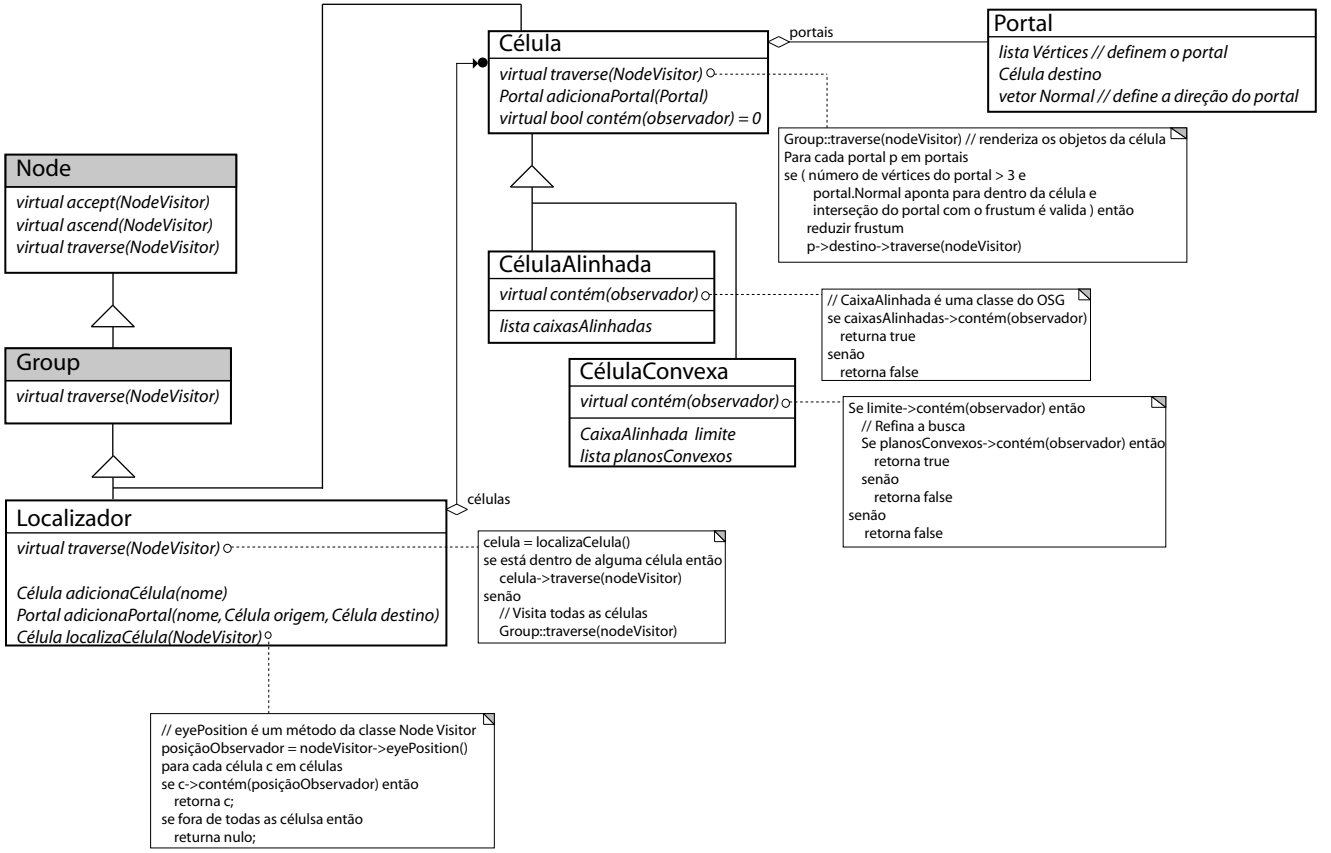
Célula
virtual traverse(NodeVisitor)
Portal adicionaPortal(Portal)
virtual bool contém(observador) = 0
portais
Portal
lista Vértices // definem o portal
Célula destino
vetor Normal // define a direção do portal
Group::traverse(nodeVisitor) // renderiza os objetos da célula
Para cada portal p em portais
se (número de vértices do portal > 3 e
portal.Normal aponta para dentro da célula e
interseção do portal com o frustum é valida) então
reduzir frustum
p->destino->traverse(nodeVisitor)
Node
virtual accept(NodeVisitor)
virtual ascend(NodeVisitor)
virtual traverse(NodeVisitor)
CélulaAlinhada
virtual contém(observador)
lista caixasAlinhadas
// CaixaAlinhada é uma classe do OSG
se caixasAlinhadas->contém(observador)
returna true
senão
retorna false
Group
virtual traverse(NodeVisitor)
CélulaConvexa
virtual contém(observador)
CaixaAlinhada limite
lista planosConvexos
Se limite->contém(observador) então
// Refina a busca
Se planosConvexos->contém(observador) então
retorna true
senão
retorna false
senão
retorna false
células
Localizador
virtual traverse(NodeVisitor)
Célula adicionaCélula(nome)
Portal adicionaPortal(nome, Célula origem, Célula destino)
Célula localizaCélula(NodeVisitor)
celula = localizaCelula()
se está dentro de alguma célula então
celula->traverse(nodeVisitor)
senão
// Visita todas as células
Group::traverse(nodeVisitor)
// eyePosition é um método da classe Node Visitor
posiçãoObservador = nodeVisitor->eyePosition()
para cada célula c em células
se c->contém(posiçãoObservador) então
retorna c;
se fora de todas as célulsa então
returna nulo;