Scene Graph structure (example)

properties (SGObjectProperty)

root/Child1 (sub-instance SGObject)

localTransform (Mat44)

Model1 (main SGObject)

Model (owner reference)

*localBBox* (Box3, *generated*)

*globalBBox* (*generated*)

*Color* (inherited from root)

*globalTransform* (*generated*)

root (main SGObject)

Geometry1 (main SGObject)

properties (SGObjectProperty)

localTransform (Mat44)

properties (SGObjectProperty)

*localBBox* (Box3, *generated*)

localBBox (Box3, *generated from the PolygonMesh*)

Child1 (owner reference)

Geometry (PolygonMesh)

root/Child1/model (sub-instance SGObject)

Child2 (owner reference)

Color (default propagated)

*Color* (inherited from root)

*globalTransform* (*generated*)

globalTransform (*generated*)

*globalBBox* (*generated*)

globalBBox (*generated*)

root/Child2/model (sub-instance SGObject)

Model2 (main SGObject)

…

Color (overrides root color)

Model (owner reference)

*Color* (inherited from model2)

globalTransform (*generated*)

globalBBox (*generated*)

root/Child2 (sub-instance SGObject)

…

SGBaseObjectWrapper

SGBaseObjectWrapper

SGBaseObjectWrapper

SGTransformed

SGTransformed

SGTransformed

SGInstanceGroup

MyAlembicGeomWrapper

SGGeometry

SGInstance

MyAlembicInstanceWrapper

SGGeneratedInstanceGroup

MyAlembicGroupWrapper

*DynamicGroupOctree*

RealTime Renderer tasks

*Note: the graph is built in KL (DFG to come later)*

Scene to RTR example

Scene Graph

Camera cull

*(RTRCullFrom VolumeTask)*

*x*LightToRTR *(adaptor)*

Query lights

*(SGBaseLight InstanceFilter)*

*x*InstanceToRTR *(adaptor)*

CameraToRTR *(adaptor)*

Visible filter *(Visible Property GroupFilter)*

Query geometry

*(SGGeometry InstanceFilter)*

Camera cull

*(RTRCullFrom VolumeTask)*

*DynamicGroupOctree*

Fills and incrementally updates a SGInstanceQuery that contains a filter selecting geometry instances.

For all input scene element references, gets the “globalBBox” property and updates an Octree, accelerating filters such as RTRCullFromVolumeTask.

BaseDynamicGroup

BaseSWDynamicGroup

DynamicGroupOctree

SWElementReferenceArray

Resulting references to SceneGraph objects

3

root/model1/geom/globalBBox

root/model2/geom/globalBBox

root/model2/geom2/globalBBox

…

0

1

2

SWDynamicGroupPropertySet

SWElementReferenceArray

Result references to SceneGraph “globalBBox” properties

Octree

SGGeometryInstanceFilter

SGInstanceQuery

2

1

0

root/model1/geom

root/model2/geom

3

…

root/model2/geom2

**Adaptor object**

SGDirectionalLight

SGInstance

“RTROGL”

RTRSWInstance

“RTROGL”

SGDirectionalLightToRTR

**To**

**From**

AdaptorRegistry

SceneGraph to RTR adaptors example

“RTROGL”

Vec3Attribute

…

PolygonMeshToRTR

“RTROGL”

Vec3AttributeToRTR

“RTROGL”

…

PolygonMesh

2. Create registered “RTROGL” adaptor

SceneGraph

RTRSWGroupToInstanceTask

SGDirectionalLight

1. Request “RTROGL” adaptor

RTRSWInstance

*(for root/model1/geom)*

3. Store adaptor

PolygonMeshToRTR

RTROGLAdaptor *(SGObjectProperty)*

For each input scene reference, get or update an adaptor for the RTR target (“RTROGL”) for the instance, its geometry, and parameters such as “positions”.

QueryGeometryInstances *(SGInstanceQuery)*

…

SWElementReferenceArray

root/model1/geom

Vec3AttributeToRTR

*(for positions attribute)*

RTRSWInstance

…

…

Adaptors are stored as SceneGraph properties so they are shared for all references to that element (SWElementReference). Additionally, this allows the adaptor data to be stored per frame if caching animation.

SGGeometry

PolygonMeshToRTR *(SGObjectProperty)*

RTRCreate SubInstanceTask

*(task executed in viewport + camera RTRContext)*

RTRDrawInstance data chain

RTROGLProgramDrawInstance

RTRCamera reference

*(from viewport context)*

RTRMaterial

model transform

OGLProgram *(Phong)*

RTROGLProgram

Default parameter values

sharedValues

modelViewMatrix

modelViewProjMatrix

OGLProgramParamValues

RTR scene objects

RTR draw instances

RTROGLProgramDrawInstance

RTRMaterial

OGLProgramParamValues

normals OGLBuffer

positions OGLBuffer

OGLVertexArrays

RTRMaterial

RTROGLProgramDrawInstance

sharedValues

shininess = *20.0*

color = *(1,0,0,1)*

OGLProgramParamValues

Vec3AttributeAdaptor *(positions)*

(

Vec3AttributeAdaptor *(normals)*

(

PolygonMeshToRTR

RTRDrawInstance

RTRSWGeometry

Tasks

RTR scene objects

RTRDrawInstance

SWElementReference *(root/car1/mesh)*

Main RTRContext

Viewport + camera RTRContext

…

RTRSWGeometryInstance

SWElementReference *(root/car1)*

sharedRTRGeometry

RTRDrawInstance

RTRInstance

*(root/car1)*

RTRSWGroupTo InstanceTask

*(task executed in main RTRContext)*

RTRAdaptor

mesh

RTRAdaptor

car1

root

SceneGraph

RTROGLAmbientLight

RTROGLShadowSpotLightInstance

RTRStandardLitSubInstanceTask

*(assign lights to lit instances)*

RTRDrawInstance

RTRMaterial

shininess = *10.0*

color = *OGLTexture*

RTRDrawInstance *(lights added)*

ambientLight\_color = *(1,1,1,1)*

shSpotLight\_color =*(1,0.5,0.5,1)*

shininess = *10.0*

color = *OGLTexture*

shSpotLight\_shadowMap =*OGLTexture*

RTRGenericMaterial

RTRMaterial

baseMaterial *(RTROGLProgram)*

*= phong* *(“phong\_\*.glsl”) defining code insertion points*

Variation “color\_tex2D|ambientLight|shadowSpotLight”

*(RTROGLProgramVariation)*

*= phong* *(“phong\_\*.glsl”)*

*+ inserted GLSL code / parameters:*

color: “texture2D” source

“ambientLight” source

“shSpotLight” source

Variation “color\_tex2D”

*(RTROGLProgramVariation)*

*= phong* *(“phong\_\*.glsl”)*

*+ inserted GLSL code / parameters:*

color: “texture2D” source

RTRGenericMaterial *(phong)*

`

RTRTaskScheduler

viewportContext

*(RTROGLContext)*

Create viewport RTRDrawInstance, assign lights *(RTRStandardLit SubInstanceTask)*

Create OpenGL light adaptor

Light prepare tasks (create shadow maps)

Clear viewport task

Scene geometry

*(Base Dynamic Group)*

Scene lights

*(Base Dynamic Group)*

Camera frustum cull

*(RTRCullFrom CameraTask)*

Create RTRInstance

*(RTRSWGroupTo InstanceTask)*

Camera frustum cull

*(RTRCullFrom CameraTask)*

GLBaseRTR

BaseRTR

SGCamera ToRTR (adaptor)

Main RTR components

*(GLStandardRTR example)*

RTROGLMaterialLibrary

RTROGLViewport

mainContext

*(RTROGLContext)*

Scene Graph

GLStandardRTR

SceneGraphTo GLStandardRTR

SWContext

*(scene context)*

Property value storage

Frames (contextual values)

All scene parameters

Static values

Stored frames

(SceneGraph.defineCurrentContext, SceneGraph.pinFrameValues)

Pre-cached

values

(SGObjectProperty

.enableAllFramesCaching)

Root/ch1/bbox

Root/ch1/material

Root/bbox

Root/ch1/color

Root/ch2/bbox

Root/ch2/color

…

…

4

3

2

1

0

Scene Graph

Scene Hub

Renderer API

Adaptors

Offline Renderer

OpenGL

Adaptors

Render Tasks

Frustum cull

Filter 2

Filter 3

Filter 1

Merge

Scene Assembly

RealTime Renderer

My Wrapper

Wrapper

My Asset