

VAAPI

Generated by Doxygen 1.8.16

1 Visual-auditory API	1
1.1 Introduction	1
1.2 Optix engine and VTK	1
1.3 TODO	2
2 Approach overview	3
2.1 Rendering optical model and geometry	3
2.2 Auditory rendering	3
3 Tutorial: FRep basics	5
3.1 FRep primitive creation and rendering	5
4 FRep tutorial	11
5 Main logic and similarities with VTK API	13
5.1 Data readers and SDF geometry creation	13
5.2 Mapper, Actor, Renderer	14
5.3 GUI and interaction	14
6 Namespace Index	17
6.1 Namespace List	17
7 Hierarchical Index	19
7.1 Class Hierarchy	19
8 Class Index	21
8.1 Class List	21
9 File Index	23
9.1 File List	23
10 Namespace Documentation	25
10.1 stkSound Namespace Reference	25
10.1.1 Function Documentation	25
10.1.1.1 ApplyADSR()	25
10.1.1.2 ApplySin()	26
10.1.1.3 CloseAL()	26
10.1.1.4 CreateWave()	26
10.1.1.5 ExitEnv()	26
10.1.1.6 GenerateNoise()	26
10.1.1.7 GeneratePlucked()	26
10.1.1.8 InitAL()	27
10.1.1.9 InitEnv()	27
10.1.1.10 MoveSounds()	27
10.1.1.11 PlayBuffer() [1/2]	27
10.1.1.12 PlayBuffer() [2/2]	27

10.1.1.13 PlaySelSound()	27
10.1.1.14 RotateAngle()	28
10.1.1.15 RotateTest()	28
10.1.1.16 WaitForSound()	28
11 Class Documentation	29
11.1 auditoryModel Class Reference	29
11.1.1 Detailed Description	30
11.1.2 Constructor & Destructor Documentation	30
11.1.2.1 auditoryModel()	30
11.1.2.2 ~auditoryModel()	30
11.1.3 Member Function Documentation	30
11.1.3.1 BindBuffer()	31
11.1.3.2 ComputeSoundRaycast()	31
11.1.3.3 ConfigureHRTF()	31
11.1.3.4 GetBuffer()	31
11.1.3.5 GetHeight()	31
11.1.3.6 GetOutput()	31
11.1.3.7 GetWidth()	31
11.1.3.8 Render()	32
11.1.3.9 Reshape()	32
11.1.4 Member Data Documentation	32
11.1.4.1 m_bufferOutput	32
11.1.4.2 m_height	32
11.1.4.3 m_width	32
11.2 basicOpticalModel Class Reference	33
11.2.1 Constructor & Destructor Documentation	34
11.2.1.1 basicOpticalModel()	35
11.2.1.2 ~basicOpticalModel()	35
11.2.2 Member Function Documentation	35
11.2.2.1 BindBuffer()	35
11.2.2.2 GetHeight()	35
11.2.2.3 GetOutput()	35
11.2.2.4 GetWidth()	35
11.2.2.5 InitOpenGL()	35
11.2.2.6 Render()	36
11.2.2.7 Reshape()	36
11.2.2.8 SetBufferSize()	36
11.2.2.9 SetDim()	36
11.2.2.10 UpdateOpticalBuffer()	36
11.2.3 Member Data Documentation	36
11.2.3.1 m_bufferOutput	36

11.2.3.2 m_height	37
11.2.3.3 m_interop	37
11.2.3.4 m_width	37
11.3 glslRayCast Class Reference	37
11.3.1 Constructor & Destructor Documentation	38
11.3.1.1 glslRayCast()	38
11.3.1.2 ~glslRayCast()	38
11.3.2 Member Function Documentation	38
11.3.2.1 ActivateTexture()	38
11.3.2.2 Display()	39
11.3.2.3 initGLSL()	39
11.3.2.4 initTexture()	39
11.3.3 Member Data Documentation	39
11.3.3.1 fsSource	39
11.3.3.2 m_glslFS	39
11.3.3.3 m_glslProgram	39
11.3.3.4 m_glslVS	39
11.3.3.5 m_hdrTexture	40
11.3.3.6 vsSource	40
11.4 MaterialDesc Struct Reference	40
11.4.1 Constructor & Destructor Documentation	40
11.4.1.1 MaterialDesc()	41
11.4.2 Member Data Documentation	41
11.4.2.1 auditory	41
11.4.2.2 dynamic	41
11.5 opticalModel Class Reference	41
11.5.1 Constructor & Destructor Documentation	44
11.5.1.1 opticalModel()	44
11.5.1.2 ~opticalModel()	44
11.5.2 Member Function Documentation	44
11.5.2.1 BindBuffer()	44
11.5.2.2 InitOpenGL()	44
11.5.2.3 Modified()	44
11.5.2.4 Render()	45
11.5.2.5 Reshape()	45
11.5.2.6 UpdateOpticalBuffer()	45
11.5.3 Member Data Documentation	45
11.5.3.1 glsl	45
11.5.3.2 m_pboOutputBuffer	45
11.6 optixBasicActor Class Reference	46
11.6.1 Detailed Description	48
11.6.2 Constructor & Destructor Documentation	48

11.6.2.1 optixBasicActor()	48
11.6.2.2 ~optixBasicActor()	48
11.6.3 Member Function Documentation	48
11.6.3.1 AddMapper()	48
11.6.3.2 GetAcceleration()	48
11.6.3.3 GetOutput()	49
11.6.3.4 PrintInfo()	49
11.6.3.5 RebuildAccel()	49
11.6.3.6 SetAccelerationProperties()	49
11.6.3.7 SetAccelerationType()	49
11.6.3.8 SetAuditoryModel()	49
11.6.3.9 SetContext()	49
11.6.3.10 SetOpticalModel()	50
11.6.3.11 SetTime()	50
11.6.3.12 Update()	50
11.6.4 Member Data Documentation	50
11.6.4.1 gg	50
11.6.4.2 m_builder	50
11.7 optixBasicRenderer Class Reference	51
11.7.1 Detailed Description	54
11.7.2 Member Enumeration Documentation	54
11.7.2.1 anonymous enum	54
11.7.2.2 RenderModes	54
11.7.3 Constructor & Destructor Documentation	54
11.7.3.1 optixBasicRenderer()	54
11.7.3.2 ~optixBasicRenderer()	55
11.7.4 Member Function Documentation	55
11.7.4.1 AddActor()	55
11.7.4.2 Display()	55
11.7.4.3 GetActor()	55
11.7.4.4 GetAudioDim()	55
11.7.4.5 GetAuditoryExceptionProgName()	55
11.7.4.6 GetAuditoryMissProgName()	55
11.7.4.7 GetAuditoryRayGenerationProgName()	56
11.7.4.8 GetExceptionProgName()	56
11.7.4.9 GetMissProgName()	56
11.7.4.10 GetMode()	56
11.7.4.11 GetNumOfActors()	56
11.7.4.12 GetOutputSoundBuffer()	56
11.7.4.13 GetRayGenerationProgName()	56
11.7.4.14 GetTime()	56
11.7.4.15 InitAcceleration()	57

11.7.4.16 InitializePrograms()	57
11.7.4.17 InitProg()	57
11.7.4.18 InitRenderer()	57
11.7.4.19 isAuditory()	57
11.7.4.20 isDynamic()	57
11.7.4.21 LaunchAuditoryContext()	57
11.7.4.22 LaunchOpticContext()	58
11.7.4.23 PlayAnimation()	58
11.7.4.24 Render()	58
11.7.4.25 Reshape()	58
11.7.4.26 setAudioBuffer()	58
11.7.4.27 SetAuditory()	58
11.7.4.28 SetAuditoryExceptionProg()	58
11.7.4.29 SetAuditoryExceptionProgName()	59
11.7.4.30 SetAuditoryMissProg()	59
11.7.4.31 SetAuditoryMissProgName()	59
11.7.4.32 SetAuditoryRayGenerationProg()	59
11.7.4.33 SetAuditoryRayGenerationProgName()	59
11.7.4.34 SetDynamic()	59
11.7.4.35 SetExceptionProg()	59
11.7.4.36 SetExceptionProgName()	60
11.7.4.37 SetMissProg()	60
11.7.4.38 SetMissProgName()	60
11.7.4.39 SetMode()	60
11.7.4.40 SetOpticalDims()	60
11.7.4.41 SetPrograms()	60
11.7.4.42 SetRayGenerationProg()	60
11.7.4.43 SetRayGenerationProgName()	61
11.7.4.44 SetTime()	61
11.7.4.45 SetUpRenderer()	61
11.7.4.46 Update()	61
11.7.5 Member Data Documentation	61
11.7.5.1 audioM	61
11.7.5.2 m_mapOfPrograms	61
11.7.5.3 opticM	61
11.8 optixMapper Class Reference	62
11.8.1 Detailed Description	64
11.8.2 Constructor & Destructor Documentation	64
11.8.2.1 optixMapper()	64
11.8.2.2 ~optixMapper()	64
11.8.3 Member Function Documentation	64
11.8.3.1 AddMaterial()	64

11.8.3.2 GetMaterial()	64
11.8.3.3 GetMaterialDesc()	65
11.8.3.4 GetOutput()	65
11.8.3.5 Modified()	65
11.8.3.6 PrintInfo()	65
11.8.3.7 SetAuditoryModel()	65
11.8.3.8 SetDescInput()	65
11.8.3.9 SetInput()	65
11.8.3.10 SetMaterialParameters()	66
11.8.3.11 SetOpticalModel()	66
11.8.3.12 SetTime()	66
11.8.3.13 Update()	66
11.9 optixReader< T > Class Template Reference	66
11.9.1 Detailed Description	67
11.9.2 Constructor & Destructor Documentation	67
11.9.2.1 optixReader()	67
11.9.2.2 ~optixReader()	67
11.9.3 Member Function Documentation	67
11.9.3.1 GetOutput()	68
11.9.3.2 Modified()	68
11.9.3.3 opxGetMacro()	68
11.9.3.4 opxSetMacro()	68
11.9.3.5 ReadFile()	68
11.9.3.6 Update()	68
11.10 optixSDFBinaryOp Class Reference	69
11.10.1 Detailed Description	71
11.10.2 Constructor & Destructor Documentation	71
11.10.2.1 optixSDFBinaryOp()	71
11.10.2.2 ~optixSDFBinaryOp()	71
11.10.3 Member Function Documentation	71
11.10.3.1 AddOperand1()	72
11.10.3.2 AddOperand2()	72
11.10.3.3 AdjustCenterAndBoundingBox()	72
11.10.3.4 CreateGeometry()	72
11.10.3.5 GetInput1()	72
11.10.3.6 GetInput2()	72
11.10.3.7 GetOutput()	72
11.10.3.8 InitCallableProg()	73
11.10.3.9 Initialize()	73
11.10.3.10 SetCallableProg()	73
11.10.3.11 SetContext()	73
11.10.3.12 SetInput1()	73

11.10.3.13 SetInput2()	73
11.10.3.14 SetMainPrograms()	74
11.10.3.15 Update()	74
11.10.4 Member Data Documentation	74
11.10.4.1 geoDesc1	74
11.10.4.2 geoDesc2	74
11.10.4.3 m_sdf1	74
11.10.4.4 m_sdf2	74
11.11 optixSDFGeometry Class Reference	75
11.11.1 Detailed Description	77
11.11.2 Constructor & Destructor Documentation	78
11.11.2.1 optixSDFGeometry()	78
11.11.2.2 ~optixSDFGeometry()	78
11.11.3 Member Function Documentation	78
11.11.3.1 CreateGeometry()	78
11.11.3.2 GetBoundingBoxProgName()	78
11.11.3.3 GetCallableProg()	78
11.11.3.4 GetCallableProgName()	78
11.11.3.5 GetIntersectionProgName()	79
11.11.3.6 GetMaterialType()	79
11.11.3.7 GetOutput()	79
11.11.3.8 GetOutputDesc()	79
11.11.3.9 GetTime()	79
11.11.3.10 Initialize()	79
11.11.3.11 InitializeInputBuffer()	79
11.11.3.12 InitProg()	80
11.11.3.13 isDynamic()	80
11.11.3.14 Modified()	80
11.11.3.15 SetBoundingBoxProg()	80
11.11.3.16 SetBoundingBoxProgName()	80
11.11.3.17 SetCallableProg()	80
11.11.3.18 SetCallableProgManually()	80
11.11.3.19 SetCallableProgName()	81
11.11.3.20 SetContext()	81
11.11.3.21 SetDynamic()	81
11.11.3.22 SetIntersectionProg()	81
11.11.3.23 SetIntersectionProgName()	81
11.11.3.24 SetMainPrograms()	81
11.11.3.25 SetMaterialType()	82
11.11.3.26 SetParameters()	82
11.11.3.27 SetTime()	82
11.11.3.28 Update()	82

11.12 optixSDFPrimitive Class Reference	82
11.12.1 Detailed Description	85
11.12.2 Constructor & Destructor Documentation	85
11.12.2.1 optixSDFPrimitive()	85
11.12.2.2 ~optixSDFPrimitive()	85
11.12.3 Member Function Documentation	85
11.12.3.1 CreateGeometry()	85
11.12.3.2 GetCenter()	86
11.12.3.3 GetOutput()	86
11.12.3.4 GetRadius()	86
11.12.3.5 InitCallableProg()	86
11.12.3.6 Initialize()	86
11.12.3.7 SetCallableProg()	86
11.12.3.8 SetCenter()	87
11.12.3.9 SetRadius()	87
11.12.4 Member Data Documentation	87
11.12.4.1 center	87
11.12.4.2 radius	87
11.13 optixSDFUnaryOp Class Reference	87
11.13.1 Detailed Description	90
11.13.2 Constructor & Destructor Documentation	90
11.13.2.1 optixSDFUnaryOp()	90
11.13.2.2 ~optixSDFUnaryOp()	90
11.13.3 Member Function Documentation	90
11.13.3.1 AddOperand()	91
11.13.3.2 AdjustCenterAndBoundingBox()	91
11.13.3.3 CreateGeometry()	91
11.13.3.4 GetInput()	91
11.13.3.5 GetOutput()	91
11.13.3.6 GetOutputSdfObject()	91
11.13.3.7 InitCallableProg()	91
11.13.3.8 Initialize()	92
11.13.3.9 SetCallableProg()	92
11.13.3.10 SetContext()	92
11.13.3.11 SetInput()	92
11.13.3.12 SetMainPrograms()	92
11.13.3.13 Update()	93
11.13.4 Member Data Documentation	93
11.13.4.1 m_sdf	93
11.14 PerRayData Struct Reference	93
11.14.1 Member Data Documentation	94
11.14.1.1 cur_prim	94

11.14.1.2 depth	94
11.14.1.3 Distances	94
11.14.1.4 f_over_pdf	94
11.14.1.5 flags	95
11.14.1.6 isDynamic	95
11.14.1.7 isSoundRay	95
11.14.1.8 numS	95
11.14.1.9 pdf	95
11.14.1.10 pos	95
11.14.1.11 primitives	95
11.14.1.12 radiance	95
11.14.1.13 result	96
11.14.1.14 rnd	96
11.14.1.15 seed	96
11.14.1.16 TimeSound	96
11.14.1.17 wi	96
11.14.1.18 wo	96
11.15 primDes Struct Reference	97
11.15.1 Constructor & Destructor Documentation	97
11.15.1.1 primDes()	97
11.15.2 Member Data Documentation	97
11.15.2.1 data	98
11.15.2.2 x	98
11.15.2.3 y	98
11.16 sdfParams Struct Reference	98
11.16.1 Constructor & Destructor Documentation	99
11.16.1.1 sdfParams() [1/3]	99
11.16.1.2 sdfParams() [2/3]	99
11.16.1.3 sdfParams() [3/3]	99
11.16.2 Member Data Documentation	99
11.16.2.1 lvShift	99
11.16.2.2 texSize	99
11.17 stkSound::soundDesc Struct Reference	100
11.17.1 Constructor & Destructor Documentation	100
11.17.1.1 soundDesc()	100
11.17.2 Member Data Documentation	100
11.17.2.1 buffer	100
11.17.2.2 source	101
11.18 visAuditoryMaterial Class Reference	101
11.18.1 Constructor & Destructor Documentation	103
11.18.1.1 visAuditoryMaterial()	103
11.18.1.2 ~visAuditoryMaterial()	103

11.18.2 Member Function Documentation	103
11.18.2.1 GetAnyHitProgName()	103
11.18.2.2 GetClosestHitProgName()	103
11.18.2.3 GetDynamicProgName()	104
11.18.2.4 GetOutput()	104
11.18.2.5 GetProgName()	104
11.18.2.6 GetType()	104
11.18.2.7 Initialize()	104
11.18.2.8 InitProg()	104
11.18.2.9 isAnyHit()	104
11.18.2.10 isAuditory()	105
11.18.2.11 isClosestHit()	105
11.18.2.12 isDynamic()	105
11.18.2.13 SetAnyHit()	105
11.18.2.14 SetAnyHitProg()	105
11.18.2.15 SetAnyHitProgName()	105
11.18.2.16 SetAuditoryTypeOff()	105
11.18.2.17 SetAuditoryTypeOn()	106
11.18.2.18 SetClosestHit()	106
11.18.2.19 SetClosestHitProg()	106
11.18.2.20 SetClosestHitProgName()	106
11.18.2.21 SetDynamicProg()	106
11.18.2.22 SetDynamicProgName()	106
11.18.2.23 SetMaterialParameters()	106
11.18.2.24 Update()	107
11.19 visParams Struct Reference	107
11.19.1 Member Data Documentation	107
11.19.1.1 brightness	107
11.19.1.2 density	108
11.19.1.3 time	108
11.19.1.4 transferOffset	108
11.19.1.5 transferScale	108
12 File Documentation	109
12.1 doc/approach.md File Reference	109
12.2 doc/freptutorial.md File Reference	109
12.3 doc/freptutorial_test.md File Reference	109
12.4 doc/vtkstyle.md File Reference	109
12.5 optixAbstractMaterial.cpp File Reference	109
12.6 optixAbstractMaterial.h File Reference	109
12.7 optixAudioOptic.cpp File Reference	110
12.7.1 Macro Definition Documentation	111

12.7.1.1 LOAD_PROC	111
12.8 optixAudioOptic.h File Reference	111
12.8.1 Macro Definition Documentation	112
12.8.1.1 optixAudioOptic_h	112
12.9 optixBasicActor.cpp File Reference	113
12.10 optixBasicActor.h File Reference	113
12.11 optixBasicRenderer.cpp File Reference	114
12.12 optixBasicRenderer.h File Reference	115
12.13 optixReader.h File Reference	115
12.14 optixSDFBasicOperations.cpp File Reference	116
12.15 optixSDFBasicOperations.h File Reference	116
12.16 optixSDFBasicPrimitives.cpp File Reference	117
12.17 optixSDFBasicPrimitives.h File Reference	118
12.18 optixSDFGeometry.cpp File Reference	119
12.18.1 Function Documentation	119
12.18.1.1 getFormat()	119
12.18.1.2 getFormat< int >()	119
12.19 optixSDFGeometry.h File Reference	120
12.20 Readme.md File Reference	120
12.21 renderTypes.h File Reference	120
12.22 shaders/renderer/per_ray_data.h File Reference	121
12.22.1 Macro Definition Documentation	122
12.22.1.1 PER_RAY_DATA_H	122
12.22.2 Typedef Documentation	122
12.22.2.1 auditoryPrim	122
12.22.3 Variable Documentation	122
12.22.3.1 MAX_PRIM_ALONG_RAY	122
12.23 stkSound.cpp File Reference	123
12.24 stkSound.h File Reference	124

Chapter 1

Visual-auditory API

1.1 Introduction

Scalar fields are used in many research areas, where computer simulations or experimental studies are involved, such as computational chemistry, medical data analysis and physical phenomena studies. Visualisation of scalar fields usually employs Volume Rendering techniques as for the computer systems the scalar fields data is often converted to data volumes stored in the texture memory. The visual analysis of scalar fields is not always straightforward, especially when a complex dynamic phenomenon is represented. In Volume Rendering the Multidimensional Transfer functions (TF) and consideration of more complex object-light interaction processes in the optical model is used to address the issues of visual analysis quality improvement and highlight features of interest.

However, the visual perception limitations not always can be overcome solely with the enhancement of the optical model. The visual system can be overloaded and perturbed due to fatigue. An additional introduction of auditory sensory stimuli to address a problem of visual analysis limitations is a well-known technique, called *sonification*.

This API targets the problems of the visual-auditory interactive study of the dynamic scalar fields using the concept of a heterogeneous object influencing various sensory stimuli. The API takes advantages of the similarities between light and sound propagation to suggest a novel procedure of a visual-auditory rendering based on the ray-casting procedure. It makes it possible to conduct a simultaneous visual analysis along with spatial positioning/measuring. The API suggests rendering and modelling of both surface and solid geometry with visual and auditory properties. However mainly it is concentrated on the Volume scene representation based on the HyperVolume (HV) model and take advantage of the Signed Distance Fields (SDF) for rendering.

The concept introduces three separate parts of the HV model for Volume Rendering that can be evaluated independently on demand to speed up the rendering process of complex dynamic volume objects and a general visual-auditory scene representation for interactive Volume Rendering.

For more details, on heterogeneous objects modelling and HyperVolume approach look at **Approach Overview** (p.3) Section:

1.2 Optix engine and VTK

The API is based on Optix Engine and takes advantage of its native rendering acceleration structures like BVH. The API syntaxis is mainly a "copy" of *Visualization Toolkit*. As well it much inherits the standard visualisation concepts reflected in VTK API, like visualisation pipeline, replacing it with notion of visual-auditory pipeline, unified on base of ray-casting procedure. Thus, API targets to provide high level access to geometric

modelling and rendering, allowing the researcher or programmer to operate familiar visualisation concepts and terminology. At the same time it is very lightweight and fast as all core procedures are CUDA kernels running on NVIDIA GPU card. The API is mostly oriented to support geometry modelling on base of `FRep` concept and targets various aspects of multisensory interaction, like collision detection, interactive manipulations and etc.

For concept of basic use, similarities and differences to VTK see **API basics** (p. 13) For FRep modelling example look **FRep tutorial** (p. 5)

1.3 TODO

- Haptic implementation
 - Collision detection
- Practical examples
 - Molecular haptics -Visual-auditory tutorials -Extending API (advanced tutorials)

Chapter 2

Approach overview

A scalar field is a mapping $f : x \rightarrow \mathbb{R}, x \in \mathbb{R}^n$, which associates any point in space with a scalar value. The conventional Volume Rendering scheme suggests that volume boundary (or scalar field domain) is defined as a bounding box and the ray/volume intersection is computed to define the ray's start and end points to solve the Volume Rendering equation with the ray-marching procedure. In the general case, the scalar field can be defined on an arbitrary domain and the object boundary can have a more complex representation and can change dynamically in time.

Let us consider the case of the molecular structure, that is a result of quantum simulation. The molecule is usually defined with electron density field and electrostatic potential fields. The entire structure can be treated as heterogeneous object, where first field defines the molecule interaction boundary $F(X)$, and the second field represents the physical property, namely the charge distribution $S_1(X)$. We take advantage of the Hyper Volume (HV) model to define the initial structure and the visual-auditory pipeline, where initial data is represented with HV model: $o(t) = (G(t), A_1(t)) : (F(X|t), S_1(X|t))$

, where t - is time parameter as in general case the researcher has to deal with dynamic molecules.

Via application of functional mapping procedures, that define optic and auditory transfer function, we receive a HV representation: $m(t) = (G(t), A_o(t), A_s(t)) : (F(X|t), S_o(X|t), S_s(X|t))$

, that represents an abstract heterogeneous object with visual and auditory properties.

2.1 Rendering optical model and geometry

Optix engine, optical materials

2.2 Auditory rendering

Optix engine, auditory materials

Chapter 3

Tutorial: FRep basics

This tutorial demonstrates basics of geometry creation according to FRep modelling approach. At the same time tutorial demonstrates the basics of API use and main concept that is much similar to Visualisation Toolkit. The core of the VTK is a visualisation pipeline of data, that starts from data source and ends up with image rendered on screen. The basic classes of VTK API are:

- Sources
- Filters
- Mappers
- Actors
- Renderers
- Interactors
- Windows

The Visual-auditory API implements basic Mappers, Actors, Renderers and Windows that are intended to support ray-casting of mainly Signed Distance Fields, although triangulated geometry is also supported and can be integrated within pipeline. The scalar field data is provided by file Readers that either output structured data or directly create Optix Buffers and TextureSamplers on GPU.

The GUI procedures are implemented on base of GLFW and ImGui libraries. A custom GUI can be easily implemented on base of those libraries and integrated to API pipeline.

3.1 FRep primitive creation and rendering

Below is simple program, creating SDF box primitive. Similar to VTK API, the Camera, Renderer, GLFW Window and Interactor to process interactive manipulations with scene are created. For syntaxis comparison with simple VTK primitive creation look [wiki](#):

```
#include "optixWindow.h"
#include "optixSDFPrimitives.h"
int main(int argc, char *argv[])
{
    int windowWidth = 512;
    int windowHeight = 512;

    PinholeCamera pinholeCamera; //creates basic camera
    contextManager m;
    m.Update(); //creates context
    optixRenderer ren;
```

```

ren.SetValid(m.GetValid());
ren.SetContext(m.GetOutput()); //m returns the context
ren.SetOpticalDims(windowWidth, windowHeight);
ren.SetCamera(&pinholeCamera);
//set not dynamic
ren.SetDynamic(false);
ren.SetAuditory(false);

//window procedure
GLFW_Window optixWindowProc;

optixWindowProc.SetDim(windowWidth, windowHeight);
optixWindowProc.SetRenderer(&ren);
optixWindowProc.SetContext(m.GetOutput());
optixWindowProc.SetCamera(&pinholeCamera);
RenderWindowInteractor iren; //TODO always check that basic still works
iren.SetWindow(&optixWindowProc);

std::cout << "DONE WITH WINDOW" << std::endl;

if (iren.SetUp()) //update of WindowProc is successful, we can start scene creation with predefined
    rendering parameters
{
    std::cout << "START MAIN LOOP" << std::endl;
    //Scene creation
    try
    {
        optixSDFBox sdf;
        sdf.SetContext(m.GetOutput());
        sdf.SetCenter1(optix::make_float3(0));
        sdf.SetDims(optix::make_float3(0.3,1.0,3.0));
        sdf.Update();
        std::cout << "BOX created" << std::endl;
        SDFMaterial mSdf;
        mSdf.SetContext(m.GetOutput());
        mSdf.Update();
        std::cout << "MATERIAL created" << std::endl;
        optixMapper map21;
        map21.SetContext(m.GetOutput());
        map21.SetInput(sdf.GetOutput());
        map21.AddMaterial(mSdf.GetOutput(), mSdf.GetType());
        map21.Update();
        optixSdfActor acSdf1;
        acSdf1.SetContext(m.GetOutput());
        acSdf1.AddMapper(&map21);
        acSdf1.Update();
        ren.AddActor(&acSdf1);
    }
    catch (optix::Exception& e)
    {
        std::cerr << e.getErrorString() << std::endl;
    }
    std::cout << "DONE WITH SCENE" << std::endl;
    //Main loop
    iren.Start();
    std::cout << "MAIN LOOP STARTED" << std::endl;
}

return 0;
}

```

The output of the program is:

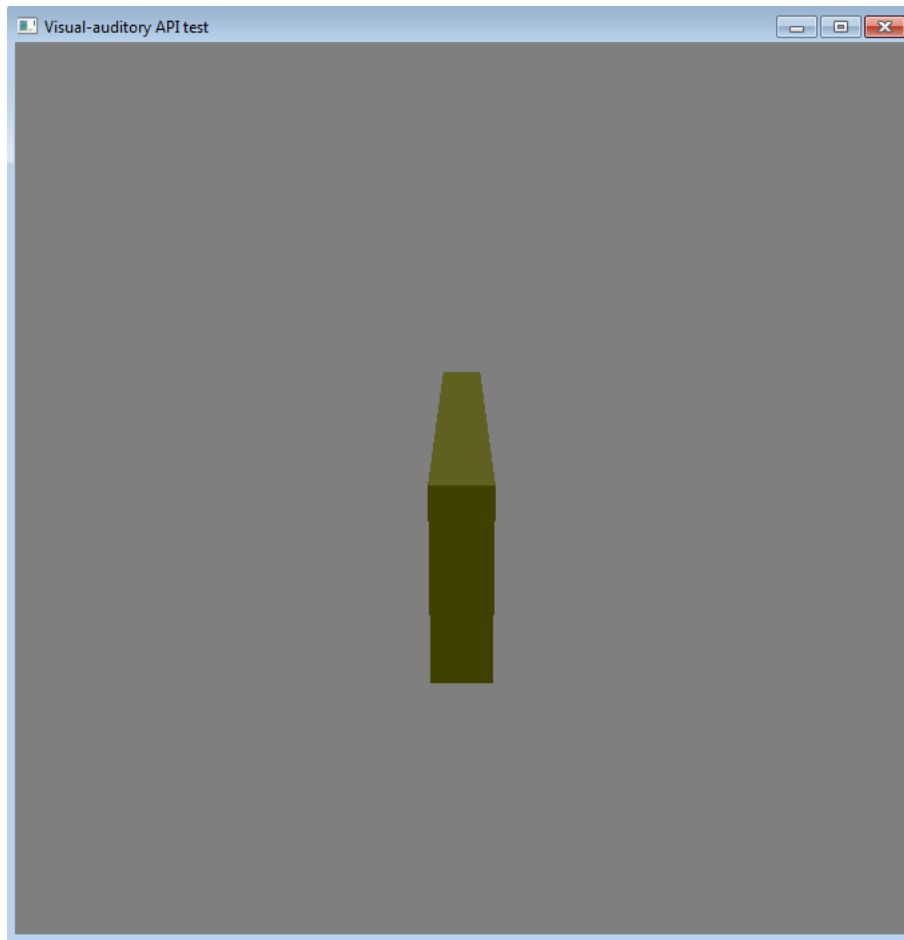


Figure 3.1 Rendered SDF box

A more complex example is below:

```
#include "optixWindow.h"
#include "optixSDFPrimitives.h"
#include "optixSDFOperations.h"
int main(int argc, char *argv[])
{
    int windowHeight = 512;
    int windowWidth = 512;

    PinholeCamera pinholeCamera; //creates basic camera
    contextManager m;
    m.Update(); //creates context
    optixRenderer ren;
    ren.SetValid(m.GetValid());
    ren.SetContext(m.GetOutput());
    ren.SetOpticalDims(windowWidth, windowHeight);
    ren.SetCamera(&pinholeCamera);
    //set not dynamic
    ren.SetDynamic(false);
    ren.SetAuditory(false);

    //window procedure
    GLFW_Window optixWindowProc;

    optixWindowProc.SetDim(windowWidth, windowHeight);
    optixWindowProc.SetRenderer(&ren);
    optixWindowProc.SetContext(m.GetOutput()); //returns context
    optixWindowProc.SetCamera(&pinholeCamera);
    RenderWindowInteractor iren; //TODO always check that basic still works
    iren.SetWindow(&optixWindowProc);

    std::cout << "DONE WITH WINDOW" << std::endl;

    if (iren.SetUp()) //update of WindowProc
    {
        std::cout << "START MAIN LOOP" << std::endl;
        //Scene creation
    }
}
```

```

try
{
    optixSDFBox sdf;
    sdf.SetContext(m.GetOutput());
    sdf.SetCenter1(optix::make_float3(1.0));
    sdf.SetDims(optix::make_float3(0.3));
    sdf.Update();
    const int nums = 10;
    SDFRoundingOp round[nums];
    round[0].SetContext(m.GetOutput());
    round[0].AddOperand(&sdf);
    round[0].SetKcoeff(0.01);
    round[0].Update();
    for (int i = 1; i < nums; i++)
    {
        round[i].SetContext(m.GetOutput());
        round[i].AddOperand(round[i - 1].GetOutputSdfObject()); // &sdf;
        round[i].SetKcoeff(0.01*i);
        round[i].Update();
    }
    optixSDFTorus sdfT;
    sdfT.SetContext(m.GetOutput());
    sdfT.SetCenter1(optix::make_float3(0.0));
    sdfT.SetRadius1(optix::make_float2(0.4, 0.1));
    sdfT.Update();
    SDFElongateOp el;
    el.SetContext(m.GetOutput());
    el.AddOperand(&sdfT);
    el.SetHKcoeff(optix::make_float3(0.0, 1.0, 2.1));
    el.Update();
    SDFBlendUnionOp opBlend;
    opBlend.SetContext(m.GetOutput());
    opBlend.AddOperand1(&sdf);
    opBlend.AddOperand2(el.GetOutputSdfObject());
    opBlend.SetKcoeff(0.3);
    opBlend.Update();

    SDFMaterial mSdf;
    mSdf.SetContext(m.GetOutput());
    mSdf.Update();
    optixMapper map21;
    map21.SetContext(m.GetOutput());
    map21.SetInput(sdf.GetOutput());
    map21.AddMaterial(mSdf.GetOutput(), mSdf.GetType());
    map21.Update();
    optixSdfActor acSdf1;
    acSdf1.SetContext(m.GetOutput());
    acSdf1.AddMapper(&map21);
    acSdf1.Update();
    ren.AddActor(&acSdf1);
}
catch (optix::Exception& e)
{
    std::cerr << e.getErrorString() << std::endl;
}
std::cout << "DONE WITH SCENE" << std::endl;
//Main loop
iren.Start();
std::cout << "MAIN LOOP STARTED" << std::endl;
}

return 0;
}

```

The output of the program is :

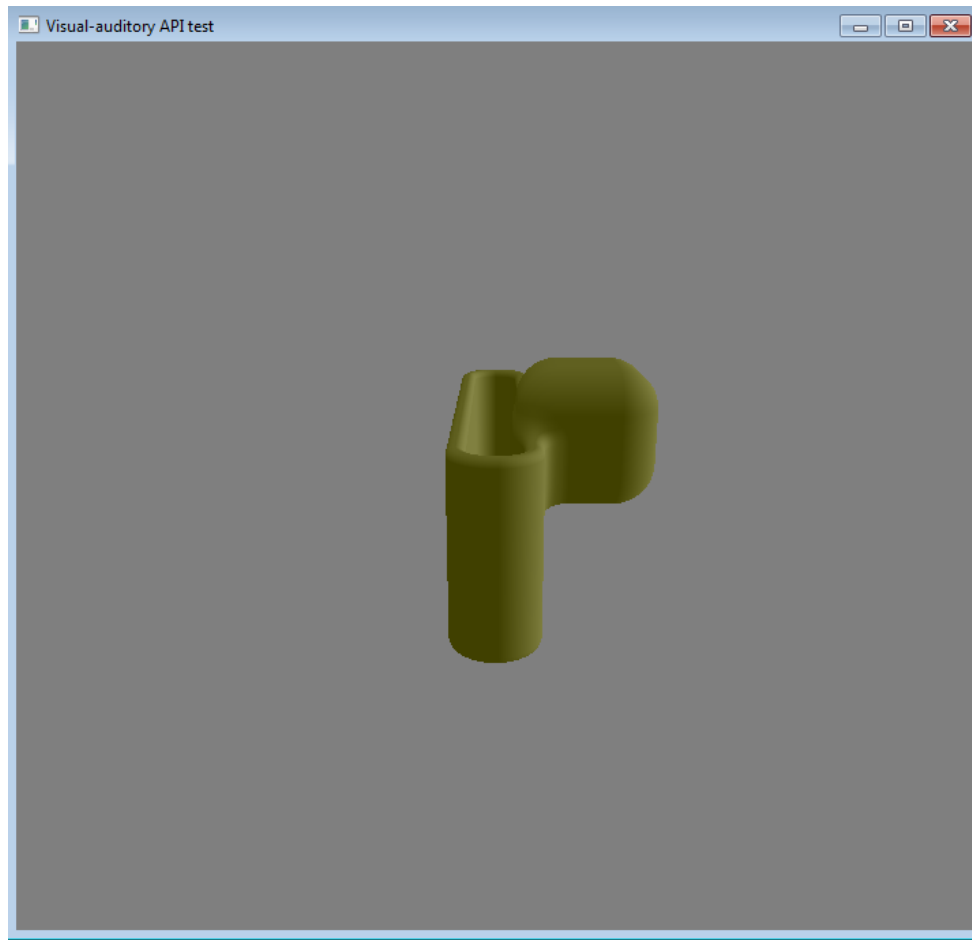


Figure 3.2 Rendered FRep modelled geometry

Chapter 4

FRep tutorial

Below is a basic example of creation of the FRep tree

Chapter 5

Main logic and similarities with VTK API

The basics of API use and main concept that is much similar to `Visualisation Toolkit`. The core of the VTK is a visualisation pipeline of data, that starts from data source and ends up with image rendered on screen. The basic classes of VTK API are:

- Sources
- Filters
- Mappers
- Actors
- Renderers
- Interactors
- Windows

The Visual-auditory API implements basic Mappers, Actors, Renderers and Windows that are intended to support ray-casting of mainly Signed Distance Fields, although triangulated geometry is also supported and can be integrated within pipeline. Similarly to VTK the pipeline is constructed through "input-output" concept. The scalar field data is provided by file Readers that either output structured data or directly create Optix Buffers and Texture Samplers on GPU. The inputs and outputs are usually Optix objects, like `optix::Geometry`, `optix::Material`, `optix::GeometryInstance` and etc.. As all this objects are created through `optix::Context`, the reference to it is given to each class. For context creation a `ContextManager` class is responsible.

5.1 Data readers and SDF geometry creation

In the example below, the Reader that reads molecular data described in `XYZ` file format and transfers it to "ball" like representation, when atoms represented as spheres (balls). This is current version of `spacefill` molecule representation. For info on molecule representation types.

```
contextManager m;
m.Update(); //creates context
optixXYZReader read; //reads data
read.Setfile("molecule.xyz");
read.Update();
optixSDFMolBalls mol; //
mol.SetContext(m.GetOutput());
mol.SetCenter(read.GetOutput1()); //centers, data stored in std::vector
mol.SetTypes(read.GetOutput2()); //set types of atoms, data stored in std::vector
mol.SetMaterialType(0);
mol.Update();
```

The other example is reading dynamic scalar data, that represents . Gamess files. The reader performs signed distance transform of data on base of Image Processing Toolkit. It generates optix Buffers and Textures Samplers. The optixSDFTexture gets Texture Samplers describing initial and final states of molecule dynamic electron density field and forms dynamic SDF geometry object.

```
contextManager m;
m.Update();//creates context
optixSDFTextureReader<float> readSDFTex1;
readSDFTex1.SetContext(m.GetOutput());
readSDFTex1.SetSize(139, 150, 160); //data dimensions
readSDFTex1.SetThreshold(0.1); //isolevel of scalar field
readSDFTex1.Setfile("ed1.txt");
readSDFTex1.Update();
optixSDFTextureReader<float> readSDFTex2;
readSDFTex2.SetContext(m.GetOutput());
readSDFTex2.SetSize(138, 150, 160);
readSDFTex2.SetThreshold(0.1);
readSDFTex2.Setfile("ed2.txt");
readSDFTex2.Update();
optixSDFTexture tex; //dynamic SDF object creation
tex.SetContext(m.GetOutput());
tex.SetTexture(readSDFTex1.GetTexture(), readSDFTex1.GetParam());
tex.SetTexture(readSDFTex2.GetTexture(), readSDFTex1.GetParam());
tex.Update();
```

5.2 Mapper, Actor, Renderer

Mapper, Actor and Renderer classes are much similar to VTK in concept and syntacsis. The Mapper applies a material to the objects (in current implementation with optical or auditory properties). The Mapper is can activate optical or auditory material, depending on type of rendering procedure: rendering optical or auditory properties. For this, in current implementation the Mapper receives optix::Material from Material class (material.GetOutput() procedure) and description of it's properties in , material.GetType() procedure.

The Renderer is responsible for rendering visual and auditory properties. Whether optical or auditory properties should be rendered, the Renderer operates its actors to activate the necessary visual or auditory model for all created pipelines of data.

Below the code demonstrating the above concepts and syntacsis of Mapper, Actor and Renderer classes basic usage.

```
SDFMaterial mSdf;
mSdf.SetContext(m.GetOutput());
mSdf.Update();
optixMapper map21;
map21.SetContext(m.GetOutput());
map21.SetInput(sdf.GetOutput());
map21.AddMaterial(mSdf.GetOutput(), mSdf.GetType());
map21.Update();
optixSdfActor acSdf1;
acSdf1.SetContext(m.GetOutput());
acSdf1.AddMapper(&map21);
acSdf1.Update();
ren.AddActor(&acSdf1);
....
ren.LaunchAuditoryContext(); //compute auditory ray-tracing
...
ren.LaunchOpticContext();// compute optical ray-tracing
```

5.3 GUI and interaction

The GUI procedures are implemented on base of GLFW and ImGUI libraries. A custom GUI can be easily implemented on base of those libraries and integrated to API pipeline.

The API provides a basic Renderer, GLFW window and Interactor:

```
contextManager m;
m.Update();//creates context
PinholeCamera pinholeCamera; //basic camera
optixRenderer ren;
ren.SetValid(m.GetValid());
```

```
ren.SetContext(m.GetOutput());
ren.SetOpticalDims(windowWidth, windowHeight);
ren.SetCamera(&pinholeCamera);
//set not dynamic
ren.SetDynamic(false);
ren.SetAuditory(false);
//window procedure
GLFW_Window optixWindowProc;
optixWindowProc.SetDim(windowWidth, windowHeight);
optixWindowProc.SetRenderer(&ren);
optixWindowProc.SetContext(m.GetOutput()); //returns context
optixWindowProc.SetCamera(&pinholeCamera);
RenderWindowInteractor iren; //TODO always check that basic still works
iren.SetWindow(&optixWindowProc);
```


Chapter 6

Namespace Index

6.1 Namespace List

Here is a list of all namespaces with brief descriptions:

stkSound	25
-----------------	-------	----

Chapter 7

Hierarchical Index

7.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

auditoryModel	29
basicOpticalModel	33
opticalModel	41
glslRayCast	37
MaterialDesc	40
optixObject	
optixBasicActor	46
optixBasicRenderer	51
optixMapper	62
optixSDFGeometry	75
optixSDFBinaryOp	69
optixSDFPrimitive	82
optixSDFUnaryOp	87
visAuditoryMaterial	101
optixReader< T >	66
PerRayData	93
primDes	97
sdfParams	98
stkSound::soundDesc	100
visParams	107

Chapter 8

Class Index

8.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

auditoryModel	
Architecture much inspired by VTK Renderer(http://vtk.org) implementation The classes are mainly responsible for openGL/openAL interop	29
basicOpticalModel	33
glslRayCast	37
MaterialDesc	40
opticalModel	41
optixBasicActor	
Basic abstract actor class	46
optixBasicRenderer	
Architecture much inspired by VTK Renderer(http://vtk.org) implementation Abstract visual-auditory renderer	51
optixMapper	
A basic mapper that suggest application of one visual and one auditory material to one geometry instance	62
optixReader< T >	
The class creates a basic triangular plane for Optix use. Architecture much inspired by VTK (http://vtk.org) implementation	66
optixSDFBinaryOp	
Binary operation	69
optixSDFGeometry	
Abstract class for SDF Geometry. A basic class. All SDF geometry inherits it's functions	75
optixSDFPrimitive	
A basic class for all sdf primitives	82
optixSDFUnaryOp	
Unary operation	87
PerRayData	93
primDes	97
sdfParams	98
stkSound::soundDesc	100
visAuditoryMaterial	101
visParams	107

Chapter 9

File Index

9.1 File List

Here is a list of all files with brief descriptions:

optixAbstractMaterial.cpp	109
optixAbstractMaterial.h	109
optixAudioOptic.cpp	110
optixAudioOptic.h	111
optixBasicActor.cpp	113
optixBasicActor.h	113
optixBasicRenderer.cpp	114
optixBasicRenderer.h	115
optixReader.h	115
optixSDFBasicOperations.cpp	116
optixSDFBasicOperations.h	116
optixSDFBasicPrimitives.cpp	117
optixSDFBasicPrimitives.h	118
optixSDFGeometry.cpp	119
optixSDFGeometry.h	120
renderTypes.h	120
stkSound.cpp	123
stkSound.h	124
shaders/renderer/per_ray_data.h	121

Chapter 10

Namespace Documentation

10.1 stkSound Namespace Reference

Classes

- struct **soundDesc**

Functions

- int **InitAL** (char ***argv, int *argc)
- void **CloseAL** (void)
- void **ApplySin** (ALfloat *data, ALdouble g, ALuint srates, ALuint freq)
- void **CreateWave** (StkFloat *ptr, int srates2, int duration, ALuint &buffer)
- void **InitEnv** ()
- void **ExitEnv** ()
- StkFrames **GenerateNoise** (double time, double srates2, float seed)
- void **WaitforSound** (float scale)
- void **PlaySelSound** (ALuint source)
- void **MoveSounds** (std::vector< **stkSound::soundDesc** > &descs, std::vector< float > &xpos, int scale)
- void **RotateTest** (int scale)
- **soundDesc** **PlayBuffer** (StkFloat *ptr, int time, int size, int j, optix::float2 xpos)
- void **RotateAngle** (double angle, ALuint &source)
- void **ApplyADSR** (stk::StkFrames &frames, int releaseCount)
- StkFrames **GeneratePlucked** (int time, double srates2, double freq, double amplitude)
- **soundDesc** **PlayBuffer** (stk::StkFloat *ptr, int time, int size, int j, optix::float2 xpos)

10.1.1 Function Documentation

10.1.1.1 ApplyADSR()

```
void stkSound::ApplyADSR (
    stk::StkFrames & frames,
    int releaseCount )
```

10.1.1.2 ApplySin()

```
void stkSound::ApplySin (
    ALfloat * data,
    ALdouble g,
    ALuint srate,
    ALuint freq )
```

10.1.1.3 CloseAL()

```
void stkSound::CloseAL (
    void )
```

10.1.1.4 CreateWave()

```
void stkSound::CreateWave (
    StkFloat * ptr,
    int srate2,
    int duration,
    ALuint & buffer )
```

10.1.1.5 ExitEnv()

```
void stkSound::ExitEnv ( )
```

10.1.1.6 GenerateNoise()

```
stk::StkFrames stkSound::GenerateNoise (
    double time,
    double srate2,
    float seed )
```

10.1.1.7 GeneratePlucked()

```
stk::StkFrames stkSound::GeneratePlucked (
    int time,
    double srate2,
    double freq,
    double amplitude )
```


10.1.1.8 InitAL()

```
int stkSound::InitAL (
    char *** argv,
    int * argc )
```

10.1.1.9 InitEnv()

```
void stkSound::InitEnv ( )
```

10.1.1.10 MoveSounds()

```
void stkSound::MoveSounds (
    std::vector< stkSound::soundDesc > & descSs,
    std::vector< float > & xpos,
    int scale )
```

10.1.1.11 PlayBuffer() [1/2]

```
soundDesc stkSound::PlayBuffer (
    stk::StkFloat * ptr,
    int time,
    int size,
    int j,
    optix::float2 xpos )
```

10.1.1.12 PlayBuffer() [2/2]

```
soundDesc stkSound::PlayBuffer (
    StkFloat * ptr,
    int time,
    int size,
    int j,
    optix::float2 xpos )
```

10.1.1.13 PlaySelSound()

```
void stkSound::PlaySelSound (
    ALuint source )
```

10.1.1.14 RotateAngle()

```
void stkSound::RotateAngle (
    double angle,
    ALuint & source )
```

10.1.1.15 RotateTest()

```
void stkSound::RotateTest (
    int scale )
```

10.1.1.16 WaitForSound()

```
void stkSound::WaitforSound (
    float scale )
```

Chapter 11

Class Documentation

11.1 auditoryModel Class Reference

Architecture much inspired by VTK Renderer(<http://vtk.org>) implementation The classes are mainly responsible for openGL/openAL interop.

```
#include <optixAudioOptic.h>
```

Collaboration diagram for auditoryModel:

auditoryModel
m_bufferOutput # m_width # m_height
+ auditoryModel() + ~auditoryModel() + Render() + BindBuffer() + GetBuffer() + Reshape() + GetWidth() + GetHeight() + GetOutput() # ComputeSoundRaycast() # ConfigureHRTF()

Public Member Functions

- **auditoryModel** ()
- **~auditoryModel** ()

- void **Render** ()
- void **BindBuffer** (optix::Context context)
- void **GetBuffer** ()
- void **Reshape** (int width, int height)
- int **GetWidth** ()
- int **GetHeight** ()
- optix::Buffer **GetOutput** ()

Protected Member Functions

- void **ComputeSoundRaycast** (float distNorm, float distComp)
- void **ConfigureHRTF** ()

Protected Attributes

- optix::Buffer **m_bufferOutput**
- int **m_width**
- int **m_height**

11.1.1 Detailed Description

Architecture much inspired by VTK Renderer(<http://vtk.org>) implementation The classes are mainly responsible for openGL/openAL interop.

11.1.2 Constructor & Destructor Documentation

11.1.2.1 auditoryModel()

```
auditoryModel::auditoryModel ( ) [inline]
```

11.1.2.2 ~auditoryModel()

```
auditoryModel::~~auditoryModel ( ) [inline]
```

11.1.3 Member Function Documentation

11.1.3.1 BindBuffer()

```
void auditoryModel::BindBuffer (
    optix::Context context )
```

11.1.3.2 ComputeSoundRaycast()

```
void auditoryModel::ComputeSoundRaycast (
    float distNorm,
    float distComp ) [protected]
```

11.1.3.3 ConfigureHRTF()

```
void auditoryModel::ConfigureHRTF ( ) [protected]
```

11.1.3.4 GetBuffer()

```
void auditoryModel::GetBuffer ( )
```

11.1.3.5 GetHeight()

```
int auditoryModel::GetHeight ( ) [inline]
```

11.1.3.6 GetOutput()

```
optix::Buffer auditoryModel::GetOutput ( ) [inline]
```

11.1.3.7 GetWidth()

```
int auditoryModel::GetWidth ( ) [inline]
```

11.1.3.8 Render()

```
void auditoryModel::Render ( ) [inline]
```

11.1.3.9 Reshape()

```
void auditoryModel::Reshape (
    int width,
    int height )
```

11.1.4 Member Data Documentation

11.1.4.1 m_bufferOutput

```
optix::Buffer auditoryModel::m_bufferOutput [protected]
```

11.1.4.2 m_height

```
int auditoryModel::m_height [protected]
```

11.1.4.3 m_width

```
int auditoryModel::m_width [protected]
```

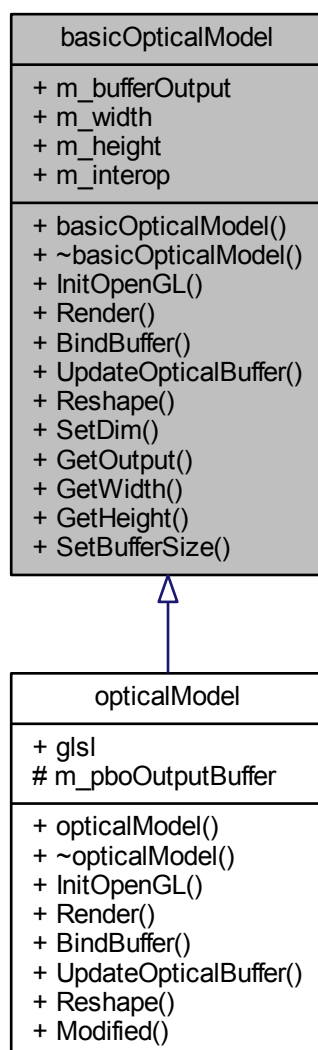
The documentation for this class was generated from the following files:

- **optixAudioOptic.h**
- **optixAudioOptic.cpp**

11.2 basicOpticalModel Class Reference

```
#include <optixAudioOptic.h>
```

Inheritance diagram for basicOpticalModel:



Collaboration diagram for basicOpticalModel:

basicOpticalModel
+ m_bufferOutput + m_width + m_height + m_interop
+ basicOpticalModel() + ~basicOpticalModel() + InitOpenGL() + Render() + BindBuffer() + UpdateOpticalBuffer() + Reshape() + SetDim() + GetOutput() + GetWidth() + GetHeight() + SetBufferSize()

Public Member Functions

- **basicOpticalModel** ()
- **~basicOpticalModel** ()
- virtual void **InitOpenGL** ()
- virtual void **Render** ()
- virtual void **BindBuffer** (optix::Context context)
- virtual void **UpdateOpticalBuffer** ()
- virtual void **Reshape** (int width, int height)
- void **SetDim** (int width, int height)
- optix::Buffer **GetOutput** ()
- int **GetWidth** ()
- int **GetHeight** ()
- void **SetBufferSize** ()

Public Attributes

- optix::Buffer **m_bufferOutput**
- int **m_width**
- int **m_height**
- bool **m_interop**

11.2.1 Constructor & Destructor Documentation

11.2.1.1 basicOpticalModel()

```
basicOpticalModel::basicOpticalModel ( ) [inline]
```

11.2.1.2 ~basicOpticalModel()

```
basicOpticalModel::~~basicOpticalModel ( ) [inline]
```

11.2.2 Member Function Documentation

11.2.2.1 BindBuffer()

```
virtual void basicOpticalModel::BindBuffer (
    optix::Context context ) [inline], [virtual]
```

Reimplemented in **opticalModel** (p. 44).

11.2.2.2 GetHeight()

```
int basicOpticalModel::GetHeight ( ) [inline]
```

11.2.2.3 GetOutput()

```
optix::Buffer basicOpticalModel::GetOutput ( ) [inline]
```

11.2.2.4 GetWidth()

```
int basicOpticalModel::GetWidth ( ) [inline]
```

11.2.2.5 InitOpenGL()

```
virtual void basicOpticalModel::InitOpenGL ( ) [inline], [virtual]
```

Reimplemented in **opticalModel** (p. 44).

11.2.2.6 Render()

```
virtual void basicOpticalModel::Render ( ) [inline], [virtual]
```

Reimplemented in **opticalModel** (p. 44).

11.2.2.7 Reshape()

```
virtual void basicOpticalModel::Reshape (
    int width,
    int height ) [inline], [virtual]
```

Reimplemented in **opticalModel** (p. 45).

11.2.2.8 SetBufferSize()

```
void basicOpticalModel::SetBufferSize ( ) [inline]
```

11.2.2.9 SetDim()

```
void basicOpticalModel::SetDim (
    int width,
    int height ) [inline]
```

11.2.2.10 UpdateOpticalBuffer()

```
virtual void basicOpticalModel::UpdateOpticalBuffer ( ) [inline], [virtual]
```

Reimplemented in **opticalModel** (p. 45).

11.2.3 Member Data Documentation

11.2.3.1 m_bufferOutput

```
optix::Buffer basicOpticalModel::m_bufferOutput
```

11.2.3.2 m_height

```
int basicOpticalModel::m_height
```

11.2.3.3 m_interop

```
bool basicOpticalModel::m_interop
```

11.2.3.4 m_width

```
int basicOpticalModel::m_width
```

The documentation for this class was generated from the following file:

- **optixAudioOptic.h**

11.3 glslRayCast Class Reference

```
#include <optixAudioOptic.h>
```

Collaboration diagram for glslRayCast:

glslRayCast
+ vsSource + fsSource # m_glslVS # m_glslFS # m_glslProgram # m_hdrTexture
+ glslRayCast() + ~glslRayCast() + initTexture() + initGLSL() + Display() + ActivateTexture()

Public Member Functions

- **glslRayCast** ()
- **~glslRayCast** ()
- void **initTexture** ()
- void **initGLSL** ()
- void **Display** ()
- void **ActivateTexture** ()

Public Attributes

- std::string **vsSource**
- std::string **fsSource**

Protected Attributes

- GLuint **m_glslVS**
- GLuint **m_glslFS**
- GLuint **m_glslProgram**
- GLuint **m_hdrTexture**

11.3.1 Constructor & Destructor Documentation

11.3.1.1 glslRayCast()

```
glslRayCast::glslRayCast ( )
```

11.3.1.2 ~glslRayCast()

```
glslRayCast::~~glslRayCast ( ) [inline]
```

11.3.2 Member Function Documentation

11.3.2.1 ActivateTexture()

```
void glslRayCast::ActivateTexture ( ) [inline]
```

11.3.2.2 Display()

```
void glslRayCast::Display ( )
```

11.3.2.3 initGLSL()

```
void glslRayCast::initGLSL ( )
```

11.3.2.4 initTexture()

```
void glslRayCast::initTexture ( )
```

11.3.3 Member Data Documentation

11.3.3.1 fsSource

```
std::string glslRayCast::fsSource
```

11.3.3.2 m_glslFS

```
GLuint glslRayCast::m_glslFS [protected]
```

11.3.3.3 m_glslProgram

```
GLuint glslRayCast::m_glslProgram [protected]
```

11.3.3.4 m_glslVS

```
GLuint glslRayCast::m_glslVS [protected]
```

11.3.3.5 m_hdrTexture

```
GLuint glslRayCast::m_hdrTexture [protected]
```

11.3.3.6 vsSource

```
std::string glslRayCast::vsSource
```

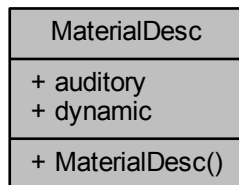
The documentation for this class was generated from the following files:

- **optixAudioOptic.h**
- **optixAudioOptic.cpp**

11.4 MaterialDesc Struct Reference

```
#include <optixAbstractMaterial.h>
```

Collaboration diagram for MaterialDesc:



Public Member Functions

- **MaterialDesc** (bool au, bool dy)

Public Attributes

- bool **auditory**
- bool **dynamic**

11.4.1 Constructor & Destructor Documentation

11.4.1.1 MaterialDesc()

```
MaterialDesc::MaterialDesc (
    bool au,
    bool dy ) [inline]
```

11.4.2 Member Data Documentation

11.4.2.1 auditory

```
bool MaterialDesc::auditory
```

11.4.2.2 dynamic

```
bool MaterialDesc::dynamic
```

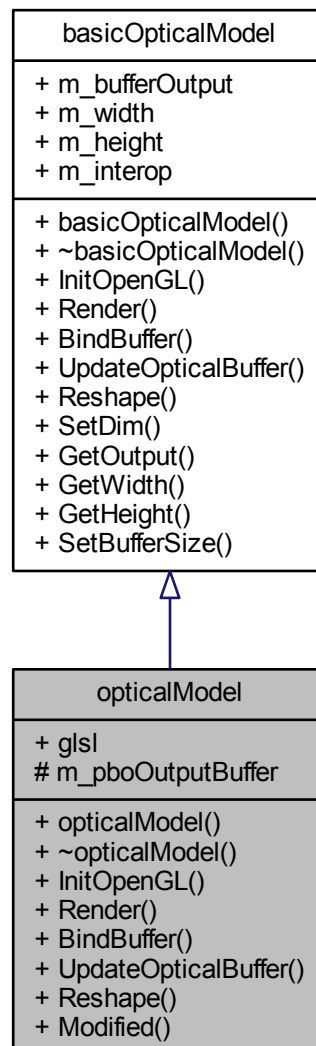
The documentation for this struct was generated from the following file:

- `optixAbstractMaterial.h`

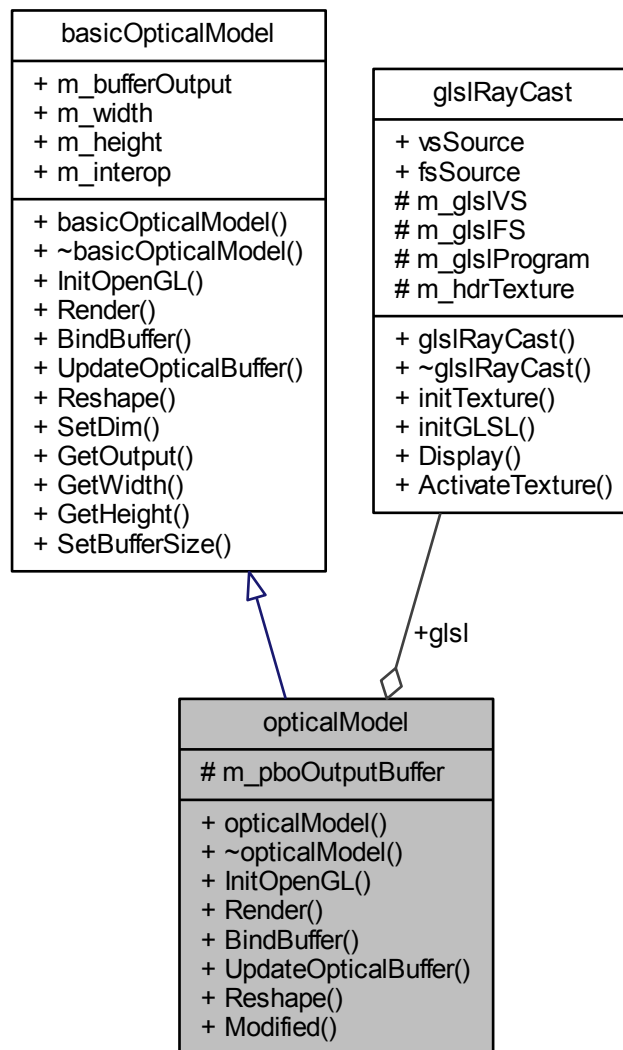
11.5 opticalModel Class Reference

```
#include <optixAudioOptic.h>
```

Inheritance diagram for opticalModel:



Collaboration diagram for opticalModel:



Public Member Functions

- **opticalModel** ()
- **~opticalModel** ()
- virtual void **InitOpenGL** ()
- virtual void **Render** ()
- virtual void **BindBuffer** (optix::Context context)
- virtual void **UpdateOpticalBuffer** ()
- virtual void **Reshape** (int width, int height)
- void **Modified** ()

Public Attributes

- **glslRayCast glsl**

Protected Attributes

- GLuint `m_pboOutputBuffer`

11.5.1 Constructor & Destructor Documentation

11.5.1.1 `opticalModel()`

```
opticalModel::opticalModel ( ) [inline]
```

11.5.1.2 `~opticalModel()`

```
opticalModel::~~opticalModel ( ) [inline]
```

11.5.2 Member Function Documentation

11.5.2.1 `BindBuffer()`

```
void opticalModel::BindBuffer (
    optix::Context context ) [virtual]
```

Reimplemented from **basicOpticalModel** (p. 35).

11.5.2.2 `InitOpenGL()`

```
void opticalModel::InitOpenGL ( ) [virtual]
```

Reimplemented from **basicOpticalModel** (p. 35).

11.5.2.3 `Modified()`

```
void opticalModel::Modified ( ) [inline]
```

11.5.2.4 Render()

```
virtual void opticalModel::Render ( ) [inline], [virtual]
```

Reimplemented from **basicOpticalModel** (p. 35).

11.5.2.5 Reshape()

```
void opticalModel::Reshape (
    int width,
    int height ) [virtual]
```

Reimplemented from **basicOpticalModel** (p. 36).

11.5.2.6 UpdateOpticalBuffer()

```
void opticalModel::UpdateOpticalBuffer ( ) [virtual]
```

Reimplemented from **basicOpticalModel** (p. 36).

11.5.3 Member Data Documentation

11.5.3.1 glsl

```
glslRayCast opticalModel::glsl
```

11.5.3.2 m_pboOutputBuffer

```
GLuint opticalModel::m_pboOutputBuffer [protected]
```

The documentation for this class was generated from the following files:

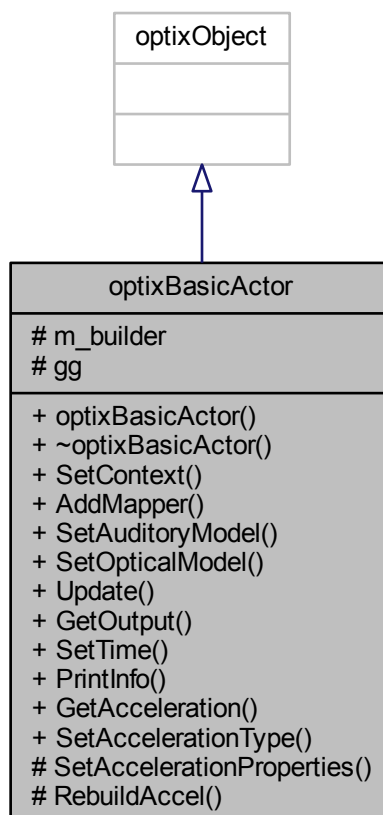
- **optixAudioOptic.h**
- **optixAudioOptic.cpp**

11.6 optixBasicActor Class Reference

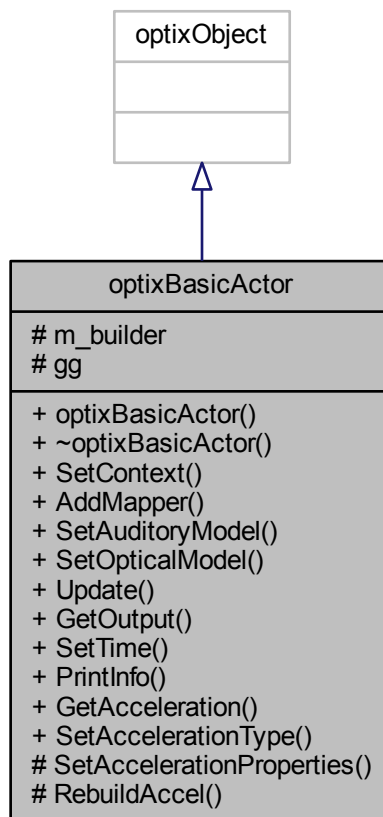
basic abstract actor class

```
#include <optixBasicActor.h>
```

Inheritance diagram for optixBasicActor:



Collaboration diagram for optixBasicActor:



Public Member Functions

- `optixBasicActor ()`
- `~optixBasicActor ()`
- virtual void **SetContext** (optix::Context &context)
- void **AddMapper** (`optixMapper` *map)
- void **SetAuditoryModel** ()
- void **SetOpticalModel** ()
- void **Update** ()
- optix::GeometryGroup **GetOutput** ()
- void **SetTime** (float time)
- void **PrintInfo** ()
- optix::Acceleration **GetAcceleration** ()
- void **SetAccelerationType** (std::string type)

Protected Member Functions

- virtual void **SetAccelerationProperties** ()
- virtual void **RebuildAccel** ()

Protected Attributes

- `std::string m_builder`
- `optix::GeometryGroup gg`

11.6.1 Detailed Description

basic abstract actor class

The class creates a basic Actor for Optix use. Architecture much inspired by VTK (<http://vtk.org>) implementation

11.6.2 Constructor & Destructor Documentation

11.6.2.1 `optixBasicActor()`

```
optixBasicActor::optixBasicActor ( ) [inline]
```

11.6.2.2 `~optixBasicActor()`

```
optixBasicActor::~~optixBasicActor ( ) [inline]
```

11.6.3 Member Function Documentation

11.6.3.1 `AddMapper()`

```
void optixBasicActor::AddMapper (
    optixMapper * map ) [inline]
```

11.6.3.2 `GetAcceleration()`

```
optix::Acceleration optixBasicActor::GetAcceleration ( ) [inline]
```

11.6.3.3 GetOutput()

```
optix::GeometryGroup optixBasicActor::GetOutput ( ) [inline]
```

11.6.3.4 PrintInfo()

```
void optixBasicActor::PrintInfo ( ) [inline]
```

11.6.3.5 RebuildAccel()

```
virtual void optixBasicActor::RebuildAccel ( ) [inline], [protected], [virtual]
```

11.6.3.6 SetAccelerationProperties()

```
virtual void optixBasicActor::SetAccelerationProperties ( ) [inline], [protected], [virtual]
```

11.6.3.7 SetAccelerationType()

```
void optixBasicActor::SetAccelerationType (
    std::string type ) [inline]
```

11.6.3.8 SetAuditoryModel()

```
void optixBasicActor::SetAuditoryModel ( ) [inline]
```

11.6.3.9 SetContext()

```
virtual void optixBasicActor::SetContext (
    optix::Context & context ) [inline], [virtual]
```

11.6.3.10 SetOpticalModel()

```
void optixBasicActor::SetOpticalModel ( ) [inline]
```

11.6.3.11 SetTime()

```
void optixBasicActor::SetTime (
    float time ) [inline]
```

11.6.3.12 Update()

```
void optixBasicActor::Update ( )
```

11.6.4 Member Data Documentation

11.6.4.1 gg

```
optix::GeometryGroup optixBasicActor::gg [protected]
```

11.6.4.2 m_builder

```
std::string optixBasicActor::m_builder [protected]
```

The documentation for this class was generated from the following files:

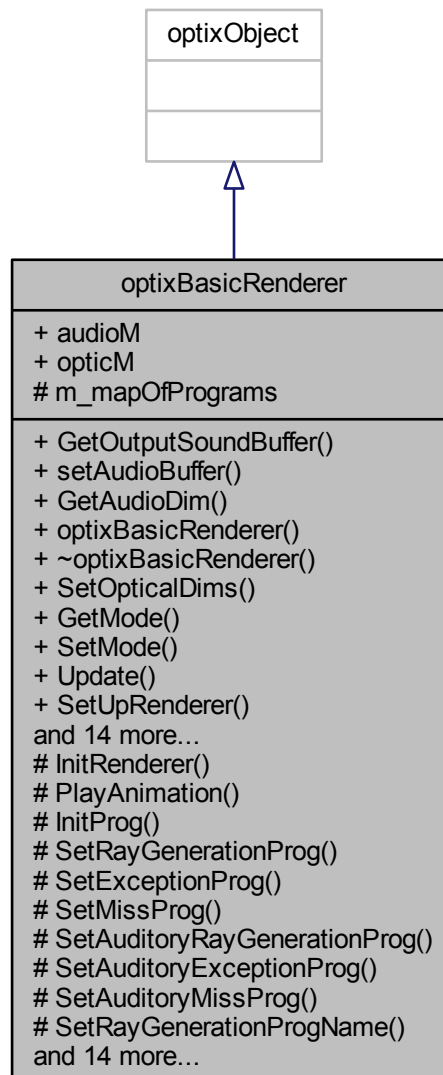
- **optixBasicActor.h**
- **optixBasicActor.cpp**

11.7 optixBasicRenderer Class Reference

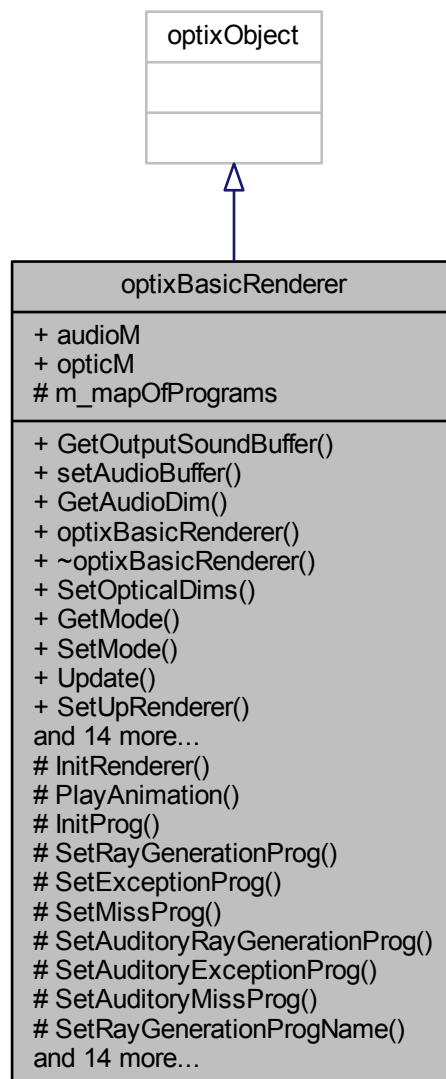
Architecture much inspired by VTK Renderer(<http://vtk.org>) implementation Abstract visual-auditory renderer.

```
#include <optixBasicRenderer.h>
```

Inheritance diagram for optixBasicRenderer:



Collaboration diagram for optixBasicRenderer:



Public Types

- enum { **OPTIC_RAYCASTING**, **AUDITORY_RAYCASTING** }
- enum **RenderModes** { **INTERACTIVE_CAMERA**, **COMPUTE_SOUND**, **PLAY_ANIMATION** }

Public Member Functions

- `optix::Buffer` **GetOutputSoundBuffer** ()
- `void` **setAudioBuffer** ()
- `void` **GetAudioDim** (int &w, int &h)
- `optixBasicRenderer` ()

- `~optixBasicRenderer ()`
- `void SetOpticalDims (int width, int height)`
- `RenderModes GetMode ()`
- `void SetMode (RenderModes m)`
- `virtual void Update ()`
- `void SetUpRenderer ()`
- `void SetDynamic (bool type)`
- `void SetAuditory (bool type)`
- `bool isDynamic ()`
- `bool isAuditory ()`
- `void SetTime (float time)`
- `float GetTime ()`
- `void AddActor (optixBasicActor *act)`
- `optixBasicActor * GetActor (int i)`
- `int GetNumOfActors ()`
- `void LaunchOpticContext ()`
- `void LaunchAuditoryContext ()`
- `virtual void Reshape (int width, int height)`
- `virtual bool Render ()`
- `virtual void Display ()`

Public Attributes

- `std::unique_ptr< auditoryModel > audioM`
- `std::unique_ptr< basicOpticalModel > opticM`

Protected Member Functions

- `virtual void InitRenderer ()`
- `bool PlayAnimation (float play_time)`
- `void InitProg (std::string prog, std::string file, std::string name)`
- `void SetRayGenerationProg ()`
- `void SetExceptionProg ()`
- `void SetMissProg ()`
- `void SetAuditoryRayGenerationProg ()`
- `void SetAuditoryExceptionProg ()`
- `void SetAuditoryMissProg ()`
- `void SetRayGenerationProgName (std::string name)`
- `void SetExceptionProgName (std::string name)`
- `void SetMissProgName (std::string name)`
- `void SetAuditoryRayGenerationProgName (std::string name)`
- `void SetAuditoryExceptionProgName (std::string name)`
- `void SetAuditoryMissProgName (std::string name)`
- `std::string GetRayGenerationProgName ()`
- `std::string GetExceptionProgName ()`
- `std::string GetMissProgName ()`
- `std::string GetAuditoryRayGenerationProgName ()`
- `std::string GetAuditoryExceptionProgName ()`
- `std::string GetAuditoryMissProgName ()`
- `void InitAcceleration ()`
- `void SetPrograms ()`
- `void InitializePrograms ()`

Protected Attributes

- `std::map< std::string, optix::Program > m_mapOfPrograms`

11.7.1 Detailed Description

Architecture much inspired by VTK Renderer(<http://vtk.org>) implementation Abstract visual-auditory renderer.

11.7.2 Member Enumeration Documentation

11.7.2.1 anonymous enum

anonymous enum

Enumerator

OPTIC_RAYCASTING	
AUDITORY_RAYCASTING	

11.7.2.2 RenderModes

enum `optixBasicRenderer::RenderModes`

Enumerator

INTERACTIVE_CAMERA	
COMPUTE_SOUND	
PLAY_ANIMATION	

11.7.3 Constructor & Destructor Documentation

11.7.3.1 optixBasicRenderer()

`optixBasicRenderer::optixBasicRenderer () [inline]`

11.7.3.2 ~optixBasicRenderer()

```
optixBasicRenderer::~~optixBasicRenderer ( ) [inline]
```

11.7.4 Member Function Documentation

11.7.4.1 AddActor()

```
void optixBasicRenderer::AddActor (
    optixBasicActor * act )
```

11.7.4.2 Display()

```
virtual void optixBasicRenderer::Display ( ) [inline], [virtual]
```

11.7.4.3 GetActor()

```
optixBasicActor* optixBasicRenderer::GetActor (
    int i ) [inline]
```

11.7.4.4 GetAudioDim()

```
void optixBasicRenderer::GetAudioDim (
    int & w,
    int & h ) [inline]
```

11.7.4.5 GetAuditoryExceptionProgName()

```
std::string optixBasicRenderer::GetAuditoryExceptionProgName ( ) [inline], [protected]
```

11.7.4.6 GetAuditoryMissProgName()

```
std::string optixBasicRenderer::GetAuditoryMissProgName ( ) [inline], [protected]
```

11.7.4.7 GetAuditoryRayGenerationProgName()

```
std::string optixBasicRenderer::GetAuditoryRayGenerationProgName ( ) [inline], [protected]
```

11.7.4.8 GetExceptionProgName()

```
std::string optixBasicRenderer::GetExceptionProgName ( ) [inline], [protected]
```

11.7.4.9 GetMissProgName()

```
std::string optixBasicRenderer::GetMissProgName ( ) [inline], [protected]
```

11.7.4.10 GetMode()

```
RenderModes optixBasicRenderer::GetMode ( ) [inline]
```

11.7.4.11 GetNumOfActors()

```
int optixBasicRenderer::GetNumOfActors ( ) [inline]
```

11.7.4.12 GetOutputSoundBuffer()

```
optix::Buffer optixBasicRenderer::GetOutputSoundBuffer ( ) [inline]
```

11.7.4.13 GetRayGenerationProgName()

```
std::string optixBasicRenderer::GetRayGenerationProgName ( ) [inline], [protected]
```

11.7.4.14 GetTime()

```
float optixBasicRenderer::GetTime ( ) [inline]
```

11.7.4.15 InitAcceleration()

```
void optixBasicRenderer::InitAcceleration ( ) [protected]
```

11.7.4.16 InitializePrograms()

```
void optixBasicRenderer::InitializePrograms ( ) [protected]
```

11.7.4.17 InitProg()

```
void optixBasicRenderer::InitProg (
    std::string prog,
    std::string file,
    std::string name ) [protected]
```

11.7.4.18 InitRenderer()

```
virtual void optixBasicRenderer::InitRenderer ( ) [inline], [protected], [virtual]
```

11.7.4.19 isAuditory()

```
bool optixBasicRenderer::isAuditory ( ) [inline]
```

11.7.4.20 isDynamic()

```
bool optixBasicRenderer::isDynamic ( ) [inline]
```

11.7.4.21 LaunchAuditoryContext()

```
void optixBasicRenderer::LaunchAuditoryContext ( )
```

11.7.4.22 LaunchOpticContext()

```
void optixBasicRenderer::LaunchOpticContext ( )
```

11.7.4.23 PlayAnimation()

```
bool optixBasicRenderer::PlayAnimation (
    float play_time ) [protected]
```

11.7.4.24 Render()

```
virtual bool optixBasicRenderer::Render ( ) [inline], [virtual]
```

11.7.4.25 Reshape()

```
virtual void optixBasicRenderer::Reshape (
    int width,
    int height ) [inline], [virtual]
```

11.7.4.26 setAudioBuffer()

```
void optixBasicRenderer::setAudioBuffer ( ) [inline]
```

11.7.4.27 SetAuditory()

```
void optixBasicRenderer::SetAuditory (
    bool type ) [inline]
```

11.7.4.28 SetAuditoryExceptionProg()

```
void optixBasicRenderer::SetAuditoryExceptionProg ( ) [protected]
```


11.7.4.29 SetAuditoryExceptionProgName()

```
void optixBasicRenderer::SetAuditoryExceptionProgName (
    std::string name ) [inline], [protected]
```

11.7.4.30 SetAuditoryMissProg()

```
void optixBasicRenderer::SetAuditoryMissProg ( ) [protected]
```

11.7.4.31 SetAuditoryMissProgName()

```
void optixBasicRenderer::SetAuditoryMissProgName (
    std::string name ) [inline], [protected]
```

11.7.4.32 SetAuditoryRayGenerationProg()

```
void optixBasicRenderer::SetAuditoryRayGenerationProg ( ) [protected]
```

11.7.4.33 SetAuditoryRayGenerationProgName()

```
void optixBasicRenderer::SetAuditoryRayGenerationProgName (
    std::string name ) [inline], [protected]
```

11.7.4.34 SetDynamic()

```
void optixBasicRenderer::SetDynamic (
    bool type ) [inline]
```

11.7.4.35 SetExceptionProg()

```
void optixBasicRenderer::SetExceptionProg ( ) [protected]
```

11.7.4.36 SetExceptionProgName()

```
void optixBasicRenderer::SetExceptionProgName (
    std::string name ) [inline], [protected]
```

11.7.4.37 SetMissProg()

```
void optixBasicRenderer::SetMissProg ( ) [protected]
```

11.7.4.38 SetMissProgName()

```
void optixBasicRenderer::SetMissProgName (
    std::string name ) [inline], [protected]
```

11.7.4.39 SetMode()

```
void optixBasicRenderer::SetMode (
    RenderModes m ) [inline]
```

11.7.4.40 SetOpticalDims()

```
void optixBasicRenderer::SetOpticalDims (
    int width,
    int height ) [inline]
```

11.7.4.41 SetPrograms()

```
void optixBasicRenderer::SetPrograms ( ) [protected]
```

11.7.4.42 SetRayGenerationProg()

```
void optixBasicRenderer::SetRayGenerationProg ( ) [protected]
```

11.7.4.43 SetRayGenerationProgName()

```
void optixBasicRenderer::SetRayGenerationProgName (
    std::string name ) [inline], [protected]
```

11.7.4.44 SetTime()

```
void optixBasicRenderer::SetTime (
    float time ) [inline]
```

11.7.4.45 SetUpRenderer()

```
void optixBasicRenderer::SetUpRenderer ( )
```

11.7.4.46 Update()

```
virtual void optixBasicRenderer::Update ( ) [inline], [virtual]
```

11.7.5 Member Data Documentation

11.7.5.1 audioM

```
std::unique_ptr< auditoryModel> optixBasicRenderer::audioM
```

11.7.5.2 m_mapOfPrograms

```
std::map<std::string, optix::Program> optixBasicRenderer::m_mapOfPrograms [protected]
```

11.7.5.3 opticM

```
std::unique_ptr< basicOpticalModel> optixBasicRenderer::opticM
```

The documentation for this class was generated from the following files:

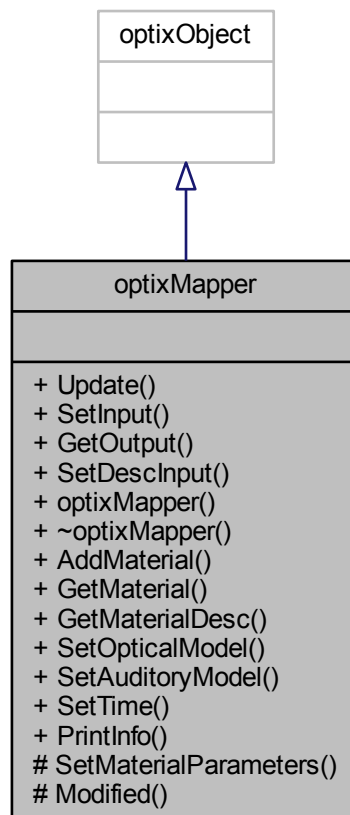
- **optixBasicRenderer.h**
- **optixBasicRenderer.cpp**

11.8 optixMapper Class Reference

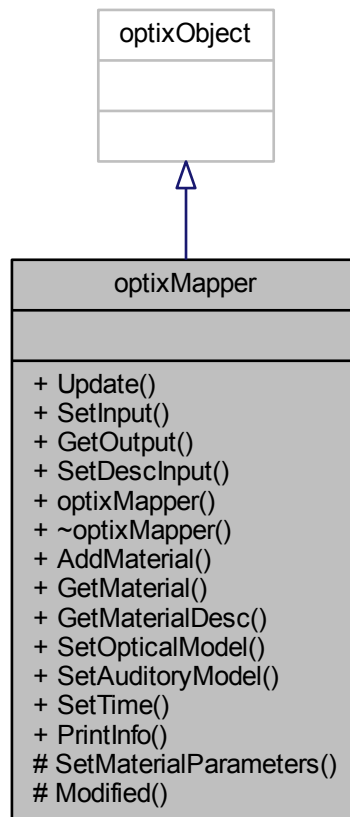
A basic mapper that suggest application of one visual and one auditory material to one geometry instance.

```
#include <optixAbstractMaterial.h>
```

Inheritance diagram for optixMapper:



Collaboration diagram for optixMapper:



Public Member Functions

- virtual void **Update** ()
- void **SetInput** (optix::Geometry g)
- optix::GeometryInstance **GetOutput** ()
- void **SetDescInput** (sdfGeo *g)
- **optixMapper** ()
- **~optixMapper** ()
- void **AddMaterial** (optix::Material m, **MaterialDesc** desc)
- optix::Material **GetMaterial** (int i)
- **MaterialDesc** **GetMaterialDesc** (int i)
- void **SetOpticalModel** ()
- void **SetAuditoryModel** ()
- void **SetTime** (float time)
- void **PrintInfo** ()

Protected Member Functions

- virtual void **SetMaterialParameters** (int i)
- virtual void **Modified** ()

11.8.1 Detailed Description

A basic mapper that suggest application of one visual and one auditory material to one geometry instance.

The codes inspired by vtkMapper class. Returns optix::GeometryInstance with assigned material. Each material suggests definition of optical model , that treats geometry defined object as a surface or takes geometry and heterogeneously defined attributes and maps them to optical and/or auditory properties for more details see ..heterogeneous objects modelling

The class takes optix geometry as input, assigns materials and returns optix GeometryInstance. Class also handles switching between optical and auditory material to perform visual and auditory rendering accordingly

11.8.2 Constructor & Destructor Documentation

11.8.2.1 optixMapper()

```
optixMapper::optixMapper ( ) [inline]
```

11.8.2.2 ~optixMapper()

```
optixMapper::~~optixMapper ( ) [inline]
```

11.8.3 Member Function Documentation

11.8.3.1 AddMaterial()

```
void optixMapper::AddMaterial (
    optix::Material m,
    MaterialDesc desc )
```

11.8.3.2 GetMaterial()

```
optix::Material optixMapper::GetMaterial (
    int i )
```

11.8.3.3 GetMaterialDesc()

```
MaterialDesc optixMapper::GetMaterialDesc (
    int i ) [inline]
```

11.8.3.4 GetOutput()

```
optix::GeometryInstance optixMapper::GetOutput ( ) [inline]
```

11.8.3.5 Modified()

```
virtual void optixMapper::Modified ( ) [inline], [protected], [virtual]
```

11.8.3.6 PrintInfo()

```
void optixMapper::PrintInfo ( ) [inline]
```

11.8.3.7 SetAuditoryModel()

```
void optixMapper::SetAuditoryModel ( )
```

11.8.3.8 SetDescInput()

```
void optixMapper::SetDescInput (
    sdfGeo * g ) [inline]
```

11.8.3.9 SetInput()

```
void optixMapper::SetInput (
    optix::Geometry g ) [inline]
```

11.8.3.10 SetMaterialParameters()

```
virtual void optixMapper::SetMaterialParameters (
    int i ) [inline], [protected], [virtual]
```

11.8.3.11 SetOpticalModel()

```
void optixMapper::SetOpticalModel ( )
```

11.8.3.12 SetTime()

```
void optixMapper::SetTime (
    float time ) [inline]
```

11.8.3.13 Update()

```
void optixMapper::Update ( ) [virtual]
```

The documentation for this class was generated from the following files:

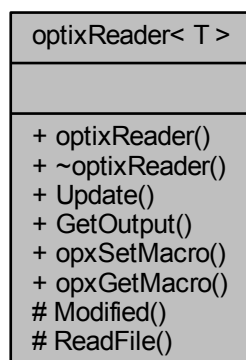
- **optixAbstractMaterial.h**
- **optixAbstractMaterial.cpp**

11.9 optixReader< T > Class Template Reference

The class creates a basic triangular plane for Optix use. Architecture much inspired by VTK (<http://vtk.org>) implementation.

```
#include <optixReader.h>
```

Collaboration diagram for optixReader< T >:



Public Member Functions

- **optixReader** ()
- **~optixReader** ()
- virtual void **Update** ()
- virtual std::shared_ptr< T > **GetOutput** ()
- **opxSetMacro** (file, std::string)
- **opxGetMacro** (file, std::string)

Protected Member Functions

- virtual void **Modified** ()
- virtual void **ReadFile** ()

11.9.1 Detailed Description

```
template<class T>
class optixReader< T >
```

The class creates a basic triangular plane for Optix use. Architecture much inspired by VTK (<http://vtk.org>) implementation.

11.9.2 Constructor & Destructor Documentation

11.9.2.1 optixReader()

```
template<class T >
optixReader< T >:: optixReader ( ) [inline]
```

11.9.2.2 ~optixReader()

```
template<class T >
optixReader< T >::~ ~optixReader ( ) [inline]
```

11.9.3 Member Function Documentation

11.9.3.1 GetOutput()

```
template<class T >
virtual std::shared_ptr<T> optixReader< T >::GetOutput ( ) [inline], [virtual]
```

11.9.3.2 Modified()

```
template<class T >
virtual void optixReader< T >::Modified ( ) [inline], [protected], [virtual]
```

11.9.3.3 opxGetMacro()

```
template<class T >
optixReader< T >::opxGetMacro (
    file ,
    std::string )
```

11.9.3.4 opxSetMacro()

```
template<class T >
optixReader< T >::opxSetMacro (
    file ,
    std::string )
```

11.9.3.5 ReadFile()

```
template<class T >
virtual void optixReader< T >::ReadFile ( ) [inline], [protected], [virtual]
```

11.9.3.6 Update()

```
template<class T >
virtual void optixReader< T >::Update ( ) [inline], [virtual]
```

The documentation for this class was generated from the following file:

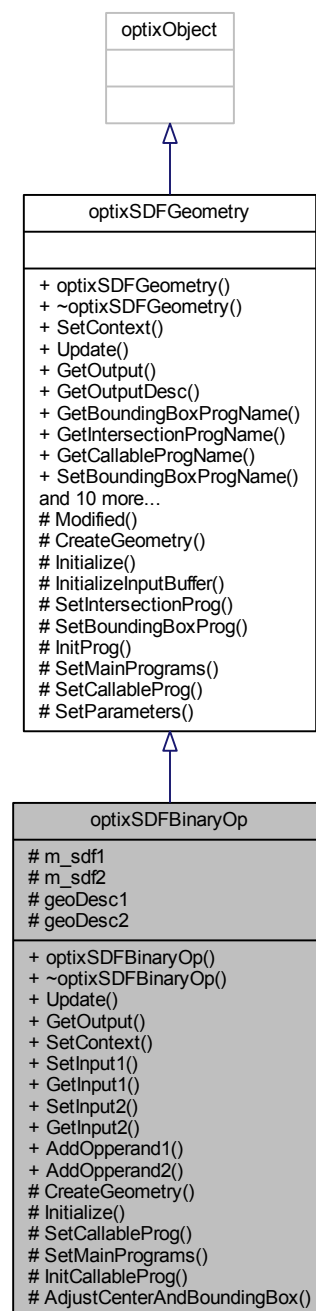
- **optixReader.h**

11.10 optixSDFBinaryOp Class Reference

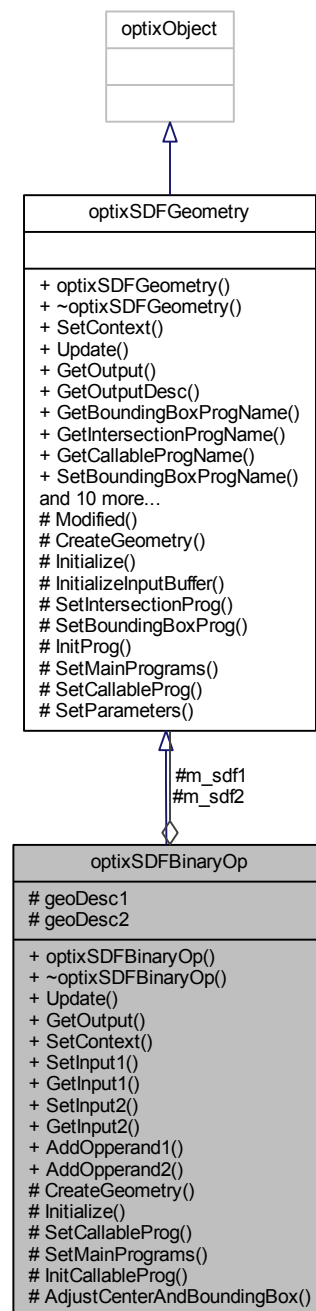
Binary operation.

```
#include <optixSDFBasicOperations.h>
```

Inheritance diagram for optixSDFBinaryOp:



Collaboration diagram for optixSDFBinaryOp:



Public Member Functions

- `optixSDFBinaryOp ()`
- `~optixSDFBinaryOp ()`
- virtual void **Update** ()
- virtual `optix::Geometry` **GetOutput** ()
- virtual void **SetContext** (`optix::Context &context`)

- virtual void **SetInput1** (optix::Geometry geo1)
- optix::Geometry **GetInput1** ()
- virtual void **SetInput2** (optix::Geometry geo1)
- optix::Geometry **GetInput2** ()
- void **AddOperand1** (**optixSDFGeometry** *sdf)
- void **AddOperand2** (**optixSDFGeometry** *sdf)

Protected Member Functions

- virtual void **CreateGeometry** ()
- virtual void **Initialize** ()
- virtual void **SetCallableProg** ()
- virtual void **SetMainPrograms** ()
- virtual void **InitCallableProg** ()
- virtual void **AdjustCenterAndBoundingBox** ()

Protected Attributes

- **optixSDFGeometry** * **m_sdf1**
- **optixSDFGeometry** * **m_sdf2**
- sdfGeo * **geoDesc1**
- sdfGeo * **geoDesc2**

11.10.1 Detailed Description

Binary operation.

Abstract class. Defines sdf Binary operation for FRep tree construction pipeline

11.10.2 Constructor & Destructor Documentation

11.10.2.1 optixSDFBinaryOp()

```
optixSDFBinaryOp::optixSDFBinaryOp ( ) [inline]
```

11.10.2.2 ~optixSDFBinaryOp()

```
optixSDFBinaryOp::~~optixSDFBinaryOp ( ) [inline]
```

11.10.3 Member Function Documentation

11.10.3.1 AddOpoperand1()

```
void optixSDFBinaryOp::AddOpoperand1 (
    optixSDFGeometry * sdf ) [inline]
```

11.10.3.2 AddOpoperand2()

```
void optixSDFBinaryOp::AddOpoperand2 (
    optixSDFGeometry * sdf ) [inline]
```

11.10.3.3 AdjustCenterAndBoundingBox()

```
virtual void optixSDFBinaryOp::AdjustCenterAndBoundingBox ( ) [inline], [protected], [virtual]
```

11.10.3.4 CreateGeometry()

```
void optixSDFBinaryOp::CreateGeometry ( ) [protected], [virtual]
```

Reimplemented from **optixSDFGeometry** (p. 78).

11.10.3.5 GetInput1()

```
optix::Geometry optixSDFBinaryOp::GetInput1 ( ) [inline]
```

11.10.3.6 GetInput2()

```
optix::Geometry optixSDFBinaryOp::GetInput2 ( ) [inline]
```

11.10.3.7 GetOutput()

```
virtual optix::Geometry optixSDFBinaryOp::GetOutput ( ) [inline], [virtual]
```

Reimplemented from **optixSDFGeometry** (p. 79).

11.10.3.8 InitCallableProg()

```
virtual void optixSDFBinaryOp::InitCallableProg ( ) [inline], [protected], [virtual]
```

11.10.3.9 Initialize()

```
void optixSDFBinaryOp::Initialize ( ) [protected], [virtual]
```

TODO: potentially read file with points here

Reimplemented from **optixSDFGeometry** (p. 79).

11.10.3.10 SetCallableProg()

```
void optixSDFBinaryOp::SetCallableProg ( ) [protected], [virtual]
```

Is called for setting callable prog

Reimplemented from **optixSDFGeometry** (p. 80).

11.10.3.11 SetContext()

```
void optixSDFBinaryOp::SetContext (
    optix::Context & context ) [virtual]
```

Sets optix context that will store all generated optix geometry

Reimplemented from **optixSDFGeometry** (p. 81).

11.10.3.12 SetInput1()

```
virtual void optixSDFBinaryOp::SetInput1 (
    optix::Geometry geo1 ) [inline], [virtual]
```

11.10.3.13 SetInput2()

```
virtual void optixSDFBinaryOp::SetInput2 (
    optix::Geometry geo1 ) [inline], [virtual]
```

11.10.3.14 SetMainPrograms()

```
void optixSDFBinaryOp::SetMainPrograms ( ) [protected], [virtual]
```

sets all main programs

Reimplemented from **optixSDFGeometry** (p. 81).

11.10.3.15 Update()

```
virtual void optixSDFBinaryOp::Update ( ) [inline], [virtual]
```

Runs the process of computation of optix geometry and setting up all necessary cuda programs and geometry parameters

Reimplemented from **optixSDFGeometry** (p. 82).

11.10.4 Member Data Documentation

11.10.4.1 geoDesc1

```
sdfGeo* optixSDFBinaryOp::geoDesc1 [protected]
```

11.10.4.2 geoDesc2

```
sdfGeo* optixSDFBinaryOp::geoDesc2 [protected]
```

11.10.4.3 m_sdf1

```
optixSDFGeometry* optixSDFBinaryOp::m_sdf1 [protected]
```

11.10.4.4 m_sdf2

```
optixSDFGeometry* optixSDFBinaryOp::m_sdf2 [protected]
```

The documentation for this class was generated from the following files:

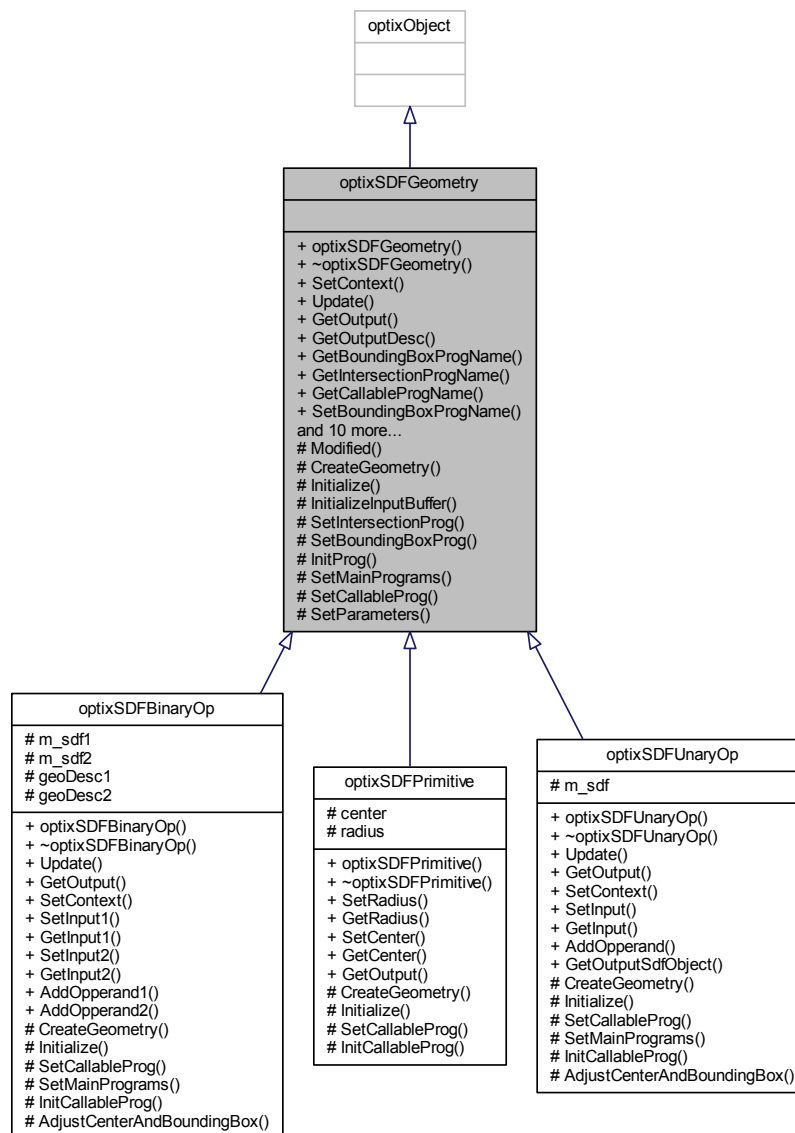
- **optixSDFBasicOperations.h**
- **optixSDFBasicOperations.cpp**

11.11 optixSDFGeometry Class Reference

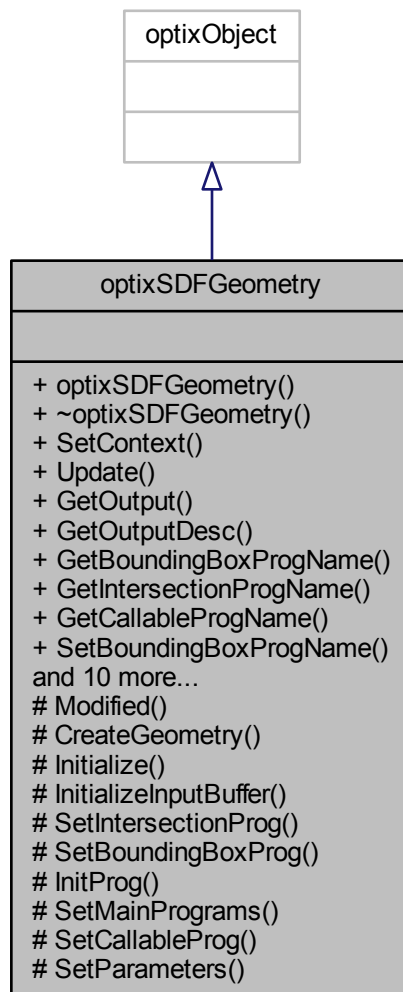
Abstract class for SDF Geometry. A basic class. All SDF geometry inherits it's functions.

```
#include <optixSDFGeometry.h>
```

Inheritance diagram for optixSDFGeometry:



Collaboration diagram for optixSDFGeometry:



Public Member Functions

- `optixSDFGeometry ()`
- `~optixSDFGeometry ()`
- virtual void **SetContext** (optix::Context &context)
- virtual void **Update** ()
- virtual optix::Geometry **GetOutput** ()
- virtual sdfGeo * **GetOutputDesc** ()
- std::string **GetBoundingBoxProgName** ()
- std::string **GetIntersectionProgName** ()
- std::string **GetCallableProgName** ()
- void **SetBoundingBoxProgName** (std::string bb)
- void **SetIntersectionProgName** (std::string inter)
- void **SetCallableProgName** (std::string c)

- optix::Program **GetCallableProg** ()
- void **SetCallableProgManually** (optix::Program pr)
- void **SetMaterialType** (int type)
- int **GetMaterialType** ()
- void **SetDynamic** (bool d)
- bool **isDynamic** ()
- virtual void **SetTime** (float time)
- virtual float **GetTime** ()

Protected Member Functions

- virtual void **Modified** ()
- virtual void **CreateGeometry** ()
- virtual void **Initialize** ()
- template<class T >
optix::Buffer **InitializeInputBuffer** (T Attributes, std::vector< T > attributes, optix::RTbuffermapflag mode)
- void **SetIntersectionProg** ()
- void **SetBoundingBoxProg** ()
- void **InitProg** (std::string prog, std::string file, std::string name)
- virtual void **SetMainPrograms** ()
- virtual void **SetCallableProg** ()
- virtual void **SetParameters** ()

11.11.1 Detailed Description

Abstract class for SDF Geometry. A basic class. All SDF geometry inherits it's functions.

All SDF geometry inherits it's functions, that includes:

1)SDF dynamic textures

2)FRep modelling primitives and operations that construct dynamically Construction Tree see <http://hyperfun.org> for details

3)Large objects, like molecular structures, that consist of lot's of objects and use BVH for rendering

...

may be something else

Architecture much inspired by VTK (<http://vtk.org>) implementation. Main operations similar to most of objects are much similar to VTK classes structure:

virtual void **SetContext**(optix::Context &context) (p. 81);

virtual void **Update**() (p. 82);

virtual optix::Geometry **GetOutput**() (p. 79)

virtual sdfGeo* **GetOutputDesc**() (p. 79)

See Tutorial for more info.

11.11.2 Constructor & Destructor Documentation

11.11.2.1 `optixSDFGeometry()`

```
optixSDFGeometry::optixSDFGeometry ( )
```

Constructor and destructor

11.11.2.2 `~optixSDFGeometry()`

```
optixSDFGeometry::~~optixSDFGeometry ( ) [inline]
```

11.11.3 Member Function Documentation

11.11.3.1 `CreateGeometry()`

```
virtual void optixSDFGeometry::CreateGeometry ( ) [inline], [protected], [virtual]
```

Reimplemented in **optixSDFBinaryOp** (p.72), **optixSDFPrimitive** (p.85), and **optixSDFUnaryOp** (p.91).

11.11.3.2 `GetBoundingBoxProgName()`

```
std::string optixSDFGeometry::GetBoundingBoxProgName ( ) [inline]
```

Returns description, a more detailed info, that is optix geometry and description wheter it is dynamic, auditory and etc.

11.11.3.3 `GetCallableProg()`

```
optix::Program optixSDFGeometry::GetCallableProg ( )
```

11.11.3.4 `GetCallableProgName()`

```
std::string optixSDFGeometry::GetCallableProgName ( ) [inline]
```

11.11.3.5 GetIntersectionProgName()

```
std::string optixSDFGeometry::GetIntersectionProgName ( ) [inline]
```

11.11.3.6 GetMaterialType()

```
int optixSDFGeometry::GetMaterialType ( )
```

11.11.3.7 GetOutput()

```
virtual optix::Geometry optixSDFGeometry::GetOutput ( ) [inline], [virtual]
```

Reimplemented in **optixSDFBinaryOp** (p.72), **optixSDFPrimitive** (p.86), and **optixSDFUnaryOp** (p.91).

11.11.3.8 GetOutputDesc()

```
virtual sdfGeo* optixSDFGeometry::GetOutputDesc ( ) [inline], [virtual]
```

Returns ready to use output, that is optix geometry

11.11.3.9 GetTime()

```
virtual float optixSDFGeometry::GetTime ( ) [inline], [virtual]
```

11.11.3.10 Initialize()

```
virtual void optixSDFGeometry::Initialize ( ) [inline], [protected], [virtual]
```

TODO: potentially read file with points here

Reimplemented in **optixSDFBinaryOp** (p.73), **optixSDFPrimitive** (p.86), and **optixSDFUnaryOp** (p.91).

11.11.3.11 InitializeInputBuffer()

```
template<class T >
optix::Buffer optixSDFGeometry::InitializeInputBuffer (
    T Attributes,
    std::vector< T > attributes,
    optix::RTbuffermapflag mode ) [protected]
```

read cuda procedure for SDF generation, the process is linked to name

11.11.3.12 InitProg()

```
void optixSDFGeometry::InitProg (
    std::string prog,
    std::string file,
    std::string name ) [protected]
```

11.11.3.13 isDynamic()

```
bool optixSDFGeometry::isDynamic ( ) [inline]
```

11.11.3.14 Modified()

```
virtual void optixSDFGeometry::Modified ( ) [inline], [protected], [virtual]
```

11.11.3.15 SetBoundingBoxProg()

```
void optixSDFGeometry::SetBoundingBoxProg ( ) [protected]
```

11.11.3.16 SetBoundingBoxProgName()

```
void optixSDFGeometry::SetBoundingBoxProgName (
    std::string bb ) [inline]
```

11.11.3.17 SetCallableProg()

```
virtual void optixSDFGeometry::SetCallableProg ( ) [inline], [protected], [virtual]
```

Is called for setting callable prog

Reimplemented in **optixSDFBinaryOp** (p. 73), **optixSDFPrimitive** (p. 86), and **optixSDFUnaryOp** (p. 92).

11.11.3.18 SetCallableProgManually()

```
void optixSDFGeometry::SetCallableProgManually (
    optix::Program pr )
```

11.11.3.19 SetCallableProgName()

```
void optixSDFGeometry::SetCallableProgName (
    std::string c ) [inline]
```

11.11.3.20 SetContext()

```
void optixSDFGeometry::SetContext (
    optix::Context & context ) [virtual]
```

Sets optix context that will store all generated optix geometry

Reimplemented in **optixSDFBinaryOp** (p.73), and **optixSDFUnaryOp** (p.92).

11.11.3.21 SetDynamic()

```
void optixSDFGeometry::SetDynamic (
    bool d ) [inline]
```

11.11.3.22 SetIntersectionProg()

```
void optixSDFGeometry::SetIntersectionProg ( ) [protected]
```

11.11.3.23 SetIntersectionProgName()

```
void optixSDFGeometry::SetIntersectionProgName (
    std::string inter ) [inline]
```

11.11.3.24 SetMainPrograms()

```
void optixSDFGeometry::SetMainPrograms ( ) [protected], [virtual]
```

sets all main programs

Reimplemented in **optixSDFBinaryOp** (p.73), and **optixSDFUnaryOp** (p.92).

11.11.3.25 SetMaterialType()

```
void optixSDFGeometry::SetMaterialType (
    int type )
```

11.11.3.26 SetParameters()

```
virtual void optixSDFGeometry::SetParameters ( ) [inline], [protected], [virtual]
```

for setting custom cuda variables values

11.11.3.27 SetTime()

```
virtual void optixSDFGeometry::SetTime (
    float time ) [inline], [virtual]
```

11.11.3.28 Update()

```
void optixSDFGeometry::Update ( ) [virtual]
```

Runs the process of computation of optix geometry and setting up all necessary cuda programs and geometry parameters

Reimplemented in **optixSDFBinaryOp** (p. 74), and **optixSDFUnaryOp** (p. 92).

The documentation for this class was generated from the following files:

- **optixSDFGeometry.h**
- **optixSDFGeometry.cpp**

11.12 optixSDFPrimitive Class Reference

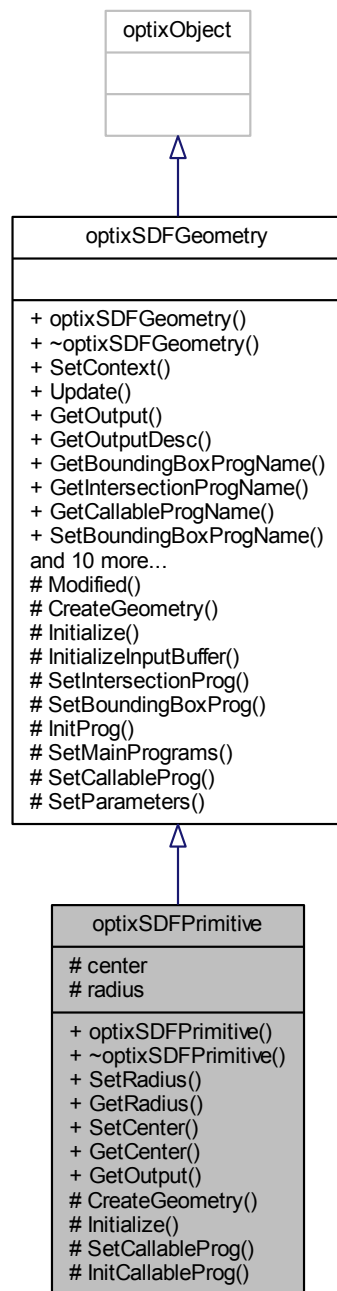
A basic class for all sdf primitives.

```
#include <optixSDFBasicPrimitives.h>
```


Inheritance diagram for optixSDFPrimitive:



Collaboration diagram for optixSDFPrimitive:



Public Member Functions

- `optixSDFPrimitive ()`
- `~optixSDFPrimitive ()`
- void `SetRadius` (optix::float3 rad)
- optix::float3 `GetRadius` ()
- void `SetCenter` (optix::float3 c)

- optix::float3 **GetCenter** ()
- virtual optix::Geometry **GetOutput** ()

Protected Member Functions

- virtual void **CreateGeometry** ()
- virtual void **Initialize** ()
- virtual void **SetCallableProg** ()
- virtual void **InitCallableProg** ()

Protected Attributes

- optix::float3 **center**
- optix::float3 **radius**

11.12.1 Detailed Description

A basic class for all sdf primitives.

The class creates a basic triangular plane for Optix use. Codes from Optix advanced samples. Architecture much inspired by VTK (<http://vtk.org>) implementation

11.12.2 Constructor & Destructor Documentation

11.12.2.1 optixSDFPrimitive()

```
optixSDFPrimitive::optixSDFPrimitive ( ) [inline]
```

11.12.2.2 ~optixSDFPrimitive()

```
optixSDFPrimitive::~~optixSDFPrimitive ( ) [inline]
```

11.12.3 Member Function Documentation

11.12.3.1 CreateGeometry()

```
void optixSDFPrimitive::CreateGeometry ( ) [protected], [virtual]
```

Reimplemented from **optixSDFGeometry** (p. 78).

11.12.3.2 GetCenter()

```
optix::float3 optixSDFPrimitive::GetCenter ( ) [inline]
```

11.12.3.3 GetOutput()

```
virtual optix::Geometry optixSDFPrimitive::GetOutput ( ) [inline], [virtual]
```

Reimplemented from **optixSDFGeometry** (p. 79).

11.12.3.4 GetRadius()

```
optix::float3 optixSDFPrimitive::GetRadius ( ) [inline]
```

11.12.3.5 InitCallableProg()

```
virtual void optixSDFPrimitive::InitCallableProg ( ) [inline], [protected], [virtual]
```

11.12.3.6 Initialize()

```
void optixSDFPrimitive::Initialize ( ) [protected], [virtual]
```

TODO: potentially read file with points here

Reimplemented from **optixSDFGeometry** (p. 79).

11.12.3.7 SetCallableProg()

```
void optixSDFPrimitive::SetCallableProg ( ) [protected], [virtual]
```

Is called for setting callable prog

Reimplemented from **optixSDFGeometry** (p. 80).

11.12.3.8 SetCenter()

```
void optixSDFPrimitive::SetCenter (
    optix::float3 c ) [inline]
```

11.12.3.9 SetRadius()

```
void optixSDFPrimitive::SetRadius (
    optix::float3 rad ) [inline]
```

11.12.4 Member Data Documentation

11.12.4.1 center

```
optix::float3 optixSDFPrimitive::center [protected]
```

11.12.4.2 radius

```
optix::float3 optixSDFPrimitive::radius [protected]
```

The documentation for this class was generated from the following files:

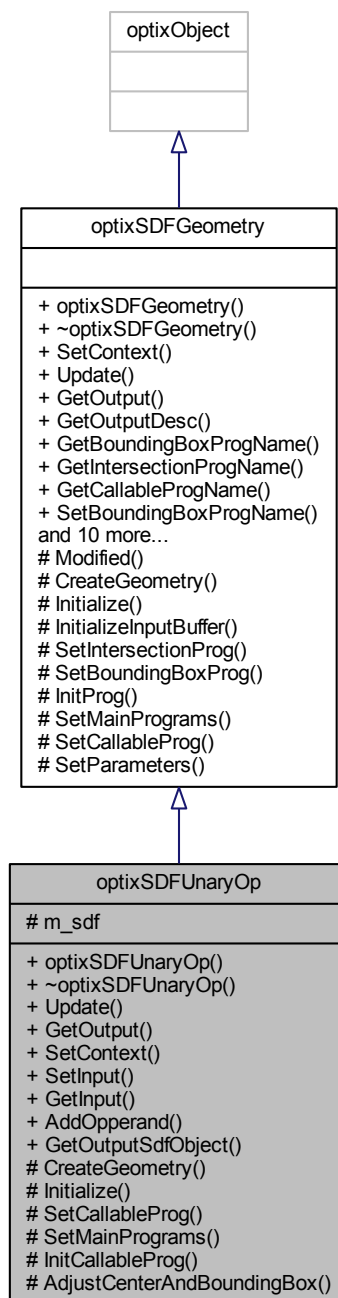
- **optixSDFBasicPrimitives.h**
- **optixSDFBasicPrimitives.cpp**

11.13 optixSDFUnaryOp Class Reference

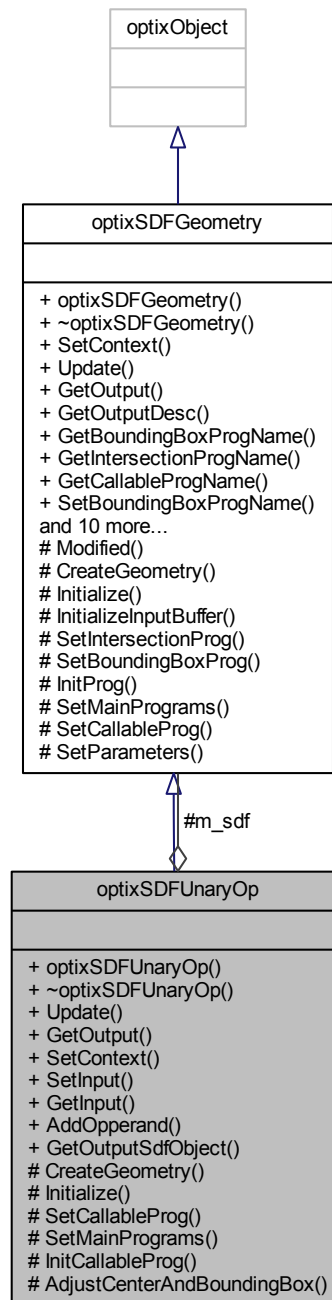
Unary operation.

```
#include <optixSDFBasicOperations.h>
```

Inheritance diagram for optixSDFUnaryOp:



Collaboration diagram for optixSDFUnaryOp:



Public Member Functions

- **optixSDFUnaryOp** ()
- **~optixSDFUnaryOp** ()
- virtual void **Update** ()
- virtual optix::Geometry **GetOutput** ()
- virtual void **SetContext** (optix::Context &context)

- virtual void **SetInput** (optix::Geometry geo1)
- optix::Geometry **GetInput** ()
- void **AddOppoperand** (optixSDFGeometry *sdf)
- optixSDFGeometry * **GetOutputSdfObject** ()

Protected Member Functions

- virtual void **CreateGeometry** ()
- virtual void **Initialize** ()
- virtual void **SetCallableProg** ()
- virtual void **SetMainPrograms** ()
- virtual void **InitCallableProg** ()
- virtual void **AdjustCenterAndBoundingBox** ()

Protected Attributes

- optixSDFGeometry * **m_sdf**

11.13.1 Detailed Description

Unary operation.

Abstract class. Defines sdf unary operation for FRep tree construction pipeline

11.13.2 Constructor & Destructor Documentation

11.13.2.1 optixSDFUnaryOp()

```
optixSDFUnaryOp::optixSDFUnaryOp ( ) [inline]
```

11.13.2.2 ~optixSDFUnaryOp()

```
optixSDFUnaryOp::~optixSDFUnaryOp ( ) [inline]
```

11.13.3 Member Function Documentation

11.13.3.1 AddOppoperand()

```
void optixSDFUnaryOp::AddOppoperand (
    optixSDFGeometry * sdf ) [inline]
```

11.13.3.2 AdjustCenterAndBoundingBox()

```
virtual void optixSDFUnaryOp::AdjustCenterAndBoundingBox ( ) [inline], [protected], [virtual]
```

11.13.3.3 CreateGeometry()

```
void optixSDFUnaryOp::CreateGeometry ( ) [protected], [virtual]
```

Reimplemented from **optixSDFGeometry** (p. 78).

11.13.3.4 GetInput()

```
optix::Geometry optixSDFUnaryOp::GetInput ( ) [inline]
```

11.13.3.5 GetOutput()

```
virtual optix::Geometry optixSDFUnaryOp::GetOutput ( ) [inline], [virtual]
```

Reimplemented from **optixSDFGeometry** (p. 79).

11.13.3.6 GetOutputSdfObject()

```
optixSDFGeometry* optixSDFUnaryOp::GetOutputSdfObject ( ) [inline]
```

11.13.3.7 InitCallableProg()

```
virtual void optixSDFUnaryOp::InitCallableProg ( ) [inline], [protected], [virtual]
```

11.13.3.8 Initialize()

```
void optixSDFUnaryOp::Initialize ( ) [protected], [virtual]
```

TODO: potentially read file with points here

Reimplemented from **optixSDFGeometry** (p. 79).

11.13.3.9 SetCallableProg()

```
void optixSDFUnaryOp::SetCallableProg ( ) [protected], [virtual]
```

Is called for setting callable prog

Reimplemented from **optixSDFGeometry** (p. 80).

11.13.3.10 SetContext()

```
void optixSDFUnaryOp::SetContext (
    optix::Context & context ) [virtual]
```

Sets optix context that will store all generated optix geometry

Reimplemented from **optixSDFGeometry** (p. 81).

11.13.3.11 SetInput()

```
virtual void optixSDFUnaryOp::SetInput (
    optix::Geometry geom ) [inline], [virtual]
```

11.13.3.12 SetMainPrograms()

```
void optixSDFUnaryOp::SetMainPrograms ( ) [protected], [virtual]
```

sets all main programs

Reimplemented from **optixSDFGeometry** (p. 81).

11.13.3.13 Update()

```
virtual void optixSDFUnaryOp::Update ( ) [inline], [virtual]
```

Runs the process of computation of optix geometry and setting up all necessary cuda programs and geometry parameters

Reimplemented from **optixSDFGeometry** (p. 82).

11.13.4 Member Data Documentation

11.13.4.1 m_sdf

```
optixSDFGeometry* optixSDFUnaryOp::m_sdf [protected]
```

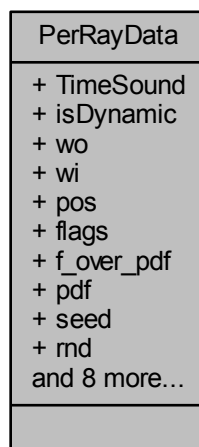
The documentation for this class was generated from the following files:

- **optixSDFBasicOperations.h**
- **optixSDFBasicOperations.cpp**

11.14 PerRayData Struct Reference

```
#include <per_ray_data.h>
```

Collaboration diagram for PerRayData:



Public Attributes

- float **TimeSound**
- int **isDynamic**
- optix::float3 **wo**
- optix::float3 **wi**
- optix::float3 **pos**
- int **flags**
- optix::float3 **f_over_pdf**
- float **pdf**
- unsigned int **seed**
- float **rnd**
- int **depth**
- optix::float3 **radiance**
- optix::float4 **result**
- int **cur_prim**
- bool **isSoundRay**
- int **numS**
- **auditoryPrim** primitives
- optix::float3 **Distances** [**MAX_PRIM_ALONG_RAY**]

11.14.1 Member Data Documentation

11.14.1.1 cur_prim

```
int PerRayData::cur_prim
```

11.14.1.2 depth

```
int PerRayData::depth
```

11.14.1.3 Distances

```
optix::float3 PerRayData::Distances[ MAX_PRIM_ALONG_RAY]
```

11.14.1.4 f_over_pdf

```
optix::float3 PerRayData::f_over_pdf
```

11.14.1.5 flags

```
int PerRayData::flags
```

11.14.1.6 isDynamic

```
int PerRayData::isDynamic
```

11.14.1.7 isSoundRay

```
bool PerRayData::isSoundRay
```

11.14.1.8 numS

```
int PerRayData::numS
```

11.14.1.9 pdf

```
float PerRayData::pdf
```

11.14.1.10 pos

```
optix::float3 PerRayData::pos
```

11.14.1.11 primitives

```
auditoryPrim PerRayData::primitives
```

11.14.1.12 radiance

```
optix::float3 PerRayData::radiance
```

11.14.1.13 result

```
optix::float4 PerRayData::result
```

11.14.1.14 rnd

```
float PerRayData::rnd
```

11.14.1.15 seed

```
unsigned int PerRayData::seed
```

11.14.1.16 TimeSound

```
float PerRayData::TimeSound
```

11.14.1.17 wi

```
optix::float3 PerRayData::wi
```

11.14.1.18 wo

```
optix::float3 PerRayData::wo
```

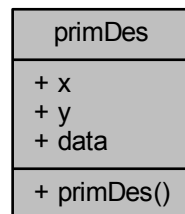
The documentation for this struct was generated from the following file:

- `shaders/renderer/per_ray_data.h`

11.15 primDes Struct Reference

```
#include <per_ray_data.h>
```

Collaboration diagram for primDes:



Public Member Functions

- `primDes` (int xx, int yy, `auditoryPrim` data_t)

Public Attributes

- int `x`
- int `y`
- `auditoryPrim` `data`

11.15.1 Constructor & Destructor Documentation

11.15.1.1 `primDes()`

```
primDes::primDes (  
    int xx,  
    int yy,  
    auditoryPrim data_t ) [inline]
```

11.15.2 Member Data Documentation

11.15.2.1 data

```
auditoryPrim primDes::data
```

11.15.2.2 x

```
int primDes::x
```

11.15.2.3 y

```
int primDes::y
```

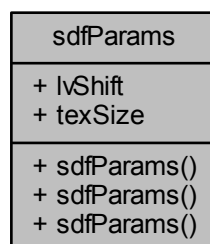
The documentation for this struct was generated from the following file:

- shaders/renderer/ **per_ray_data.h**

11.16 sdfParams Struct Reference

```
#include <renderTypes.h>
```

Collaboration diagram for sdfParams:



Public Member Functions

- **sdfParams** ()
- **sdfParams** (float l, int tex)
- **sdfParams** (**sdfParams** *param)

Public Attributes

- float **lvShift**
- int **texSize**

11.16.1 Constructor & Destructor Documentation

11.16.1.1 sdfParams() [1/3]

```
sdfParams::sdfParams ( ) [inline]
```

11.16.1.2 sdfParams() [2/3]

```
sdfParams::sdfParams (
    float l,
    int tex ) [inline]
```

11.16.1.3 sdfParams() [3/3]

```
sdfParams::sdfParams (
    sdfParams * param ) [inline]
```

11.16.2 Member Data Documentation

11.16.2.1 lvShift

```
float sdfParams::lvShift
```

11.16.2.2 texSize

```
int sdfParams::texSize
```

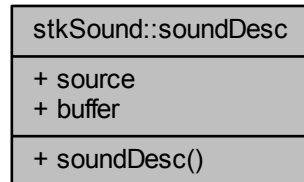
The documentation for this struct was generated from the following file:

- **renderTypes.h**

11.17 stkSound::soundDesc Struct Reference

```
#include <stkSound.h>
```

Collaboration diagram for stkSound::soundDesc:



Public Member Functions

- **soundDesc** (ALuint s, ALuint b)

Public Attributes

- ALuint **source**
- ALuint **buffer**

11.17.1 Constructor & Destructor Documentation

11.17.1.1 soundDesc()

```
stkSound::soundDesc::soundDesc (  
    ALuint s,  
    ALuint b ) [inline]
```

11.17.2 Member Data Documentation

11.17.2.1 buffer

```
ALuint stkSound::soundDesc::buffer
```

11.17.2.2 source

```
ALuint stkSound::soundDesc::source
```

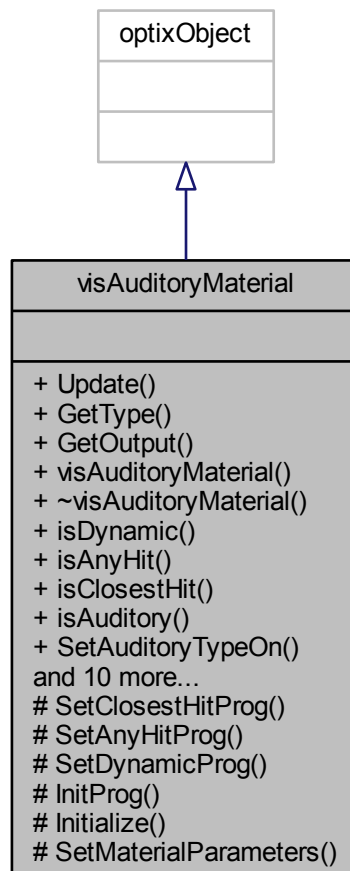
The documentation for this struct was generated from the following file:

- `stkSound.h`

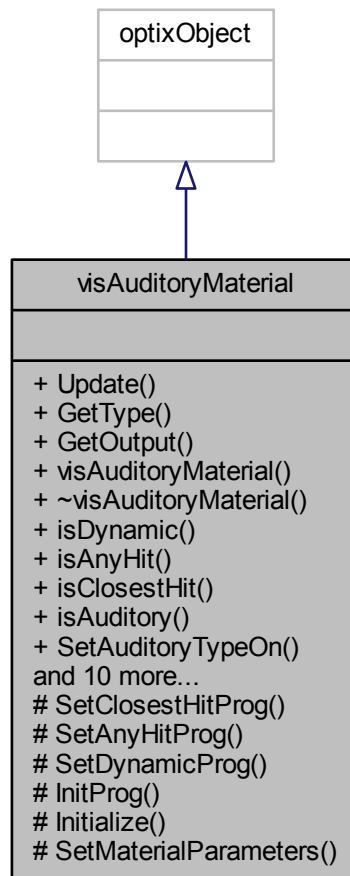
11.18 visAuditoryMaterial Class Reference

```
#include <optixAbstractMaterial.h>
```

Inheritance diagram for visAuditoryMaterial:



Collaboration diagram for visAuditoryMaterial:



Public Member Functions

- virtual void **Update** ()
- **MaterialDesc** **GetType** ()
- optix::Material **GetOutput** ()
- **visAuditoryMaterial** ()
- **~visAuditoryMaterial** ()
- bool **isDynamic** ()
- bool **isAnyHit** ()
- bool **isClosestHit** ()
- bool **isAuditory** ()
- void **SetAuditoryTypeOn** ()
- void **SetAuditoryTypeOff** ()
- void **SetClosestHit** (bool type)
- void **SetAnyHit** (bool type)
- std::string **GetClosestHitProgName** ()
- std::string **GetAnyHitProgName** ()
- std::string **GetDynamicProgName** ()

- void **SetClosestHitProgName** (std::string name)
- void **SetAnyHitProgName** (std::string name)
- void **SetDynamicProgName** (std::string name)
- std::string **GetProgName** (int i)

Protected Member Functions

- void **SetClosestHitProg** (optix::Material m)
- void **SetAnyHitProg** (optix::Material m)
- void **SetDynamicProg** (optix::Material m)
- void **InitProg** (std::string prog, std::string file, std::string name)
- virtual void **Initialize** ()
- virtual void **SetMaterialParameters** ()

11.18.1 Constructor & Destructor Documentation

11.18.1.1 visAuditoryMaterial()

```
visAuditoryMaterial::visAuditoryMaterial ( ) [inline]
```

11.18.1.2 ~visAuditoryMaterial()

```
visAuditoryMaterial::~~visAuditoryMaterial ( ) [inline]
```

11.18.2 Member Function Documentation

11.18.2.1 GetAnyHitProgName()

```
std::string visAuditoryMaterial::GetAnyHitProgName ( ) [inline]
```

11.18.2.2 GetClosestHitProgName()

```
std::string visAuditoryMaterial::GetClosestHitProgName ( ) [inline]
```

11.18.2.3 GetDynamicProgName()

```
std::string visAuditoryMaterial::GetDynamicProgName ( ) [inline]
```

11.18.2.4 GetOutput()

```
optix::Material visAuditoryMaterial::GetOutput ( ) [inline]
```

11.18.2.5 GetProgName()

```
std::string visAuditoryMaterial::GetProgName (
    int i ) [inline]
```

11.18.2.6 GetType()

```
MaterialDesc visAuditoryMaterial::GetType ( ) [inline]
```

11.18.2.7 Initialize()

```
virtual void visAuditoryMaterial::Initialize ( ) [inline], [protected], [virtual]
```

11.18.2.8 InitProg()

```
void visAuditoryMaterial::InitProg (
    std::string prog,
    std::string file,
    std::string name ) [protected]
```

11.18.2.9 isAnyHit()

```
bool visAuditoryMaterial::isAnyHit ( ) [inline]
```

11.18.2.10 isAuditory()

```
bool visAuditoryMaterial::isAuditory ( ) [inline]
```

11.18.2.11 isClosestHit()

```
bool visAuditoryMaterial::isClosestHit ( ) [inline]
```

11.18.2.12 isDynamic()

```
bool visAuditoryMaterial::isDynamic ( ) [inline]
```

11.18.2.13 SetAnyHit()

```
void visAuditoryMaterial::SetAnyHit (
    bool type ) [inline]
```

11.18.2.14 SetAnyHitProg()

```
void visAuditoryMaterial::SetAnyHitProg (
    optix::Material m ) [protected]
```

11.18.2.15 SetAnyHitProgName()

```
void visAuditoryMaterial::SetAnyHitProgName (
    std::string name ) [inline]
```

11.18.2.16 SetAuditoryTypeOff()

```
void visAuditoryMaterial::SetAuditoryTypeOff ( ) [inline]
```

11.18.2.17 SetAuditoryTypeOn()

```
void visAuditoryMaterial::SetAuditoryTypeOn ( ) [inline]
```

11.18.2.18 SetClosestHit()

```
void visAuditoryMaterial::SetClosestHit (
    bool type ) [inline]
```

11.18.2.19 SetClosestHitProg()

```
void visAuditoryMaterial::SetClosestHitProg (
    optix::Material m ) [protected]
```

11.18.2.20 SetClosestHitProgName()

```
void visAuditoryMaterial::SetClosestHitProgName (
    std::string name ) [inline]
```

11.18.2.21 SetDynamicProg()

```
void visAuditoryMaterial::SetDynamicProg (
    optix::Material m ) [protected]
```

11.18.2.22 SetDynamicProgName()

```
void visAuditoryMaterial::SetDynamicProgName (
    std::string name ) [inline]
```

11.18.2.23 SetMaterialParameters()

```
virtual void visAuditoryMaterial::SetMaterialParameters ( ) [inline], [protected], [virtual]
```


11.18.2.24 Update()

```
void visAuditoryMaterial::Update ( ) [virtual]
```

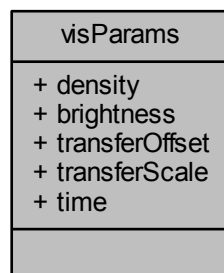
The documentation for this class was generated from the following files:

- `optixAbstractMaterial.h`
- `optixAbstractMaterial.cpp`

11.19 visParams Struct Reference

```
#include <renderTypes.h>
```

Collaboration diagram for visParams:



Public Attributes

- float **density**
- float **brightness**
- float **transferOffset**
- float **transferScale**
- float **time**

11.19.1 Member Data Documentation

11.19.1.1 brightness

```
float visParams::brightness
```

11.19.1.2 density

```
float visParams::density
```

11.19.1.3 time

```
float visParams::time
```

11.19.1.4 transferOffset

```
float visParams::transferOffset
```

11.19.1.5 transferScale

```
float visParams::transferScale
```

The documentation for this struct was generated from the following file:

- **renderTypes.h**

Chapter 12

File Documentation

12.1 doc/approach.md File Reference

12.2 doc/freptutorial.md File Reference

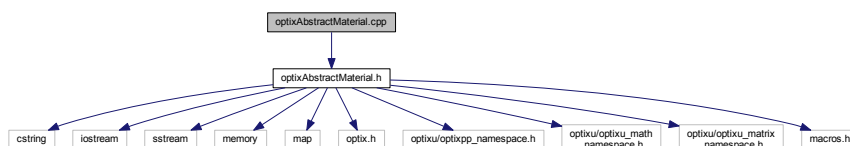
12.3 doc/freptutorial_test.md File Reference

12.4 doc/vtkstyle.md File Reference

12.5 optixAbstractMaterial.cpp File Reference

```
#include "optixAbstractMaterial.h"
```

Include dependency graph for optixAbstractMaterial.cpp:

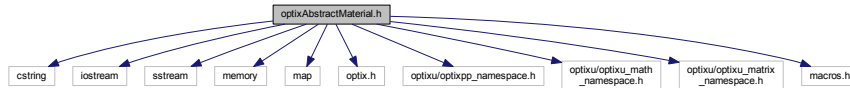


12.6 optixAbstractMaterial.h File Reference

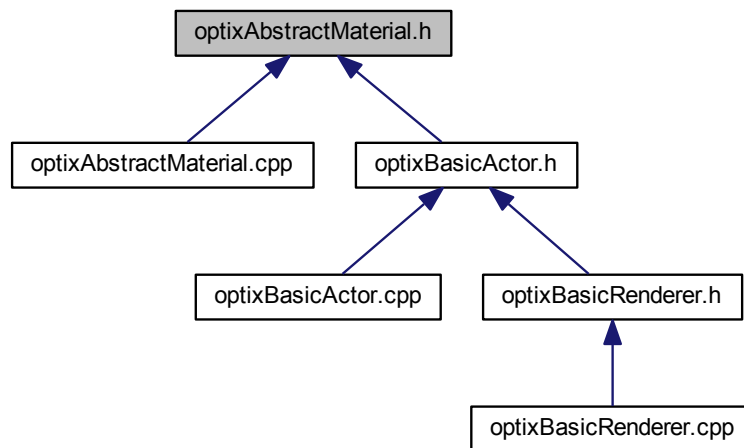
```
#include <cstring>
#include <iostream>
#include <sstream>
#include <memory>
#include <map>
#include <optix.h>
#include <optixu/optixpp_namespace.h>
```

```
#include <optixu/optixu_math_namespace.h>
#include <optixu/optixu_matrix_namespace.h>
#include "macros.h"
```

Include dependency graph for optixAbstractMaterial.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct **MaterialDesc**
- class **visAuditoryMaterial**
- class **optixMapper**

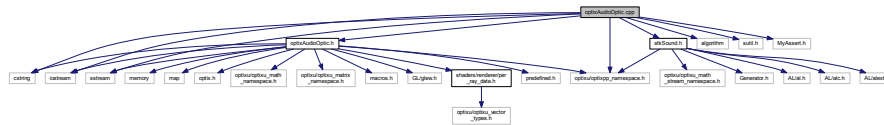
A basic mapper that suggest application of one visual and one auditory material to one geometry instance.

12.7 optixAudioOptic.cpp File Reference

```
#include "optixAudioOptic.h"
#include <algorithm>
#include <cstring>
#include <iostream>
#include <sstream>
#include <sutil.h>
#include <optixu/optixpp_namespace.h>
#include "MyAssert.h"
```

```
#include "stkSound.h"
```

Include dependency graph for optixAudioOptic.cpp:



Macros

- #define **LOAD_PROC**(d, x, y) ((x) = (y)alcGetProcAddress((d), #x))

12.7.1 Macro Definition Documentation

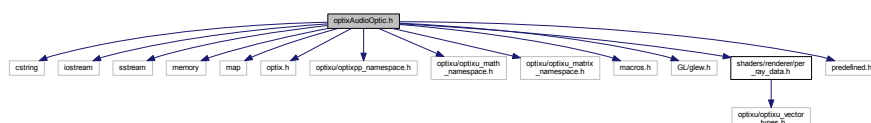
12.7.1.1 LOAD_PROC

```
#define LOAD_PROC(
    d,
    x,
    y ) ((x) = (y)alcGetProcAddress((d), #x))
```

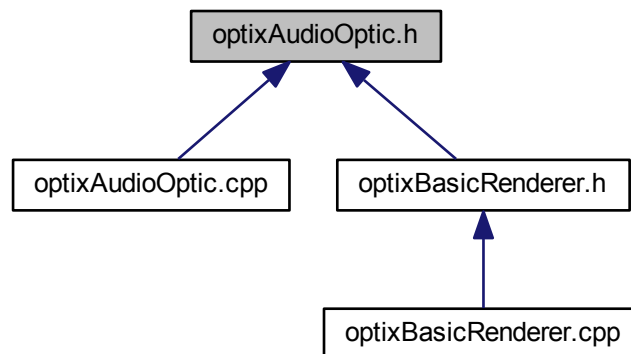
12.8 optixAudioOptic.h File Reference

```
#include <cstring>
#include <iostream>
#include <sstream>
#include <memory>
#include <map>
#include <optix.h>
#include <optixu/optixpp_namespace.h>
#include <optixu/optixu_math_namespace.h>
#include <optixu/optixu_matrix_namespace.h>
#include "macros.h"
#include <GL/glew.h>
#include "shaders/renderer/per_ray_data.h"
#include "predefined.h"
```

Include dependency graph for optixAudioOptic.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **auditoryModel**

Architecture much inspired by VTK Renderer(<http://vtk.org>) implementation The classes are mainly responsible for openGL/openAL interop.

- class **basicOpticalModel**
- class **glslRayCast**
- class **opticalModel**

Macros

- **#define optixAudioOptic_h**

12.8.1 Macro Definition Documentation

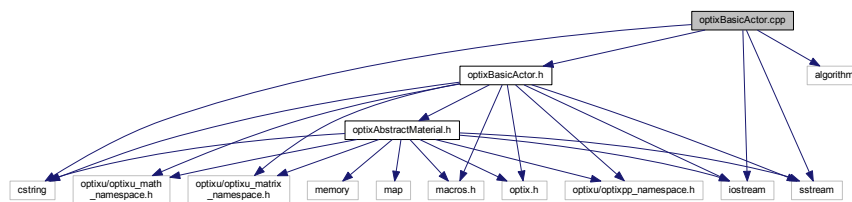
12.8.1.1 optixAudioOptic_h

```
#define optixAudioOptic_h
```

12.9 optixBasicActor.cpp File Reference

```
#include "optixBasicActor.h"
#include <algorithm>
#include <cstring>
#include <iostream>
#include <sstream>
```

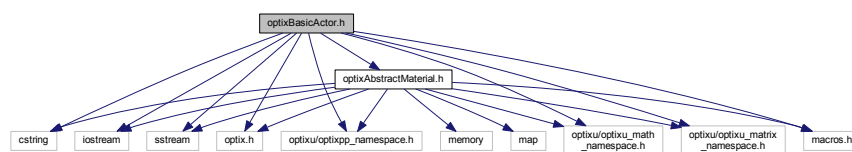
Include dependency graph for optixBasicActor.cpp:



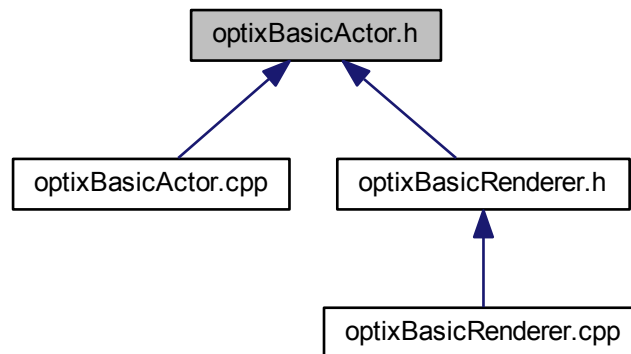
12.10 optixBasicActor.h File Reference

```
#include <cstring>
#include <iostream>
#include <sstream>
#include <optix.h>
#include <optixu/optixpp_namespace.h>
#include <optixu/optixu_math_namespace.h>
#include <optixu/optixu_matrix_namespace.h>
#include "macros.h"
#include "optixAbstractMaterial.h"
```

Include dependency graph for optixBasicActor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **optixBasicActor**
basic abstract actor class

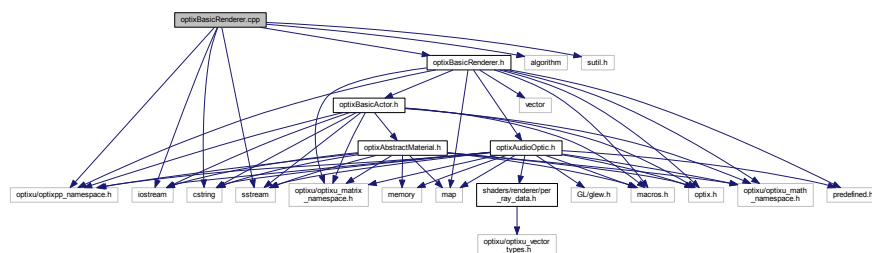
12.11 optixBasicRenderer.cpp File Reference

```

#include "optixBasicRenderer.h"
#include <algorithm>
#include <cstring>
#include <iostream>
#include <sstream>
#include <sutil.h>
#include <optixu/optixpp_namespace.h>

```

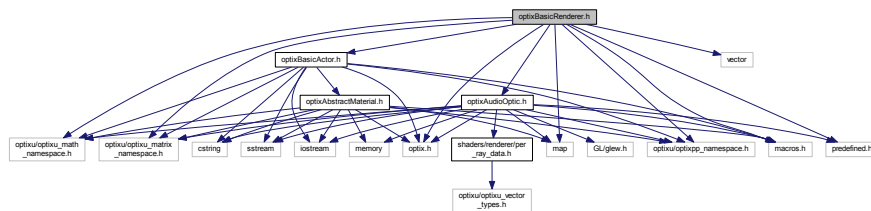
Include dependency graph for **optixBasicRenderer.cpp**:



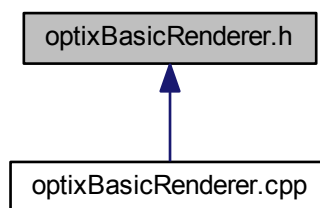
12.12 optixBasicRenderer.h File Reference

```
#include <optix.h>
#include <optixu/optixpp_namespace.h>
#include <optixu/optixu_math_namespace.h>
#include <optixu/optixu_matrix_namespace.h>
#include <vector>
#include <map>
#include "macros.h"
#include "predefined.h"
#include "optixAudioOptic.h"
#include "optixBasicActor.h"
```

Include dependency graph for optixBasicRenderer.h:



This graph shows which files directly or indirectly include this file:



Classes

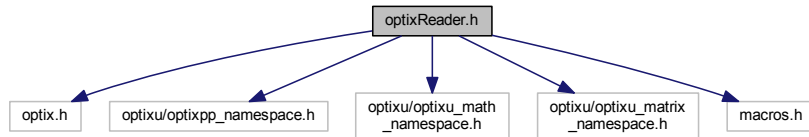
- class **optixBasicRenderer**

Architecture much inspired by VTK Renderer(<http://vtk.org>) implementation Abstract visual-auditory renderer.

12.13 optixReader.h File Reference

```
#include <optix.h>
#include <optixu/optixpp_namespace.h>
#include <optixu/optixu_math_namespace.h>
```

```
#include <optixu/optixu_matrix_namespace.h>
#include "macros.h"
Include dependency graph for optixReader.h:
```



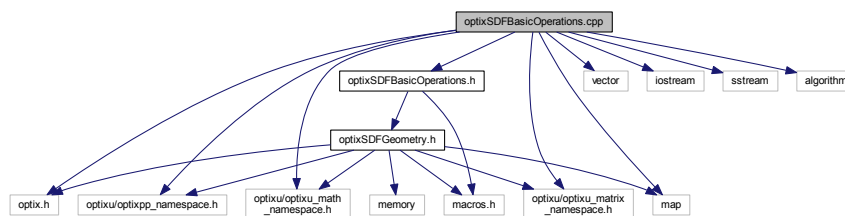
Classes

- class **optixReader**< T >

The class creates a basic triangular plane for Optix use. Architecture much inspired by VTK (<http://vtk.org>) implementation.

12.14 optixSDFBasicOperations.cpp File Reference

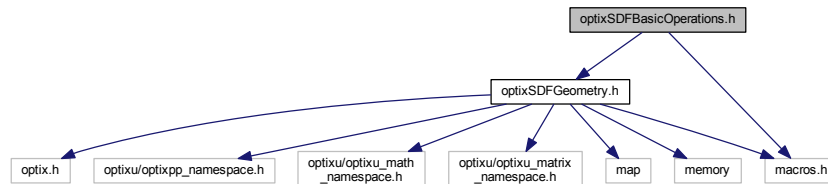
```
#include <optix.h>
#include <optixu/optixpp_namespace.h>
#include <optixu/optixu_math_namespace.h>
#include <optixu/optixu_matrix_namespace.h>
#include <vector>
#include <map>
#include <iostream>
#include <sstream>
#include <algorithm>
#include "optixSDFBasicOperations.h"
Include dependency graph for optixSDFBasicOperations.cpp:
```



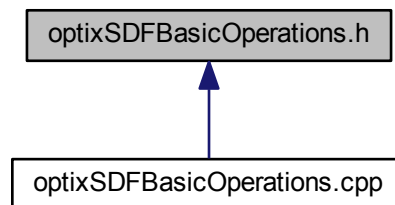
12.15 optixSDFBasicOperations.h File Reference

```
#include "optixSDFGeometry.h"
#include "macros.h"
```

Include dependency graph for optixSDFBasicOperations.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **optixSDFUnaryOp**
Unary operation.
- class **optixSDFBinaryOp**
Binary operation.

12.16 optixSDFBasicPrimitives.cpp File Reference

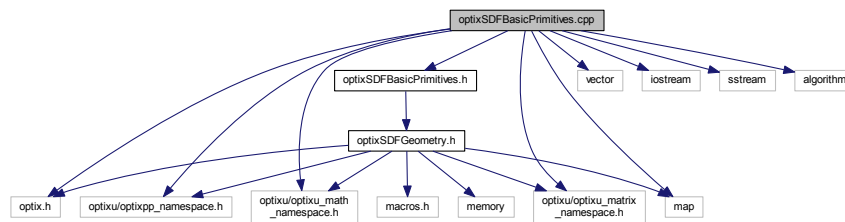
```

#include <optix.h>
#include <optixu/optixpp_namespace.h>
#include <optixu/optixu_math_namespace.h>
#include <optixu/optixu_matrix_namespace.h>
#include <vector>
#include <map>
#include <iostream>
#include <sstream>
#include <algorithm>

```

```
#include "optixSDFBasicPrimitives.h"
```

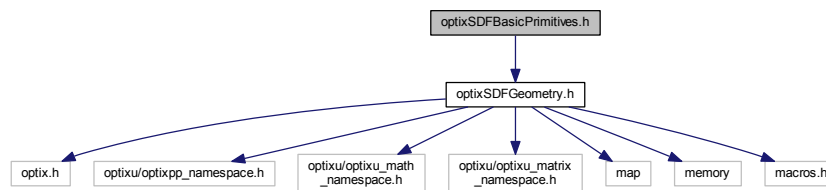
Include dependency graph for optixSDFBasicPrimitives.cpp:



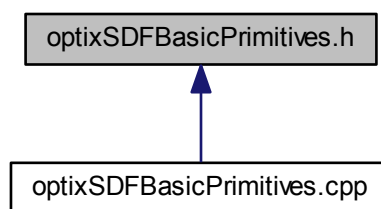
12.17 optixSDFBasicPrimitives.h File Reference

```
#include "optixSDFGeometry.h"
```

Include dependency graph for optixSDFBasicPrimitives.h:



This graph shows which files directly or indirectly include this file:

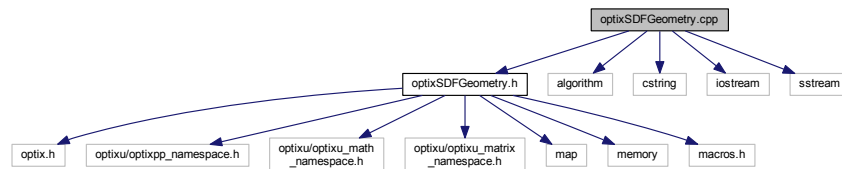


Classes

- class **optixSDFPrimitive**
A basic class for all sdf primitives.

12.18 optixSDFGeometry.cpp File Reference

```
#include "optixSDFGeometry.h"
#include <algorithm>
#include <cstring>
#include <iostream>
#include <sstream>
Include dependency graph for optixSDFGeometry.cpp:
```



Functions

- `template<class T >`
RTformat **getFormat** ()
- `template<>`
RTformat **getFormat**< int > ()

12.18.1 Function Documentation

12.18.1.1 getFormat()

```
template<class T >
RTformat getFormat ( )
```

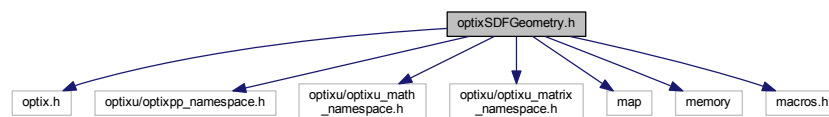
12.18.1.2 getFormat< int >()

```
template<>
RTformat getFormat< int > ( )
```

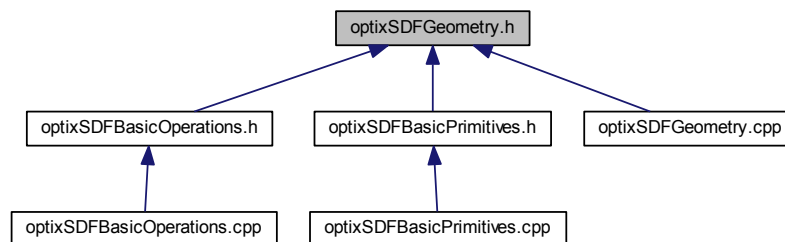
12.19 optixSDFGeometry.h File Reference

```
#include <optix.h>
#include <optixu/optixpp_namespace.h>
#include <optixu/optixu_math_namespace.h>
#include <optixu/optixu_matrix_namespace.h>
#include <map>
#include <memory>
#include "macros.h"
```

Include dependency graph for optixSDFGeometry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **optixSDFGeometry**

Abstract class for SDF Geometry. A basic class. All SDF geometry inherits it's functions.

12.20 Readme.md File Reference

12.21 renderTypes.h File Reference

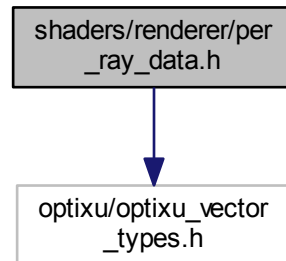
Classes

- struct **sdfParams**
- struct **visParams**

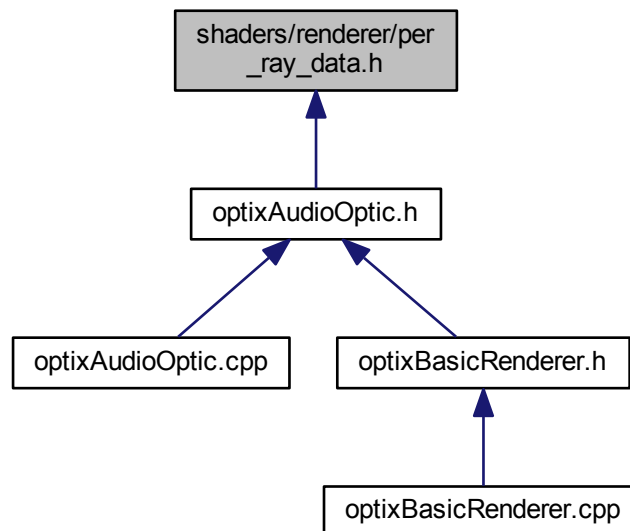
12.22 shaders/renderer/per_ray_data.h File Reference

```
#include <optixu/optixu_vector_types.h>
```

Include dependency graph for per_ray_data.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct **primDes**
- struct **PerRayData**

Macros

- `#define PER_RAY_DATA_H`

Typedefs

- `typedef optix::float2 auditoryPrim[MAX_PRIM_ALONG_RAY]`

Variables

- `const int MAX_PRIM_ALONG_RAY = 10`

12.22.1 Macro Definition Documentation

12.22.1.1 PER_RAY_DATA_H

```
#define PER_RAY_DATA_H
```

12.22.2 Typedef Documentation

12.22.2.1 auditoryPrim

```
typedef optix::float2 auditoryPrim[ MAX_PRIM_ALONG_RAY]
```

12.22.3 Variable Documentation

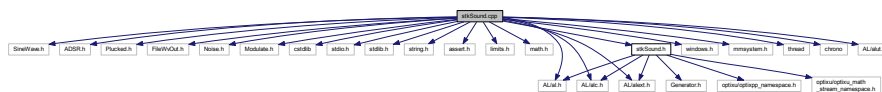
12.22.3.1 MAX_PRIM_ALONG_RAY

```
const int MAX_PRIM_ALONG_RAY = 10
```


12.23 stkSound.cpp File Reference

```
#include "SineWave.h"
#include "ADSR.h"
#include "Plucked.h"
#include "FileWvOut.h"
#include "Noise.h"
#include "Modulate.h"
#include <cstdlib>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include <limits.h>
#include <math.h>
#include "AL/al.h"
#include "AL/alc.h"
#include "AL/alext.h"
#include <windows.h>
#include <mmsystem.h>
#include <thread>
#include <chrono>
#include "AL/alut.h"
#include "stkSound.h"
```

Include dependency graph for stkSound.cpp:



Namespaces

- **stkSound**

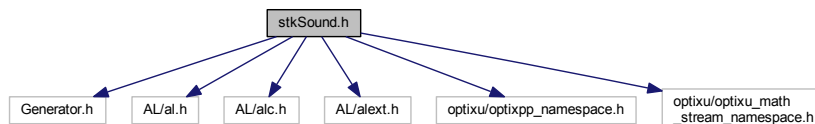
Functions

- int **stkSound::InitAL** (char ***argv, int *argc)
- void **stkSound::CloseAL** (void)
- void **stkSound::ApplySin** (ALfloat *data, ALdouble g, ALuint srte, ALuint freq)
- void **stkSound::CreateWave** (StkFloat *ptr, int srte2, int duration, ALuint &buffer)
- void **stkSound::InitEnv** ()
- void **stkSound::ExitEnv** ()
- StkFrames **stkSound::GenerateNoise** (double time, double srte2, float seed)
- void **stkSound::WaitforSound** (float scale)
- void **stkSound::PlaySelSound** (ALuint source)
- void **stkSound::MoveSounds** (std::vector< **stkSound::soundDesc** > &descSs, std::vector< float > &xpos, int scale)
- void **stkSound::RotateTest** (int scale)
- soundDesc **stkSound::PlayBuffer** (StkFloat *ptr, int time, int size, int j, optix::float2 xpos)
- void **stkSound::RotateAngle** (double angle, ALuint &source)
- void **stkSound::ApplyADSR** (stk::StkFrames &frames, int releaseCount)
- StkFrames **stkSound::GeneratePlucked** (int time, double srte2, double freq, double amplitude)

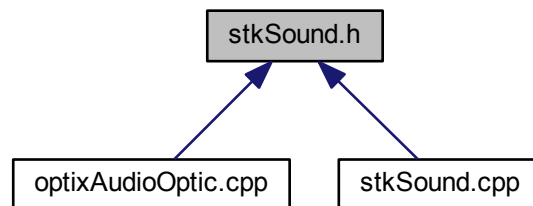
12.24 stkSound.h File Reference

```
#include "Generator.h"
#include "AL/al.h"
#include "AL/alc.h"
#include "AL/alext.h"
#include <optixu/optixpp_namespace.h>
#include <optixu/optixu_math_stream_namespace.h>
```

Include dependency graph for stkSound.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct **stkSound::soundDesc**

Namespaces

- **stkSound**

Functions

- StkFrames **stkSound::GeneratePlucked** (int time, double srate2, double freq, double amplitude)
- StkFrames **stkSound::GenerateNoise** (double time, double srate2, float seed)
- void **stkSound::ApplyADSR** (stk::StkFrames &frames, int releaseCount)
- soundDesc **stkSound::PlayBuffer** (stk::StkFloat *ptr, int time, int size, int j, optix::float2 xpos)
- void **stkSound::InitEnv** ()
- void **stkSound::ExitEnv** ()

- void **stkSound::RotateAngle** (double angle, ALuint &source)
- void **stkSound::PlaySelSound** (ALuint source)
- void **stkSound::MoveSounds** (std::vector< **stkSound::soundDesc** > &descSs, std::vector< float > &xpos, int scale)
- void **stkSound::WaitforSound** (float scale)
- void **stkSound::RotateTest** (int scale)

Index

- ~auditoryModel
 - auditoryModel, 30
- ~basicOpticalModel
 - basicOpticalModel, 35
- ~glslRayCast
 - glslRayCast, 38
- ~opticalModel
 - opticalModel, 44
- ~optixBasicActor
 - optixBasicActor, 48
- ~optixBasicRenderer
 - optixBasicRenderer, 54
- ~optixMapper
 - optixMapper, 64
- ~optixReader
 - optixReader< T >, 67
- ~optixSDFBinaryOp
 - optixSDFBinaryOp, 71
- ~optixSDFGeometry
 - optixSDFGeometry, 78
- ~optixSDFPrimitive
 - optixSDFPrimitive, 85
- ~optixSDFUnaryOp
 - optixSDFUnaryOp, 90
- ~visAuditoryMaterial
 - visAuditoryMaterial, 103
- ActivateTexture
 - glslRayCast, 38
- AddActor
 - optixBasicRenderer, 55
- AddMapper
 - optixBasicActor, 48
- AddMaterial
 - optixMapper, 64
- AddOpoperand
 - optixSDFUnaryOp, 90
- AddOpoperand1
 - optixSDFBinaryOp, 71
- AddOpoperand2
 - optixSDFBinaryOp, 72
- AdjustCenterAndBoundingBox
 - optixSDFBinaryOp, 72
 - optixSDFUnaryOp, 91
- ApplyADSR
 - stkSound, 25
- ApplySin
 - stkSound, 25
- audioM
 - optixBasicRenderer, 61
- auditory
 - MaterialDesc, 41
- AUDITORY_RAYCASTING
 - optixBasicRenderer, 54
- auditoryModel, 29
 - ~auditoryModel, 30
 - auditoryModel, 30
 - BindBuffer, 30
 - ComputeSoundRaycast, 31
 - ConfigureHRTF, 31
 - GetBuffer, 31
 - GetHeight, 31
 - GetOutput, 31
 - GetWidth, 31
 - m_bufferOutput, 32
 - m_height, 32
 - m_width, 32
 - Render, 31
 - Reshape, 32
- auditoryPrim
 - per_ray_data.h, 122
- basicOpticalModel, 33
 - ~basicOpticalModel, 35
 - basicOpticalModel, 34
 - BindBuffer, 35
 - GetHeight, 35
 - GetOutput, 35
 - GetWidth, 35
 - InitOpenGL, 35
 - m_bufferOutput, 36
 - m_height, 36
 - m_interop, 37
 - m_width, 37
 - Render, 35
 - Reshape, 36
 - SetBufferSize, 36
 - SetDim, 36
 - UpdateOpticalBuffer, 36
- BindBuffer
 - auditoryModel, 30
 - basicOpticalModel, 35
 - opticalModel, 44
- brightness
 - visParams, 107
- buffer
 - stkSound::soundDesc, 100
- center
 - optixSDFPrimitive, 87

- CloseAL
 - stkSound, 26
- COMPUTE_SOUND
 - optixBasicRenderer, 54
- ComputeSoundRaycast
 - auditoryModel, 31
- ConfigureHRTF
 - auditoryModel, 31
- CreateGeometry
 - optixSDFBinaryOp, 72
 - optixSDFGeometry, 78
 - optixSDFPrimitive, 85
 - optixSDFUnaryOp, 91
- CreateWave
 - stkSound, 26
- cur_prim
 - PerRayData, 94
- data
 - primDes, 97
- density
 - visParams, 107
- depth
 - PerRayData, 94
- Display
 - glslRayCast, 38
 - optixBasicRenderer, 55
- Distances
 - PerRayData, 94
- doc/approach.md, 109
- doc/freptutorial.md, 109
- doc/freptutorial_test.md, 109
- doc/vtkstyle.md, 109
- dynamic
 - MaterialDesc, 41
- ExitEnv
 - stkSound, 26
- f_over_pdf
 - PerRayData, 94
- flags
 - PerRayData, 94
- fsSource
 - glslRayCast, 39
- GenerateNoise
 - stkSound, 26
- GeneratePlucked
 - stkSound, 26
- geoDesc1
 - optixSDFBinaryOp, 74
- geoDesc2
 - optixSDFBinaryOp, 74
- GetAcceleration
 - optixBasicActor, 48
- GetActor
 - optixBasicRenderer, 55
- GetAnyHitProgName
 - visAuditoryMaterial, 103
- GetAudioDim
 - optixBasicRenderer, 55
- GetAuditoryExceptionProgName
 - optixBasicRenderer, 55
- GetAuditoryMissProgName
 - optixBasicRenderer, 55
- GetAuditoryRayGenerationProgName
 - optixBasicRenderer, 55
- GetBoundingBoxProgName
 - optixSDFGeometry, 78
- GetBuffer
 - auditoryModel, 31
- GetCallableProg
 - optixSDFGeometry, 78
- GetCallableProgName
 - optixSDFGeometry, 78
- GetCenter
 - optixSDFPrimitive, 85
- GetClosestHitProgName
 - visAuditoryMaterial, 103
- GetDynamicProgName
 - visAuditoryMaterial, 103
- GetExceptionProgName
 - optixBasicRenderer, 56
- getFormat
 - optixSDFGeometry.cpp, 119
- getFormat< int >
 - optixSDFGeometry.cpp, 119
- GetHeight
 - auditoryModel, 31
 - basicOpticalModel, 35
- GetInput
 - optixSDFUnaryOp, 91
- GetInput1
 - optixSDFBinaryOp, 72
- GetInput2
 - optixSDFBinaryOp, 72
- GetIntersectionProgName
 - optixSDFGeometry, 78
- GetMaterial
 - optixMapper, 64
- GetMaterialDesc
 - optixMapper, 64
- GetMaterialType
 - optixSDFGeometry, 79
- GetMissProgName
 - optixBasicRenderer, 56
- GetMode
 - optixBasicRenderer, 56
- GetNumOfActors
 - optixBasicRenderer, 56
- GetOutput
 - auditoryModel, 31
 - basicOpticalModel, 35
 - optixBasicActor, 48
 - optixMapper, 65
 - optixReader< T >, 67

- optixSDFBinaryOp, 72
 - optixSDFGeometry, 79
 - optixSDFPrimitive, 86
 - optixSDFUnaryOp, 91
 - visAuditoryMaterial, 104
- GetOutputDesc
 - optixSDFGeometry, 79
- GetOutputSdfObject
 - optixSDFUnaryOp, 91
- GetOutputSoundBuffer
 - optixBasicRenderer, 56
- GetProgName
 - visAuditoryMaterial, 104
- GetRadius
 - optixSDFPrimitive, 86
- GetRayGenerationProgName
 - optixBasicRenderer, 56
- GetTime
 - optixBasicRenderer, 56
 - optixSDFGeometry, 79
- GetType
 - visAuditoryMaterial, 104
- GetWidth
 - auditoryModel, 31
 - basicOpticalModel, 35
- gg
 - optixBasicActor, 50
- gsl
 - opticalModel, 45
- gslRayCast, 37
 - ~gslRayCast, 38
 - ActivateTexture, 38
 - Display, 38
 - fsSource, 39
 - gslRayCast, 38
 - initGLSL, 39
 - initTexture, 39
 - m_gslIFS, 39
 - m_gslProgram, 39
 - m_gslVS, 39
 - m_hdrTexture, 39
 - vsSource, 40
- InitAcceleration
 - optixBasicRenderer, 56
- InitAL
 - stkSound, 26
- InitCallableProg
 - optixSDFBinaryOp, 72
 - optixSDFPrimitive, 86
 - optixSDFUnaryOp, 91
- InitEnv
 - stkSound, 27
- initGLSL
 - gslRayCast, 39
- Initialize
 - optixSDFBinaryOp, 73
 - optixSDFGeometry, 79
 - optixSDFPrimitive, 86
 - optixSDFUnaryOp, 91
 - visAuditoryMaterial, 104
- InitializeInputBuffer
 - optixSDFGeometry, 79
- InitializePrograms
 - optixBasicRenderer, 57
- InitOpenGL
 - basicOpticalModel, 35
 - opticalModel, 44
- InitProg
 - optixBasicRenderer, 57
 - optixSDFGeometry, 79
 - visAuditoryMaterial, 104
- InitRenderer
 - optixBasicRenderer, 57
- initTexture
 - gslRayCast, 39
- INTERACTIVE_CAMERA
 - optixBasicRenderer, 54
- isAnyHit
 - visAuditoryMaterial, 104
- isAuditory
 - optixBasicRenderer, 57
 - visAuditoryMaterial, 104
- isClosestHit
 - visAuditoryMaterial, 105
- isDynamic
 - optixBasicRenderer, 57
 - optixSDFGeometry, 80
 - PerRayData, 95
 - visAuditoryMaterial, 105
- isSoundRay
 - PerRayData, 95
- LaunchAuditoryContext
 - optixBasicRenderer, 57
- LaunchOpticContext
 - optixBasicRenderer, 57
- LOAD_PROC
 - optixAudioOptic.cpp, 111
- lvShift
 - sdfParams, 99
- m_bufferOutput
 - auditoryModel, 32
 - basicOpticalModel, 36
- m_builder
 - optixBasicActor, 50
- m_gslIFS
 - gslRayCast, 39
- m_gslProgram
 - gslRayCast, 39
- m_gslVS
 - gslRayCast, 39
- m_hdrTexture
 - gslRayCast, 39
- m_height
 - auditoryModel, 32
 - basicOpticalModel, 36

- m_interop
 - basicOpticalModel, 37
- m_mapOfPrograms
 - optixBasicRenderer, 61
- m_pboOutputBuffer
 - opticalModel, 45
- m_sdf
 - optixSDFUnaryOp, 93
- m_sdf1
 - optixSDFBinaryOp, 74
- m_sdf2
 - optixSDFBinaryOp, 74
- m_width
 - auditoryModel, 32
 - basicOpticalModel, 37
- MaterialDesc, 40
 - auditory, 41
 - dynamic, 41
 - MaterialDesc, 40
- MAX_PRIM_ALONG_RAY
 - per_ray_data.h, 122
- Modified
 - opticalModel, 44
 - optixMapper, 65
 - optixReader< T >, 68
 - optixSDFGeometry, 80
- MoveSounds
 - stkSound, 27
- numS
 - PerRayData, 95
- OPTIC_RAYCASTING
 - optixBasicRenderer, 54
- opticalModel, 41
 - ~opticalModel, 44
 - BindBuffer, 44
 - glsl, 45
 - InitOpenGL, 44
 - m_pboOutputBuffer, 45
 - Modified, 44
 - opticalModel, 44
 - Render, 44
 - Reshape, 45
 - UpdateOpticalBuffer, 45
- opticM
 - optixBasicRenderer, 61
- optixAbstractMaterial.cpp, 109
- optixAbstractMaterial.h, 109
- optixAudioOptic.cpp, 110
 - LOAD_PROC, 111
- optixAudioOptic.h, 111
 - optixAudioOptic_h, 112
- optixAudioOptic_h
 - optixAudioOptic.h, 112
- optixBasicActor, 46
 - ~optixBasicActor, 48
 - AddMapper, 48
 - GetAcceleration, 48
 - GetOutput, 48
 - gg, 50
 - m_builder, 50
 - optixBasicActor, 48
 - PrintInfo, 49
 - RebuildAccel, 49
 - SetAccelerationProperties, 49
 - SetAccelerationType, 49
 - SetAuditoryModel, 49
 - SetContext, 49
 - SetOpticalModel, 49
 - SetTime, 50
 - Update, 50
- optixBasicActor.cpp, 113
- optixBasicActor.h, 113
- optixBasicRenderer, 51
 - ~optixBasicRenderer, 54
 - AddActor, 55
 - audioM, 61
 - AUDITORY_RAYCASTING, 54
 - COMPUTE_SOUND, 54
 - Display, 55
 - GetActor, 55
 - GetAudioDim, 55
 - GetAuditoryExceptionProgName, 55
 - GetAuditoryMissProgName, 55
 - GetAuditoryRayGenerationProgName, 55
 - GetExceptionProgName, 56
 - GetMissProgName, 56
 - GetMode, 56
 - GetNumOfActors, 56
 - GetOutputSoundBuffer, 56
 - GetRayGenerationProgName, 56
 - GetTime, 56
 - InitAcceleration, 56
 - InitializePrograms, 57
 - InitProg, 57
 - InitRenderer, 57
 - INTERACTIVE_CAMERA, 54
 - isAuditory, 57
 - isDynamic, 57
 - LaunchAuditoryContext, 57
 - LaunchOpticContext, 57
 - m_mapOfPrograms, 61
 - OPTIC_RAYCASTING, 54
 - opticM, 61
 - optixBasicRenderer, 54
 - PLAY_ANIMATION, 54
 - PlayAnimation, 58
 - Render, 58
 - RenderModes, 54
 - Reshape, 58
 - setAudioBuffer, 58
 - SetAuditory, 58
 - SetAuditoryExceptionProg, 58
 - SetAuditoryExceptionProgName, 58
 - SetAuditoryMissProg, 59
 - SetAuditoryMissProgName, 59

- SetAuditoryRayGenerationProg, 59
- SetAuditoryRayGenerationProgName, 59
- SetDynamic, 59
- SetExceptionProg, 59
- SetExceptionProgName, 59
- SetMissProg, 60
- SetMissProgName, 60
- SetMode, 60
- SetOpticalDims, 60
- SetPrograms, 60
- SetRayGenerationProg, 60
- SetRayGenerationProgName, 60
- SetTime, 61
- SetUpRenderer, 61
- Update, 61
- optixBasicRenderer.cpp, 114
- optixBasicRenderer.h, 115
- optixMapper, 62
 - ~optixMapper, 64
 - AddMaterial, 64
 - GetMaterial, 64
 - GetMaterialDesc, 64
 - GetOutput, 65
 - Modified, 65
 - optixMapper, 64
 - PrintInfo, 65
 - SetAuditoryModel, 65
 - SetDescInput, 65
 - SetInput, 65
 - SetMaterialParameters, 65
 - SetOpticalModel, 66
 - SetTime, 66
 - Update, 66
- optixReader
 - optixReader< T >, 67
- optixReader< T >, 66
 - ~optixReader, 67
 - GetOutput, 67
 - Modified, 68
 - optixReader, 67
 - opxGetMacro, 68
 - opxSetMacro, 68
 - ReadFile, 68
 - Update, 68
- optixReader.h, 115
- optixSDFBasicOperations.cpp, 116
- optixSDFBasicOperations.h, 116
- optixSDFBasicPrimitives.cpp, 117
- optixSDFBasicPrimitives.h, 118
- optixSDFBinaryOp, 69
 - ~optixSDFBinaryOp, 71
 - AddOperand1, 71
 - AddOperand2, 72
 - AdjustCenterAndBoundingBox, 72
 - CreateGeometry, 72
 - geoDesc1, 74
 - geoDesc2, 74
 - GetInput1, 72
 - GetInput2, 72
 - GetOutput, 72
 - InitCallableProg, 72
 - Initialize, 73
 - m_sdf1, 74
 - m_sdf2, 74
 - optixSDFBinaryOp, 71
 - SetCallableProg, 73
 - SetContext, 73
 - SetInput1, 73
 - SetInput2, 73
 - SetMainPrograms, 73
 - Update, 74
- optixSDFGeometry, 75
 - ~optixSDFGeometry, 78
 - CreateGeometry, 78
 - GetBoundingBoxProgName, 78
 - GetCallableProg, 78
 - GetCallableProgName, 78
 - GetIntersectionProgName, 78
 - GetMaterialType, 79
 - GetOutput, 79
 - GetOutputDesc, 79
 - GetTime, 79
 - Initialize, 79
 - InitializeInputBuffer, 79
 - InitProg, 79
 - isDynamic, 80
 - Modified, 80
 - optixSDFGeometry, 78
 - SetBoundingBoxProg, 80
 - SetBoundingBoxProgName, 80
 - SetCallableProg, 80
 - SetCallableProgManually, 80
 - SetCallableProgName, 80
 - SetContext, 81
 - SetDynamic, 81
 - SetIntersectionProg, 81
 - SetIntersectionProgName, 81
 - SetMainPrograms, 81
 - SetMaterialType, 81
 - SetParameters, 82
 - SetTime, 82
 - Update, 82
- optixSDFGeometry.cpp, 119
 - getFormat, 119
 - getFormat< int >, 119
- optixSDFGeometry.h, 120
- optixSDFPrimitive, 82
 - ~optixSDFPrimitive, 85
 - center, 87
 - CreateGeometry, 85
 - GetCenter, 85
 - GetOutput, 86
 - GetRadius, 86
 - InitCallableProg, 86
 - Initialize, 86
 - optixSDFPrimitive, 85

- radius, 87
- SetCallableProg, 86
- SetCenter, 86
- SetRadius, 87
- optixSDFUnaryOp, 87
 - ~optixSDFUnaryOp, 90
 - AddOpperand, 90
 - AdjustCenterAndBoundingBox, 91
 - CreateGeometry, 91
 - GetInput, 91
 - GetOutput, 91
 - GetOutputSdfObject, 91
 - InitCallableProg, 91
 - Initialize, 91
 - m_sdf, 93
 - optixSDFUnaryOp, 90
 - SetCallableProg, 92
 - SetContext, 92
 - SetInput, 92
 - SetMainPrograms, 92
 - Update, 92
- opxGetMacro
 - optixReader< T >, 68
- opxSetMacro
 - optixReader< T >, 68
- pdf
 - PerRayData, 95
- per_ray_data.h
 - auditoryPrim, 122
 - MAX_PRIM_ALONG_RAY, 122
 - PER_RAY_DATA_H, 122
- PER_RAY_DATA_H
 - per_ray_data.h, 122
- PerRayData, 93
 - cur_prim, 94
 - depth, 94
 - Distances, 94
 - f_over_pdf, 94
 - flags, 94
 - isDynamic, 95
 - isSoundRay, 95
 - numS, 95
 - pdf, 95
 - pos, 95
 - primitives, 95
 - radiance, 95
 - result, 95
 - rnd, 96
 - seed, 96
 - TimeSound, 96
 - wi, 96
 - wo, 96
- PLAY_ANIMATION
 - optixBasicRenderer, 54
- PlayAnimation
 - optixBasicRenderer, 58
- PlayBuffer
 - stkSound, 27
- PlaySelSound
 - stkSound, 27
- pos
 - PerRayData, 95
- primDes, 97
 - data, 97
 - primDes, 97
 - x, 98
 - y, 98
- primitives
 - PerRayData, 95
- PrintInfo
 - optixBasicActor, 49
 - optixMapper, 65
- radiance
 - PerRayData, 95
- radius
 - optixSDFPrimitive, 87
- ReadFile
 - optixReader< T >, 68
- Readme.md, 120
- RebuildAccel
 - optixBasicActor, 49
- Render
 - auditoryModel, 31
 - basicOpticalModel, 35
 - opticalModel, 44
 - optixBasicRenderer, 58
- RenderModes
 - optixBasicRenderer, 54
- renderTypes.h, 120
- Reshape
 - auditoryModel, 32
 - basicOpticalModel, 36
 - opticalModel, 45
 - optixBasicRenderer, 58
- result
 - PerRayData, 95
- rnd
 - PerRayData, 96
- RotateAngle
 - stkSound, 27
- RotateTest
 - stkSound, 28
- sdfParams, 98
 - lvShift, 99
 - sdfParams, 99
 - texSize, 99
- seed
 - PerRayData, 96
- SetAccelerationProperties
 - optixBasicActor, 49
- SetAccelerationType
 - optixBasicActor, 49
- SetAnyHit
 - visAuditoryMaterial, 105
- SetAnyHitProg

- visAuditoryMaterial, 105
- SetAnyHitProgName
 - visAuditoryMaterial, 105
- setAudioBuffer
 - optixBasicRenderer, 58
- SetAuditory
 - optixBasicRenderer, 58
- SetAuditoryExceptionProg
 - optixBasicRenderer, 58
- SetAuditoryExceptionProgName
 - optixBasicRenderer, 58
- SetAuditoryMissProg
 - optixBasicRenderer, 59
- SetAuditoryMissProgName
 - optixBasicRenderer, 59
- SetAuditoryModel
 - optixBasicActor, 49
 - optixMapper, 65
- SetAuditoryRayGenerationProg
 - optixBasicRenderer, 59
- SetAuditoryRayGenerationProgName
 - optixBasicRenderer, 59
- SetAuditoryTypeOff
 - visAuditoryMaterial, 105
- SetAuditoryTypeOn
 - visAuditoryMaterial, 105
- SetBoundingBoxProg
 - optixSDFGeometry, 80
- SetBoundingBoxProgName
 - optixSDFGeometry, 80
- SetBufferSize
 - basicOpticalModel, 36
- SetCallableProg
 - optixSDFBinaryOp, 73
 - optixSDFGeometry, 80
 - optixSDFPrimitive, 86
 - optixSDFUnaryOp, 92
- SetCallableProgManually
 - optixSDFGeometry, 80
- SetCallableProgName
 - optixSDFGeometry, 80
- SetCenter
 - optixSDFPrimitive, 86
- SetClosestHit
 - visAuditoryMaterial, 106
- SetClosestHitProg
 - visAuditoryMaterial, 106
- SetClosestHitProgName
 - visAuditoryMaterial, 106
- SetContext
 - optixBasicActor, 49
 - optixSDFBinaryOp, 73
 - optixSDFGeometry, 81
 - optixSDFUnaryOp, 92
- SetDescInput
 - optixMapper, 65
- SetDim
 - basicOpticalModel, 36
- SetDynamic
 - optixBasicRenderer, 59
 - optixSDFGeometry, 81
- SetDynamicProg
 - visAuditoryMaterial, 106
- SetDynamicProgName
 - visAuditoryMaterial, 106
- SetExceptionProg
 - optixBasicRenderer, 59
- SetExceptionProgName
 - optixBasicRenderer, 59
- SetInput
 - optixMapper, 65
 - optixSDFUnaryOp, 92
- SetInput1
 - optixSDFBinaryOp, 73
- SetInput2
 - optixSDFBinaryOp, 73
- SetIntersectionProg
 - optixSDFGeometry, 81
- SetIntersectionProgName
 - optixSDFGeometry, 81
- SetMainPrograms
 - optixSDFBinaryOp, 73
 - optixSDFGeometry, 81
 - optixSDFUnaryOp, 92
- SetMaterialParameters
 - optixMapper, 65
 - visAuditoryMaterial, 106
- SetMaterialType
 - optixSDFGeometry, 81
- SetMissProg
 - optixBasicRenderer, 60
- SetMissProgName
 - optixBasicRenderer, 60
- SetMode
 - optixBasicRenderer, 60
- SetOpticalDims
 - optixBasicRenderer, 60
- SetOpticalModel
 - optixBasicActor, 49
 - optixMapper, 66
- SetParameters
 - optixSDFGeometry, 82
- SetPrograms
 - optixBasicRenderer, 60
- SetRadius
 - optixSDFPrimitive, 87
- SetRayGenerationProg
 - optixBasicRenderer, 60
- SetRayGenerationProgName
 - optixBasicRenderer, 60
- SetTime
 - optixBasicActor, 50
 - optixBasicRenderer, 61
 - optixMapper, 66
 - optixSDFGeometry, 82
- SetUpRenderer

- optixBasicRenderer, 61
- shaders/renderer/per_ray_data.h, 121
- soundDesc
 - stkSound::soundDesc, 100
- source
 - stkSound::soundDesc, 100
- stkSound, 25
 - ApplyADSR, 25
 - ApplySin, 25
 - CloseAL, 26
 - CreateWave, 26
 - ExitEnv, 26
 - GenerateNoise, 26
 - GeneratePlucked, 26
 - InitAL, 26
 - InitEnv, 27
 - MoveSounds, 27
 - PlayBuffer, 27
 - PlaySelSound, 27
 - RotateAngle, 27
 - RotateTest, 28
 - WaitforSound, 28
- stkSound.cpp, 123
- stkSound.h, 124
- stkSound::soundDesc, 100
 - buffer, 100
 - soundDesc, 100
 - source, 100
- texSize
 - sdfParams, 99
- time
 - visParams, 108
- TimeSound
 - PerRayData, 96
- transferOffset
 - visParams, 108
- transferScale
 - visParams, 108
- Update
 - optixBasicActor, 50
 - optixBasicRenderer, 61
 - optixMapper, 66
 - optixReader< T >, 68
 - optixSDFBinaryOp, 74
 - optixSDFGeometry, 82
 - optixSDFUnaryOp, 92
 - visAuditoryMaterial, 106
- UpdateOpticalBuffer
 - basicOpticalModel, 36
 - opticalModel, 45
- visAuditoryMaterial, 101
 - ~visAuditoryMaterial, 103
 - GetAnyHitProgName, 103
 - GetClosestHitProgName, 103
 - GetDynamicProgName, 103
 - GetOutput, 104
 - GetProgName, 104
 - GetType, 104
 - Initialize, 104
 - InitProg, 104
 - isAnyHit, 104
 - isAuditory, 104
 - isClosestHit, 105
 - isDynamic, 105
 - SetAnyHit, 105
 - SetAnyHitProg, 105
 - SetAnyHitProgName, 105
 - SetAuditoryTypeOff, 105
 - SetAuditoryTypeOn, 105
 - SetClosestHit, 106
 - SetClosestHitProg, 106
 - SetClosestHitProgName, 106
 - SetDynamicProg, 106
 - SetDynamicProgName, 106
 - SetMaterialParameters, 106
 - Update, 106
 - visAuditoryMaterial, 103
- visParams, 107
 - brightness, 107
 - density, 107
 - time, 108
 - transferOffset, 108
 - transferScale, 108
- vsSource
 - glslRayCast, 40
- WaitforSound
 - stkSound, 28
- wi
 - PerRayData, 96
- wo
 - PerRayData, 96
- x
 - primDes, 98
- y
 - primDes, 98