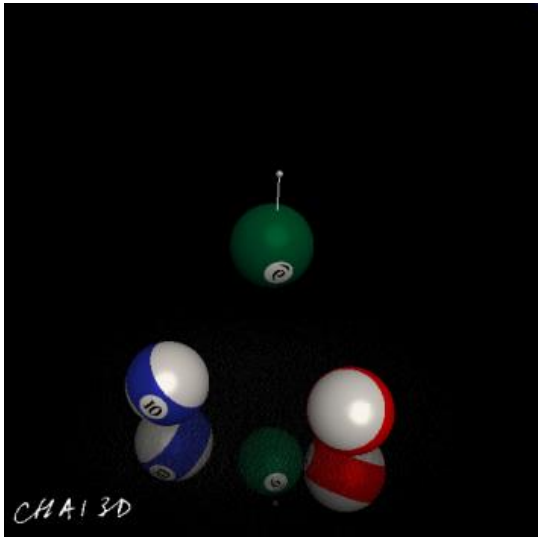


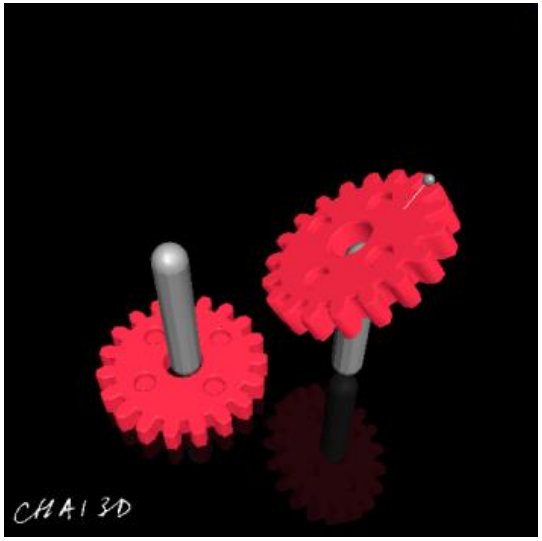


Examples





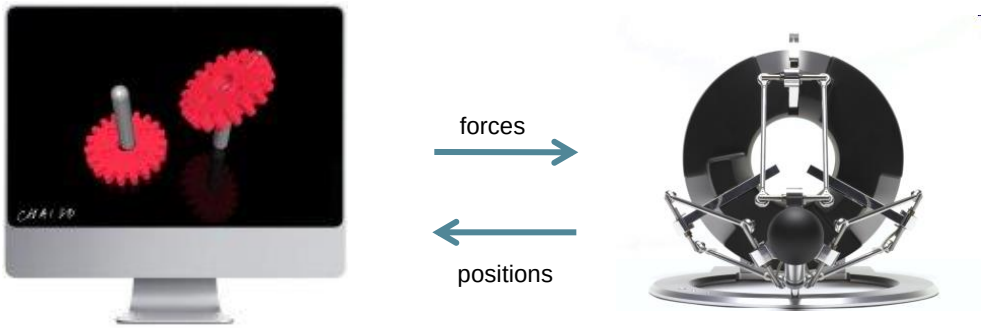




Haptic Device Handler

Haptic Devices

```
class cGenericHapticDevice

int open()
int close()

int initialize()

int getPosition(cVector3d& a_position)
int getLinearVelocity(cVector3d& a_linearVelocity)
int getRotation(cMatrix3d& a_rotation)

int setForce(cVector3d& a_force)
int getUserSwitch(int a_switchIndex, bool& a_status)

cHapticDeviceInfo getSpecifications()
```



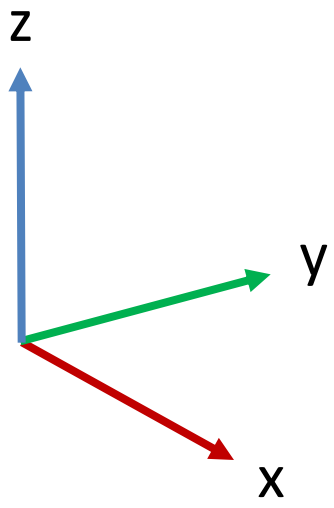
Haptic Devices

```
// create a haptic device handler
handler = new cHapticDeviceHandler();

// get access to the first available haptic device
cGenericHapticDevice* hapticDevice;
handler->getDevice(hapticDevice, 0);

// retrieve information about the current haptic device
cHapticDeviceInfo info;
if (hapticDevice)
{
    info = hapticDevice->getSpecifications();
}

(…)
```

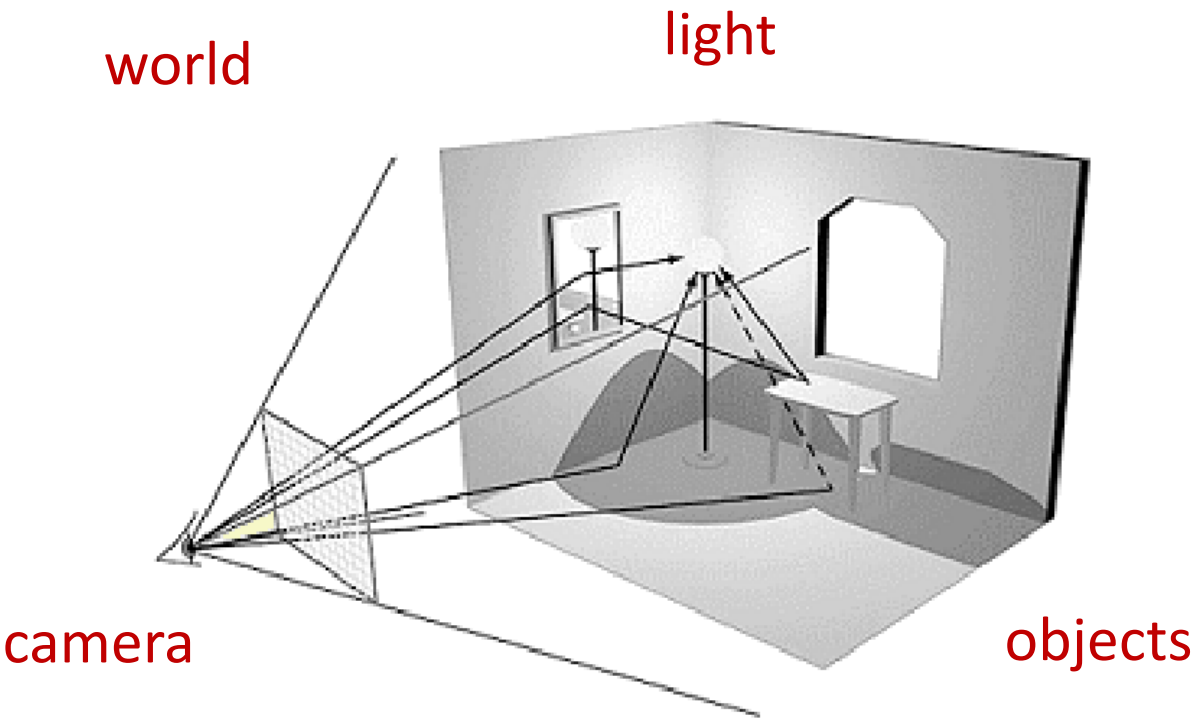
cHapticDeviceInfo

```
string m_manufacturerName;
double m_maxForce;
double m_maxForceStiffness;
double m_workspaceRadius;
bool m_sensedPosition;
bool m_sensedRotation;
bool m_actuatedPosition;
bool m_actuatedRotation;
(…)
```

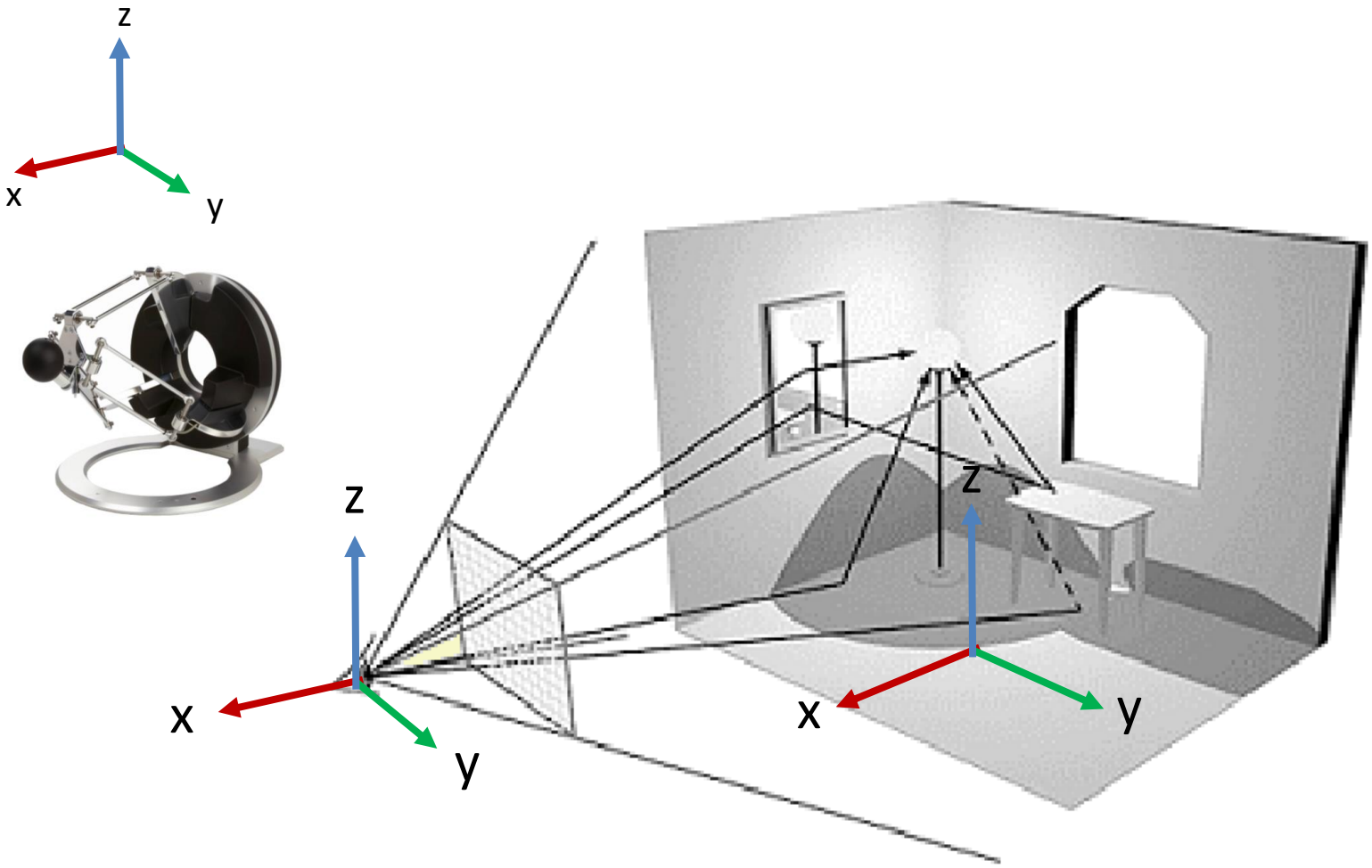
Coordinate System



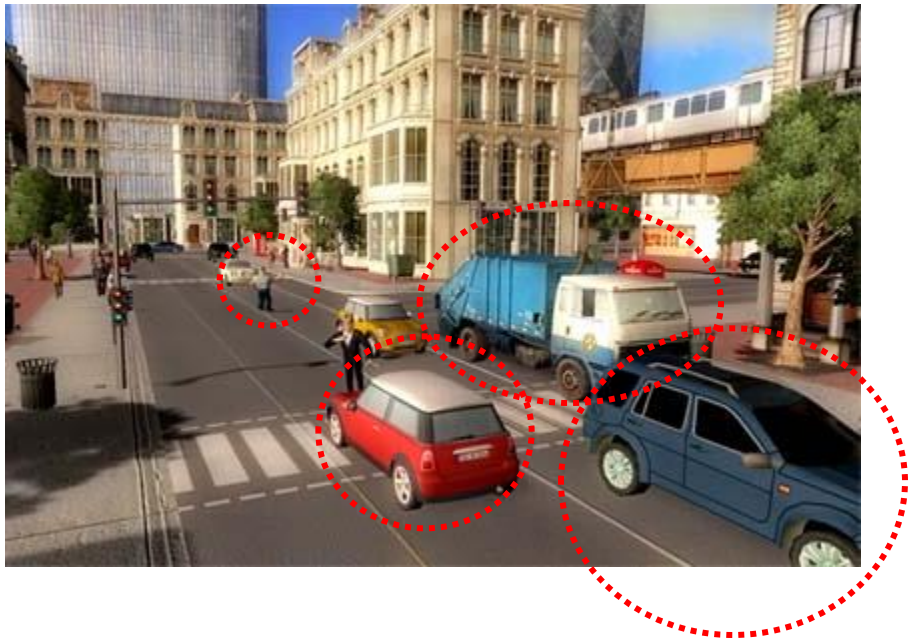
Virtual World

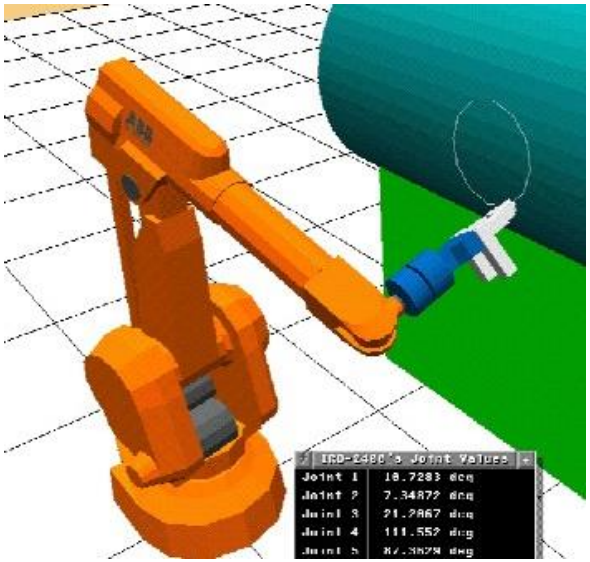


Reference frames



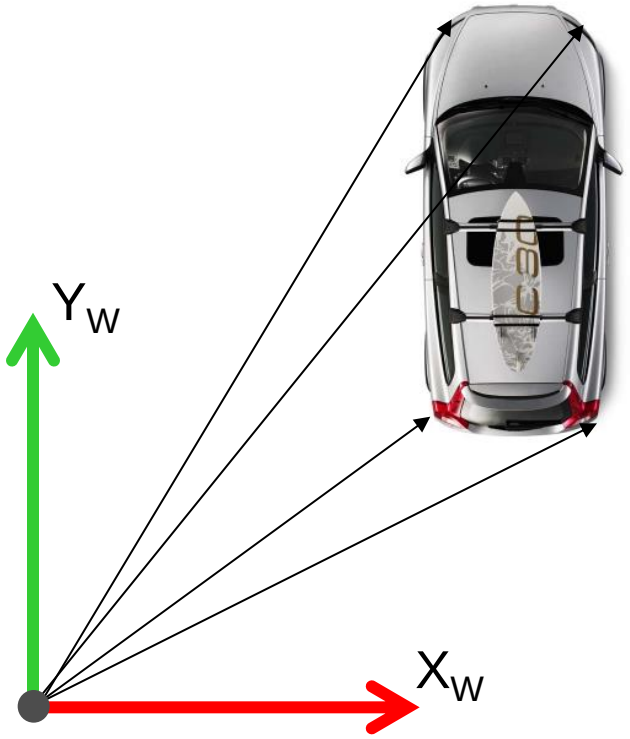
Scene





- Defining independent objects in the world
- Defining relationships between these objects

World Origin

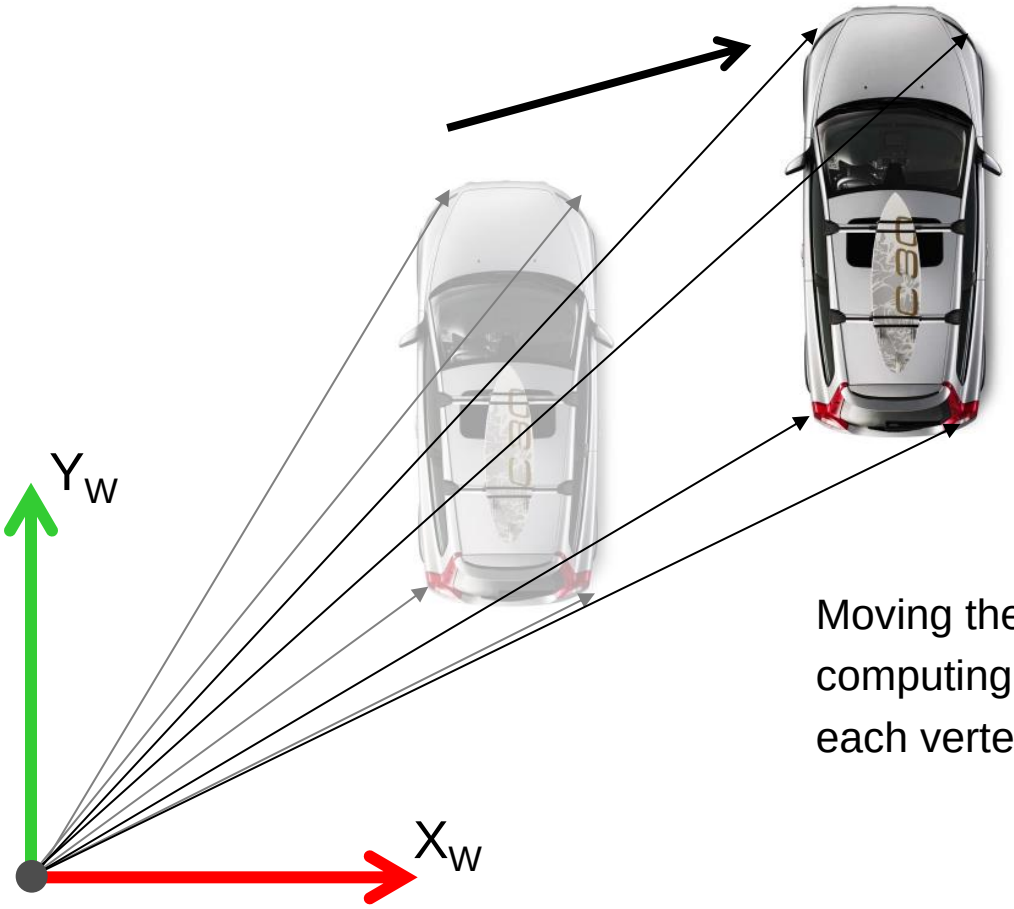


World Origin

Expressing the vertices of the object (car) in reference with the world coordinate system.

{W} : World Reference Frame

Translation

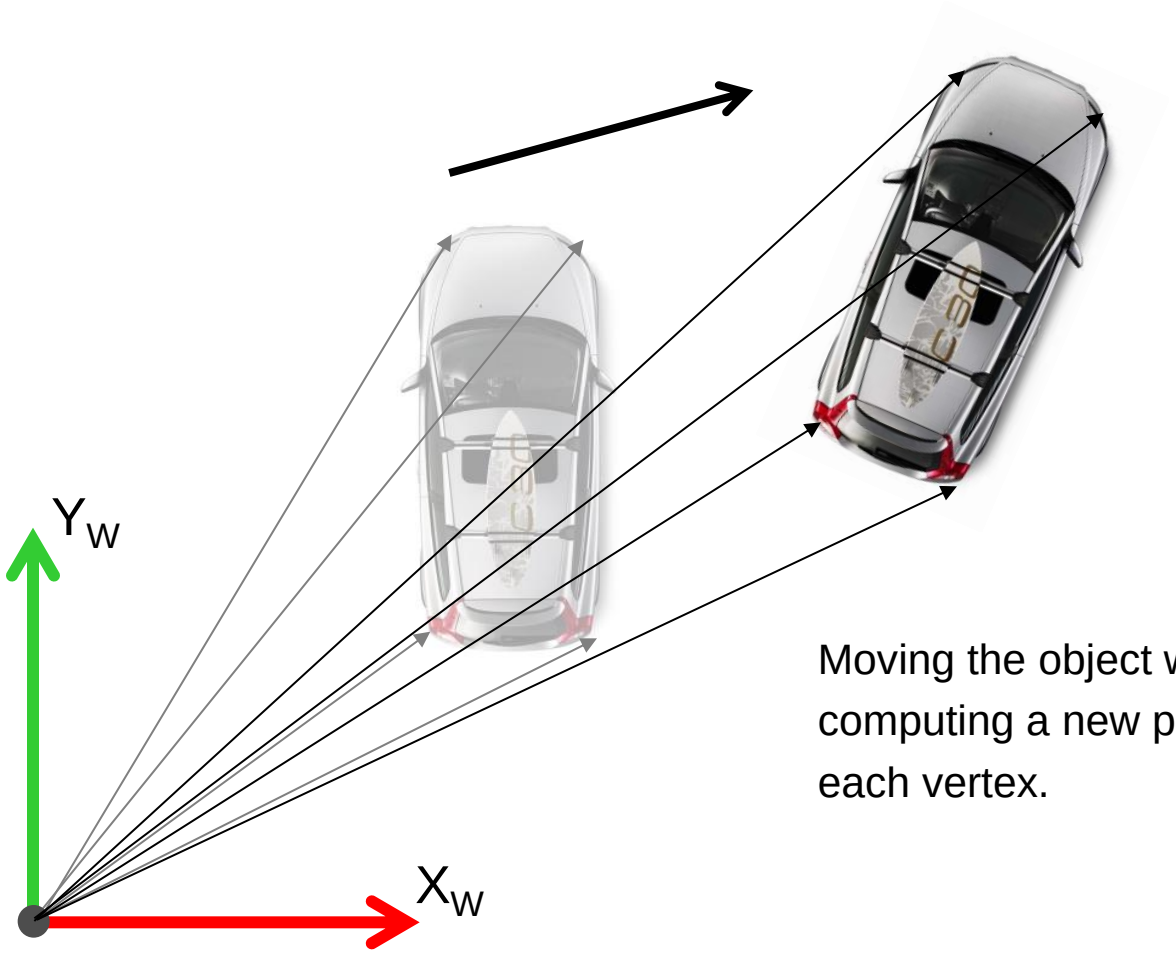


World Origin

Moving the object would require computing a new position for each vertex.

{W} : World Reference Frame

Rotation

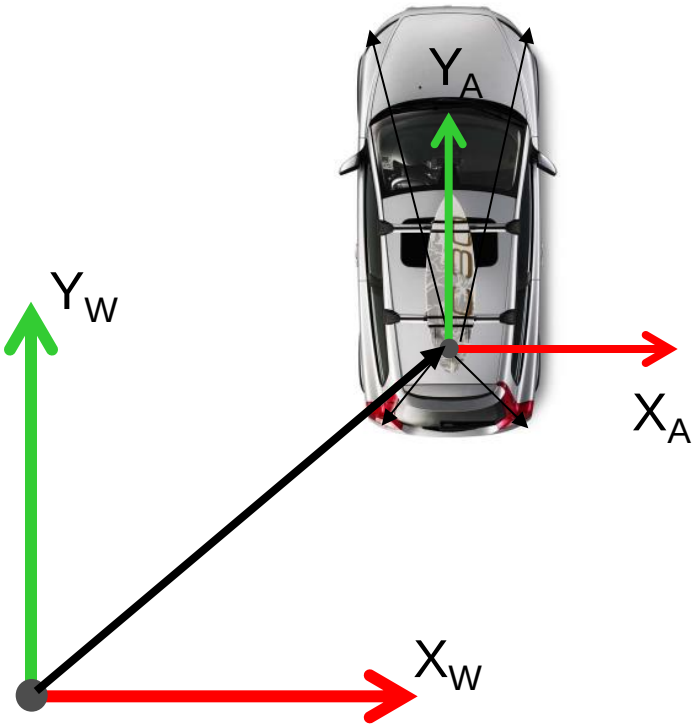


World Origin

Moving the object would require computing a new position for each vertex.

{W} : World Reference Frame

Reference Frames



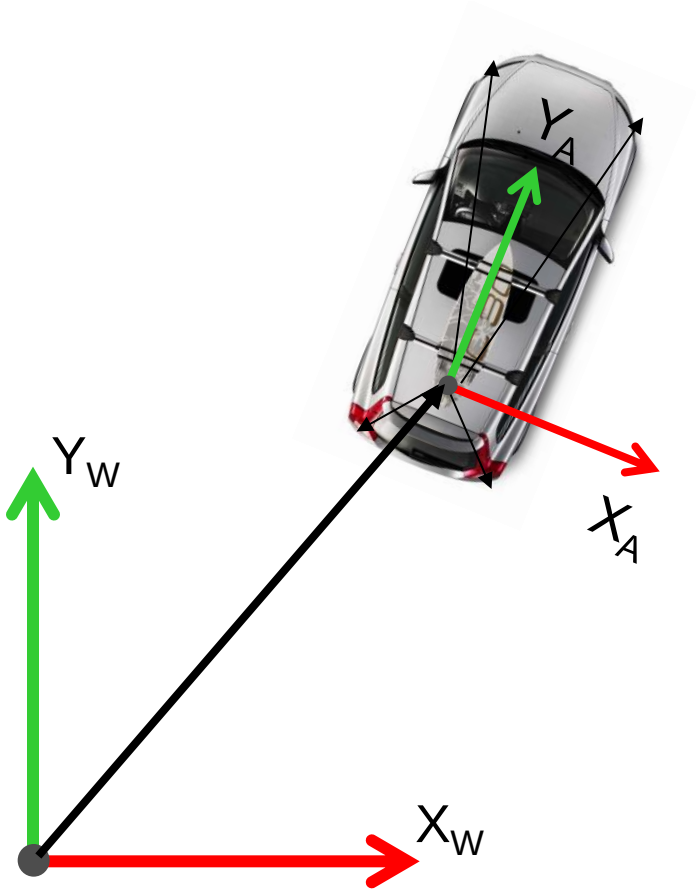
World Origin

Defining a local reference frame for each independent object.

{A} : Reference Frame of Object A

{W} : World Reference Frame

Reference Frames

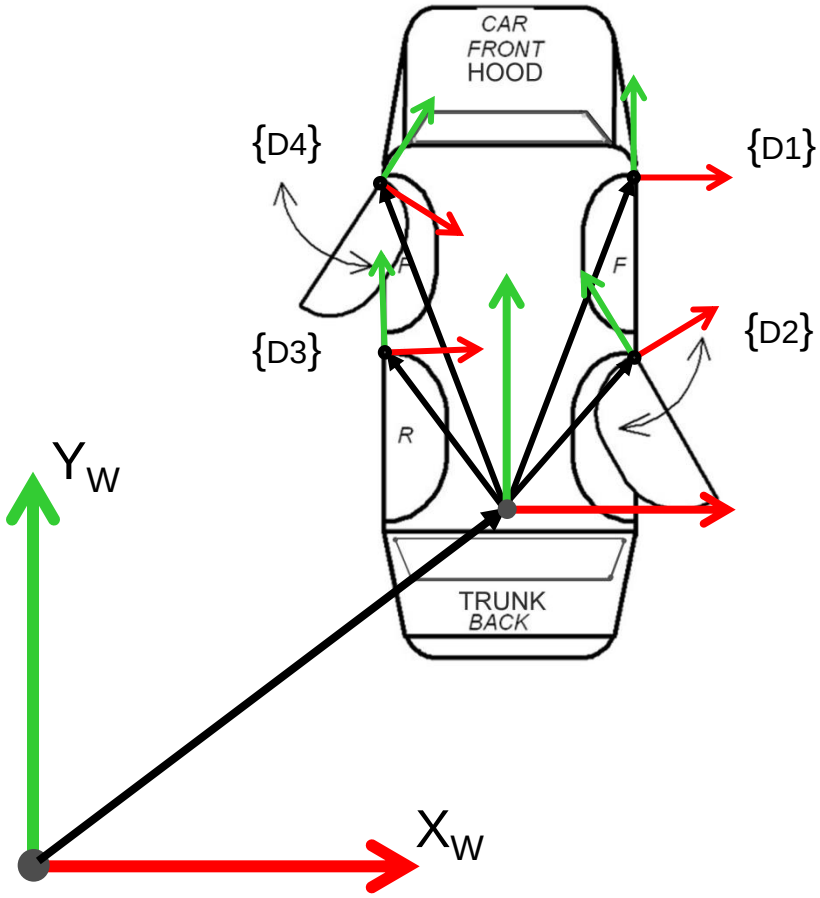


Defining a local reference frame for each independent object.

{A} : Reference Frame of Object A

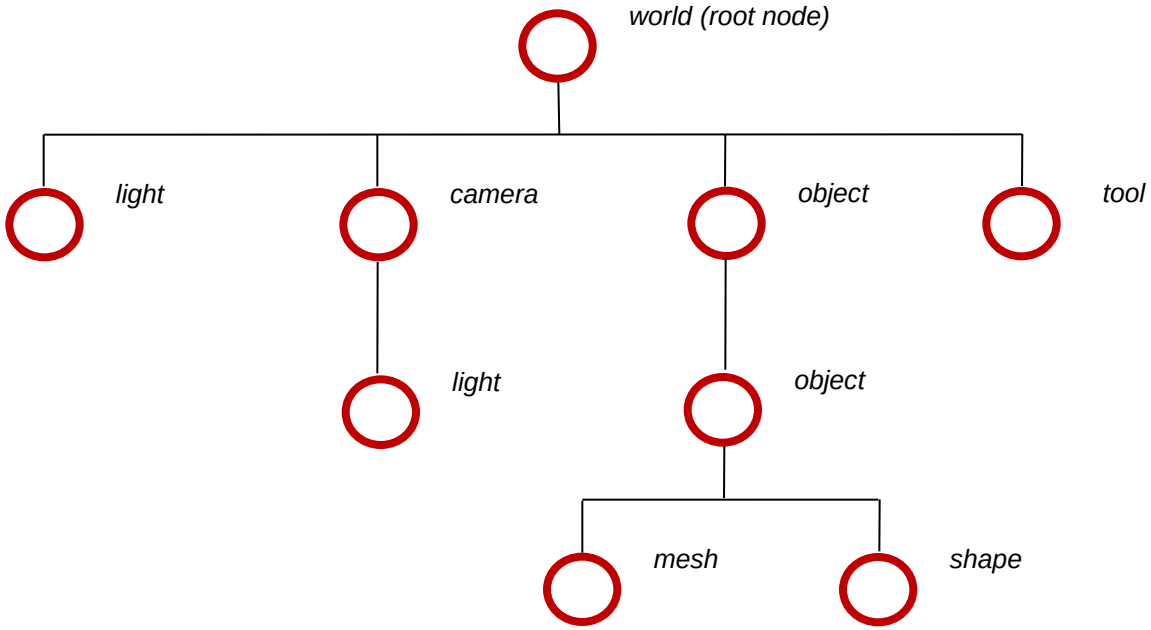
{W} : World Reference Frame

Reference Frames

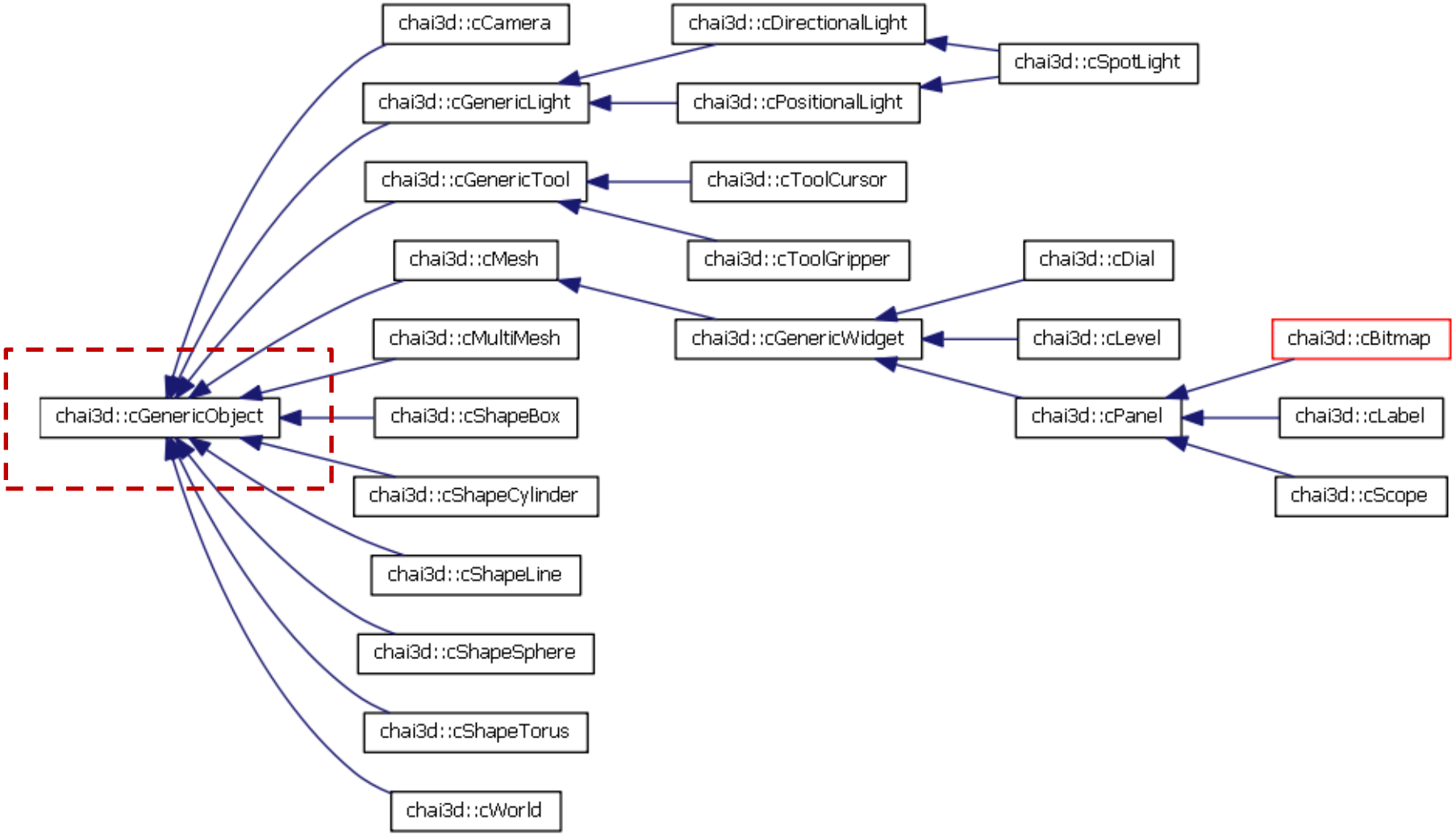


World Origin

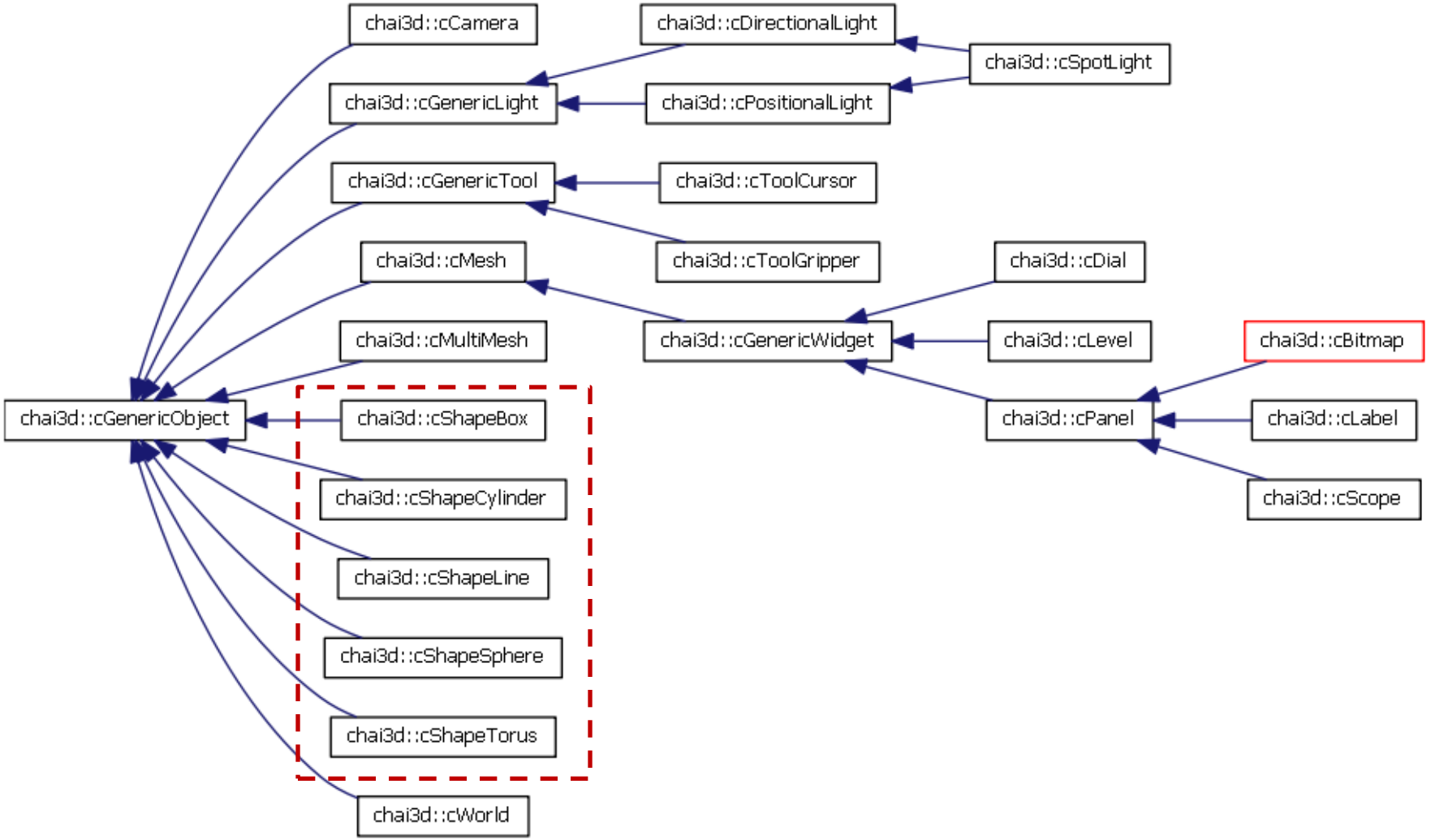
Scene Graph


world (root node)
light
camera
object
tool
light
object
mesh
shape

cGenericObject

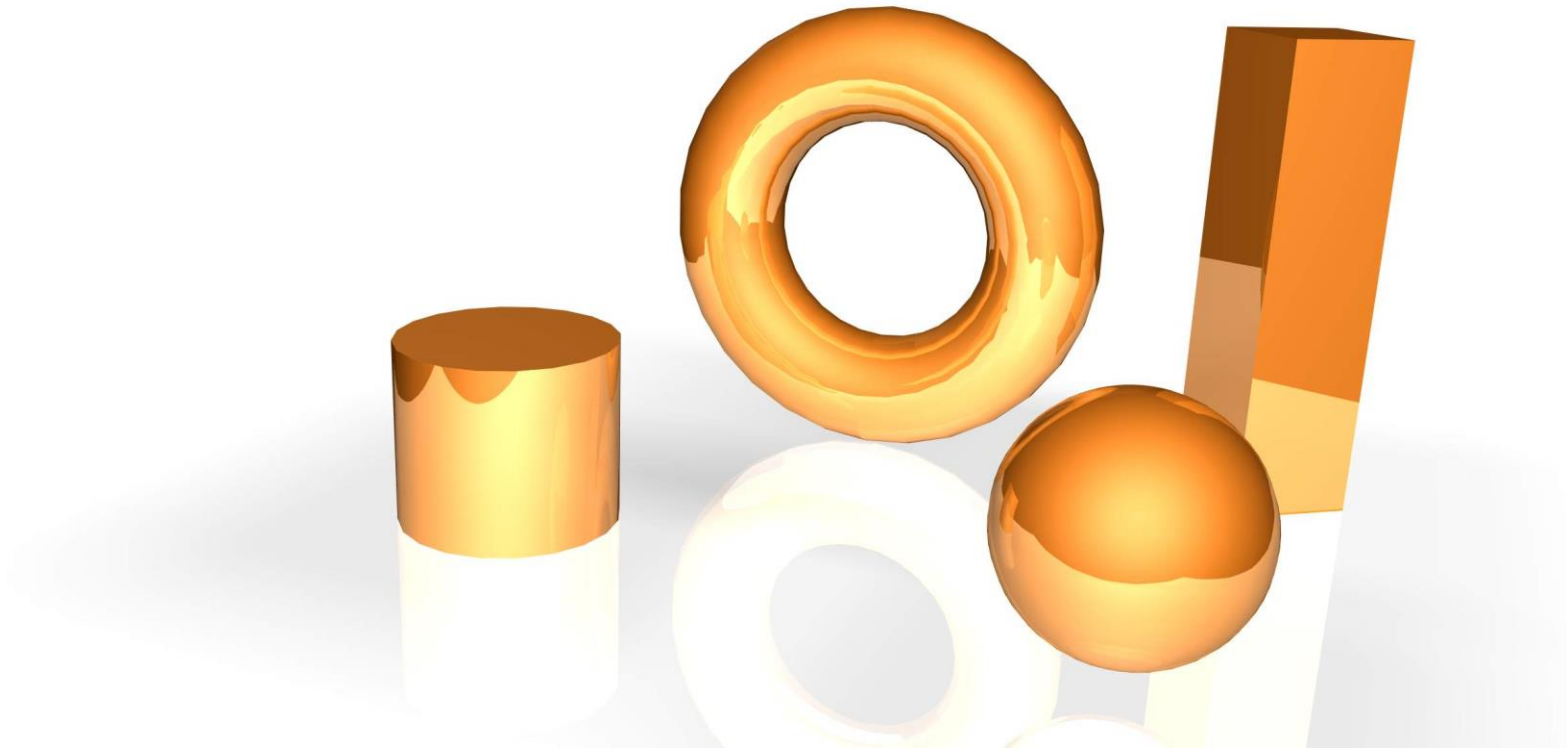


Shapes


chai3d::cCamera
chai3d::cDirectionalLight
chai3d::cSpotLight
chai3d::cGenericLight
chai3d::cPositionalLight
chai3d::cGenericTool
chai3d::cToolCursor
chai3d::cToolGripper
chai3d::cMesh
chai3d::cDial
chai3d::cMultiMesh
chai3d::cGenericWidget
chai3d::cLevel
chai3d::cBitmap
chai3d::cGenericObject
chai3d::cShapeBox
chai3d::cPanel
chai3d::cLabel
chai3d::cShapeCylinder
chai3d::cScope
chai3d::cShapeLine
chai3d::cShapeSphere
chai3d::cShapeTorus
chai3d::cWorld

Shapes

(cShapeSphere, cShapeCylinder, cShapeTorus, cShapeBox)



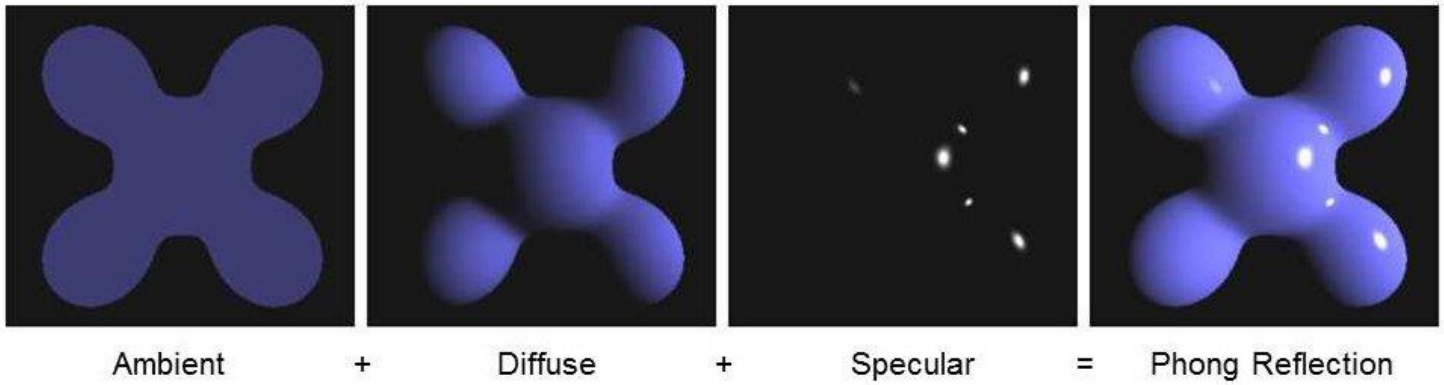
Example

Materials
(cMaterial)

```
GRAPHIC PROPERTIES:
cColorf m_ambient;                  Ambient color.
cColorf m_diffuse;                  Diffuse color.
cColorf m_specular;                 Specular color.
cColorf m_emission;                 Emissive color.
GLuint  m_shininess;                Shininess

HAPTIC PROPERTIES:
double m_viscosity;                 Viscosity constant.
double m_stiffness;                 Stiffness constant.
double m_static_friction;           Static friction constant.
double m_dynamic_friction;          Dynamic friction constant.
double m_vibrationFrequency;        Frequency of vibrations
double m_vibrationAmplitude;        Amplitude of vibrations.
double m_magnetMaxForce;            Maximum force applied by magnet effect.
double m_stickSlipForceMax;         Force threshold for stick and slip effect.
double m_stickSlipStiffness;        Spring stiffness of stick slip model.
```

Materials
(cMaterial)

```
GRAPHIC PROPERTIES:
cColorf m_ambient;          Ambient color.
cColorf m_diffuse;          Diffuse color.
cColorf m_specular;         Specular color.
cColorf m_emission;         Emissive color.
GLuint  m_shininess;        Shininess
```

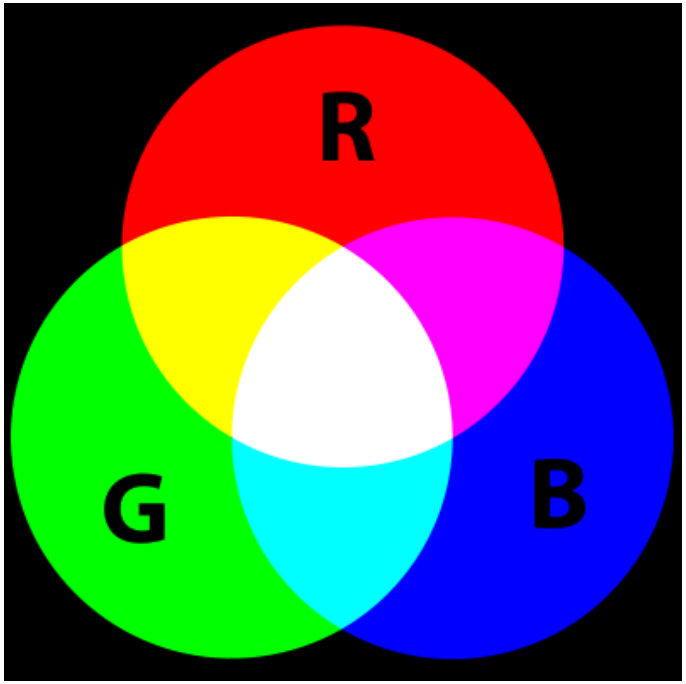


Colors

HTML name	Hex code R G B	Decimal code R G B
Pink colors		
Pink	FF C0 CB	255 192 203
LightPink	FF B6 C1	255 182 193
HotPink	FF 69 B4	255 105 180
DeepPink	FF 14 93	255 20 147
PaleVioletRed	DB 70 93	219 112 147
MediumVioletRed	C7 15 85	199 21 133
Red colors		
LightSalmon	FF A0 7A	255 160 122
Salmon	FA 80 72	250 128 114
DarkSalmon	E9 96 7A	233 150 122
LightCoral	F0 80 80	240 128 128
IndianRed	CD 5C 5C	205 92 92
Crimson	DC 14 3C	220 20 60
FireBrick	B2 22 22	178 34 34
DarkRed	8B 00 00	139 0 0
Red	FF 00 00	255 0 0
Orange colors		
OrangeRed	FF 45 00	255 69 0
Tomato	FF 63 47	255 99 71
Coral	FF 7F 50	255 127 80
DarkOrange	FF 8C 00	255 140 0
Orange	FF A5 00	255 165 0
Gold	FF D7 00	255 215 0
Yellow colors		
Yellow	FF FF 00	255 255 0
LightYellow	FF FF E0	255 255 224
LemonChiffon	FF FA CD	255 250 205
LightGoldenrodYellow	FA FA D2	250 250 210
PapayaWhip	FF EF D5	255 239 213
Moccasin	FF E4 B5	255 228 181
PeachPuff	FF DA B9	255 218 185
PaleGoldenrod	EE E8 AA	238 232 170
Khaki	F0 E6 8C	240 230 140
DarkKhaki	BD B7 6B	189 183 107
Brown colors		
Cornsilk	FF F8 DC	255 248 220
BlanchedAlmond	FF EB CD	255 235 205
Bisque	FF E4 C4	255 228 196
NavajoWhite	FF DE AD	255 222 173
Wheat	F5 DE B3	245 222 179
BurlyWood	DE B8 87	222 184 135
Tan	D2 B4 8C	210 180 140
RosyBrown	BC 8F 8F	188 143 143
SandyBrown	F4 A4 60	244 164 96
Goldenrod	DA A5 20	218 165 32
DarkGoldenrod	B8 86 0B	184 134 11
Peru	CD 85 3F	205 133 63
Chocolate	D2 69 1E	210 105 30
SaddleBrown	8B 45 13	139 69 19
Sienna	A0 52 2D	160 82 45
Brown	A5 2A 2A	165 42 42
Maroon	80 00 00	128 0 0

HTML name	Hex code R G B	Decimal code R G B
Green colors		
DarkOliveGreen	55 6B 2F	85 107 47
Olive	80 80 00	128 128 0
OliveDrab	6B 8E 23	107 142 35
YellowGreen	9A CD 32	154 205 50
LimeGreen	32 CD 32	50 205 50
Lime	00 FF 00	0 255 0
LawnGreen	7C FC 00	124 252 0
Chartreuse	7F FF 00	127 255 0
GreenYellow	AD FF 2F	173 255 47
SpringGreen	00 FF 7F	0 255 127
MediumSpringGreen	00 FA 9A	0 250 154
LightGreen	90 EE 90	144 238 144
PaleGreen	98 FB 98	152 251 152
DarkSeaGreen	8F BC 8F	143 188 143
MediumSeaGreen	3C B3 71	60 179 113
SeaGreen	2E 8B 57	46 139 87
ForestGreen	22 8B 22	34 139 34
Green	00 80 00	0 128 0
DarkGreen	00 64 00	0 100 0
Cyan colors		
MediumAquamarine	66 CD AA	102 205 170
Aqua	00 FF FF	0 255 255
Cyan	00 FF FF	0 255 255
LightCyan	E0 FF FF	224 255 255
PaleTurquoise	AF EE EE	175 238 238
Aquamarine	7F FF D4	127 255 212
Turquoise	40 E0 D0	64 224 208
MediumTurquoise	48 D1 CC	72 209 204
DarkTurquoise	00 CE D1	0 206 209
LightSeaGreen	20 B2 AA	32 178 170
CadetBlue	5F 9E A0	95 158 160
DarkCyan	00 8B 8B	0 139 139
Teal	00 80 80	0 128 128
Blue colors		
LightSteelBlue	B0 C4 DE	176 196 222
PowderBlue	B0 E0 E6	176 224 230
LightBlue	AD D8 E6	173 216 230
SkyBlue	87 CE EB	135 206 235
LightSkyBlue	87 CE FA	135 206 250
DeepSkyBlue	00 BF FF	0 191 255
DodgerBlue	1E 90 FF	30 144 255
CornflowerBlue	64 95 ED	100 149 237
SteelBlue	46 82 B4	70 130 180
RoyalBlue	41 69 E1	65 105 225
Blue	00 00 FF	0 0 255
MediumBlue	00 00 CD	0 0 205
DarkBlue	00 00 8B	0 0 139
Navy	00 00 80	0 0 128
MidnightBlue	19 19 70	25 25 112

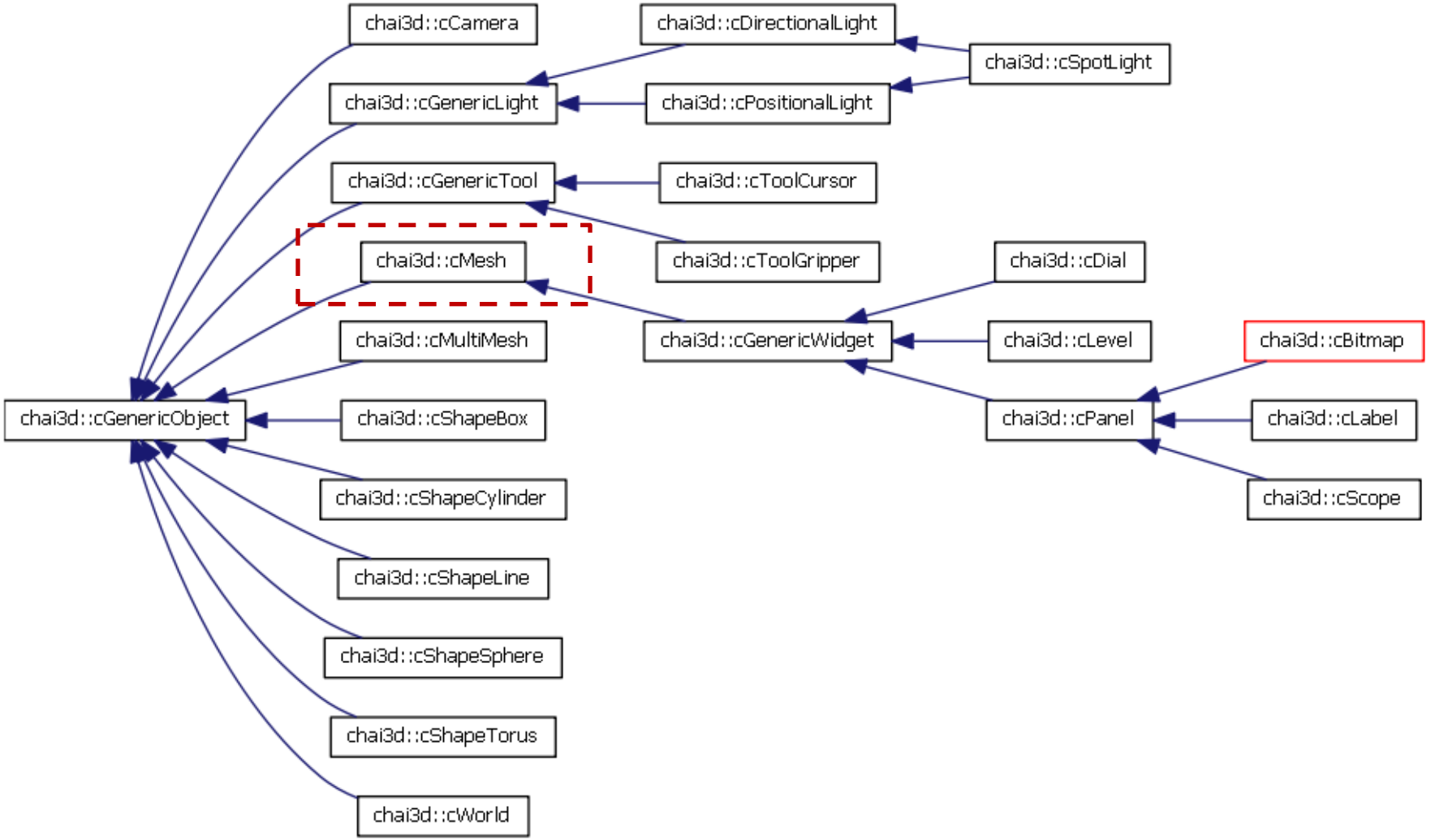
HTML name	Hex code R G B	Decimal code R G B
Purple colors		
Lavender	E6 E6 FA	230 230 250
Thistle	D8 BF D8	216 191 216
Plum	DD A0 DD	221 160 221
Violet	EE 82 EE	238 130 238
Orchid	DA 70 D6	218 112 214
Fuchsia	FF 00 FF	255 0 255
Magenta	FF 00 FF	255 0 255
MediumOrchid	BA 55 D3	186 85 211
MediumPurple	93 70 DB	147 112 219
BlueViolet	8A 2B E2	138 43 226
DarkViolet	94 00 D3	148 0 211
DarkOrchid	99 32 CC	153 50 204
DarkMagenta	8B 00 8B	139 0 139
Purple	80 00 80	128 0 128
Indigo	4B 00 82	75 0 130
DarkSlateBlue	48 3D 8B	72 61 139
SlateBlue	6A 5A CD	106 90 205
MediumSlateBlue	7B 68 EE	123 104 238
White/Gray/Grey/Black colors		
White	FF FF FF	255 255 255
Snow	FF FA FA	255 250 250
Honeydew	F0 FF F0	240 255 240
MintCream	F5 FF FA	245 255 250
Azure	F0 FF FF	240 255 255
AliceBlue	F0 F8 FF	240 248 255
GhostWhite	F8 F8 FF	248 248 255
WhiteSmoke	F5 F5 F5	245 245 245
Seashell	FF F5 EE	255 245 238
Beige	F5 F5 DC	245 245 220
OldLace	FD F5 E6	253 245 230
FloralWhite	FF FA F0	255 250 240
Ivory	FF FF F0	255 255 240
AntiqueWhite	FA EB D7	250 235 215
Linen	FA F0 E6	250 240 230
LavenderBlush	FF F0 F5	255 240 245
MistyRose	FF E4 E1	255 228 225
Gainsboro	DC DC DC	220 220 220
LightGray	D3 D3 D3	211 211 211
LightGrey	D3 D3 D3	211 211 211
Silver	C0 C0 C0	192 192 192
DarkGray	A9 A9 A9	169 169 169
DarkGrey	A9 A9 A9	169 169 169
Gray	80 80 80	128 128 128
Grey	80 80 80	128 128 128
DimGray	69 69 69	105 105 105
DimGrey	69 69 69	105 105 105
LightSlateGray	77 88 99	119 136 153
LightSlateGrey	77 88 99	119 136 153
SlateGray	70 80 90	112 128 144
SlateGrey	70 80 90	112 128 144
DarkSlateGray	2F 4F 4F	47 79 79
DarkSlateGrey	2F 4F 4F	47 79 79
Black	00 00 00	0 0 0



Colors
(cColorf)

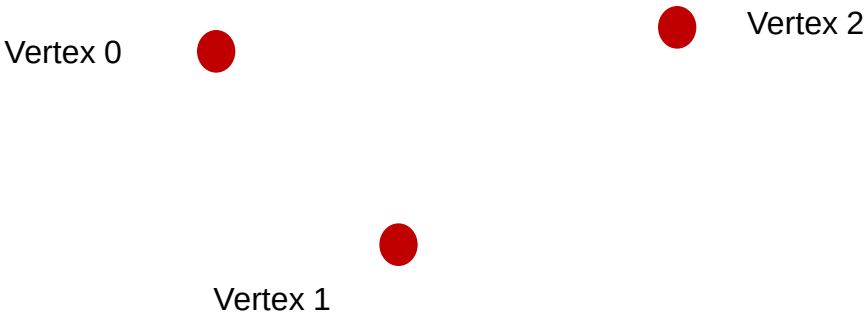
```
// set material color by name
object->m_material->setBlueCornflower();

// set material color by values (R-G-B)
object->m_material->setColorf(0.2, 0.1, 0.1);

// set material color by components (R-G-B)
object->m_material->m_ambient->set(0.2, 0.2, 0.2);
object->m_material->m_diffuse>set(0.5, 0.5, 0.5);
object->m_material->m_specular>set(1.0, 1.0, 1.0);


// define color
cColorf color;
color.setRedFireBrick();
```

Mesh


chai3d::cCamera
chai3d::cDirectionalLight
chai3d::cSpotLight
chai3d::cGenericLight
chai3d::cPositionalLight
chai3d::cGenericTool
chai3d::cToolCursor
chai3d::cMesh
chai3d::cToolGripper
chai3d::cDial
chai3d::cMultiMesh
chai3d::cGenericWidget
chai3d::cLevel
chai3d::cBitmap
chai3d::cGenericObject
chai3d::cShapeBox
chai3d::cPanel
chai3d::cLabel
chai3d::cShapeCylinder
chai3d::cScope
chai3d::cShapeLine
chai3d::cShapeSphere
chai3d::cShapeTorus
chai3d::cWorld

Creating Vertices

```
// create mesh
cMesh* mesh = new cMesh();

// add mesh to world
world->addChild(mesh);

// create 3 vertices
unsigned int vertexIndex0 = mesh->m_vertices->newVertex();
unsigned int vertexIndex1 = mesh->m_vertices->newVertex();
unsigned int vertexIndex2 = mesh->m_vertices->newVertex();
```

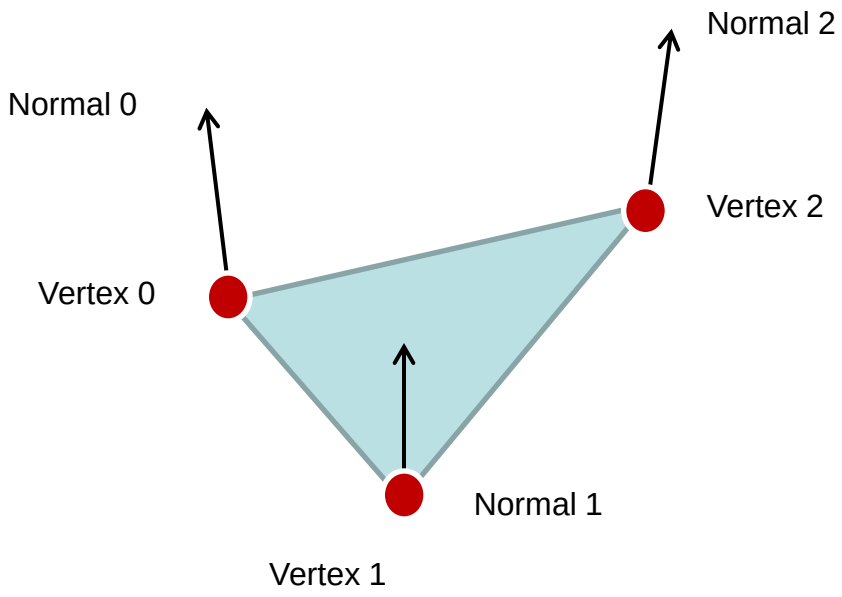


Vertex Properties

```
// define vertex position
cVector3d position(2.0, 3.0, 4.0);
mesh->m_vertices->setLocalPos(vertexIndex0, position);

// define vertex normal
cVector3d normal(1.0, 0.0, 0.0);
mesh->m_vertices->setNormal(vertexIndex0, normal);

// define texture coordinate
mesh->m_vertices->setTexCoord(vertexIndex0, 0.2, 0.3);

// define vertex color
cColorf color;
color.setBlueCadet();
mesh->m_vertices->setColor(vertexIndex0, color);
```

Creating Triangles

```
// create triangle
mesh->m_triangles->newTriangle(vertexIndex0, vertexIndex1, vertexIndex2);
```

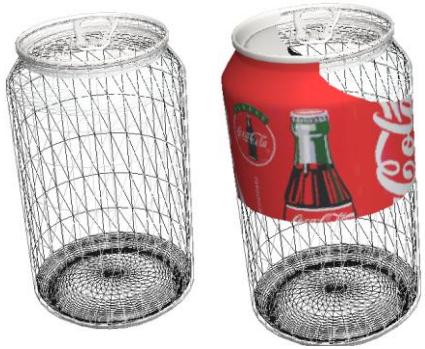


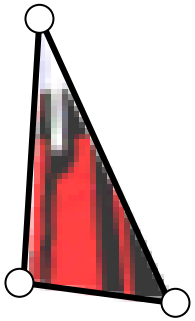
Textures

mesh object (cMesh)

texture map (cTexture2D)

mesh object with
texture properties defined



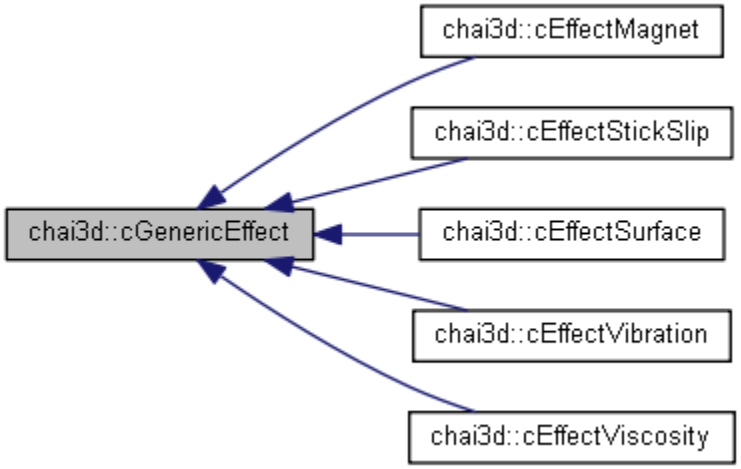


single textured triangle
and its 3 vertices

cMultiMesh

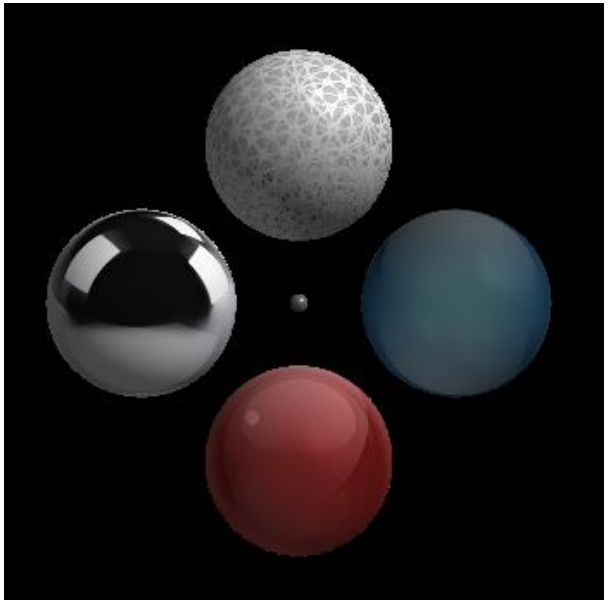
Haptic Effects
(cGenericEffect)

```
HAPTIC PROPERTIES: (cMaterial)
double m_viscosity;                 Viscosity constant.
double m_stiffness;                 Stiffness constant.
double m_static_friction;           Static friction constant.
double m_dynamic_friction;          Dynamic friction constant.
double m_vibrationFrequency;        Frequency of vibrations
double m_vibrationAmplitude;        Amplitude of vibrations.
double m_magnetMaxForce;            Maximum force applied by magnet effect.
double m_stickSlipForceMax;         Force threshold for stick and slip effect.
double m_stickSlipStiffness;        Spring stiffness of stick slip model.
```

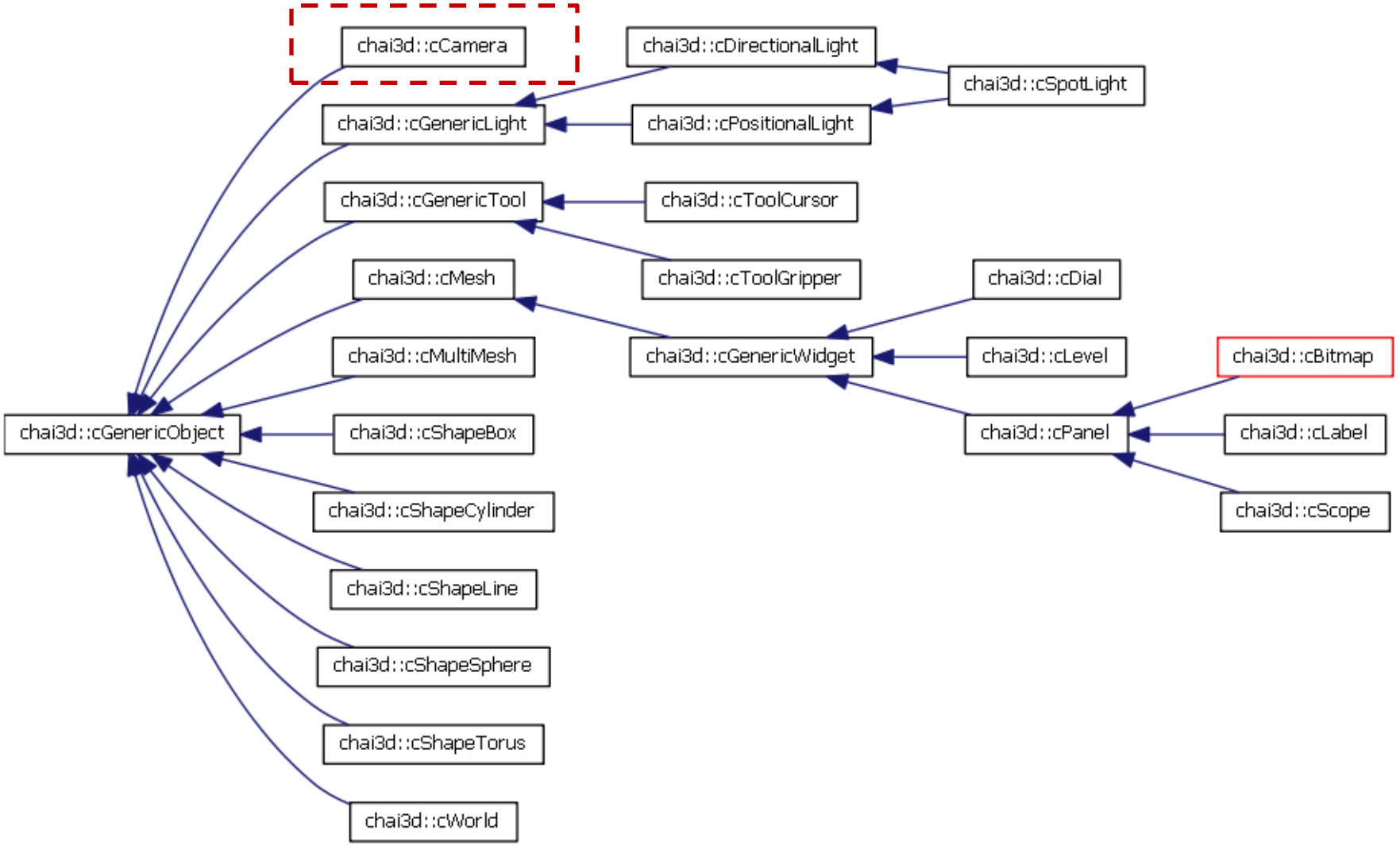


Haptic Effects
(cGenericEffect)

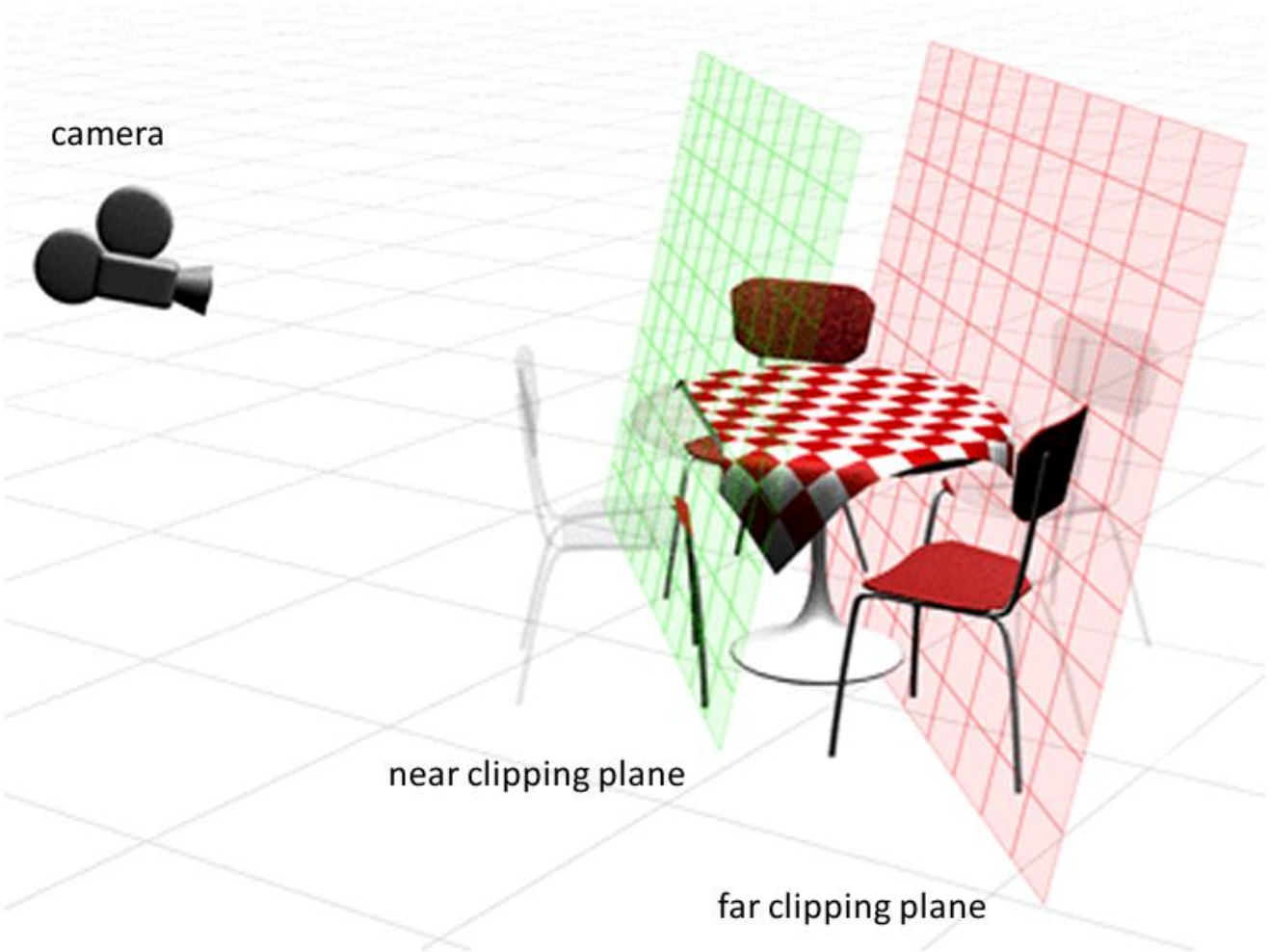
```
// create a haptic vibration effect
object->createEffectVibration();

// create a haptic surface effect
object->createEffectSurface();
```



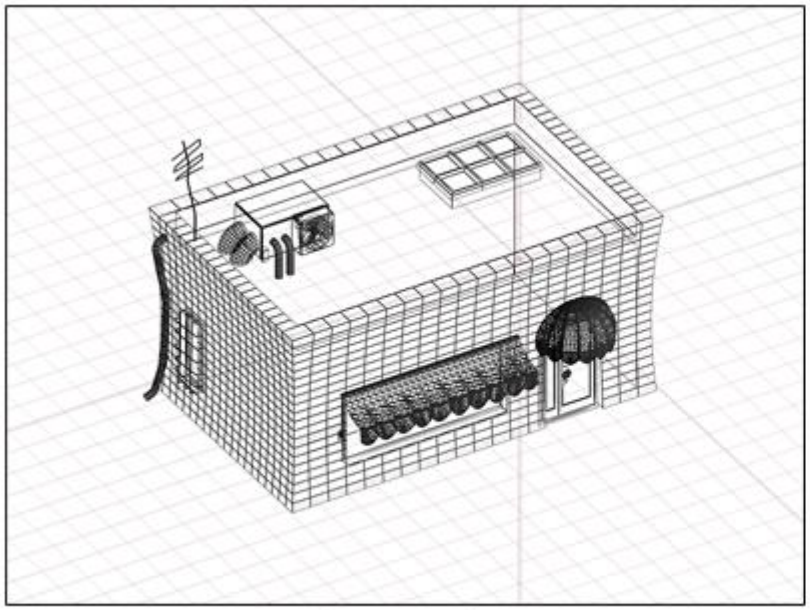
Camera


chai3d::cCamera
chai3d::cDirectionalLight
chai3d::cSpotLight
chai3d::cGenericLight
chai3d::cPositionalLight
chai3d::cGenericTool
chai3d::cToolCursor
chai3d::cToolGripper
chai3d::cMesh
chai3d::cDial
chai3d::cMultiMesh
chai3d::cGenericWidget
chai3d::cLevel
chai3d::cBitmap
chai3d::cGenericObject
chai3d::cShapeBox
chai3d::cPanel
chai3d::cLabel
chai3d::cShapeCylinder
chai3d::cScope
chai3d::cShapeLine
chai3d::cShapeSphere
chai3d::cShapeTorus
chai3d::cWorld

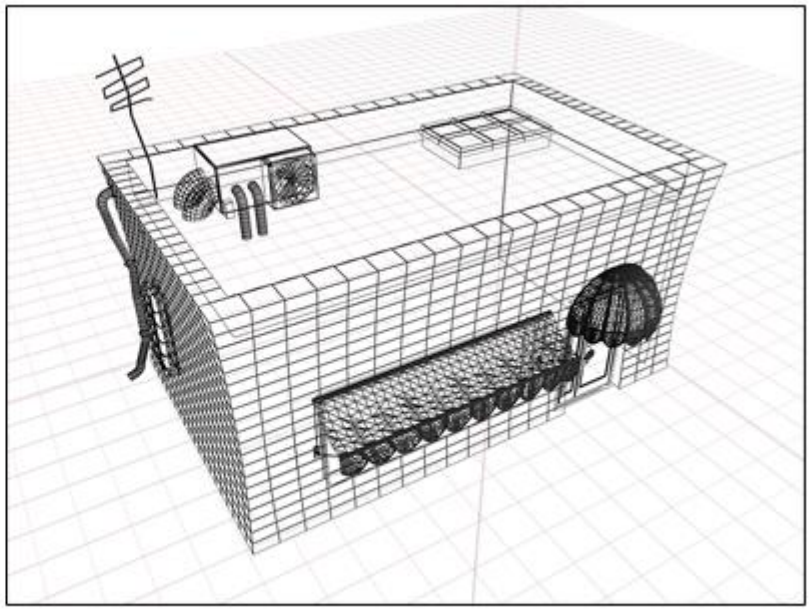
Camera



Camera



orthographic

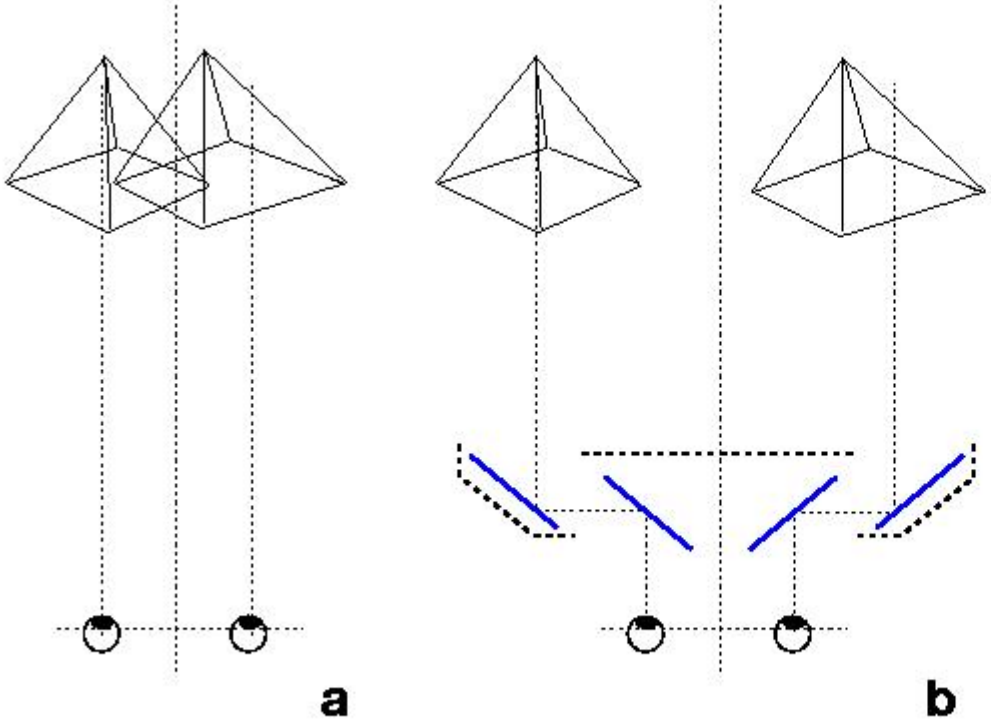


perspective

Camera
(cCamera)

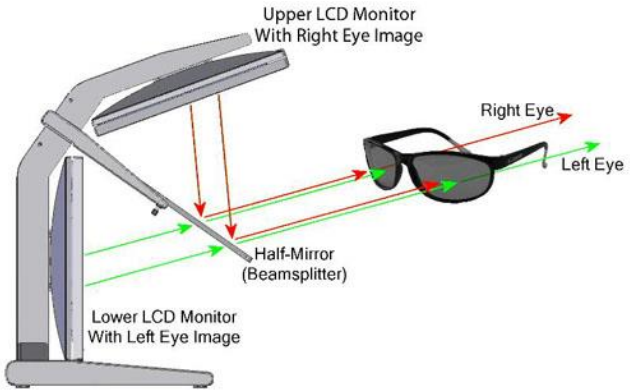
```
// create a camera and insert it into the virtual world
camera = new cCamera(world);
world->addChild(camera);

// position and orient the camera
camera->set( cVector3d (0.5, 0.0, 0.0),    // camera position (eye)
             cVector3d (0.0, 0.0, 0.0),    // look at position (target)
             cVector3d (0.0, 0.0, 1.0));   // direction of the (up) vector

// set the near and far clipping planes of the camera
camera->setClippingPlanes(0.01, 10.0);
```

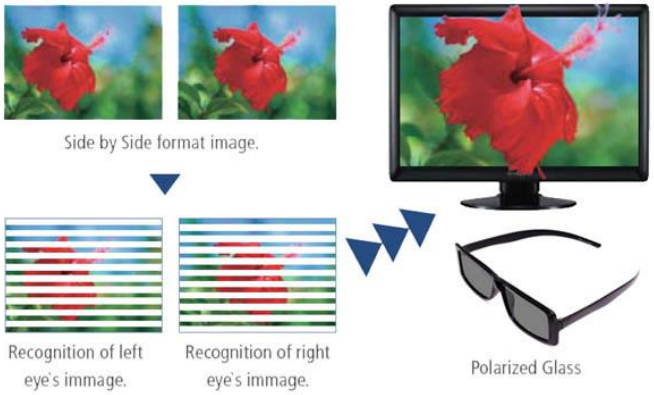
Stereo Camera



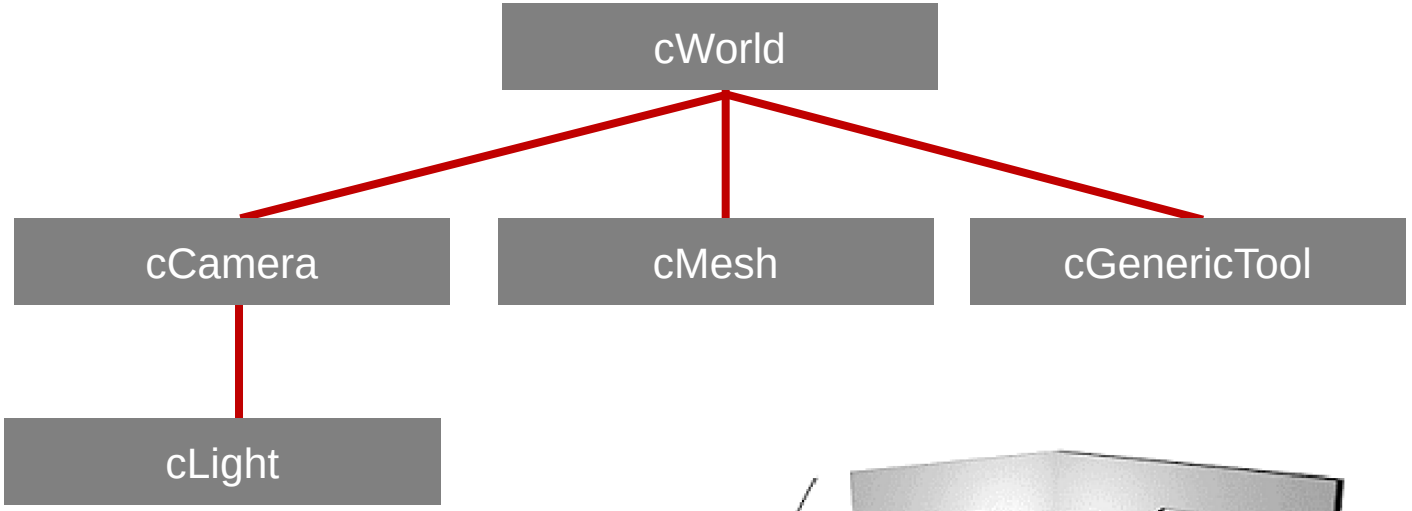
Stereo Camera
(cCamera)

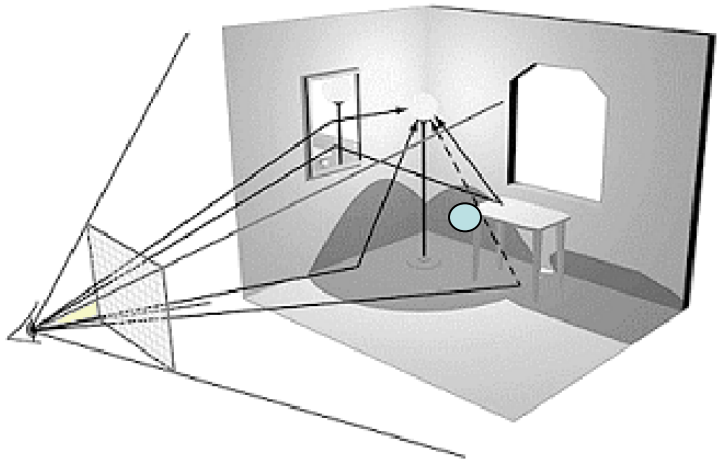
```
// set stereo mode
camera->setStereoMode(stereoMode);

// set stereo eye separation and focal length (applies only if stereo is enabled)
camera->setStereoEyeSeparation(0.005);
camera->setStereoFocalLength(0.5);

// set vertical mirrored display mode
camera->setMirrorVertical(mirroredDisplay);
```



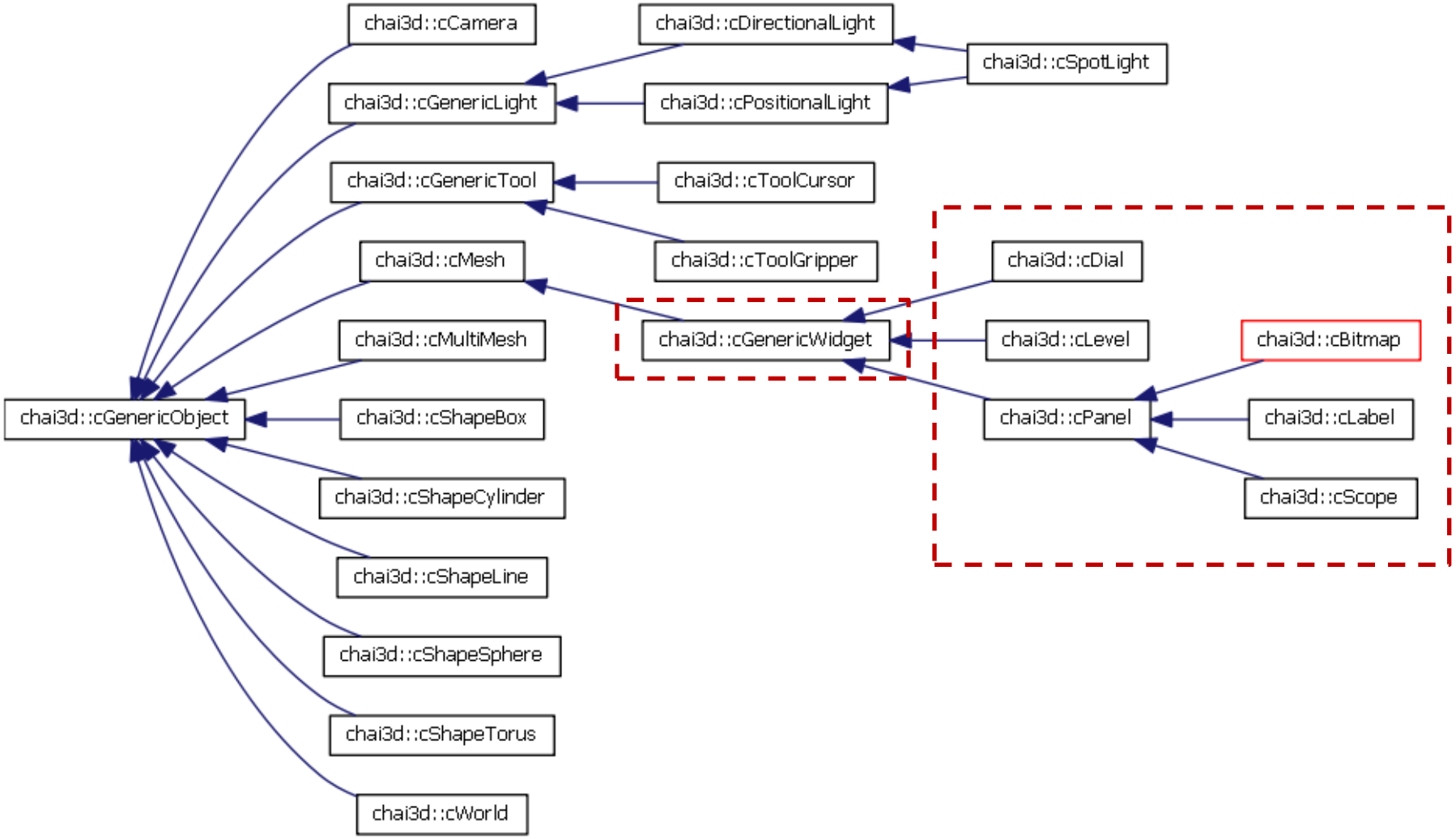


Example

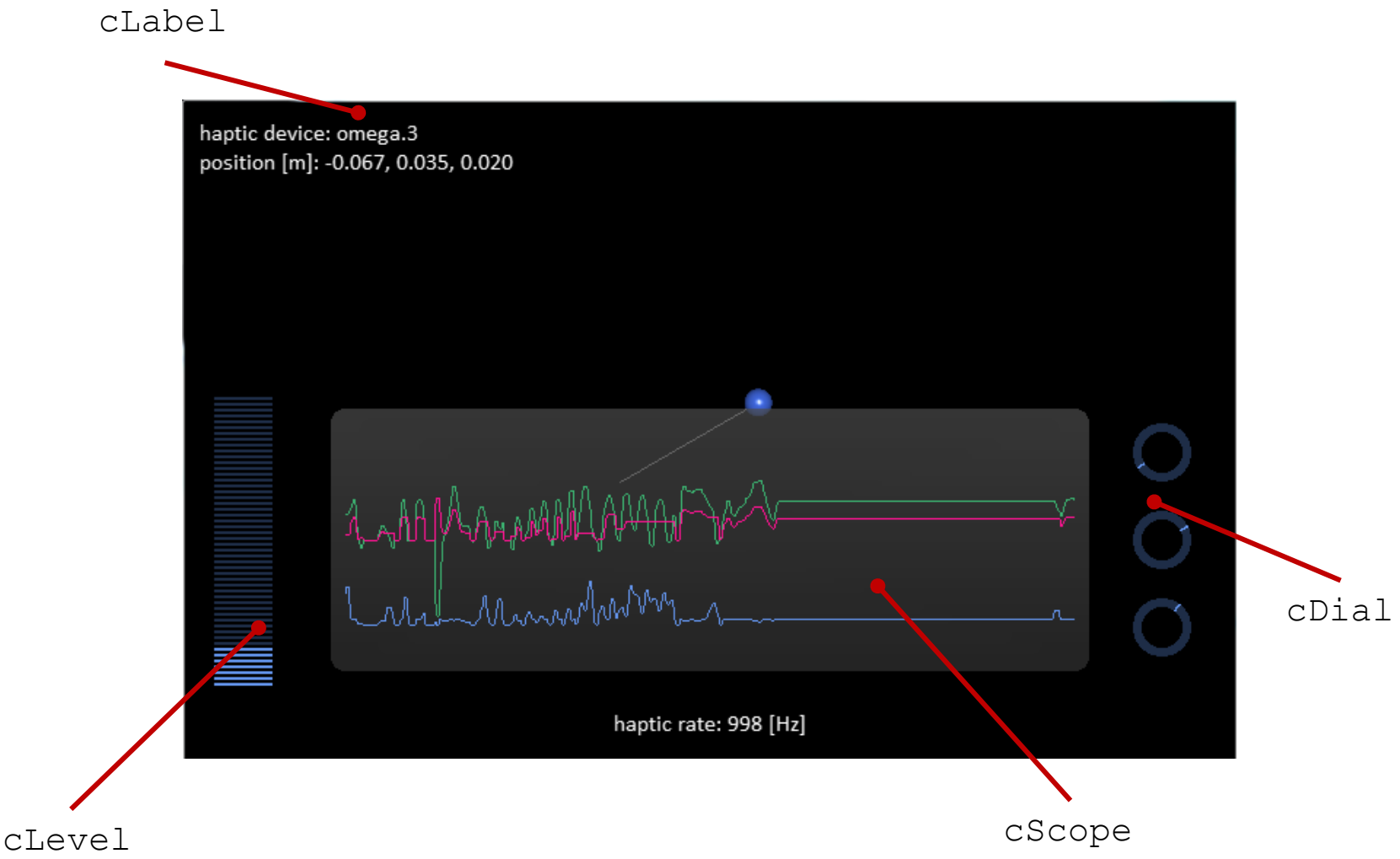




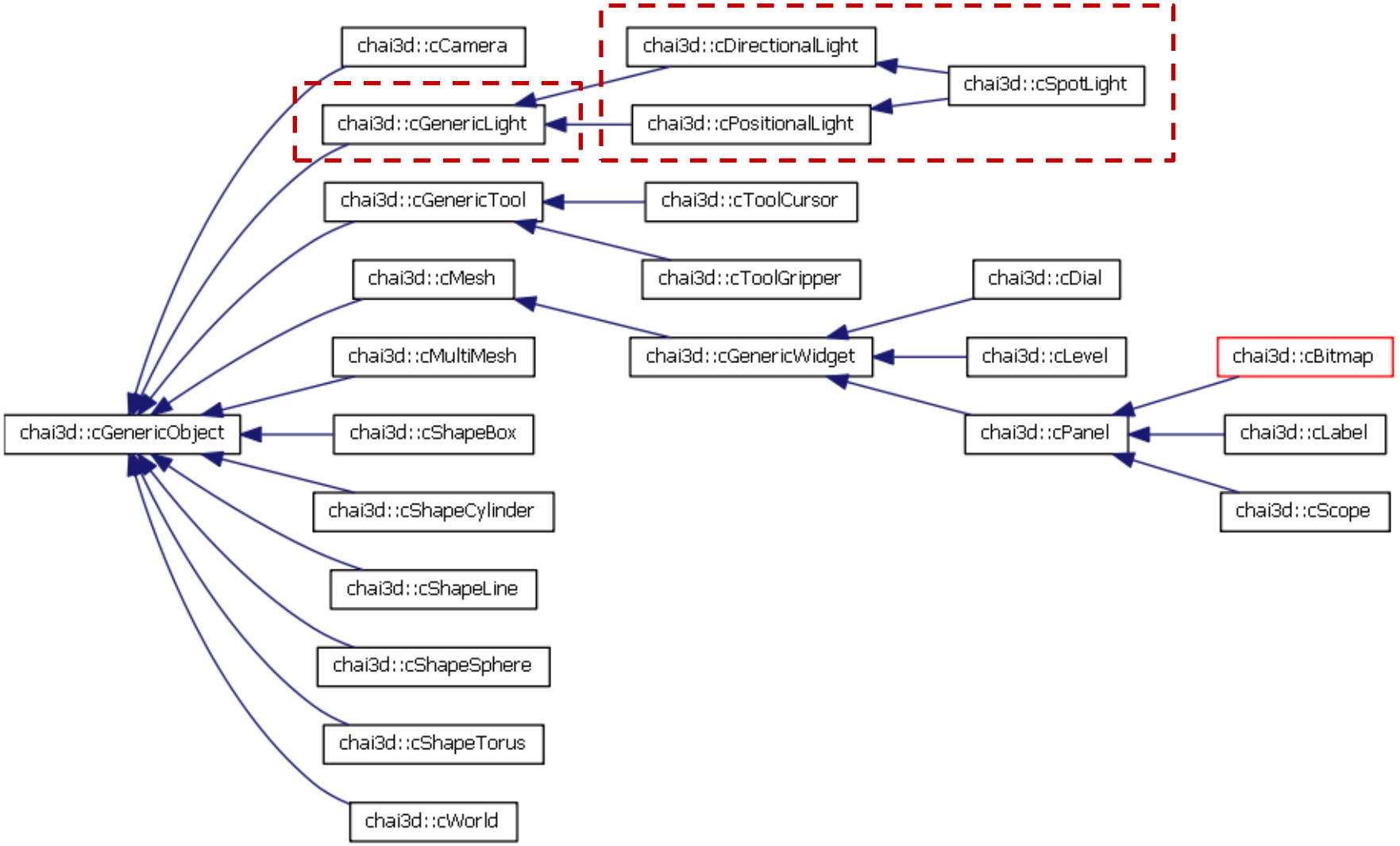
Widgets


chai3d::cCamera
chai3d::cDirectionalLight
chai3d::cSpotLight
chai3d::cGenericLight
chai3d::cPositionalLight
chai3d::cGenericTool
chai3d::cToolCursor
chai3d::cToolGripper
chai3d::cMesh
chai3d::cDial
chai3d::cGenericWidget
chai3d::cLevel
chai3d::cBitmap
chai3d::cMultiMesh
chai3d::cGenericObject
chai3d::cShapeBox
chai3d::cPanel
chai3d::cLabel
chai3d::cShapeCylinder
chai3d::cScope
chai3d::cShapeLine
chai3d::cShapeSphere
chai3d::cShapeTorus
chai3d::cWorld

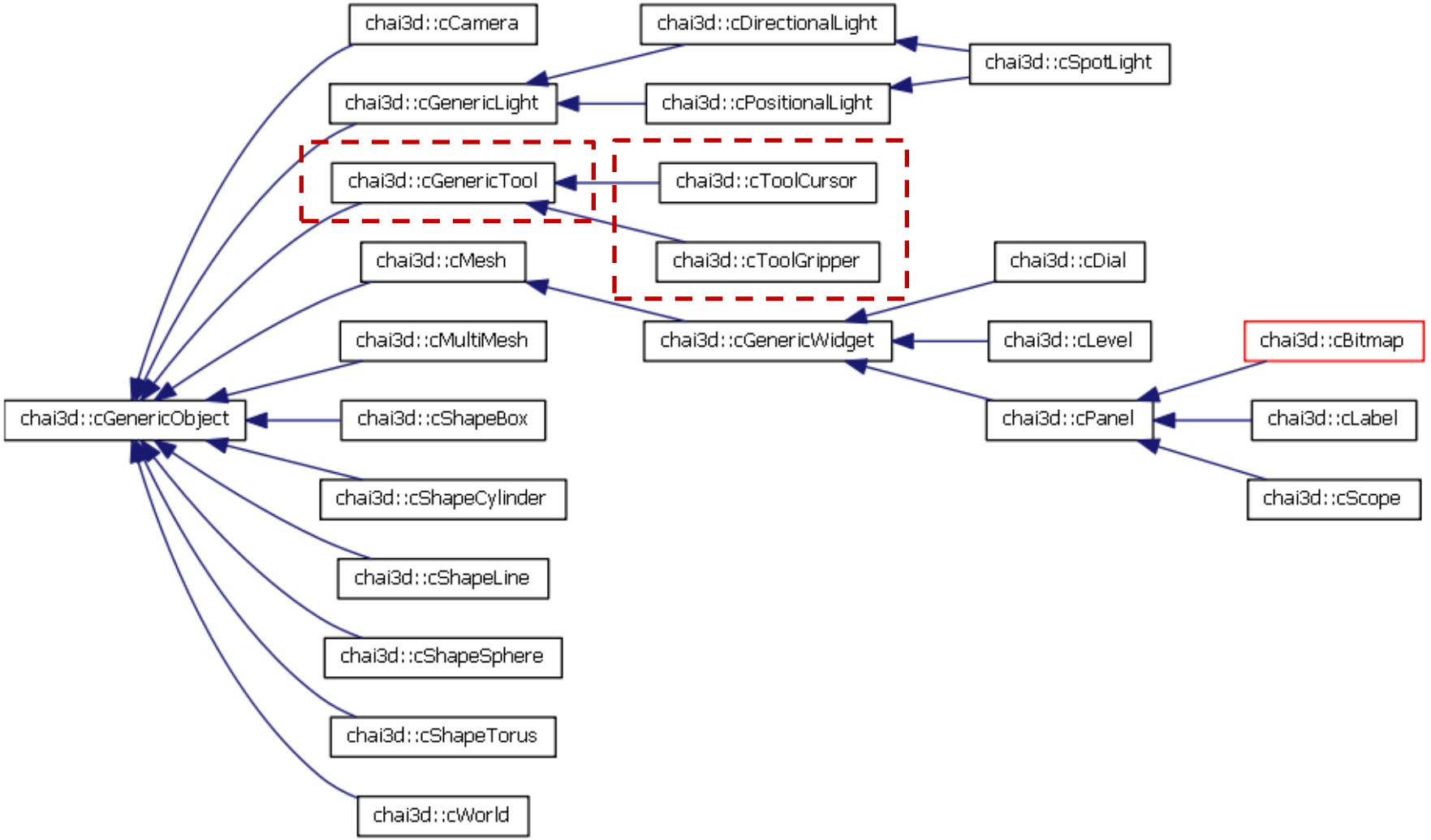
Widgets


cLabel
haptic device: omega.3
position [m]: -0.067, 0.035, 0.020
haptic rate: 998 [Hz]
cDial
cLevel
cScope

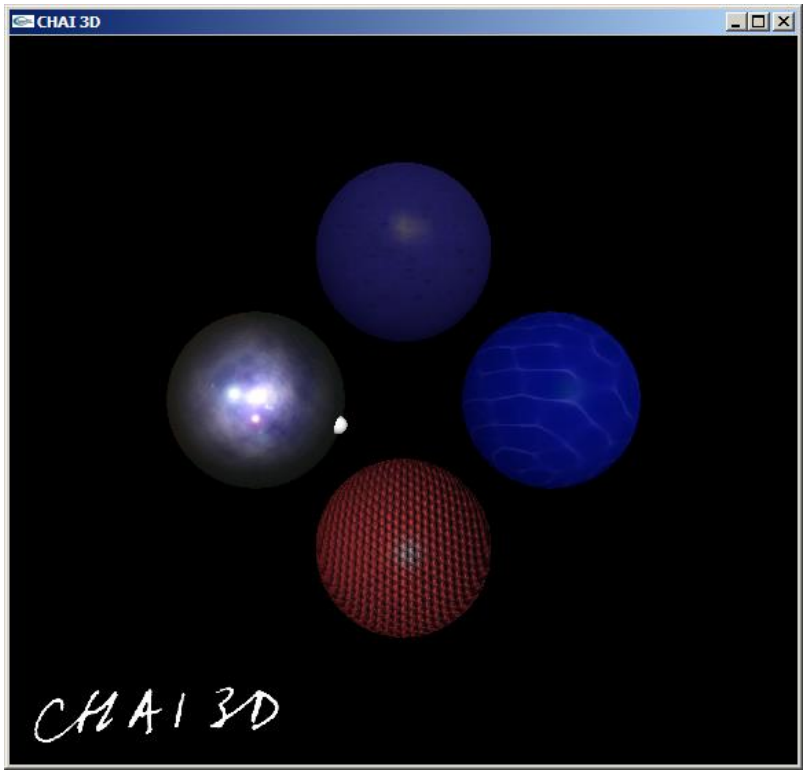
Light Sources



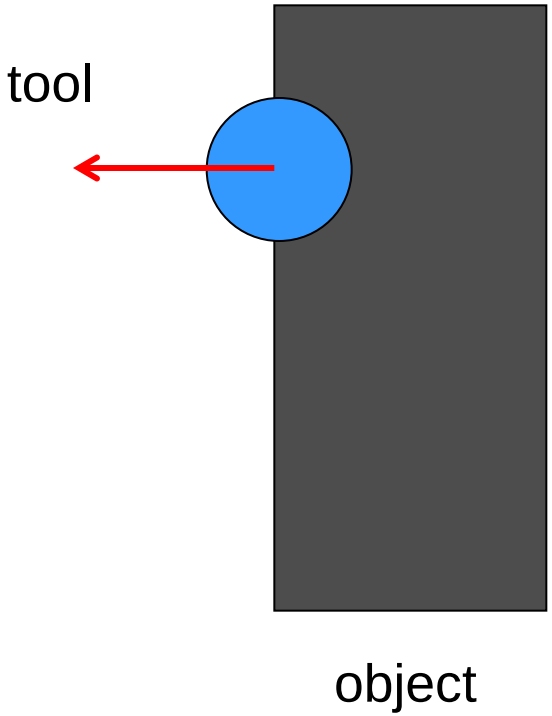
Tools


chai3d::cCamera
chai3d::cDirectionalLight
chai3d::cSpotLight
chai3d::cGenericLight
chai3d::cPositionalLight
chai3d::cGenericTool
chai3d::cToolCursor
chai3d::cToolGripper
chai3d::cMesh
chai3d::cDial
chai3d::cMultiMesh
chai3d::cGenericWidget
chai3d::cLevel
chai3d::cBitmap
chai3d::cGenericObject
chai3d::cShapeBox
chai3d::cPanel
chai3d::cLabel
chai3d::cShapeCylinder
chai3d::cScope
chai3d::cShapeLine
chai3d::cShapeSphere
chai3d::cShapeTorus
chai3d::cWorld

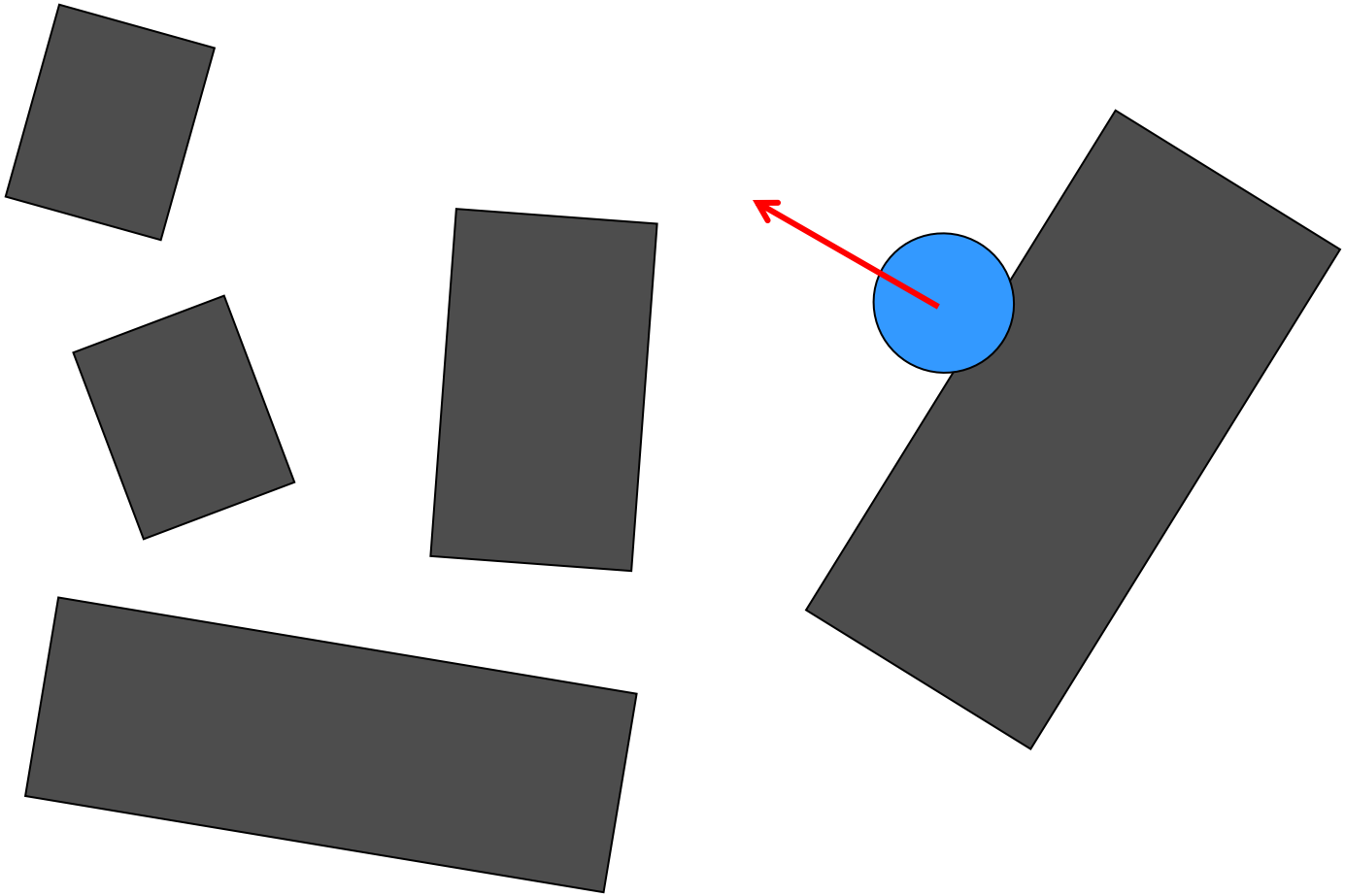
Example



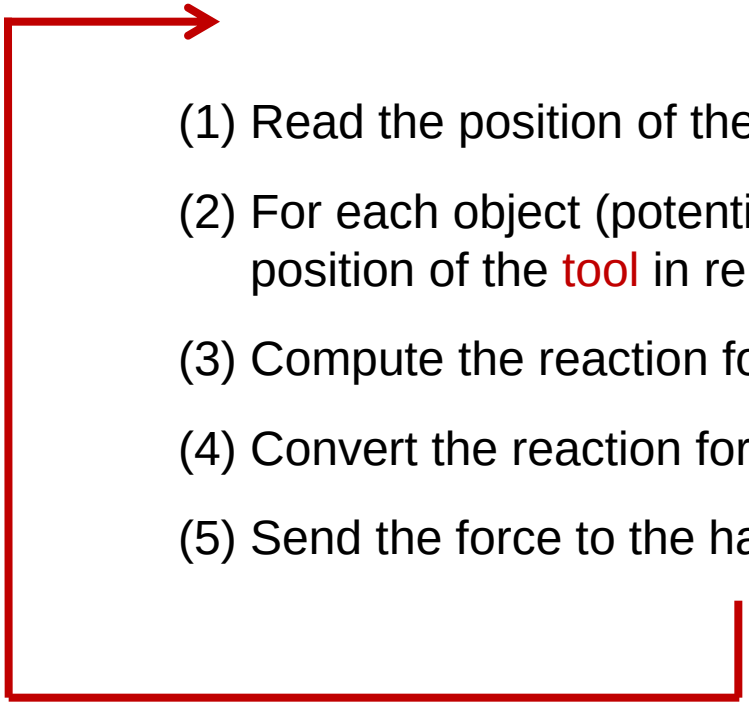
Traversing a Scene Graph



Traversing a Scene Graph



Traversing a Scene Graph



(1) Read the position of the haptic device

(2) For each object (potential field) in the environment, compute the position of the tool in relation to the local reference frame

(3) Compute the reaction force in the local frame

(4) Convert the reaction force in the world frame

(5) Send the force to the haptic device