

Voxellancer

0.2

Generated by Doxygen 1.8.6

Sun Mar 16 2014 23:43:27

Contents

1	Main Page	1
2	build instructions	3
3	luaapi	5
4	Hierarchical Index	7
4.1	Class Hierarchy	7
5	Class Index	13
5.1	Class List	13
6	Class Documentation	19
6.1	Luaw::_indices< Is > Struct Template Reference	19
6.2	Luaw::_indices_builder< N, Is > Struct Template Reference	19
6.3	Luaw::_indices_builder< 0, Is...> Struct Template Reference	19
6.4	LuaWrapper::_pop< size_t, Ts > Struct Template Reference	19
6.5	LuaWrapper::_pop< 0, Ts...> Struct Template Reference	20
6.6	LuaWrapper::_pop< 1, T > Struct Template Reference	20
6.7	AABBEenteredPoll Class Reference	21
6.8	AbstractMove Class Reference	22
6.9	AbstractPropertyCollection Class Reference	22
6.10	AbstractShape Class Reference	23
6.11	ActionKeyMapping Class Reference	24
6.12	AiBindings Class Reference	25
6.13	AiGroupTask Class Reference	26
6.14	Aimer Class Reference	28
6.15	AimHelperBudget Class Reference	29
6.16	AimHelperBudgetVoxels Class Reference	30
6.17	AiTask Class Reference	31
6.18	AiTaskFinishedPoll Class Reference	32
6.19	ArrowBudgetVoxels Class Reference	34
6.20	B Class Reference	35

6.21	BaseLuaFunction Struct Reference	36
6.22	BaseScenario Class Reference	36
6.23	BattleScenario Class Reference	38
6.24	Bindings Class Reference	39
6.25	BlitProgram Class Reference	40
6.26	Blitter Class Reference	41
6.26.1	Detailed Description	42
6.27	BoardComputer Class Reference	43
6.28	Bullet Class Reference	44
6.29	C Class Reference	45
6.30	Camera Class Reference	46
6.31	CameraDolly Class Reference	48
6.32	CameraFollowHelper Class Reference	49
6.33	CameraHead Class Reference	50
6.34	Starfield::CameraLocation Struct Reference	50
6.35	Capsule Class Reference	51
6.36	Character Class Reference	52
6.37	CircularHudget Class Reference	53
6.38	ClusterCache Class Reference	54
6.39	ClusterLoader Class Reference	55
6.40	CockpitVoxel Class Reference	55
6.41	CollisionDetector Class Reference	56
6.42	CollisionFilter Class Reference	57
6.43	CollisionFilterIgnoringCreator Class Reference	59
6.44	ColorCoder Class Reference	60
6.45	CommandLineParser Class Reference	61
6.46	CommonBindings Class Reference	61
6.47	ContextDependant Class Reference	63
6.48	ContextProvider Class Reference	64
6.49	CrossHair Class Reference	65
6.50	CrossHairElement Class Reference	67
6.51	CrossHairVoxels Class Reference	68
6.52	CrucialVoxel Class Reference	69
6.53	D Class Reference	70
6.54	DamageForwarder Class Reference	71
6.55	DamagelImpact Class Reference	72
6.56	DamagelImpactGenerator Class Reference	73
6.57	Damager Class Reference	73
6.58	DdsTexture Class Reference	74
6.59	DefaultRenderPipeline Class Reference	75

6.60	DefendAreaTask Class Reference	76
6.61	DirectoryReader Class Reference	78
6.62	DirectSuicideTask Class Reference	79
6.63	E Class Reference	80
6.64	ElasticImpulseGenerator Class Reference	81
6.65	ElasticImpulsor Class Reference	82
6.66	Engine Class Reference	83
6.67	EngineBuilder Class Reference	85
6.68	EnginePower Class Reference	86
6.69	EngineSlot Class Reference	86
6.70	EngineSlotVoxel Class Reference	88
6.71	EngineTrailGenerator Class Reference	89
6.72	Equipment Class Reference	90
6.73	EventPoll Class Reference	91
6.74	EventPoller Class Reference	92
6.75	F Class Reference	93
6.76	Faction Class Reference	94
6.77	FactionMatrix Class Reference	95
6.78	FactionRelation Class Reference	96
6.79	FightTask Class Reference	97
6.80	FlyToTask Class Reference	99
6.81	FormationMemberTask Class Reference	100
6.82	FrameBuffer Class Reference	101
6.83	FrozenGameScenario Class Reference	102
6.84	FuelVoxel Class Reference	104
6.85	G Class Reference	105
6.86	Game Class Reference	106
6.86.1	Member Function Documentation	107
6.86.1.1	update	107
6.87	GamePlay Class Reference	107
6.87.1	Detailed Description	108
6.87.2	Member Function Documentation	109
6.87.2.1	onEntered	109
6.87.2.2	onLeft	109
6.87.2.3	update	109
6.88	GamePlayPaused Class Reference	109
6.88.1	Member Function Documentation	110
6.88.1.1	onEntered	110
6.88.1.2	onLeft	110
6.88.1.3	update	110

6.89 GamePlayRunning Class Reference	111
6.89.1 Member Function Documentation	112
6.89.1.1 onEntered	112
6.89.1.2 onLeft	112
6.89.1.3 update	112
6.90 GamePlayRunningInput Class Reference	112
6.91 GamePlayScene Class Reference	114
6.92 GamePlayScript Class Reference	115
6.93 GameScenario Class Reference	116
6.94 GameState Class Reference	117
6.94.1 Detailed Description	118
6.94.2 Member Function Documentation	118
6.94.2.1 onEntered	118
6.94.2.2 onLeft	119
6.94.2.3 update	119
6.95 GarbageCollector Class Reference	119
6.96 GenericBullet Class Reference	120
6.97 GenericEngine Class Reference	121
6.98 GenericGun Class Reference	123
6.99 GenericRocket Class Reference	124
6.100 GenericRocketLauncher Class Reference	126
6.101 GenericShip Class Reference	127
6.102 GenericWorldObject Class Reference	128
6.103 GeometryHelper Class Reference	129
6.104 God Class Reference	129
6.105 GridAABB Class Reference	130
6.106 Gun Class Reference	131
6.107 Handle< T > Class Template Reference	132
6.108 HandleImpl< T > Class Template Reference	132
6.109 Hardpoint Class Reference	133
6.110 HardpointAimHelper Class Reference	134
6.111 HardpointVoxel Class Reference	135
6.112 std::hash< glm::ivec3 > Struct Template Reference	136
6.113 std::hash< pair< A, B > > Struct Template Reference	137
6.114 HMD Class Reference	137
6.115 HMDManager Class Reference	138
6.116 HUD Class Reference	139
6.117 Hudget Class Reference	140
6.118 HUDObjectDelegate Class Reference	143
6.119 Impulse Class Reference	144

6.120ImpulseAccumulator Class Reference	145
6.121InertiaFollower Class Reference	146
6.122InputConfigurator Class Reference	146
6.123InputMapping Class Reference	147
6.124KeyTrigger Class Reference	147
6.125Letter Class Reference	149
6.126Line Class Reference	150
6.127LoopingTimer Class Reference	151
6.128LuaFunction< N, Return, Args > Class Template Reference	152
6.128.1 Member Data Documentation	153
6.128.1.1 state	153
6.129LuaWrapper Class Reference	153
6.130Math Class Reference	155
6.131Metrics Class Reference	155
6.132MonoView Class Reference	156
6.133Movement Class Reference	158
6.134Mover Class Reference	159
6.135ObjectHudget Class Reference	160
6.136ObjectHudgetCornerVoxels Class Reference	161
6.137ObjectHudgetVoxels Class Reference	162
6.138ObjectInfo Class Reference	163
6.139PatrolWaypointsTask Class Reference	164
6.140Physics Class Reference	165
6.141PirateScenario Class Reference	166
6.142Player Class Reference	167
6.143Point Class Reference	168
6.144PoliceFaction Class Reference	169
6.145PostProcessingPass Class Reference	171
6.146Projectile Class Reference	173
6.147Property< T > Class Template Reference	175
6.148PropertyCollection< T > Class Template Reference	176
6.149PropertyConverter Class Reference	176
6.150PropertyDirectory Class Reference	177
6.151PropertyManager Class Reference	178
6.152RandBool Class Reference	179
6.153RandFloat Class Reference	179
6.154RandVec3 Class Reference	179
6.155Range Class Reference	180
6.156Ray Class Reference	180
6.157RenderMetaData Class Reference	181

6.158 RenderPass Class Reference	182
6.159 RenderPipeline Class Reference	183
6.160 Rocket Class Reference	184
6.161 RocketLauncher Class Reference	186
6.162 Scene Class Reference	187
6.163 ScreenBlitter Class Reference	188
6.164 ScreenQuad Class Reference	188
6.165 Script Class Reference	190
6.166 Scriptable Class Reference	191
6.167 ScriptedScenario Class Reference	192
6.168 ScriptEngine Class Reference	194
6.169 SecondaryInputValues Struct Reference	195
6.170 Ship Class Reference	195
6.171 SimpleWayfind Class Reference	196
6.172 SingleShotTimer Class Reference	197
6.173 Size< T > Class Template Reference	198
6.174 Skybox Class Reference	199
6.175 Sound Class Reference	200
6.176 SoundManager Class Reference	201
6.177 SoundProperties Class Reference	202
6.178 SpawnRequest Class Reference	203
6.179 SpecialVoxel Class Reference	204
6.180 Sphere Class Reference	205
6.181 Split Class Reference	206
6.182 SplitData Class Reference	208
6.183 SplitDetector Class Reference	209
6.184 Splitter Class Reference	210
6.185 Squad Class Reference	211
6.186 SquadBindings Class Reference	212
6.187 SquadLogic Class Reference	214
6.188 Starfield::StarData Struct Reference	214
6.189 Starfield Class Reference	215
6.190 State Class Reference	216
6.190.1 Detailed Description	218
6.190.2 Member Function Documentation	218
6.190.2.1 onEntered	218
6.190.2.2 onLeft	218
6.190.2.3 pathToDescendant	218
6.190.2.4 transit	219
6.190.2.5 update	219

6.191 StereoBlitProgram Class Reference	219
6.192 StereoRenderInfo Class Reference	220
6.193 StereoView Class Reference	222
6.194 StereoViewEye Class Reference	223
6.195 StreamRedirect Class Reference	224
6.196 snowhouse::Stringizer< glm::vec3 > Struct Template Reference	224
6.197 snowhouse::Stringizer< TAABB< T > > Struct Template Reference	225
6.198 StyleGuide Class Reference	225
6.199 TAABB< T > Class Template Reference	226
6.200 TargetSelector Class Reference	228
6.201 TestState Class Reference	228
6.201.1 Member Function Documentation	229
6.201.1.1 onEntered	229
6.201.1.2 onLeft	229
6.202 TextFieldHudget Class Reference	230
6.203 TextFieldHudgetVoxels Class Reference	231
6.204 ThreadPool< T > Class Template Reference	232
6.205 TimedTask Class Reference	233
6.206 Timer Class Reference	234
6.207 Transform Class Reference	235
6.208 Transition Class Reference	236
6.208.1 Detailed Description	238
6.209 Trigger Class Reference	238
6.210 Triggerable Class Reference	239
6.211 TriggeredTransition Class Reference	240
6.212 glow::Uniform< T > Class Template Reference	241
6.213 View Class Reference	241
6.214 Viewer Class Reference	242
6.215 Viewport Class Reference	243
6.216 Visuals Class Reference	243
6.217 Voxel Class Reference	244
6.218 VoxelAccumulator< Accumulatable > Class Template Reference	246
6.219 VoxelCluster Class Reference	247
6.220 VoxelClusterBounds Class Reference	248
6.221 VoxelCollision Class Reference	249
6.222 VoxelCollisionAccumulator Class Reference	250
6.223 VoxelCollisionParticipant Class Reference	251
6.224 VoxelDebrisGenerator Class Reference	252
6.225 VoxelExplosionGenerator Class Reference	253
6.226 VoxelFont Class Reference	254

6.227 VoxelGridCmp< highPriorityAxis, middlePriorityAxis, lowPriorityAxis > Class Template Reference	255
6.228 SplitDetector::VoxelGroup Struct Reference	256
6.229 VoxelHangman Class Reference	256
6.230 VoxelMesh Class Reference	257
6.231 VoxelNeighbourHelper Class Reference	258
6.232 VoxelParticleData Struct Reference	258
6.233 VoxelParticleEngine Class Reference	259
6.234 VoxelParticleExpireCheck Class Reference	260
6.235 VoxelParticleFutureCheck Class Reference	261
6.236 VoxelParticleIntersectionCheck Class Reference	261
6.237 VoxelParticleRemoveCheck Class Reference	262
6.238 VoxelParticleRemover Class Reference	263
6.239 VoxelParticleRenderer Class Reference	264
6.240 VoxelParticleSetup Class Reference	265
6.241 VoxelParticleSpawnBase Class Reference	266
6.242 VoxelRenderData Class Reference	267
6.243 VoxelRenderer Class Reference	269
6.244 VoxelTree Class Reference	270
6.245 VoxelTreeNode Class Reference	271
6.246 VoxelTreeQuery Class Reference	272
6.247 Weapon Class Reference	273
6.248 WeaponBuilder Class Reference	275
6.249 World Class Reference	276
6.250 WorldLogic Class Reference	277
6.251 WorldObject Class Reference	278
6.252 WorldObjectBindings Class Reference	280
6.253 WorldObjectBuilder Class Reference	281
6.254 WorldObjectCollision Class Reference	282
6.255 WorldObjectComponents Class Reference	283
6.256 WorldObjectDestroyedPoll Class Reference	284
6.257 WorldObjectModification Class Reference	286
6.258 WorldObjectPair Struct Reference	287
6.259 WorldObjectSlot Class Reference	288
6.260 WorldTree Class Reference	289
6.261 WorldTreeGeode Class Reference	289
6.262 WorldTreeNode Class Reference	290
6.263 WorldTreeQuery Class Reference	291
6.264 WorldTreeScanner Class Reference	292
Index	294

Chapter 1

Main Page

A game about voxels in space

Have fun!

Chapter 2

build instructions

git and cmake should be on the path!

windows:

Visual Studio 2013 is required

execute:

```
git clone https://github.com/voxelinc/voxellancer.git
pushd voxellancer
git submodule init
git submodule update
pushd lib
unzip lib.zip
popd
mkdir build
pushd build
cmake -G "Visual Studio 12 Win64" ..
popd
```

now start voxellancer.sln und set the debugging working directory for voxellancer to "\$(ProjectDir)../.."

linux:

(tested on ubuntu 13.10)

execute:

```
git clone https://github.com/voxelinc/voxellancer.git
cd voxellancer
./lib/setup_libs.sh
mkdir build
cd build
cmake ..
make voxellancer
```


Chapter 3

luaapi

#Lua API bindings generated from 66949b7

aibindings.h

```
"c std::string getFaction(apikey ship); int setFaction(apikey ship, const std::string& faction); float getFactionRelation(const std::string& factionA, const std::string& factionB); int setFactionRelation(const std::string& factionA, const std::string& factionB, float friendliness); apikey onAiTaskFinished(apikey aiTask, const std::string& callback); apikey createFlyToTask(apikey ship); int setTargetPoint(apikey flyToTask, const glm::vec3& point); apikey createFightTask(apikey ship); int addFightTaskTarget(apikey flyToTask, apikey worldObject); "
```

commonbindings.h

```
"c bool isValid(apikey key); int showText(const std::string& string); int showTextFor(const std::string& string, int seconds); int setEventActive(apikey eventPoll, bool active); apikey createSingleShotTimer(const std::string& callback, float delta); apikey createLoopingTimer(const std::string& callback, float delta); apikey onAABBEntered(apikey worldObject, const glm::vec3& llf, const glm::vec3& urb, const std::string& callback); "
```

squadbindings.h

```
"c apikey createSquad(apikey leader); int joinSquad(apikey squad, apikey ship); int createPatrolWaypointsTask(apikey squad); int addPatrolWaypointPoint(apikey task, const glm::vec3& point); int createDefendAreaTask(apikey squad, const glm::vec3& point, float range); int addDefendAreaPoint(apikey task, const glm::vec3& point); "
```

worldobjectbindings.h

```
"c apikey playerShip(); apikey createShip(const std::string& name); int spawn(apikey worldObject); int setPosition(apikey worldObject, const glm::vec3& position); int setOrientation(apikey worldObject, const glm::vec3& orientation); glm::vec3 position(apikey worldObject); glm::vec3 orientation(apikey worldObject); apikey onWorldObjectDestroyed(apikey worldObject, const std::string& callback); "
```


Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

Luaw::_indices< Is >	19
Luaw::_indices_builder< N, Is >	19
Luaw::_indices_builder< 0, Is...>	19
LuaWrapper::_pop< size_t, Ts >	19
LuaWrapper::_pop< 0, Ts...>	20
LuaWrapper::_pop< 1, T >	20
A	
StyleGuide	225
AbstractMove	22
AbstractPropertyCollection	22
PropertyCollection< T >	176
AbstractShape	23
Capsule	51
Line	150
Point	168
Ray	180
Sphere	205
TAABB< T >	226
TAABB< glm::mediump_float >	226
TAABB< int >	226
ActionKeyMapping	24
Aimer	28
AimHelperHudgetVoxels	30
ArrowHudgetVoxels	34
BaseLuaFunction	36
LuaFunction< N, Return, Args >	152
BaseScenario	36
BattleScenario	38
FrozenGameScenario	102
GameScenario	116
PirateScenario	166
ScriptedScenario	192
basic_streambuf	
StreamRedirect	224
Bindings	39
AiBindings	25
CommonBindings	61

SquadBindings	212
WorldObjectBindings	280
BoardComputer	43
CameraFollowHelper	49
CameraHead	50
Starfield::CameraLocation	50
Changeable	
PropertyManager	178
Character	52
ClusterCache	54
ClusterLoader	55
CollisionDetector	56
CollisionFilter	57
CollisionFilterIgnoringCreator	59
ColorCoder	60
CommandLineParser	61
ContextDependant	63
BlitProgram	40
StereoBlitProgram	219
FrameBuffer	101
PostProcessingPass	171
Blitter	41
ScreenQuad	188
Skybox	199
Starfield	215
StereoView	222
VoxelMesh	257
VoxelParticleRenderer	264
VoxelRenderData	267
VoxelRenderer	269
ContextProvider	64
CrossHairVoxels	68
DamageForwarder	71
DamagedImpact	72
DamagedImpactGenerator	73
Damager	73
DdsTexture	74
DirectoryReader	78
ElasticImpulseGenerator	81
ElasticImpulsor	82
EngineBuilder	85
EnginePower	86
EngineTrailGenerator	89
Equipment	90
Engine	83
GenericEngine	121
Weapon	273
Gun	131
GenericGun	123
RocketLauncher	186
GenericRocketLauncher	126
EventPoller	92
Faction	94
PoliceFaction	169
FactionMatrix	95
FactionRelation	96

GamePlayRunningInput	112
GarbageCollector	119
GeometryHelper	129
God	129
GridAABB	130
Handle< T >	132
Handle< AiTask >	132
Handle< EventPoll >	132
Handle< Ship >	132
Handle< WorldObject >	132
HandleImpl< T >	132
HardpointAimHelper	134
std::hash< glm::ivec3 >	136
std::hash< pair< A, B > >	137
HMD	137
HMDManager	138
HUD	139
Hudget	140
CircularHudget	53
AimHelperHudget	29
CrossHair	65
ObjectHudget	160
TextFieldHudget	230
HUDObjectDelegate	143
Impulse	144
ImpulseAccumulator	145
InertiaFollower	146
CameraDolly	48
InputConfigurator	146
InputMapping	147
LuaWrapper	153
Math	155
Metrics	155
Movement	158
Mover	159
ObjectHudgetCornerVoxels	161
ObjectHudgetVoxels	162
ObjectInfo	163
Physics	165
Player	167
Program	
BlitProgram	40
Property< T >	175
Property< bool >	175
Property< float >	175
Property< glm::vec3 >	175
Property< InputMapping >	175
Property< int >	175
Property< uint32_t >	175
PropertyConverter	176
PropertyDirectory	177
RandBool	179
RandFloat	179
RandVec3	179
Range	180
RenderMetaData	181
RenderPass	182
PostProcessingPass	171

RenderPipeline	183
DefaultRenderPipeline	75
Starfield	215
Scene	187
GamePlayScene	114
ScreenBlitter	188
Script	190
GamePlayScript	115
Scriptable	191
AiGroupTask	26
DefendAreaTask	76
PatrolWaypointsTask	164
AiTask	31
DirectSuicideTask	79
FightTask	97
FlyToTask	99
FormationMemberTask	100
EventPoll	91
AABBEnteredPoll	21
AiTaskFinishedPoll	32
Timer	234
LoopingTimer	151
SingleShotTimer	197
WorldObjectDestroyedPoll	284
Squad	211
WorldObject	278
GenericWorldObject	128
Projectile	173
Bullet	44
GenericBullet	120
Rocket	184
GenericRocket	124
Ship	195
GenericShip	127
Split	206
ScriptEngine	194
SecondaryInputValues	195
SimpleWayfind	196
Size< T >	198
Size< int >	198
Sound	200
SoundManager	201
SoundProperties	202
SpawnRequest	203
SplitData	208
SplitDetector	209
Splitter	210
SquadLogic	214
Starfield::StarData	214
State	216
GameState	117
Game	106
GamePlay	107
GamePlayPaused	109
GamePlayRunning	111
TestState	228

B	35
StyleGuide	225
C	45
StyleGuide	225
D	70
E	80
F	93
G	105
StereoRenderInfo	220
StereoViewEye	223
snowhouse::Stringizer< glm::vec3 >	224
snowhouse::Stringizer< TAABB< T > >	225
TargetSelector	228
TextFieldHudgetVoxels	231
ThreadPool< T >	232
ThreadPool< VoxelParticleData >	232
TimedTask	233
Transform	235
Camera	46
Transition	236
TriggeredTransition	240
Trigger	238
KeyTrigger	147
Triggerable	239
TriggeredTransition	240
glow::Uniform< T >	241
glow::Uniform< float >	241
glow::Uniform< glm::mat4 >	241
View	241
MonoView	156
StereoView	222
Viewer	242
Viewport	243
Visuals	243
Voxel	244
SpecialVoxel	204
CockpitVoxel	55
CrucialVoxel	69
EngineSlotVoxel	88
FuelVoxel	104
HardpointVoxel	135
VoxelAccumulator< Accumulatable >	246
VoxelAccumulator< DamageImpact >	246
VoxelCluster	247
CrossHairElement	67
Letter	149
WorldObject	278
VoxelClusterBounds	248
VoxelCollision	249
VoxelCollisionAccumulator	250
VoxelCollisionParticipant	251
VoxelFont	254
VoxelGridCmp< highPriorityAxis, middlePriorityAxis, lowPriorityAxis >	255
SplitDetector::VoxelGroup	256
VoxelHangman	256
VoxelNeighbourHelper	258

VoxelParticleData	258
VoxelParticleEngine	259
VoxelParticleFutureCheck	261
VoxelParticleRemoveCheck	262
VoxelParticleExpireCheck	260
VoxelParticleIntersectionCheck	261
VoxelParticleRemover	263
VoxelParticleSetup	265
VoxelParticleSpawnBase	266
VoxelDebrisGenerator	252
VoxelExplosionGenerator	253
VoxelTree	270
VoxelTreeNode	271
VoxelTreeQuery	272
WeaponBuilder	275
World	276
WorldLogic	277
WorldObjectBuilder	281
WorldObjectCollision	282
WorldObjectComponents	283
WorldObjectModification	286
WorldObjectPair	287
WorldObjectSlot	288
EngineSlot	86
Hardpoint	133
WorldTree	289
WorldTreeGeode	289
WorldTreeNode	290
WorldTreeQuery	291
WorldTreeScanner	292

Chapter 5

Class Index

5.1 Class List

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

Luaw::_indices< Is >	19
Luaw::_indices_builder< N, Is >	19
Luaw::_indices_builder< 0, Is...>	19
LuaWrapper::_pop< size_t, Ts >	19
LuaWrapper::_pop< 0, Ts...>	20
LuaWrapper::_pop< 1, T >	20
AABBEnteredPoll	21
AbstractMove	22
AbstractPropertyCollection	22
AbstractShape	23
ActionKeyMapping	24
AiBindings	25
AiGroupTask	26
Aimer	28
AimHelperHudget	29
AimHelperHudgetVoxels	30
AiTask	31
AiTaskFinishedPoll	32
ArrowHudgetVoxels	34
B	35
BaseLuaFunction	36
BaseScenario	36
BattleScenario	38
Bindings	39
BlitProgram	40
Blitter	41
BoardComputer	43
Bullet	44
C	45
Camera	46
CameraDolly	48
CameraFollowHelper	49
CameraHead	50
Starfield::CameraLocation	50
Capsule	51
Character	52
CircularHudget	53
ClusterCache	54

ClusterLoader	55
CockpitVoxel	55
CollisionDetector	56
CollisionFilter	57
CollisionFilterIgnoringCreator	59
ColorCoder	60
CommandLineParser	61
CommonBindings	61
ContextDependant	63
ContextProvider	64
CrossHair	65
CrossHairElement	67
CrossHairVoxels	68
CrucialVoxel	69
D	70
DamageForwarder	71
DamagedImpact	72
DamagedImpactGenerator	73
Damager	73
DdsTexture	74
DefaultRenderPipeline	75
DefendAreaTask	76
DirectoryReader	78
DirectSuicideTask	79
E	80
ElasticImpulseGenerator	81
ElasticImpulsor	82
Engine	83
EngineBuilder	85
EnginePower	86
EngineSlot	86
EngineSlotVoxel	88
EngineTrailGenerator	89
Equipment	90
EventPoll	91
EventPoller	92
F	93
Faction	94
FactionMatrix	95
FactionRelation	96
FightTask	97
FlyToTask	99
FormationMemberTask	100
FrameBuffer	101
FrozenGameScenario	102
FuelVoxel	104
G	105
Game	106
GamePlay	107
GamePlayPaused	109
GamePlayRunning	111
GamePlayRunningInput	112
GamePlayScene	114
GamePlayScript	115
GameScenario	116
GameState	117
GarbageCollector	119
GenericBullet	120

GenericEngine	121
GenericGun	123
GenericRocket	124
GenericRocketLauncher	126
GenericShip	127
GenericWorldObject	128
GeometryHelper	129
God	129
GridAABB	130
Gun	131
Handle< T >	132
HandleImpl< T >	132
Hardpoint	133
HardpointAimHelper	134
HardpointVoxel	135
std::hash< glm::ivec3 >	136
std::hash< pair< A, B > >	137
HMD	137
HMDManager	138
HUD	139
Hudget	140
HUDObjectDelegate	143
Impulse	144
ImpulseAccumulator	145
InertiaFollower	146
InputConfigurator	146
InputMapping	147
KeyTrigger	147
Letter	149
Line	150
LoopingTimer	151
LuaFunction< N, Return, Args >	152
LuaWrapper	153
Math	155
Metrics	155
MonoView	156
Movement	158
Mover	159
ObjectHudget	160
ObjectHudgetCornerVoxels	161
ObjectHudgetVoxels	162
ObjectInfo	163
PatrolWaypointsTask	164
Physics	165
PirateScenario	166
Player	167
Point	168
PoliceFaction	169
PostProcessingPass	171
Projectile	173
Property< T >	175
PropertyCollection< T >	176
PropertyConverter	176
PropertyDirectory	177
PropertyManager	178
RandBool	179
RandFloat	179
RandVec3	179

Range	180
Ray	180
RenderMetaData	181
RenderPass	182
RenderPipeline	183
Rocket	184
RocketLauncher	186
Scene	187
ScreenBlitter	188
ScreenQuad	188
Script	190
Scriptable	191
ScriptedScenario	192
ScriptEngine	194
SecondaryInputValues	195
Ship	195
SimpleWayfind	196
SingleShotTimer	197
Size< T >	198
Skybox	199
Sound	200
SoundManager	201
SoundProperties	202
SpawnRequest	203
SpecialVoxel	204
Sphere	205
Split	206
SplitData	208
SplitDetector	209
Splitter	210
Squad	211
SquadBindings	212
SquadLogic	214
Starfield::StarData	214
Starfield	215
State	216
StereoBlitProgram	219
StereoRenderInfo	220
StereoView	222
StereoViewEye	223
StreamRedirect	224
snowhouse::Stringizer< glm::vec3 >	224
snowhouse::Stringizer< TAABB< T > >	225
StyleGuide	225
TAABB< T >	226
TargetSelector	228
TestState	228
TextFieldHudget	230
TextFieldHudgetVoxels	231
ThreadPool< T >	232
TimedTask	233
Timer	234
Transform	235
Transition	236
Trigger	238
Triggerable	239
TriggeredTransition	240
glow::Uniform< T >	241

View	241
Viewer	242
Viewport	243
Visuals	243
Voxel	244
VoxelAccumulator< Accumulatable >	246
VoxelCluster	247
VoxelClusterBounds	248
VoxelCollision	249
VoxelCollisionAccumulator	250
VoxelCollisionParticipant	251
VoxelDebrisGenerator	252
VoxelExplosionGenerator	253
VoxelFont	254
VoxelGridCmp< highPriorityAxis, middlePriorityAxis, lowPriorityAxis >	255
SplitDetector::VoxelGroup	256
VoxelHangman	256
VoxelMesh	257
VoxelNeighbourHelper	258
VoxelParticleData	258
VoxelParticleEngine	259
VoxelParticleExpireCheck	260
VoxelParticleFutureCheck	261
VoxelParticleIntersectionCheck	261
VoxelParticleRemoveCheck	262
VoxelParticleRemover	263
VoxelParticleRenderer	264
VoxelParticleSetup	265
VoxelParticleSpawnBase	266
VoxelRenderData	267
VoxelRenderer	269
VoxelTree	270
VoxelTreeNode	271
VoxelTreeQuery	272
Weapon	273
WeaponBuilder	275
World	276
WorldLogic	277
WorldObject	278
WorldObjectBindings	280
WorldObjectBuilder	281
WorldObjectCollision	282
WorldObjectComponents	283
WorldObjectDestroyedPoll	284
WorldObjectModification	286
WorldObjectPair	287
WorldObjectSlot	288
WorldTree	289
WorldTreeGeode	289
WorldTreeNode	290
WorldTreeQuery	291
WorldTreeScanner	292

Chapter 6

Class Documentation

6.1 Luaw::_indices< Is > Struct Template Reference

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

- src/scripting/elematelua/luawrapperfunction.h

6.2 Luaw::_indices_builder< N, Is > Struct Template Reference

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

- src/scripting/elematelua/luawrapperfunction.h

6.3 Luaw::_indices_builder< 0, Is...> Struct Template Reference

Public Types

- using **type** = [_indices< Is...>](#)

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

- src/scripting/elematelua/luawrapperfunction.h

6.4 LuaWrapper::_pop< size_t, Ts > Struct Template Reference

Public Types

- **typedef std::tuple< Ts...> type**

Static Public Member Functions

- template<typename T >
static std::tuple< T > **worker** (const [LuaWrapper](#) &instance, const int index)
- template<typename T1 , typename T2 , typename... Rest>
static std::tuple< T1, T2, Rest...> **worker** (const [LuaWrapper](#) &instance, const int index)
- static type **apply** ([LuaWrapper](#) &instance)

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

- src/scripting/elematelua/luawrapper.h

6.5 `LuaWrapper::pop< 0, Ts...>` Struct Template Reference

Public Types

- `typedef void type`

Static Public Member Functions

- static type `apply (LuaWrapper &)`

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

- src/scripting/elematelua/luawrapper.h

6.6 `LuaWrapper::pop< 1, T >` Struct Template Reference

Public Types

- `typedef T type`

Static Public Member Functions

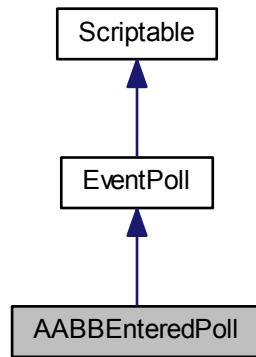
- static T `apply (LuaWrapper &instance)`

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

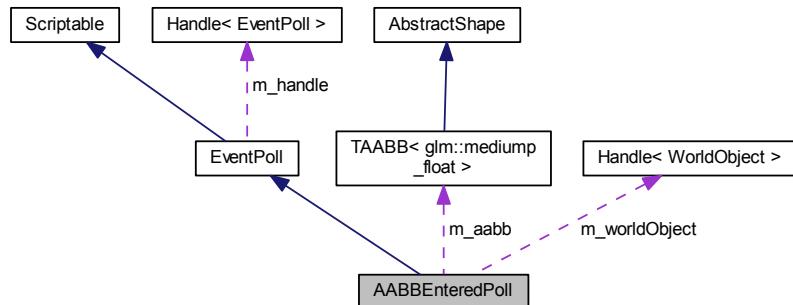
- src/scripting/elematelua/luawrapper.h

6.7 AABBEenteredPoll Class Reference

Inheritance diagram for AABBEenteredPoll:



Collaboration diagram for AABBEenteredPoll:



Public Member Functions

- **AABBEenteredPoll** (`WorldObject` *worldObject, const `AABB` &aabb, const std::function< void()> &callback)

Protected Member Functions

- virtual bool **poll** () override

Protected Attributes

- `Handle< WorldObject > m_worldObject`
- `AABB m_aabb`
- `bool m_lastEntered`

Additional Inherited Members

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

- src/events/aabbenteredpoll.h
- src/events/aabbenteredpoll.cpp

6.8 AbstractMove Class Reference

Public Member Functions

- **AbstractMove** (const glm::vec3 &directional, const glm::vec3 &angular)
- void **clear** ()
- const glm::vec3 & **directional** () const
- void **setDirectional** (const glm::vec3 &directional)
- const glm::vec3 & **angular** () const
- void **setAngular** (const glm::vec3 &angular)
- **AbstractMove** & **operator+=** (const **AbstractMove** &other)
- **AbstractMove** **operator*** (const **AbstractMove** &other) const
- **AbstractMove** **operator*** (float multiplier) const
- **AbstractMove** **operator/** (float divisor) const

Protected Attributes

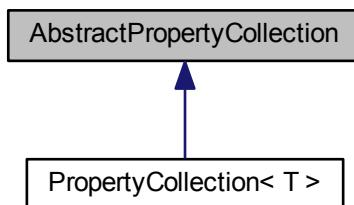
- glm::vec3 **m_directional**
- glm::vec3 **m_angular**

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

- src/geometry/abstractmove.h
- src/geometry/abstractmove.cpp

6.9 AbstractPropertyCollection Class Reference

Inheritance diagram for AbstractPropertyCollection:



Public Member Functions

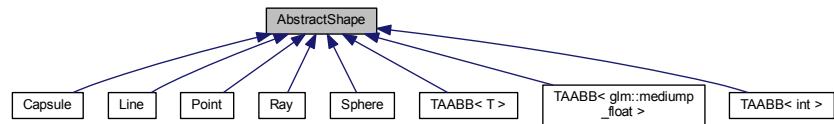
- virtual bool **update** (const std::string &key, const std::string &svalue)=0

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

- src/property/propertycollection.h

6.10 AbstractShape Class Reference

Inheritance diagram for AbstractShape:



Public Member Functions

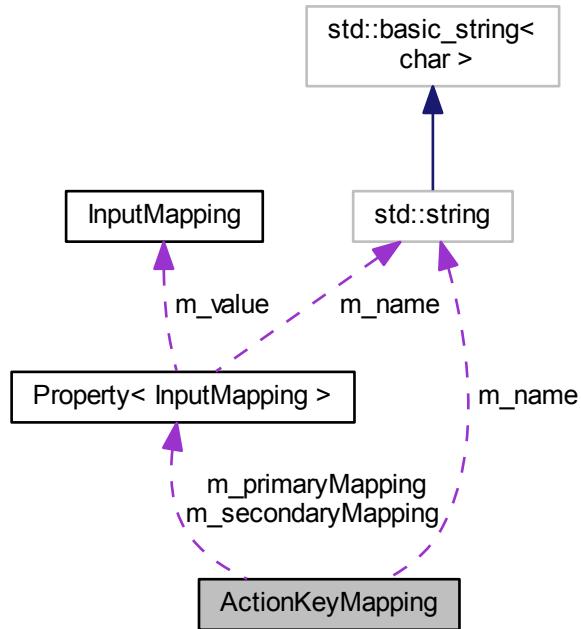
- virtual bool **intersects** (const [Sphere](#) &sphere) const =0
- virtual bool **nearTo** (const [TAABB< int >](#) &aabb) const =0
- virtual bool **containedBy** (const [TAABB< int >](#) &aabb) const =0

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

- src/geometry/abstractshape.h

6.11 ActionKeyMapping Class Reference

Collaboration diagram for ActionKeyMapping:



Public Member Functions

- **ActionKeyMapping** (`std::string primary, std::string secondary, std::string name)`
- **ActionKeyMapping** (`std::string primary, std::string secondary, std::string name, bool toggleAction)`
- [InputMapping mapping](#) (`InputClass inputClass`)
- void [setMapping](#) (`InputMapping mapping, InputClass inputClass`)
- `std::string name ()`
- `bool toggleAction ()`
- `bool toggleStatus ()`
- void [setToggleStatus](#) (`bool status`)

Protected Attributes

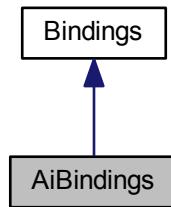
- [Property< InputMapping > m_primaryMapping](#)
- [Property< InputMapping > m_secondaryMapping](#)
- `bool m_toggleAction`
- `bool m_toggleStatus`
- `std::string m_name`

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

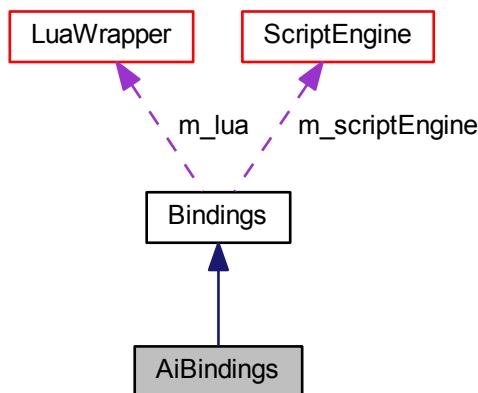
- `src/ui/actionkeymapping.h`
- `src/ui/actionkeymapping.cpp`

6.12 AiBindings Class Reference

Inheritance diagram for AiBindings:



Collaboration diagram for AiBindings:



Public Member Functions

- **AiBindings** ([GamePlayScript](#) &script)

Protected Member Functions

- virtual void **initialize** ()
- std::string **apiGetFaction** (apikey ship)
- int **apiSetFaction** (apikey ship, const std::string &faction)
- float **apiGetFactionRelation** (const std::string &factionA, const std::string &factionB)
- int **apiSetFactionRelation** (const std::string &factionA, const std::string &factionB, float friendliness)
- apikey **apiOnAiTaskFinished** (apikey aiTask, const std::string &callback)
- apikey **apiCreateFlyToTask** (apikey ship)

- int **apiSetTargetPoint** (apikey flyToTask, const glm::vec3 &point)
- apikey **apiCreateFightTask** (apikey ship)
- int **apiAddFightTaskTarget** (apikey flyToTask, apikey worldObject)

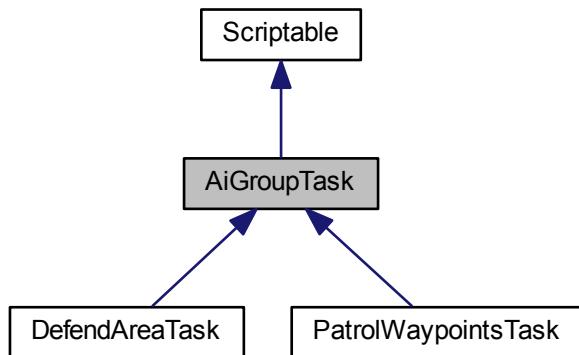
Additional Inherited Members

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

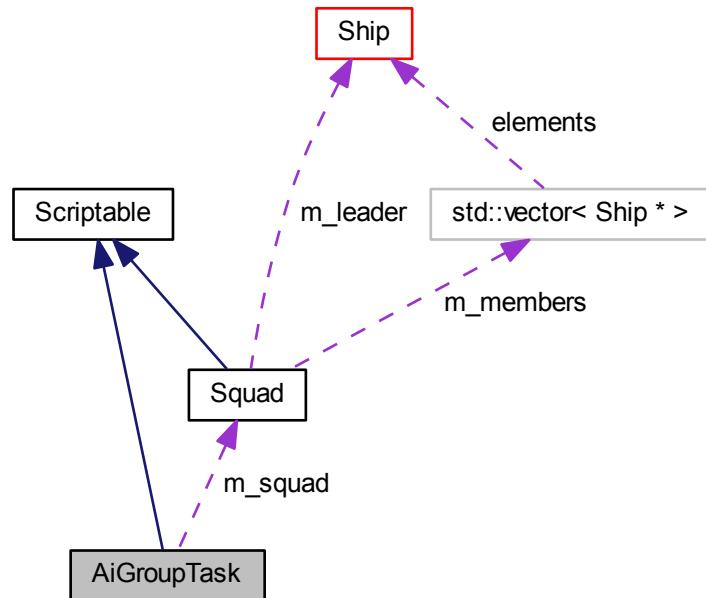
- src/scripting/bindings/aibindings.h
- src/scripting/bindings/aibindings.cpp

6.13 AiGroupTask Class Reference

Inheritance diagram for AiGroupTask:



Collaboration diagram for AiGroupTask:



Public Member Functions

- `AiGroupTask (Squad &squad)`
- `virtual void update (float deltaSec)`
- `virtual bool isInProgress ()`

Protected Member Functions

- `virtual void onMemberJoin (Ship *member)`
- `virtual void onMemberLeave (Ship *member)`
- `virtual void onNewLeader (Ship *leader)`
- `void setLeaderTask (std::shared_ptr< AiTask > task)`
- `void setMembersToFollowLeader ()`

Protected Attributes

- `Squad & m_squad`

Friends

- `class Squad`

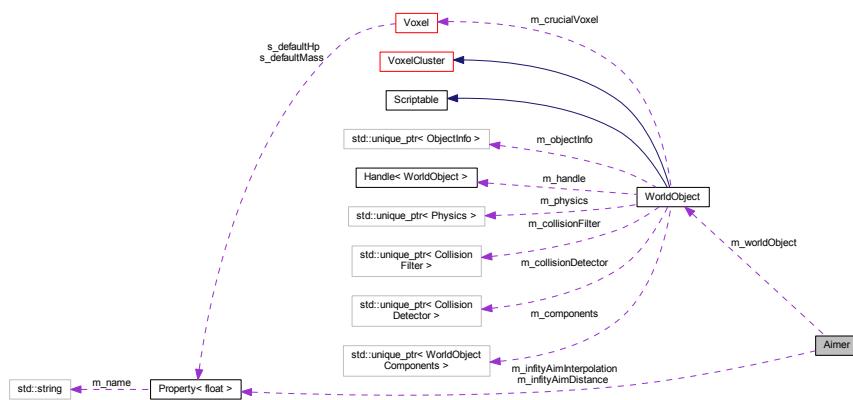
Additional Inherited Members

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

- src/ai/aigrouptask.h
- src/ai/aigrouptask.cpp

6.14 Aimer Class Reference

Collaboration diagram for Aimer:



Public Member Functions

- **Aimer** ([WorldObject](#) *worldObject)
- void **update** (float deltaSec)
- glm::vec3 **aim** (const [Ray](#) &ray)
- void **setWorldObject** ([WorldObject](#) *worldObject)

Protected Member Functions

- glm::vec3 **nearestTarget** (const std::unordered_set<[Voxel](#) *> &voxels, const glm::vec3 &origin) const
- float **distanceTo** ([Voxel](#) *voxel, const glm::vec3 &origin) const
- glm::vec3 **infinity** (const [Ray](#) &ray) const

Protected Attributes

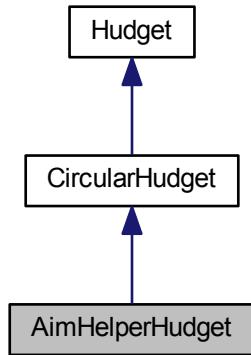
- [WorldObject](#) * **m_worldObject**
- [Property](#)< float > **m_infinityAimDistance**
- [Property](#)< float > **m_infinityAimInterpolation**
- float **m_lastDistance**

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

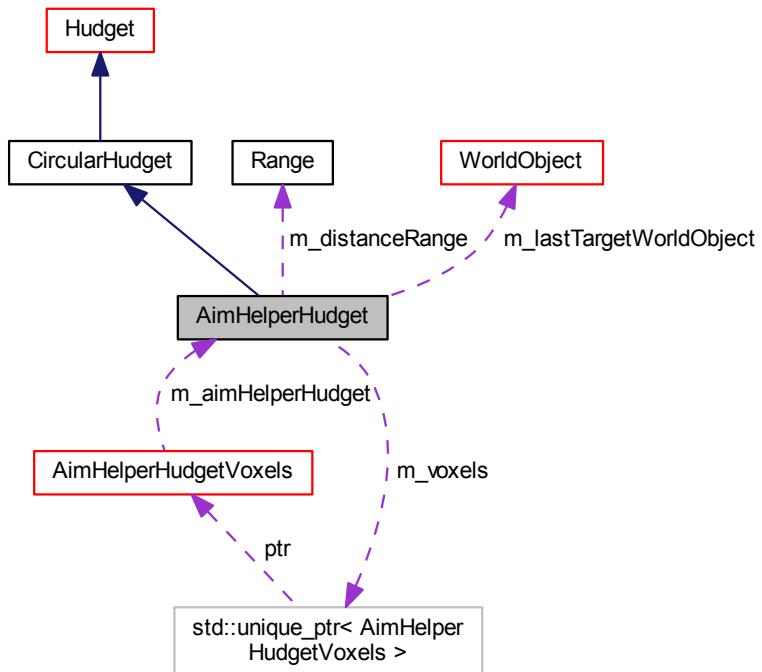
- src/utils/aimer.h
- src/utils/aimer.cpp

6.15 AimHelperHudget Class Reference

Inheritance diagram for AimHelperHudget:



Collaboration diagram for AimHelperHudget:



Public Member Functions

- **AimHelperHudget** ([HUD](#) *hud)

- const glm::vec3 & **targetPoint** () const
- virtual void **update** (float deltaSec) override
- virtual void **draw** ()
- virtual bool **isAt** (const Ray &ray) const override

Protected Member Functions

- void **calculateTargetPoint** (WorldObject *targetObject)
- void **calculatedDirection** ()

Protected Attributes

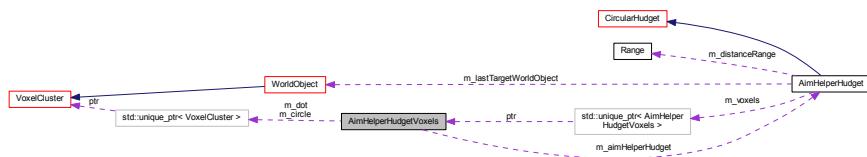
- std::unique_ptr< AimHelperHudgetVoxels > **m_voxels**
- glm::vec3 **m_targetPoint**
- glm::vec3 **m_smoothTargetPoint**
- WorldObject * **m_lastTargetWorldObject**
- bool **m_lastVisible**
- Range **m_distanceRange**

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

- src/ui/hud/aimhelperhudget.h
- src/ui/hud/aimhelperhudget.cpp

6.16 AimHelperHudgetVoxels Class Reference

Collaboration diagram for AimHelperHudgetVoxels:



Public Member Functions

- **AimHelperHudgetVoxels** (AimHelperHudget *aimHelperHudget)
- void **draw** ()

Protected Attributes

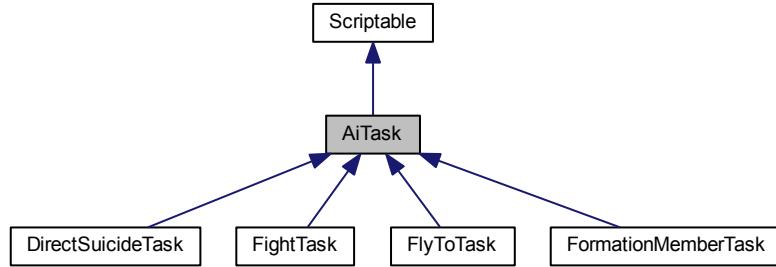
- AimHelperHudget * **m_aimHelperHudget**
- std::unique_ptr< VoxelCluster > **m_dot**
- std::unique_ptr< VoxelCluster > **m_circle**

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

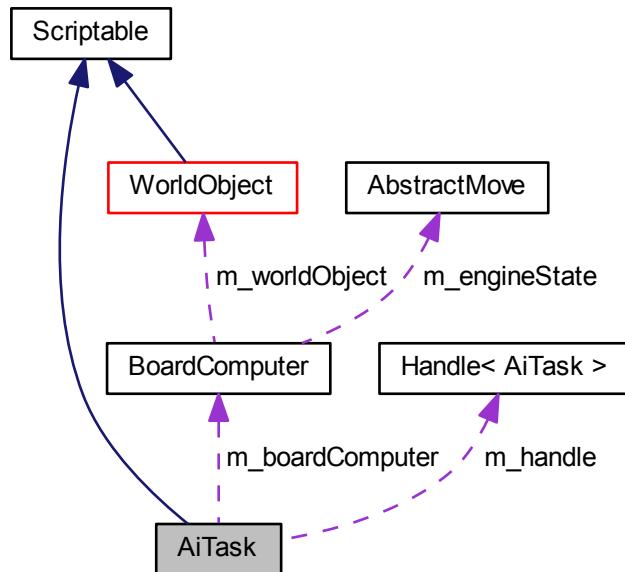
- src/ui/hud/aimhelperhudgetvoxels.h
- src/ui/hud/aimhelperhudgetvoxels.cpp

6.17 AiTask Class Reference

Inheritance diagram for AiTask:



Collaboration diagram for AiTask:



Public Member Functions

- `AiTask (BoardComputer *boardComputer)`
- `BoardComputer * boardComputer ()`
- `virtual void update (float deltaSec)`
- `virtual bool isFinished ()`
- `Handle< AiTask > & handle ()`

Protected Attributes

- Handle< AiTask > **m_handle**
- BoardComputer * **m_boardComputer**

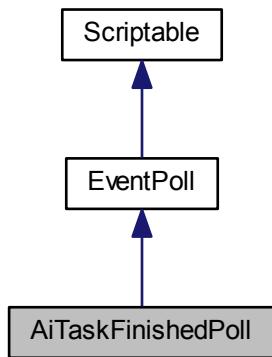
Additional Inherited Members

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

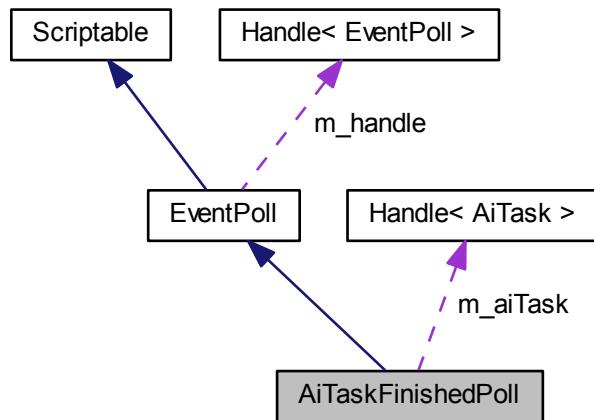
- src/ai/aitask.h
- src/ai/aitask.cpp

6.18 AiTaskFinishedPoll Class Reference

Inheritance diagram for AiTaskFinishedPoll:



Collaboration diagram for AiTaskFinishedPoll:



Public Member Functions

- `AiTaskFinishedPoll` (`AiTask *aitask, const std::function< void()> &callback)`

Protected Member Functions

- `virtual bool poll () override`
- `virtual bool isDead () override`

Protected Attributes

- `Handle< AiTask > m_aiTask`

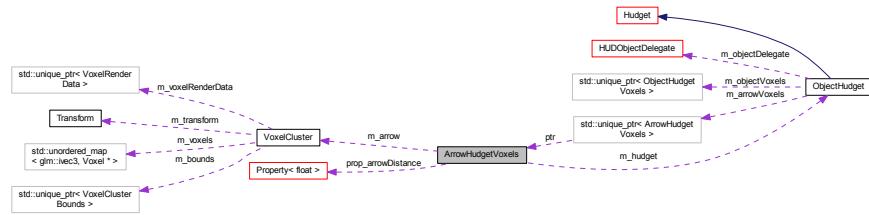
Additional Inherited Members

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

- `src/events/aitaskfinishedpoll.h`
- `src/events/aitaskfinishedpoll.cpp`

6.19 ArrowBudgetVoxels Class Reference

Collaboration diagram for ArrowBudgetVoxels:



Public Member Functions

- **ArrowBudgetVoxels** ([ObjectBudget](#) ***hudget**)
- **ObjectBudget** * **hudget** ()
- void **draw** ()
- void **setTargeted** (bool targeted)
- void **updateDirection** (glm::vec3 direction)
- bool **findPointOnEdge** ()
- virtual bool **isAt** (const [Ray](#) &ray) const

Protected Member Functions

- bool **findPoint** ()
- float **vectorAngleToPlane** (glm::vec3 vector, glm::vec3 planeNormal)
- float **vectorAngleToVector** (glm::vec3 vector, glm::vec3 vector2)

Protected Attributes

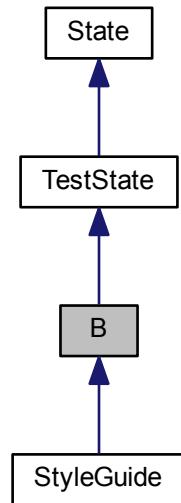
- [ObjectBudget](#) * **m_hudget**
- [VoxelCluster](#) **m_arrow**
- [Property< float >](#) **prop_arrowDistance**
- glm::vec3 **m_targetPoint**

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

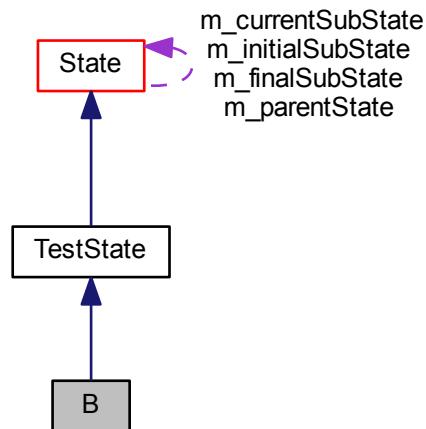
- src/ui/hud/arrowbudgetvoxels.h
- src/ui/hud/arrowbudgetvoxels.cpp

6.20 B Class Reference

Inheritance diagram for B:



Collaboration diagram for B:



Public Member Functions

- **B (State *parent)**

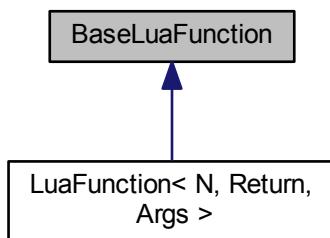
Additional Inherited Members

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

- test/statemachine/teststatemachine.cpp

6.21 BaseLuaFunction Struct Reference

Inheritance diagram for BaseLuaFunction:



Public Member Functions

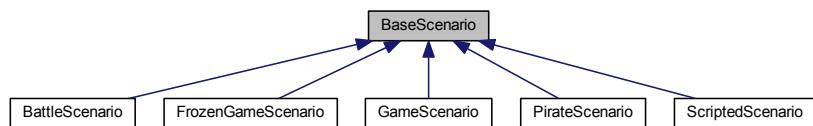
- virtual int **apply** (lua_State *state)=0

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

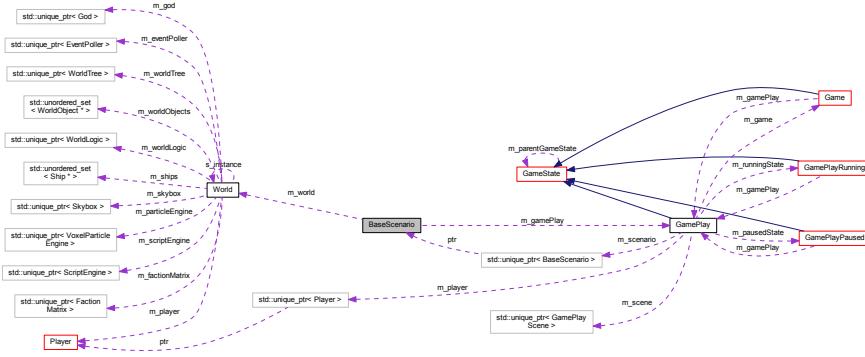
- src/scripting/elematelua/luawrapperfunction.h

6.22 BaseScenario Class Reference

Inheritance diagram for BaseScenario:



Collaboration diagram for BaseScenario:



Public Member Functions

- **BaseScenario** (`GamePlay` *`gamePlay`)
- void **load** ()
- void **clear** ()
- void **reset** ()

Protected Member Functions

- virtual void **createWorld** ()
- virtual void **populateWorld** ()

Protected Attributes

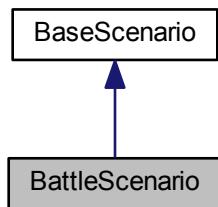
- `GamePlay` * **m_gamePlay**
- `World` * **m_world**

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

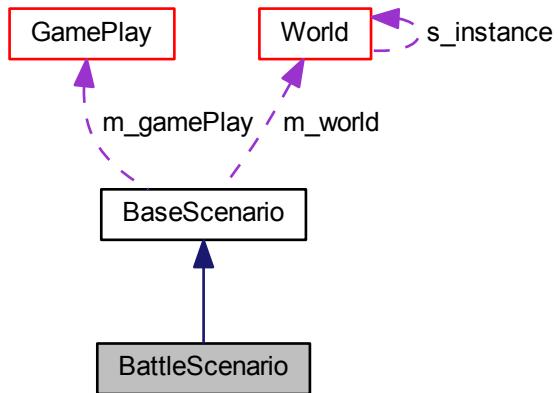
- `src/scenarios/basescenario.h`
- `src/scenarios/basescenario.cpp`

6.23 BattleScenario Class Reference

Inheritance diagram for BattleScenario:



Collaboration diagram for BattleScenario:



Public Member Functions

- **BattleScenario** (`GamePlay *gamePlay`)

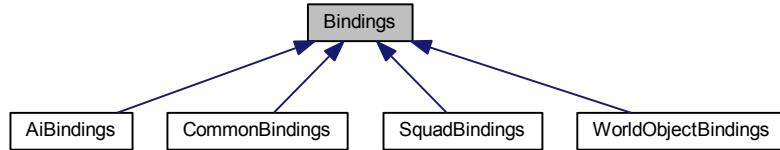
Additional Inherited Members

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

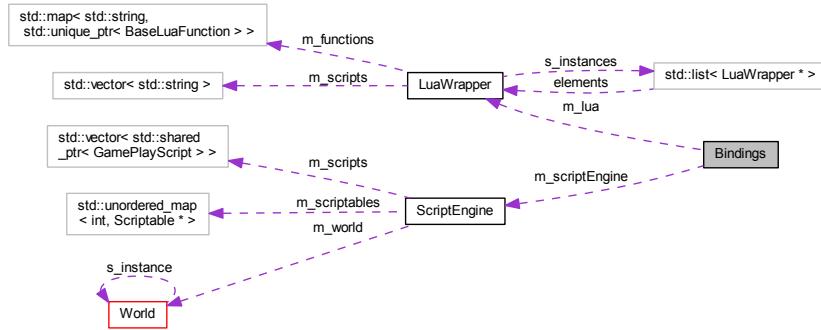
- `src/scenarios/battlescenario.h`
- `src/scenarios/battlescenario.cpp`

6.24 Bindings Class Reference

Inheritance diagram for Bindings:



Collaboration diagram for Bindings:



Public Member Functions

- **Bindings** ([GamePlayScript](#) &gamePlayScript)
- virtual void **initialize** ()=0

Protected Attributes

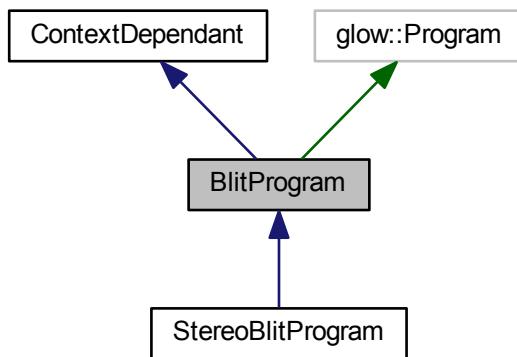
- [LuaWrapper](#) & **m_lua**
- [ScriptEngine](#) & **m_scriptEngine**

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

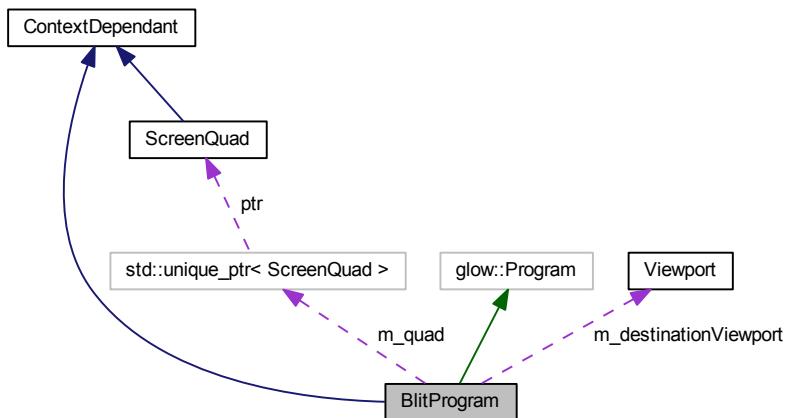
- src/scripting/bindings/bindings.h
- src/scripting/bindings/bindings.cpp

6.25 BlitProgram Class Reference

Inheritance diagram for BlitProgram:



Collaboration diagram for BlitProgram:



Public Member Functions

- `void setSource (glow::Texture *source)`
- `void setDestination (glow::FrameBufferObject *targetFBO, const Viewport &m_targetViewpoer)`
- `virtual void blit ()`

Public Attributes

- `const GLint TEXTURE_LOCATION = 0`

Protected Member Functions

- virtual void **initialize** ()
- virtual void **initializeShaders** ()=0
- template<typename T >
void **setUniform** (const std::string &name, const T &value)
- virtual void **beforeContextDestroy** () override
- virtual void **afterContextRebuild** () override

Protected Attributes

- glow::Texture * **m_source**
- glow::FrameBufferObject * **m_destinationFBO**
- std::unique_ptr< [ScreenQuad](#) > **m_quad**
- [Viewport](#) **m_destinationViewport**
- bool **m_initialized**

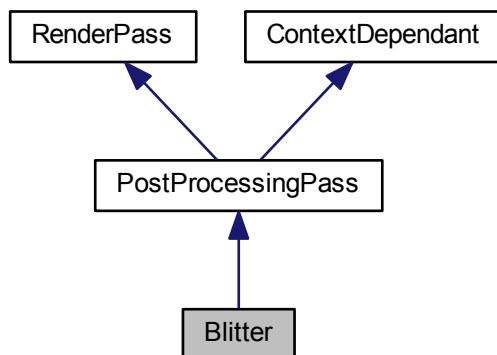
The documentation for this class was generated from the following files:

- src/programs/blitprogram.h
- src/programs/blitprogram.cpp
- src/programs/blitprogram.inl

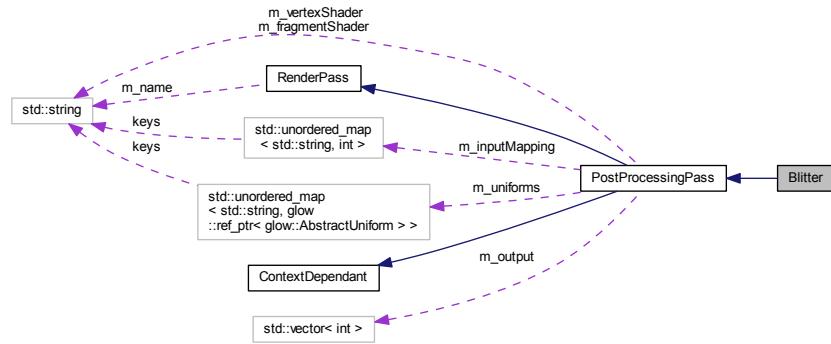
6.26 Blitter Class Reference

```
#include <blitter.h>
```

Inheritance diagram for Blitter:



Collaboration diagram for Blitter:



Public Member Functions

- virtual void **apply** (FrameBuffer &frameBuffer, glow::FrameBufferObject *target)

Additional Inherited Members

6.26.1 Detailed Description

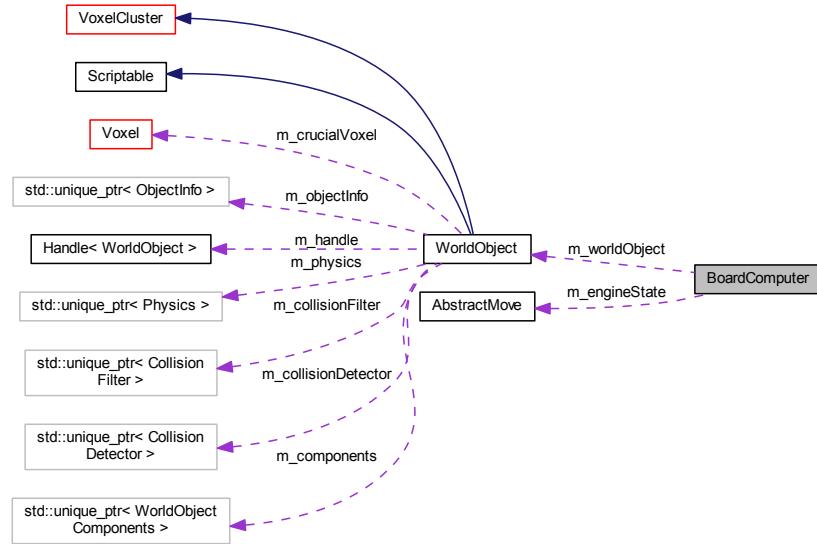
the mono/stereo blitter does some additional stuff that is not useful for just copying one framebuffer to another. Please tell me if there is a better way.

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

- src/display/rendering/blitter.h
 - src/display/rendering/blitter.cpp

6.27 BoardComputer Class Reference

Collaboration diagram for BoardComputer:



Public Member Functions

- `BoardComputer (WorldObject *worldObject)`
- `WorldObject * worldObject ()`
- `const EngineState & engineState () const`
- `void moveTo (const glm::vec3 &position, bool decelerate=true)`
- `void rotateTo (const glm::vec3 &position, const glm::vec3 &up=glm::vec3(0, 0, 0))`
- `void shootBullet (const std::vector< Handle< WorldObject >> &targets)`
- `void shootRockets (Handle< WorldObject > &target)`
- `void update (float deltaSec)`

Protected Member Functions

- `glm::vec3 rotateUpTo (const glm::vec3 &up)`
- `glm::vec3 rotateUpAuto (const glm::quat &rotation)`

Protected Attributes

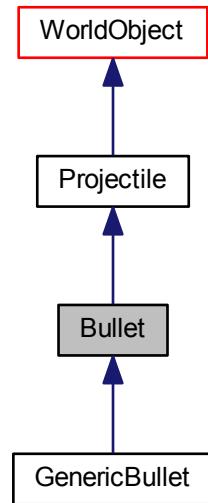
- `WorldObject * m_worldObject`
- `EngineState m_engineState`
- `bool m_overwriteEngineState`

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

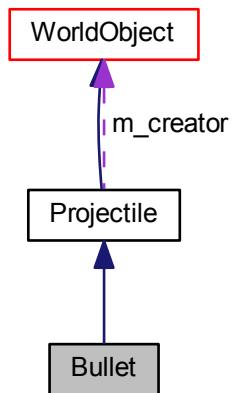
- `src/ai/boardcomputer.h`
- `src/ai/boardcomputer.cpp`

6.28 Bullet Class Reference

Inheritance diagram for Bullet:



Collaboration diagram for Bullet:



Public Member Functions

- virtual WorldObjectType **objectType () const override**
- virtual void **update (float deltaSec) override**

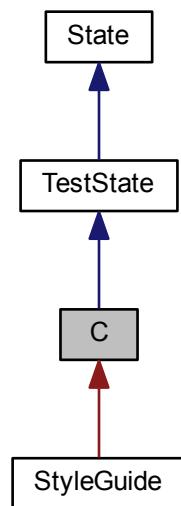
Additional Inherited Members

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

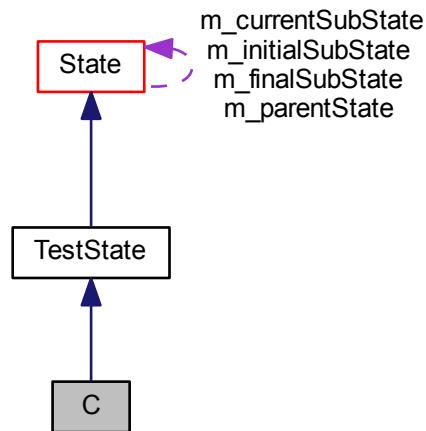
- src/equipment/weapons/bullet.h
- src/equipment/weapons/bullet.cpp

6.29 C Class Reference

Inheritance diagram for C:



Collaboration diagram for C:



Public Member Functions

- **C (State *parent)**

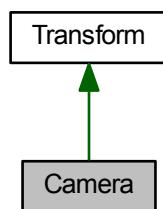
Additional Inherited Members

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

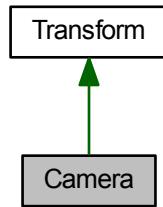
- test/statemachine/teststatemachine.cpp

6.30 Camera Class Reference

Inheritance diagram for Camera:



Collaboration diagram for Camera:



Public Member Functions

- **Camera** (int viewportWidth, int viewportHeight)
- void **move** (glm::vec3 dist)
- void **setPosition** (glm::vec3 pos)
- void **rotateX** (float rot)
- void **rotateY** (float rot)
- void **rotateZ** (float rot)
- void **setOrientation** (glm::quat quat)
- const glm::mat4 & **view** () const
- const glm::mat4 & **viewInverted** () const
- const glm::quat & **orientation** () const
- const glm::vec3 & **position** () const
- float **zNear** () const
- void **setZNear** (float zNear)
- float **zFar** () const
- void **setZFar** (float zFar)
- float **fovy** () const
- void **setFovy** (float fovy)
- const glm::ivec2 **viewport** () const
- void **setViewport** (const glm::ivec2 &viewport)
- const glm::vec3 & **projectionOffset** () const
- void **setProjectionOffset** (const glm::vec3 &projectionOffset)
- float **aspectRatio** () const
- const glm::mat4 & **projection** () const
- const glm::mat4 & **viewProjection** () const

Protected Member Functions

- void **viewDirty** ()
- void **projectionDirty** ()

Protected Attributes

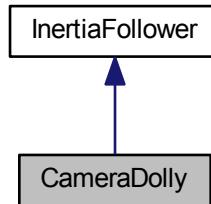
- float **m_fovy**
- float **m_aspect**
- float **m_zNear**
- float **m_zFar**
- glm::ivec2 **m_viewport**
- glm::vec3 **m_projectionOffset**
- glm::mat4 **m_view**
- glm::mat4 **m_projection**
- glm::mat4 **m_viewProjection**

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

- src/camera/camera.h
- src/camera/camera.cpp

6.31 CameraDolly Class Reference

Inheritance diagram for CameraDolly:



Collaboration diagram for CameraDolly:



Public Member Functions

- **CameraHead & cameraHead ()**
- const **CameraHead & cameraHead () const**
- void **followWorldObject (WorldObject *m_followWorldObject)**
- void **update (float deltaSec)**

Protected Attributes

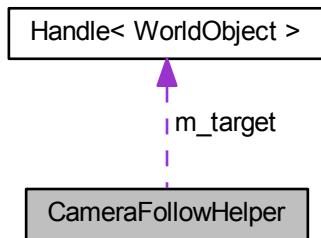
- std::unique_ptr< CameraHead > **m_cameraHead**
- std::unique_ptr< CameraFollowHelper > **m_followHelper**

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

- src/camera/cameradolly.h
- src/camera/cameradolly.cpp

6.32 CameraFollowHelper Class Reference

Collaboration diagram for CameraFollowHelper:



Public Member Functions

- `WorldObject * target ()`
- `void setTarget (WorldObject *target)`
- `glm::vec3 followPosition ()`

Protected Attributes

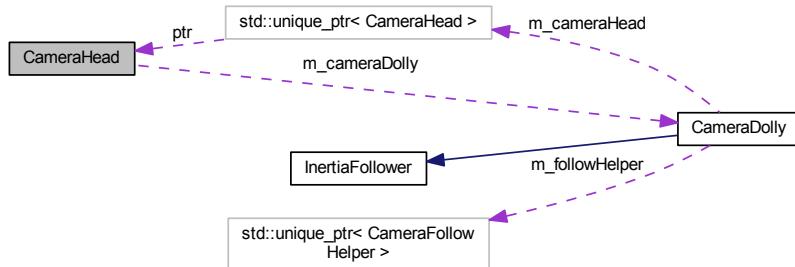
- `Handle< WorldObject > m_target`

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

- src/camera/camerafollowhelper.h
- src/camera/camerafollowhelper.cpp

6.33 CameraHead Class Reference

Collaboration diagram for CameraHead:



Public Member Functions

- **CameraHead** (`CameraDolly *cameraDolly`)
- **CameraDolly *** `cameraDolly ()`
- const `glm::quat & relativeOrientation () const`
- void `setRelativeOrientation (const glm::quat &relativeOrientation)`
- `glm::vec3 position () const`
- `glm::quat orientation () const`

Protected Attributes

- `CameraDolly * m_cameraDolly`
- `glm::quat m_relativeOrientation`

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

- `src/camera/camerahead.h`
- `src/camera/camerahead.cpp`

6.34 Starfield::CameraLocation Struct Reference

Public Attributes

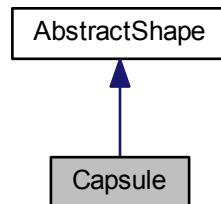
- float `time`
- `glm::vec3 position`
- `glm::quat orientation`

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

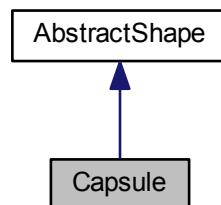
- `src/display/rendering/starfield.h`

6.35 Capsule Class Reference

Inheritance diagram for Capsule:



Collaboration diagram for Capsule:



Public Member Functions

- **Capsule** (const glm::vec3 &origin, const glm::vec3 &direction, const float radius)
- const glm::vec3 & **origin** () const
- void **setOrigin** (const glm::vec3 &origin)
- const glm::vec3 & **direction** () const
- void **setDirection** (const glm::vec3 &direction)
- const float **radius** () const
- void **setRadius** (const float radius)
- virtual bool **intersects** (const [Sphere](#) &sphere) const override
- virtual bool **nearTo** (const [TAABB](#)< int > &aabb) const override
- virtual bool **containedBy** (const [TAABB](#)< int > &aabb) const override

Protected Attributes

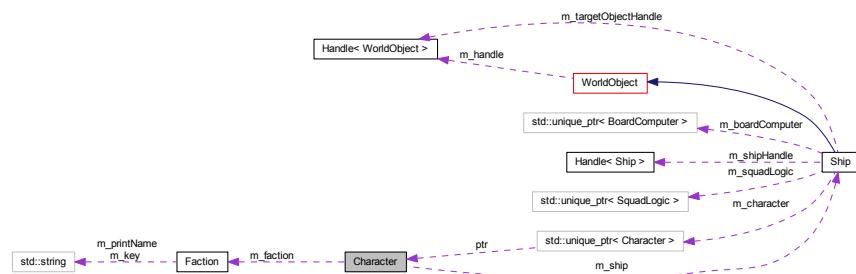
- glm::vec3 **m_origin**
- glm::vec3 **m_direction**
- float **m_radius**

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

- src/geometry/capsule.h
- src/geometry/capsule.cpp

6.36 Character Class Reference

Collaboration diagram for Character:



Public Member Functions

- **Character** ([Ship](#) &[ship](#), [Faction](#) &[faction](#))
- [Faction](#) & [faction](#) ()
- void [setFactio](#)n ([Faction](#) &[faction](#))
- void [setTask](#) (std::shared_ptr<[AiTask](#)> task)
- std::shared_ptr<[AiTask](#)> [task](#) ()
- virtual void [update](#) (float deltaSec)

Protected Attributes

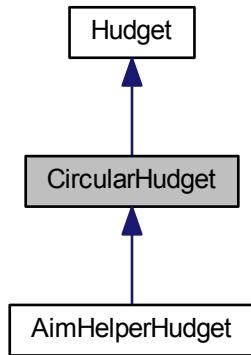
- [Ship](#) & [m_ship](#)
- [Faction](#) * [m_factio](#)n
- std::shared_ptr<[AiTask](#)> [m_task](#)

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

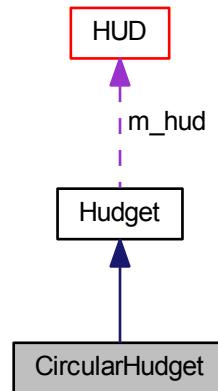
- src/ai/character.h
- src/ai/character.cpp

6.37 CircularHudget Class Reference

Inheritance diagram for CircularHudget:



Collaboration diagram for CircularHudget:



Public Member Functions

- **CircularHudget** (`HUD *hud, float radius`)
- float **radius** () const
- virtual bool **isAt** (const `Ray &ray`) const override

Protected Attributes

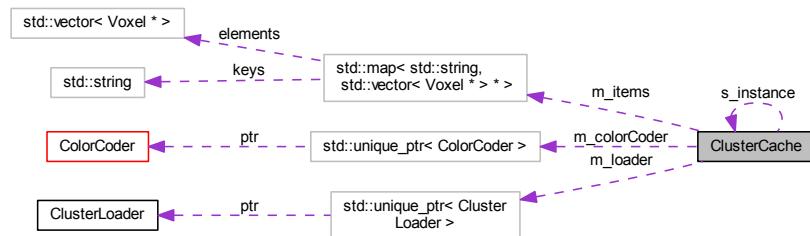
- float **m_radius**

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

- src/ui/hud/circularhudget.h
- src/ui/hud/circularhudget.cpp

6.38 ClusterCache Class Reference

Collaboration diagram for ClusterCache:



Public Member Functions

- void **fillObject** ([WorldObject](#) *worldObject, const std::string &filename)
- void **fillCluster** ([VoxelCluster](#) *cluster, const std::string &filename)

Static Public Member Functions

- static [ClusterCache](#) * **instance** ()

Protected Member Functions

- std::vector< [Voxel](#) * > * **getOrCreate** (const std::string &filename)

Protected Attributes

- std::map< std::string, std::vector< [Voxel](#) * > * > **m_items**
- std::unique_ptr< [ClusterLoader](#) > **m_loader**
- std::unique_ptr< [ColorCoder](#) > **m_colorCoder**

Static Protected Attributes

- static [ClusterCache](#) * **s_instance** = nullptr

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

- src/resource/clustercache.h
- src/resource/clustercache.cpp

6.39 ClusterLoader Class Reference

Public Member Functions

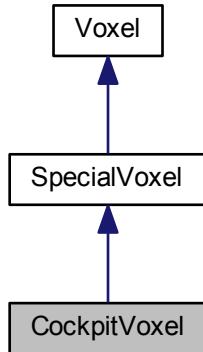
- void **load** (const std::string &filename, std::vector< Voxel * > *list)

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

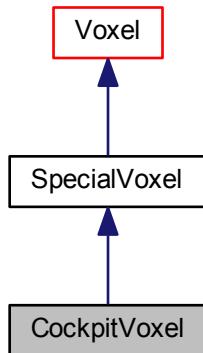
- src/resource/clusterloader.h
- src/resource/clusterloader.cpp

6.40 CockpitVoxel Class Reference

Inheritance diagram for CockpitVoxel:



Collaboration diagram for CockpitVoxel:



Public Member Functions

- **CockpitVoxel** (const glm::ivec3 &gridCell, int index)
- virtual void **addToObject** ([WorldObject](#) *object) override
- virtual void **onRemoval** () override
- virtual void **onDestruction** () override

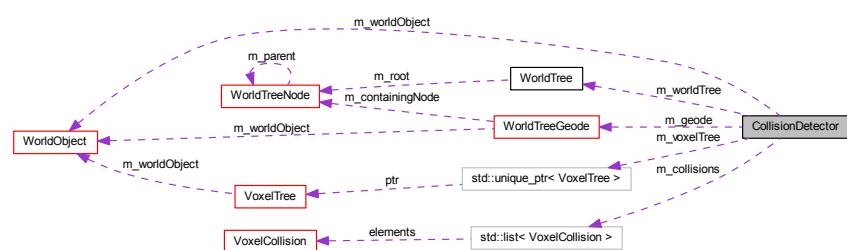
Additional Inherited Members

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

- src/voxel/specialvoxels/cockpitvoxel.h
- src/voxel/specialvoxels/cockpitvoxel.cpp

6.41 CollisionDetector Class Reference

Collaboration diagram for CollisionDetector:



Public Member Functions

- **CollisionDetector** ([WorldObject](#) &worldObject)
- void **addVoxel** ([Voxel](#) *voxel)
- void **removeVoxel** ([Voxel](#) *voxel)
- std::list<[VoxelCollision](#)> & **checkCollisions** ()
- std::list<[VoxelCollision](#)> & **lastCollisions** ()
- void **reset** ()
- [WorldTreeGeode](#) * **geode** ()
- void **setGeode** ([WorldTreeGeode](#) *geode)
- void **setWorldTree** ([WorldTree](#) *worldTree)
- [WorldTree](#) * **worldTree** ()
- [VoxelTree](#) & **voxelTree** ()
- void **updateGeode** ()

Protected Member Functions

- void **checkCollisions** ([VoxelTreeNode](#) *nodeA, [VoxelTreeNode](#) *nodeB)
- const [Sphere](#) & **getOrCreateSphere** ([VoxelTreeNode](#) *node)

Protected Attributes

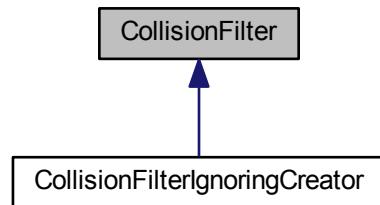
- std::unique_ptr<[VoxelTree](#)> **m voxelTree**
- [WorldObject](#) & **m worldObject**
- [WorldTreeGeode](#) * **m geode**
- [WorldTree](#) * **m worldTree**
- std::list<[VoxelCollision](#)> **m collisions**

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

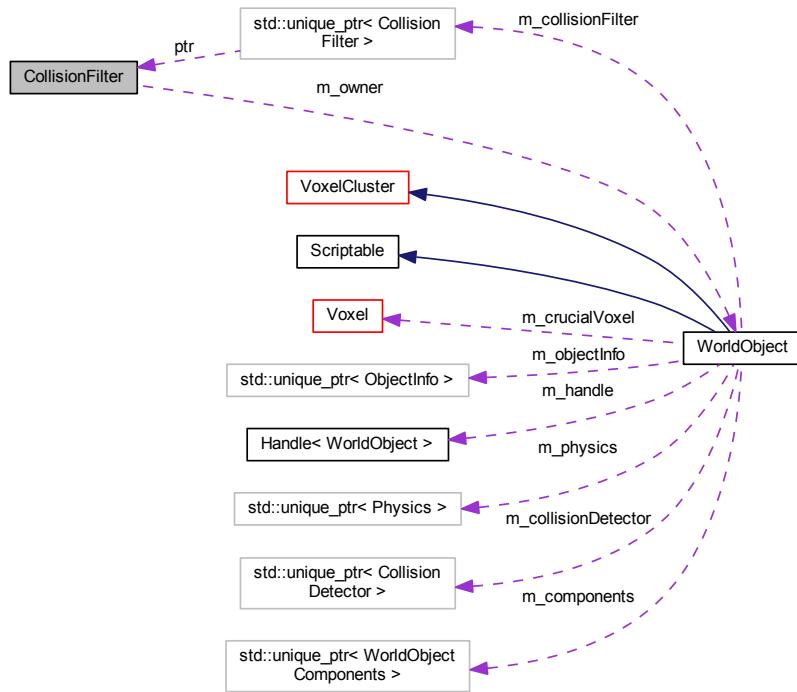
- src/collision/collisiondetector.h
- src/collision/collisiondetector.cpp

6.42 CollisionFilter Class Reference

Inheritance diagram for CollisionFilter:



Collaboration diagram for CollisionFilter:



Public Member Functions

- **CollisionFilter** (`WorldObject *owner, uint32_t collisionMask=0xFFFFFFFF`)
- `uint32_t collisionMask () const`
- `void setCollideableWith (WorldObjectType objectType, bool collides)`
- `bool isCollideableWith (const CollisionFilter *other) const`
- `virtual WorldObject * owner () const`
- `virtual WorldObject * creator () const`

Protected Member Functions

- `bool areMasksCollidable (const CollisionFilter *other) const`
- `virtual bool specialIsCollideableWith (const CollisionFilter *other) const`

Protected Attributes

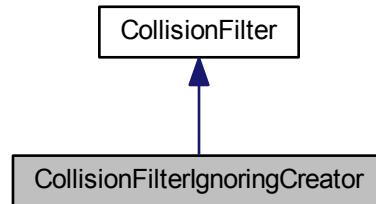
- `uint32_t m_collisionMask`
- `WorldObject * m_owner`

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

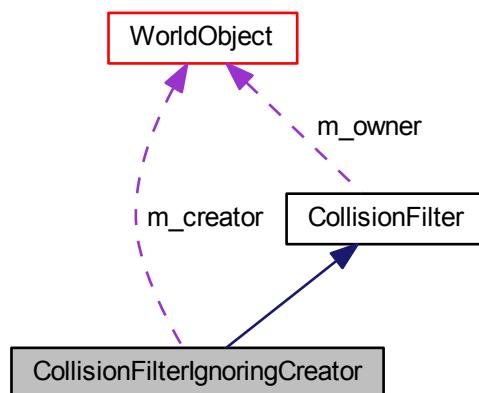
- `src/collision/collisionfilter.h`
- `src/collision/collisionfilter.cpp`

6.43 CollisionFilterIgnoringCreator Class Reference

Inheritance diagram for CollisionFilterIgnoringCreator:



Collaboration diagram for CollisionFilterIgnoringCreator:



Public Member Functions

- **CollisionFilterIgnoringCreator** ([WorldObject](#) *owner, [WorldObject](#) *creator, uint32_t collisionMask=0xFFFF-FFFF)
- virtual [WorldObject](#) * **creator** () const override

Protected Member Functions

- virtual bool **specialIsCollideableWith** (const [CollisionFilter](#) *other) const override

Protected Attributes

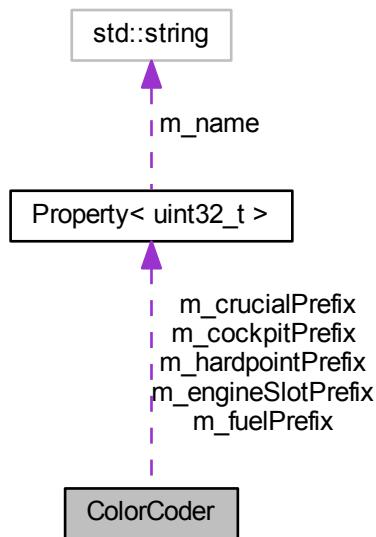
- [WorldObject](#) * **m_creator**

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

- src/collision/collisionfilterignoringcreator.h
- src/collision/collisionfilterignoringcreator.cpp

6.44 ColorCoder Class Reference

Collaboration diagram for ColorCoder:



Public Member Functions

- `Voxel * newCodedVoxel (const Voxel &voxel)`

Protected Attributes

- `Property< uint32_t > m_engineSlotPrefix`
- `Property< uint32_t > m_hardpointPrefix`
- `Property< uint32_t > m_cockpitPrefix`
- `Property< uint32_t > m_fuelPrefix`
- `Property< uint32_t > m_crucialPrefix`

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

- src/resource/colorcoder.h
- src/resource/colorcoder.cpp

6.45 CommandLineParser Class Reference

Public Member Functions

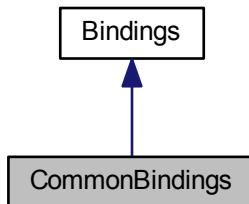
- void **parse** (int argc, char *argv[])
- bool **hmd** () const
- bool **stereoView** () const
- bool **fullScreen** () const

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

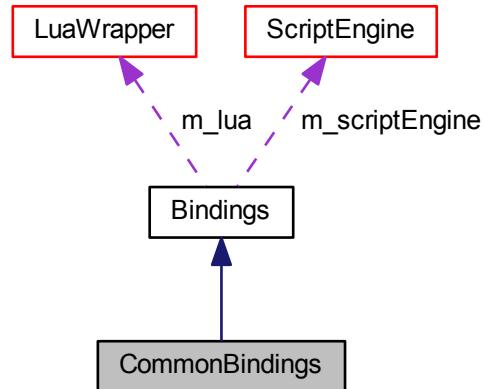
- src/etc/cli/commandlineparser.h
- src/etc/cli/commandlineparser.cpp

6.46 CommonBindings Class Reference

Inheritance diagram for CommonBindings:



Collaboration diagram for CommonBindings:



Public Member Functions

- **CommonBindings** ([GamePlayScript](#) &script)

Protected Member Functions

- virtual void **initialize** ()
- bool **apiIsKeyValid** (apikey key)
- int **apiShowText** (const std::string &string)
- int **apiShowTextFor** (const std::string &string, int seconds)
- int **apiSetEventActive** (apikey eventPoll, bool active)
- apikey **apiCreateSingleShotTimer** (const std::string &callback, float delta)
- apikey **apiCreateLoopingTimer** (const std::string &callback, float delta)
- apikey **apiOnAABBEntered** (apikey worldObject, const glm::vec3 &llf, const glm::vec3 &urb, const std::string &callback)

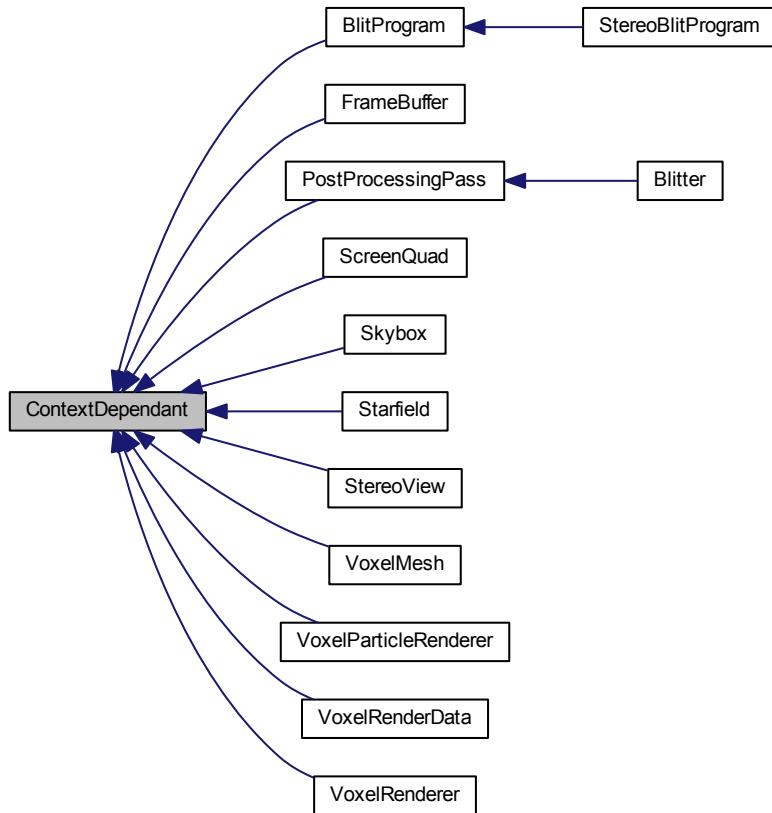
Additional Inherited Members

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

- src/scripting/bindings/commonbindings.h
- src/scripting/bindings/commonbindings.cpp

6.47 ContextDependant Class Reference

Inheritance diagram for ContextDependant:



Protected Member Functions

- virtual void **beforeContextDestroy ()=0**
- virtual void **afterContextRebuild ()=0**

Friends

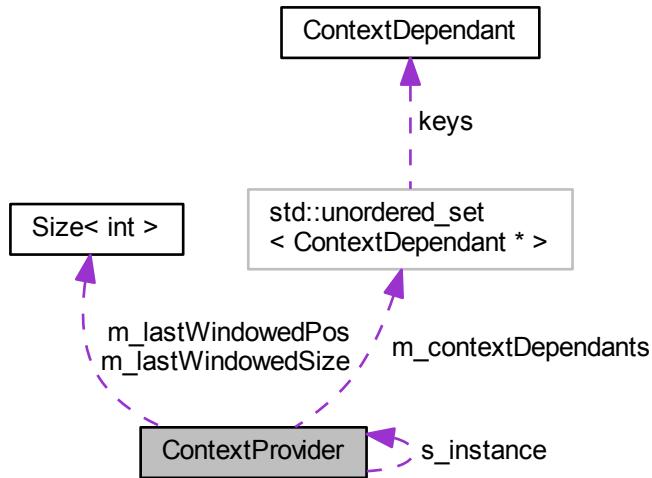
- class **ContextProvider**

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

- src/etc/contextdependant.h
- src/etc/contextdependant.cpp

6.48 ContextProvider Class Reference

Collaboration diagram for ContextProvider:



Public Member Functions

- void **setRequiredGLVersion** (int majorVersionRequire, int minorVersionRequire)
- void **initWindowed** ()
- void **initWindowed** (const `Size< int >` &resolution)
- void **initWindowed** (const `Size< int >` &resolution, const `Size< int >` &position)
- void **initFullScreen** (int monitorIndex=0)
- void **toggleFullScreen** ()
- void **shutdown** ()
- bool **fullScreen** () const
- `Size< int >` **resolution** () const
- `Viewport` **viewport** () const
- float **aspectRatio** () const
- `std::vector< GLFWmonitor * >` **monitors** () const
- int **currentMonitor** () const
- void **registerContextDependant** (`ContextDependant` *dependant)
- void **unregisterContextDependant** (`ContextDependant` *dependant)

Static Public Member Functions

- static `ContextProvider` * **instance** ()

Protected Member Functions

- `Size< int >` **currentResolution** (`GLFWmonitor` *monitor)
- void **setWindowHints** ()

Protected Attributes

- std::unordered_set
 < ContextDependant * > **m_contextDependants**
- bool **m_fullScreen**
- int **m_majorVersionRequire**
- int **m_minorVersionRequire**
- int **m_lastFullScreenMonitorIndex**
- Size< int > **m_lastWindowedPos**
- Size< int > **m_lastWindowedSize**

Static Protected Attributes

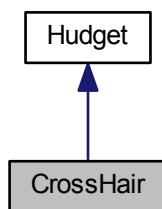
- static ContextProvider * **s_instance** = nullptr

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

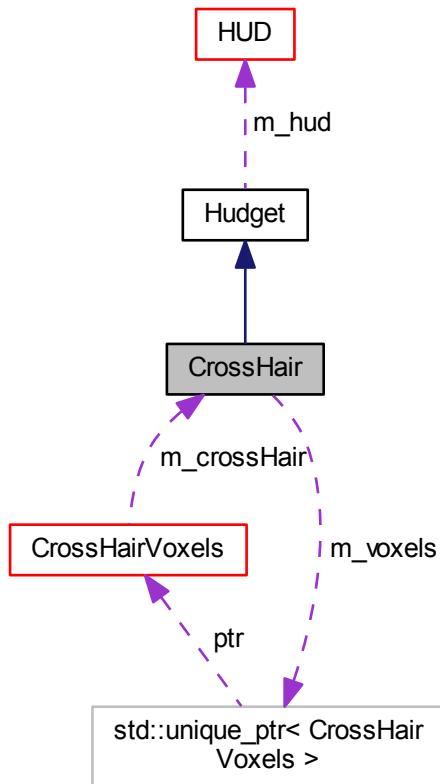
- src/etc/contextprovider.h
- src/etc/contextprovider.cpp

6.49 CrossHair Class Reference

Inheritance diagram for CrossHair:



Collaboration diagram for CrossHair:



Public Member Functions

- **CrossHair (HUD *hud)**
- bool **actionActive () const**
- void **setActionActive (bool actionActive)**
- virtual void **update (float deltaSec) override**
- virtual void **draw () override**

Protected Attributes

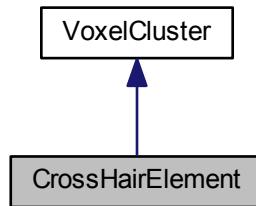
- bool **m_actionActive**
- std::unique_ptr< **CrossHairVoxels** > **m_voxels**

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

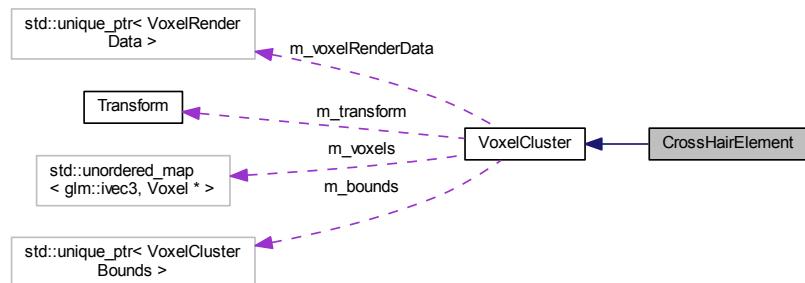
- src/ui/hud/crosshair.h
- src/ui/hud/crosshair.cpp

6.50 CrossHairElement Class Reference

Inheritance diagram for CrossHairElement:



Collaboration diagram for CrossHairElement:



Public Attributes

- `glm::quat relativeOrientation`
- `float zOrientation`

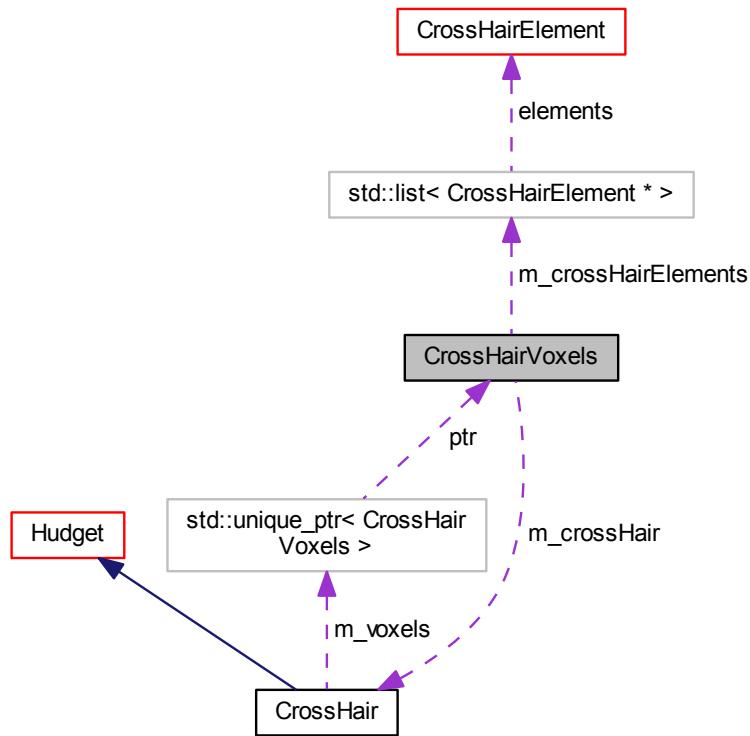
Additional Inherited Members

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

- `src/ui/hud/crosshairvoxels.cpp`

6.51 CrossHairVoxels Class Reference

Collaboration diagram for CrossHairVoxels:



Public Member Functions

- `CrossHairVoxels (CrossHair *crossHair)`
- `void update (float deltaSec)`
- `void draw ()`

Protected Attributes

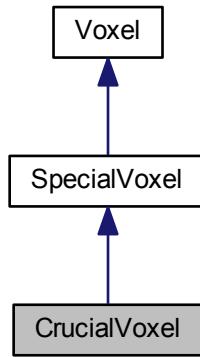
- `CrossHair * m_crossHair`
- `std::list< CrossHairElement * > m_crossHairElements`

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

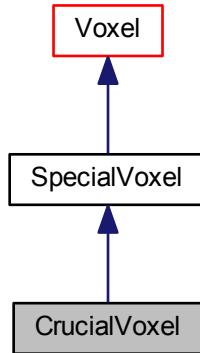
- `src/ui/hud/crosshairvoxels.h`
- `src/ui/hud/crosshairvoxels.cpp`

6.52 CrucialVoxel Class Reference

Inheritance diagram for CrucialVoxel:



Collaboration diagram for CrucialVoxel:



Public Member Functions

- **CrucialVoxel** (const glm::ivec3 &gridCell, int index)
- virtual void **addToObject** ([WorldObject](#) *worldObject)
- virtual void **onRemoval** ()
- virtual void **onDestruction** ()

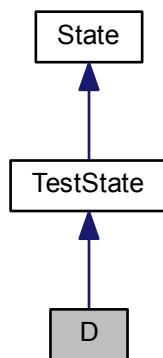
Additional Inherited Members

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

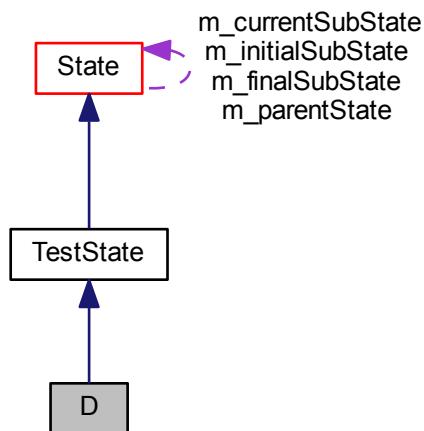
- src/voxel/specialvoxels/crucialvoxel.h
- src/voxel/specialvoxels/crucialvoxel.cpp

6.53 D Class Reference

Inheritance diagram for D:



Collaboration diagram for D:



Public Member Functions

- **D** ([State](#) *parent)

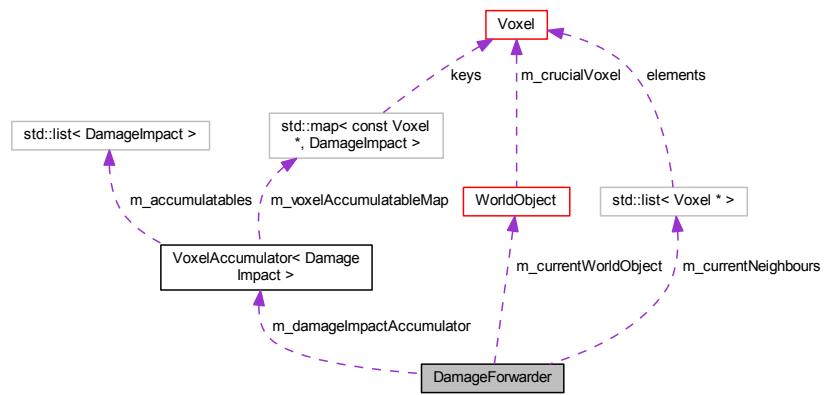
Additional Inherited Members

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

- test/statemachine/teststatemachine.cpp

6.54 DamageForwarder Class Reference

Collaboration diagram for DamageForwarder:



Public Member Functions

- void **forwardDamageImpacts** (`std::list< DamageImpact > &dampedDeadlyDamageImpacts`)
- void **dontForwardTo** (`std::list< Voxel * > &deadVoxels`)
- `std::list< DamageImpact > forwardedDamageImpacts ()`

Protected Member Functions

- float **forwardFactor** (float dotProduct, float fieldOfDamage, int neighbours)

Protected Attributes

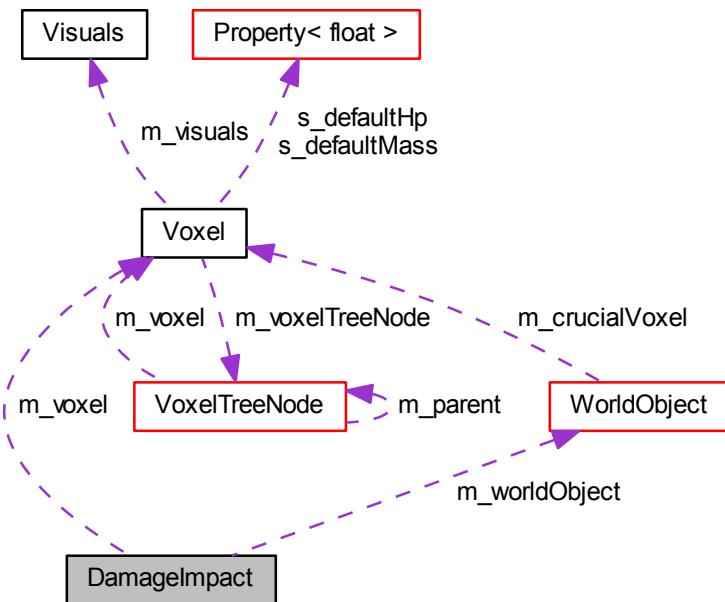
- `WorldObject * m_currentWorldObject`
- `std::list< Voxel * > * m_currentNeighbours`
- `glm::ivec3 m_currentGridCell`
- `VoxelAccumulator< DamageImpact > m_damageImpactAccumulator`

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

- `src/world/handler/damageforwarder.h`
- `src/world/handler/damageforwarder.cpp`

6.55 DamageImpact Class Reference

Collaboration diagram for DamageImpact:



Public Member Functions

- `DamageImpact (WorldObject *worldObject, Voxel *voxel, const glm::vec3 &damageVec, float fieldOfDamage)`
- `WorldObject * worldObject ()`
- `const WorldObject * worldObject () const`
- `Voxel * voxel ()`
- `const Voxel * voxel () const`
- `const glm::vec3 & damageVec () const`
- `float damage () const`
- `float fieldOfDamage () const`
- `void add (const DamageImpact &damageImpact)`

Protected Attributes

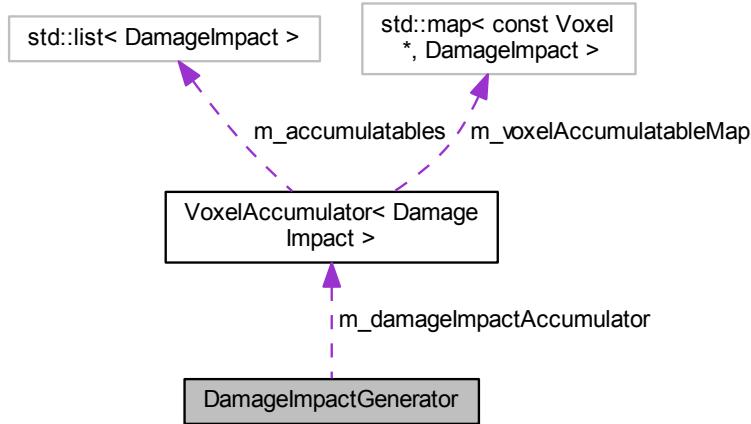
- `WorldObject * m_worldObject`
- `Voxel * m voxel`
- `glm::vec3 m_damageVec`
- `float m_fieldOfDamage`

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

- `src/world/helper/damageimpact.h`
- `src/world/helper/damageimpact.cpp`

6.56 DamageImpactGenerator Class Reference

Collaboration diagram for DamageImpactGenerator:



Public Member Functions

- void `parse` (`std::list< WorldObjectCollision > &worldObjectCollisions)`
- `std::list< DamageImpact > & damageImpacts ()`

Protected Attributes

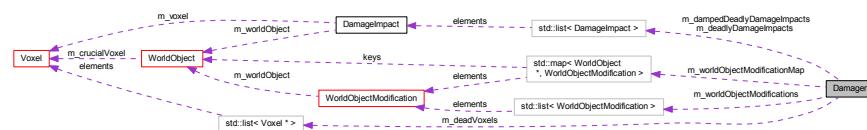
- `VoxelAccumulator< DamageImpact > m_damageImpactAccumulator`

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

- `src/world/handler/damageimpactgenerator.h`
- `src/world/handler/damageimpactgenerator.cpp`

6.57 Damager Class Reference

Collaboration diagram for Damager:



Public Member Functions

- void `applyDamages` (`std::list< DamageImpact > &damagelImpacts)`

- void **reset** ()
- std::list< [DamageImpact](#) > & **dampedDeadlyDamageImpacts** ()
- std::list< [DamageImpact](#) > & **deadlyDamageImpacts** ()
- std::list< [Voxel](#) * > & **deadVoxels** ()
- std::list< [WorldObjectModification](#) > & **worldObjectModifications** ()

Protected Member Functions

- [DamageImpact](#) **dampDamageImpact** ([DamageImpact](#) &undamped, float factor)

Protected Attributes

- std::list< [DamageImpact](#) > **m_dampedDeadlyDamageImpacts**
- std::list< [DamageImpact](#) > **m_deadlyDamageImpacts**
- std::list< [Voxel](#) * > **m_deadVoxels**
- std::map< [WorldObject](#) *, [WorldObjectModification](#) > **m_worldObjectModificationMap**
- std::list< [WorldObjectModification](#) > **m_worldObjectModifications**

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

- src/world/handler/damager.h
- src/world/handler/damager.cpp

6.58 DdsTexture Class Reference

Static Public Member Functions

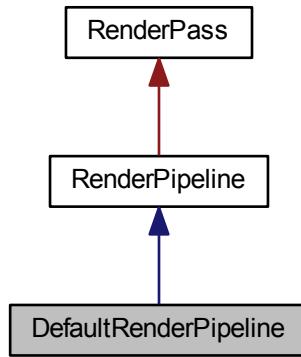
- static bool **loadImage2d** (glow::Texture *texture, std::string path)
- static bool **loadImageCube** (glow::Texture *texture, std::string pathXp, std::string pathXn, std::string pathYp, std::string pathYn, std::string pathZp, std::string pathZn)

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

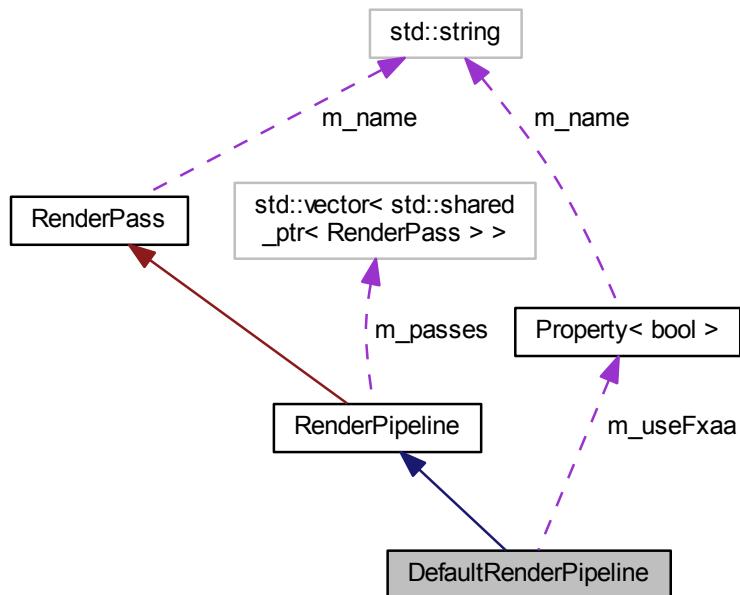
- src/resource/ddstexture.h
- src/resource/ddstexture.cpp

6.59 DefaultRenderPipeline Class Reference

Inheritance diagram for DefaultRenderPipeline:



Collaboration diagram for DefaultRenderPipeline:



Public Member Functions

- virtual void **apply** (FrameBuffer &frameBuffer, const RenderMetaData &metadata) override
- virtual void **setup** () override

- virtual int **bufferCount** () override
- void **addFXAA** ()
- void **addEmissivenessBlurVertical** ()
- void **addEmissivenessBlurHorizontal** ()
- void **addFinalization** ()

Protected Attributes

- std::shared_ptr< ScreenQuad > **m_quad**
- std::shared_ptr< PostProcessingPass > **m_fxaa**
- std::shared_ptr< PostProcessingPass > **m_finalization**
- Property< bool > **m_useFxaa**

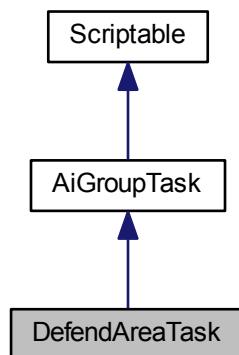
Additional Inherited Members

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

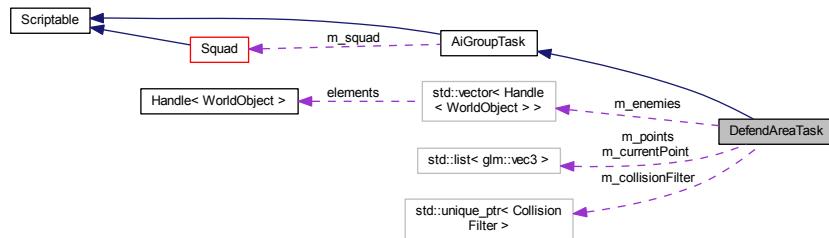
- src/display/rendering/defaultrendertask.h
- src/display/rendering/defaultrendertask.cpp

6.60 DefendAreaTask Class Reference

Inheritance diagram for DefendAreaTask:



Collaboration diagram for DefendAreaTask:



Public Member Functions

- **DefendAreaTask** ([Squad](#) &squad, [std::list<glm::vec3>](#) points, float defendRange)
- void **addPoint** (const [glm::vec3](#) &point)
- const [std::list<glm::vec3>](#) & **points** ()
- float **range** ()
- virtual void **update** (float deltaSec) override

Protected Member Functions

- virtual void **onNewLeader** ([Ship](#) *leader) override
- virtual void **onMemberJoin** ([Ship](#) *member) override
- void **updatePatrol** ()
- void **updateFight** ()
- bool **isEnemyInRange** ()

Protected Attributes

- [std::unique_ptr<CollisionFilter>](#) **m_collisionFilter**
- [std::shared_ptr<FlyToTask>](#) **m_leaderFlyTask**
- [std::shared_ptr<FightTask>](#) **m_fightTask**
- [std::list<glm::vec3>](#) **m_points**
- [std::list<glm::vec3>::iterator](#) **m_currentPoint**
- [std::vector<Handle<WorldObject>>](#) **m_enemies**
- float **m_defendRange**

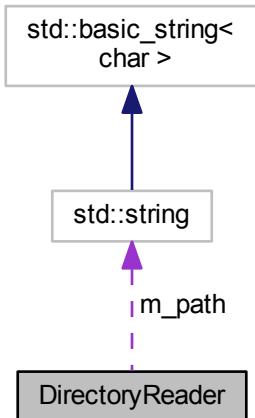
Additional Inherited Members

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

- [src/ai/grouptasks/defendareatask.h](#)
- [src/ai/grouptasks/defendareatask.cpp](#)

6.61 DirectoryReader Class Reference

Collaboration diagram for DirectoryReader:



Public Member Functions

- `DirectoryReader (const std::string &path)`
- `std::list< std::string > read () const`

Protected Attributes

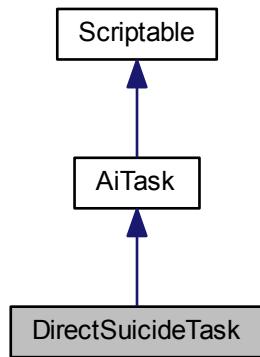
- `std::string m_path`

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

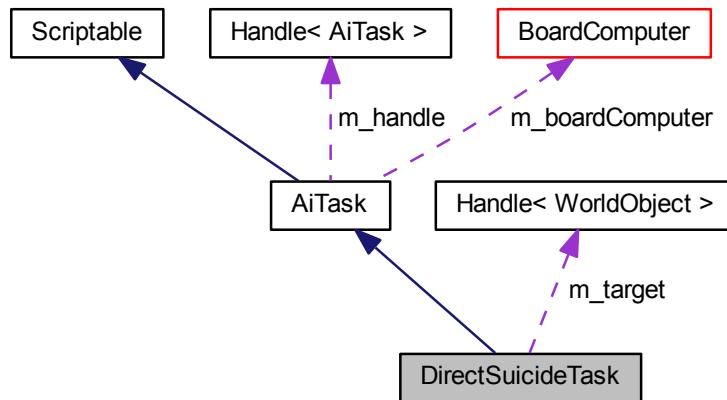
- `src/utils/directoryreader.h`
- `src/utils/directoryreader.cpp`

6.62 DirectSuicideTask Class Reference

Inheritance diagram for DirectSuicideTask:



Collaboration diagram for DirectSuicideTask:



Public Member Functions

- **DirectSuicideTask** (`BoardComputer` *boardComputer, `WorldObject` *target)
- void **setTarget** (`WorldObject` *target)
- virtual void **update** (float deltaSec)

Protected Attributes

- `Handle< WorldObject > m_target`

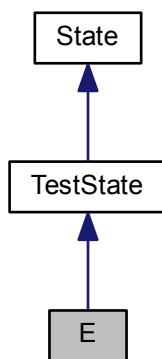
Additional Inherited Members

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

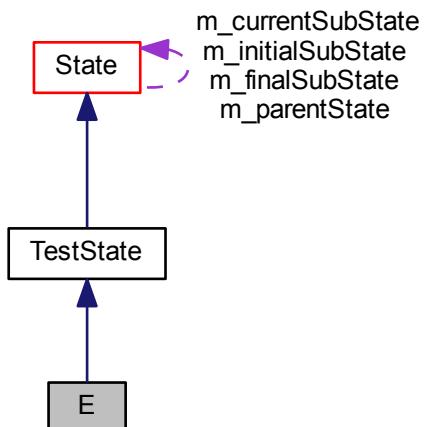
- src/ai/basictasks/directsuicidetask.h
- src/ai/basictasks/directsuicidetask.cpp

6.63 E Class Reference

Inheritance diagram for E:



Collaboration diagram for E:



Public Member Functions

- **E** ([State](#) *parent)

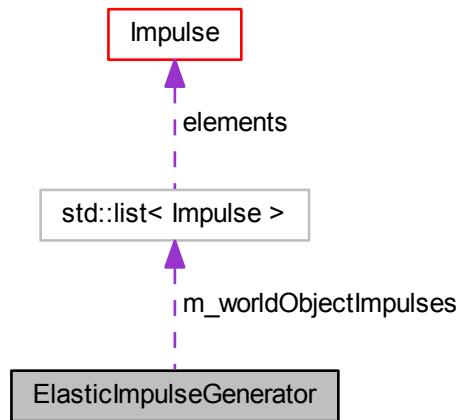
Additional Inherited Members

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

- test/statemachine/teststatemachine.cpp

6.64 ElasticImpulseGenerator Class Reference

Collaboration diagram for ElasticImpulseGenerator:



Public Member Functions

- void **parse** (`std::list< WorldObjectCollision > &worldObjectCollisions)`
- `std::list< Impulse > & worldObjectImpulses ()`

Protected Member Functions

- void **generateImpulse** (`VoxelCollisionParticipant &from, VoxelCollisionParticipant &to`)

Protected Attributes

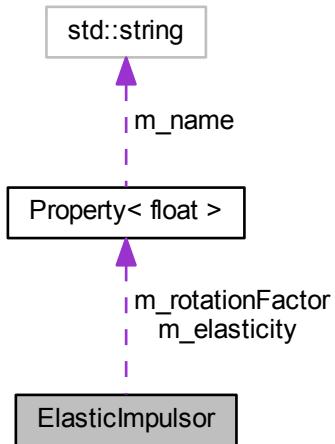
- `std::list< Impulse > m_worldObjectImpulses`

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

- `src/world/handler/elasticimpulsegenerator.h`
- `src/world/handler/elasticimpulsegenerator.cpp`

6.65 ElasticImpulsor Class Reference

Collaboration diagram for ElasticImpulsor:



Public Member Functions

- void `parse` (`std::list< Impulse > &worldObjectImpulses)`

Protected Attributes

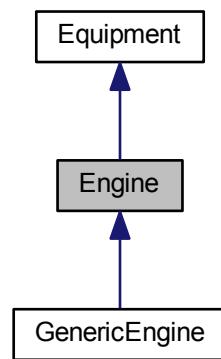
- `Property< float > m_rotationFactor`
- `Property< float > m_elasticity`

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

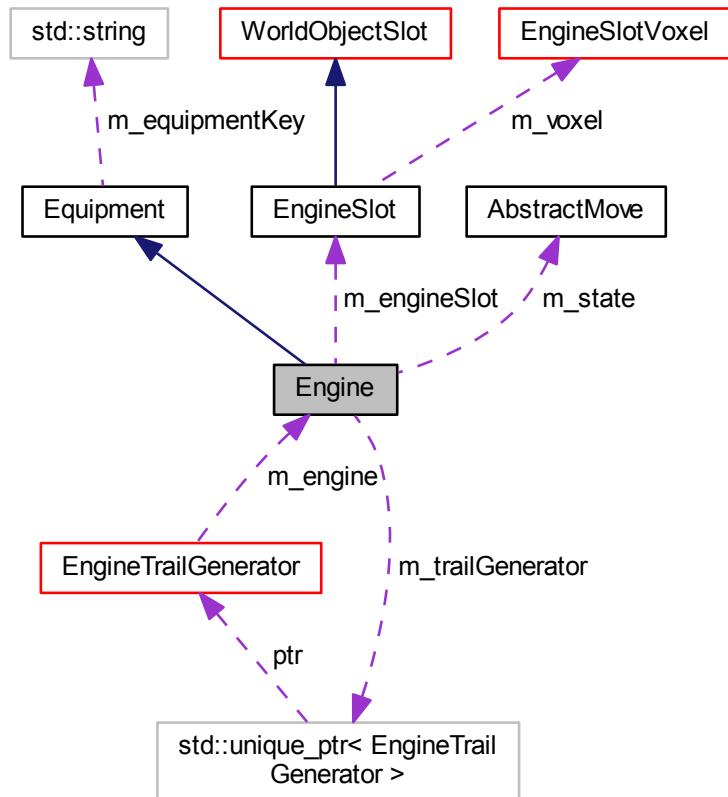
- `src/world/handler/elasticimpulsor.h`
- `src/world/handler/elasticimpulsor.cpp`

6.66 Engine Class Reference

Inheritance diagram for Engine:



Collaboration diagram for Engine:



Public Member Functions

- **Engine** (const std::string &equipmentKey)
- virtual const **Visuals** & **visuals** () const =0
- virtual const **SoundProperties** & **sound** () const =0
- **EngineSlot** * **engineSlot** ()
- const **EngineSlot** * **engineSlot** () const
- void **setEngineSlot** (**EngineSlot** *engineSlot)
- virtual **EnginePower** **power** () const =0
- const **EngineState** & **state** () const
- void **setState** (const **EngineState** &state)
- **Acceleration** **currentAcceleration** () const
- virtual void **update** (float deltaSec)

Protected Member Functions

- void **setupTrail** ()

Protected Attributes

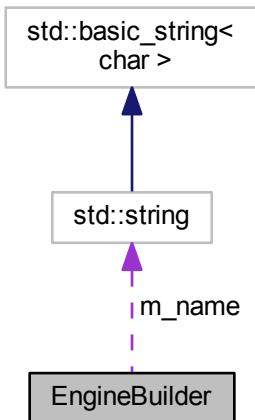
- std::unique_ptr<EngineTrailGenerator> **m_trailGenerator**
- std::shared_ptr<Sound> **m_sound**
- EngineSlot * **m_engineSlot**
- EngineState **m_state**

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

- src/equipment/engine.h
- src/equipment/engine.cpp

6.67 EngineBuilder Class Reference

Collaboration diagram for EngineBuilder:



Public Member Functions

- **EngineBuilder** (const std::string &name)
- **Engine * build ()**

Protected Attributes

- std::string **m_name**

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

- src/resource/enginebuilder.h
- src/resource/enginebuilder.cpp

6.68 EnginePower Class Reference

Public Member Functions

- **EnginePower** (const glm::vec4 &directional, const glm::vec3 &angular)
- const glm::vec4 & **directional** () const
- void **setDirectional** (const glm::vec4 &directional)
- const glm::vec3 & **angular** () const
- void **setAngular** (const glm::vec3 &angular)
- Acceleration **accelerationAt** (const EngineState &engineState)
- EnginePower & **operator+=** (const EnginePower &other)

Static Public Member Functions

- static EnginePower **fromProperties** (const std::string &prefix)

Protected Attributes

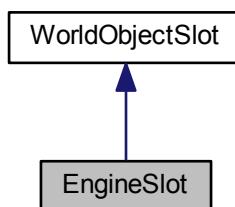
- glm::vec4 **m_directional**
- glm::vec3 **m_angular**

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

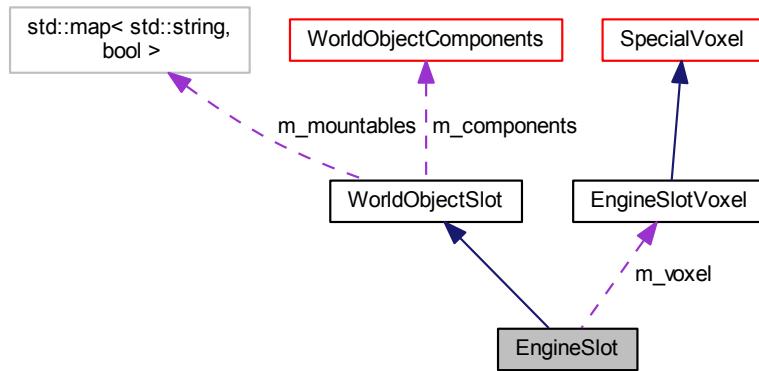
- src/equipment/enginepower.h
- src/equipment/enginepower.cpp

6.69 EngineSlot Class Reference

Inheritance diagram for EngineSlot:



Collaboration diagram for EngineSlot:



Public Member Functions

- `EngineSlot (WorldObjectComponents *components, EngineSlotVoxel *voxel)`
- const `EngineSlotVoxel * voxel () const`
- const `glm::vec3 & direction ()`
- void `setDirection (const glm::vec3 &direction)`
- const `std::shared_ptr< Engine > & engine ()`
- void `setEngine (const std::shared_ptr< Engine > &engine)`
- void `update (float deltaSec)`
- void `onVoxelRemoval ()`

Protected Attributes

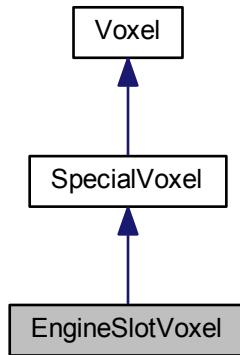
- `EngineSlotVoxel * m voxel`
- `std::shared_ptr< Engine > m engine`
- `glm::vec3 m direction`

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

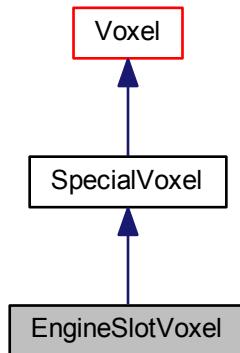
- `src/equipment/engineslot.h`
- `src/equipment/engineslot.cpp`

6.70 EngineSlotVoxel Class Reference

Inheritance diagram for EngineSlotVoxel:



Collaboration diagram for EngineSlotVoxel:



Public Member Functions

- **EngineSlotVoxel** (const glm::ivec3 &gridCell, int index)
- virtual [Visuals visuals](#) () const override
- virtual void [addToObject](#) ([WorldObject](#) *worldObject) override
- virtual void [onRemoval](#) () override
- virtual void [onDestruction](#) () override

Protected Attributes

- std::shared_ptr<[EngineSlot](#)> **m_engineSlot**

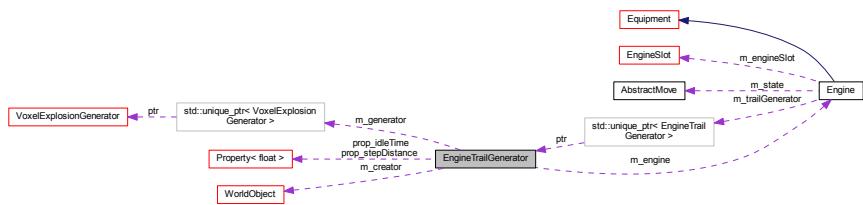
Additional Inherited Members

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

- src/voxel/specialvoxels/engineslotvoxel.h
- src/voxel/specialvoxels/engineslotvoxel.cpp

6.71 EngineTrailGenerator Class Reference

Collaboration diagram for EngineTrailGenerator:



Public Member Functions

- **EngineTrailGenerator** (`Engine &engine, const WorldObject &creator`)
- `void setLifetime (float lifetime)`
- `void setColor (int color)`
- `void setEmissiveness (float emissiveness)`
- `void update (float deltaSec)`

Protected Member Functions

- `void spawnTrail ()`
- `void updateTrailSettings ()`
- `glm::vec3 calculateSpawnPosition ()`
- `void spawnAt (glm::vec3 position)`

Protected Attributes

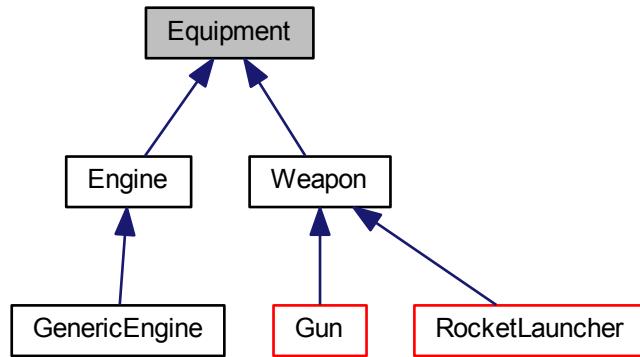
- `const WorldObject & m_creator`
- `const Engine & m_engine`
- `std::unique_ptr<VoxelExplosionGenerator> m_generator`
- `glm::vec3 m_lastSpawnPoint`
- `bool m_lastValid`
- `float m_stepRest`
- `double m_timeSinceLastSpawn`
- `float m_spawnOffset`
- `Property<float> prop_stepDistance`
- `Property<float> prop_idleTime`

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

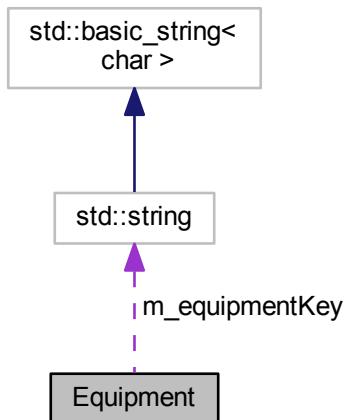
- src/voxeleffect/enginetrailgenerator.h
- src/voxeleffect/enginetrailgenerator.cpp

6.72 Equipment Class Reference

Inheritance diagram for Equipment:



Collaboration diagram for Equipment:



Public Member Functions

- **Equipment** (const std::string &equipmentKey)
- const std::string & **equipmentKey** () const

Protected Attributes

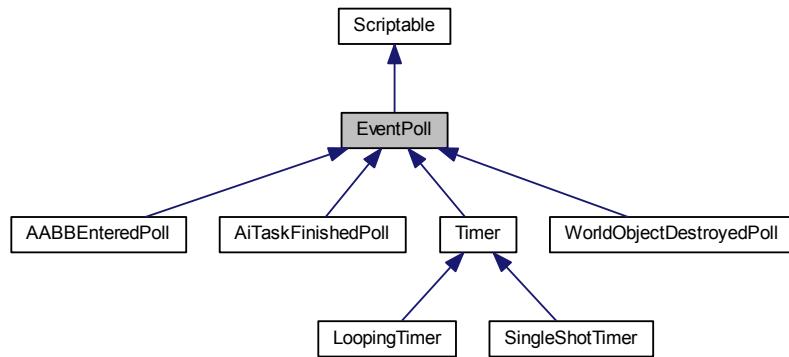
- std::string **m_equipmentKey**

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

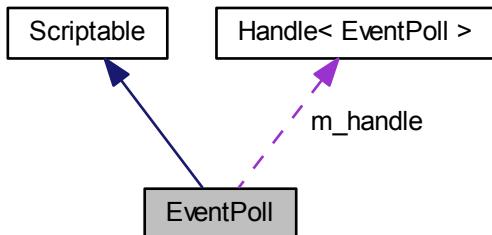
- src/equipment/equipment.h
- src/equipment/equipment.cpp

6.73 EventPoll Class Reference

Inheritance diagram for EventPoll:



Collaboration diagram for EventPoll:



Public Member Functions

- **EventPoll** (const std::function< void()> &callback)
- virtual bool **isDead** ()
- virtual void **update** (float deltaSec)
- bool **isActive** () const
- void **setActive** (bool active)
- [Handle< EventPoll > & handle \(\)](#)

Protected Member Functions

- void **doCallback** ()
- virtual bool **poll** ()=0
- virtual void **specialOnCallback** ()

Protected Attributes

- std::function< void()> **m_callback**
- Handle< EventPoll > **m_handle**
- bool **m_active**

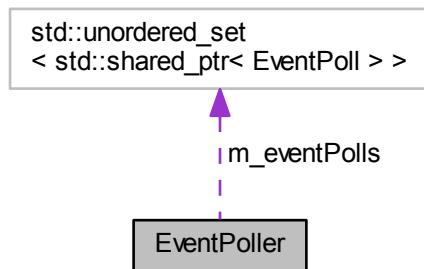
Additional Inherited Members

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

- src/events/eventpoll.h
- src/events/eventpoll.cpp

6.74 EventPoller Class Reference

Collaboration diagram for EventPoller:



Public Member Functions

- void **addPoll** (std::shared_ptr< EventPoll > eventPoll)
- void **removePoll** (std::shared_ptr< EventPoll > eventPoll)
- void **update** (float deltaSec)

Protected Attributes

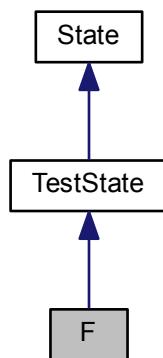
- std::unordered_set<std::shared_ptr<EventPoll>> **m_eventPolls**

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

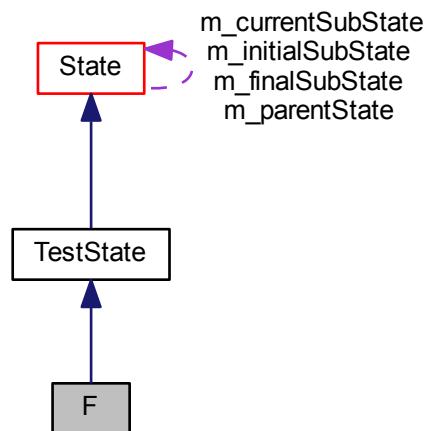
- src/events/eventpoller.h
- src/events/eventpoller.cpp

6.75 F Class Reference

Inheritance diagram for F:



Collaboration diagram for F:



Public Member Functions

- **F (State *parent)**

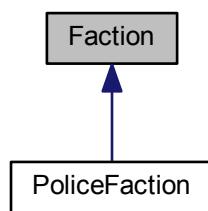
Additional Inherited Members

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

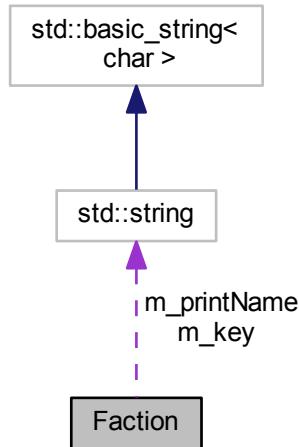
- test/statemachine/teststatemachine.cpp

6.76 Faction Class Reference

Inheritance diagram for Faction:



Collaboration diagram for Faction:



Public Member Functions

- **Faction** (const std::string &key, const std::string &printName)
- const std::string & **key** () const
- void **setPrintName** (const std::string &printName)

- const std::string & **printName** () const
- **FactionRelation** & **relationTo** (**Faction** &other)

Protected Attributes

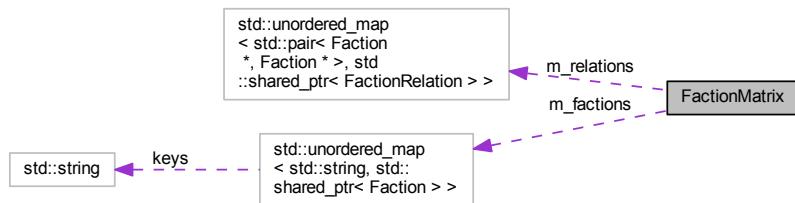
- std::string **m_key**
- std::string **m_printName**

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

- src/factions/faction.h
- src/factions/faction.cpp

6.77 FactionMatrix Class Reference

Collaboration diagram for FactionMatrix:



Public Member Functions

- **Faction** & **pirateFacton** ()
- **Faction** & **policeFacton** ()
- **Faction** & **playerFacton** ()
- **Faction** & **unknownFacton** ()
- **Faction** & **getFacton** (const std::string &factionName)
- void **addFacton** (std::shared_ptr<Faction> facton)
- **FactionRelation** & **getRelation** (**Faction** &factionA, **Faction** &factionB)
- **FactionRelation** & **getRelationToPlayer** (**Faction** &faction)

Protected Member Functions

- void **setupRelations** ()
- std::pair<Faction*, Fraction*> **uniquePair** (Faction &factionA, Fraction &factionB)

Protected Attributes

- std::unordered_map<std::string, std::shared_ptr<Fraction>> **m_factions**

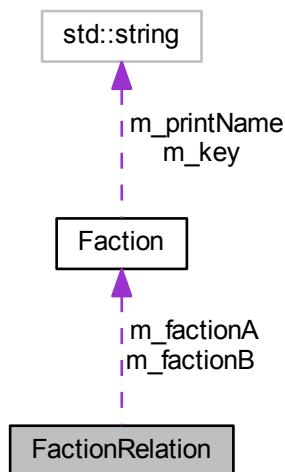
- std::unordered_map< std::pair< [Faction](#) *, [Faction](#) * >, std::shared_ptr< [FactionRelation](#) > > **m_relations**

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

- src/factions/factionmatrix.h
- src/factions/factionmatrix.cpp

6.78 [FactionRelation Class Reference](#)

Collaboration diagram for [FactionRelation](#):



Public Member Functions

- [FactionRelation \(Faction &factionA, Faction &factionB, float friendliness\)](#)
- [Fraction & **factionA** \(\)](#)
- [Fraction & **factionB** \(\)](#)
- float [friendliness \(\) const](#)
- void [setFriendliness \(float friendliness\)](#)
- [FactionRelationType **type** \(\) const](#)

Static Public Member Functions

- static std::string [typeName \(FactionRelationType type\)](#)

Protected Attributes

- [Fraction & **m_factionA**](#)

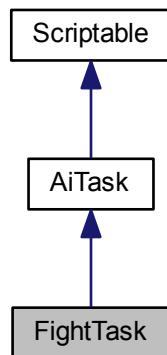
- **Faction & m_factioN**
- float **m_friendliness**
- FactionRelationType **m_type**

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

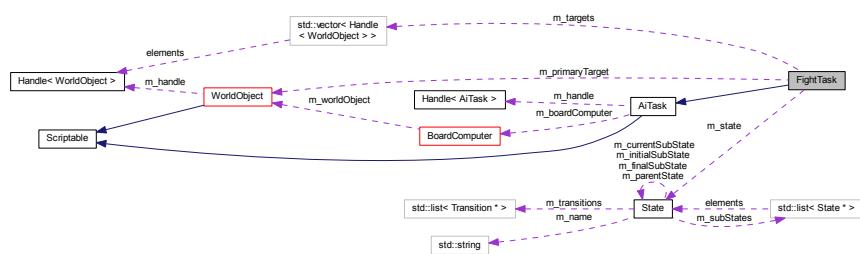
- src/factions/factionrelation.h
- src/factions/factionrelation.cpp

6.79 FightTask Class Reference

Inheritance diagram for FightTask:



Collaboration diagram for FightTask:



Public Member Functions

- **FightTask** (BoardComputer *boardComputer, const std::vector<Handle<WorldObject>> &targets)
- virtual void **update** (float deltaSec)
- std::vector<Handle<WorldObject>> & **targets ()**
- virtual void **addTarget** (const Handle<WorldObject> &targets)
- virtual void **setTargets** (const std::vector<Handle<WorldObject>> &targets)
- virtual bool **isFinished ()**

Protected Types

- enum **State** { **IDLE**, **APPROACH**, **ENGAGE**, **EVADE** }

Protected Member Functions

- void **updateTargets** ()
- void **updateState** ()
- void **setState** ([State](#) newState)
- glm::vec3 **findRandomEvasionPoint** ()
- float **targetDistance** ()
- float **pointDistance** (glm::vec3 point)
- float **angleToTarget** ()

Protected Attributes

- std::vector< [Handle](#) < [WorldObject](#) > > **m_targets**
- [WorldObject](#) * **m_primaryTarget**
- [State](#) **m_state**
- bool **m_stateChanged**
- float **m_maxFireDistance**
- float **m_maxRocketDistance**
- float **m_minEnemyDistance**
- glm::vec3 **m_positionBehindTarget**

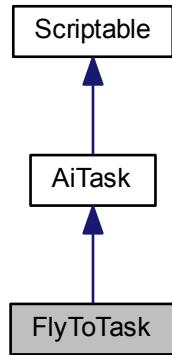
Additional Inherited Members

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

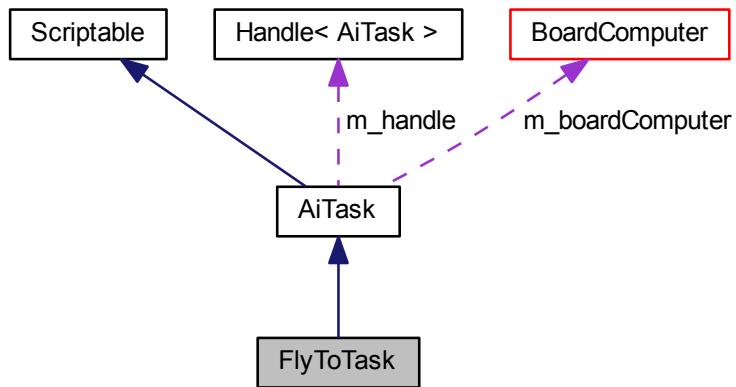
- src/ai/basictasks/fighttask.h
- src/ai/basictasks/fighttask.cpp

6.80 FlyToTask Class Reference

Inheritance diagram for FlyToTask:



Collaboration diagram for FlyToTask:



Public Member Functions

- **FlyToTask** ([BoardComputer](#) *boardComputer)
- void **setTargetPoint** (const [glm::vec3](#) &point, const [glm::vec3](#) &up=[glm::vec3\(0, 0, 0\)](#))
- virtual void **update** (float deltaSec) override
- virtual bool **isFinished** ()

Protected Attributes

- [glm::vec3 m_targetPoint](#)

- `glm::vec3 m_targetUp`
- `float m_minDistance`

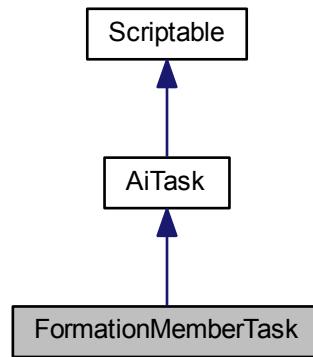
Additional Inherited Members

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

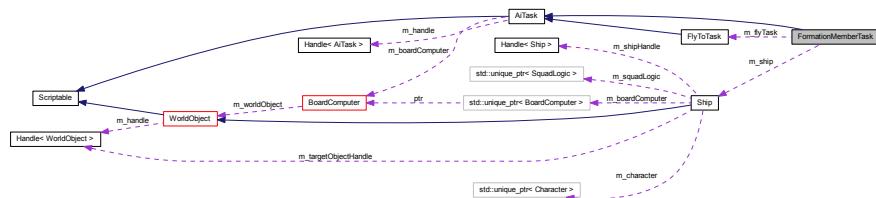
- `src/ai/basictasks/flytotask.h`
- `src/ai/basictasks/flytotask.cpp`

6.81 FormationMemberTask Class Reference

Inheritance diagram for FormationMemberTask:



Collaboration diagram for FormationMemberTask:



Public Member Functions

- **FormationMemberTask** (`Ship &ship`)
- virtual void **update** (float deltaSec) override

Protected Attributes

- `FlyToTask m_flyTask`
- `Ship & m_ship`

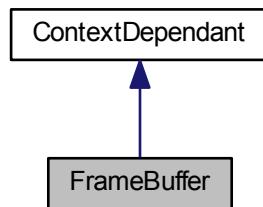
Additional Inherited Members

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

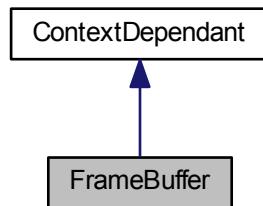
- src/ai/basictasks/formationmembertask.h
- src/ai/basictasks/formationmembertask.cpp

6.82 FrameBuffer Class Reference

Inheritance diagram for FrameBuffer:



Collaboration diagram for FrameBuffer:



Public Member Functions

- **FrameBuffer** (int colorAttachments=1, bool depthAttachment=true)
- void **bind** ()
- void **unbind** ()
- void **clear** ()
- glow::FrameBufferObject & **get** ()
- void **setDrawBuffers** (const std::vector< int > &buffers)
- void **setResolution** (const glm::ivec2 &resolution)
- const glm::ivec2 & **resolution** ()
- glow::Texture * **texture** (int i)

Protected Member Functions

- void **setupFBO** ()
- virtual void **beforeContextDestroy** () override
- virtual void **afterContextRebuild** () override

Protected Attributes

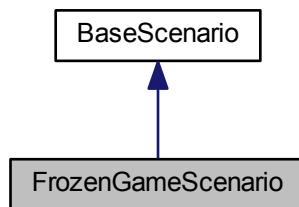
- int **m_colorAttachmentCount**
- bool **m_useDepthAttachment**
- glm::ivec2 **m_resolution**
- glow::ref_ptr<glow::FrameBufferObject> **m_fbo**

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

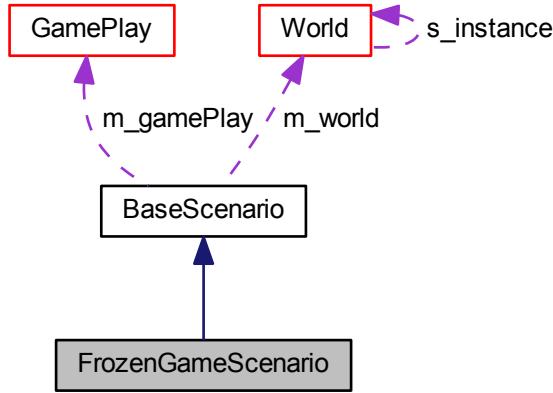
- src/display/rendering/framebuffer.h
- src/display/rendering/framebuffer.cpp

6.83 FrozenGameScenario Class Reference

Inheritance diagram for FrozenGameScenario:



Collaboration diagram for FrozenGameScenario:



Public Member Functions

- **FrozenGameScenario** (`GamePlay` *`inGame`)

Protected Member Functions

- `virtual void populateWorld () override`

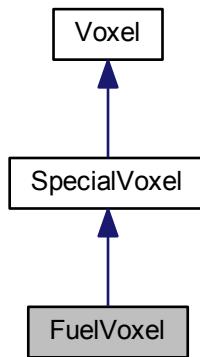
Additional Inherited Members

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

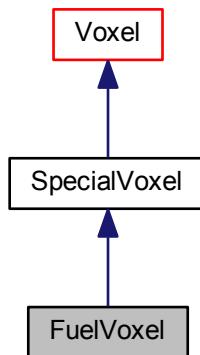
- `src/scenarios/frozengamescenario.h`
- `src/scenarios/frozengamescenario.cpp`

6.84 FuelVoxel Class Reference

Inheritance diagram for FuelVoxel:



Collaboration diagram for FuelVoxel:



Public Member Functions

- **FuelVoxel** (const glm::ivec3 &gridCell, int index)
- virtual void **addToObject** (WorldObject *worldObject) override
- virtual float **damageForwardingDestructionDamage** () override
- virtual void **onRemoval** () override
- virtual void **onDestruction** () override

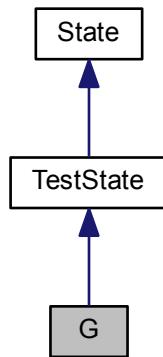
Additional Inherited Members

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

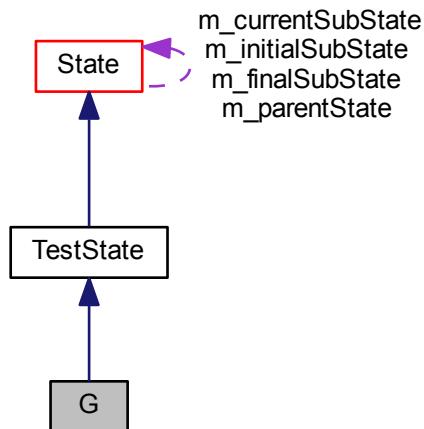
- src/voxel/specialvoxels/fuelvoxel.h
- src/voxel/specialvoxels/fuelvoxel.cpp

6.85 G Class Reference

Inheritance diagram for G:



Collaboration diagram for G:



Public Member Functions

- **G** ([State](#) *parent)

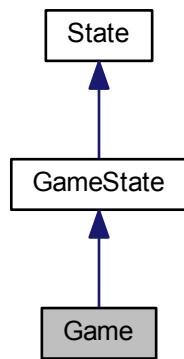
Additional Inherited Members

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

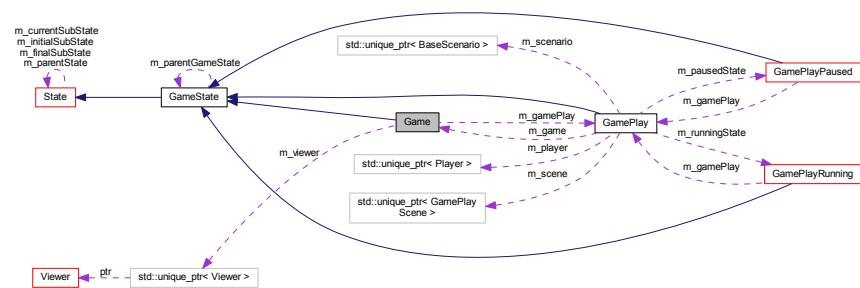
- test/statemachine/teststatemachine.cpp

6.86 Game Class Reference

Inheritance diagram for Game:



Collaboration diagram for Game:



Public Member Functions

- **GamePlay & gamePlay ()**
- virtual const **Scene & scene ()** const override
- virtual const **CameraHead & cameraHead ()** const override
- **HMDManager & hmdManager ()**
- **Viewer & viewer ()**
- virtual void **update** (float deltaSec) override
- void **draw ()**

Protected Attributes

- std::shared_ptr< HMDManager > **m_hmdManager**
- std::unique_ptr< Viewer > **m_viewer**
- GamePlay * **m_gamePlay**

Additional Inherited Members

6.86.1 Member Function Documentation

6.86.1.1 void Game::update (float deltaSec) [override], [virtual]

Performs a [Transition](#) from the currentSubState, if such isPossible()

Reimplemented from [GameState](#).

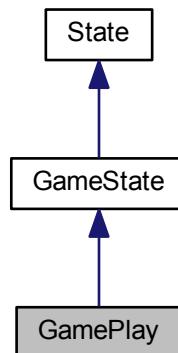
The documentation for this class was generated from the following files:

- src/gamestate/game.h
- src/gamestate/game.cpp

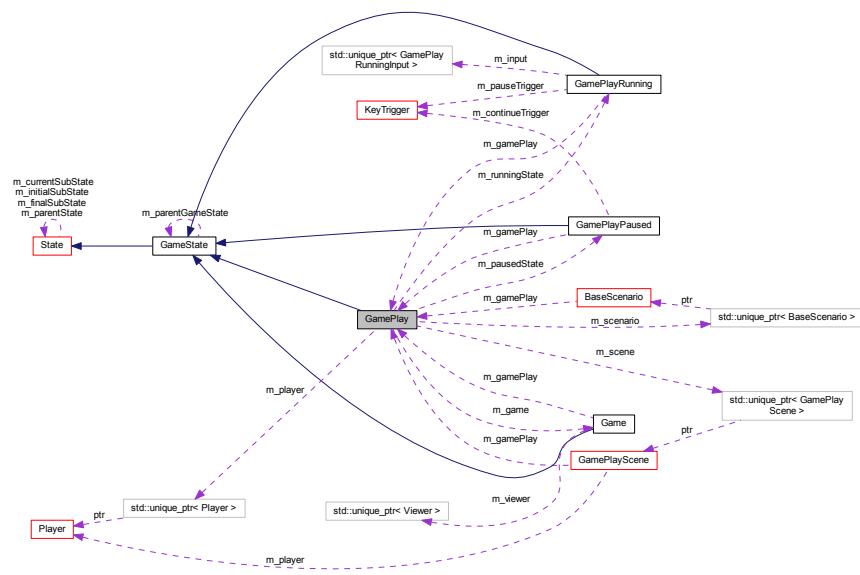
6.87 GamePlay Class Reference

```
#include <gameplay.h>
```

Inheritance diagram for GamePlay:



Collaboration diagram for GamePlay:



Public Member Functions

- **GamePlay (Game *game)**
- **Game * game ()**
- **GamePlayScene & scene ()**
- **GamePlayRunning & running ()**
- **GamePlayPaused & paused ()**
- virtual const Scene & scene () const override
- virtual const CameraHead & cameraHead () const override
- **Player & player ()**
- **SoundManager & soundManager ()**
- void **loadScenario (int i)**
- virtual void **update (float deltaSec)** override
- virtual void **onEntered ()** override
- virtual void **onLeft ()** override

Protected Attributes

- **Game * m_game**
- std::unique_ptr<Player> **m_player**
- std::unique_ptr<GamePlayScene> **m_scene**
- std::unique_ptr<BaseScenario> **m_scenario**
- std::shared_ptr<SoundManager> **m_soundManager**
- **GamePlayRunning * m_runningState**
- **GamePlayPaused * m_pausedState**

Additional Inherited Members

6.87.1 Detailed Description

State that is active whenever the the game is actually played and not in some menustate etc.

6.87.2 Member Function Documentation

6.87.2.1 `void GamePlay::onEntered() [override], [virtual]`

Overrideable method that is called whenever a state or any of its substates come to be currentSubState This happens recursively up to the root-state

Reimplemented from [GameState](#).

6.87.2.2 `void GamePlay::onLeft() [override], [virtual]`

Overrideable method that is called whenever a state ceases to be currentSubState This happens recursively up to the root-state

Reimplemented from [GameState](#).

6.87.2.3 `void GamePlay::update(float deltaSec) [override], [virtual]`

Performs a [Transition](#) from the currentSubState, if such isPossible()

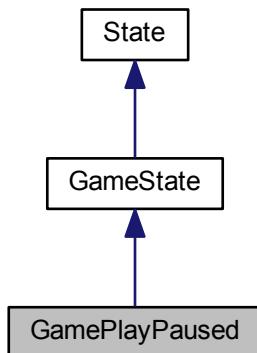
Reimplemented from [GameState](#).

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

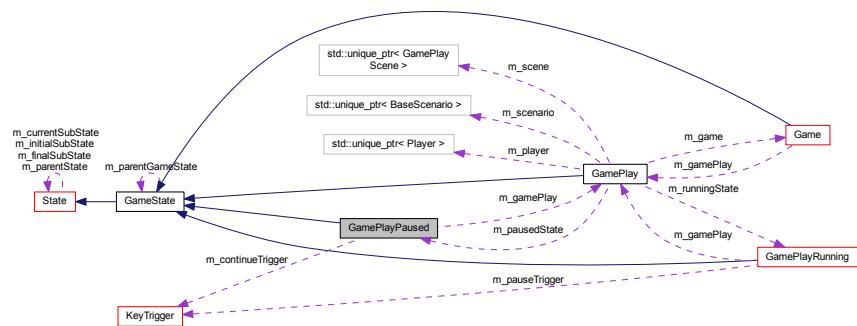
- `src/gamestate/gameplay/gameplay.h`
- `src/gamestate/gameplay/gameplay.cpp`

6.88 GamePlayPaused Class Reference

Inheritance diagram for GamePlayPaused:



Collaboration diagram for GamePlayPaused:



Public Member Functions

- **GamePlayPaused** (`GamePlay *gamePlay`)
 - **Trigger & continueTrigger ()**
 - virtual void `update` (float `deltaSec`) override
 - virtual void `onEntered` () override
 - virtual void `onLeft` () override

Protected Attributes

- GamePlay * m_gamePlay
 - KeyTrigger m_continueTrigger

Additional Inherited Members

6.88.1 Member Function Documentation

6.88.1.1 void GamePlayPaused::onEntered() [override], [virtual]

Overrideable method that is called whenever a state or any of its substates come to be currentSubState This happens recursively up to the root-state

Reimplemented from [GameState](#).

6.88.1.2 void GamePlayPaused::onLeft() [override], [virtual]

Overrideable method that is called whenever a state ceases to be currentSubState This happens recursively up to the root-state

Reimplemented from [GameState](#).

6.88.1.3 void GamePlayPaused::update (float deltaSec) [override], [virtual]

Performs a [Transition](#) from the currentSubState, if such isPossible()

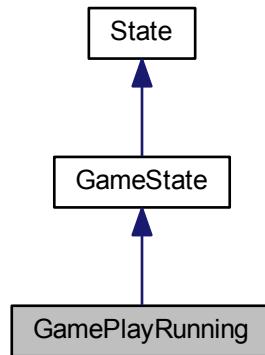
Reimplemented from [GameState](#).

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

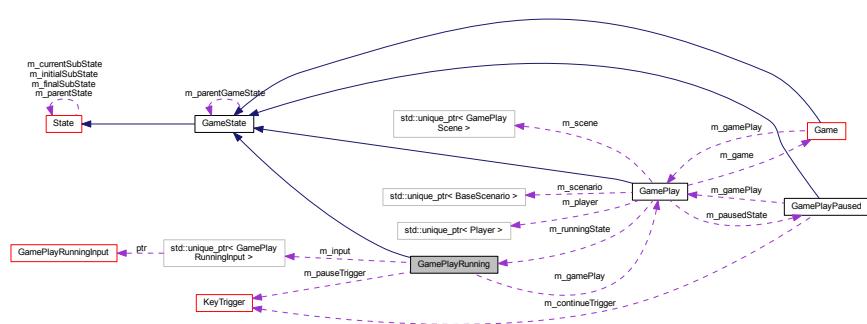
- src/gamestate/gameplay/paused/gameplaypaused.h
- src/gamestate/gameplay/paused/gameplaypaused.cpp

6.89 GamePlayRunning Class Reference

Inheritance diagram for GamePlayRunning:



Collaboration diagram for GamePlayRunning:



Public Member Functions

- GamePlayRunning** (`GamePlay` *`gamePlay`)
- GamePlayRunningInput** & `input()`
- Trigger** & `pauseTrigger()`
- virtual void `update` (float deltaSec) override
- virtual void `onEntered()` override
- virtual void `onLeft()` override

Protected Attributes

- GamePlay** * `m_gamePlay`

- KeyTrigger **m_pauseTrigger**
- std::unique_ptr
< GamePlayRunningInput > **m_input**

Additional Inherited Members

6.89.1 Member Function Documentation

6.89.1.1 void GamePlayRunning::onEntered() [override], [virtual]

Overrideable method that is called whenever a state or any of its substates come to be currentSubState This happens recursively up to the root-state

Reimplemented from [GameState](#).

6.89.1.2 void GamePlayRunning::onLeft() [override], [virtual]

Overrideable method that is called whenever a state ceases to be currentSubState This happens recursively up to the root-state

Reimplemented from [GameState](#).

6.89.1.3 void GamePlayRunning::update(float deltaSec) [override], [virtual]

Performs a [Transition](#) from the currentSubState, if such isPossible()

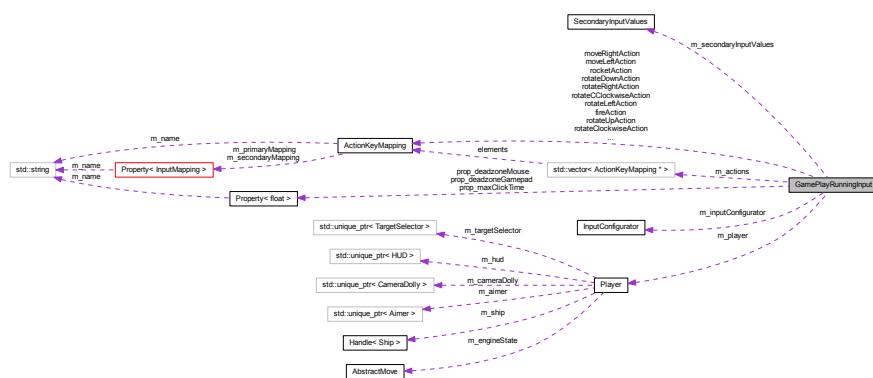
Reimplemented from [GameState](#).

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

- src/gamestate/gameplay/running/gameplayrunning.h
- src/gamestate/gameplay/running/gameplayrunning.cpp

6.90 GamePlayRunningInput Class Reference

Collaboration diagram for GamePlayRunningInput:



Public Member Functions

- **GamePlayRunningInput** ([Player](#) *player)

- void **resizeEvent** (const unsigned int width, const unsigned int height)
- void **keyCallback** (int key, int scanCode, int action, int mods)
- void **mouseButtonCallback** (int button, int action, int mods)
- void **update** (float deltaSec)

Protected Member Functions

- void **toggleControls** ()
- void **processUpdate** ()
- void **processMouseUpdate** (float deltaSec)
- void **processHMDUpdate** ()
- void **applyUpdates** ()
- void **processFireActions** ()
- void **processMoveActions** ()
- void **processRotateActions** ()
- void **processTargetSelectActions** ()
- float **getInputValue** ([ActionKeyMapping](#) *action)
- float **getInputValue** ([InputMapping](#) mapping)
- void **addActionToVector** ()
- void **setupJoystickControls** ()
- void **retrieveInputValues** ()
- void **placeCrossHair** (double winX, double winY)

Protected Attributes

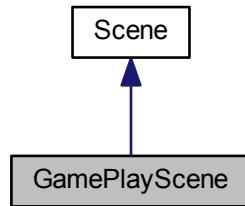
- [Player](#) * **m_player**
- [InputConfigurator](#) * **m_inputConfigurator**
- [SecondaryInputValues](#) **m_secondaryInputValues**
- std::vector< [ActionKeyMapping](#) * > **m_actions**
- bool **m_mouseControl**
- int **m_cursorMaxDistance**
- int **m_lastfocus**
- float **m_currentTimePressed**
- [Property](#)< float > **prop_deadzoneMouse**
- [Property](#)< float > **prop_deadzoneGamepad**
- [Property](#)< float > **prop_maxClickTime**
- [ActionKeyMapping](#) **fireAction**
- [ActionKeyMapping](#) **rocketAction**
- [ActionKeyMapping](#) **moveLeftAction**
- [ActionKeyMapping](#) **moveRightAction**
- [ActionKeyMapping](#) **moveForwardAction**
- [ActionKeyMapping](#) **moveBackwardAction**
- [ActionKeyMapping](#) **rotateLeftAction**
- [ActionKeyMapping](#) **rotateRightAction**
- [ActionKeyMapping](#) **rotateUpAction**
- [ActionKeyMapping](#) **rotateDownAction**
- [ActionKeyMapping](#) **rotateClockwiseAction**
- [ActionKeyMapping](#) **rotateCClockwiseAction**
- [ActionKeyMapping](#) **selectNextAction**
- [ActionKeyMapping](#) **selectPreviousAction**
- [glm::vec3](#) **m_moveUpdate**
- [glm::vec3](#) **m_rotateUpdate**
- bool **m_fireUpdate**
- bool **m_rocketUpdate**

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

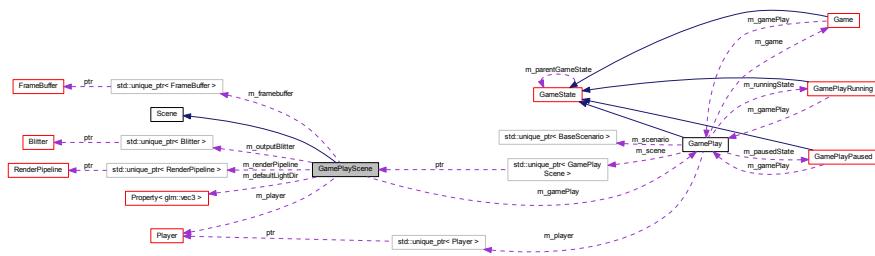
- src/gamestate/gameplay/running/gameplayrunninginput.h
 - src/gamestate/gameplay/running/gameplayrunninginput.cpp

6.91 GamePlayScene Class Reference

Inheritance diagram for GamePlayScene:



Collaboration diagram for GamePlayScene:



Public Member Functions

- **GamePlayScene** (`GamePlay &gamePlay, Player &player`)
 - `virtual void draw (const Camera &camera, glow::FrameBufferObject *target, const Viewport &destinationViewport, EyeSide side=EyeSide::None) const override`
 - `virtual void update (float deltaSec) override`
 - `void setOutputBuffer (int i)`

Protected Member Functions

- void **drawGame** (const Camera &camera) const

Protected Attributes

- std::unique_ptr< Blitter > m_outputBlitter
 - std::unique_ptr< RenderPipeline > m renderPipeline

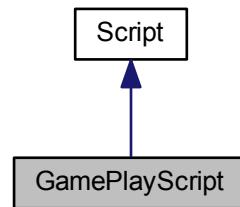
- std::unique_ptr< FrameBuffer > **m_framebuffer**
- std::shared_ptr< VoxelRenderer > **m_voxelRenderer**
- std::shared_ptr< Starfield > **m_starField**
- **GamePlay** & **m_gamePlay**
- **Player** & **m_player**
- **Property**< glm::vec3 > **m_defaultLightDir**
- int **m_currentOutputBuffer**

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

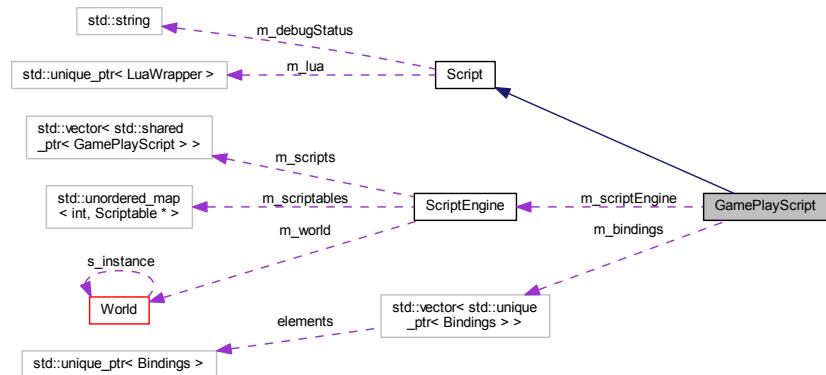
- src/gamestate/gameplay/gameplayscene.h
- src/gamestate/gameplay/gameplayscene.cpp

6.92 GamePlayScript Class Reference

Inheritance diagram for GamePlayScript:



Collaboration diagram for GamePlayScript:



Public Member Functions

- **GamePlayScript** (**ScriptEngine** *scriptEngine)

- void **initializeBindings ()**
- **ScriptEngine & scriptEngine ()**
- **LuaWrapper & luaWrapper ()**

Protected Attributes

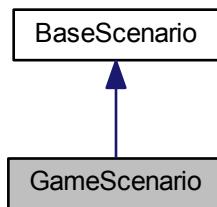
- **ScriptEngine * m_scriptEngine**
- std::vector< std::unique_ptr< Bindings > > **m_bindings**

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

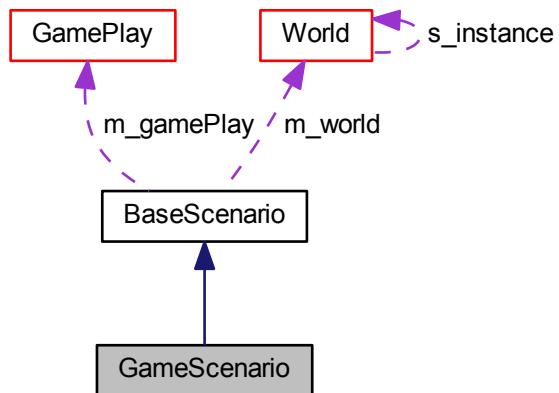
- src/scripting/gameplayscript.h
- src/scripting/gameplayscript.cpp

6.93 GameScenario Class Reference

Inheritance diagram for GameScenario:



Collaboration diagram for GameScenario:



Public Member Functions

- **GameScenario** ([GamePlay *inGame](#))

Protected Member Functions

- virtual void **populateWorld** () override
- void **createArmada** ()

Additional Inherited Members

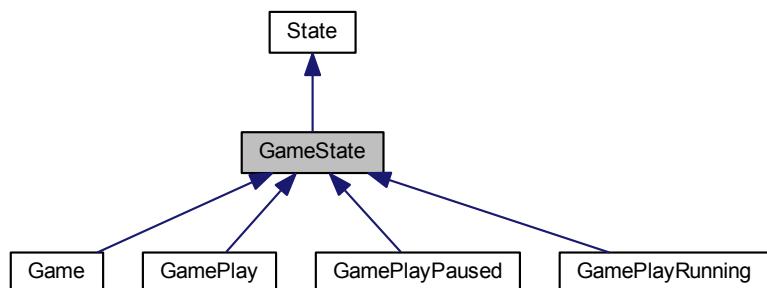
The documentation for this class was generated from the following files:

- [src/scenarios/gamescenario.h](#)
- [src/scenarios/gamescenario.cpp](#)

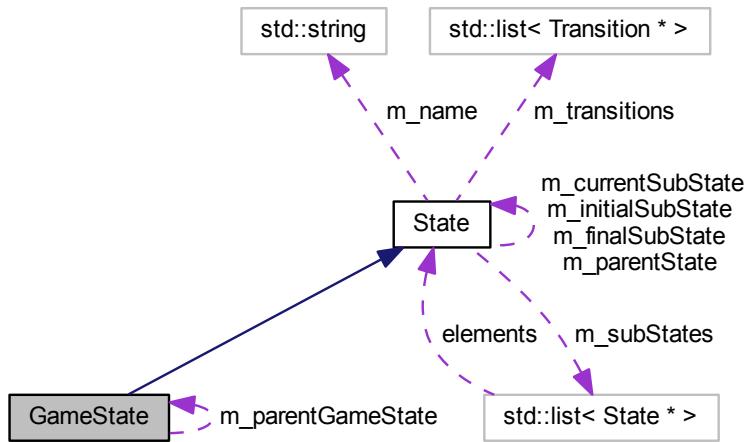
6.94 GameState Class Reference

```
#include <gamestate.h>
```

Inheritance diagram for GameState:



Collaboration diagram for GameState:



Public Member Functions

- `GameState` (const std::string &name, `GameState` *parent)
- `GameState` * `parentGameState` ()
- virtual const `Scene` & `scene` () const
- virtual const `CameraHead` & `cameraHead` () const
- virtual void `update` (float deltaSec) override
- virtual void `onEntered` () override
- virtual void `onLeft` () override

Protected Attributes

- `GameState` * `m_parentGameState`

Additional Inherited Members

6.94.1 Detailed Description

Base class for a `State` the game can be in - for example Menu, OptionsMenu, normal Gameplay or ingame menu

A `GameState` has to provide a `Scene` and a `CameraHead` so that it can be displayed by the `Viewer` of `Game`. Per default these 2 return their parents `Scene` and `CameraHead` so that GameStates can be arbitrarily nested

6.94.2 Member Function Documentation

6.94.2.1 void GameState::onEntered () [override], [virtual]

Overrideable method that is called whenever a state or any of its substates come to be currentSubState This happens recursively up to the root-state

Reimplemented from `State`.

Reimplemented in `GamePlay`, `GamePlayRunning`, and `GamePlayPaused`.

6.94.2.2 void GameState::onLeft() [override], [virtual]

Overrideable method that is called whenever a state ceases to be currentSubState This happens recursively up to the root-state

Reimplemented from [State](#).

Reimplemented in [GamePlay](#), [GamePlayRunning](#), and [GamePlayPaused](#).

6.94.2.3 void GameState::update(float deltaSec) [override], [virtual]

Performs a [Transition](#) from the currentSubState, if such isPossible()

Reimplemented from [State](#).

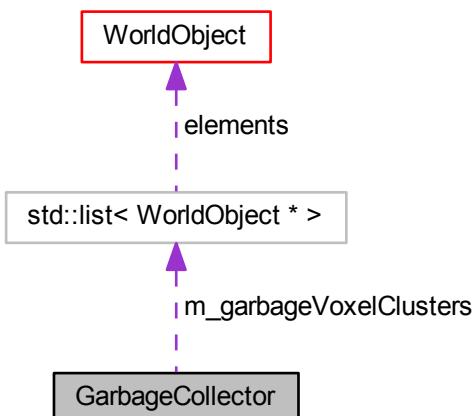
Reimplemented in [GamePlay](#), [Game](#), [GamePlayRunning](#), and [GamePlayPaused](#).

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

- src/gamestate/gamestate.h
- src/gamestate/gamestate.cpp

6.95 GarbageCollector Class Reference

Collaboration diagram for GarbageCollector:



Public Member Functions

- void **check** (std::unordered_set<[WorldObject](#) * > &modifiedVoxelClusters)
- std::list<[WorldObject](#) * > & **garbageVoxelClusters** ()

Protected Attributes

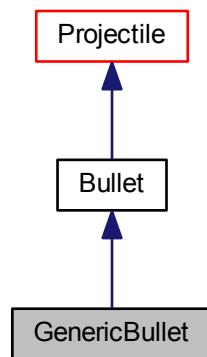
- std::list<[WorldObject](#) * > **m_garbageVoxelClusters**

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

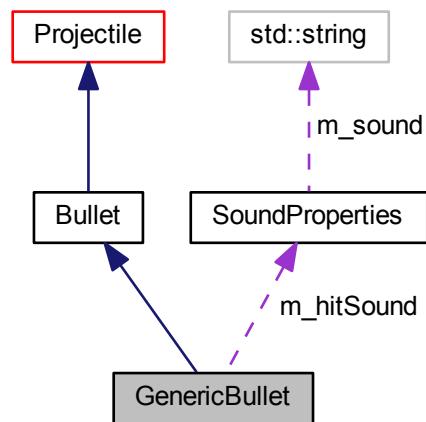
- src/world/handler/garbagecollector.h
- src/world/handler/garbagecollector.cpp

6.96 GenericBullet Class Reference

Inheritance diagram for GenericBullet:



Collaboration diagram for GenericBullet:



Public Member Functions

- virtual float **emissiveness () const override**

- void **setEmissiveness** (float emissiveness)
- virtual const [SoundProperties](#) & **hitSound** () const override
- void **setHitSound** (const [SoundProperties](#) &hitSound)

Protected Member Functions

- virtual void **spawnExplosion** () override

Protected Attributes

- float **m_emissiveness**
- [SoundProperties](#) **m_hitSound**

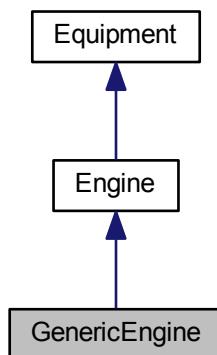
Additional Inherited Members

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

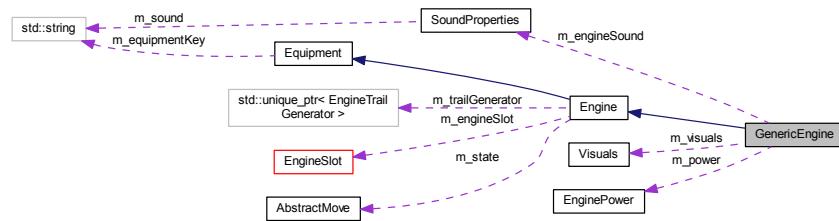
- src/equipment/weapons/genericbullet.h
- src/equipment/weapons/genericbullet.cpp

6.97 GenericEngine Class Reference

Inheritance diagram for GenericEngine:



Collaboration diagram for GenericEngine:



Public Member Functions

- **GenericEngine** (const std::string &equipmentKey)
- virtual const **Visuals** & **visuals** () const override
- void **setVisuals** (const **Visuals** &visuals)
- virtual const **SoundProperties** & **sound** () const override
- void **setEngineSound** (const **SoundProperties** &engineSound)
- virtual **EnginePower** **power** () const override
- void **setPower** (const **EnginePower** &power)
- virtual void **update** (float deltaSec) override

Protected Attributes

- **EnginePower** **m_power**
- **Visuals** **m_visuals**
- **SoundProperties** **m_engineSound**

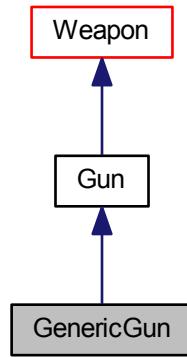
Additional Inherited Members

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

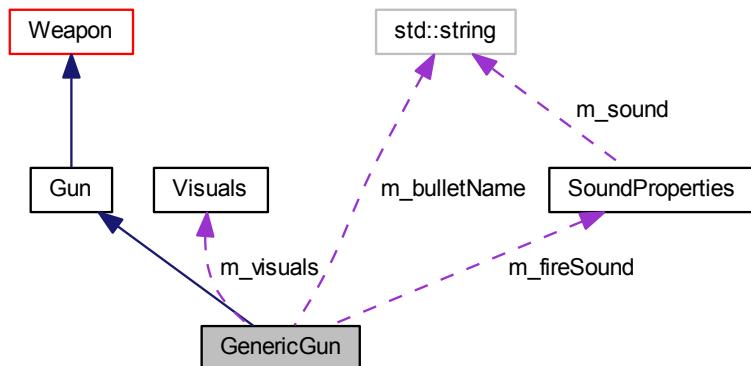
- src/equipment/engines/genericengine.h
- src/equipment/engines/genericengine.cpp

6.98 GenericGun Class Reference

Inheritance diagram for GenericGun:



Collaboration diagram for GenericGun:



Public Member Functions

- **GenericGun** (const std::string &name)
- virtual float **bulletLifetime** () const override
- void **setBulletLifetime** (float bulletLifetime)
- virtual float **bulletSpeed** () const override
- void **setBulletSpeed** (float bulletSpeed)
- virtual const **Visuals** & **visuals** () const override
- void **setVisuals** (const **Visuals** &visuals)
- virtual const **SoundProperties** & **fireSound** () const override
- void **setFireSound** (const **SoundProperties** &fireSound)

- virtual float **cooldownTime** () const override
- void **setCooldownTime** (float cooldownTime)
- const std::string & **bulletName** () const
- void **setBulletName** (const std::string &bulletName)

Protected Member Functions

- virtual Bullet * **createBullet** () override

Protected Attributes

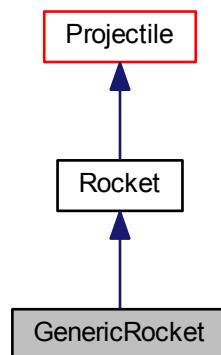
- float **m_bulletSpeed**
- float **m_bulletLifetime**
- float **m_cooldownTime**
- Visuals **m_visuals**
- SoundProperties **m_fireSound**
- std::string **m_bulletName**

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

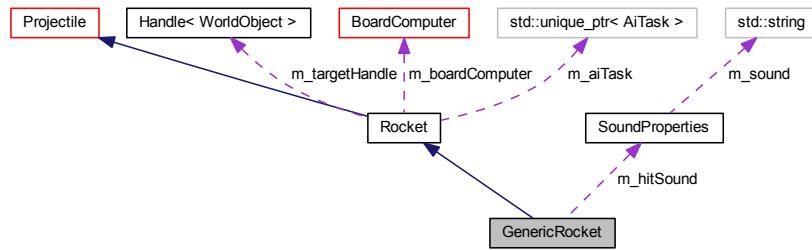
- src/equipment/weapons/genericgun.h
- src/equipment/weapons/genericgun.cpp

6.99 GenericRocket Class Reference

Inheritance diagram for GenericRocket:



Collaboration diagram for GenericRocket:



Public Member Functions

- virtual const `SoundProperties` & `hitSound` () const override
- void `setHitSound` (const `SoundProperties` &hitSound)

Protected Member Functions

- virtual void `spawnExplosion` () override
- virtual void `onLifetimeOver` () override

Protected Attributes

- `SoundProperties m_hitSound`

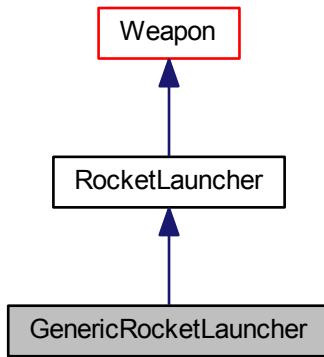
Additional Inherited Members

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

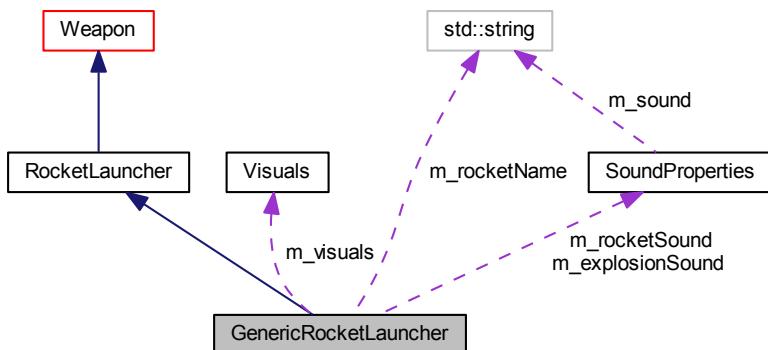
- src/equipment/weapons/genericrocket.h
- src/equipment/weapons/genericrocket.cpp

6.100 GenericRocketLauncher Class Reference

Inheritance diagram for GenericRocketLauncher:



Collaboration diagram for GenericRocketLauncher:



Public Member Functions

- **GenericRocketLauncher** (const std::string &name)
- virtual const **Visuals** & **visuals** () const override
- void **setVisuals** (const **Visuals** &visuals)
- virtual float **cooldownTime** () const override
- void **setCooldownTime** (float cooldownTime)
- const **Rocket** * **rocketPrototype** () const
- void **setRocketPrototype** (**Rocket** *rocketPrototype)
- const std::string & **rocketName** () const
- void **setRocketName** (const std::string &rocketName)

Protected Member Functions

- virtual `Rocket * createRocket ()` override

Protected Attributes

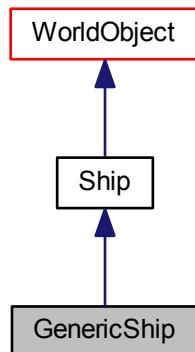
- float `m_cooldownTime`
- `SoundProperties m_rocketSound`
- `SoundProperties m_explosionSound`
- `Visuals m_visuals`
- `std::string m_rocketName`

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

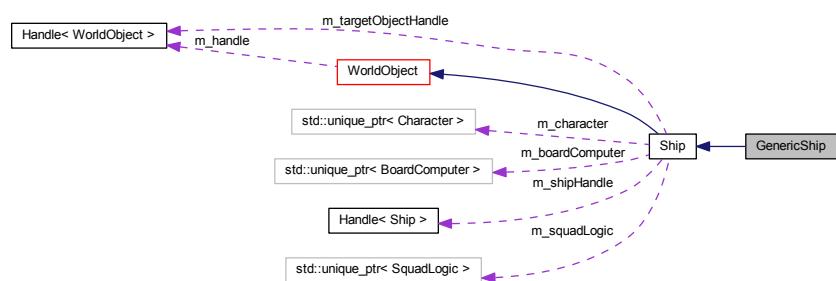
- `src/equipment/weapons/genericrocketlauncher.h`
- `src/equipment/weapons/genericrocketlauncher.cpp`

6.101 GenericShip Class Reference

Inheritance diagram for GenericShip:



Collaboration diagram for GenericShip:



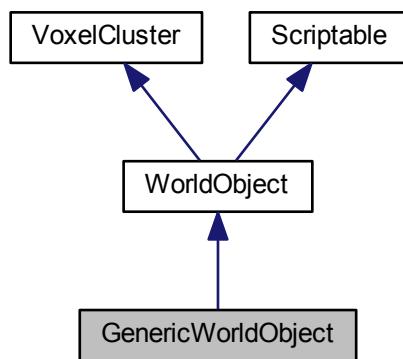
Additional Inherited Members

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

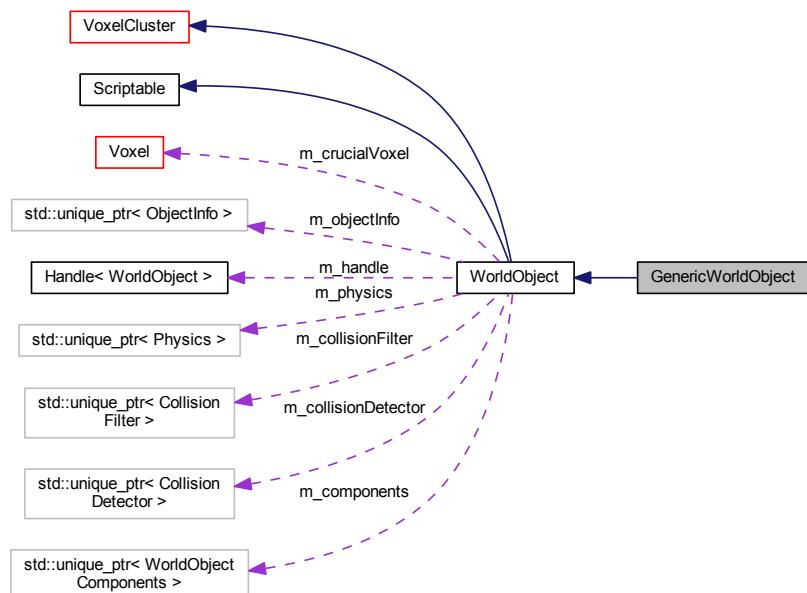
- src/worldobject/genericship.h
- src/worldobject/genericship.cpp

6.102 GenericWorldObject Class Reference

Inheritance diagram for GenericWorldObject:



Collaboration diagram for GenericWorldObject:



Additional Inherited Members

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

- src/worldobject/genericworldobject.h
- src/worldobject/genericworldobject.cpp

6.103 GeometryHelper Class Reference

Static Public Member Functions

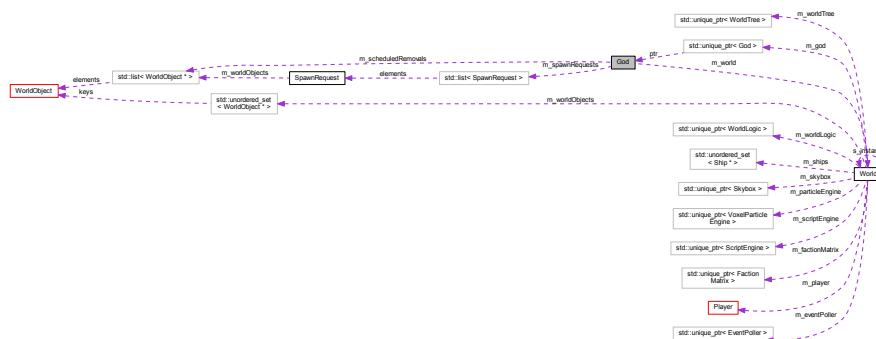
- static bool **intersectRectangle** (const [Ray](#) *ray, const [glm::vec3](#) &p, const [glm::vec3](#) &q, const [glm::vec3](#) &r, const [glm::vec3](#) &s)
- static bool **intersectRectangle** (const [Ray](#) *ray, const [glm::vec3](#) &p, const [glm::vec3](#) &q, const [glm::vec3](#) &r, const [glm::vec3](#) &s, [glm::vec3](#) &intersection)
- static [glm::vec3](#) **plane** (const [glm::vec3](#) &p, const [glm::vec3](#) &q, const [glm::vec3](#) &r)
- static float **angleBetween** (const [glm::vec3](#) &u, const [glm::vec3](#) &v)
- static float **angleBetweenVectorPlane** (const [glm::vec3](#) &u, const [glm::vec3](#) &v)
- static [glm::quat](#) **quatFromTo** (const [glm::vec3](#) &u, const [glm::vec3](#) &v)
- static [glm::quat](#) **quatFromViewDirection** (const [glm::vec3](#) &dir)
- static [WorldObject](#) * **closestObject** ([WorldObject](#) &self, std::unordered_set<[WorldObject](#) * > *objects)
- template<typename T >
static T **safeNormalize** (const T &value)

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

- src/utils/geometryhelper.h
- src/utils/geometryhelper.cpp
- src/utils/geometryhelper.inl

6.104 God Class Reference

Collaboration diagram for God:



Public Member Functions

- God** ([World](#) &world)
- void **scheduleSpawn** ([SpawnRequest](#) spawnRequest)

- void **scheduleRemoval** (`WorldObject` *worldObject)
- void **scheduleRemovals** (const std::list< `WorldObject` * > &removals)
- void **spawn** ()
- void **remove** ()

Protected Attributes

- `World` & **m_world**
- std::list< `SpawnRequest` > **m_spawnRequests**
- std::list< `WorldObject` * > **m_scheduledRemovals**

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

- src/world/god.h
- src/world/god.cpp

6.105 GridAABB Class Reference

Public Member Functions

- **GridAABB** (const `glm::ivec3` &llf, const `glm::ivec3` &rub)
- const `glm::ivec3` & **llf** () const
- void **setLLF** (const `glm::ivec3` &llf)
- const `glm::ivec3` & **rub** () const
- void **setRUB** (const `glm::ivec3` &rub)
- bool **contains** (const `glm::ivec3` &cell) const
- int **extent** (Axis axis) const
- float **diameter** () const
- bool **operator==** (const `GridAABB` &other) const

Protected Attributes

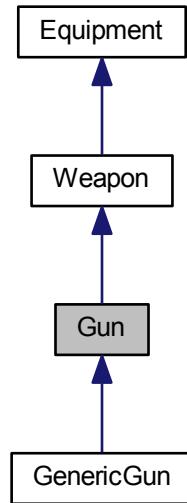
- `glm::ivec3` **m_llf**
- `glm::ivec3` **m_rub**

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

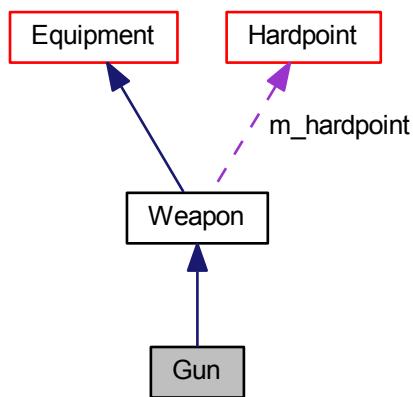
- src/geometry/gridaabb.h
- src/geometry/gridaabb.cpp

6.106 Gun Class Reference

Inheritance diagram for Gun:



Collaboration diagram for Gun:



Public Member Functions

- **Gun** (const std::string &equipmentKey)
- virtual const **SoundProperties** & **fireSound** () const =0
- virtual float **bulletLifetime** () const =0

- virtual float **bulletSpeed** () const =0
- virtual void **fireAtPoint** (const glm::vec3 &point)
- virtual void **update** (float deltaSec) override

Protected Member Functions

- virtual **Bullet** * **createBullet** ()=0
- void **setupBullet** (**Bullet** *bullet, const glm::vec3 &point)

Additional Inherited Members

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

- src/equipment/weapons/gun.h
- src/equipment/weapons/gun.cpp

6.107 Handle< T > Class Template Reference

Public Member Functions

- **Handle** (T *object)
- T * **get** ()
- const T * **get** () const
- T * **operator->** ()
- const T * **operator->** () const
- T * **operator*** ()
- const T * **operator*** () const
- bool **valid** () const
- void **invalidate** ()

Protected Attributes

- std::shared_ptr< **HandleImpl**< T > > **m_impl**

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

- src/utils/handle/handle.h
- src/utils/handle/handle.inl

6.108 HandleImpl< T > Class Template Reference

Public Member Functions

- **HandleImpl** (T *object)
- T * **get** ()
- bool **valid** () const
- void **invalidate** ()

Protected Attributes

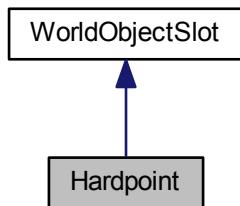
- `T * m_object`

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

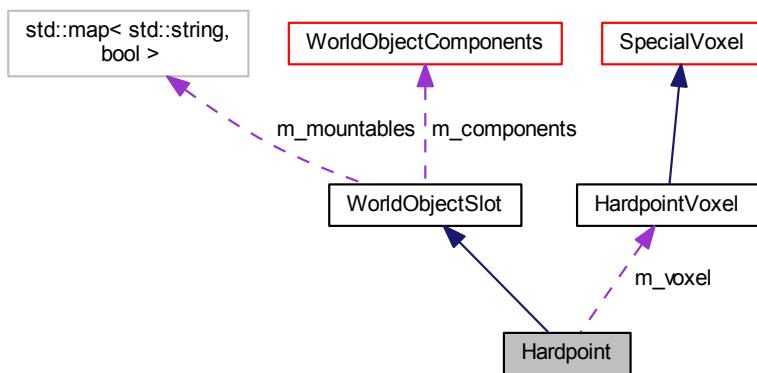
- `src/utils/handle/handle.h`
- `src/utils/handle/handleimpl.h`
- `src/utils/handle/handleimpl.inl`

6.109 Hardpoint Class Reference

Inheritance diagram for Hardpoint:



Collaboration diagram for Hardpoint:



Public Member Functions

- `Hardpoint (WorldObjectComponents *components, HardpointVoxel *voxel)`
- `HardpointVoxel * voxel ()`
- `const std::shared_ptr< Weapon > & weapon ()`

- void **setWeapon** (const std::shared_ptr< Weapon > &weapon)
- const glm::vec3 & **direction** () const
- void **setDirection** (const glm::vec3 &direction)
- float **fieldOfAim** () const
- void **setFieldOfAim** (float fieldOfAim)
- bool **inFieldOfAim** (const glm::vec3 &point)
- void **update** (float deltaSec)
- void **onVoxelRemoval** ()

Protected Attributes

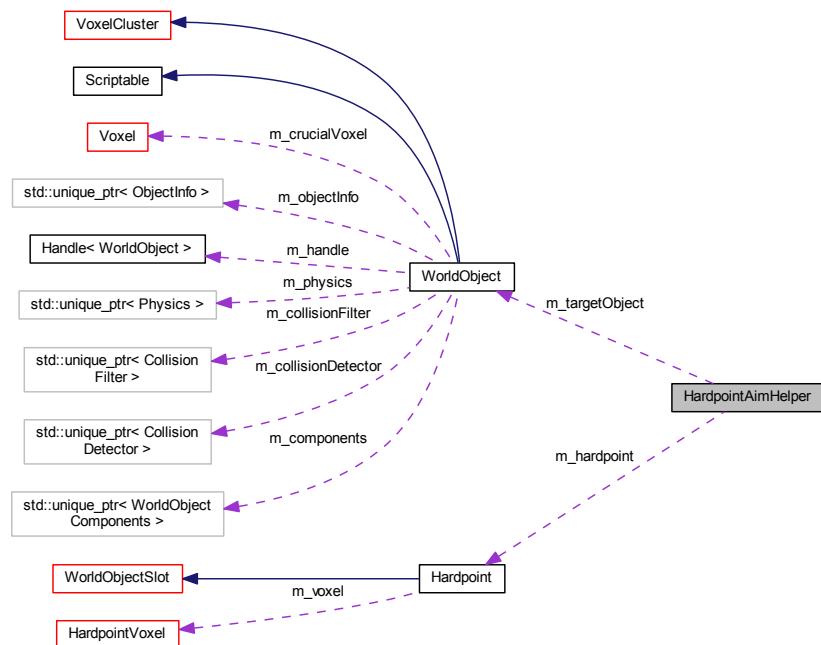
- HardpointVoxel * **m voxel**
- std::shared_ptr< Weapon > **m weapon**
- glm::vec3 **m direction**
- float **m fieldOfAim**

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

- src/equipment/hardpoint.h
- src/equipment/hardpoint.cpp

6.110 HardpointAimHelper Class Reference

Collaboration diagram for HardpointAimHelper:



Public Member Functions

- **HardpointAimHelper** (**Hardpoint** *hardpoint, **WorldObject** *targetObject)

- void **aim**()
- bool **isHitable**()
- const glm::vec3 & **direction**()
- const glm::vec3 & **point**()

Protected Member Functions

- float **bulletTravelTime**(const glm::vec3 &point)
- glm::vec3 **targetPositionIn**(float deltaSec)
- float **bulletSpeedInDirection**(const glm::vec3 &direction)

Protected Attributes

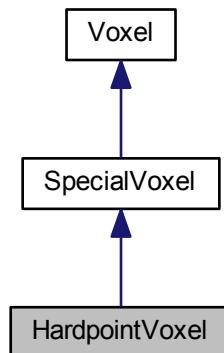
- [Hardpoint](#) * **m_hardpoint**
- [WorldObject](#) * **m_targetObject**
- bool **m_aimed**
- glm::vec3 **m_hardpointPosition**
- glm::vec3 **m_targetPosition**
- glm::vec3 **m_targetSpeed**
- float **m_bulletSpeed**
- float **m_bulletLifetime**
- bool **m_hitable**
- glm::vec3 **m_direction**
- glm::vec3 **m_point**

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

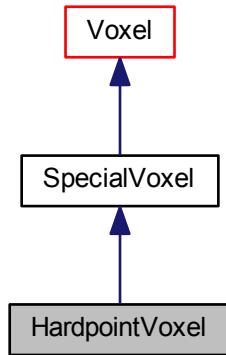
- src/worldobject/helper/hardpointaimhelper.h
- src/worldobject/helper/hardpointaimhelper.cpp

6.111 HardpointVoxel Class Reference

Inheritance diagram for HardpointVoxel:



Collaboration diagram for HardpointVoxel:



Public Member Functions

- **HardpointVoxel** (const glm::ivec3 &gridCell, int index)
- virtual **Visuals visuals** () const override
- virtual void **addToObject** (WorldObject *object) override
- virtual void **onRemoval** () override
- virtual void **onDestruction** () override

Protected Attributes

- std::shared_ptr< Hardpoint > **m_hardpoint**

Additional Inherited Members

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

- src/voxel/specialvoxels/hardpointvoxel.h
- src/voxel/specialvoxels/hardpointvoxel.cpp

6.112 std::hash< glm::ivec3 > Struct Template Reference

Public Member Functions

- std::size_t **operator()** (const glm::ivec3 &v) const

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

- src/utils/vec3hash.h

6.113 std::hash< pair< A, B > > Struct Template Reference

Public Member Functions

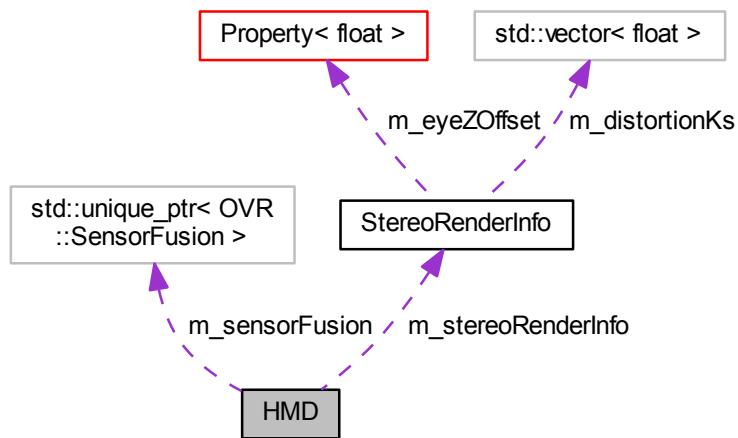
- `size_t operator()(const pair< A, B > &arg) const`

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

- `src/utils/pairhash.h`

6.114 HMD Class Reference

Collaboration diagram for HMD:



Public Member Functions

- `HMD (OVR::HMDDevice *hmdDevice)`
- `glm::quat orientation ()`
- `const StereoRenderInfo & stereoRenderInfo () const`

Protected Attributes

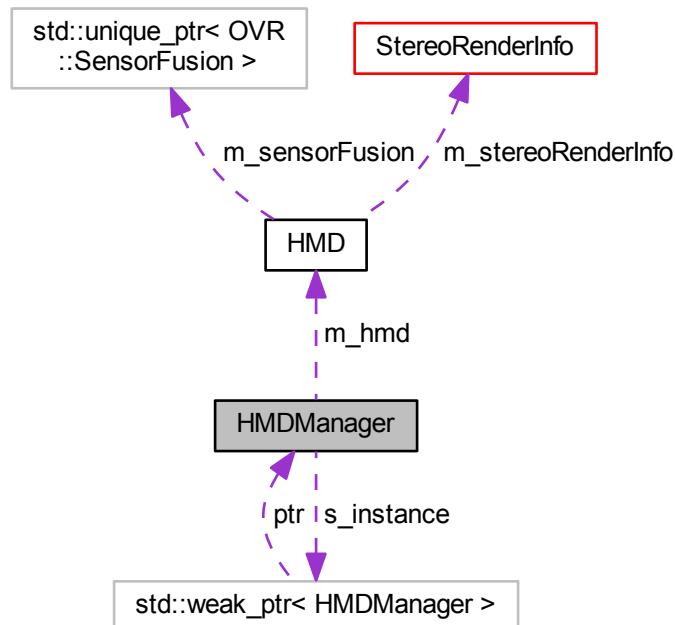
- `std::unique_ptr< OVR::SensorFusion > m_sensorFusion`
- `OVR::HMDDevice * m_hmdDevice`
- `OVR::SensorDevice * m_sensorDevice`
- `StereoRenderInfo m_stereoRenderInfo`

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

- `src/etc/hmd/hmd.h`
- `src/etc/hmd/hmd.cpp`

6.115 HMDManager Class Reference

Collaboration diagram for HMDManager:



Public Member Functions

- void `setupHMD` (`Viewer` &viewer)
- `HMD * hmd ()`

Static Public Member Functions

- static `std::shared_ptr< HMDManager > instance ()`

Protected Attributes

- `HMD * m_hmd`
- `OVR::DeviceManager * m_deviceManager`

Static Protected Attributes

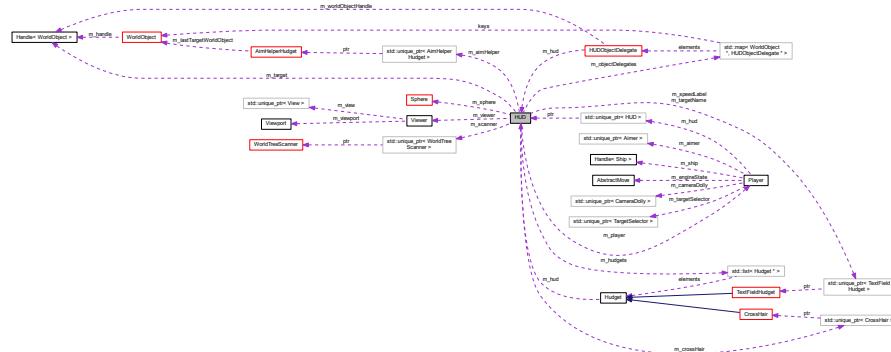
- static `std::weak_ptr< HMDManager > s_instance`

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

- `src/etc/hmd/hmdmanager.h`
- `src/etc/hmd/hmdmanager.cpp`

6.116 HUD Class Reference

Collaboration diagram for HUD:



Public Member Functions

- `HUD (Player *player)`
- `Player * player ()`
- `const Sphere & sphere () const`
- `CrossHair & crossHair ()`
- `AimHelperBudget & aimHelper ()`
- `glm::vec3 centerOfView () const`
- `glm::vec3 position () const`
- `glm::quat orientation () const`
- `void addBudget (Budget *budget)`
- `void removeBudget (Budget *budget)`
- `void addObjectDelegate (HUDOObjectDelegate *objectDelegate)`
- `void removeObjectDelegate (HUDOObjectDelegate *objectDelegate)`
- `HUDOObjectDelegate * objectDelegate (WorldObject *worldObject)`
- `void setCrossHairOffset (const glm::vec2 &mousePosition)`
- `void setTarget (WorldObject *target)`
- `WorldObject * target ()`
- `void onClick (ClickType clickType)`
- `void update (float deltaSec)`
- `void draw ()`
- `glm::vec3 applyTo (const glm::vec3 &vertex) const`
- `Viewer * viewer () const`
- `void setViewer (Viewer &viewer)`
- `float fovy () const`
- `float fovx () const`

Protected Member Functions

- `void updateScanner (float deltaSec)`
- `void updateFov ()`

Protected Attributes

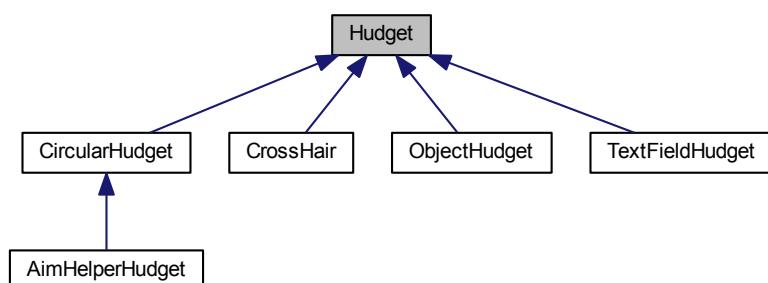
- `Player * m_player`
- `Viewer * m_viewer`
- `Sphere m_sphere`
- `Handle< WorldObject > m_target`
- `float m_fovy`
- `float m_fovx`
- `std::unique_ptr< AimHelperBudget > m_aimHelper`
- `std::unique_ptr< CrossHair > m_crossHair`
- `std::unique_ptr< WorldTreeScanner > m_scanner`
- `std::unique_ptr< TextFieldBudget > m_speedLabel`
- `std::unique_ptr< TextFieldBudget > m_targetName`
- `std::list< Hudget * > m_hudgets`
- `std::map< WorldObject *, HUDObjectDelegate * > m_objectDelegates`

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

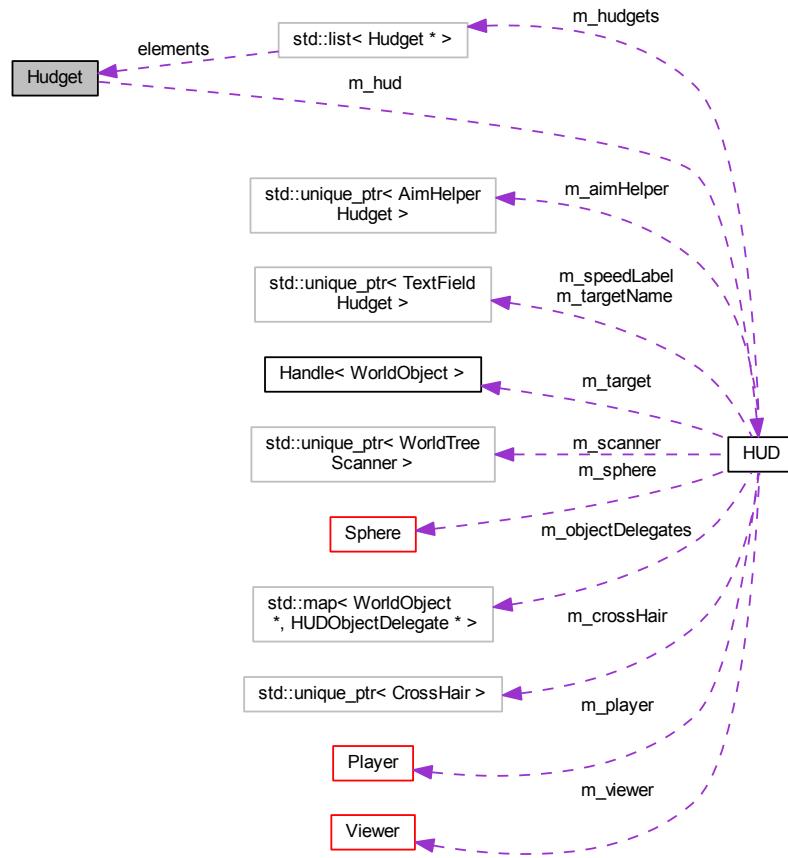
- `src/ui/hud/hud.h`
- `src/ui/hud/hud.cpp`

6.117 Hudget Class Reference

Inheritance diagram for Hudget:



Collaboration diagram for Hudget:



Public Member Functions

- **Hudget (HUD *hud)**
- **HUD * hud ()**
- **bool visible () const**
- **void setVisible (bool visible)**
- **bool pressed () const**
- **bool hovered () const**
- **bool clicked () const**
- **bool released () const**
- **bool entered () const**
- **bool left () const**
- **virtual void onClick (ClickType clickType)**
- **virtual bool isAt (const Ray &ray) const**
- **virtual void pointerAt (const Ray &ray, bool pressed)**
- **void setRelativeDistance (float relativeDistance)**
- **void pointToWorldPoint (const glm::vec3 &worldPoint)**
- **void pointToLocalPoint (const glm::vec3 &localPoint)**
- **glm::vec3 localDirection () const**
- **glm::vec3 worldDirection () const**
- **float directionAngle () const**

- void **setDirectionAngle** (float directionAngle)
- glm::vec3 **worldPosition** () const
- glm::vec3 **worldPosition** (const glm::vec3 &localVector) const
- glm::quat **worldOrientation** () const
- glm::quat **worldOrientation** (const glm::vec3 &localVector) const
- virtual void **update** (float deltaSec)=0
- virtual void **draw** ()=0

Protected Attributes

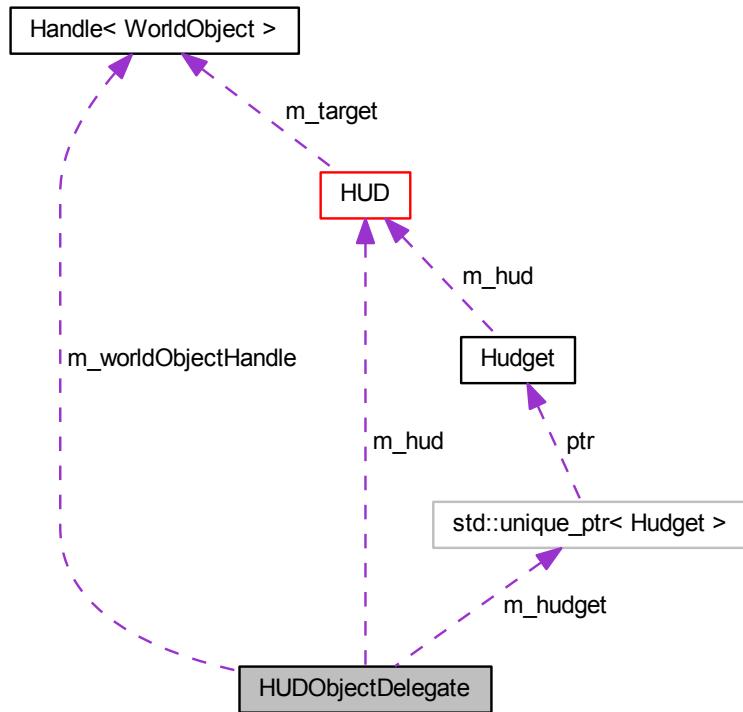
- **HUD * m_hud**
- glm::vec3 **m_direction**
- float **m_directionAngle**
- float **m_relativeDistance**
- bool **m_visible**
- bool **m_pressed**
- bool **m_hovered**
- bool **m_clicked**
- bool **m_released**
- bool **m_entered**
- bool **m_left**

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

- src/ui/hud/hudget.h
- src/ui/hud/hudget.cpp

6.118 HUDObjectDelegate Class Reference

Collaboration diagram for HUDObjectDelegate:



Public Member Functions

- `HUDObjectDelegate (HUD *hud, WorldObject *worldObject)`
- `HUD * hud ()`
- `WorldObject * worldObject ()`
- `Hudget & hudget ()`

Protected Attributes

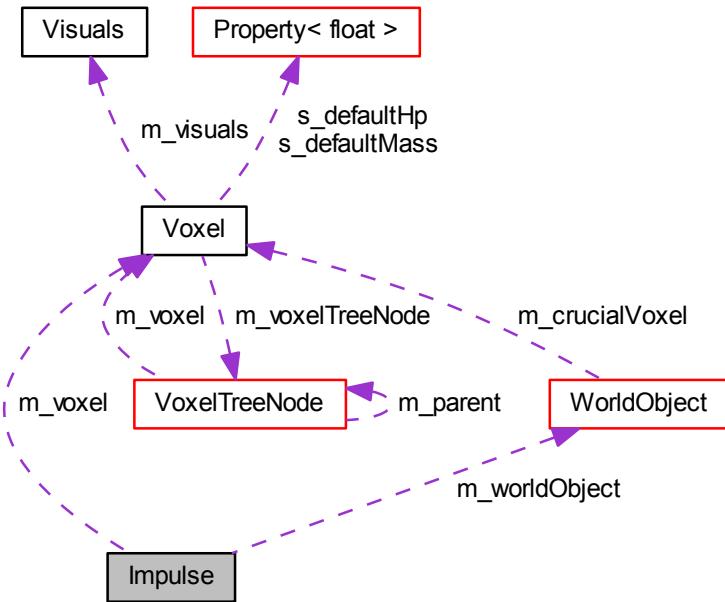
- `HUD * m_hud`
- `Handle< WorldObject > m_worldObjectHandle`
- `std::unique_ptr< Budget > m_hudget`

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

- `src/ui/hud/hudobjectdelegate.h`
- `src/ui/hud/hudobjectdelegate.cpp`

6.119 Impulse Class Reference

Collaboration diagram for Impulse:



Public Member Functions

- **Impulse** (`WorldObject` *`worldObject`, `Voxel` *`voxel`, const `glm::vec3` &`speed`, float `mass`, const `glm::vec3` &`normal`)
- **WorldObject** * `worldObject` ()
- const `WorldObject` * `worldObject` () const
- **Voxel** * `voxel` ()
- const `Voxel` * `voxel` () const
- const `glm::vec3` & `speed` () const
- float `mass` () const
- const `glm::vec3` & `normal` () const
- void `add` (const `Impulse` &`impulse`)

Protected Attributes

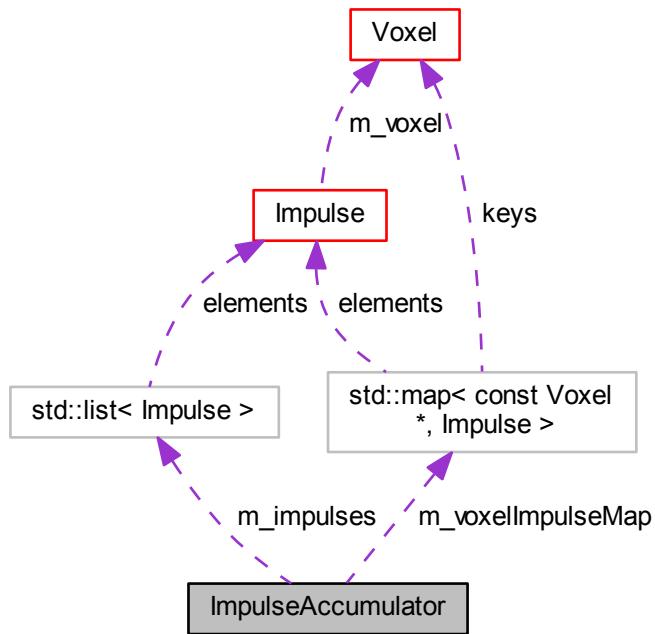
- `WorldObject` * `m_worldObject`
- `Voxel` * `m voxel`
- `glm::vec3` `m speed`
- float `m mass`
- `glm::vec3` `m normal`

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

- `src/physics/impulse.h`
- `src/physics/impulse.cpp`

6.120 ImpulseAccumulator Class Reference

Collaboration diagram for ImpulseAccumulator:



Public Member Functions

- void **clear** ()
- void **parse** (const std::list<**Impulse**> &impulses)
- void **dontImpulse** (const std::list<**Voxel*** > &voxels)
- std::list<**Impulse**> & **impulses** ()

Protected Member Functions

- void **parseCollision** (**WorldObject** *worldObject, const **VoxelCollision** &collision)

Protected Attributes

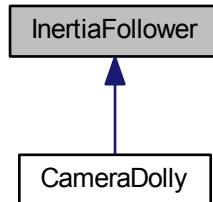
- std::map<const **Voxel** *, **Impulse**> **m_voxellImpulseMap**
- std::list<**Impulse**> **m_impulses**

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

- src/world/handler/impulseaccumulator.h
- src/world/handler/impulseaccumulator.cpp

6.121 InertiaFollower Class Reference

Inheritance diagram for InertiaFollower:



Public Member Functions

- **InertiaFollower** (float directionalInertia, float angularInertia)
- const glm::vec3 & **position** () const
- void **setPosition** (const glm::vec3 &position)
- const glm::quat & **orientation** () const
- void **setOrientation** (const glm::quat &orientation)
- void **follow** (const glm::vec3 &targetPosition, const glm::quat &targetOrientation, float deltaSec)

Protected Attributes

- glm::vec3 **m_position**
- float **m_directionalInertia**
- glm::quat **m_orientation**
- float **m_angularInertia**

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

- src/utils/inertiafollower.h
- src/utils/inertiafollower.cpp

6.122 InputConfigurator Class Reference

Public Member Functions

- **InputConfigurator** (std::vector< [ActionKeyMapping](#) * > *actions, [SecondaryInputValues](#) *secondaryInputValues, [Property](#)< float > *deadzone, [HUD](#) *hud)
- void **startConfiguration** (InputClass inputClass)
- bool **isConfiguring** ()
- void **update** ()
- void **setActions** (std::vector< [ActionKeyMapping](#) * > *actions)
- void **setSecondaryInputValues** ([SecondaryInputValues](#) *values)
- void **setLastInput** ([InputMapping](#) lastInput, InputClass inputClass)
- [InputMapping](#) **lastInput** (InputClass inputClass)

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

- src/ui/inputconfigurator.h
- src/ui/inputconfigurator.cpp

6.123 InputMapping Class Reference

Public Member Functions

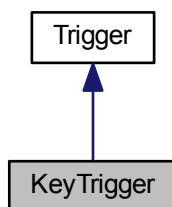
- **InputMapping** (InputType type, int index, float maxValue, float idleValue)
- InputType **type** ()
- int **index** ()
- float **maxValue** ()
- float **idleValue** ()

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

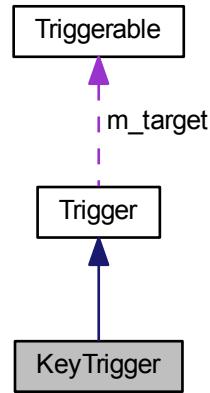
- src/input/inputmapping.h
- src/input/inputmapping.cpp

6.124 KeyTrigger Class Reference

Inheritance diagram for KeyTrigger:



Collaboration diagram for KeyTrigger:



Public Member Functions

- **KeyTrigger** (int glfwKey)
- int **key** () const
- void **setKey** (int glfwKey)
- virtual void **update** (float deltaSec) override

Protected Attributes

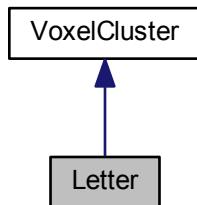
- int **m_glfwKey**
- int **m_lastState**

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

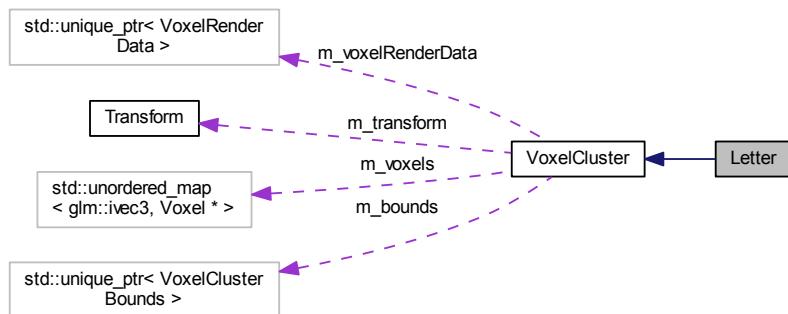
- src/utils/statemachine/keytrigger.h
- src/utils/statemachine/keytrigger.cpp

6.125 Letter Class Reference

Inheritance diagram for Letter:



Collaboration diagram for Letter:



Public Member Functions

- **Letter** (float scale=1.0)

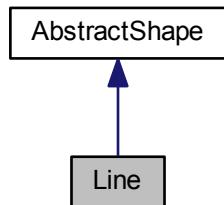
Additional Inherited Members

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

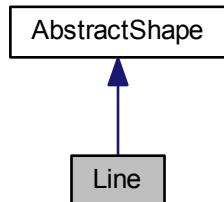
- src/ui/letter.h
- src/ui/letter.cpp

6.126 Line Class Reference

Inheritance diagram for Line:



Collaboration diagram for Line:



Public Member Functions

- **Line** (const glm::vec3 &a, const glm::vec3 &b)
- const glm::vec3 & **a** () const
- void **setA** (const glm::vec3 &a)
- const glm::vec3 & **b** () const
- void **setB** (const glm::vec3 &b)
- virtual bool **intersects** (const [Sphere](#) &sphere) const override
- virtual bool **nearTo** (const [TAABB](#)< int > &aabb) const override
- virtual bool **containedBy** (const [TAABB](#)< int > &aabb) const override

Protected Attributes

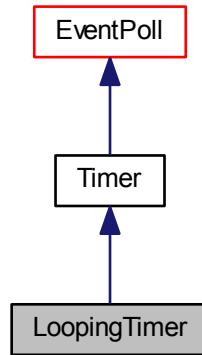
- glm::vec3 **m_a**
- glm::vec3 **m_b**

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

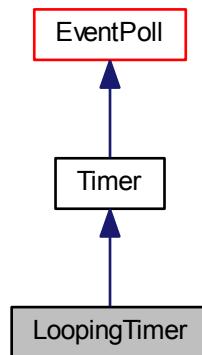
- src/geometry/line.h
- src/geometry/line.cpp

6.127 LoopingTimer Class Reference

Inheritance diagram for LoopingTimer:



Collaboration diagram for LoopingTimer:



Public Member Functions

- **LoopingTimer** (float interval, const std::function< void() > &callback)

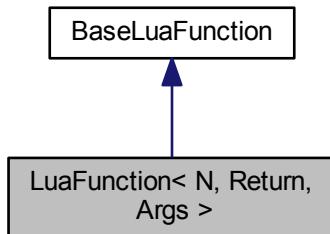
Additional Inherited Members

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

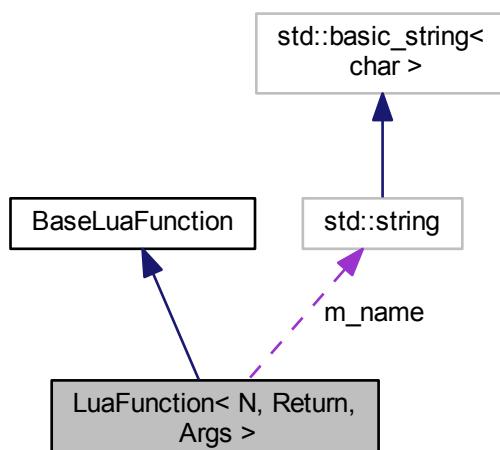
- src/events/loopingtimer.h
- src/events/loopingtimer.cpp

6.128 `LuaFunction< N, Return, Args >` Class Template Reference

Inheritance diagram for `LuaFunction< N, Return, Args >`:



Collaboration diagram for `LuaFunction< N, Return, Args >`:



Public Member Functions

- `LuaFunction (lua_State *&state, const std::string &name, Return(*function)(Args...))`
- `m_name (name)`
- `Lua::pushclosure (state,&Luaw::_lua_dispatcher, 1)`
- `Lua::setglobal (state, name.c_str())`
- `int apply (lua_State *state)`
- `void operator= (LuaFunction &) = delete`

Public Attributes

- `m_state state`

Protected Attributes

- std::function< Return(Args...)> **m_function**
- std::string **m_name**
- lua_State ** **m_state**

6.128.1 Member Data Documentation

6.128.1.1 template<int N, typename Return , typename... Args> m_state **LuaFunction< N, Return, Args >**::state

Initial value:

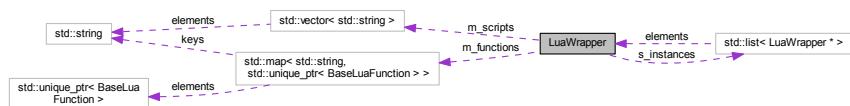
```
{
    Lua::pushlightuserdata(state, (void *)static_cast<BaseLuaFunction *>(this))}
```

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

- src/scripting/elematelua/luawrapperfunction.h

6.129 LuaWrapper Class Reference

Collaboration diagram for LuaWrapper:



Classes

- struct **_pop**
- struct **_pop< 0, Ts...>**
- struct **_pop< 1, T >**

Public Member Functions

- void **loadScript** (const std::string &script)
- void **loadString** (const std::string &script)
- void **removeScript** (const std::string &script)
- void **reloadScripts** ()
- template<typename... Ret, typename... Args>
 _pop< sizeof...(Ret), Ret...>::type **call** (const std::string &fun, const Args &...args)
- template<typename Return , typename Class , typename... Args>
 void **Register** (const std::string &name, Class *obj, Return(Class::*const method)(Args...))
- template<typename Return , typename... Args>
 void **Register** (const std::string &name, std::function< Return(Args...)> function)
- template<typename Return , typename... Args>
 void **Register** (const std::string &name, Return(*function)(Args...))
- template<typename... Return, typename... Args>
 void **Register** (const std::string &name, std::function< std::tuple<Return...> (Args...)> function)

- template<typename... Return, typename... Args>
void **Register** (const std::string &name, std::tuple<Return...>(*function)(Args...))
- void **Unregister** (const std::string &name)
- void **operator=** ([LuaWrapper](#) &) = delete

Static Public Member Functions

- static void **reloadAll** ()

Protected Member Functions

- void **luaError** ()
- void **pushFunc** (const std::string &func) const
- void **callFunc** (const int numArgs, const int numRet)
- void **popStack** (const int index)
- template<typename T, typename... Ts>
void **push** (const T value, const Ts...values) const
- void **push** () const
- void **push** (const std::string &value) const
- void **push** (const char *value) const
- void **push** (const int8_t value) const
- void **push** (const uint8_t value) const
- void **push** (const int16_t value) const
- void **push** (const uint16_t value) const
- void **push** (const int32_t value) const
- void **push** (const uint32_t value) const
- void **push** (const int64_t value) const
- void **push** (const uint64_t value) const
- void **push** (const float value) const
- void **push** (const double value) const
- void **push** (const bool value) const
- void **push** (const glm::vec3 &value) const
- template<typename T>
T **fetch** (const int index) const
- template<typename... T>
[_pop](#)< sizeof...(T), T...>::type **pop** ()
- template<>
int **fetch** (const int index) const
- template<>
double **fetch** (const int index) const
- template<>
float **fetch** (const int index) const
- template<>
unsigned long **fetch** (const int index) const
- template<>
bool **fetch** (const int index) const

Protected Attributes

- **lua_State** * **m_state**
- std::vector<std::string> **m_scripts**
- int **m_err**
- std::map<std::string,
std::unique_ptr
<[BaseLuaFunction](#)>> **m_functions**

Static Protected Attributes

- static std::list< [LuaWrapper](#) * > **s_instances**

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

- [src/scripting/elematelua/luawrapper.h](#)
- [src/scripting/elematelua/luawrapper.cpp](#)

6.130 Math Class Reference

Static Public Member Functions

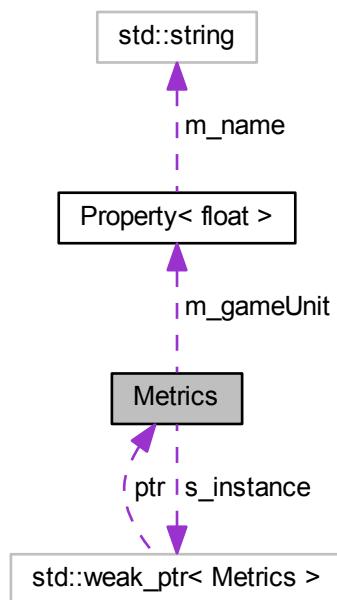
- static uint32_t **nextPowerOf2** (uint32_t n)

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

- [src/utils/math.h](#)
- [src/utils/math.cpp](#)

6.131 Metrics Class Reference

Collaboration diagram for Metrics:



Public Member Functions

- float **gameUnit** ()
- void **setGameUnit** (float gameUnit)
- float **toGameUnits** (float metre)
- float **toMetres** (float gameUnits)

Static Public Member Functions

- static std::shared_ptr< Metrics > **instance** ()

Protected Attributes

- **Property**< float > **m_gameUnit**

Static Protected Attributes

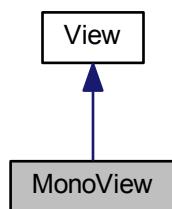
- static std::weak_ptr< Metrics > **s_instance**

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

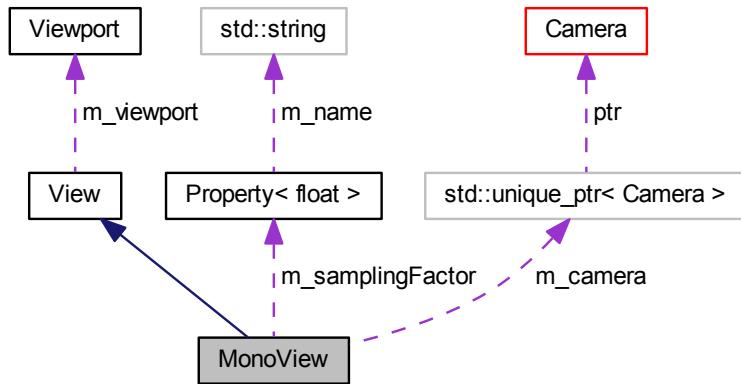
- src/utils/metrics.h
- src/utils/metrics.cpp

6.132 MonoView Class Reference

Inheritance diagram for MonoView:



Collaboration diagram for MonoView:



Public Member Functions

- **MonoView** (const [Viewport](#) &viewport)
- virtual void **setViewport** (const [Viewport](#) &viewport) override
- virtual float **fovy** () const override
- virtual float **zNear** () const override
- virtual float **aspectRatio** () const override
- virtual void **draw** (const [Scene](#) &scene, const [CameraHead](#) &cameraHead) override

Protected Attributes

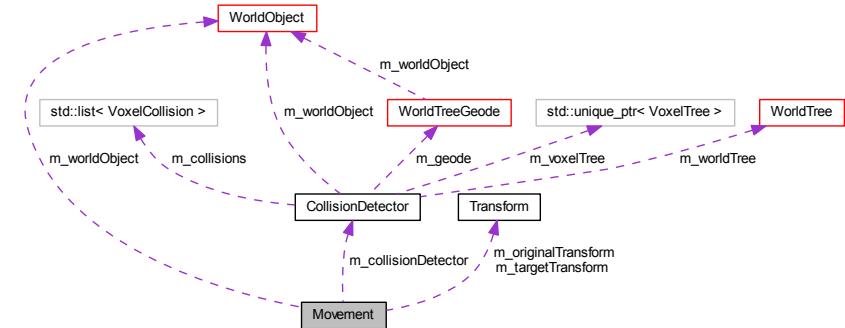
- std::unique_ptr< [Camera](#) > **m_camera**
- [Property](#)< float > **m_samplingFactor**

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

- src/display/monoview.h
- src/display/monoview.cpp

6.133 Movement Class Reference

Collaboration diagram for Movement:



Public Member Functions

- **Movement** ([WorldObject](#) &worldObject, const [Transform](#) &originalTransform, const [Transform](#) &targetTransform)
- bool **perform** ()

Protected Member Functions

- bool **performSplitted** ()
- bool **performStepped** ()
- int **calculateStepCount** ()
- [Transform](#) **calculateStep** (int s, int stepCount) const

Protected Attributes

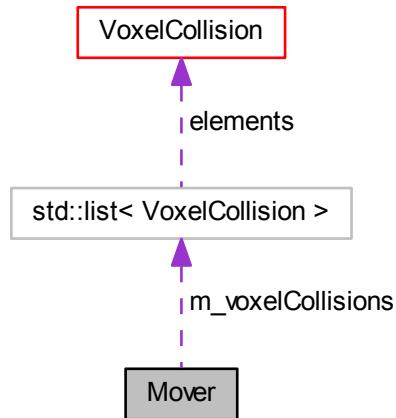
- [WorldObject](#) & **m_worldObject**
- [CollisionDetector](#) & **m_collisionDetector**
- [Transform](#) **m_originalTransform**
- [Transform](#) **m_targetTransform**
- float **m_distance**

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

- src/physics/movement.h
- src/physics/movement.cpp

6.134 Mover Class Reference

Collaboration diagram for Mover:



Public Member Functions

- void **moveWorldObjects** (float deltaSec)
- std::list< [VoxelCollision](#) > & **voxelCollisions** ()

Protected Attributes

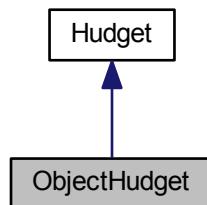
- std::list< [VoxelCollision](#) > **m_voxelCollisions**

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

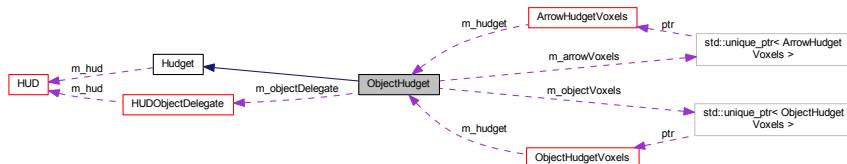
- src/world/handler/mover.h
- src/world/handler/mover.cpp

6.135 ObjectHudget Class Reference

Inheritance diagram for ObjectHudget:



Collaboration diagram for ObjectHudget:



Public Member Functions

- **ObjectHudget** (`HUD *hud, HUDObjectDelegate *objectDelegate`)
- virtual void **update** (float deltaSec) override
- virtual void **draw** () override
- virtual bool **isAt** (const `Ray` &ray) const override
- virtual void **onClick** (ClickType clickType) override
- `HUDObjectDelegate * objectDelegate ()`

Protected Member Functions

- void **updateTargeted** ()
- bool **isInsideFov** ()
- `glm::vec3 closestPointInsideFov ()`
- void **calculateOpeningAngle** ()
- void **updateFov** ()

Protected Attributes

- `HUDObjectDelegate * m_objectDelegate`
- `std::unique_ptr<ObjectHudgetVoxels> m_objectVoxels`
- `std::unique_ptr<ArrowHudgetVoxels> m_arrowVoxels`

- bool **m_targeted**
 - bool **m_insideFov**
 - float **m_fovy**
 - float **m_fovx**

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

- src/ui/hud/objecthudget.h
 - src/ui/hud/objecthudget.cpp

6.136 ObjectHudgetCornerVoxels Class Reference

Collaboration diagram for ObjectBudgetCornerVoxels:



Public Member Functions

- **ObjectHudgetCornerVoxels** (`ObjectHudgetVoxels *objectHudgetVoxels, const glm::ivec3 &baseOffset`)
 - const `glm::vec3 & position () const`
 - const `glm::quat & orientation () const`
 - void `update (float deltaSec)`
 - void `draw (int index)`

Protected Member Functions

- void **addIndex** (int index, uint32_t color, float scale)

Protected Attributes

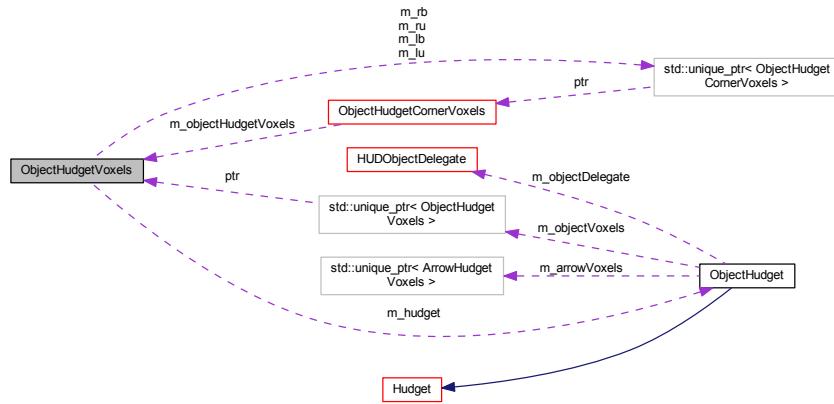
- `ObjectHudgetVoxels * m_objectHudgetVoxels`
 - `glm::ivec3 m_baseOffset`
 - `glm::vec3 m_position`
 - `glm::quat m_orientation`
 - `std::vector< std::unique_ptr< VoxelCluster > > m_voxelClusters`

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

- src/ui/hud/objecthudgetcornervoxels.h
 - src/ui/hud/objecthudgetcornervoxels.cpp

6.137 ObjectHudgetVoxels Class Reference

Collaboration diagram for ObjectHudgetVoxels:



Public Member Functions

- **ObjectHudgetVoxels** (*ObjectHudget* **hudget*)
- **ObjectHudget** * *hudget* ()
- void **setTargetHighlight** (bool *targetHighlight*)
- void **setRelationType** (FactionRelationType *relationType*)
- float **openingAngle** () const
- void **setOpeningAngle** (float *openingAngle*)
- void **update** (float *deltaSec*)
- void **draw** ()
- bool **isAt** (const *Ray* &*ray*) const

Protected Attributes

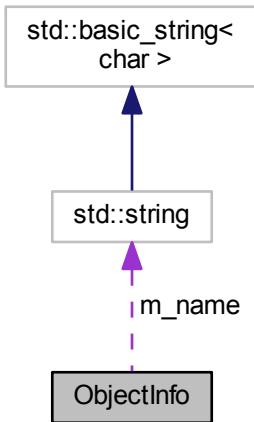
- **ObjectHudget** * **m_hudget**
- std::unique_ptr<ObjectHudgetCornerVoxels> **m_lu**
- std::unique_ptr<ObjectHudgetCornerVoxels> **m_lb**
- std::unique_ptr<ObjectHudgetCornerVoxels> **m_ru**
- std::unique_ptr<ObjectHudgetCornerVoxels> **m_rb**
- float **m_openingAngle**
- bool **m_targetHighlight**
- FactionRelationType **m_relationType**

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

- src/ui/hud/objecthudgetvoxels.h
- src/ui/hud/objecthudgetvoxels.cpp

6.138 ObjectInfo Class Reference

Collaboration diagram for ObjectInfo:



Public Member Functions

- `std::string name ()`
- `void setName (const std::string &name)`
- `bool showOnHud ()`
- `void setShowOnHud (bool show)`
- `bool canLockOn ()`
- `void setCanLockOn (bool canLockOn)`

Protected Attributes

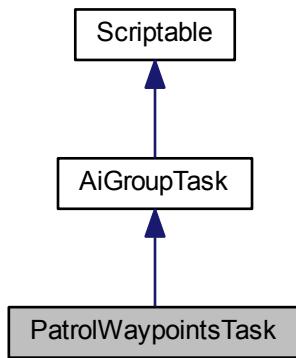
- `std::string m_name`
- `bool m_showOnHud`
- `bool m_canLockOn`

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

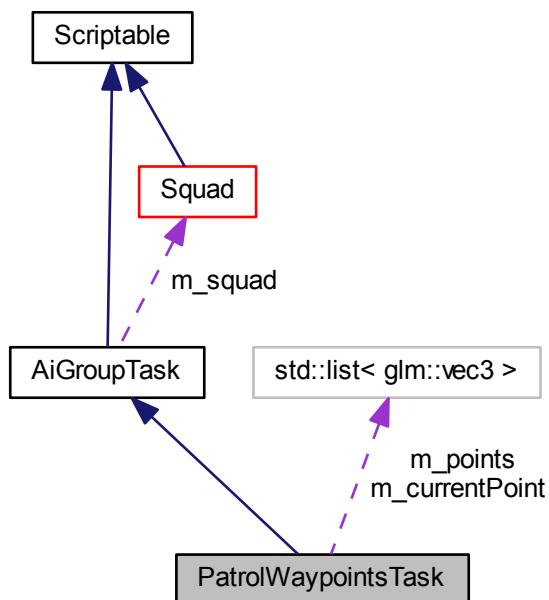
- `src/ui/objectinfo.h`
- `src/ui/objectinfo.cpp`

6.139 PatrolWaypointsTask Class Reference

Inheritance diagram for PatrolWaypointsTask:



Collaboration diagram for PatrolWaypointsTask:



Public Member Functions

- **PatrolWaypointsTask** ([Squad](#) &squad)
- **PatrolWaypointsTask** ([Squad](#) &squad, const [std::list<glm::vec3>](#) &points)

- void **appendWaypoint** (const glm::vec3 &point)
- virtual void **update** (float deltaSec) override
- const glm::vec3 * **currentTargetPoint** ()

Protected Member Functions

- virtual void **onNewLeader** ([Ship](#) *leader) override
- virtual void **onMemberJoin** ([Ship](#) *member) override

Protected Attributes

- std::shared_ptr< [FlyToTask](#) > **m_leaderFlyTask**
- std::list< glm::vec3 > **m_points**
- std::list< glm::vec3 >::iterator **m_currentPoint**

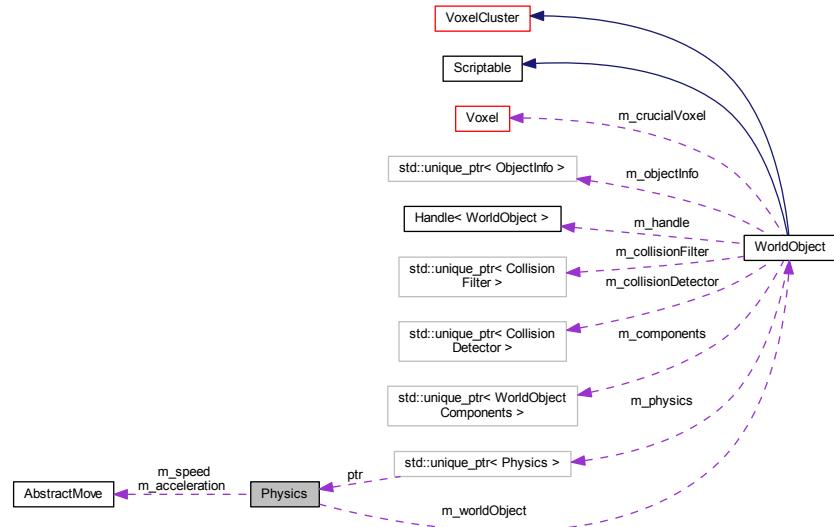
Additional Inherited Members

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

- src/ai/grouptasks/patrolwaypointtask.h
- src/ai/grouptasks/patrolwaypointtask.cpp

6.140 Physics Class Reference

Collaboration diagram for Physics:



Public Member Functions

- **Physics** ([WorldObject](#) &worldObject, float scale)
- float **directionalDampening** () const
- void **setDirectionalDampening** (float directionalDampening)

- float **angularDampening** () const
- void **setAngularDampening** (float angularDampening)
- const **Speed** & **speed** () const
- void **setSpeed** (const **Speed** &speed)
- const **Acceleration** & **acceleration** () const
- void **setAcceleration** (const **Acceleration** &acceleration)
- float **mass** () const
- const **Transform** **projectedTransformIn** (float deltaSec)
- std::list< **VoxelCollision** > & **move** (float deltaSec)
- void **addVoxel** (**Voxel** *voxel)
- void **removeVoxel** (**Voxel** *voxel)

Protected Member Functions

- void **voxelChanged** (**Voxel** *voxel, bool isAdd)
- virtual void **updateSpeed** (float deltaSec)

Protected Attributes

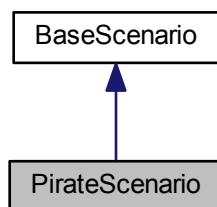
- **WorldObject** & **m_worldObject**
- **Speed** **m_speed**
- **Acceleration** **m_acceleration**
- float **m_directionalDampening**
- float **m_angularDampening**
- float **m_mass**
- glm::vec3 **m_accumulatedMassVec**
- float **m_massScaleFactor**

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

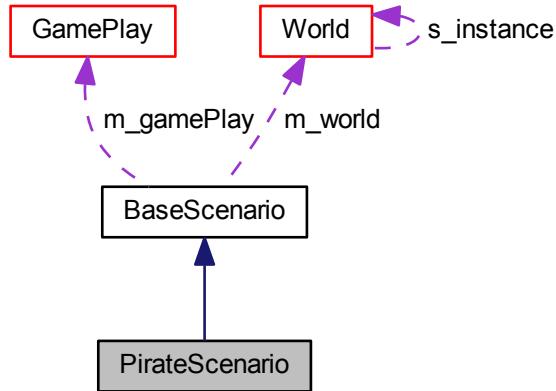
- src/physics/physics.h
- src/physics/physics.cpp

6.141 PirateScenario Class Reference

Inheritance diagram for PirateScenario:



Collaboration diagram for PirateScenario:



Public Member Functions

- **PirateScenario** (`GamePlay *inGame`)

Protected Member Functions

- virtual void **populateWorld** () override
- void **createArmada** ()

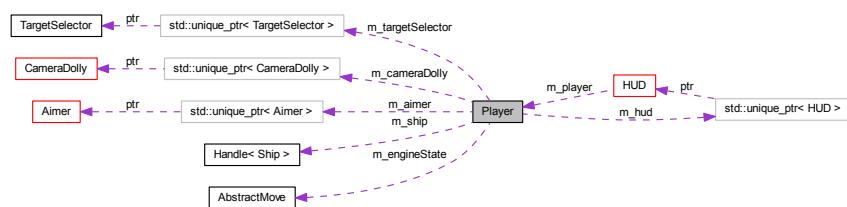
Additional Inherited Members

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

- `src/scenarios/piratescenario.h`
- `src/scenarios/piratescenario.cpp`

6.142 Player Class Reference

Collaboration diagram for Player:



Public Member Functions

- `Ship * ship ()`
- `void setShip (Ship *ship)`
- `void update (float deltaSec)`
- `CameraHead & cameraHead ()`
- `HUD & hud ()`
- `void fire ()`
- `void move (const glm::vec3 &vec)`
- `void rotate (const glm::vec3 &euler)`
- `void selectTarget (bool next)`
- `void setTarget (WorldObject *target)`

Protected Attributes

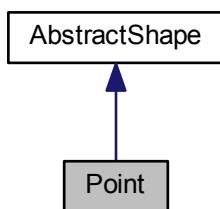
- `Handle< Ship > m_ship`
- `std::unique_ptr< CameraDolly > m_cameraDolly`
- `std::unique_ptr< HUD > m_hud`
- `std::unique_ptr< TargetSelector > m_targetSelector`
- `std::unique_ptr< Aimer > m_aimer`
- `EngineState m_engineState`

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

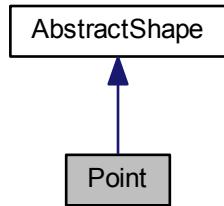
- `src/player.h`
- `src/player.cpp`

6.143 Point Class Reference

Inheritance diagram for Point:



Collaboration diagram for Point:



Public Member Functions

- **Point** (const `glm::vec3 &pos)`
- const `glm::vec3 & position () const`
- void **setPosition** (const `glm::vec3 &pos)`
- virtual bool **intersects** (const `Sphere &sphere)` const override
- virtual bool **nearTo** (const `TAABB< int > &aabb)` const override
- virtual bool **containedBy** (const `TAABB< int > &aabb)` const override

Protected Attributes

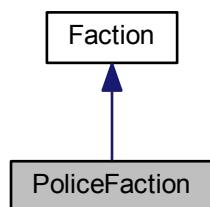
- `glm::vec3 m_position`

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

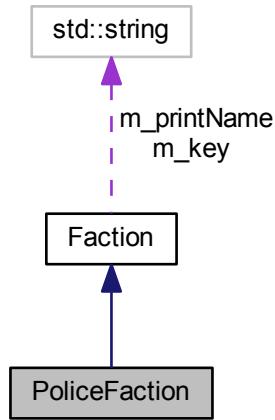
- `src/geometry/point.h`
- `src/geometry/point.cpp`

6.144 PoliceFaction Class Reference

Inheritance diagram for PoliceFaction:



Collaboration diagram for PoliceFaction:



Public Member Functions

- `PoliceFaction (FactionMatrix *factionMatrix)`

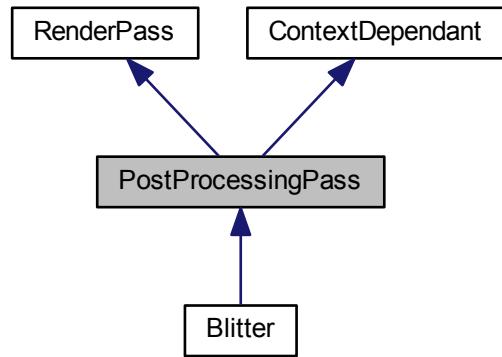
Additional Inherited Members

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

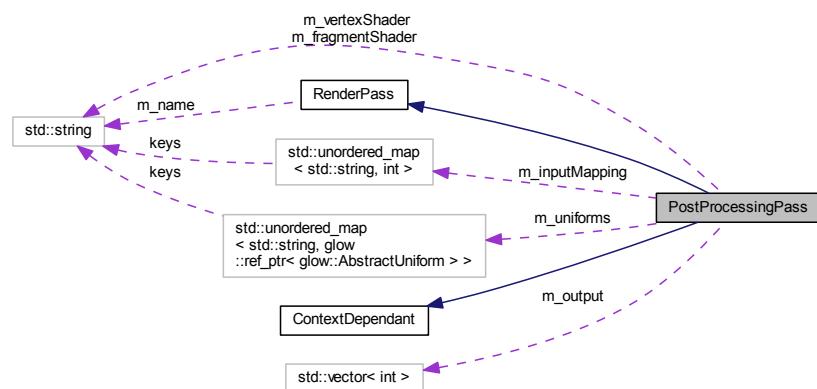
- `src/factions/policefaction.h`

6.145 PostProcessingPass Class Reference

Inheritance diagram for PostProcessingPass:



Collaboration diagram for PostProcessingPass:



Public Member Functions

- **PostProcessingPass** (const std::string &name, std::shared_ptr< ScreenQuad > quad)
- virtual void **apply** (FrameBuffer &frameBuffer, const RenderMetaData &metadata) override
- void **beforeDraw** (FrameBuffer &frameBuffer)
- void **setInputMapping** (const std::unordered_map< std::string, int > &inputMapping)
- void **setOutput** (const std::vector< int > &output)
- void **setFragmentShader** (const std::string &output)
- template<typename T>
void **setUniform** (const std::string &name, const T &value)
- bool **isEnabled** ()
- void **setEnabled** (bool enabled)

Protected Member Functions

- void **initialize** ()
- void **restoreUniforms** ()
- virtual void **beforeContextDestroy** () override
- virtual void **afterContextRebuild** () override

Protected Attributes

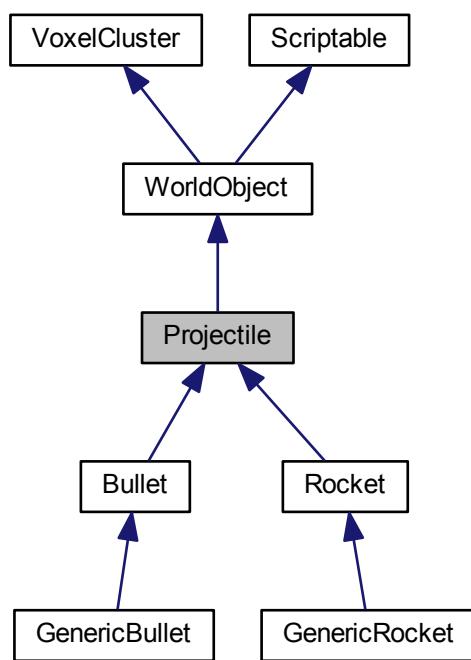
- std::unordered_map
 < std::string, glow::ref_ptr
 < glow::AbstractUniform > > **m_uniforms**
- glow::ref_ptr< glow::Program > **m_program**
- std::shared_ptr< ScreenQuad > **m_quad**
- std::unordered_map
 < std::string, int > **m_inputMapping**
- std::vector< int > **m_output**
- std::string **m_fragmentShader**
- std::string **m_vertexShader**
- bool **m_enabled**

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

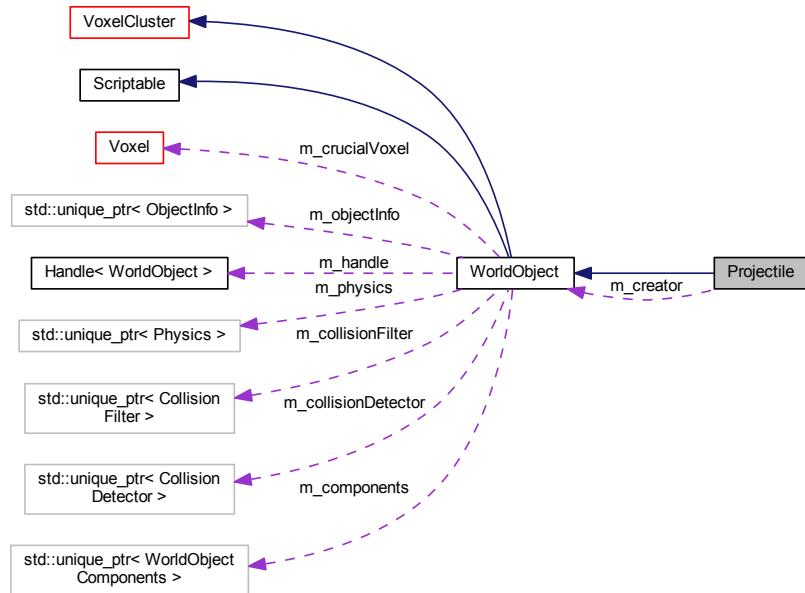
- src/display/rendering/postprocessingpass.h
- src/display/rendering/postprocessingpass.cpp
- src/display/rendering/postprocessingpass.inl

6.146 Projectile Class Reference

Inheritance diagram for Projectile:



Collaboration diagram for Projectile:



Public Member Functions

- `virtual const SoundProperties & hitSound () const =0`
- `WorldObject * creator ()`
- `void setCreator (WorldObject *creator)`
- `float lifetime () const`
- `void setLifetime (float lifetime)`
- `virtual void update (float deltaSec) override`
- `virtual void onCollision () override`
- `virtual void onSpawnFail () override`

Protected Member Functions

- `virtual void onLifetimeOver ()`
- `virtual void spawnExplosion ()=0`

Protected Attributes

- `WorldObject * m_creator`
- `float m_lifetime`

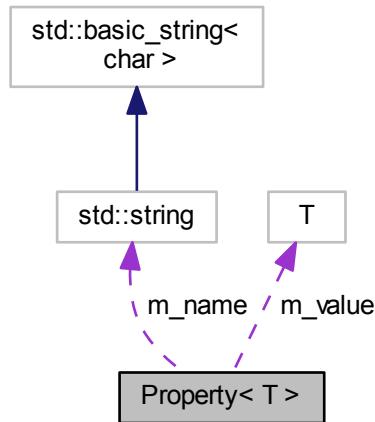
Additional Inherited Members

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

- `src/equipment/weapons/projectile.h`
- `src/equipment/weapons/projectile.cpp`

6.147 Property< T > Class Template Reference

Collaboration diagram for Property< T >:



Public Member Functions

- **Property** (const std::string &name)
- **Property** (const std::string &name, const T &defaultValue)
- const std::string & **name** () const
- T **get** () const
- void **set** (const T &value)
- **operator T** () const
- T * **operator->** ()

Static Public Member Functions

- static T **get** (const std::string &name)
- static T **get** (const std::string &name, const T &defaultValue)

Protected Attributes

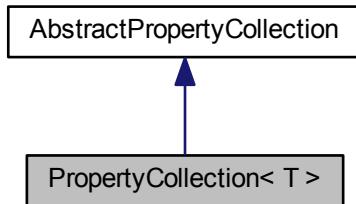
- std::string **m_name**
- T **m_value**

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

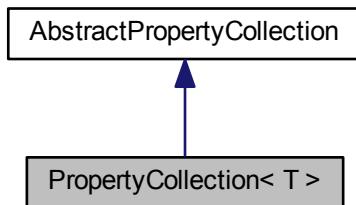
- src/property/property.h
- src/property/property.inl

6.148 PropertyCollection< T > Class Template Reference

Inheritance diagram for PropertyCollection< T >:



Collaboration diagram for PropertyCollection< T >:



Public Member Functions

- **PropertyCollection** (regexns::regex regex, std::function< T(const std::string &) > converter)
- void **registerProperty** ([Property](#)< T > *prop)
- void **registerProperty** ([Property](#)< T > *prop, const T &defaultValue)
- void **unregisterProperty** ([Property](#)< T > *prop)
- virtual bool **update** (const std::string &key, const std::string &svalue) override
- void **set** (const std::string &key, const T &value)
- T **get** (const std::string &name) const
- T **get** (const std::string &name, const T &defaultValue) const

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

- src/property/propertycollection.h
- src/property/propertycollection.inl

6.149 PropertyConverter Class Reference

Static Public Member Functions

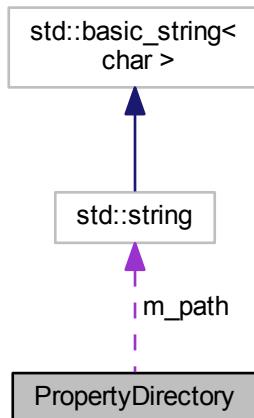
- static float **floatConverter** (const std::string &s)
- static glm::vec2 **vec2Converter** (const std::string &s)
- static glm::vec3 **vec3Converter** (const std::string &s)
- static glm::vec4 **vec4Converter** (const std::string &s)
- static std::list< std::string > **listConverter** (const std::string &s)
- static [InputMapping](#) **inputMappingConverter** (const std::string &s)

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

- src/property/propertyconverter.h
- src/property/propertyconverter.cpp

6.150 PropertyDirectory Class Reference

Collaboration diagram for PropertyDirectory:



Public Member Functions

- **PropertyDirectory** (const std::string &path)
- void **read** ()

Protected Attributes

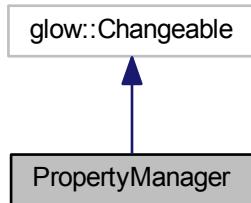
- std::string **m_path**

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

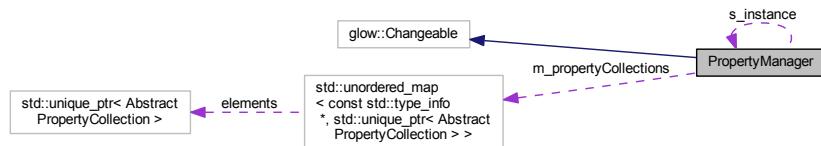
- src/property/propertydirectory.h
- src/property/propertydirectory.cpp

6.151 PropertyManager Class Reference

Inheritance diagram for PropertyManager:



Collaboration diagram for PropertyManager:



Public Member Functions

- void **load** (const std::string &file, const std::string &prefix="")
- template<typename T>
void **registerProperty** ([Property](#)< T > *prop)
- template<typename T>
void **registerProperty** ([Property](#)< T > *prop, const T &defaultValue)
- template<typename T>
void **unregisterProperty** ([Property](#)< T > *prop)
- template<typename T>
T **get** (const std::string &name)
- template<typename T>
T **get** (const std::string &name, const T &defaultValue)

Static Public Member Functions

- static [PropertyManager](#) * **instance** ()
- static void **reset** ()

Protected Member Functions

- template<typename T>
[PropertyCollection](#)< T > * **getPropertyCollection** ()
- template<typename T>
void **addPropertyCollection** ([PropertyCollection](#)< T > *collection)

Protected Attributes

- std::unordered_map< const std::type_info *, std::unique_ptr <[AbstractPropertyCollection](#)>> **m_propertyCollections**

Static Protected Attributes

- static [PropertyManager](#) * **s_instance**

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

- src/property/propertymanager.h
- src/property/propertymanager.cpp
- src/property/propertymanager.inl

6.152 RandBool Class Reference

Static Public Member Functions

- static bool **rand** (float trueProbability)

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

- src/utils/randbool.h
- src/utils/rand.cpp

6.153 RandFloat Class Reference

Static Public Member Functions

- static float **rand** (float from, float to)
- static float **randomize** (float value, float randomization)

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

- src/utils/randfloat.h
- src/utils/rand.cpp

6.154 RandVec3 Class Reference

Static Public Member Functions

- static glm::vec3 **rand** (float from, float to)
- static glm::vec3 **randUnitVec** ()

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

- src/utils/randvec.h
- src/utils/rand.cpp

6.155 Range Class Reference

Public Member Functions

- **Range** (float min, float max)
- float **min** () const
- void **setMin** (float min)
- float **max** () const
- void **setMax** (float max)
- float **clamp** (float value)

Protected Attributes

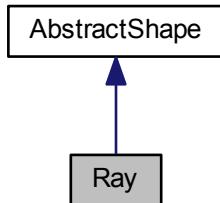
- float **m_min**
- float **m_max**

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

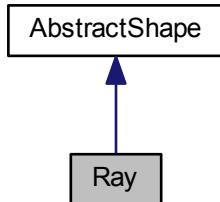
- src/geometry/range.h
- src/geometry/range.cpp

6.156 Ray Class Reference

Inheritance diagram for Ray:



Collaboration diagram for Ray:



Public Member Functions

- **Ray** (const glm::vec3 &origin, const glm::vec3 &direction)
- const glm::vec3 & **origin** () const
- void **setOrigin** (const glm::vec3 &origin)
- const glm::vec3 & **direction** () const
- void **setDirection** (const glm::vec3 &direction)
- virtual bool **intersects** (const [Sphere](#) &sphere) const override
- virtual bool **nearTo** (const [TAABB](#)< int > &aabb) const override
- virtual bool **containedBy** (const [TAABB](#)< int > &aabb) const override

Static Public Member Functions

- static [Ray](#) **fromTo** (const glm::vec3 &from, const glm::vec3 &to)

Protected Attributes

- glm::vec3 **m_origin**
- glm::vec3 **m_direction**

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

- src/geometry/ray.h
- src/geometry/ray.cpp

6.157 RenderMetaData Class Reference

Public Member Functions

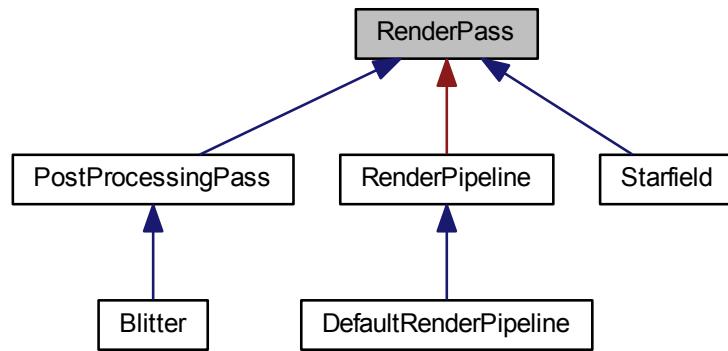
- **RenderMetaData** (const [Camera](#) &camera, EyeSide eyeside)
- const [Camera](#) & **camera** () const
- EyeSide **eyeside** () const

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

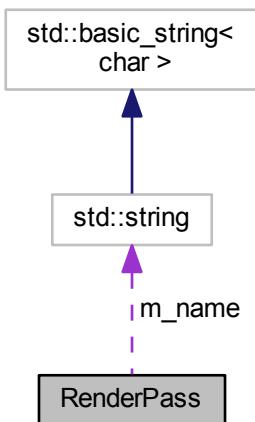
- src/display/rendering/rendermetadata.h
- src/display/rendering/rendermetadata.cpp

6.158 RenderPass Class Reference

Inheritance diagram for RenderPass:



Collaboration diagram for RenderPass:



Public Member Functions

- `RenderPass` (const `std::string` &`name`)
- virtual void `apply` (`FrameBuffer` &`frameBuffer`, const `RenderMetaData` &`metadata`)=0
- const `std::string` & `name` () const

Protected Attributes

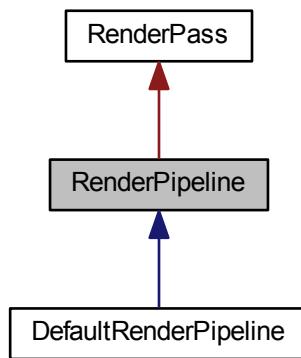
- `std::string m_name`

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

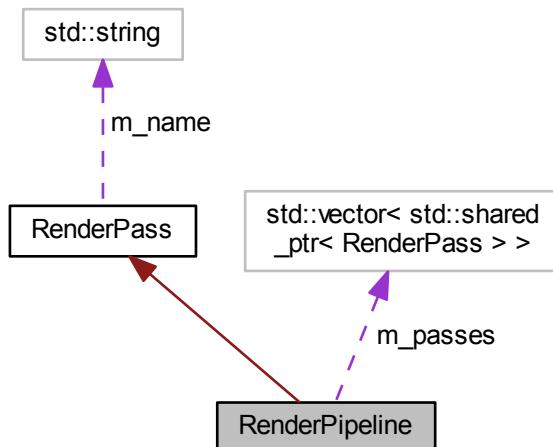
- src/display/rendering/renderpass.h
- src/display/rendering/renderpass.cpp

6.159 RenderPipeline Class Reference

Inheritance diagram for RenderPipeline:



Collaboration diagram for RenderPipeline:



Public Member Functions

- **RenderPipeline** (const std::string &name)

- virtual void **apply** (FrameBuffer &frameBuffer, const RenderMetaData &metadata) override
- virtual void **setup** ()=0
- virtual int **bufferCount** ()=0
- void **add** (std::shared_ptr< RenderPass > pass, int index=-1)
- void **insertAfter** (std::shared_ptr< RenderPass > pass, const std::string &after)

Static Public Member Functions

- static RenderPipeline * **getDefault** ()

Protected Attributes

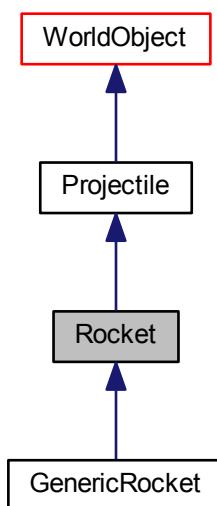
- std::vector< std::shared_ptr< RenderPass > > **m_passes**
- bool **m_initialized**

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

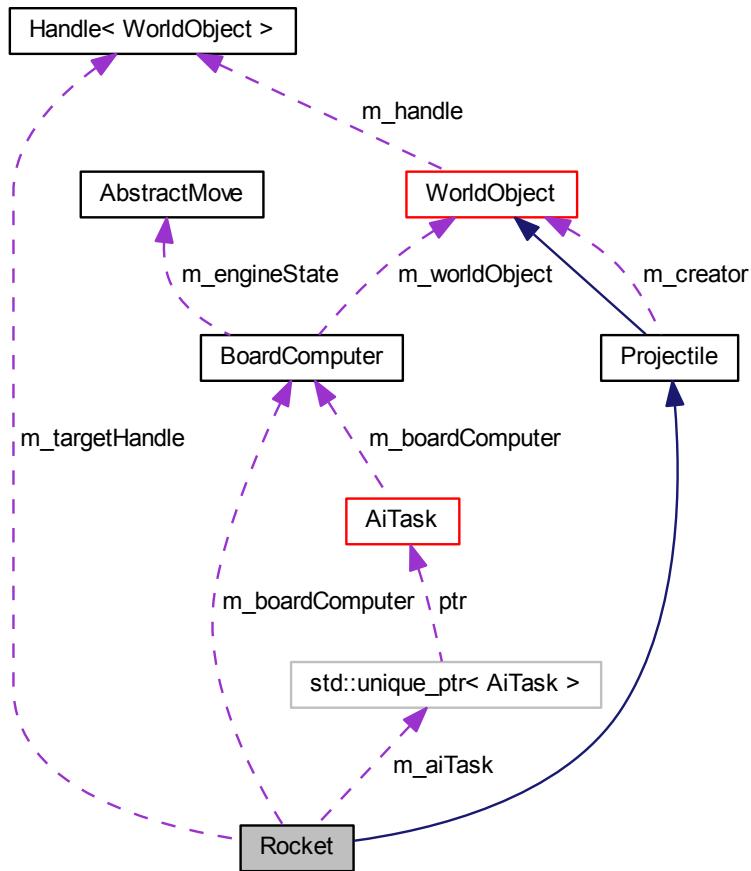
- src/display/rendering/renderpipeline.h
- src/display/rendering/renderpipeline.cpp

6.160 Rocket Class Reference

Inheritance diagram for Rocket:



Collaboration diagram for Rocket:



Public Member Functions

- virtual WorldObjectType **objectType** () const override
- [WorldObject * target \(\)](#)
- void **setTarget** ([WorldObject *targetObject](#))
- virtual void **update** (float deltaSec) override

Protected Attributes

- [Handle< WorldObject > m_targetHandle](#)
- [BoardComputer m_boardComputer](#)
- [std::unique_ptr< AiTask > m_aiTask](#)

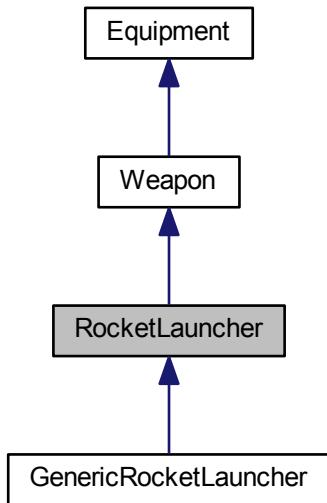
Additional Inherited Members

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

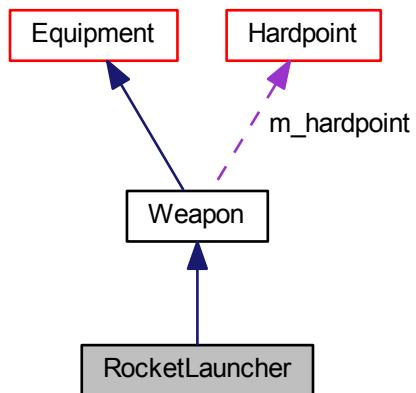
- src/equipment/weapons/rocket.h
- src/equipment/weapons/rocket.cpp

6.161 RocketLauncher Class Reference

Inheritance diagram for RocketLauncher:



Collaboration diagram for RocketLauncher:



Public Member Functions

- **RocketLauncher** (const std::string &equipmentKey)
- virtual void **fireAtObject** ([WorldObject](#) *target)
- virtual void **update** (float deltaSec) override

Protected Member Functions

- virtual `Rocket * createRocket ()=0`
- void `setupRocket (Rocket *rocket, WorldObject *target)`

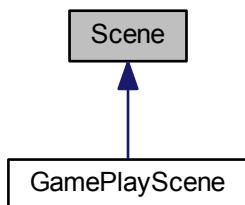
Additional Inherited Members

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

- `src/equipment/weapons/rocketlauncher.h`
- `src/equipment/weapons/rocketlauncher.cpp`

6.162 Scene Class Reference

Inheritance diagram for Scene:



Public Member Functions

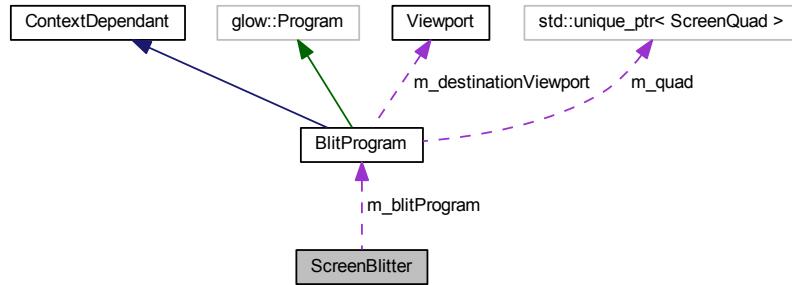
- virtual void `draw (const Camera &camera, glow::FrameBufferObject *destination, const Viewport &destinationViewport, EyeSide side=EyeSide::None) const =0`
- virtual void `update (float deltaSec)=0`

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

- `src/display/scene.h`

6.163 ScreenBlitter Class Reference

Collaboration diagram for ScreenBlitter:



Public Member Functions

- void **setProgram** (`BlitProgram` &blitProgram)
- void **blit** (`FrameBuffer` &source, const `Viewport` &viewport)

Protected Attributes

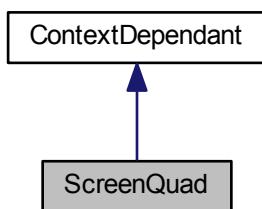
- `BlitProgram * m.blitProgram`

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

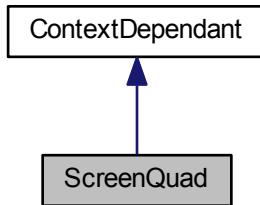
- src/display/screenblitter.h
- src/display/screenblitter.cpp

6.164 ScreenQuad Class Reference

Inheritance diagram for ScreenQuad:



Collaboration diagram for ScreenQuad:



Public Member Functions

- void **draw** ()

Static Public Attributes

- static const int **VERTEX_ATTRIBUTE_LOCATION** = 0

Protected Member Functions

- void **initialize** ()
- void **beforeContextDestroy** () override
- void **afterContextRebuild** () override

Protected Attributes

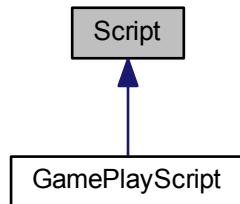
- glow::ref_ptr<glow::VertexArrayObject> **m_vertexArrayObject**
- glow::ref_ptr<glow::Buffer> **m_vertexBuffer**

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

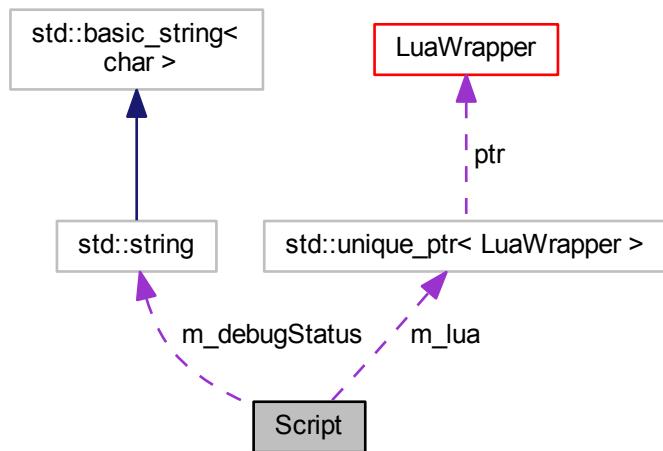
- src/display/rendering/screenquad.h
- src/display/rendering/screenquad.cpp

6.165 Script Class Reference

Inheritance diagram for Script:



Collaboration diagram for Script:



Public Member Functions

- bool **started () const**
- virtual void **load (const std::string &path)**
- virtual void **loadString (const std::string &script)**
- void **start ()**
- const std::string & **debugStatus ()**
- int **apiSetDebugStatus (const std::string &string)**

Protected Attributes

- std::unique_ptr<LuaWrapper> **m_lua**

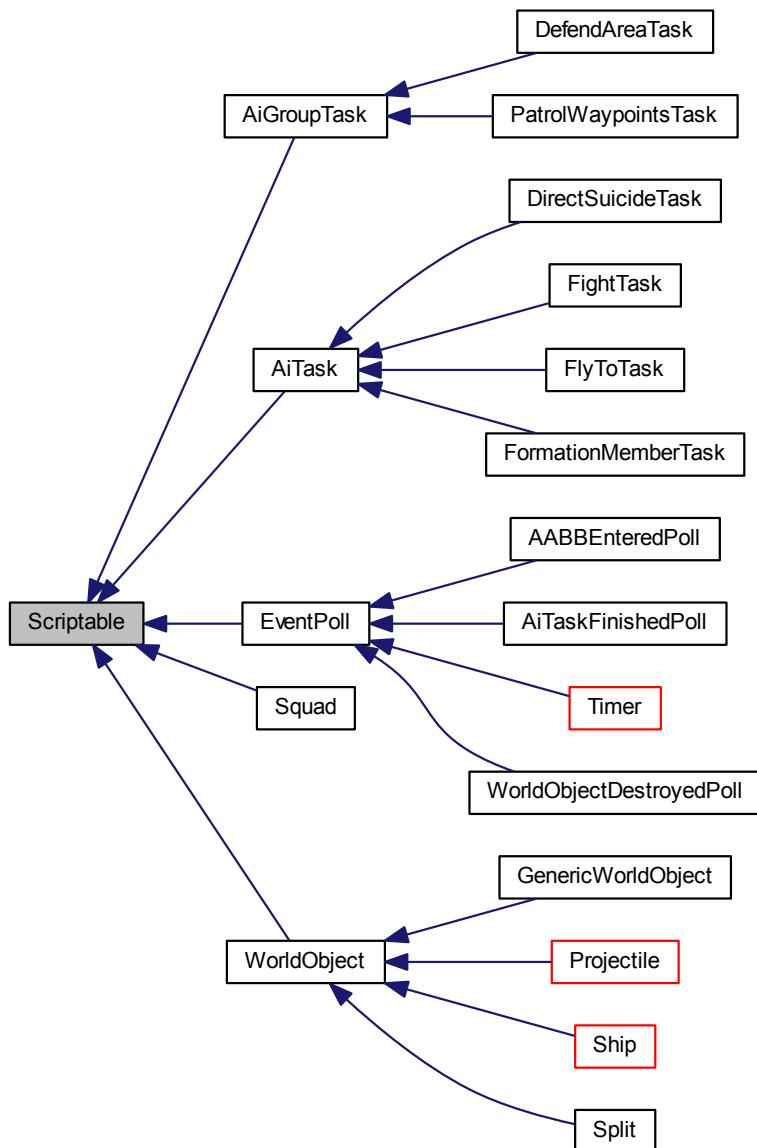
- bool **m_started**
- std::string **m_debugStatus**

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

- src/scripting/script.h
- src/scripting/script.cpp

6.166 Scriptable Class Reference

Inheritance diagram for Scriptable:



Public Member Functions

- int **scriptKey () const**
- void **setScriptKey (int key)**

Static Public Attributes

- static const int **INVALID_KEY = -1**

Protected Attributes

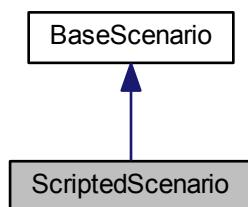
- int **m_key**

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

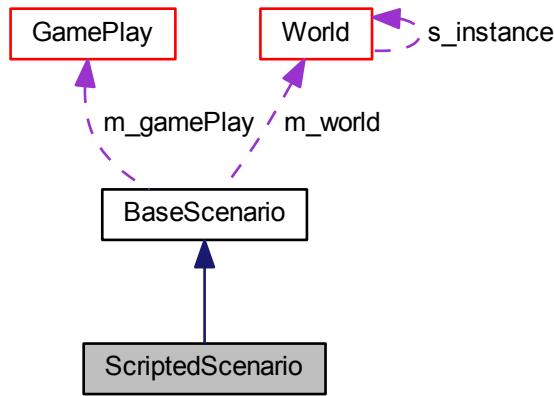
- src/scripting/scriptable.h
- src/scripting/scriptable.cpp

6.167 ScriptedScenario Class Reference

Inheritance diagram for ScriptedScenario:



Collaboration diagram for ScriptedScenario:



Public Member Functions

- **ScriptedScenario** ([GamePlay](#) *gamePlay, const std::string &path)

Protected Member Functions

- virtual void **populateWorld** () override

Protected Attributes

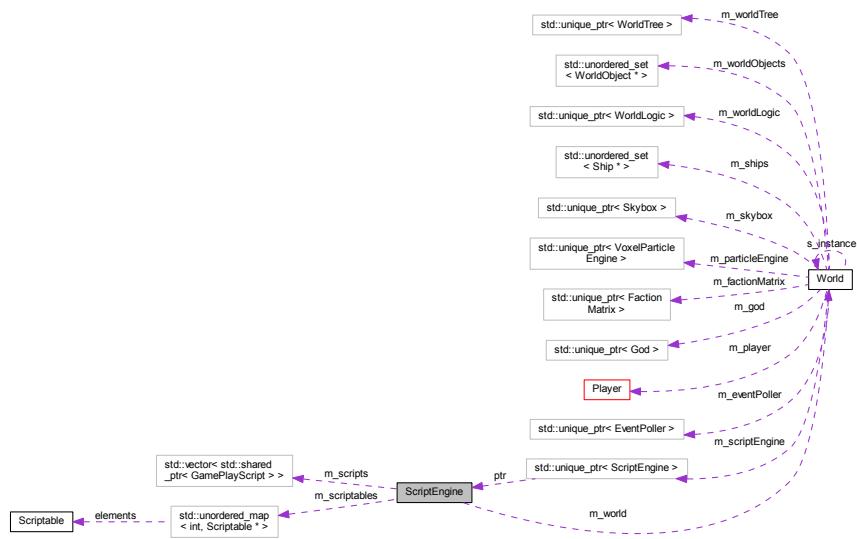
- std::shared_ptr< [GamePlayScript](#) > **m_script**

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

- src/scenarios/scriptedscenario.h
- src/scenarios/scriptedscenario.cpp

6.168 ScriptEngine Class Reference

Collaboration diagram for ScriptEngine:



Public Member Functions

- `ScriptEngine (World *world)`
- `void addScript (std::shared_ptr< GamePlayScript > script)`
- `void start ()`
- `void stop ()`
- `void registerScriptable (Scriptable *scriptable)`
- `void unregisterScriptable (Scriptable *scriptable)`
- template<class T>
 `T * get (int key)`
- `void update (float deltaSec)`

Protected Member Functions

- `Scriptable * getScriptable (int key)`

Protected Attributes

- `World * m_world`
- `std::vector< std::shared_ptr< GamePlayScript > > m_scripts`
- `std::unordered_map< int, Scriptable * > m_scriptables`
- `int m_keyIncrementor`
- `bool m_running`

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

- `src/scripting/scriptengine.h`
- `src/scripting/scriptengine.cpp`
- `src/scripting/scriptengine.inl`

6.169 SecondaryInputValues Struct Reference

Public Attributes

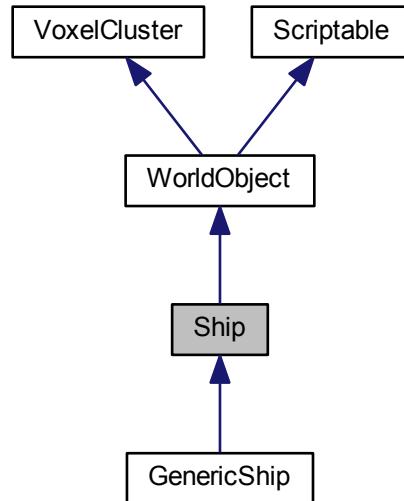
- int **buttonCnt**
- int **axisCnt**
- const unsigned char * **buttonValues**
- const float * **axisValues**

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

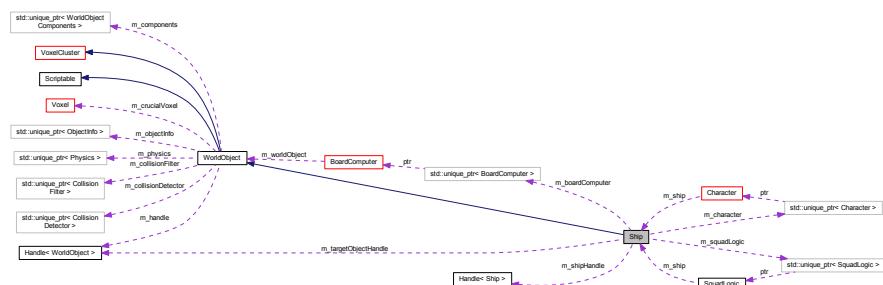
- src/ui/actionkeymapping.h
- src/gamestate/gameplay/running/gameplayrunninginput.cpp

6.170 Ship Class Reference

Inheritance diagram for Ship:



Collaboration diagram for Ship:



Public Member Functions

- virtual WorldObjectType **objectType** () const override
- virtual void **update** (float deltaSec) override
- **Handle< Ship > handle** ()
- void **setTargetObject** ([WorldObject](#) *target)
- [WorldObject](#) * **targetObject** ()
- [BoardComputer](#) * **boardComputer** ()
- [SquadLogic](#) * **squadLogic** ()
- void **setCharacter** ([Character](#) *character)
- [Character](#) * **character** ()

Protected Member Functions

- [Ship](#) ([CollisionFilter](#) *collisionFilter)
- void **updateEnginePosition** ()

Protected Attributes

- std::unique_ptr< [Character](#) > **m_character**
- std::unique_ptr< [BoardComputer](#) > **m_boardComputer**
- std::unique_ptr< [SquadLogic](#) > **m_squadLogic**
- **Handle< Ship > m_shipHandle**
- **Handle< WorldObject > m_targetObjectHandle**

Additional Inherited Members

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

- src/worldobject/ship.h
- src/worldobject/ship.cpp

6.171 SimpleWayfind Class Reference

Static Public Member Functions

- static glm::vec3 **calculateTravelPoint** ([WorldObject](#) &object, glm::vec3 targetPoint)

Static Protected Member Functions

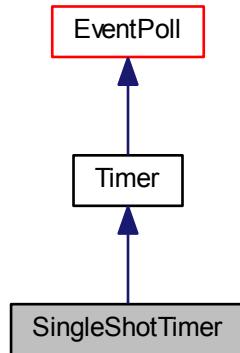
- static glm::vec3 **calculateEvasionDirectionFor** ([WorldObject](#) &self, [WorldObject](#) &obstacle, const glm::vec3 &targetPoint)
- static glm::vec3 **calculateEvasionPointFor** ([WorldObject](#) &self, [WorldObject](#) &obstacle, const glm::vec3 &targetPoint)

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

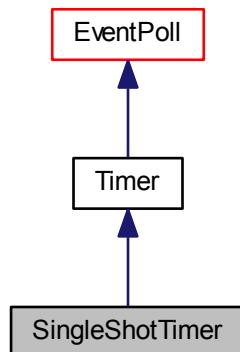
- src/utils/simplewayfind.h
- src/utils/simplewayfind.cpp

6.172 SingleShotTimer Class Reference

Inheritance diagram for SingleShotTimer:



Collaboration diagram for SingleShotTimer:



Public Member Functions

- **SingleShotTimer** (float interval, const std::function< void()> &callback)
- virtual bool **isDead** () override

Protected Member Functions

- virtual void **specialOnCallback** () override

Protected Attributes

- bool **m_ticksing**

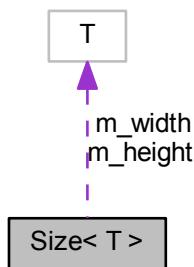
Additional Inherited Members

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

- src/events/singleshottimer.h
- src/events/singleshottimer.cpp

6.173 Size< T > Class Template Reference

Collaboration diagram for Size< T >:



Public Member Functions

- **Size** (T width, T height)
- T **width** () const
- void **setWidth** (T width)
- T **height** () const
- void **setHeight** (T height)

Protected Attributes

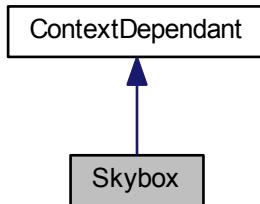
- T **m_width**
- T **m_height**

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

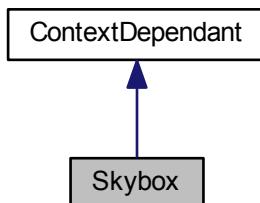
- src/geometry/size.h
- src/geometry/size.inl

6.174 Skybox Class Reference

Inheritance diagram for Skybox:



Collaboration diagram for Skybox:



Public Member Functions

- void **draw** (const [Camera](#) &camera)

Protected Member Functions

- void **initialize** ()
- virtual void **beforeContextDestroy** () override
- virtual void **afterContextRebuild** () override

Protected Attributes

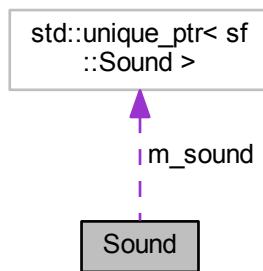
- glow::ref_ptr< glow::Texture > **m_texture**
- glow::ref_ptr< glow::Program > **m_shaderProgram**
- glow::ref_ptr< glow::VertexArrayObject > **m_vertexArrayObject**
- glow::ref_ptr< glow::Buffer > **m_vertexBuffer**

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

- src/skybox.h
- src/skybox.cpp

6.175 Sound Class Reference

Collaboration diagram for Sound:



Public Types

- enum **Status** { **Paused**, **Playing**, **Stopped**, **Disabled** }

Public Member Functions

- **Sound** (const sf::SoundBuffer &sound)
- Status **status** ()
- void **play** ()
- void **stop** ()
- void **pause** ()
- **Sound** & **setPosition** (const glm::vec3 &position)
- **Sound** & **setVolume** (float volume)
- **Sound** & **setAttenuation** (float attenuation)
- **Sound** & **setLooping** (bool loop)
- **Sound** & **setRelativeToListener** (bool relative)
- **Sound** & **setMinDistance** (float distance)

Protected Attributes

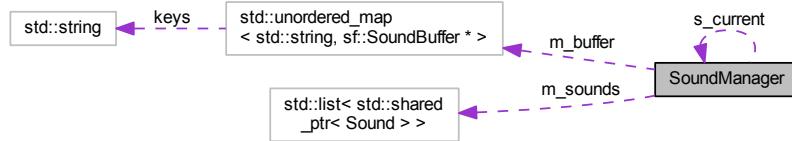
- std::unique_ptr<sf::Sound> **m_sound**

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

- src/sound/sound.h
- src/sound/sound.cpp

6.176 SoundManager Class Reference

Collaboration diagram for SoundManager:



Public Member Functions

- void **setListener** (const `glm::vec3` &position, const `glm::quat` &orientation)
- `std::shared_ptr< Sound >` **create** (`std::string` soundFile)
- `std::shared_ptr< Sound >` **play** (`std::string` soundFile, const `glm::vec3` &position, `bool` relative=false)
- `std::shared_ptr< Sound >` **play** (`const SoundProperties &soundProperties`, const `glm::vec3` &position, `bool` relative=false)
- void **activate** ()
- void **deactivate** ()

Static Public Member Functions

- static `SoundManager * current` ()

Protected Member Functions

- `sf::SoundBuffer * obtain` (`std::string` soundFile)
- void **cleanUp** ()

Protected Attributes

- `std::unordered_map< std::string, sf::SoundBuffer * >` **m_buffer**
- `std::list< std::shared_ptr< Sound > >` **m_sounds**
- int **m_nextCleanup**

Static Protected Attributes

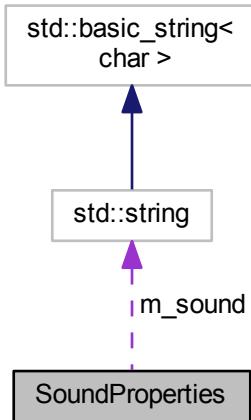
- static `SoundManager * s_current`

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

- `src/sound/soundmanager.h`
- `src/sound/soundmanager.cpp`

6.177 SoundProperties Class Reference

Collaboration diagram for SoundProperties:



Public Member Functions

- **SoundProperties** (const std::string &sound, float volume, float attenuation, bool repeating)
- const std::string & **sound** () const
- const float **volume** () const
- const float **attenuation** () const
- const bool **looping** () const

Static Public Member Functions

- static **SoundProperties fromProperties** (const std::string &prefix)

Protected Attributes

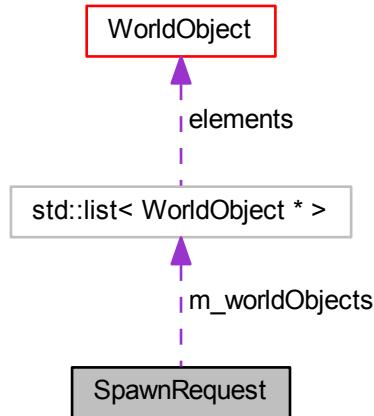
- std::string **m_sound**
- float **m_volume**
- float **m_attenuation**
- bool **m_looping**

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

- src/sound/soundproperties.h
- src/sound/soundproperties.cpp

6.178 SpawnRequest Class Reference

Collaboration diagram for SpawnRequest:



Public Member Functions

- `SpawnRequest (WorldObject *worldObject, bool deleteOnRejection=true)`
- `SpawnRequest (const std::list< WorldObject * > &worldObjects, bool deleteOnRejection=true)`
- `std::list< WorldObject * > & worldObjects ()`
- `bool deleteOnRejection () const`

Protected Attributes

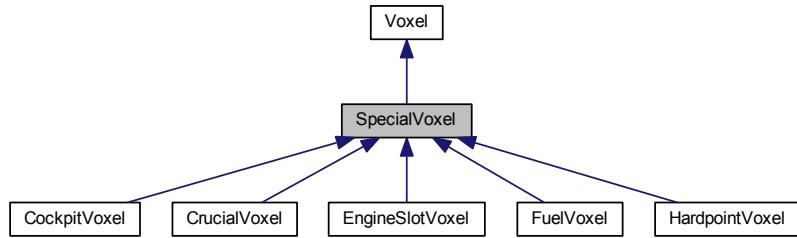
- `std::list< WorldObject * > m_worldObjects`
- `bool m_deleteOnRejection`

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

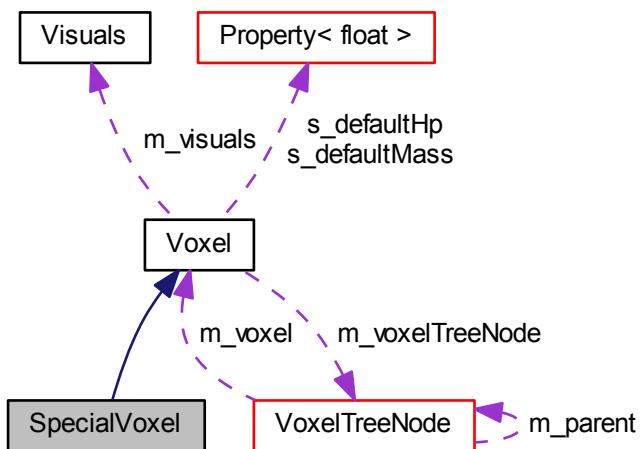
- `src/world/spawnrequest.h`
- `src/world/spawnrequest.cpp`

6.179 SpecialVoxel Class Reference

Inheritance diagram for SpecialVoxel:



Collaboration diagram for SpecialVoxel:



Public Member Functions

- **SpecialVoxel** (const `glm::ivec3 &gridCell`, int index, `uint32_t` color, float mass, float hp)
- int **index** () const

Protected Attributes

- int **m_index**

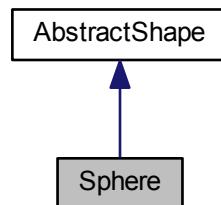
Additional Inherited Members

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

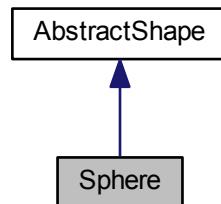
- src/voxel/specialvoxel.h
- src/voxel/specialvoxel.cpp

6.180 Sphere Class Reference

Inheritance diagram for Sphere:



Collaboration diagram for Sphere:



Public Member Functions

- **Sphere** (const glm::vec3 &position, float radius)
- float **radius** () const
- void **setRadius** (float radius)
- const glm::vec3 & **position** () const
- void **setPosition** (const glm::vec3 &position)
- bool **contains** (const [Sphere](#) &other) const
- virtual bool **intersects** (const [Sphere](#) &other) const override
- virtual bool **nearTo** (const [TAABB](#)< int > &aabb) const override
- virtual bool **containedBy** (const [TAABB](#)< int > &aabb) const override

Static Public Member Functions

- template<typename T >
static [Sphere](#) **containing** (const [TAABB](#)< T > &aabb)

Protected Attributes

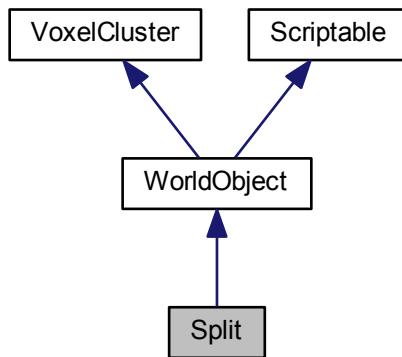
- `glm::vec3 m_position`
- `float m_radius`

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

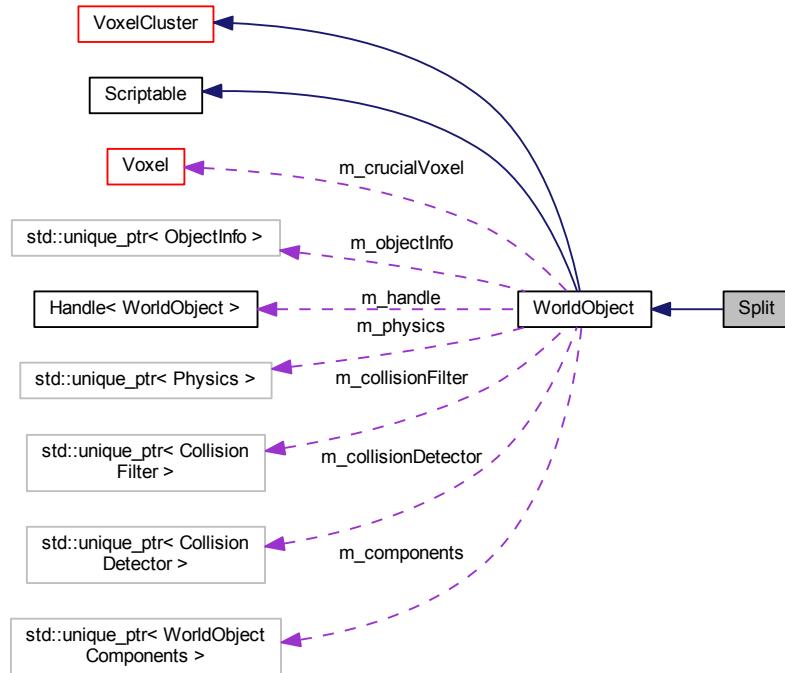
- `src/geometry/sphere.h`
- `src/geometry/sphere.cpp`
- `src/geometry/sphere.inl`

6.181 Split Class Reference

Inheritance diagram for Split:



Collaboration diagram for Split:



Public Member Functions

- **Split** (const [Transform](#) &transform)

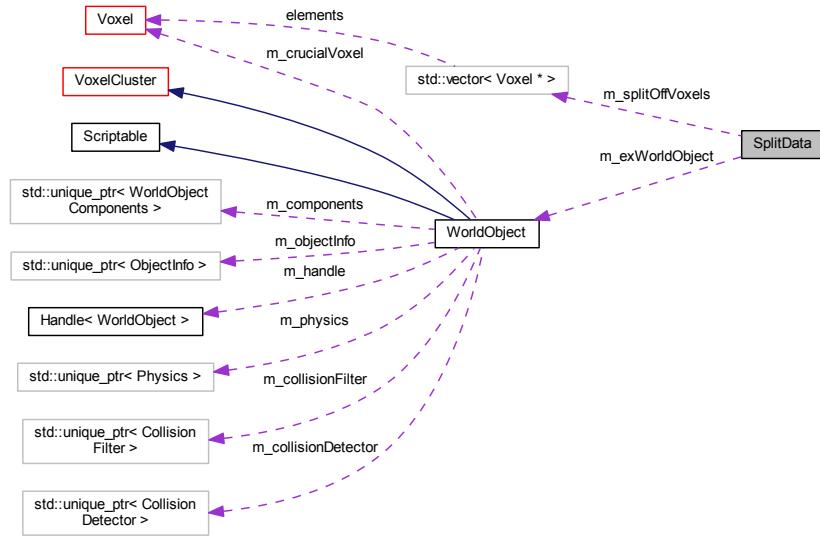
Additional Inherited Members

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

- [src/worldobject/split.h](#)
- [src/worldobject/split.cpp](#)

6.182 SplitData Class Reference

Collaboration diagram for SplitData:



Public Member Functions

- **SplitData** ([WorldObject](#) *worldObject)
- void **addVoxel** ([Voxel](#) *voxel)
- [WorldObject](#) * **exWorldObject** ()
- [std::vector<Voxel*>](#) **splitOffVoxels** ()
- [glm::ivec3](#) **llf** ()

Protected Attributes

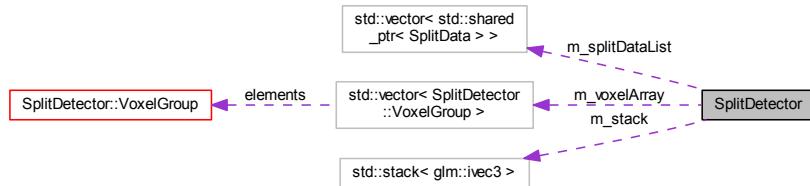
- [WorldObject](#) * **m_exWorldObject**
- [std::vector<Voxel*>](#) **m_splitOffVoxels**
- [glm::ivec3](#) **m_llf**

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

- src/world/helper/splittedata.h
- src/world/helper/splittedata.cpp

6.183 SplitDetector Class Reference

Collaboration diagram for SplitDetector:



Classes

- struct [VoxelGroup](#)

Public Member Functions

- void **searchSplitOffs** (std::list<[WorldObjectModification](#)> &worldObjectModifications)
- std::vector< std::shared_ptr<[SplitData](#) > > & **splitDataList** ()

Protected Member Functions

- void **clear** ()
- void **findSplits** ([WorldObject](#) *worldObject)
- void **createSplitData** ([WorldObject](#) *worldObject)
- void **init** ([WorldObject](#) *worldObject)
- int **address** (const glm::ivec3 &pos)
- [VoxelGroup](#) * **voxelGroup** (const glm::ivec3 &pos)
- void **fillColor** (const glm::ivec3 &start, int groupId)
- void **visit** (const glm::ivec3 &p)

Protected Attributes

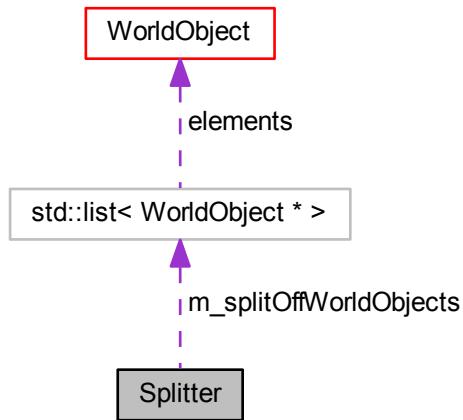
- std::vector< std::shared_ptr<[SplitData](#) > > **m_splitDataList**
- std::vector< [VoxelGroup](#) > **m_voxelArray**
- std::stack< glm::ivec3 > **m_stack**
- int **m_xy**
- int **m_x**
- glm::ivec3 **m_llf**
- glm::ivec3 **m_size**
- int **m_nextGroupId**

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

- src/world/handler/splitdetector.h
- src/world/handler/splitdetector.cpp

6.184 Splitter Class Reference

Collaboration diagram for Splitter:



Public Member Functions

- void **split** (std::vector< std::shared_ptr< [SplitData](#) >> &splits)
- std::list< [WorldObject](#) * > & **splitOffWorldObjects** ()

Protected Member Functions

- [WorldObject](#) * **createWorldObjectFromSplitOff** (std::shared_ptr< [SplitData](#) > split)
- void **removeExtractedVoxelsFromEx** (std::shared_ptr< [SplitData](#) > split)

Protected Attributes

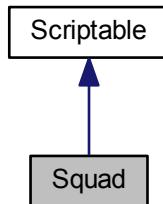
- std::list< [WorldObject](#) * > **m_splitOffWorldObjects**

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

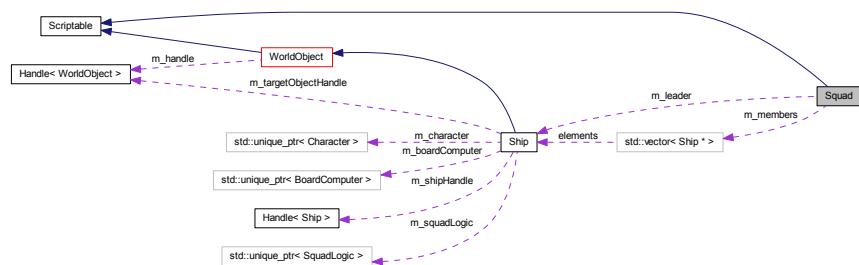
- src/world/handler/splitter.h
- src/world/handler/splitter.cpp

6.185 Squad Class Reference

Inheritance diagram for Squad:



Collaboration diagram for Squad:



Public Member Functions

- **Squad** (`Ship *leader=nullptr`)
- **Ship * leader ()**
- void **setLeader** (`Ship *leader`)
- `std::shared_ptr< AiGroupTask > task ()`
- void **setTask** (`std::shared_ptr< AiGroupTask > task`)
- const `std::vector< Ship * > & members ()`

Protected Member Functions

- void **onMemberJoin** (`Ship *member`)
- void **onMemberLeave** (`Ship *member`)
- `glm::vec3 formationPositionFor (Ship *member)`
- `glm::vec3 formationUpFor (Ship *member)`
- void **chooseNewLeader ()**
- `glm::vec3 calculateFormationPosition (Ship *member, int position)`

Protected Attributes

- `Ship * m_leader`
- `std::vector<Ship * > m_members`
- `std::shared_ptr<AiGroupTask > m_task`

Friends

- class `SquadLogic`

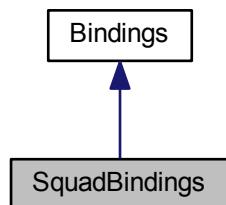
Additional Inherited Members

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

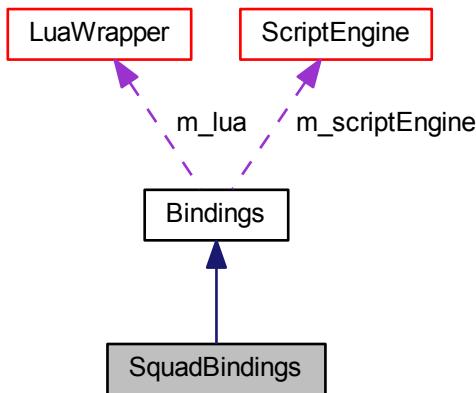
- `src/ai/squad.h`
- `src/ai/squad.cpp`

6.186 SquadBindings Class Reference

Inheritance diagram for SquadBindings:



Collaboration diagram for SquadBindings:



Public Member Functions

- `SquadBindings (GamePlayScript &script)`

Protected Member Functions

- `virtual void initialize ()`
- `apikey apiCreateSquad (apikey leader)`
- `int apiJoinSquad (apikey squad, apikey ship)`
- `int apiCreatePatrolWaypointsTask (apikey squad)`
- `int apiAddPatrolWaypointPoint (apikey task, const glm::vec3 &point)`
- `int apiCreateDefendAreaTask (apikey squad, const glm::vec3 &point, float range)`
- `int apiAddDefendAreaPoint (apikey task, const glm::vec3 &point)`

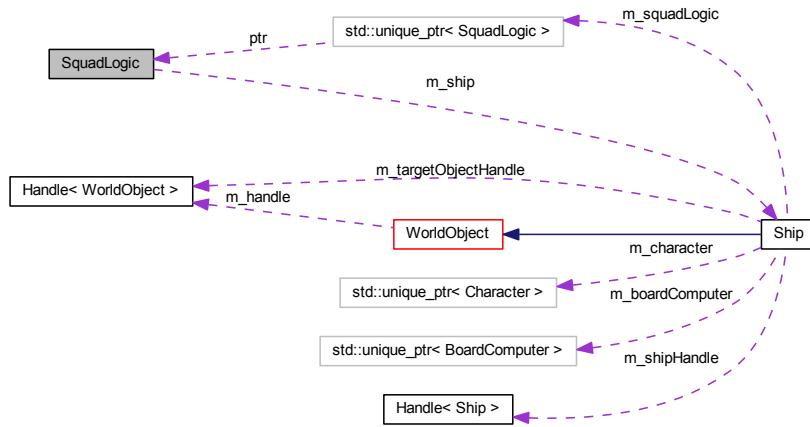
Additional Inherited Members

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

- `src/scripting/bindings/squadbindings.h`
- `src/scripting/bindings/squadbindings.cpp`

6.187 SquadLogic Class Reference

Collaboration diagram for SquadLogic:



Public Member Functions

- `SquadLogic (Ship &ship)`
- `Ship * ship ()`
- `void joinSquadOf (Ship *leader)`
- `void joinSquad (std::shared_ptr< Squad > squad)`
- `void leaveSquad ()`
- `std::shared_ptr< Squad > squad ()`
- `bool inSquad ()`
- `bool isSquadLeader ()`
- `glm::vec3 formationPosition ()`
- `glm::vec3 formationUp ()`

Protected Attributes

- `Ship & m_ship`
- `std::shared_ptr< Squad > m_squad`

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

- `src/ai/squadlogic.h`
- `src/ai/squadlogic.cpp`

6.188 Starfield::StarData Struct Reference

Public Attributes

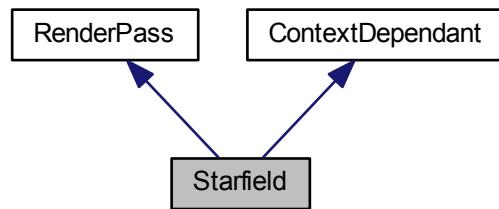
- `glm::vec3 pos`
- `float brightness`
- `float size`

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

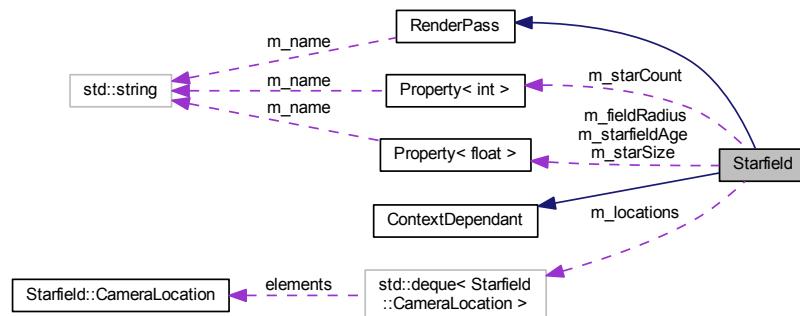
- src/display/rendering/starfield.h

6.189 Starfield Class Reference

Inheritance diagram for Starfield:



Collaboration diagram for Starfield:



Classes

- struct [CameraLocation](#)
- struct [StarData](#)

Public Member Functions

- virtual void [update](#) (float deltaSec, const glm::vec3 &cameraPosition)
- virtual void [apply](#) ([FrameBuffer](#) &frameBuffer, const [RenderMetaData](#) &metadata) override

Protected Member Functions

- void [createAndSetupShaders](#) ()

- void **createAndSetupGeometry** ()
- virtual void **beforeContextDestroy** () override
- virtual void **afterContextRebuild** () override
- void **createBinding** (int index, std::string name, int offset, int size)
- void **addLocation** (const [Camera](#) &camera, int side)
- glm::mat4 **getMatrixFromPast** (const [Camera](#) &camera, int side)
- void **cleanUp** (int side)

Protected Attributes

- std::deque< [CameraLocation](#) > **m_locations** [2]
- float **m_time**
- [Property](#)< float > **m_starfieldAge**
- [Property](#)< float > **m_starSize**
- [Property](#)< int > **m_starCount**
- [Property](#)< float > **m_fieldRadius**
- float **m_oldFieldRadius**
- glow::ref_ptr< glow::Program > **m_shaderProgram**
- glow::ref_ptr< glow::VertexArrayObject > **m_vertexArrayObject**
- glow::ref_ptr< glow::Buffer > **m_gpuBuffer**
- glow::Array< [StarData](#) > **m_cpuBuffer**

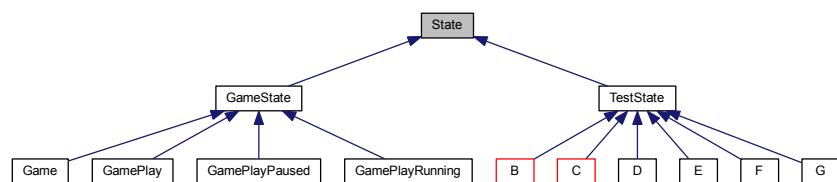
The documentation for this class was generated from the following files:

- src/display/rendering/starfield.h
- src/display/rendering/starfield.cpp

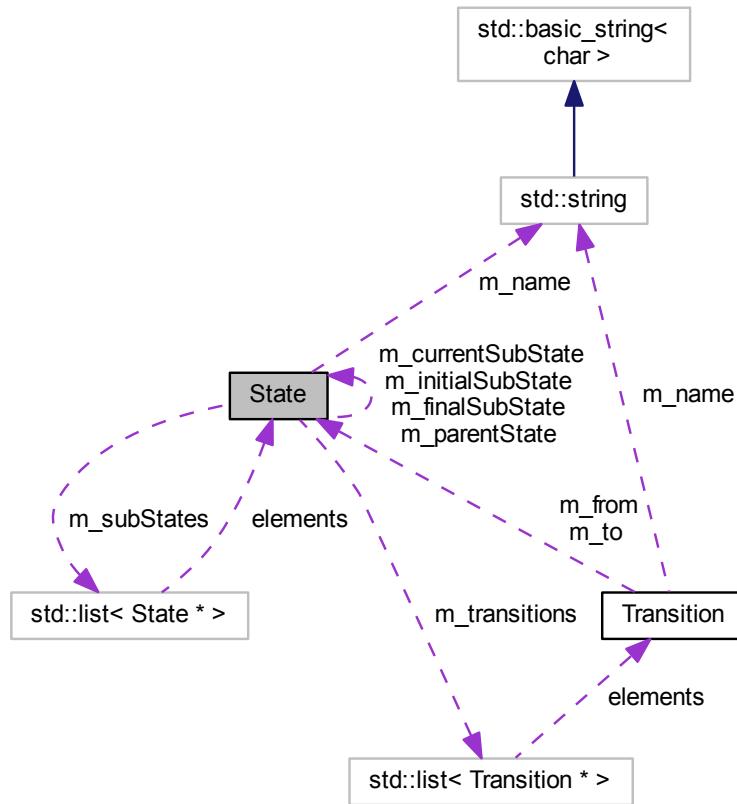
6.190 State Class Reference

```
#include <state.h>
```

Inheritance diagram for State:



Collaboration diagram for State:



Public Member Functions

- **State** (`State *parent=nullptr`)
- **State** (const `std::string &name, State *parent=nullptr`)
- const `std::string &name () const`
- void **setName** (const `std::string &name`)
- **State * parentState ()**
- const **State * parentState () const**
- **State * initialSubState ()**
- const **State * initialSubState () const**
- void **setInitialSubState** (`State *initialSubState`)
- **State * finalSubState ()**
- const **State * finalSubState () const**
- void **setFinalSubState** (`State *finalSubState`)
- **State * currentSubState ()**
- const **State * currentSubState () const**
- void **setCurrentSubState** (`State *substate`)
- bool **finished () const**
- `std::list<State *> & substates ()`
- const `std::list<State *> & substates () const`
- void **addSubState** (`State *state`)

- void **removeSubState** (*State *state*)
- std::list< *Transition ** > & **transitions** ()
- const std::list< *Transition ** > & **transitions** () const
- void **addTransition** (*Transition *transition*)
- void **removeTransition** (*Transition *transition*)
- virtual void **update** (float *deltaSec*)
- virtual void **onEntered** ()
- virtual void **onLeft** ()

Protected Member Functions

- *State * pathToDescendant* (*State *descendant*)
- void **transit** (*State *target*)
- void **leave** ()

Protected Attributes

- std::string **m_name**
- *State * m_parentState*
- std::list< *State ** > **m_subStates**
- std::list< *Transition ** > **m_transitions**
- *State * m_initialSubState*
- *State * m_finalSubState*
- *State * m_currentSubState*

6.190.1 Detailed Description

Abstract *State* that can function as a StateMachine This way you can nest states as you wish and still maintain the interface for the SubStates that you specify in ActualState (see [GameState](#) for a usage-example)

6.190.2 Member Function Documentation

6.190.2.1 void State::onEntered() [virtual]

Overrideable method that is called whenever a state or any of its substates come to be currentSubState This happens recursively up to the root-state

Reimplemented in [GamePlay](#), [GameState](#), [TestState](#), [GamePlayRunning](#), and [GamePlayPaused](#).

6.190.2.2 void State::onLeft() [virtual]

Overrideable method that is called whenever a state ceases to be currentSubState This happens recursively up to the root-state

Reimplemented in [GamePlay](#), [GameState](#), [TestState](#), [GamePlayRunning](#), and [GamePlayPaused](#).

6.190.2.3 *State * State::pathToDescendant(State * descendant)* [protected]

Returns direct substate having &descendant as a descendant Returns *m_self* if descendant is substate of this Returns nullptr if &descendant is no descendant of this

6.190.2.4 void State::transit (State * target) [protected]

Ensures the graph of m_currentSubState points from the root to targ calls `onLeft()` on every state left and `onEntered()` on every entered

6.190.2.5 void State::update (float deltaSec) [virtual]

Performs a [Transition](#) from the currentSubState, if such isPossible()

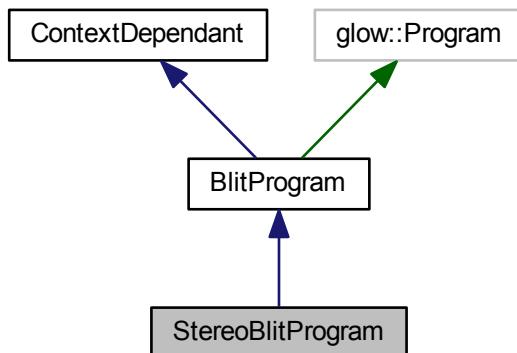
Reimplemented in [GamePlay](#), [Game](#), [GameState](#), [GamePlayRunning](#), and [GamePlayPaused](#).

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

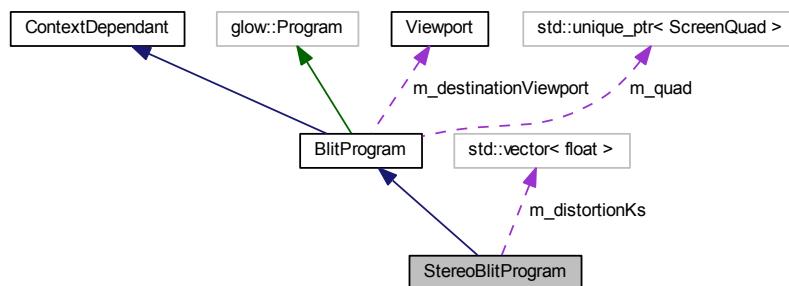
- src/utils/statemachine/state.h
- src/utils/statemachine/state.cpp

6.191 StereoBlitProgram Class Reference

Inheritance diagram for StereoBlitProgram:



Collaboration diagram for StereoBlitProgram:



Public Member Functions

- void **setDistortionKs** (std::vector< float > distortionKs)
- void **setDistortionScale** (float distortionScale)
- void **setLensCenter** (glm::vec2 lensCenter)
- virtual void **blit** () override

Protected Member Functions

- virtual void **initializeShaders** () override

Protected Attributes

- std::vector< float > **m_distortionKs**
- float **m_distortionScale**
- glm::vec2 **m_lensCenter**

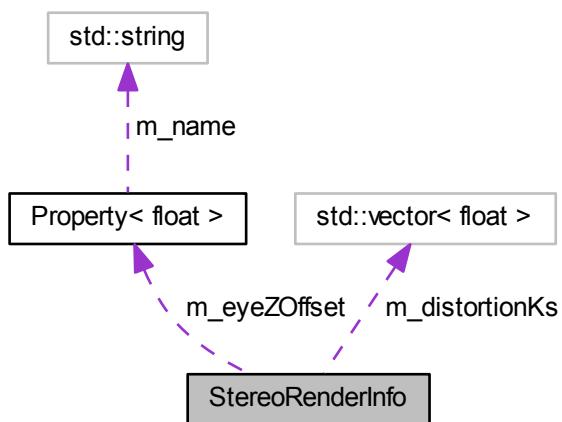
Additional Inherited Members

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

- src/programs/stereoblitprogram.h
- src/programs/stereoblitprogram.cpp

6.192 StereoRenderInfo Class Reference

Collaboration diagram for StereoRenderInfo:



Public Member Functions

- float **hScreenSize** () const
- float **vScreenSize** () const
- float **vScreenCenter** () const
- float **eyeToScreenDistance** () const
- float **lensSeparationDistance** () const
- float **interpupillaryDistance** () const
- int **hResolution** () const
- int **vResolution** () const
- float **distortionK** (int index) const
- std::vector< float > **distortionKs** () const
- float **distortionScale** () const
- float **fovy** () const
- glm::vec2 **leftEyeLensCenter** () const
- glm::vec2 **rightEyeLensCenter** () const
- glm::vec3 **leftEyeOffset** () const
- glm::vec3 **rightEyeOffset** () const
- glm::vec3 **leftEyeProjectionOffset** () const
- glm::vec3 **rightEyeProjectionOffset** () const

Static Public Member Functions

- static **StereoRenderInfo fromOVRInfo** (const OVR::HMDInfo &hmdInfo)
- static **StereoRenderInfo dummy** ()

Protected Member Functions

- float **projectionCenterOffset** () const

Protected Attributes

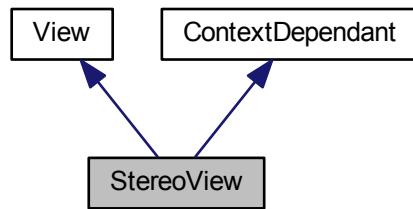
- float **m_hScreenSize**
- float **m_vScreenSize**
- float **m_vScreenCenter**
- float **m_eyeToScreenDistance**
- float **m_lensSeparationDistance**
- float **m_interpupillaryDistance**
- int **m_hResolution**
- int **m_vResolution**
- std::vector< float > **m_distortionKs**
- float **m_distortionScale**
- float **m_fovy**
- Property< float > **m_eyeZOffset**

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

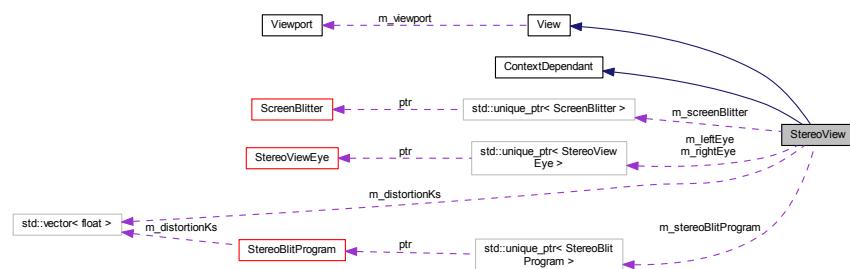
- src/display/stereorenderinfo.h
- src/display/stereorenderinfo.cpp

6.193 StereoView Class Reference

Inheritance diagram for StereoView:



Collaboration diagram for StereoView:



Public Member Functions

- **StereoView** (const [Viewport](#) &viewport, const [StereoRenderInfo](#) &stereoRenderInfo)
- virtual void **setViewport** (const [Viewport](#) &viewport) override
- virtual float **fovy** () const override
- virtual float **zNear** () const override
- virtual float **aspectRatio** () const override
- virtual void **draw** (const [Scene](#) &scene, const [CameraHead](#) &cameraHead) override

Protected Member Functions

- void **initialize** ()
- virtual void **beforeContextDestroy** () override
- virtual void **afterContextRebuild** () override

Protected Attributes

- [std::unique_ptr< StereoViewEye >](#) **m_leftEye**
- [std::unique_ptr< StereoViewEye >](#) **m_rightEye**
- [std::unique_ptr< ScreenBlitter >](#) **m_screenBlitter**

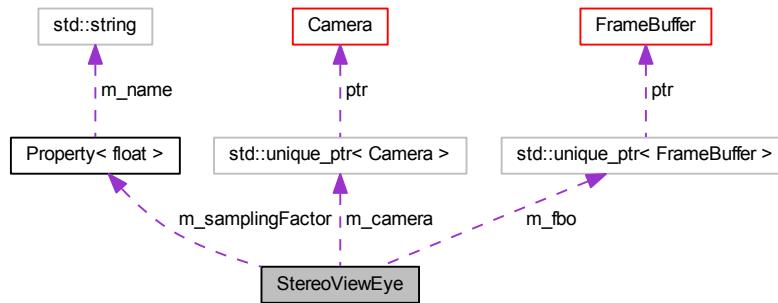
- std::unique_ptr< [StereoBlitProgram](#) > m_stereoBlitProgram
- glm::vec2 m_leftEyeLensCenter
- glm::vec2 m_rightEyeLensCenter
- std::vector< float > m_distortionKs
- float m_distortionScale

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

- src/display/stereoview.h
- src/display/stereoview.cpp

6.194 StereoViewEye Class Reference

Collaboration diagram for StereoViewEye:



Public Member Functions

- `StereoViewEye` (const `glm::ivec2` &viewportResolution, const [StereoRenderInfo](#) &stereoRenderInfo, EyeSide side)
- `FrameBuffer` & `fbo` ()
- const `Camera` & `camera` () const
- void `setViewportResolution` (const `glm::ivec2` &viewportResolution)
- void `draw` (const `Scene` &scene, const `CameraHead` &cameraHead)

Protected Attributes

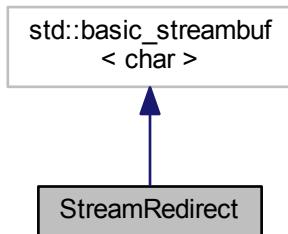
- `glm::vec3` `m_offset`
- `EyeSide` `m_side`
- `Property< float >` `m_samplingFactor`
- float `m_distortionScale`
- `glm::ivec2` `m_textureSize`
- `std::unique_ptr< Camera >` `m_camera`
- `std::unique_ptr< FrameBuffer >` `m_fbo`

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

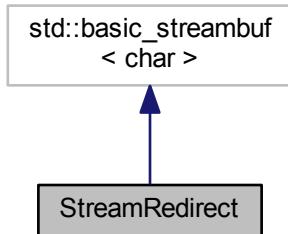
- src/display/stereovieweye.h
- src/display/stereovieweye.cpp

6.195 StreamRedirect Class Reference

Inheritance diagram for StreamRedirect:



Collaboration diagram for StreamRedirect:



Public Member Functions

- **StreamRedirect** (std::ostream &stream, [HUD](#) *hud, bool copy=false)

Protected Member Functions

- virtual int_type **overflow** (int_type v) override
- virtual std::streamsize **xspoutn** (const char *p, std::streamsize n) override

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

- [src/ui/streamredirect.h](#)
- [src/ui/streamredirect.cpp](#)

6.196 [snowhouse::Stringizer< glm::vec3 >](#) Struct Template Reference

Static Public Member Functions

- static std::string **ToString** (const glm::vec3 &value)

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

- test/bandit_extension/vec3helper.h

6.197 snowhouse::Stringizer< TAABB< T > > Struct Template Reference

Static Public Member Functions

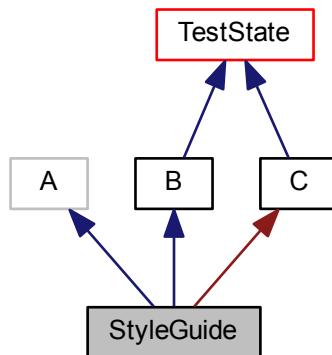
- static std::string **ToString** (const TAABB< T > &value)

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

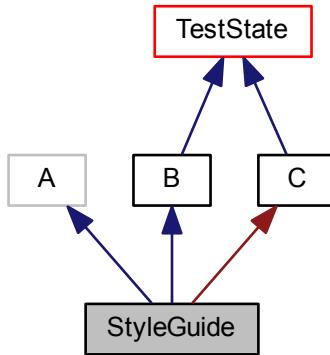
- test/bandit_extension/aabbhelper.h

6.198 StyleGuide Class Reference

Inheritance diagram for StyleGuide:



Collaboration diagram for StyleGuide:



Public Member Functions

- const Value & **value** () const
- void **setValue** (const Value &value)
- virtual void **update** () override

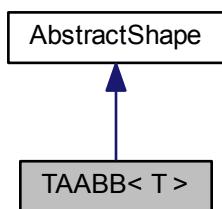
Additional Inherited Members

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

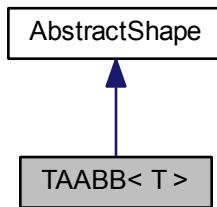
- doc/styleguide.h
- doc/styleguide.cpp

6.199 TAABB< T > Class Template Reference

Inheritance diagram for TAABB< T >:



Collaboration diagram for TAABB< T >:



Public Member Functions

- **TAABB** (const glm::detail::tvec3< T > &llf, const glm::detail::tvec3< T > &rub)
- template<typename OtherT >
 TAABB (const **TAABB**< OtherT > &other)
- const glm::detail::tvec3< T > & **llf** () const
- void **setLLF** (const glm::detail::tvec3< T > &llf)
- const glm::detail::tvec3< T > & **rub** () const
- void **setRUB** (const glm::detail::tvec3< T > &rub)
- T **axisMin** (Axis axis) const
- T **axisMax** (Axis axis) const
- glm::detail::tvec3< T > **middle** () const
- virtual T **extent** (Axis axis) const
- T **diameter** () const
- **TAABB**< T > **moved** (Axis axis, T delta) const
- **TAABB**< T > **moved** (const glm::detail::tvec3< T > &delta) const
- void **move** (Axis axis, T delta)
- void **move** (const glm::detail::tvec3< T > &delta)
- void **expand** (Axis axis, T delta)
- **TAABB**< T > **expanded** (Axis axis, T delta) const
- template<typename OtherT >
 bool **intersects** (const **TAABB**< OtherT > &other) const
- bool **contains** (const **TAABB**< T > &other) const
- template<typename OtherT >
 bool **contains** (const glm::detail::tvec3< OtherT > &vec) const
- virtual bool **intersects** (const **Sphere** &sphere) const override
- virtual bool **nearTo** (const **TAABB**< int > &other) const override
- virtual bool **containedBy** (const **TAABB**< int > &other) const override
- **TAABB**< T > **united** (const **TAABB**< T > &other) const
- void **unite** (const **TAABB**< T > &other)
- std::list< **TAABB**< T > > **split** (Axis axis) const
- void **split** (**TAABB**< T > &a, **TAABB**< T > &b, Axis axis) const
- std::list< **TAABB**< T > > **recursiveSplit** (int recursions, Axis axis) const
- bool **operator==** (const **TAABB**< T > &other) const
- void **extend** (const glm::detail::tvec3< T > &point)

Static Public Member Functions

- static **TAABB**< float > **containing** (const **Sphere** &sphere)

Protected Attributes

- `glm::detail::tvec3< T > m_llf`
- `glm::detail::tvec3< T > m_rub`

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

- `src/geometry/aabb.h`
- `src/geometry/aabb.inl`

6.200 TargetSelector Class Reference

Public Member Functions

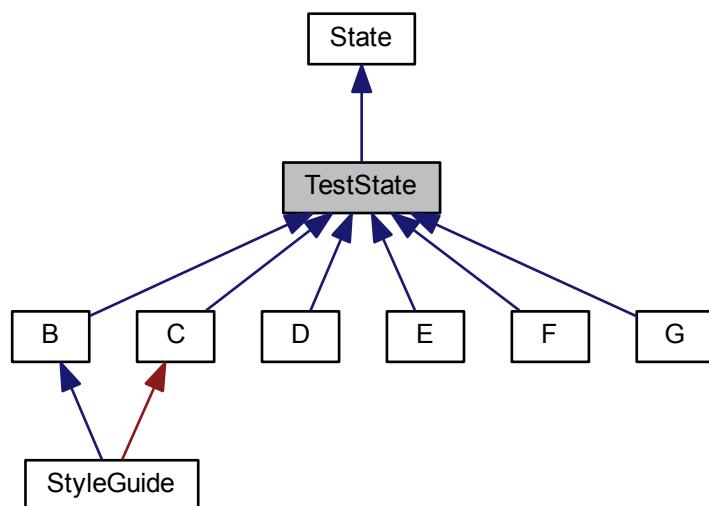
- `TargetSelector (Player *player)`
- `void selectTarget (bool next)`

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

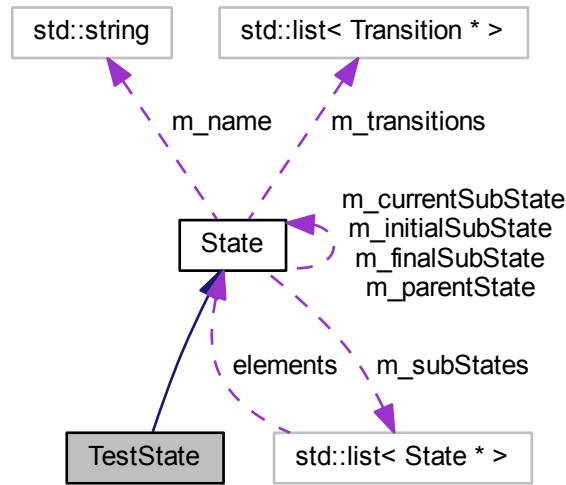
- `src/ui/targetselector.h`
- `src/ui/targetselector.cpp`

6.201 TestState Class Reference

Inheritance diagram for TestState:



Collaboration diagram for TestState:



Public Member Functions

- **TestState** (const std::string &name, [State](#) *parent)
- virtual void [onLeft\(\)](#) override
- virtual void [onEntered\(\)](#) override

Additional Inherited Members

6.201.1 Member Function Documentation

6.201.1.1 virtual void [TestState::onEntered\(\)](#) [inline], [override], [virtual]

Overrideable method that is called whenever a state or any of its substates come to be currentSubState This happens recursively up to the root-state

Reimplemented from [State](#).

6.201.1.2 virtual void [TestState::onLeft\(\)](#) [inline], [override], [virtual]

Overrideable method that is called whenever a state ceases to be currentSubState This happens recursively up to the root-state

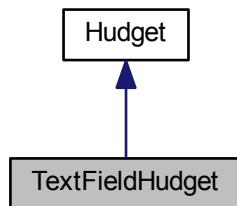
Reimplemented from [State](#).

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

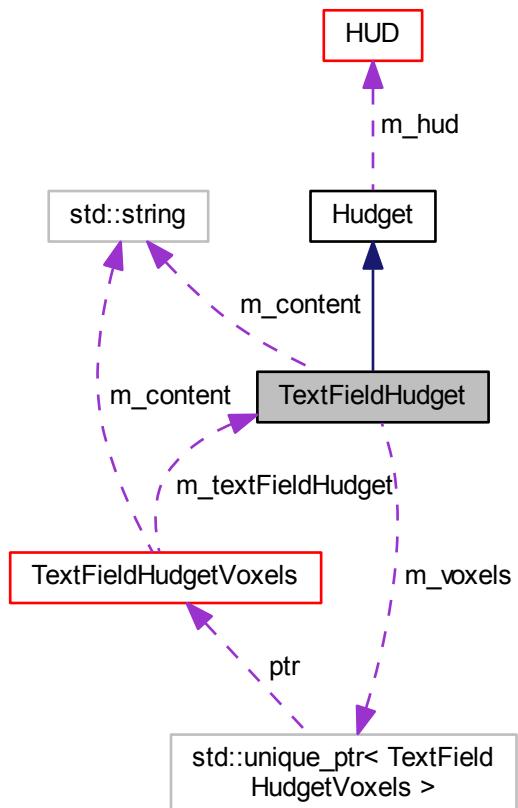
- test/statemachine/teststatemachine.cpp

6.202 TextFieldHudget Class Reference

Inheritance diagram for TextFieldHudget:



Collaboration diagram for TextFieldHudget:



Public Member Functions

- **TextFieldHudget** (`HUD *hud, glm::vec3 direction, float scale=0.5f, std::string content="", FontSize fontSize=FontSize::SIZE5x7)`
- void **setContent** (std::string content)
- virtual void **update** (float deltaSec) override
- virtual void **draw** () override
- virtual bool **isAt** (const `Ray &ray`) const override
- virtual void **onClick** (ClickType clickType) override

Protected Attributes

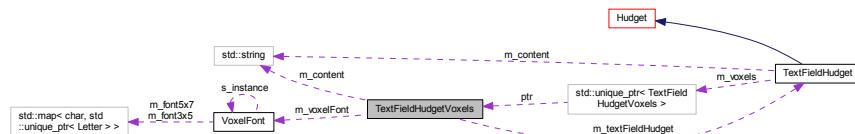
- std::string **m_content**
- std::unique_ptr
`<TextFieldHudgetVoxels>` **m_voxels**

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

- src/ui/hud/textfieldhudget.h
- src/ui/hud/textfieldhudget.cpp

6.203 TextFieldHudgetVoxels Class Reference

Collaboration diagram for `TextFieldHudgetVoxels`:



Public Member Functions

- **TextFieldHudgetVoxels** (`TextFieldHudget *textFieldHudget, glm::vec3 direction, float scale=0.5f, std::string content="", FontSize fontSize=FontSize::SIZE5x7)`
- void **setContent** (std::string content)
- void **update** (float deltaSec)
- void **draw** ()
- virtual bool **isAt** (const `Ray &ray`) const
- float **width** ()
- float **height** ()
- float **scale** ()

Protected Member Functions

- const `glm::vec3 upperLeft () const`
- const `glm::vec3 lowerLeft () const`
- const `glm::vec3 upperRight () const`
- const `glm::vec3 lowerRight () const`
- `glm::vec3 worldPosition () const`
- `glm::quat worldOrientation () const`
- const `glm::vec3 offsetToCenter (bool upper, bool left) const`

Protected Attributes

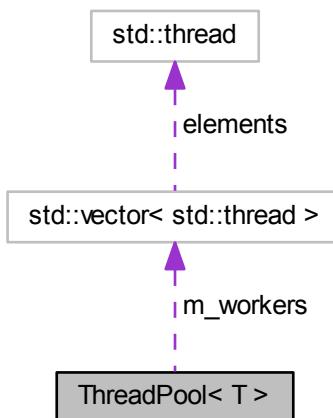
- FontSize **m_fontSize**
- **TextFieldHudget** * **m_textFieldHudget**
- std::string **m_content**
- **VoxelFont** * **m voxelFont**
- glm::vec3 **m_direction**
- float **m_width**
- float **m_height**
- float **m_scale**
- float **m_offset**

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

- src/ui/hud/textfieldhudgetvoxels.h
- src/ui/hud/textfieldhudgetvoxels.cpp

6.204 ThreadPool< T > Class Template Reference

Collaboration diagram for ThreadPool< T >:



Public Member Functions

- **ThreadPool** (int threadcount=4, int chunksize=100)
- void **map** (std::function< void(T &) > function, std::vector< T > &data)
- void **map** (std::function< void(T &) > function, std::vector< T > &data, int start, int end)

Protected Member Functions

- void **startWorkers** ()
- void **worker** ()
- int **getTask** ()

Protected Attributes

- std::vector< T > * **m_tasks**
- std::function< void(T &) > **m_function**
- std::vector< std::thread > **m_workers**
- std::condition_variable **m_startSignal**
- std::condition_variable **m_stopSignal**
- std::mutex **m_mutex**
- std::atomic_int **m_currentIndex**
- int **m_endIndex**
- int **m_chunksize**
- bool **m_exit**
- std::atomic_int **m_startWorkers**
- std::atomic_int **m_stoppedWorkers**

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

- src/utils/threadpool.h
- src/utils/threadpool.inl

6.205 TimedTask Class Reference

Public Member Functions

- **TimedTask** (std::chrono::duration< float > intervalSecs)
- std::chrono::duration< float > **intervalSecs** () const
- void **setIntervalSecs** (std::chrono::duration< float > intervalSecs)
- virtual bool **isDue** () const final
- virtual void **nudge** () final
- virtual void **exec** ()

Protected Attributes

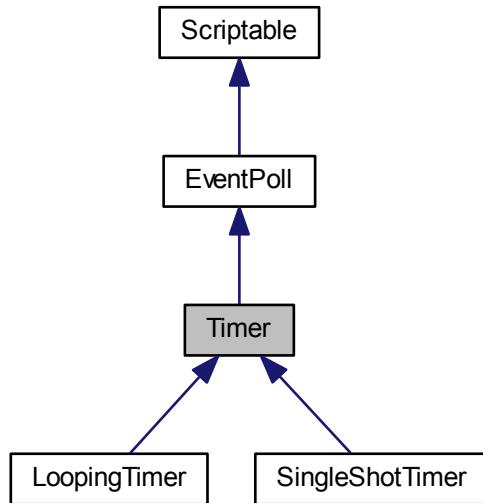
- std::chrono::duration< float > **m_intervalSecs**
- std::chrono::high_resolution_clock::time_point **m_lastExecution**

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

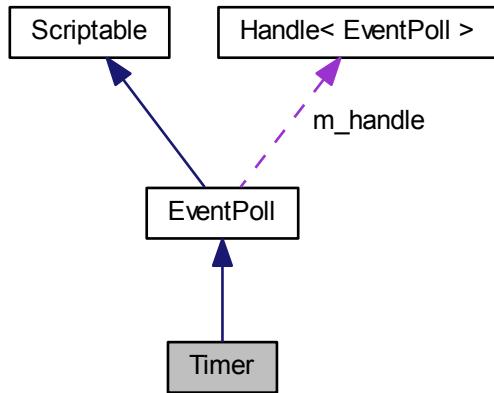
- src/utils/timedtask.h
- src/utils/timedtask.cpp

6.206 Timer Class Reference

Inheritance diagram for Timer:



Collaboration diagram for Timer:



Public Member Functions

- **Timer** (float interval, const std::function< void()> &callback)
- virtual void **update** (float deltaSec) override

Protected Member Functions

- virtual bool **poll** () override
- virtual void **specialOnCallback** () override

Protected Attributes

- float **m_interval**
- float **m_countdown**
- int **m_scheduledCalls**

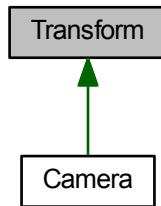
Additional Inherited Members

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

- src/events/timer.h
- src/events/timer.cpp

6.207 Transform Class Reference

Inheritance diagram for Transform:



Public Member Functions

- **Transform** (glm::vec3 center=glm::vec3(0), float scale=1.0)
- **Transform** (const [Transform](#) &transform, const glm::vec3 &positionDelta, const glm::quat &orientationDelta)
- const glm::vec3 & **position** () const
- void **setPosition** (const glm::vec3 &pos)
- const glm::quat & **orientation** () const
- void **setOrientation** (const glm::quat &quat)
- const glm::vec3 & **center** () const
- void **setCenter** (const glm::vec3 ¢er)
- void **setCenterAndAdjustPosition** (const glm::vec3 &newCenter)
- float **scale** () const
- void **setScale** (float scale)
- void **move** (const glm::vec3 &dist)
- void **moveWorld** (const glm::vec3 &dist)
- void **rotate** (const glm::quat &qrot)

- void **rotateWorld** (const glm::quat &qrot)
- bool **operator==** (const [Transform](#) &other) const
- bool **operator!=** (const [Transform](#) &other) const
- const glm::mat4 **matrix** () const
- glm::vec3 **applyTo** (const glm::vec3 &vertex) const
- glm::vec3 **inverseApplyTo** (const glm::vec3 &vertex) const

Protected Attributes

- glm::vec3 **m_position**
- glm::quat **m_orientation**
- glm::vec3 **m_center**
- float **m_scale**

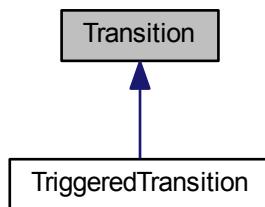
The documentation for this class was generated from the following files:

- src/geometry/transform.h
- src/geometry/transform.cpp

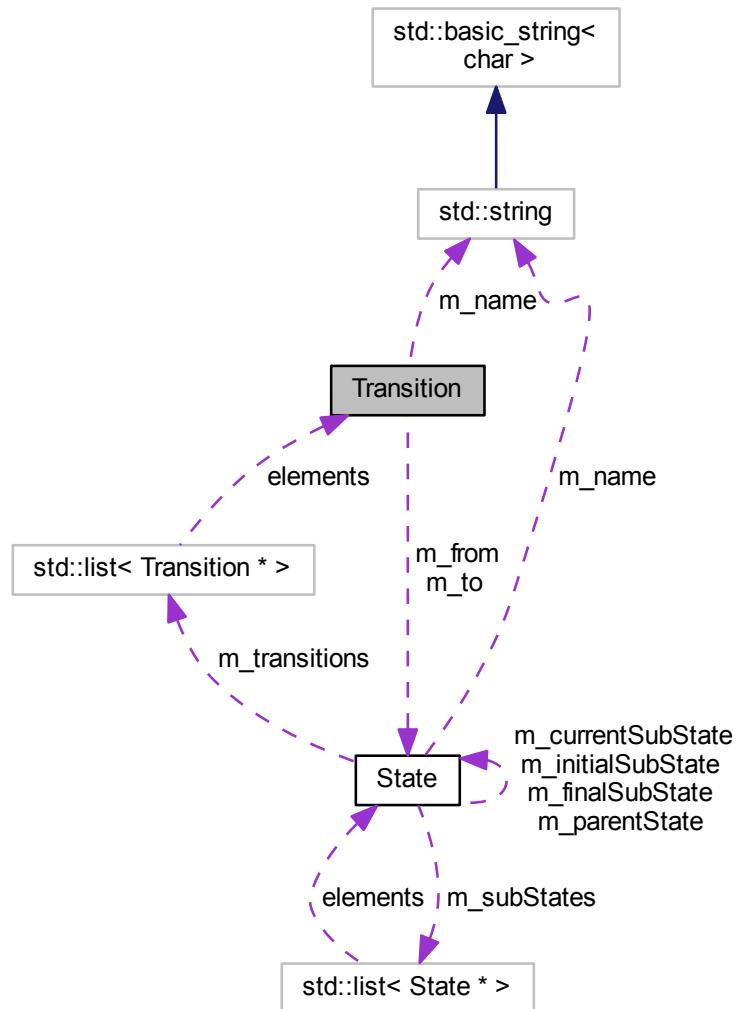
6.208 Transition Class Reference

```
#include <transition.h>
```

Inheritance diagram for Transition:



Collaboration diagram for Transition:



Public Member Functions

- **Transition** (`State *from, State *to`)
 - **Transition** (`State *from, State *to, const std::string &name`)
 - `const std::string & name () const`
 - `void setName (const std::string &name)`
 - **State * from ()**
 - **State * to ()**
 - `virtual bool isPossible () const =0`
 - `virtual void onPerformed ()`

Protected Attributes

- State * **m_from**

- `State * m_to`
- `std::string m_name`

6.208.1 Detailed Description

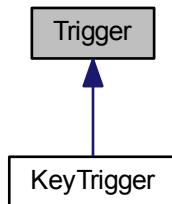
Abstract base class for a transition from one state to another. Derived classes need to implement `isPossible()`. The actual transition is performed by the `update()` of the parentstate of from-. Transitions register themselves at from, no need to call `addTransition()` on the `State`. to isn't informed it's "target" of a transition. Note that Transitions can go from any `State` to any other `State`, no matter which `State` they are nested in.

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

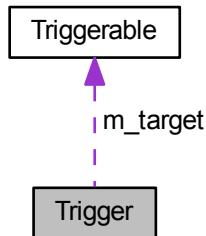
- `src/utils/statemachine/transition.h`
- `src/utils/statemachine/transition.cpp`

6.209 Trigger Class Reference

Inheritance diagram for Trigger:



Collaboration diagram for Trigger:



Public Member Functions

- `Trigger (Triggerable *target=nullptr)`

- `Triggerable * target ()`
- `void setTarget (Triggerable *target)`
- `void trigger ()`
- `virtual void update (float deltaSec)`

Protected Attributes

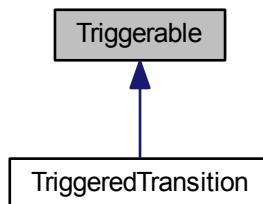
- `Triggerable * m_target`

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

- `src/utils/statemachine/trigger.h`
- `src/utils/statemachine/trigger.cpp`

6.210 Triggerable Class Reference

Inheritance diagram for Triggerable:



Public Member Functions

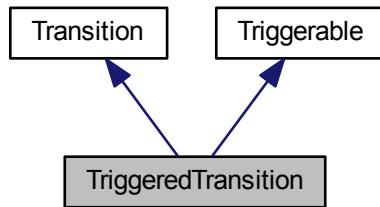
- `virtual void trigger ()=0`

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

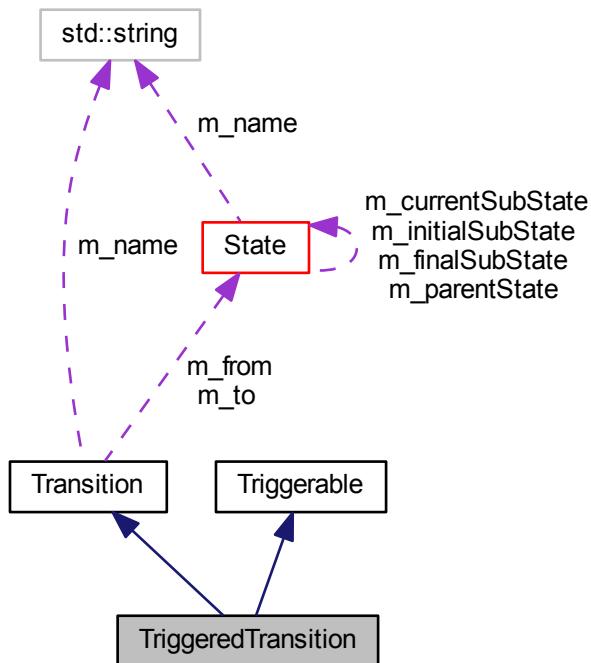
- `src/utils/statemachine/triggerable.h`

6.211 TriggeredTransition Class Reference

Inheritance diagram for TriggeredTransition:



Collaboration diagram for TriggeredTransition:



Public Member Functions

- **TriggeredTransition** (`State *from, State *to`)
- **TriggeredTransition** (`State *from, State *to, const std::string &name`)
- `virtual bool isPossible () const override`
- `virtual void trigger () override`
- `virtual void onPerformed () override`

Protected Attributes

- bool **m_triggered**

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

- src/utils/statemachine/triggeredtransition.h
- src/utils/statemachine/triggeredtransition.cpp

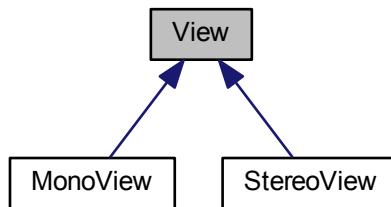
6.212 glow::Uniform< T > Class Template Reference

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

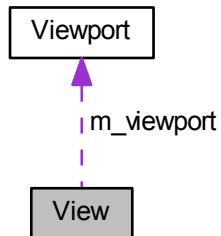
- src/voxel/voxelrenderer.h

6.213 View Class Reference

Inheritance diagram for View:



Collaboration diagram for View:



Public Member Functions

- **View** (const [Viewport](#) &viewport)
- virtual float **fovy** () const =0
- virtual float **zNear** () const =0
- virtual float **aspectRatio** () const =0
- virtual void **setViewport** (const [Viewport](#) &viewport)
- virtual void **draw** (const [Scene](#) &scene, const [CameraHead](#) &cameraHead)=0

Protected Attributes

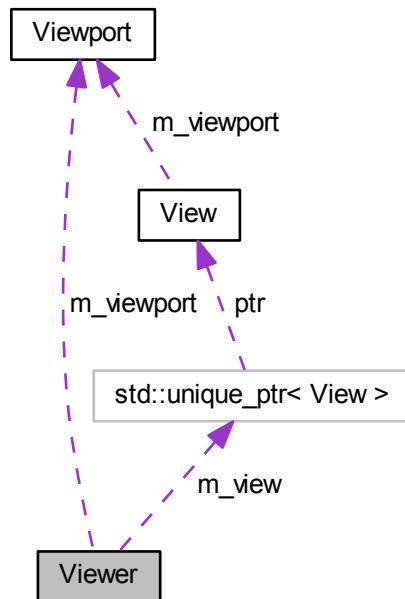
- [Viewport](#) **m_viewport**

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

- src/display/view.h
- src/display/view.cpp

6.214 Viewer Class Reference

Collaboration diagram for Viewer:



Public Member Functions

- **Viewer** (const [Viewport](#) &viewport)
- const [View](#) & **view** () const
- void **setViewport** (const [Viewport](#) &viewport)

- void **switchToMonoView** ()
- void **switchToStereoView** (const **StereoRenderInfo** &stereoRenderInfo)
- void **update** (float deltaSec)
- void **draw** (const **Scene** &scene, const **CameraHead** &cameraHead)

Protected Attributes

- std::unique_ptr<**View**> **m_view**
- **Viewport** **m_viewport**

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

- src/display/viewer.h
- src/display/viewer.cpp

6.215 Viewport Class Reference

Public Member Functions

- **Viewport** (int x, int y, int width, int height)
- int **x** () const
- int **y** () const
- int **width** () const
- int **height** () const
- glm::vec2 **offset** () const
- glm::vec2 **scale** () const

Protected Attributes

- int **m_x**
- int **m_y**
- int **m_width**
- int **m_height**

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

- src/geometry/viewport.h
- src/geometry/viewport.cpp

6.216 Visuals Class Reference

Public Member Functions

- **Visuals** (uint32_t color, float emissiveness)
- uint32_t **color** () const
- void **setColor** (uint32_t color)
- float **emissiveness** () const
- void **setEmissiveness** (float emissiveness)

Static Public Member Functions

- static **Visuals fromProperties** (const std::string &prefix)

Protected Attributes

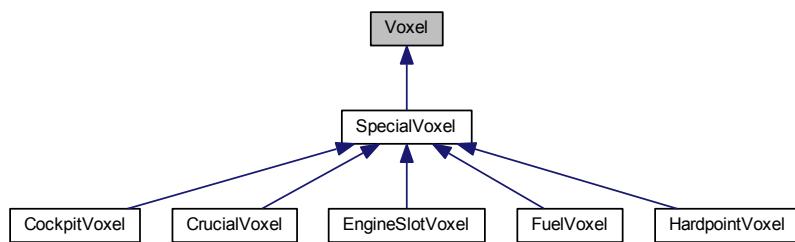
- uint32_t **m_color**
- float **m_emissiveness**

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

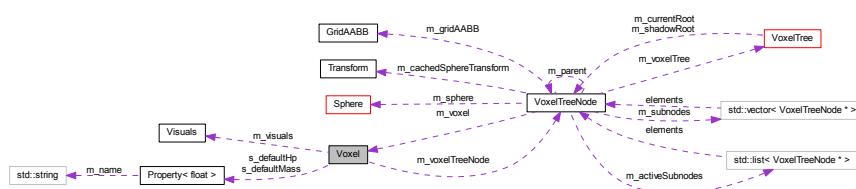
- src/display/rendering/visuals.h
- src/display/rendering/visuals.cpp

6.217 Voxel Class Reference

Inheritance diagram for Voxel:



Collaboration diagram for Voxel:



Public Member Functions

- **Voxel** (const glm::ivec3 &gridCell, uint32_t color=0xFFFFFFFF, float mass=defaultMass(), float hp=defaultHp(), float emissiveness=0)
- **Voxel** (const **Voxel** &other)
- const glm::ivec3 & **gridCell** () const
- glm::vec3 **position** () const
- **VoxelTreeNode** * **voxelTreeNode** ()
- void **setVoxelTreeNode** (**VoxelTreeNode** *voxelTreeNode)
- virtual void **addToCluster** (**VoxelCluster** *cluster)

- virtual void **addToObject** ([WorldObject](#) *object)
- virtual [Visuals](#) **visuals** () const
- float **hp** () const
- void **applyDamage** (float deltaHp)
- virtual float **damageForwardingDestructionDamage** ()
- float **normalizedMass** () const
- virtual void **onRemoval** ()
- virtual void **onDestruction** ()

Static Protected Member Functions

- static float **defaultMass** ()
- static float **defaultHp** ()

Protected Attributes

- `glm::ivec3 m_gridCell`
- `VoxelTreeNode * m_voxelTreeNode`
- [Visuals](#) **m_visuals**
- float **m_hp**
- float **m_normalizedMass**

Static Protected Attributes

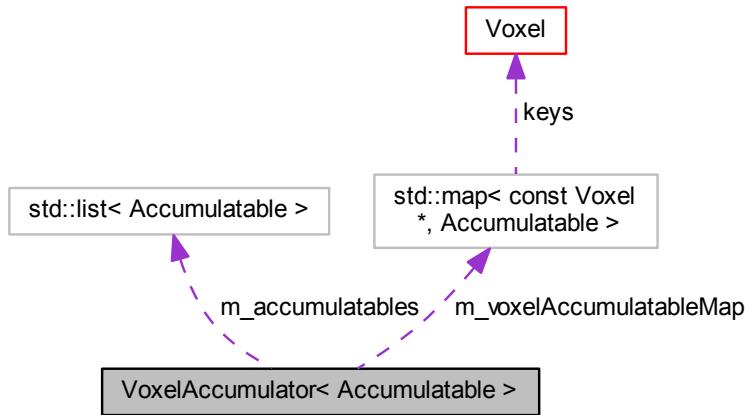
- static `Property< float > * s_defaultMass`
- static `Property< float > * s_defaultHp`

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

- `src/voxel/voxel.h`
- `src/voxel/voxel.cpp`

6.218 VoxelAccumulator< Accumulatable > Class Template Reference

Collaboration diagram for VoxelAccumulator< Accumulatable >:



Public Member Functions

- void **clear** ()
- void **parse** (const Accumulatable &accumulatable)
- void **parse** (const std::list< Accumulatable > &accumulatables)
- void **dontAffect** (const std::list< Voxel * > &voxels)
- std::list< Accumulatable > & **accumulatables** ()

Protected Attributes

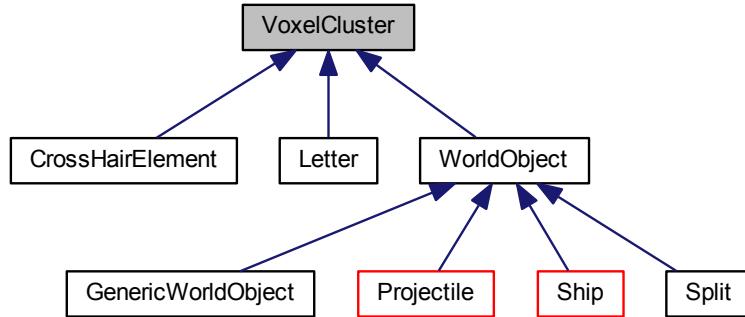
- std::map< const Voxel *, Accumulatable > **m_voxelAccumulatableMap**
- std::list< Accumulatable > **m_accumulatables**

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

- src/world/helper/voxelaccumulator.h
- src/world/helper/voxelaccumulator.inl

6.219 VoxelCluster Class Reference

Inheritance diagram for VoxelCluster:



Collaboration diagram for VoxelCluster:



Public Member Functions

- **VoxelCluster** (float scale)
- **VoxelClusterBounds & bounds ()**
- **Transform & transform ()**
- const **Transform & transform () const**
- void **setTransform** (const **Transform** &transform)
- const **glm::vec3 & position () const**
- const **glm::quat & orientation () const**
- **Voxel * voxel** (const **glm::ivec3 &position**)
- const **Voxel * voxel** (const **glm::ivec3 &position**) const
- virtual void **addVoxel** (**Voxel** *voxel)
- virtual void **removeVoxel** (**Voxel** *voxel)
- const std::unordered_map<glm::ivec3, **Voxel** *> & **voxelMap () const**
- int **voxelCount () const**
- **VoxelRenderData * voxelRenderData ()**
- virtual float **emissiveness () const**

Protected Attributes

- std::unordered_map<glm::ivec3, **Voxel** *> **m_voxels**
- std::unique_ptr<**VoxelRenderData**> **m_voxelRenderData**

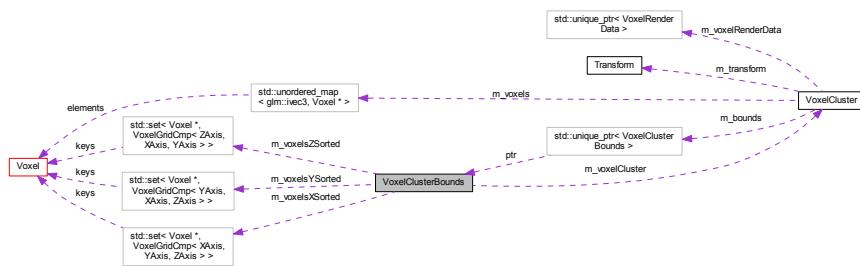
- std::unique_ptr<VoxelClusterBounds> **m_bounds**
- Transform **m_transform**

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

- src/voxel/voxelcluster.h
- src/voxel/voxelcluster.cpp

6.220 VoxelClusterBounds Class Reference

Collaboration diagram for VoxelClusterBounds:



Public Member Functions

- **VoxelClusterBounds** (VoxelCluster *voxelCluster)
- void **addVoxel** (Voxel *voxel)
- void **removeVoxel** (Voxel *voxel)
- const GridAABB & **minimalGridAABB** ()
- const Sphere & **minimalGridSphere** ()
- const IAABB & **aabb** ()
- IAABB **aabb** (const Transform &transform)
- const Sphere & **sphere** ()
- Sphere **sphere** (const Transform &transform)

Protected Member Functions

- void **calculateMinimalGridAABB** ()
- void **calculateMinimalGridSphere** ()
- const IAABB **calculateAABB** (const Transform &transform)

Protected Attributes

- VoxelCluster * **m_voxelCluster**
- std::set< Voxel
*, VoxelGridCmp< XAxis, YAxis,
ZAxis >> **m_voxelsXSorted**
- std::set< Voxel
*, VoxelGridCmp< YAxis, XAxis,
ZAxis >> **m_voxelsYSorted**

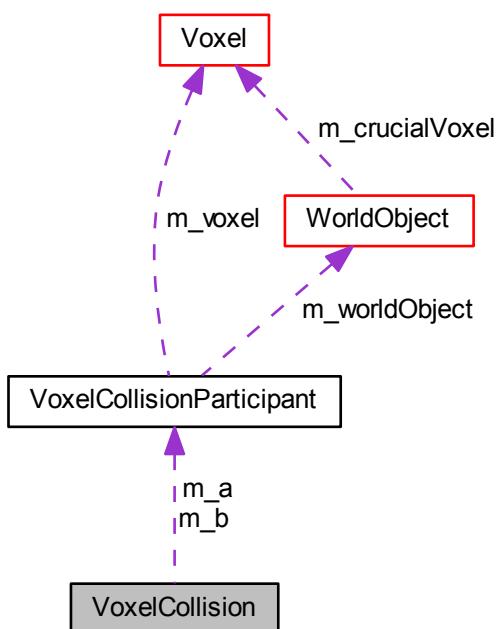
- std::set< [Voxel](#)
*, [VoxelGridCmp](#)< ZAxis, XAxis,
YAxis > > **m_voxelsZSorted**

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

- src/voxel/voxelclusterbounds.h
- src/voxel/voxelclusterbounds.cpp

6.221 VoxelCollision Class Reference

Collaboration diagram for VoxelCollision:



Public Member Functions

- **VoxelCollision** (const [VoxelCollisionParticipant](#) &a, const [VoxelCollisionParticipant](#) &b)
- [VoxelCollisionParticipant](#) & **a** ()
- [VoxelCollisionParticipant](#) & **b** ()

Protected Attributes

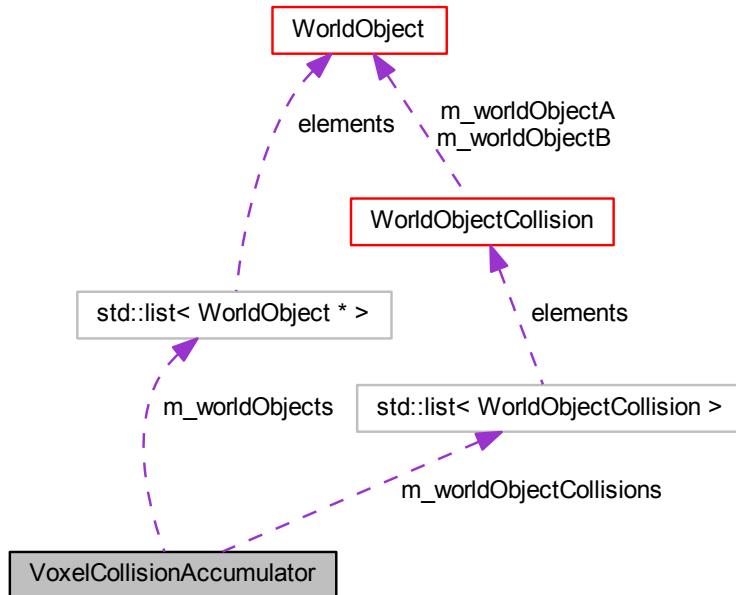
- [VoxelCollisionParticipant](#) **m_a**
- [VoxelCollisionParticipant](#) **m_b**

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

- src/collision/voxelcollision.h
- src/collision/voxelcollision.cpp

6.222 VoxelCollisionAccumulator Class Reference

Collaboration diagram for VoxelCollisionAccumulator:



Public Member Functions

- void **parse** (`std::list< VoxelCollision > &voxelCollisions)`
- void **applyOnCollisionHooks** ()
- `std::list< WorldObjectCollision > & worldObjectCollisions ()`
- `std::list< WorldObject * > & worldObjects ()`

Protected Attributes

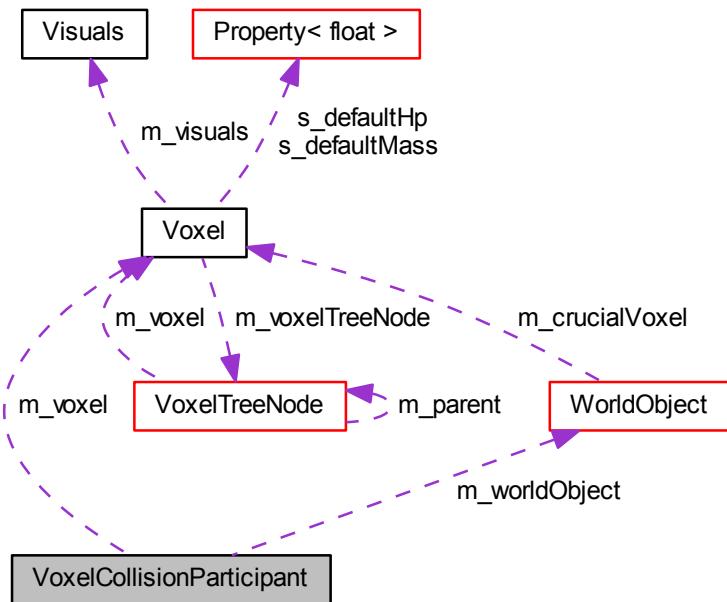
- `std::list< WorldObjectCollision > m_worldObjectCollisions`
- `std::list< WorldObject * > m_worldObjects`

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

- `src/world/handler/voxelcollisionaccumulator.h`
- `src/world/handler/voxelcollisionaccumulator.cpp`

6.223 VoxelCollisionParticipant Class Reference

Collaboration diagram for VoxelCollisionParticipant:



Public Member Functions

- `VoxelCollisionParticipant (WorldObject *worldObject, Voxel *voxel)`
- `WorldObject * worldObject ()`
- `Voxel * voxel ()`

Protected Attributes

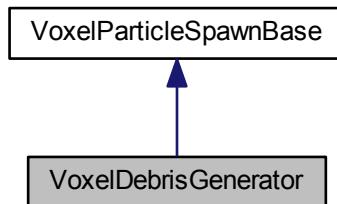
- `WorldObject * m_worldObject`
- `Voxel * m_voxel`

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

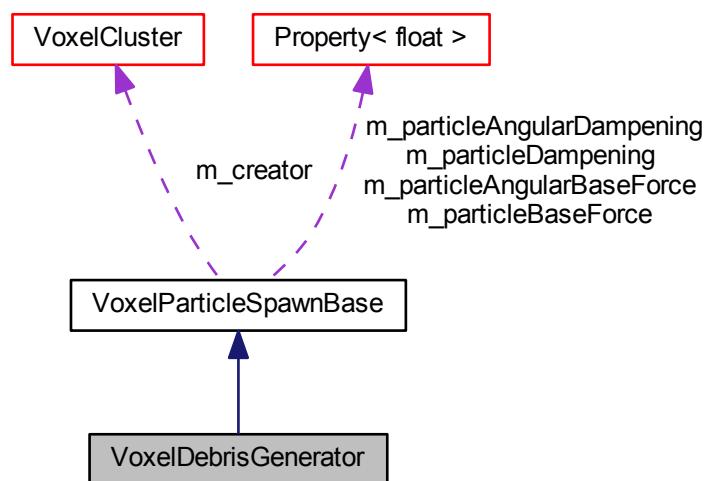
- `src/collision/voxelcollision.h`
- `src/collision/voxelcollision.cpp`

6.224 VoxelDebrisGenerator Class Reference

Inheritance diagram for VoxelDebrisGenerator:



Collaboration diagram for VoxelDebrisGenerator:



Public Member Functions

- **VoxelDebrisGenerator** (const [VoxelCluster](#) *creator)
- void **setOrientation** (const [glm::quat](#) &orientation)
- void **setDensity** (int density)
- void **setSpawnProbability** (float spawnProbability)
- void **spawn** ()

Protected Member Functions

- float **createScale** ()

Protected Attributes

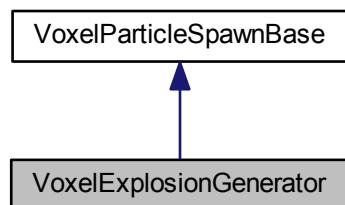
- `glm::quat m_orientation`
- `int m_density`
- `float m_spawnProbability`

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

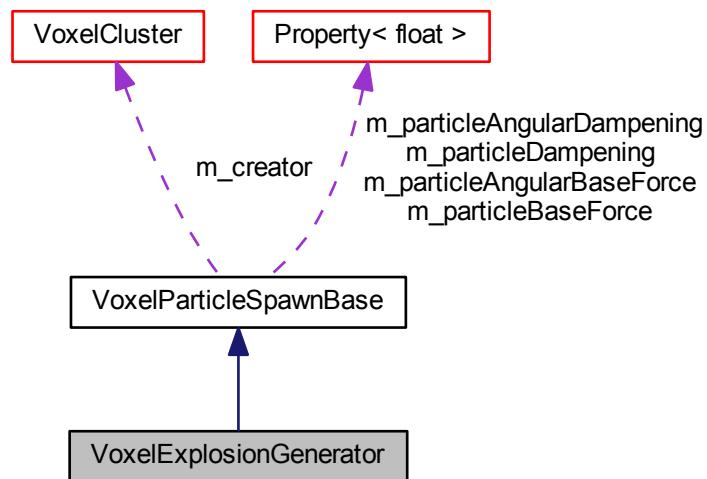
- `src/voxeleffect/voxeldebrisgenerator.h`
- `src/voxeleffect/voxeldebrisgenerator.cpp`

6.225 VoxelExplosionGenerator Class Reference

Inheritance diagram for VoxelExplosionGenerator:



Collaboration diagram for VoxelExplosionGenerator:



Public Member Functions

- **VoxelExplosionGenerator** (const [VoxelCluster](#) *creator)
- void **setRadius** (float radius)
- void **setCount** (int count)
- void **spawn** ()

Protected Member Functions

- float **createScale** ()

Protected Attributes

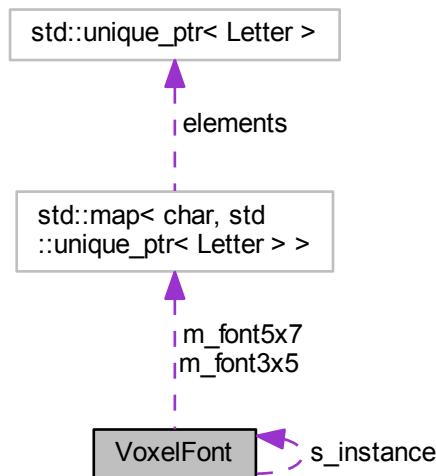
- float **m_radius**
- int **m_count**

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

- src/voxeleffect/voxelexplosiongenerator.h
- src/voxeleffect/voxelexplosiongenerator.cpp

6.226 VoxelFont Class Reference

Collaboration diagram for VoxelFont:



Public Member Functions

- void **drawString** (std::string text, glm::vec3 position, glm::quat orientation, [FontSize](#) size=[FontSize::SIZE5x7](#), float scale=1.f, [FontAlign](#) align=[FontAlign::CENTER](#))
- int **letterWidth** ([FontSize](#) size)
- int **letterHeight** ([FontSize](#) size)

Static Public Member Functions

- static `VoxelFont * instance ()`

Protected Member Functions

- void `loadFont` (const std::string &identifier, glm::vec3 offset, std::map< char, std::unique_ptr< `Letter` >> *map)
- void `loadChar` (const std::string &filename, glm::vec3 offset, const char index, std::map< char, std::unique_ptr< `Letter` >> *map)

Protected Attributes

- std::map< char, std::unique_ptr< `Letter` >> `m_font3x5`
- std::map< char, std::unique_ptr< `Letter` >> `m_font5x7`

Static Protected Attributes

- static `VoxelFont * s_instance = nullptr`

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

- src/ui/voxelfont.h
- src/ui/voxelfont.cpp

6.227 VoxelGridCmp< highPriorityAxis, middlePriorityAxis, lowPriorityAxis > Class Template Reference

Public Member Functions

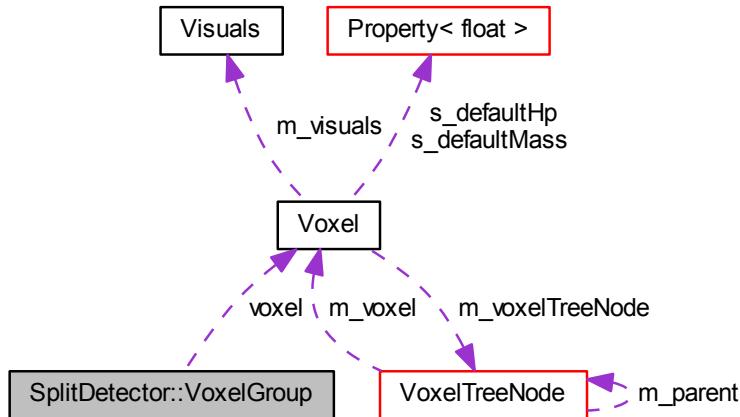
- bool `operator()` (const `Voxel` *voxel1, const `Voxel` *voxel2) const

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

- src/voxel/voxelgridcmp.h
- src/voxel/voxelgridcmp.inl

6.228 SplitDetector::VoxelGroup Struct Reference

Collaboration diagram for SplitDetector::VoxelGroup:



Public Attributes

- `Voxel * voxel`
- `int groupId`

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

- `src/world/handler/splitdetector.h`
- `src/world/handler/splitdetector.cpp`

6.229 VoxelHangman Class Reference

Public Member Functions

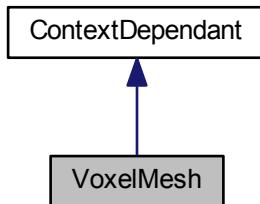
- `void applyOnDestructionHooks (std::list< DamageImpact > &deadlyDamageImpacts)`
- `void removeDestroyedVoxels (std::list< DamageImpact > &deadlyDamageImpacts)`

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

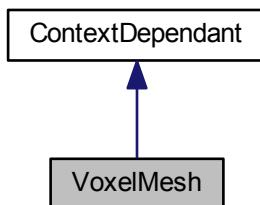
- `src/world/handler/voxelhangman.h`
- `src/world/handler/voxelhangman.cpp`

6.230 VoxelMesh Class Reference

Inheritance diagram for VoxelMesh:



Collaboration diagram for VoxelMesh:



Public Member Functions

- void **bindTo** (glow::Program *program, glow::VertexArrayObject *vao, int bindingIndex)

Protected Member Functions

- void **setupVertexAttribute** (glow::Program *program, glow::VertexArrayObject *vao, const std::string &name, GLboolean normalised, int bindingNum, GLint offset)
- void **initialize** ()
- virtual void **beforeContextDestroy** () override
- virtual void **afterContextRebuild** () override

Protected Attributes

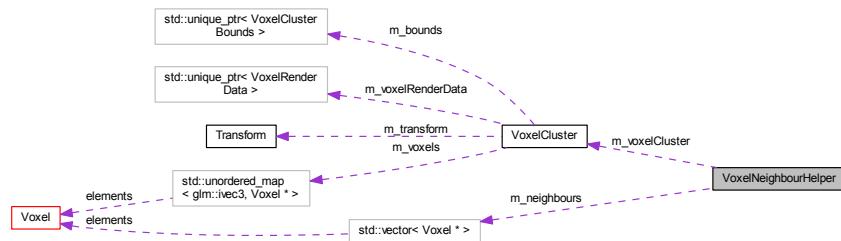
- bool **m_initialized**
- glow::ref_ptr< glow::Buffer > **m_vertexBuffer**

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

- src/voxeleffect/voxelmesh.h
- src/voxeleffect/voxelmesh.cpp

6.231 VoxelNeighbourHelper Class Reference

Collaboration diagram for VoxelNeighbourHelper:



Public Member Functions

- **VoxelNeighbourHelper** (`VoxelCluster` *`voxelCluster`, `bool includeDiagonals=true`)
- const `std::vector< Voxel * > & neighbours` (`const glm::ivec3 &pos`)
- const `std::vector< Voxel * > & neighbours` (`Voxel *voxel`)

Protected Member Functions

- void **considerNeighbour** (`const glm::ivec3 &pos`, `const glm::ivec3 &offset`)

Protected Attributes

- `VoxelCluster * m_voxelCluster`
- `std::vector< Voxel * > m_neighbours`
- `bool m_includeDiagonals`

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

- src/voxel/voxelneighbourhelper.h
- src/voxel/voxelneighbourhelper.cpp

6.232 VoxelParticleData Struct Reference

Public Types

- enum **Status** { `Removed`, `Alive`, `Dead` }

Public Attributes

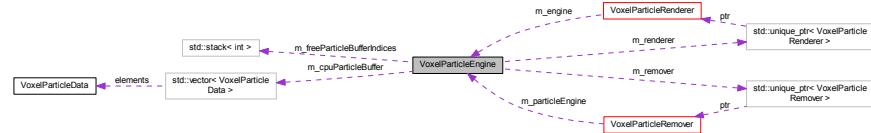
- Status **status**
- glm::vec3 **creationPosition**
- glm::vec3 **creationEulers**
- glm::vec3 **directionalSpeed**
- glm::vec3 **angularSpeed**
- float **creationTime**
- float **deathTime**
- float **scale**
- uint32_t **color**
- float **emissiveness**

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

- src/voxeleffect/voxelparticledata.h

6.233 VoxelParticleEngine Class Reference

Collaboration diagram for VoxelParticleEngine:



Public Member Functions

- float **time () const**
- int **particleDataCount () const**
- **VoxelParticleData * particleData (int index)**
- **std::vector< VoxelParticleData > & particleDataVector ()**
- void **setPlayer (Player &m_player)**
- void **addParticle (const VoxelParticleSetup &particleSetup, const VoxelCluster *creator)**
- void **removeParticle (int index)**
- void **update (float deltaSec)**
- void **draw (const Camera &camera)**

Protected Member Functions

- void **setBufferSize (int bufferSize)**
- void **particleChanged (int bufferIndex)**
- void **updateGPUBuffers (int begin, int end)**

Protected Attributes

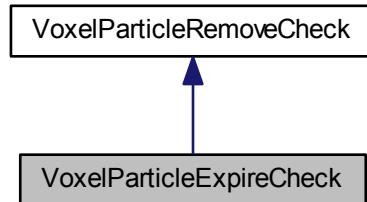
- float **m_time**
- bool **m_initialized**
- std::unique_ptr
 < VoxelParticleRenderer > **m_renderer**
- std::unique_ptr
 < VoxelParticleRemover > **m_remover**
- std::vector< VoxelParticleData > **m_cpuParticleBuffer**
- std::stack< int > **m_freeParticleBufferIndices**
- bool **m_gpuParticleBufferInvalid**
- int **m_gpuParticleBufferInvalidBegin**
- int **m_gpuParticleBufferInvalidEnd**

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

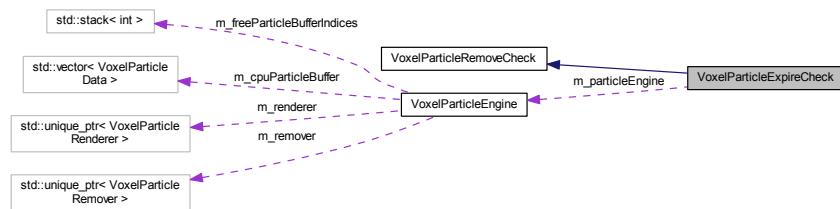
- src/voxeleffect/voxelparticleengine.h
- src/voxeleffect/voxelparticleengine.cpp

6.234 VoxelParticleExpireCheck Class Reference

Inheritance diagram for VoxelParticleExpireCheck:



Collaboration diagram for VoxelParticleExpireCheck:



Public Member Functions

- **VoxelParticleExpireCheck** (const [VoxelParticleEngine](#) &engine)
- virtual bool **isDead** (const [VoxelParticleData](#) &particle) override

Protected Attributes

- const [VoxelParticleEngine](#) & **m_particleEngine**

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

- src/voxeleffect/particlechecks/voxelparticleexpirecheck.h
- src/voxeleffect/particlechecks/voxelparticleexpirecheck.cpp

6.235 VoxelParticleFutureCheck Class Reference

Static Public Member Functions

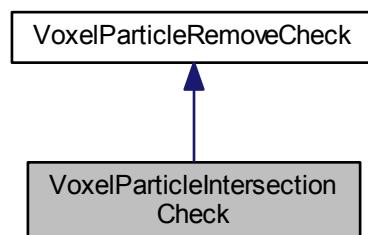
- static bool **intersectsIn** (const [VoxelParticleData](#) &particle, float futureSecs, const [VoxelCluster](#) &against)

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

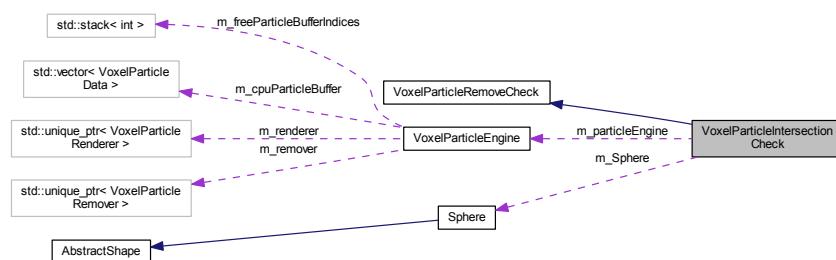
- src/voxeleffect/particlechecks/voxelparticlefuturecheck.h
- src/voxeleffect/particlechecks/voxelparticlefuturecheck.cpp

6.236 VoxelParticleIntersectionCheck Class Reference

Inheritance diagram for VoxelParticleIntersectionCheck:



Collaboration diagram for VoxelParticleIntersectionCheck:



Public Member Functions

- **VoxelParticleIntersectionCheck** (const [VoxelParticleEngine](#) &engine)
- virtual bool **isDead** (const [VoxelParticleData](#) &particle) override

Protected Member Functions

- virtual void **beforeCheck** ()

Protected Attributes

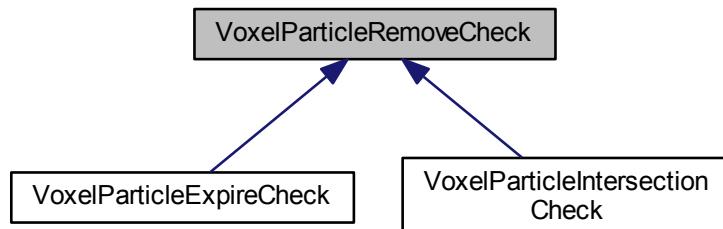
- const [VoxelParticleEngine](#) & **m_particleEngine**
- [Sphere](#) **m_Sphere**

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

- src/voxeleffect/particlechecks/voxelparticleintersectioncheck.h
- src/voxeleffect/particlechecks/voxelparticleintersectioncheck.cpp

6.237 VoxelParticleRemoveCheck Class Reference

Inheritance diagram for VoxelParticleRemoveCheck:



Public Member Functions

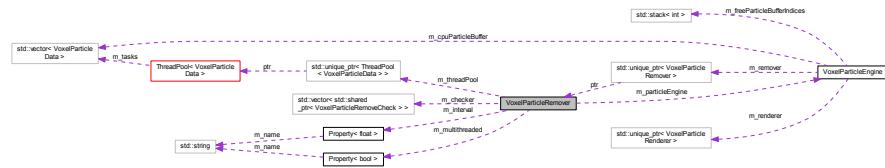
- virtual bool **isDead** (const [VoxelParticleData](#) &particle)=0
- virtual void **setPlayer** ([Player](#) &player)
- virtual void **beforeCheck** ()

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

- src/voxeleffect/particlechecks/voxelparticleremovecheck.h
- src/voxeleffect/particlechecks/voxelparticleremovecheck.cpp

6.238 VoxelParticleRemover Class Reference

Collaboration diagram for VoxelParticleRemover:



Public Member Functions

- **VoxelParticleRemover** ([VoxelParticleEngine](#) *world)
- void **addCheck** (std::shared_ptr< [VoxelParticleRemoveCheck](#) > checker)
- void **setPlayer** ([Player](#) &player)
- float **interval** () const
- void **setInterval** (float interval)
- virtual void **update** (float deltaSec)

Protected Member Functions

- void **performChecks** (int checkCount)
- bool **isDead** ([VoxelParticleData](#) &particle)
- void **beforeCheck** ()

Protected Attributes

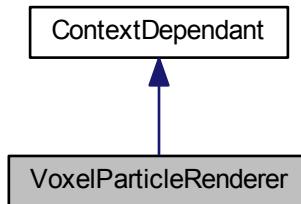
- [VoxelParticleEngine](#) * **m_particleEngine**
- std::vector< std::shared_ptr< [VoxelParticleRemoveCheck](#) > > **m_checker**
- std::unique_ptr< [ThreadPool](#) < [VoxelParticleData](#) > > **m_threadPool**
- [Property](#)< float > **m_interval**
- [Property](#)< bool > **m_multithreaded**
- int **m_currentIndex**

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

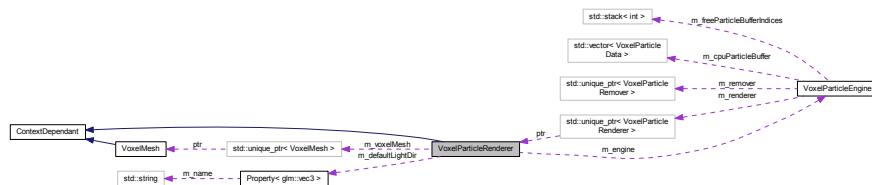
- src/voxeleffect/particlechecks/voxelparticleremover.h
- src/voxeleffect/particlechecks/voxelparticleremover.cpp

6.239 VoxelParticleRenderer Class Reference

Inheritance diagram for VoxelParticleRenderer:



Collaboration diagram for VoxelParticleRenderer:



Public Member Functions

- **VoxelParticleRenderer** ([VoxelParticleEngine](#) *engine)
- void **updateBuffer** (int begin, int end, [VoxelParticleData](#) *data)
- void **draw** (const [Camera](#) &camera)

Protected Member Functions

- void **initialize** ()
- void **loadProgram** ()
- void **setupVertexAttributes** ()
- void **setupVertexAttrib** (GLint offset, const std::string &name, int numPerVertex, GLenum type, GLboolean normalised, int bindingNum)
- void **setupVertexAttribDivisors** ()
- void **setBufferSize** (int bufferSize)
- virtual void **beforeContextDestroy** () override
- virtual void **afterContextRebuild** () override

Protected Attributes

- std::unique_ptr<[VoxelMesh](#)> **m_voxelMesh**
- bool **m_initialized**
- [VoxelParticleEngine](#) * **m_engine**
- int **m_bufferSize**

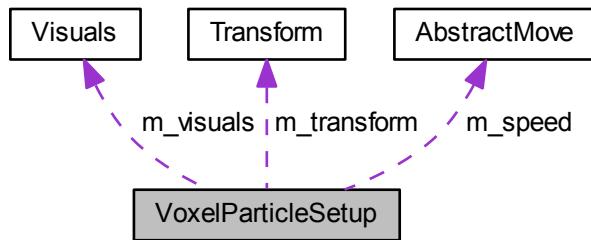
- **Property**< glm::vec3 > **m_defaultLightDir**
- glow::ref_ptr< glow::Buffer > **m_gpuParticleBuffer**
- glow::ref_ptr< glow::Program > **m_program**
- glow::ref_ptr< glow::VertexArrayObject > **m_vertexArrayObject**

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

- src/voxeleffect/voxelparticlerenderer.h
- src/voxeleffect/voxelparticlerenderer.cpp

6.240 VoxelParticleSetup Class Reference

Collaboration diagram for VoxelParticleSetup:



Public Member Functions

- **VoxelParticleSetup** (const **Transform** &transform, const **Visuals** &visuals, const **Speed** &speed, float lifetime)
- **VoxelParticleData toData** (float timeSecs) const

Protected Attributes

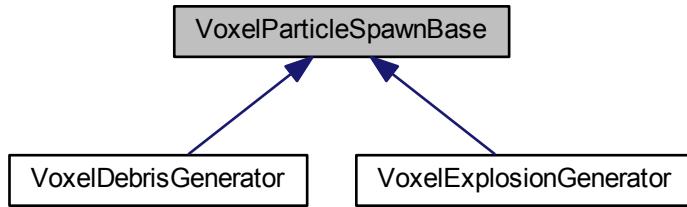
- **Transform m_transform**
- **Visuals m_visuals**
- **Speed m_speed**
- float **m_lifetime**

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

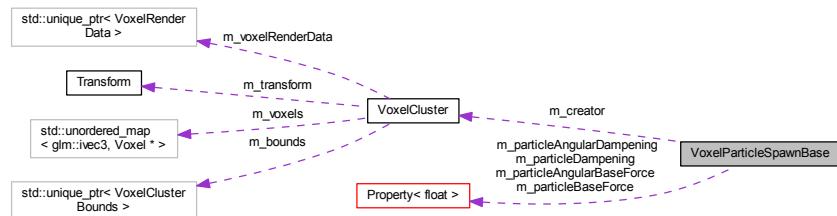
- src/voxeleffect/voxelparticlesetup.h
- src/voxeleffect/voxelparticlesetup.cpp

6.241 VoxelParticleSpawnBase Class Reference

Inheritance diagram for VoxelParticleSpawnBase:



Collaboration diagram for VoxelParticleSpawnBase:



Public Member Functions

- void **setPosition** (const glm::vec3 &position)
- void **setOrientation** (const glm::quat &orientation)
- void **setScale** (float scale, float randomization=0.0f)
- void **setForce** (float force, float randomization=0.0f)
- void **setLifetime** (float lifetime, float randomization=0.0f)
- void **setColor** (int color)
- void **setEmissiveness** (float emissiveness)
- void **setImpactVector** (const glm::vec3 &impactVector)

Protected Member Functions

- **VoxelParticleSpawnBase** (const [VoxelCluster](#) *creator, char *dampeningName, char *angularDampeningName, char *baseForceName, char *angularBaseForceName)
- glm::vec3 **createDirectionalSpeed** ()
- glm::vec3 **createAngularSpeed** ()
- float **createLifetime** ()

Protected Attributes

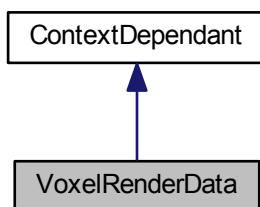
- const `VoxelCluster` * **m_creator**
- `glm::vec3 m_position`
- float **m_scale**
- float **m_scaleRandomization**
- float **m_force**
- float **m_forceRandomization**
- float **m_lifetime**
- float **m_lifetimeRandomization**
- int **m_color**
- float **m_emissiveness**
- `glm::vec3 m_impactVector`
- `Property< float > m_particleDampening`
- `Property< float > m_particleAngularDampening`
- `Property< float > m_particleBaseForce`
- `Property< float > m_particleAngularBaseForce`

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

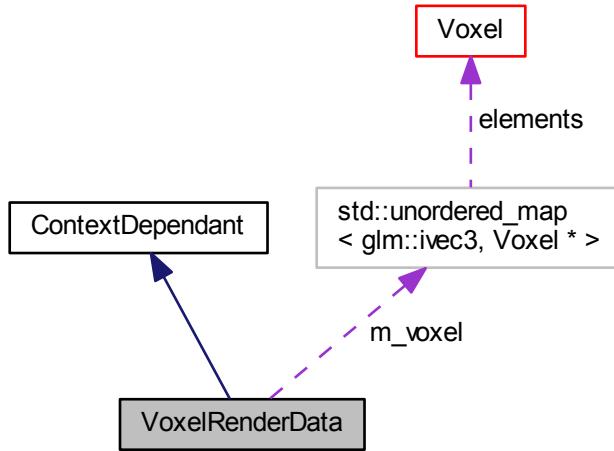
- `src/voxeleffect/voxelparticlespawnbase.h`
- `src/voxeleffect/voxelparticlespawnbase.cpp`

6.242 VoxelRenderData Class Reference

Inheritance diagram for VoxelRenderData:



Collaboration diagram for VoxelRenderData:



Public Member Functions

- **VoxelRenderData** (`std::unordered_map<glm::ivec3, Voxel *> &voxel`)
- void **invalidate** ()
- int **voxelCount** ()
- `glow::VertexArrayObject * vertexArrayObject ()`

Protected Member Functions

- void **updateBuffer** ()
- void **setupVertexAttributes** ()
- void **setupVertexAttrib** (GLint offset, const std::string &name, int numPerVertex, GLenum type, GLboolean normalised, int bindingNum)
- virtual void **beforeContextDestroy** () override
- virtual void **afterContextRebuild** () override

Protected Attributes

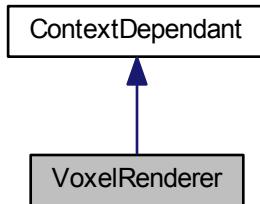
- `std::unordered_map<glm::ivec3, Voxel *> & m_voxel`
- bool **m_isDirty**
- int **m_bufferSize**
- `glow::ref_ptr<glow::Buffer> m_voxelDataBuffer`
- `glow::ref_ptr<glow::VertexArrayObject> m_vertexArrayObject`

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

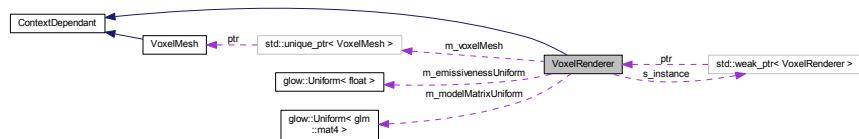
- src/voxel/voxelrenderdata.h
- src/voxel/voxelrenderdata.cpp

6.243 VoxelRenderer Class Reference

Inheritance diagram for VoxelRenderer:



Collaboration diagram for VoxelRenderer:



Public Member Functions

- void **prepareDraw** (const [Camera](#) &camera, bool withBorder=true)
- void **draw** ([VoxelCluster](#) &cluster)
- void **afterDraw** ()
- bool **prepared** ()

Static Public Member Functions

- static std::shared_ptr< [VoxelRenderer](#) > **instance** ()
- static glow::Program * **program** ()
- static [VoxelMesh](#) & **voxelMesh** ()

Protected Member Functions

- void **createAndSetupShaders** ()
- virtual void **beforeContextDestroy** () override
- virtual void **afterContextRebuild** () override

Protected Attributes

- glow::ref_ptr< glow::Program > **m_program**
- std::unique_ptr< [VoxelMesh](#) > **m_voxelMesh**

- bool **m_prepared**
- `glow::Uniform<glm::mat4> * m_modelMatrixUniform`
- `glow::Uniform<float> * m_emissivenessUniform`

Static Protected Attributes

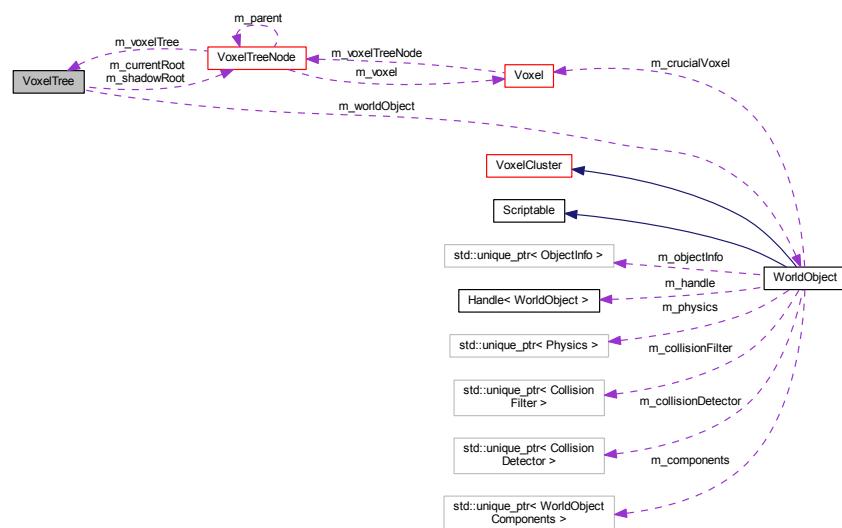
- static `std::weak_ptr<VoxelRenderer> s_instance`

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

- src/voxel/voxelrenderer.h
- src/voxel/voxelrenderer.cpp

6.244 VoxelTree Class Reference

Collaboration diagram for VoxelTree:



Public Member Functions

- `VoxelTree (WorldObject *worldObject)`
- `VoxelTreeNode * root ()`
- `void insert (Voxel *voxel)`
- `void remove (Voxel *voxel)`
- `WorldObject * worldObject ()`

Protected Attributes

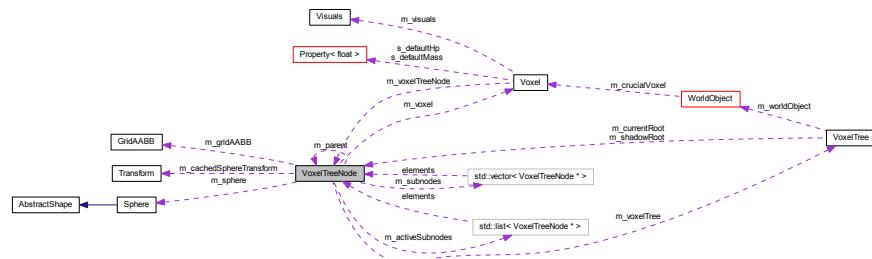
- `VoxelTreeNode * m_shadowRoot`
- `VoxelTreeNode * m_currentRoot`
- `WorldObject * m_worldObject`

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

- src/voxel/voxeltree.h
- src/voxel/voxeltree.cpp

6.245 VoxelTreeNode Class Reference

Collaboration diagram for VoxelTreeNode:



Public Member Functions

- **VoxelTreeNode** (int octIndex, **VoxelTree** *voxelTree, **VoxelTreeNode** *parent, const **GridAABB** &gridAABB)
- **VoxelTreeNode** (**VoxelTree** *voxelTree, const **GridAABB** &gridAABB, **VoxelTreeNode** *initialSubnode)
- int **octIndex** () const
- bool **isAtomic** () const
- bool **isVoxel** () const
- bool **isLeaf** () const
- bool **isEmpty** () const
- std::list< **VoxelTreeNode** * > & **subnodes** ()
- const std::list< **VoxelTreeNode** * > & **subnodes** () const
- **Voxel** * **voxel** ()
- const **Voxel** * **voxel** () const
- **VoxelTree** * **voxelTree** ()
- **VoxelTreeNode** * **parent** ()
- void **setParent** (**VoxelTreeNode** *parent)
- const **GridAABB** & **gridAABB** () const
- **Sphere** & **sphere** ()
- **Sphere** & **sphere** (const **Transform** &transform)
- bool **active** () const
- void **setActive** (bool active)
- void **insert** (**Voxel** *voxel)
- void **remove** (**Voxel** *voxel)

Protected Member Functions

- void **toGroup** ()
- void **subnodeActivated** (**VoxelTreeNode** *subnode)
- void **subnodeDeactivated** (**VoxelTreeNode** *subnode)
- **VoxelTreeNode** * **cellSubnode** (const glm::ivec3 &cell)
- void **calculateSpherePosition** (const **Transform** &transform)
- void **calculateSphereRadius** (const **Transform** &transform)

Protected Attributes

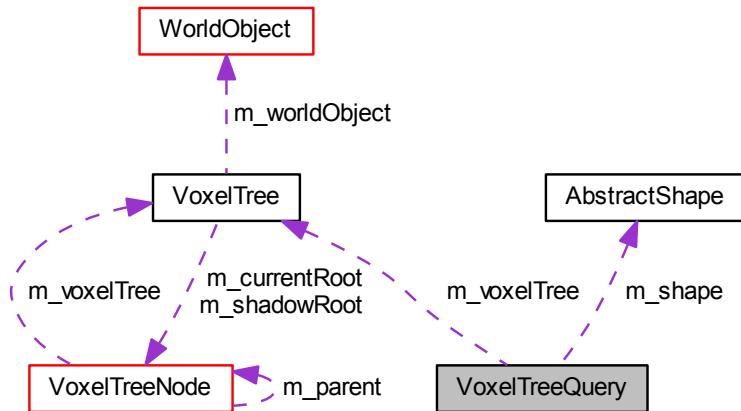
- int **m_octIndex**
- **VoxelTreeNode** * **m_parent**
- **VoxelTree** * **m_voxelTree**
- **GridAABB** **m_gridAABB**
- **Sphere** **m_sphere**
- **Transform** **m_cachedSphereTransform**
- bool **m_active**
- std::vector< **VoxelTreeNode** * > **m_subnodes**
- std::list< **VoxelTreeNode** * > **m_activeSubnodes**
- **Voxel** * **m voxel**

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

- src/voxel/voxeltreenode.h
- src/voxel/voxeltreenode.cpp

6.246 VoxelTreeQuery Class Reference

Collaboration diagram for VoxelTreeQuery:



Public Member Functions

- **VoxelTreeQuery** (**VoxelTree** *voxelTree, const **AbstractShape** *shape)
- bool **areVoxelsIntersecting** ()
- std::unordered_set< **Voxel** * > **intersectingVoxels** ()

Protected Member Functions

- void **query** (**VoxelTreeNode** *node, std::function< void(**Voxel** *)> onVoxelIntersection)

Protected Attributes

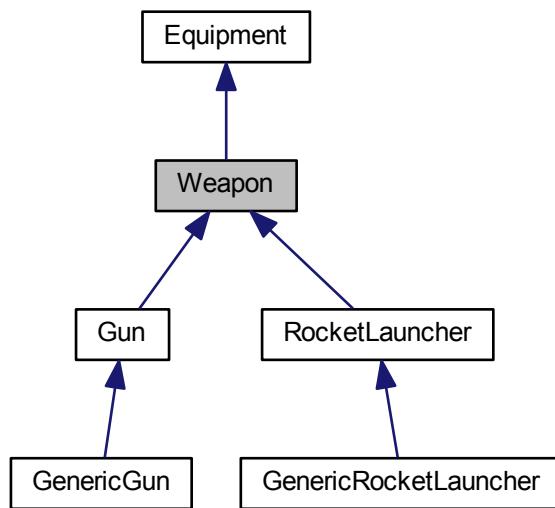
- `VoxelTree * m_voxelTree`
- const `AbstractShape * m_shape`
- bool `m_queryInterrupted`

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

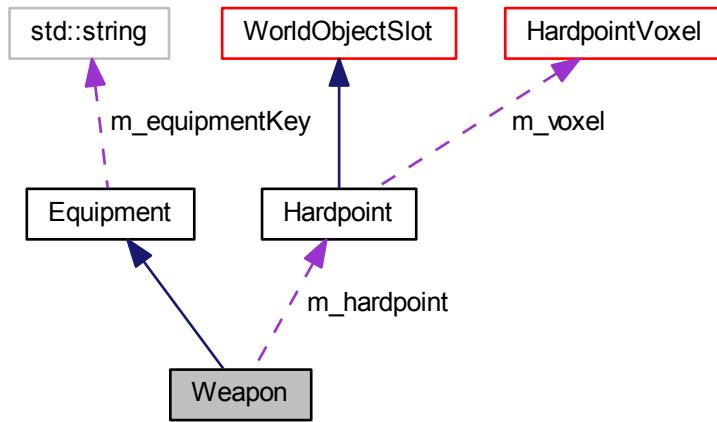
- `src/voxel/voxelequery.h`
- `src/voxel/voxelequery.cpp`

6.247 Weapon Class Reference

Inheritance diagram for Weapon:



Collaboration diagram for Weapon:



Public Member Functions

- **Weapon** (WeaponType type, const std::string &equipmentKey)
- virtual const Visuals & **visuals** () const =0
- **Hardpoint * hardpoint** ()
- void **setHardpoint** (Hardpoint *hardpoint)
- WeaponType **type** () const
- virtual float **cooldownTime** () const =0
- virtual void **update** (float deltaSec)
- bool **canFire** ()
- void **onFired** ()

Protected Attributes

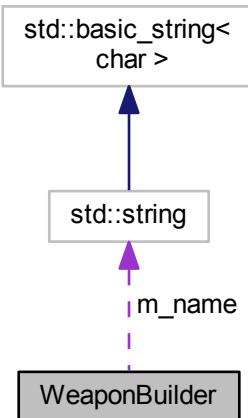
- **Hardpoint * m_hardpoint**
- WeaponType **m_type**
- float **m_cooldown**

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

- src/equipment/weapon.h
- src/equipment/weapon.cpp

6.248 WeaponBuilder Class Reference

Collaboration diagram for WeaponBuilder:



Public Member Functions

- `WeaponBuilder (const std::string &name)`
- `Weapon * build ()`
- `GenericGun * buildGenericGun ()`
- `GenericRocketLauncher * buildGenericRocketLauncher ()`

Protected Attributes

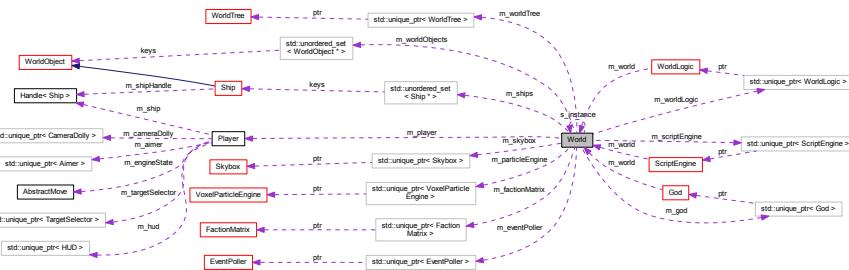
- `const std::string & m_name`

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

- `src/resource/weaponbuilder.h`
- `src/resource/weaponbuilder.cpp`

6.249 World Class Reference

Collaboration diagram for World:



Public Member Functions

- `Player & player ()`
 - `void setPlayer (Player &player)`
 - `Skybox & skybox ()`
 - `WorldLogic & worldLogic ()`
 - `God & god ()`
 - `WorldTree & worldTree ()`
 - `ScriptEngine & scriptEngine ()`
 - `VoxelParticleEngine & particleEngine ()`
 - `FactionMatrix & factionMatrix ()`
 - `EventPoller & eventPoller ()`
 - `std::unordered_set<WorldObject * > & worldObjects ()`
 - `std::unordered_set<Ship * > & ships ()`
 - `void update (float deltaSecs)`
 - `float deltaSec () const`

Static Public Member Functions

- static **World** * **instance** ()
 - static void **reset** (bool showWarning=true)

Protected Member Functions

- void **addWorldObject** (*WorldObject* *worldObject)
 - void **removeWorldObject** (*WorldObject* *worldObject)

Protected Attributes

- float **m_deltaSec**
 - Player * **m_player**
 - std::unique_ptr< Skybox > **m_skybox**
 - std::unique_ptr< WorldTree > **m_worldTree**
 - std::unique_ptr< WorldLogic > **m_worldLogic**
 - std::unique_ptr< God > **m_god**
 - std::unique_ptr< ScriptEngine > **m_scriptEngine**

- std::unique_ptr
 < VoxelParticleEngine > **m_particleEngine**
 - std::unique_ptr< FactionMatrix > **m_factionMatrix**
 - std::unique_ptr< EventPoller > **m_eventPoller**
 - std::unordered_set< WorldObject * > **m_worldObjects**
 - std::unordered_set< Ship * > **m_ships**

Static Protected Attributes

- static `World * s_instance = nullptr`

Friends

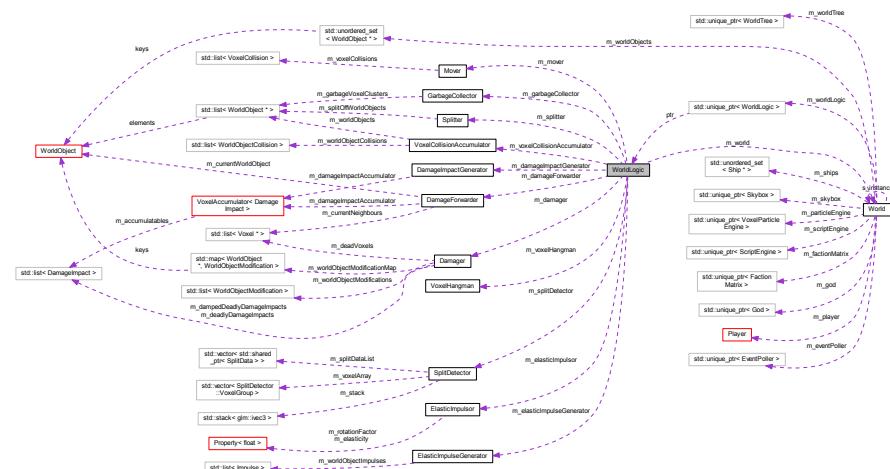
- class **God**

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

- src/world/world.h
 - src/world/world.cpp

6.250 WorldLogic Class Reference

Collaboration diagram for WorldLogic:



Public Member Functions

- **WorldLogic** ([World](#) &world)
 - void **update** (float deltaSecs)
 - **DamageForwarder** & **damageForwarder** ()

Protected Member Functions

- void **damageForwardLoop** (std::list< DamagelImpact > damageImpulses)

Protected Attributes

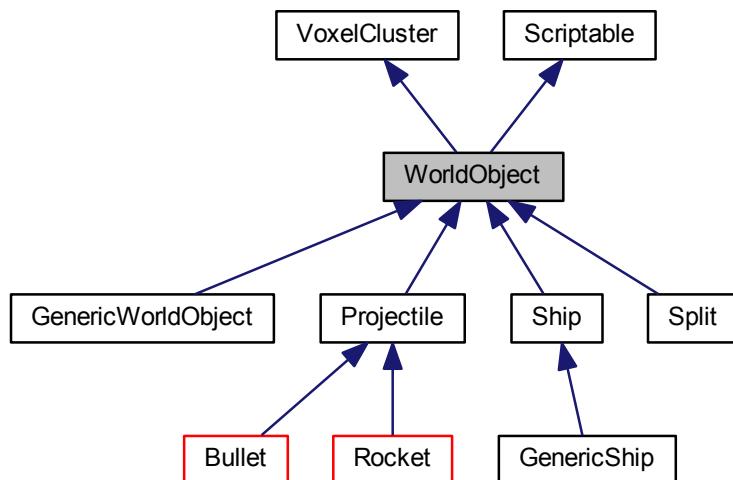
- `World & m_world`
- `Mover m_mover`
- `VoxelCollisionAccumulator m voxelCollisionAccumulator`
- `ElasticImpulseGenerator m elasticImpulseGenerator`
- `ElasticImpulsor m elasticImpulsor`
- `Damager m_damager`
- `DamageForwarder m_damageForwarder`
- `DamageImpactGenerator m_damageImpactGenerator`
- `SplitDetector m_splitDetector`
- `Splitter m_splitter`
- `GarbageCollector m_garbageCollector`
- `VoxelHangman m voxelHangman`

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

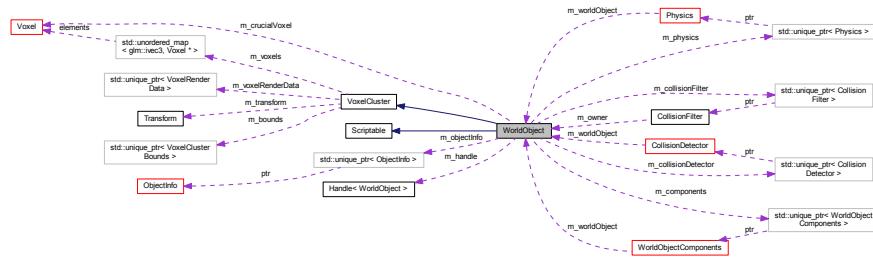
- `src/world/worldlogic.h`
- `src/world/worldlogic.cpp`

6.251 WorldObject Class Reference

Inheritance diagram for WorldObject:



Collaboration diagram for WorldObject:



Public Member Functions

- virtual WorldObjectType **objectType () const**
- SpawnState **spawnState () const**
- void **setSpawnState (SpawnState spawnState)**
- **CollisionFilter & collisionFilter ()**
- void **setCollisionFilter (CollisionFilter *collisionFilter)**
- **CollisionDetector & collisionDetector ()**
- **Physics & physics ()**
- const **Physics & physics () const**
- **ObjectInfo & objectInfo ()**
- **WorldObjectComponents & components ()**
- const **WorldObjectComponents & components () const**
- virtual void **update (float deltaSec)**
- virtual void **addVoxel (Voxel *voxel)** override
- virtual void **removeVoxel (Voxel *voxel)** override
- **Voxel * crucialVoxel ()**
- void **setCrucialVoxel (const glm::ivec3 &cell)**
- void **updateTransformAndGeode (const glm::vec3 &position, const glm::quat &orientation)**
- virtual void **onCollision ()**
- virtual void **onSpawnFail ()**
- **Handle<WorldObject> & handle ()**
- float **collisionFieldOfDamage () const**
- void **setCollisionFieldOfDamage (float collisionFieldOfDamage)**
- bool **isDestroyed () const**

Protected Member Functions

- **WorldObject (CollisionFilter *collisionFilter, float scale=1.0f)**
- virtual void **onDestruction ()**

Protected Attributes

- std::unique_ptr< CollisionFilter > **m_collisionFilter**
- std::unique_ptr< CollisionDetector > **m_collisionDetector**
- std::unique_ptr< Physics > **m_physics**
- std::unique_ptr< ObjectInfo > **m_objectInfo**
- std::unique_ptr< WorldObjectComponents > **m_components**

- Handle< WorldObject > **m_handle**
- Voxel * **m_crucialVoxel**
- float **m_collisionFieldOfDamage**
- SpawnState **m_spawnState**
- bool **m_isDestroyed**

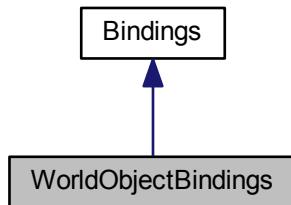
Additional Inherited Members

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

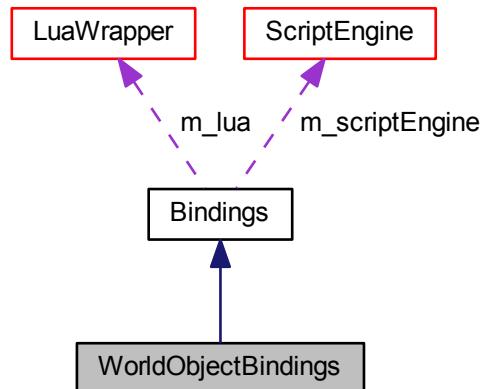
- src/worldobject/worldobject.h
- src/worldobject/worldobject.cpp

6.252 WorldObjectBindings Class Reference

Inheritance diagram for WorldObjectBindings:



Collaboration diagram for WorldObjectBindings:



Public Member Functions

- **WorldObjectBindings** ([GamePlayScript](#) &script)

Protected Member Functions

- virtual void **initialize** ()
- apikey **apiPlayerShip** ()
- apikey **apiCreateShip** (const std::string &name)
- int **apiSpawn** (apikey worldObject)
- int **apiSetPosition** (apikey worldObject, const glm::vec3 &position)
- int **apiSetOrientation** (apikey worldObject, const glm::vec3 &orientation)
- glm::vec3 **apiPosition** (apikey worldObject)
- glm::vec3 **apiOrientation** (apikey worldObject)
- apikey **apiOnWorldObjectDestroyed** (apikey worldObject, const std::string &callback)

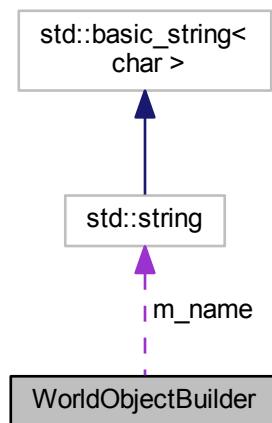
Additional Inherited Members

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

- src/scripting/bindings/worldobjectbindings.h
- src/scripting/bindings/worldobjectbindings.cpp

6.253 WorldObjectBuilder Class Reference

Collaboration diagram for WorldObjectBuilder:



Public Member Functions

- **WorldObjectBuilder** (const std::string &name)
- [WorldObject](#) * **build** ()

- `Bullet * buildBullet ()`
- `Rocket * buildRocket ()`
- `Ship * buildShip ()`
- `WorldObject * buildWorldObject ()`

Protected Member Functions

- template<typename WorldObjectType >
`WorldObjectType * makeWorldObject ()`
- `void setupVoxelCluster (WorldObject *worldObject)`
- `void setupComponents (WorldObjectComponents &components)`
- `void setupHardpoints (WorldObjectComponents &components)`
- `void setupEngineSlots (WorldObjectComponents &components)`
- `void equipSomehow (WorldObject *worldObject)`

Protected Attributes

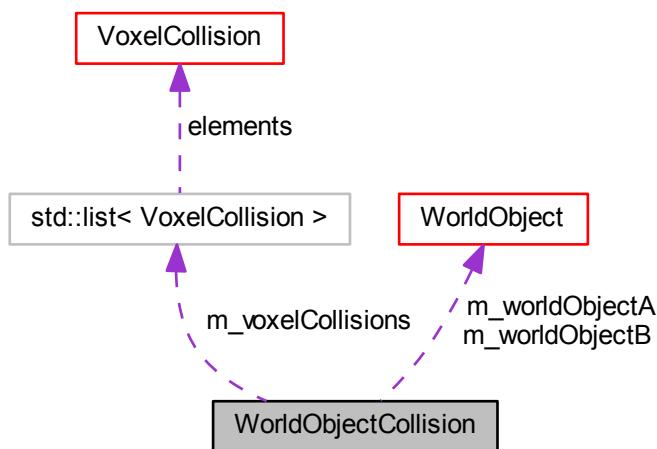
- `std::string m_name`

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

- `src/resource/worldobjectbuilder.h`
- `src/resource/worldobjectbuilder.cpp`

6.254 WorldObjectCollision Class Reference

Collaboration diagram for WorldObjectCollision:



Public Member Functions

- **WorldObjectCollision** (*WorldObject* *worldObjectA, *WorldObject* *worldObjectB)
- **WorldObject** * **worldObjectA** ()
- **WorldObject** * **worldObjectB** ()
- std::list< *VoxelCollision* > & **voxelCollisions** ()
- void **addVoxelCollision** (*VoxelCollision* & voxelCollision)

Protected Attributes

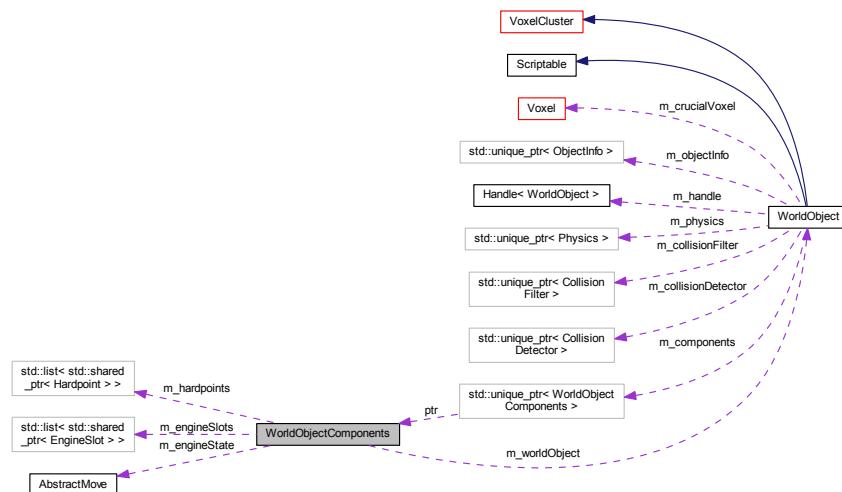
- *WorldObject* * **m_worldObjectA**
- *WorldObject* * **m_worldObjectB**
- std::list< *VoxelCollision* > **m_voxelCollisions**

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

- src/collision/worldobjectcollision.h
- src/collision/worldobjectcollision.cpp

6.255 WorldObjectComponents Class Reference

Collaboration diagram for WorldObjectComponents:



Public Member Functions

- **WorldObjectComponents** (*WorldObject* *worldObject)
- **WorldObject** * **worldObject** ()
- const **WorldObject** * **worldObject** () const
- void **addEngineSlot** (std::shared_ptr< *EngineSlot* > engineSlot)
- void **removeEngineSlot** (const *EngineSlot* *engineSlot)
- std::shared_ptr< *EngineSlot* > **engineSlot** (int index)
- std::list< std::shared_ptr< *EngineSlot* > > & **engineSlots** ()
- **EnginePower** **enginePower** () const

- Acceleration **currentAcceleration** () const
- const EngineState & **engineState** () const
- void **setEngineState** (const EngineState &engineState)
- void **addHardpoint** (std::shared_ptr< Hardpoint > hardpoint)
- void **removeHardpoint** (const Hardpoint *hardpoint)
- std::shared_ptr< Hardpoint > **hardpoint** (int index)
- std::list< std::shared_ptr< Hardpoint > > & **hardpoints** ()
- void **fireAtPoint** (const glm::vec3 &point)
- void **fireAtObject** (WorldObject *worldObject)
- void **update** (float deltaSec)

Protected Attributes

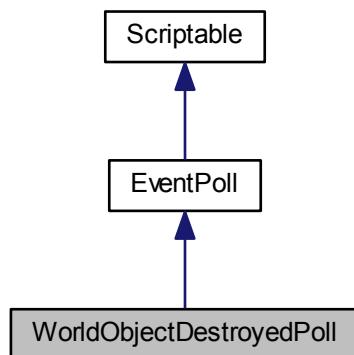
- WorldObject * **m_worldObject**
- std::list< std::shared_ptr< EngineSlot > > & **m_engineSlots**
- std::list< std::shared_ptr< Hardpoint > > & **m_hardpoints**
- EngineState **m_engineState**

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

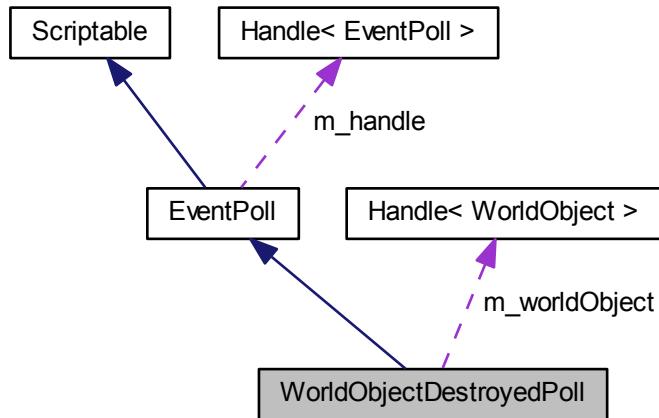
- src/worldobject/worldobjectcomponents.h
- src/worldobject/worldobjectcomponents.cpp

6.256 WorldObjectDestroyedPoll Class Reference

Inheritance diagram for WorldObjectDestroyedPoll:



Collaboration diagram for WorldObjectDestroyedPoll:



Public Member Functions

- **WorldObjectDestroyedPoll** ([WorldObject](#) *worldObject, const std::function< void() > &callback)

Protected Member Functions

- virtual bool **poll** () override
- virtual bool **isDead** () override

Protected Attributes

- [Handle< WorldObject >](#) **m_worldObject**

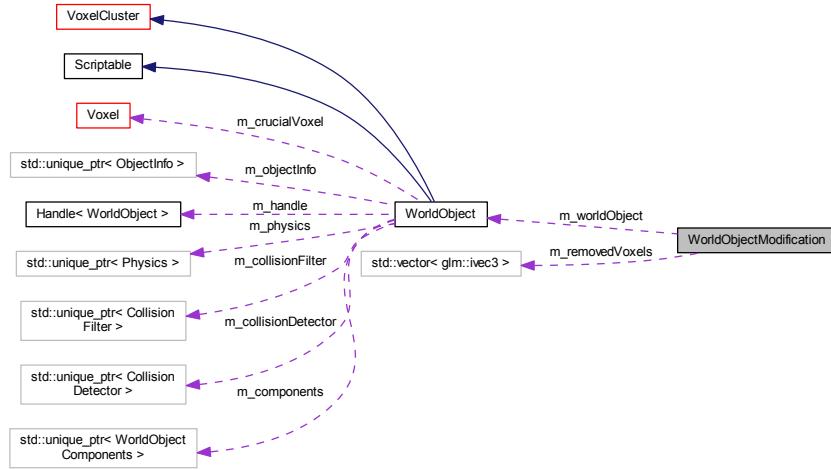
Additional Inherited Members

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

- src/events/worldobjectdestroyedpoll.h
- src/events/worldobjectdestroyedpoll.cpp

6.257 WorldObjectModification Class Reference

Collaboration diagram for WorldObjectModification:



Public Member Functions

- **WorldObjectModification** (*WorldObject* *worldObject)
- **WorldObject** * **worldObject** ()
- void **removedVoxel** (const *glm::ivec3* &pos)
- const *std::vector< glm::ivec3 >* & **removedVoxels** ()

Protected Attributes

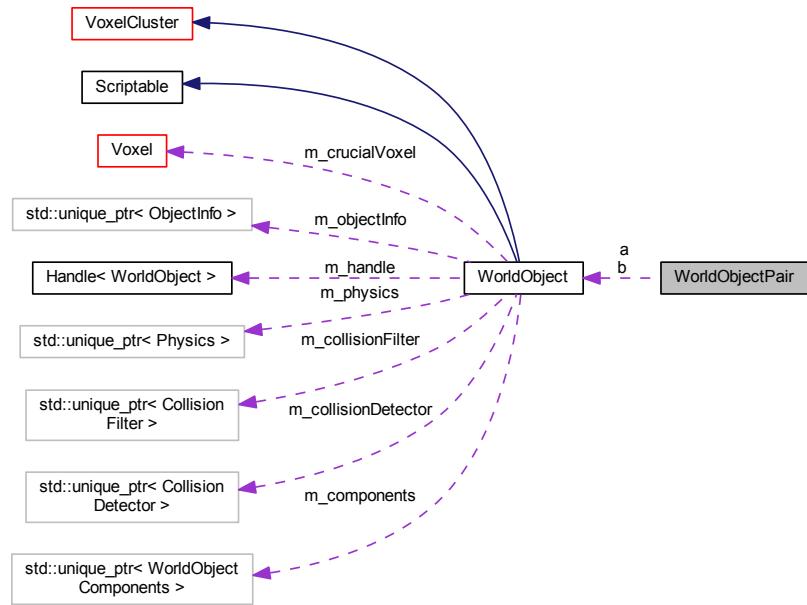
- *WorldObject* * **m_worldObject**
- *std::vector< glm::ivec3 >* **m_removedVoxels**

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

- src/world/helper/worldobjectmodification.h
- src/world/helper/worldobjectmodification.cpp

6.258 WorldObjectPair Struct Reference

Collaboration diagram for WorldObjectPair:



Public Member Functions

- **WorldObjectPair** (`WorldObject *a, WorldObject *b`)
- `bool operator<` (`const WorldObjectPair &other`) `const`

Public Attributes

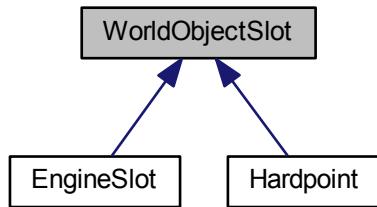
- `WorldObject * a`
- `WorldObject * b`

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

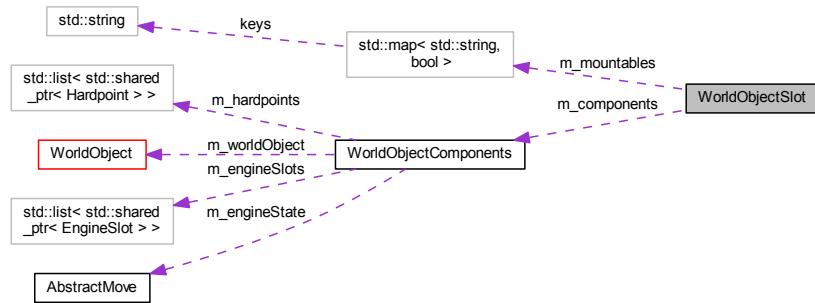
- `src/world/handler/voxelcollisionaccumulator.cpp`

6.259 WorldObjectSlot Class Reference

Inheritance diagram for WorldObjectSlot:



Collaboration diagram for WorldObjectSlot:



Public Member Functions

- **WorldObjectSlot** (*WorldObjectComponents* *components, int index)
- std::list< std::string > **mountables** () const
- bool **mountable** (const std::string &name) const
- void **setMountable** (const std::string &name, bool mountable)
- *WorldObjectComponents* * **components** ()
- const *WorldObjectComponents* * **components** () const
- int **index** () const

Protected Attributes

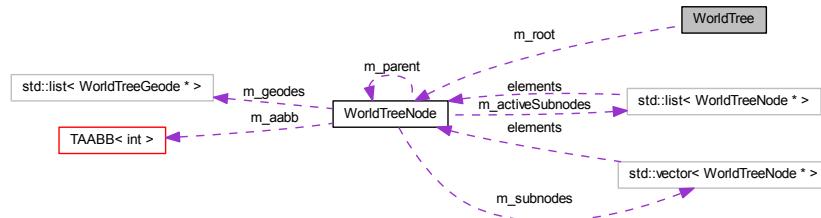
- *WorldObjectComponents* * **m_components**
- std::map< std::string, bool > **m_mountables**
- int **m_index**

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

- src/equipment/worldobjectslot.h
- src/equipment/worldobjectslot.cpp

6.260 WorldTree Class Reference

Collaboration diagram for WorldTree:



Public Member Functions

- `WorldTreeNode * root ()`
- `WorldTreeGeode * insert (WorldObject *worldObject)`
- `void insert (WorldTreeGeode *geode)`
- `void remove (WorldTreeGeode *geode)`
- `void aabbChanged (WorldTreeGeode *geode)`

Protected Member Functions

- `void extend (const IAABB &aabb)`
- `WorldTreeNode * containingNode (const IAABB &aabb, WorldTreeNode *node)`

Protected Attributes

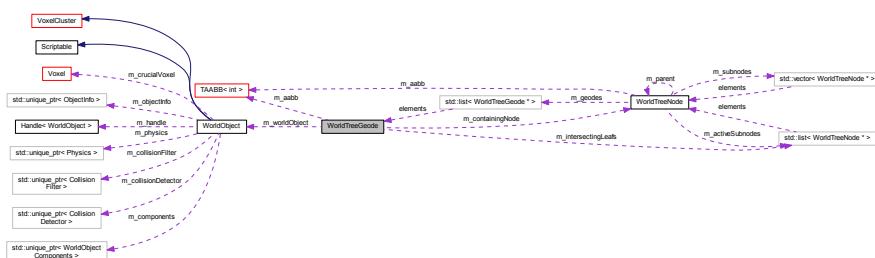
- `WorldTreeNode * m_root`

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

- `src/worldtree/worldtree.h`
- `src/worldtree/worldtree.cpp`

6.261 WorldTreeGeode Class Reference

Collaboration diagram for WorldTreeGeode:



Public Member Functions

- **WorldTreeGeode** (*WorldObject* *worldObject)
- **WorldObject** * **worldObject** ()
- const **WorldObject** * **worldObject** () const
- void **setWorldObject** (*WorldObject* *worldObject)
- **WorldTreeNode** * **containingNode** ()
- const **WorldTreeNode** * **containingNode** () const
- void **setContainingNode** (*WorldTreeNode* *node)
- const **IAABB & aabb** () const
- void **setAABB** (const **IAABB &aabb**)
- std::list< **WorldTreeNode** * > & **intersectingLeafs** ()
- void **addIntersectingLeaf** (*WorldTreeNode* *leaf)
- void **removeIntersectingLeaf** (*WorldTreeNode* *leaf)

Protected Attributes

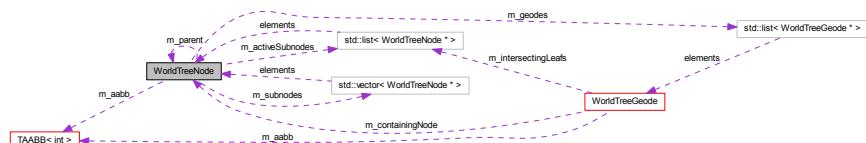
- **WorldObject** * **m_worldObject**
- **WorldTreeNode** * **m_containingNode**
- **IAABB m_aabb**
- std::list< **WorldTreeNode** * > **m_intersectingLeafs**

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

- src/worldtree/worldtreegeode.h
- src/worldtree/worldtreegeode.cpp

6.262 WorldTreeNode Class Reference

Collaboration diagram for WorldTreeNode:



Public Member Functions

- **WorldTreeNode** (int octIndex, *WorldTreeNode* *parent, const **IAABB &aabb**)
- **WorldTreeNode** (const **IAABB &aabb**, *WorldTreeNode* *initialSubnode)
- void **clear** ()
- int **octIndex** () const
- void **setOctIndex** (int octIndex)
- const **IAABB & aabb** () const
- **WorldTreeNode** * **parent** ()
- const **WorldTreeNode** * **parent** () const
- void **setParent** (*WorldTreeNode* *parent)
- bool **active** () const
- void **setActive** (bool active)

- const std::list<WorldTreeGeode * > & **geodes** () const
- const std::list<WorldTreeNode * > & **subnodes** () const
- bool **isLeaf** () const
- bool **isEmpty** () const
- bool **isRootnode** () const
- bool **isAtomic** () const
- void **insert** (WorldTreeGeode *geode)
- void **remove** (WorldTreeGeode *geode)

Protected Member Functions

- void **toGroup** (WorldTreeNode *initialSubnode=nullptr)
- void **subnodeActivated** (WorldTreeNode *subnode)
- void **subnodeDeactivated** (WorldTreeNode *subnode)

Protected Attributes

- WorldTreeNode * **m_parent**
- IAABB **m_aabb**
- int **m_octIndex**
- float **m_extent**
- bool **m_active**
- std::list<WorldTreeGeode * > **m_geodes**
- std::vector<WorldTreeNode * > **m_subnodes**
- std::list<WorldTreeNode * > **m_activeSubnodes**

Static Protected Attributes

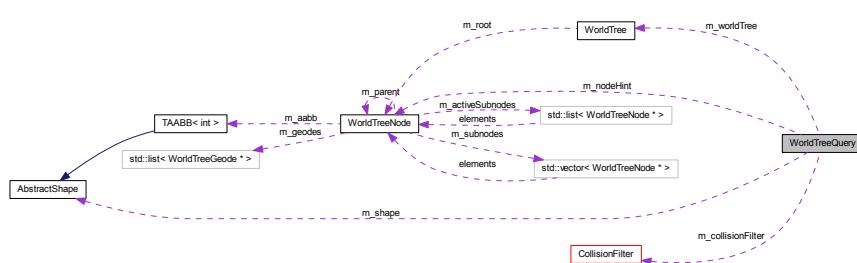
- static const int **MIN_EXTENT** = 16
- static const int **MAX_GEODES** = 4

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

- src/worldtree/worldtreenode.h
- src/worldtree/worldtreenode.cpp

6.263 WorldTreeQuery Class Reference

Collaboration diagram for WorldTreeQuery:



Public Member Functions

- **WorldTreeQuery** (`WorldTree *worldTree, const AbstractShape *shape, WorldTreeNode *nodeHint=nullptr, CollisionFilter *collisionFilter=nullptr)`
- `bool areGeodesNear ()`
- `std::unordered_set<WorldTreeGeode *> nearGeodes ()`
- `bool areVoxelsIntersecting ()`
- `std::unordered_set<Voxel *> intersectingVoxels ()`
- `std::unordered_set<WorldObject *> intersectingWorldObjects ()`

Protected Member Functions

- `WorldTreeNode * getQueryRoot (WorldTreeNode *node=nullptr) const`
- `void query (WorldTreeNode *node, std::function<void(WorldTreeGeode *)> onGeodeInteraction)`

Protected Attributes

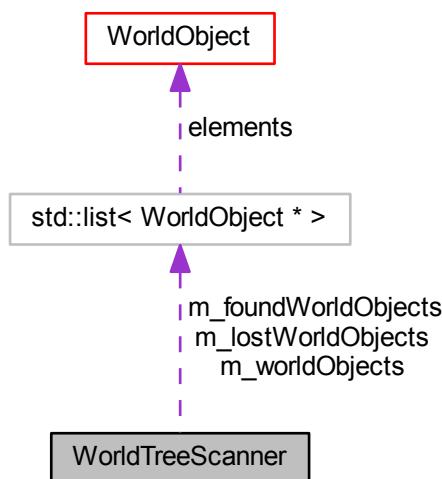
- `WorldTree * m_worldTree`
- `WorldTreeNode * m_nodeHint`
- `CollisionFilter * m_collisionFilter`
- `const AbstractShape * m_shape`
- `bool m_queryInterrupted`

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

- `src/worldtree/worldtreequery.h`
- `src/worldtree/worldtreequery.cpp`

6.264 WorldTreeScanner Class Reference

Collaboration diagram for WorldTreeScanner:



Public Member Functions

- float **scanInterval** () const
- void **setScanInterval** (float scanInterval)
- float **scanRadius** () const
- void **setScanRadius** (float scanRadius)
- const std::list< [WorldObject](#) * > & **worldObjects** ()
- const std::list< [WorldObject](#) * > & **foundWorldObjects** ()
- const std::list< [WorldObject](#) * > & **lostWorldObjects** ()
- void **update** (float deltaSec, [WorldObject](#) *worldObject)
- void **update** (float deltaSec, const glm::vec3 &position)
- virtual void **onFoundWorldObject** ([WorldObject](#) *worldObject)
- virtual void **onLostWorldObject** ([WorldObject](#) *worldObject)

Protected Member Functions

- void **update** (float deltaSec, [WorldObject](#) *worldObject, const glm::vec3 &position)
- void **scan** ([WorldObject](#) *worldObject, const glm::vec3 &position)
- void **callHooks** ()

Protected Attributes

- float **m_scanInterval**
- float **m_scanCountdown**
- float **m_scanRadius**
- std::list< [WorldObject](#) * > **m_worldObjects**
- std::list< [WorldObject](#) * > **m_foundWorldObjects**
- std::list< [WorldObject](#) * > **m_lostWorldObjects**

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

- src/worldtree/worldtreescanner.h
- src/worldtree/worldtreescanner.cpp

Index

AABBEnteredPoll, 21
AbstractMove, 22
AbstractPropertyCollection, 22
AbstractShape, 23
ActionKeyMapping, 24
AiBindings, 25
AiGroupTask, 26
AiTask, 31
AiTaskFinishedPoll, 32
AimHelperHudget, 29
AimHelperHudgetVoxels, 30
Aimer, 28
ArrowHudgetVoxels, 34

B, 35
BaseLuaFunction, 36
BaseScenario, 36
BattleScenario, 38
Bindings, 39
BlitProgram, 40
Blitter, 41
BoardComputer, 43
Bullet, 44

C, 45
Camera, 46
CameraDolly, 48
CameraFollowHelper, 49
CameraHead, 50
Capsule, 51
Character, 52
CircularHudget, 53
ClusterCache, 54
ClusterLoader, 55
CockpitVoxel, 55
CollisionDetector, 56
CollisionFilter, 57
CollisionFilterIgnoringCreator, 59
ColorCoder, 60
CommandLineParser, 61
CommonBindings, 61
ContextDependant, 63
ContextProvider, 64
CrossHair, 65
CrossHairElement, 67
CrossHairVoxels, 68
CrucialVoxel, 69

D, 70
DamageForwarder, 71

DamagedImpact, 72
DamagedImpactGenerator, 73
Damager, 73
DdsTexture, 74
DefaultRenderPipeline, 75
DefendAreaTask, 76
DirectSuicideTask, 79
DirectoryReader, 78

E, 80
ElasticImpulseGenerator, 81
ElasticImpulsor, 82
Engine, 83
EngineBuilder, 85
EnginePower, 86
EngineSlot, 86
EngineSlotVoxel, 88
EngineTrailGenerator, 89
Equipment, 90
EventPoll, 91
EventPoller, 92

F, 93
Faction, 94
FactionMatrix, 95
FactionRelation, 96
FightTask, 97
FlyToTask, 99
FormationMemberTask, 100
FrameBuffer, 101
FrozenGameScenario, 102
FuelVoxel, 104

G, 105
Game, 106
 update, 107
GamePlay, 107
 onEntered, 109
 onLeft, 109
 update, 109
GamePlayPaused, 109
 onEntered, 110
 onLeft, 110
 update, 110
GamePlayRunning, 111
 onEntered, 112
 onLeft, 112
 update, 112
GamePlayRunningInput, 112
GamePlayScene, 114

GamePlayScript, 115
GameScenario, 116
GameState, 117
 onEntered, 118
 onLeft, 118
 update, 119
GarbageCollector, 119
GenericBullet, 120
GenericEngine, 121
GenericGun, 123
GenericRocket, 124
GenericRocketLauncher, 126
GenericShip, 127
GenericWorldObject, 128
GeometryHelper, 129
glow::Uniform< T >, 241
God, 129
GridAABB, 130
Gun, 131

HMD, 137
HMDManager, 138
HUD, 139
HUDObjectDelegate, 143
Handle< T >, 132
HandleImpl< T >, 132
Hardpoint, 133
HardpointAimHelper, 134
HardpointVoxel, 135
Hudget, 140

Impulse, 144
ImpulseAccumulator, 145
InertiaFollower, 146
InputConfigurator, 146
InputMapping, 147

KeyTrigger, 147

Letter, 149
Line, 150
LoopingTimer, 151
LuaFunction
 state, 153
LuaFunction< N, Return, Args >, 152
LuaWrapper, 153
LuaWrapper::__pop< 0, Ts...>, 20
LuaWrapper::__pop< 1, T >, 20
LuaWrapper::__pop< size_t, Ts >, 19
Luaw::__indices< Is >, 19
Luaw::__indices_builder< 0, Is...>, 19
Luaw::__indices_builder< N, Is >, 19

Math, 155
Metrics, 155
MonoView, 156
Movement, 158
Mover, 159

ObjectHudget, 160
ObjectHudgetCornerVoxels, 161
ObjectHudgetVoxels, 162
ObjectInfo, 163
onEntered
 GamePlay, 109
 GamePlayPaused, 110
 GamePlayRunning, 112
 GameState, 118
 State, 218
 TestState, 229
onLeft
 GamePlay, 109
 GamePlayPaused, 110
 GamePlayRunning, 112
 GameState, 118
 State, 218
 TestState, 229

pathToDescendant
 State, 218
PatrolWaypointsTask, 164
Physics, 165
PirateScenario, 166
Player, 167
Point, 168
PoliceFaction, 169
PostProcessingPass, 171
Projectile, 173
Property< T >, 175
PropertyCollection< T >, 176
PropertyConverter, 176
PropertyDirectory, 177
PropertyManager, 178

RandBool, 179
RandFloat, 179
RandVec3, 179
Range, 180
Ray, 180
RenderMetaData, 181
RenderPass, 182
RenderPipeline, 183
Rocket, 184
RocketLauncher, 186

Scene, 187
ScreenBlitter, 188
ScreenQuad, 188
Script, 190
ScriptEngine, 194
Scriptable, 191
ScriptedScenario, 192
SecondaryInputValues, 195
Ship, 195
SimpleWayfind, 196
SingleShotTimer, 197
Size< T >, 198
Skybox, 199
snowhouse::Stringizer< glm::vec3 >, 224

snowhouse::Stringizer< TAABB< T > >, 225
 Sound, 200
 SoundManager, 201
 SoundProperties, 202
 SpawnRequest, 203
 SpecialVoxel, 204
 Sphere, 205
 Split, 206
 SplitData, 208
 SplitDetector, 209
 SplitDetector::VoxelGroup, 256
 Splitter, 210
 Squad, 211
 SquadBindings, 212
 SquadLogic, 214
 Starfield, 215
 Starfield::CameraLocation, 50
 Starfield::StarData, 214
 State, 216
 onEntered, 218
 onLeft, 218
 pathToDescendant, 218
 transit, 218
 update, 219
 state
 LuaFunction, 153
 std::hash< glm::ivec3 >, 136
 std::hash< pair< A, B > >, 137
 StereoBlitProgram, 219
 StereoRenderInfo, 220
 StereoView, 222
 StereoViewEye, 223
 StreamRedirect, 224
 StyleGuide, 225
 TAABB< T >, 226
 TargetSelector, 228
 TestState, 228
 onEntered, 229
 onLeft, 229
 TextFieldHudget, 230
 TextFieldHudgetVoxels, 231
 ThreadPool< T >, 232
 TimedTask, 233
 Timer, 234
 Transform, 235
 transit
 State, 218
 Transition, 236
 Trigger, 238
 Triggerable, 239
 TriggeredTransition, 240
 update
 Game, 107
 GamePlay, 109
 GamePlayPaused, 110
 GamePlayRunning, 112
 GameState, 119
 State, 219
 View, 241
 Viewer, 242
 Viewport, 243
 Visuals, 243
 Voxel, 244
 VoxelAccumulator< Accumulatable >, 246
 VoxelCluster, 247
 VoxelClusterBounds, 248
 VoxelCollision, 249
 VoxelCollisionAccumulator, 250
 VoxelCollisionParticipant, 251
 VoxelDebrisGenerator, 252
 VoxelExplosionGenerator, 253
 VoxelFont, 254
 VoxelGridCmp< highPriorityAxis, middlePriorityAxis, lowPriorityAxis >, 255
 VoxelHangman, 256
 VoxelMesh, 257
 VoxelNeighbourHelper, 258
 VoxelParticleData, 258
 VoxelParticleEngine, 259
 VoxelParticleExpireCheck, 260
 VoxelParticleFutureCheck, 261
 VoxelParticleIntersectionCheck, 261
 VoxelParticleRemoveCheck, 262
 VoxelParticleRemover, 263
 VoxelParticleRenderer, 264
 VoxelParticleSetup, 265
 VoxelParticleSpawnBase, 266
 VoxelRenderData, 267
 VoxelRenderer, 269
 VoxelTree, 270
 VoxelTreeNode, 271
 VoxelTreeQuery, 272
 Weapon, 273
 WeaponBuilder, 275
 World, 276
 WorldLogic, 277
 WorldObject, 278
 WorldObjectBindings, 280
 WorldObjectBuilder, 281
 WorldObjectCollision, 282
 WorldObjectComponents, 283
 WorldObjectDestroyedPoll, 284
 WorldObjectModification, 286
 WorldObjectPair, 287
 WorldObjectSlot, 288
 WorldTree, 289
 WorldTreeGeode, 289
 WorldTreeNode, 290
 WorldTreeQuery, 291
 WorldTreeScanner, 292