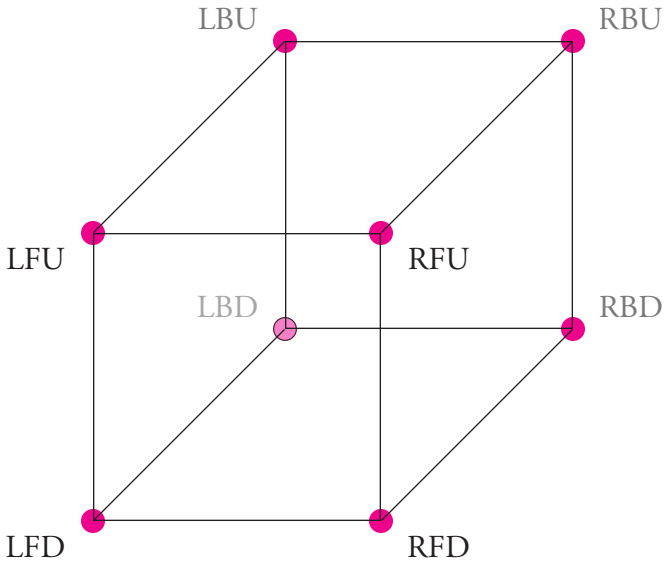
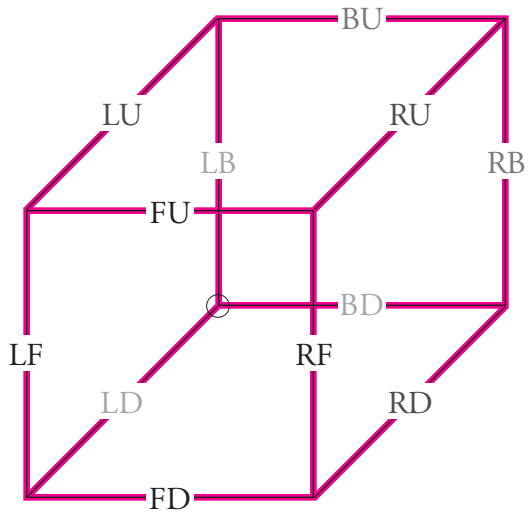


vertices
OCT_VERTEX



edges
OCT_EDGE

ONE NODE HAS:
8 VERTICES
12 EDGES
6 FACES
8 OCTANTS

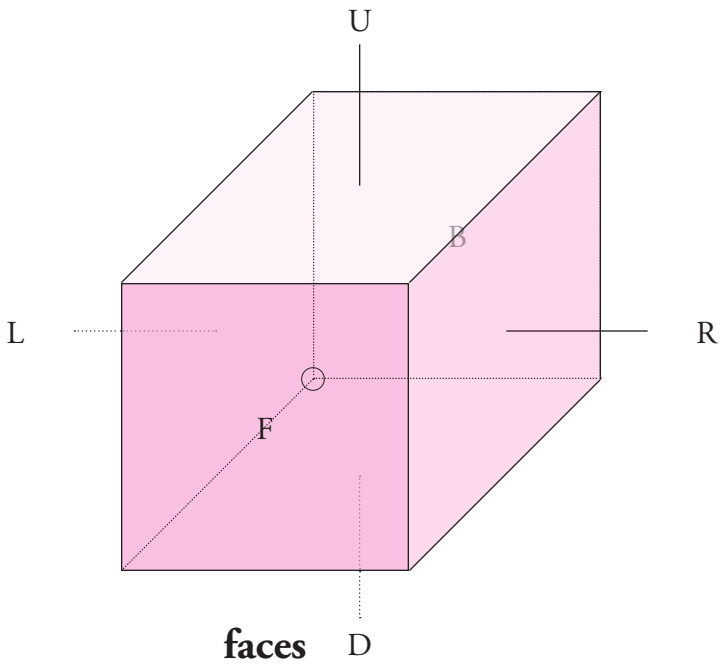
8 CHILDREN
1 PARENT
7 SIBLINGS
26 NEIGHBORS

8 VERTEX NEIGHBORS
12 EDGE NEIGHBORS
6 FACE NEIGHBORS

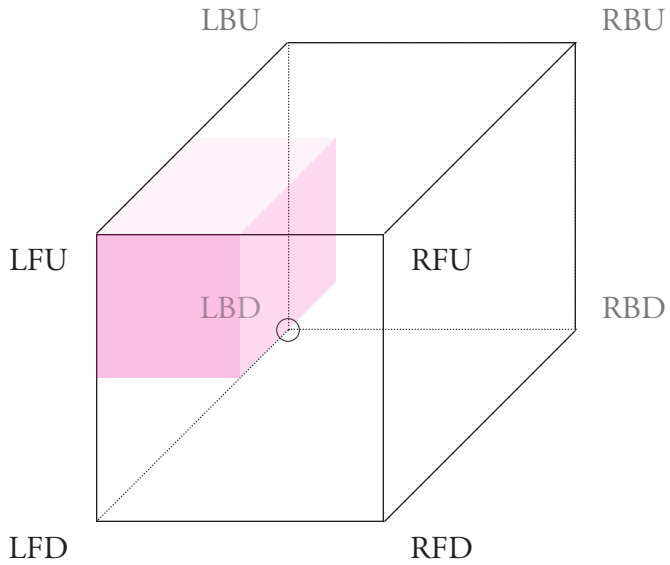
X-AXIS
- L (LEFT)
+ R (RIGHT)

Y-AXIS
- B (BACK)
+ F (FRONT)

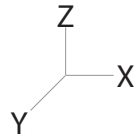
Z-AXIS
- D (DOWN)
+ U (UP)



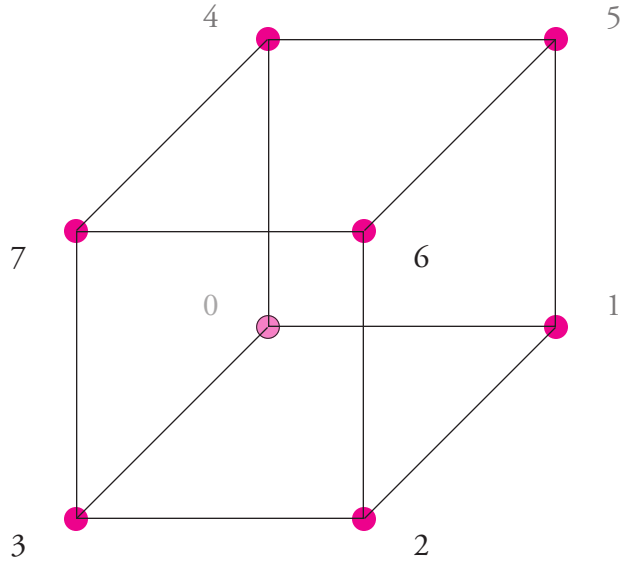
faces
OCT_FACE



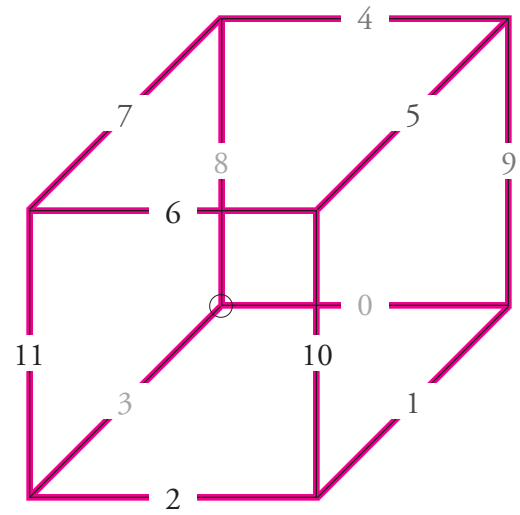
octants
OCT_OCTANT



v0.1 NAMES



vertices
OCT_VERTEX



edges
OCT_EDGE

ONE NODE HAS:
8 VERTICES
12 EDGES
6 FACES
8 OCTANTS

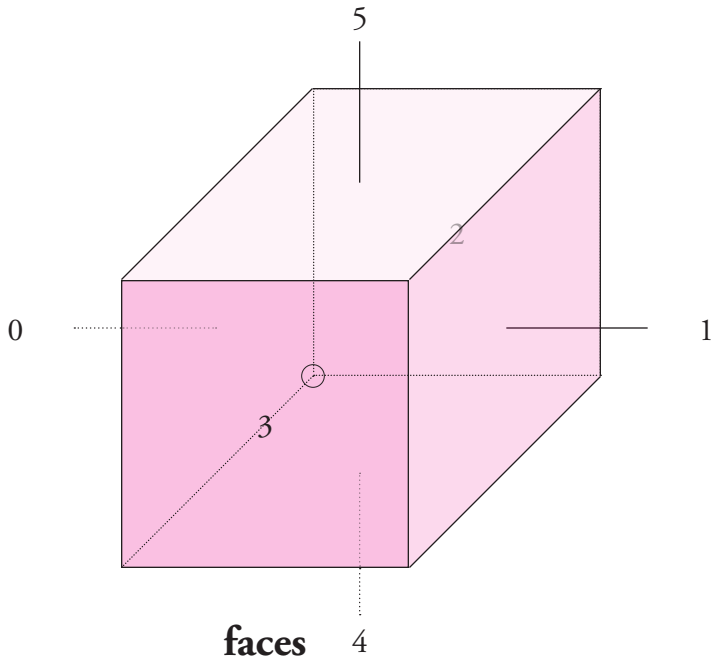
8 CHILDREN
1 PARENT
7 SIBLINGS
26 NEIGHBORS

8 VERTEX NEIGHBORS
12 EDGE NEIGHBORS
6 FACE NEIGHBORS

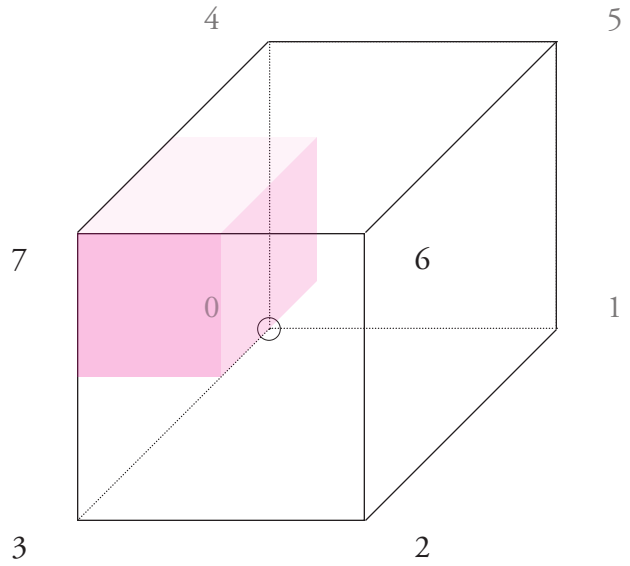
X-AXIS
- L (LEFT)
+ R (RIGHT)

Y-AXIS
- B (BACK)
+ F (FRONT)

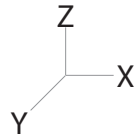
Z-AXIS
- D (DOWN)
+ U (UP)



faces
OCT_FACE



octants
OCT_OCTANT



v0.1 ORDINALS

OCT_ENUM

// enum types
OCT_VERTEX
OCT_EDGE
OCT_FACE
OCT_OCTANT
.getAll()
.get(int i)
.getOrdinal()
// direction from node center
.getR()
.getS()
.getT()
// enum name to id
.getOrdinal()
// id to enum name
.get(int i)
// direction to enum name
.get(int r, int s, int t)
// related parts of the node
.getVertices()
.getEdges()
.getFaces()
.getOctants()

OctXYZ

.toRST()

OctRST

.toXYZ()

OctNode

OctNode(codeR, codeS, codeT, level)

// code and level
.set/.getCodeR()/S()/T()/Level()
// vertex
.getVertex(OCT_VERTEX)
.getVertices(vertex list)
.getVertices() // 8 of them
.getCenter()
// octant and siblings
.getOctant() // octant type of the ndoe
.getSibling(OCT_OCTANT)
.getSiblings(octant list)
.getSiblings() // 7 of them
// parents
.getParent() // up one level
.getParent(int up)
.getAllParents()
.getAllParents(int levelsUp)
// children
.getChild(OCT_OCTANT)
.getChildren(OCT_ENUM)
.getChildren(enum list)
.getChildren() // 8 of them
.getChildren(int levelDown, OCT_ENUM)
.getChildren(int levelDown, enum list)
.getChildren(int levelDown) // 8^d of them
.getAllChildren(int levelsD, OCT_ENUM)
.getAllChildren(int levelsD, enum list)
.getAllChildren(int levelsDown)
// neighbors
.getNeighbor(OCT_ENUM)
.getNeighbors(enum list)
.getNeighbors() // 26 of them same level
.getNeighbors(same, up, down, ENUM)
.getNeighbors(same, up, down, enum list)
.getNeighbors(same, up, down)
.getAllNeighbors(s, levelsU, levelsD, ENUM)
.getAllNeighbors(s, levelsU, levelsD, enum list)
.getAllNeighbors(s, levelsU, levelsD)
// boolean
.isSiblingOf()/NeighborOf()
.isChildOf()/ParentOf()
.isWB/CodeWB/LevelWB //within bound
// draw
.drawEdges()/Faces()/Vertices()
.drawCenter()/Edge(e)/Face(f)/Vertex(v)

OctOctree

OctOctree(PApplet, dimX, dimY, dimZ)

// node list
nodeList // public HashSet
// geometry
.get/setDimensions(PVector)
.get/setOrigin(PVector)
.get/setCenter(PVector)
.get/setMin/MaxDepth(int)
// reset
.clear()/.cleatNodes
// nodes
.getNode(OctNode)
.filter(octnode list)
.nodeAdd(OctNode) // or a list
.nodeRemove(OctNode) // or a list
.nodeSubdivide(OctNode) // or a list
.nodeMerge(OctNode) // or a list
.nodeMove(OctNode) // or a list
// boolean operations
.boolSub(OctOctree)
.boolSub(octnode list)
.boolAdd(OctOctree)
.boolAdd(octnode list)
// algorithm
.algClean()
.algConstrain(int)
.algDepth()
.algGenerate(int)
.algSimplify(int)
//draw
.drawAsCenters()/AsVertices()
.drawAsEdges()/AsFaces()
.drawAsFaceSkeleton()/EdgeSkeleton()
.drawBoundingBox()
.drawAxis()

OctCuberille

OctCuberille(PApplet, OctOctree)

// setup and draw
.setup()
.draw()

OctCuberilleTri

OctCuberilleTri(PApplet, OctOctree)

// setup and draw
.setup()
.draw()

v0.1 METHODS