Voxellancer 0.3

Generated by Doxygen 1.8.6

Sun Mar 30 2014 13:32:42

# **Contents**

1	Main	Page	1
2	How	to play	3
3	Lua	API bindings	5
4	Build	d instructions	9
5	Hiera	archical Index	11
	5.1	Class Hierarchy	11
6	Clas	s Index	17
	6.1	Class List	17
7	Clas	s Documentation	23
	7.1	$\label{lem:lindices} \mbox{Luaw::\_indices} < \mbox{Is} > \mbox{Struct Template Reference}  .  .  .  .  .  .  .  .  .  $	23
	7.2	$\label{eq:Luaw::indices_builder} \mbox{Luaw::\_indices\_builder} < \mbox{N, Is} > \mbox{Struct Template Reference} \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $	23
	7.3	Luaw::_indices_builder< 0, Is> Struct Template Reference	23
	7.4	LuaWrapper::_pop< size_t, Ts > Struct Template Reference	23
	7.5	LuaWrapper::_pop< 0, Ts> Struct Template Reference	24
	7.6	$LuaWrapper::\_pop < 1, T > Struct \ Template \ Reference \\ \ \ldots \\ \ \ldots \\ \ \ldots$	24
	7.7	AABBEnteredPoll Class Reference	25
	7.8	AbstractMove Class Reference	26
	7.9	AbstractPropertyCollection Class Reference	26
	7.10	AbstractShape Class Reference	27
	7.11	ActionKeyMapping Class Reference	28
	7.12	AiBindings Class Reference	29
	7.13	AiGroupTask Class Reference	30
		7.13.1 Detailed Description	32
	7.14	Aimer Class Reference	32
	7.15	AimHelperHudget Class Reference	33
	7.16	AimHelperHudgetVoxels Class Reference	34
	7.17	AiTask Class Reference	35
		7.17.1 Detailed Description	36

iv CONTENTS

7.18	AiTaskFinishedPoll Class Reference	37
7.19	ArrowHudgetVoxels Class Reference	38
7.20	BaseLuaFunction Struct Reference	39
7.21	BaseScenario Class Reference	40
7.22	BattleScenario Class Reference	41
7.23	Bindings Class Reference	42
7.24	BlitProgram Class Reference	43
7.25	Blitter Class Reference	45
	7.25.1 Detailed Description	45
7.26	BoardComputer Class Reference	46
	7.26.1 Detailed Description	47
7.27	Bullet Class Reference	47
	7.27.1 Detailed Description	48
7.28	Camera Class Reference	48
	7.28.1 Detailed Description	50
7.29	CameraDolly Class Reference	50
7.30	CameraFollowHelper Class Reference	51
7.31	CameraHead Class Reference	52
7.32	Starfield::CameraLocation Struct Reference	53
7.33	Capsule Class Reference	53
7.34	Character Class Reference	54
	7.34.1 Detailed Description	55
7.35	CircularHudget Class Reference	55
7.36	ClusterCache Class Reference	56
7.37	ClusterLoader Class Reference	57
7.38	CockpitVoxel Class Reference	57
7.39	CollisionDetector Class Reference	58
7.40	CollisionFilter Class Reference	59
7.41	CollisionFilterIgnoringCreator Class Reference	61
7.42	ColorCoder Class Reference	62
7.43	CommandLineParser Class Reference	63
7.44	CommonBindings Class Reference	63
7.45	ContextDependant Class Reference	64
	7.45.1 Detailed Description	65
7.46	ContextProvider Class Reference	66
	7.46.1 Detailed Description	67
7.47	CrossHair Class Reference	67
7.48	CrossHairElement Class Reference	69
7.49	CrossHairVoxels Class Reference	70
7.50	CrucialVoxel Class Reference	71

CONTENTS

7.51	DamageForwarder Class Reference	72
7.52	DamageImpact Class Reference	73
7.53	DamageImpactGenerator Class Reference	74
7.54	Damager Class Reference	74
7.55	DdsTexture Class Reference	75
7.56	DefaultRenderPipeline Class Reference	75
	7.56.1 Detailed Description	77
7.57	DefendAreaTask Class Reference	77
7.58	DirectoryReader Class Reference	78
7.59	DirectSuicideTask Class Reference	79
7.60	ElasticImpulseGenerator Class Reference	81
7.61	ElasticImpulsor Class Reference	82
7.62	Engine Class Reference	83
7.63	EngineBuilder Class Reference	85
7.64	EnginePower Class Reference	86
7.65	EngineSlot Class Reference	86
7.66	EngineSlotVoxel Class Reference	88
7.67	EngineTrailGenerator Class Reference	89
7.68	Equipment Class Reference	90
7.69	EventPoll Class Reference	91
7.70	EventPoller Class Reference	92
7.71	ExternalMissionBindings Class Reference	93
	7.71.1 Detailed Description	94
7.72	Faction Class Reference	94
7.73	FactionMatrix Class Reference	96
7.74	FactionRelation Class Reference	97
7.75	FightTask Class Reference	98
7.76	FileSystem Class Reference	99
	7.76.1 Detailed Description	99
	7.76.2 Member Function Documentation	99
	7.76.2.1 userConfigDir	99
7.77	FlyToTask Class Reference	100
7.78	FormationMemberTask Class Reference	101
7.79	FrameBuffer Class Reference	102
7.80	FrozenGameScenario Class Reference	103
7.81	FuelVoxel Class Reference	105
7.82	Game Class Reference	106
	7.82.1 Detailed Description	107
	7.82.2 Member Function Documentation	107
	7.82.2.1 update	107

vi CONTENTS

7.83	GameF	ay Class Reference	107
	7.83.1	Detailed Description	109
	7.83.2	Member Function Documentation	109
		7.83.2.1 onEntered	109
		7.83.2.2 onLeft	109
		7.83.2.3 update	109
7.84	GameF	ayPaused Class Reference	110
	7.84.1	Member Function Documentation	111
		7.84.1.1 onEntered	111
		7.84.1.2 onLeft	111
		7.84.1.3 update	111
7.85	GameF	ayRunning Class Reference	111
	7.85.1	Member Function Documentation	112
		7.85.1.1 onEntered	112
		7.85.1.2 onLeft	112
		7.85.1.3 update	112
7.86	GameF	ayRunningInput Class Reference	113
	7.86.1	Member Function Documentation	114
		7.86.1.1 update	114
7.87	GameF	ayScene Class Reference	114
7.88	GameF	ayScript Class Reference	116
7.89	GameS	cenario Class Reference	117
7.90	GameS	ate Class Reference	118
	7.90.1	Detailed Description	119
	7.90.2	Member Function Documentation	119
		7.90.2.1 onEntered	119
		7.90.2.2 onLeft	119
		7.90.2.3 update	119
7.91	Garbag	Collector Class Reference	120
7.92	Generi	Bullet Class Reference	121
7.93	Generi	Engine Class Reference	122
7.94	Generi	Gun Class Reference	123
7.95	Generi	Rocket Class Reference	125
7.96	Generi	RocketLauncher Class Reference	126
7.97	Generi	Ship Class Reference	127
7.98	Generi	WorldObject Class Reference	128
			129
7.100	God Cl	ss Reference	130
	7.100.1	Detailed Description	130
7.10		BB Class Reference	131

CONTENTS vii

viii CONTENTS

7.137 Mission System Class Reference
7.138MonoView Class Reference
7.139Movement Class Reference
7.140 Mover Class Reference
7.141ObjectHudget Class Reference
7.142ObjectHudgetCornerVoxels Class Reference
7.143ObjectHudgetVoxels Class Reference
7.144ObjectInfo Class Reference
7.145PatrolWaypointsTask Class Reference
7.146 Physics Class Reference
7.147 Pirate Scenario Class Reference
7.148 Player Class Reference
7.149 Point Class Reference
7.150 PoliceFaction Class Reference
7.151 PostProcessingPass Class Reference
7.151.1 Detailed Description
7.152Projectile Class Reference
7.152.1 Detailed Description
7.153 Property < T > Class Template Reference
7.153.1 Detailed Description
7.154PropertyCollection< T > Class Template Reference
7.155PropertyConverter Class Reference
7.156PropertyDirectory Class Reference
7.156.1 Detailed Description
$7.157 Property Impl < T > Class Template Reference \dots 190$
7.158PropertyManager Class Reference
7.158.1 Detailed Description
7.159RandBool Class Reference
7.160 Rand Float Class Reference
7.161 Rand Vec 3 Class Reference
7.162Range Class Reference
7.163 Ray Class Reference
7.164RenderMetaData Class Reference
7.165RenderPass Class Reference
7.165.1 Detailed Description
7.166RenderPipeline Class Reference
7.167Rocket Class Reference
7.167.1 Detailed Description
7.168 Rocket Launcher Class Reference
7.168.1 Detailed Description

CONTENTS

7.169Scene Class Reference
7.170 Screen Blitter Class Reference
7.171 ScreenQuad Class Reference
7.172Script Class Reference
7.173Scriptable Class Reference
7.174ScriptedScenario Class Reference
7.175ScriptEngine Class Reference
7.176SecondaryInputValues Struct Reference
7.177Ship Class Reference
7.177.1 Detailed Description
7.178SimpleWayfind Class Reference
7.179SingleShotTimer Class Reference
7.180 Size < T > Class Template Reference
7.181 Skybox Class Reference
7.182Sound Class Reference
7.183SoundManager Class Reference
7.184SoundProperties Class Reference
7.185SpawnRequest Class Reference
7.185.1 Detailed Description
7.186SpecialVoxel Class Reference
7.187Sphere Class Reference
7.188SplitData Class Reference
7.189SplitDetector Class Reference
7.190 Splitter Class Reference
7.191 Squad Class Reference
7.191.1 Detailed Description
7.192SquadBindings Class Reference
7.193SquadLogic Class Reference
7.193.1 Detailed Description
7.194Starfield::StarData Struct Reference
7.195Starfield Class Reference
7.196State Class Reference
7.196.1 Detailed Description
7.196.2 Member Function Documentation
7.196.2.1 onEntered
7.196.2.2 onLeft
7.196.2.3 pathToDescendant
7.196.2.4 transit
7.196.2.5 update
7.197StereoBlitProgram Class Reference

CONTENTS

7.198StereoRenderInfo Class Reference
7.199StereoView Class Reference
7.200StereoViewEye Class Reference
7.201 StreamRedirect Class Reference
7.202TAABB< T > Class Template Reference
7.203 Target Selector Class Reference
7.204TextFieldHudget Class Reference
7.205TextFieldHudgetVoxels Class Reference
$7.206 Thread Pool < T > Class \ Template \ Reference \\  242 + 242 $
7.207TimedTask Class Reference
7.208Timer Class Reference
7.209 Transform Class Reference
7.210 Transition Class Reference
7.210.1 Detailed Description
7.211 Trigger Class Reference
7.212Triggerable Class Reference
7.213TriggeredTransition Class Reference
$7.214 glow:: Uniform < T > Class \ Template \ Reference \\ \ \ldots \\ \ \ldots \\ \ \ \ldots \\ \ \ \ \ \ \ \ \ \$
7.215 View Class Reference
7.216 Viewer Class Reference
7.217 Viewport Class Reference
7.218 Visuals Class Reference
7.218.1 Detailed Description
7.219 Voxel Class Reference
7.220 VoxelAccumulator < Accumulatable > Class Template Reference
7.221 VoxelCluster Class Reference
7.222 VoxelClusterBounds Class Reference
7.223 VoxelCollision Class Reference
7.224 VoxelCollisionAccumulator Class Reference
7.225 VoxelCollisionParticipant Class Reference
7.226 VoxelDebrisGenerator Class Reference
7.227 Voxel Explosion Generator Class Reference
7.228 VoxelFont Class Reference
$7.229 Voxel Grid Cmp < high Priority Axis, middle Priority Axis, low Priority Axis > Class \ Template \ Reference \ . \ \ 265 \ Reference \ . \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $
7.230 SplitDetector::VoxelGroup Struct Reference
7.231 VoxelHangman Class Reference
7.232 VoxelMesh Class Reference
7.233 VoxelNeighbourHelper Class Reference
7.234 VoxelParticleData Struct Reference
7.235 VoxelParticleEngine Class Reference

CONTENTS xi

7.236 VoxelParticleExpireCheck Class Reference
7.237 VoxelParticleFutureCheck Class Reference
7.238 VoxelParticleIntersectionCheck Class Reference
7.239 VoxelParticleRemoveCheck Class Reference
7.240 VoxelParticleRemover Class Reference
7.241 VoxelParticleRenderer Class Reference
7.242 VoxelParticleSetup Class Reference
7.243 VoxelParticleSpawnBase Class Reference
7.244 VoxelRenderData Class Reference
7.245 VoxelRenderer Class Reference
7.246 VoxelTree Class Reference
7.247 VoxelTreeNode Class Reference
7.248 VoxelTreeQuery Class Reference
7.249 Weapon Class Reference
7.249.1 Detailed Description
7.250 WeaponBuilder Class Reference
7.251 World Class Reference
7.252WorldLogic Class Reference
7.253WorldObject Class Reference
7.253.1 Detailed Description
7.254WorldObjectBindings Class Reference
7.255WorldObjectBuilder Class Reference
7.255.1 Detailed Description
7.256WorldObjectCollision Class Reference
7.257WorldObjectComponents Class Reference
7.257.1 Detailed Description
7.258WorldObjectDestroyedPoll Class Reference
7.259WorldObjectModification Class Reference
7.260 WorldObjectPair Struct Reference
7.261 WorldObjectSlot Class Reference
7.262WorldTree Class Reference
7.263WorldTreeGeode Class Reference
7.264WorldTreeNode Class Reference
7.264.1 Member Function Documentation
7.264.1.1 convertToGroup
7.264.1.2 moveToSubnode
7.265WorldTreeQuery Class Reference
7.266WorldTreeScanner Class Reference
7.267Wrecker Class Reference

xii CONTENTS

Index 305

# **Main Page**

A game about voxels in space Have fun! 2 Main Page

# How to play

You can fly around, fight and destroy other ships or do missions.

#### **Controls**

- · W,A,S,D direction
- · Left Mouse fire
- Right Mouse click to select target / hold to change direction
- · Space change to fight controls
- Q, E roll
- R Rocket
- P Pause
- F1-F5 change scenarios

You can configure the controls as you want. The instructions are currently only displayed in the console.

- F10 start gamepad/joystick configuration
- F11 start keyboard configuration

#### only for debugging:

- F8 reload shader and configs
- F9 world status information
- 1-9 show framebuffer

#### **Commandline parameters**

```
Usage: voxellancer [-fullscreen] [-stereo] [-hmd]

-fullscreen fullscreen mode
-stereo stereo rendering
-hmd use head mounted display (e.g. OculusRift)
automatically activates fullscreen and stereo
```

How to play

# Lua API bindings

generated from 10a4bcf

#### commonbindings.h

```
bool valid(apikey key);
int showMessage(const std::string& string);
int playVoice(const std::string& soundFile);
int setEventActive(apikey eventPoll, bool active);
apikey createSingleShotTimer(const std::string& callback, float delta);
apikey createLoopingTimer(const std::string& callback, float delta);
```

### worldobjectbindings.h

```
apikey playerShip();
apikey createShip(const std::string& name);
apikey createWorldObject(const std::string& name);
int spawn(apikey worldObject);
int remove(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);
int setShowOnHud(apikey worldObject);
int setShowOnHud(apikey worldObject, bool show);
int setCanLockOn(apikey worldObject, bool lockon);
apikey onWorldObjectDestroyed(apikey worldObject, const std::string& callback);
apikey onAABBEntered(apikey worldObject, const glm::vec3& llf, const glm::vec3& urb, const std::string& callback);
```

#### externalmissionbindings.h

```
apikey missionStart(const std::string& name);
apikey onMissionFailure(apikey missionKey, const std::string& callback);
apikey onMissionSuccess(apikey missionKey, const std::string& callback);
```

#### internalmissionbindings.h

```
int missionSucceed();
int missionFail();
int missionMessage(const std::string& message);
int missionFailureMessage(const std::string& message);
int missionSuccessMessage(const std::string& message);
```

6 Lua API bindings

#### squadbindings.h

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

#### aibindings.h

```
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 ship);
int addFightTaskTarget(apikey flyToTask, apikey worldObject);
apikey taskExecutor(apikey aiTask);
```

### Ini Objects

#### engines

- · candle
- · enginemk1
- · enginemk3
- · loudengine
- mginemk1
- · piratethruster
- · piratethrusterheavy
- · rocketthrustermk1
- · superslowengine

#### **bullets**

- · gunbullet
- ionbullet
- · pulselaser\_bullet
- · railgunbullet
- · snowball

#### rockets

hornet

#### guns

- gun
- ioncannon
- · pulselaser
- railgun
- snowcanon

### rocketlaunchers

hornetlauncher

## worldobjects

- banner
- cake
- idareyou
- · missionstart

## ships

- basicship
- bc304
- c306
- eagle
- f301
- f302
- flyingtortress
- mox
- normandy
- piratefrigatte
- pirategunboat
- pirateheavy
- piratelight
- smallpolice
- · specialbasicship
- startortress

8 Lua API bindings

### colorcodess

• engineSlot: 0x000E

• hardpoint: 0x000D

• cockpit: 0x000C

• fuel: 0x000F

• crucial: 0xC00C

# **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
```

10 **Build instructions** 

# **Hierarchical Index**

# 5.1 Class Hierarchy

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

Luaw::_indices < ls >
$\label{luaw::indices_builder} Luaw::\_indices\_builder < N, \ ls > \ \dots \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$
Luaw::_indices_builder< 0, ls>
LuaWrapper::_pop< size_t, Ts >
LuaWrapper::_pop< 0, Ts>
$LuaWrapper::\_pop < 1, T > \dots \dots$
AbstractMove
AbstractPropertyCollection
PropertyCollection< T >
AbstractShape
Capsule
Line
Point
Ray
Sphere
TAABB< T >
$TAABB < glm : mediump\_float > } . } . } 236$
TAABB< int >
ActionKeyMapping
Aimer
AimHelperHudgetVoxels
ArrowHudgetVoxels
BaseLuaFunction
LuaFunction < N, Return, Args >
BaseScenario
BattleScenario
FrozenGameScenario
GameScenario
MissionScenario
PirateScenario
ScriptedScenario
basic_streambuf
StreamRedirect
Bindings
AiBindings
CommonBindings
ExternalMissionBindings

12 Hierarchical Index

InternalMissionBindings				
SquadBindings				. 225
WorldObjectBindings				. 290
BoardComputer	 	 		46
CameraFollowHelper				
CameraHead				
Starfield::CameraLocation				
Changeable				
PropertyManager				. 190
Character				
ClusterCache				
ClusterLoader				
CollisionDetector				
CollisionFilter				
CollisionFilterIgnoringCreator				
ColorCoder				
CommandLineParser				
ContextDependant				
BlitProgram				
StereoBlitProgram				
FrameBuffer				
PostProcessingPass				. 182
Blitter				. 45
ScreenQuad				. 202
Skybox				
Starfield				. 228
StereoView				. 235
VoxelMesh				. 267
VoxelParticleEngine				. 269
VoxelParticleRenderer				. 274
VoxelRenderData				. 277
VoxelRenderer				. 279
ContextProvider	 	 		66
CrossHairVoxels				
DamageForwarder				
DamageImpact				
DamageImpactGenerator				74
Damager				74
DdsTexture	 	 		75
DirectoryReader	 	 		78
ElasticImpulseGenerator				81
ElasticImpulsor				82
EngineBuilder				85
EnginePower	 	 		86
Engine Trail Generator	 	 		89
Equipment				90
Engine				83
GenericEngine				
Weapon				
•				
Gun				
GenericGun				
RocketLauncher				
GenericRocketLauncher	 			. 126
EventPoller		 		92
Faction		 		94
PoliceFaction				. 181
	 •		•	

5.1 Class Hierarchy 13

FactionMatrix	96
FactionRelation	97
FileSystem	99
GamePlayRunningInput	
GarbageCollector	
GeometryHelper	
God	
GridAABB	
Handle < T >	
Handle AiTask >	
Handle < EventPoll >	
Handle < Mission >	
Handle Ship >	
Handle < World Object >	
$HandleImpl < T > \dots \dots$	
HardpointAimHelper	
$std :: hash < glm :: ivec3 > \dots $	
$std::hash < pair < A, B >> \dots$	138
HMD	138
HMDManager	139
HUD	140
HUDElements	141
Hudget	142
CircularHudget	
•	
AimHelperHudget	
CrossHair	
ObjectHudget	
TextFieldHudget	240
HudgetAnimation	1//
	144
HudgetHideAnimation	
HudgetHideAnimation	146
HUDObjectDelegate	146
HUDObjectDelegate	146
HUDObjectDelegate	146 147 148
HUDObjectDelegate	
HUDObjectDelegate Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigurator InputConfigWriter InputMapping LuaWrapper	
HUDObjectDelegate Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigurator InputConfigWriter InputMapping LuaWrapper Math	
HUDObjectDelegate Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigWriter InputMapping LuaWrapper Math Metrics	
HUDObjectDelegate Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigWriter InputMapping LuaWrapper Math Metrics MissionSystem	
HUDObjectDelegate Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigWriter InputMapping LuaWrapper Math Metrics MissionSystem Movement	
HUDObjectDelegate Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigWriter InputMapping LuaWrapper Math Metrics MissionSystem Movement Mover	
HUDObjectDelegate Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigWriter InputMapping LuaWrapper Math Metrics MissionSystem Movement Mover ObjectHudgetCornerVoxels	
HUDObjectDelegate Impulse Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigWriter InputMapping LuaWrapper Math Metrics MissionSystem Movement Mover ObjectHudgetCornerVoxels ObjectHudgetVoxels	
HUDObjectDelegate Impulse Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigWriter InputMapping LuaWrapper Math Metrics MissionSystem Movement Mover ObjectHudgetCornerVoxels ObjectInfo	
HUDObjectDelegate Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigWriter InputMapping LuaWrapper Math Metrics MissionSystem Movement Mover ObjectHudgetCornerVoxels ObjectInfo Physics	
HUDObjectDelegate Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigWriter InputMapping LuaWrapper Math Metrics MissionSystem Movement Mover ObjectHudgetCornerVoxels ObjectInfo Physics Player	
HUDObjectDelegate Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigWriter InputMapping LuaWrapper Math Metrics MissionSystem Movement Mover ObjectHudgetCornerVoxels ObjectInfo Physics Player Program	
HUDObjectDelegate Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigWriter InputMapping LuaWrapper Math Metrics MissionSystem Movement Mover ObjectHudgetCornerVoxels ObjectHudgetVoxels ObjectInfo Physics Player Program BlitProgram	
HUDObjectDelegate Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigWriter InputMapping LuaWrapper Math Metrics MissionSystem Movement Mover ObjectHudgetCornerVoxels ObjectInfo Physics Player Program	
HUDObjectDelegate Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigWriter InputMapping LuaWrapper Math Metrics MissionSystem Movement Mover ObjectHudgetCornerVoxels ObjectHudgetVoxels ObjectInfo Physics Player Program BlitProgram	
HUDObjectDelegate Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigWriter InputMapping LuaWrapper Math Metrics MissionSystem Movement Mover ObjectHudgetCornerVoxels ObjectHudgetVoxels ObjectInfo Physics Player Program BlitProgram Property< T >	
HUDObjectDelegate   Impulse   ImputConfigwer   ImputConfigurator   ImputConfigWriter   ImputMapping   ImputMapping	
HUDObjectDelegate Impulse Impulse	
HUDObjectDelegate Impulse Impulse Impulse Impulse Impulse Impulse ImpulseAccumulator InertiaFollower CameraDolly InputConfigurator InputConfigurator InputMapping LuaWrapper Math Metrics MissionSystem Movement Mover ObjectHudgetCornerVoxels ObjectHudgetVoxels ObjectHudgetVoxels ObjectInfo Physics Player Program BlitProgram Property < T > Property < book by the first property < float > Property < glm::vec3 > Property < glm::vec3 > Property < glm::vec3 > Property < glm::vec3 >	

14 Hierarchical Index

$\label{property} {\sf Property} < {\sf std::string} > \ \ldots \$	186
$\label{eq:property} Property < uint 32\_t > \dots $	
PropertyConverter	
PropertyDirectory	
$PropertyImpl < T > \dots \dots$	
$PropertyImpl < bool > \dots $	
$PropertyImpl < float > \dots $	
$PropertyImpl < glm::vec3 > \dots $	
PropertyImpl < InputMapping >	
PropertyImpl< int >	
PropertyImpl < std::string >	
PropertyImpl < uint32_t >	
RandBool	
RandFloat	
RandVec3	
Range	
RenderMetaData	
RenderPass	
PostProcessingPass	
RenderPipeline	
DefaultRenderPipeline	
Starfield	228
Scene	201
GamePlayScene	114
ScreenBlitter	202
Script	
GamePlayScript	
MissionScript	
·	
Scriptable	
AiGroupTask	
DefendAreaTask	
PatrolWaypointsTask	
AiTask	
DirectSuicideTask	
FightTask	
FlyToTask	
FormationMemberTask	
EventPoll	
AABBEnteredPoll	
AiTaskFinishedPoll	
MissionStatePoll	
Timer	
LoopingTimer	
SingleShotTimer	
WorldObjectDestroyedPoll	
Mission	
Squad	_
WorldObject	
GenericWorldObject	
Projectile	184
Bullet	47
GenericBullet	121
Rocket	197
GenericRocket	125
Ship	209
GenericShip	
•	

5.1 Class Hierarchy 15

ScriptEngine	208
SecondaryInputValues	209
SimpleWayfind	211
$Size \! < T \! > \dots \dots$	213
$Size < int > \ \ldots \$	
Sound	
SoundManager	
SoundProperties	
SpawnRequest	
SplitData	
SplitDetector	
Splitter	
SquadLogic      Starfield::StarData	
State	
GameState	
GamePlay	
GamePlayPaused	
GamePlayRunning	
StereoRenderInfo	
StereoViewEye	
TargetSelector	
TextFieldHudgetVoxels	
ThreadPool < T >	
ThreadPool < VoxelParticleData >	
TimedTask	
Transform	
Camera	48
Transition	
TriggeredTransition	
Trigger	
KeyTrigger	
Triggerable	
TriggeredTransition	
glow::Uniform< T >	
glow::Uniform< float >	
glow::Uniform< glm::mat4 >	
View	
MonoView	
StereoView	
	252
- P	253
Visuals	253
SpecialVoxel	
CockpitVoxel	
CrucialVoxel	
FuelVoxel	
HardpointVoxel	
•	256
	256 256
VoxelCluster	
CrossHairElement	
20101	100

16 Hierarchical Index

WorldObject
VoxelClusterBounds
VoxelCollision
VoxelCollisionAccumulator
VoxelCollisionParticipant
VoxelFont
VoxelGridCmp< highPriorityAxis, middlePriorityAxis, lowPriorityAxis>
SplitDetector::VoxelGroup
VoxelHangman
VoxelNeighbourHelper
VoxelParticleData
VoxelParticleFutureCheck
VoxelParticleRemoveCheck
VoxelParticleExpireCheck
VoxelParticleIntersectionCheck
VoxelParticleRemover
VoxelParticleSetup
VoxelParticleSpawnBase
VoxelDebrisGenerator
VoxelExplosionGenerator
VoxelTree
VoxelTreeNode
VoxelTreeQuery
WeaponBuilder
World
WorldLogic
WorldObjectBuilder
WorldObjectCollision
WorldObjectComponents
WorldObjectModification
WorldObjectPair
WorldObjectSlot
EngineSlot
Hardpoint
WorldTree
WorldTreeGeode
WorldTreeNode         300           WorldTreeOuters         300
WorldTreeQuery
WorldTreeScanner
Wrecker

# **Class Index**

## 6.1 Class List

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

Luaw::_indices < ls >
Luaw::_indices_builder< N, ls >
Luaw::_indices_builder< 0, ls>
LuaWrapper::_pop< size_t, Ts >
LuaWrapper::_pop< 0, Ts>
LuaWrapper::_pop< 1, T >
AABBEnteredPoll
AbstractMove
AbstractPropertyCollection
AbstractShape
ActionKeyMapping
AiBindings
AiGroupTask
Aimer
AimHelperHudget
AimHelperHudgetVoxels
AiTask
AiTaskFinishedPoll
ArrowHudgetVoxels
BaseLuaFunction
BaseScenario
BattleScenario
Bindings
BlitProgram
Blitter
BoardComputer
Bullet 47
Camera
CameraDolly
CameraFollowHelper
CameraHead
Starfield::CameraLocation
Capsule
Character
CircularHudget
ClusterCache
ClusterLoader
CockpitVoxel 57

18 Class Index

CollisionDetector	 		58
CollisionFilter	 		59
CollisionFilterIgnoringCreator	 		61
ColorCoder			
CommandLineParser			
CommonBindings			
ContextDependant			
ContextProvider			
CrossHair			
CrossHairElement			
CrossHairVoxels			
CrucialVoxel			
DamageForwarder			
DamageImpact			
DamageImpact			
Damager			
DdsTexture			
DefaultRenderPipeline			
DefendAreaTask			
DirectoryReader			
DirectSuicideTask			
ElasticImpulseGenerator			
ElasticImpulsor			
Engine			
EngineBuilder			
EnginePower			
EngineSlot			
EngineSlotVoxel			
Engine Trail Generator	 		89
Equipment	 		90
EventPoll	 		91
EventPoller	 		92
External Mission Bindings	 		93
Faction	 		94
FactionMatrix	 		96
FactionRelation	 		97
FightTask	 		98
FileSystem	 		99
FlyToTask	 		100
FormationMemberTask	 		101
FrameBuffer	 		102
FrozenGameScenario	 		103
FuelVoxel	 		105
Game	 		106
GamePlay			107
GamePlayPaused			110
GamePlayRunning			111
GamePlayRunningInput			113
GamePlayScene			114
•		٠.	116
GamePlayScript			_
GameScenario			117
GameState			118
GarbageCollector			120
GenericBullet			121
GenericEngine			122
GenericGun			123
GenericRocket			125
GenericRocketLauncher	 		126

6.1 Class List

GenericShip	127
GenericWorldObject	128
GeometryHelper	129
God	130
GridAABB	131
Gun	131
Handle < T >	132
HandleImpl< T >	133
Hardpoint	133
HardpointAimHelper	135 135
HardpointVoxel	136
std::hash< glm::ivec3 >	137
std::hash< pair< A, B >>	138
HMD	138
HMDManager	139
HUD	140
HUDElements	141
Hudget	142
HudgetAnimation	144
HudgetHideAnimation	146
HUDObjectDelegate	147
Impulse	148
ImpulseAccumulator	149
InertiaFollower	150
InputConfigurator	150
InputConfigWriter	150
InputMapping	
Internal Mission Bindings	
KeyTrigger	
Letter	
Line	
LoopingTimer	
LuaFunction < N, Return, Args >	
LuaWrapper	159
Math	161
Metrics	162
Mission	163
MissionScenario	164
MissionScript	165
MissionStatePoll	167
MissionSystem	168
MonoView	169
Movement	170
Mover	171
ObjectHudget	171 172
ObjectHudgetCornerVoxels	172
	_
ObjectHudgetVoxels	174
ObjectInfo	175
PatrolWaypointsTask	176
Physics	177
PirateScenario	178
Player	179
Point	180
PoliceFaction	181
PostProcessingPass	182
Projectile	184
Property < T >	186
PropertyCollection < T >	

20 Class Index

PropertyConverter	. 188
PropertyDirectory	. 188
PropertyImpl < T >	. 190
PropertyManager	. 190
RandBool	. 192
RandFloat	. 192
RandVec3	. 192
Range	. 193
Ray	. 193
RenderMetaData	. 194
RenderPass	. 195
RenderPipeline	. 196
Rocket	. 197
RocketLauncher	. 200
Scene	. 201
ScreenBlitter	. 202
ScreenQuad	. 202
Script	. 204
Scriptable	. 206
ScriptedScenario	. 207
ScriptEngine	. 208
SecondaryInputValues	. 209
Ship	. 209
SimpleWayfind	. 211
SingleShotTimer	. 212
Size < T >	. 213
Skybox	. 214
Sound	. 215
SoundManager	. 216
SoundProperties	
SpawnRequest	
SpecialVoxel	
Sphere	
SplitData	
SplitDetector	
Splitter	
Squad	
SquadBindings	. 225
SquadLogic	. 226
Starfield::StarData	. 227
Starfield	. 228
	. 220
	. 232
StereoBlitProgram	
StereoRenderInfo	. 233
StereoView	. 235
StereoViewEye	. 236
StreamRedirect	. 237
TAABB< T >	. 238
TargetSelector	. 239
TextFieldHudget	. 240
TextFieldHudgetVoxels	. 241
$\label{eq:ThreadPool} \mbox{ThreadPool} < \mbox{T} > \dots $	. 242
TimedTask	. 243
Timer	. 244
Transform	. 245
Transition	. 246
Trigger	. 248
Triggerable	. 249

6.1 Class List 21

TriggeredTransition
glow::Uniform< T >
View
Viewer
Viewport
Visuals
Voxel
VoxelAccumulator< Accumulatable >
VoxelCluster
VoxelClusterBounds
VoxelCollision
VoxelCollisionAccumulator
VoxelCollisionParticipant
VoxelDebrisGenerator
VoxelExplosionGenerator
VoxelFont
VoxelGridCmp< highPriorityAxis, middlePriorityAxis, lowPriorityAxis >
SplitDetector::VoxelGroup
VoxelHangman
VoxelMesh
VoxelNeighbourHelper
VoxelParticleData
VoxelParticleEngine
VoxelParticleExpireCheck
VoxelParticleFutureCheck
VoxelParticleIntersectionCheck
VoxelParticleRemoveCheck
VoxelParticleRemover
VoxelParticleRenderer
VoxelParticleSetup
VoxelParticleSpawnBase
VoxelRenderData
VoxelRenderer
VoxelTree
VoxelTreeNode
VoxelTreeQuery
Weapon
WeaponBuilder
World
WorldLogic
WorldObject
WorldObjectBindings
,
<b> </b>
WorldObjectCollision
WorldObjectComponents
WorldObjectDestroyedPoll 29
WorldObjectModification
WorldObjectPair
WorldObjectSlot         29
WorldTree
WorldTreeGeode
WorldTreeNode         30
WorldTreeQuery
WorldTreeScanner
Wrecker

22 Class Index

# **Class Documentation**

## 7.1 Luaw::\_indices < Is > Struct Template Reference

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

• src/scripting/elematelua/luawrapperfunction.h

## 7.2 Luaw::\_indices\_builder< N, Is > Struct Template Reference

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

· src/scripting/elematelua/luawrapperfunction.h

### 7.3 Luaw::\_indices\_builder< 0, Is...> Struct Template Reference

#### **Public Types**

• using **type** = <u>indices</u>< ls...>

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

· src/scripting/elematelua/luawrapperfunction.h

## 7.4 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)

24 Class Documentation

The documentation	for this c	etrijet wae	nanaratad	from the	following	filo.

• src/scripting/elematelua/luawrapper.h

## 7.5 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

## 7.6 LuaWrapper::\_pop< 1, T > Struct Template Reference

### **Public Types**

• typedef T type

#### **Static Public Member Functions**

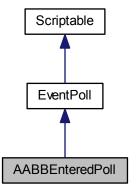
• static T apply (LuaWrapper &instance)

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

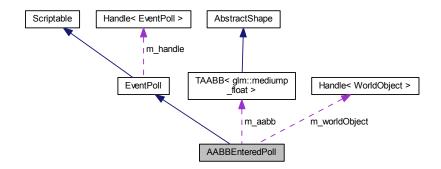
· src/scripting/elematelua/luawrapper.h

# 7.7 AABBEnteredPoll Class Reference

Inheritance diagram for AABBEnteredPoll:



Collaboration diagram for AABBEnteredPoll:



#### **Public Member Functions**

- AABBEnteredPoll (WorldObject \*worldObject, const AABB &aabb, const std::function< void()> &callback)
- virtual bool isDead () override

# **Protected Member Functions**

- virtual bool poll () override
- virtual void specialOnCallback () override

#### **Protected Attributes**

Handle < WorldObject > m\_worldObject

- · AABB m\_aabb
- · bool m\_entered

#### **Additional Inherited Members**

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

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

#### 7.8 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 dividor) const

## **Protected Attributes**

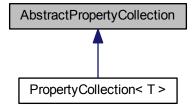
- glm::vec3 m\_directional
- glm::vec3 m\_angular

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

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

# 7.9 AbstractPropertyCollection Class Reference

Inheritance diagram for AbstractPropertyCollection:



**Public Member Functions** 

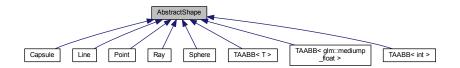
• virtual bool update (const std::string &key, const std::string &svalue)=0

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

• src/property/propertycollection.h

# 7.10 AbstractShape Class Reference

Inheritance diagram for AbstractShape:



# **Public Member Functions**

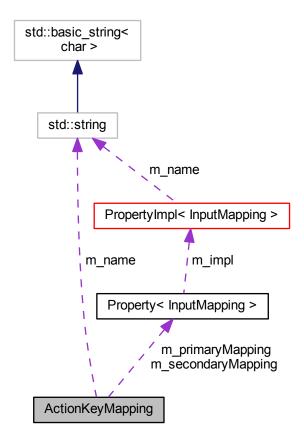
- virtual bool **intersects** (const Sphere &sphere) const =0
- virtual bool nearTo (const TAABB< int > &aabb) const =0
- virtual bool containedBy (const TAABB< int > &aabb) const =0

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

• src/geometry/abstractshape.h

# 7.11 ActionKeyMapping Class Reference

Collaboration diagram for ActionKeyMapping:



#### **Public Member Functions**

- ActionKeyMapping (std::string primary, std::string secondary, std::string name, bool toggleAction=false)
- InputMapping mapping (InputClass inputClass)
- void **setMapping** (InputClass inputClass, InputMapping mapping)
- const std::string & name ()
- const std::string & mappingName (InputClass inputClass)
- bool toggleAction ()
- bool toggleStatus ()
- void setToggleStatus (bool status)

## **Protected Attributes**

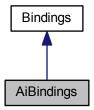
- Property < InputMapping > m\_primaryMapping
- Property < InputMapping > m\_secondaryMapping
- bool m\_toggleAction
- bool m\_toggleStatus
- std::string m\_name

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

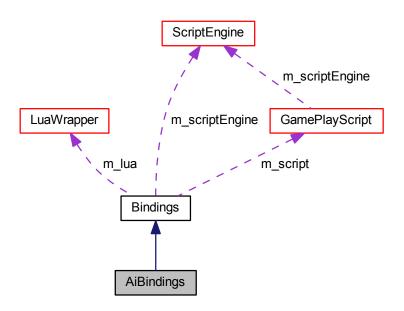
- · src/ui/actionkeymapping.h
- · src/ui/actionkeymapping.cpp

# 7.12 AiBindings Class Reference

Inheritance diagram for AiBindings:



Collaboration diagram for AiBindings:



# **Public Member Functions**

• AiBindings (GamePlayScript &script)

#### **Protected Member Functions**

- · virtual void bind () override
- 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)
- apikey apiTaskExecutor (apikey aiTask)

#### **Additional Inherited Members**

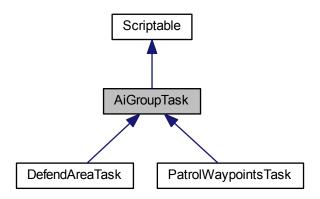
The documentation for this class was generated from the following files:

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

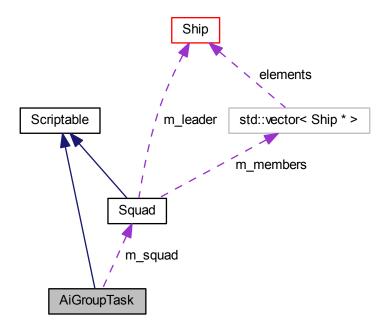
# 7.13 AiGroupTask Class Reference

#include <aigrouptask.h>

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**

#### 7.13.1 Detailed Description

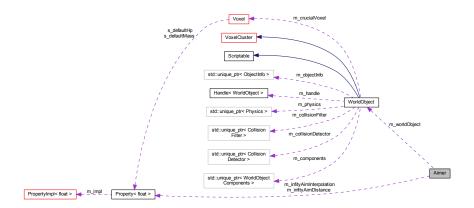
AiGroupTasks are AiTasks on the level of Squads, e.g. can be executed by multiple Characters / Ships They work by setting the Tasks of the single Characters

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

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

# 7.14 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**

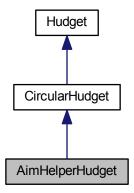
- WorldObject \* m\_worldObject
- Property< float > m\_infityAimDistance
- Property< float > m\_infityAimInterpolation
- · float m lastDistance

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

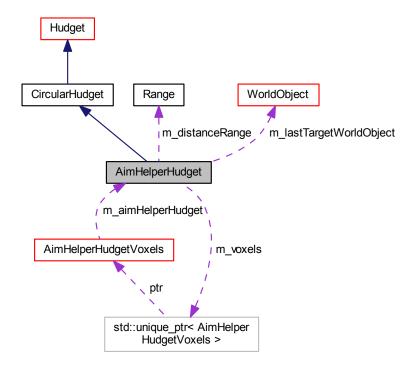
- · src/utils/aimer.h
- · src/utils/aimer.cpp

# 7.15 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**

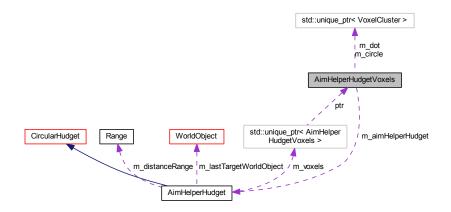
- std::unique\_ptr< AimHelperHudgetVoxels > m voxels
- glm::vec3 m\_targetPoint
- glm::vec3 m\_smoothTargetPoint
- WorldObject \* m\_lastTargetWorldObject
- bool m\_lastVisible
- Range m\_distanceRange

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

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

# 7.16 AimHelperHudgetVoxels Class Reference

Collaboration diagram for AimHelperHudgetVoxels:



#### **Public Member Functions**

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

# **Protected Attributes**

- AimHelperHudget \* m\_aimHelperHudget
- $std::unique\_ptr < VoxelCluster > m\_dot$
- $std::unique\_ptr < VoxelCluster > m\_circle$

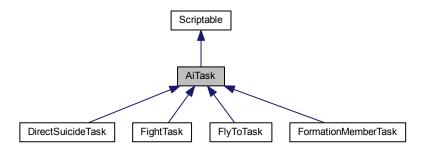
The documentation for this class was generated from the following files:

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

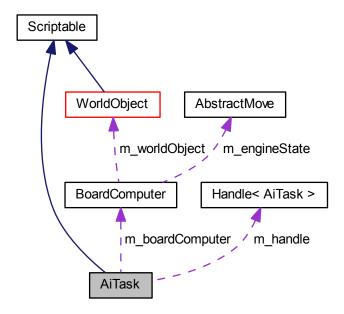
# 7.17 AiTask Class Reference

#include <aitask.h>

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**

## 7.17.1 Detailed Description

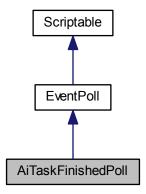
Basic unit for Al. A Ship's Character can always execute only one task.

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

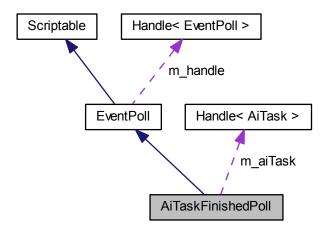
- src/ai/aitask.h
- · src/ai/aitask.cpp

# 7.18 AiTaskFinishedPoll Class Reference

Inheritance diagram for AiTaskFinishedPoll:



Collaboration diagram for AiTaskFinishedPoll:



#### **Public Member Functions**

• AiTaskFinishedPoII (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

# 7.19 ArrowHudgetVoxels Class Reference

Collaboration diagram for ArrowHudgetVoxels:



#### **Public Member Functions**

- ArrowHudgetVoxels (ObjectHudget \*hudget)
- ObjectHudget \* 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**

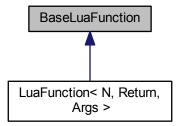
- ObjectHudget \* m\_hudget
- VoxelCluster m\_arrow
- Property< float > prop\_arrowDistance
- glm::vec3 m targetPoint

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

- src/ui/hud/arrowhudgetvoxels.h
- · src/ui/hud/arrowhudgetvoxels.cpp

# 7.20 BaseLuaFunction Struct Reference

Inheritance diagram for BaseLuaFunction:



### **Public Member Functions**

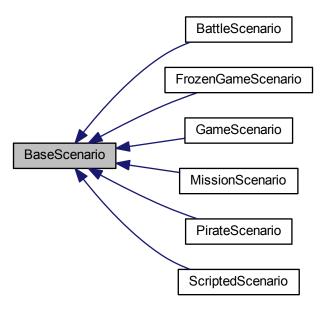
• virtual int apply (lua\_State \*state)=0

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

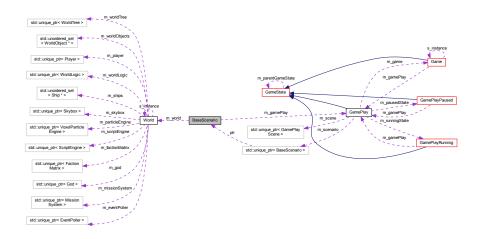
• src/scripting/elematelua/luawrapperfunction.h

# 7.21 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**

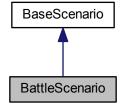
- GamePlay \* m\_gamePlay
- World \* m world

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

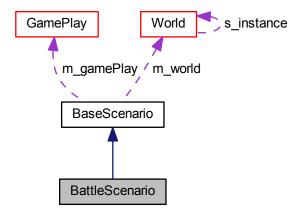
- src/scenarios/basescenario.h
- src/scenarios/basescenario.cpp

# 7.22 BattleScenario Class Reference

Inheritance diagram for BattleScenario:



Collaboration diagram for BattleScenario:



# **Public Member Functions**

• BattleScenario (GamePlay \*gamePlay)

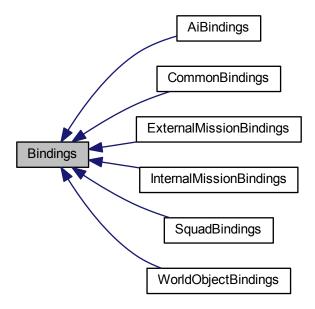
#### **Additional Inherited Members**

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

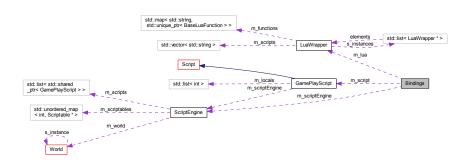
- · src/scenarios/battlescenario.h
- src/scenarios/battlescenario.cpp

# 7.23 Bindings Class Reference

Inheritance diagram for Bindings:



Collaboration diagram for Bindings:



# **Public Member Functions**

- Bindings (GamePlayScript &gamePlayScript)
- virtual void bind ()=0

# **Protected Attributes**

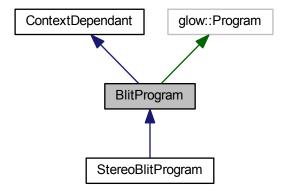
- LuaWrapper & m\_lua
- GamePlayScript & m\_script
- ScriptEngine & m\_scriptEngine

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

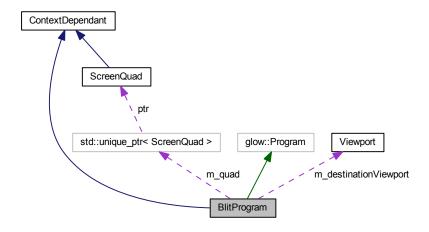
- src/scripting/bindings/bindings.h
- src/scripting/bindings/bindings.cpp

# 7.24 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:

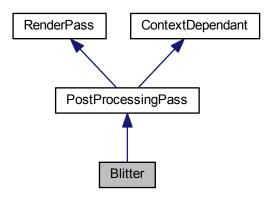
- · src/programs/blitprogram.h
- · src/programs/blitprogram.cpp
- · src/programs/blitprogram.inl

7.25 Blitter Class Reference 45

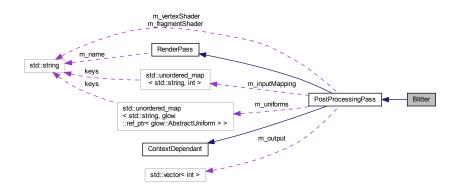
# 7.25 Blitter Class Reference

#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**

# 7.25.1 Detailed Description

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

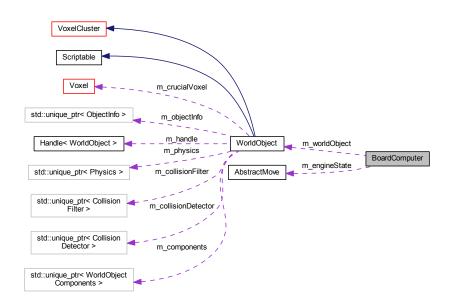
The documentation for this class was generated from the following files:

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

# 7.26 BoardComputer Class Reference

#include <boardcomputer.h>

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

7.27 Bullet Class Reference 47

# 7.26.1 Detailed Description

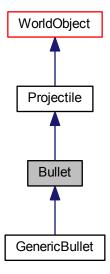
The BoardComputer abstracts the Ship's systems to a level usable to AiTask, Character and Player The documentation for this class was generated from the following files:

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

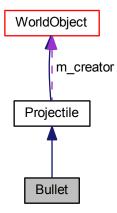
# 7.27 Bullet Class Reference

#include <bullet.h>

Inheritance diagram for Bullet:



#### Collaboration diagram for Bullet:



# **Public Member Functions**

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

### **Additional Inherited Members**

# 7.27.1 Detailed Description

Base class for every Projectile that flies with a constant undamped speed on a target and is destroyed on collision. Also, not displayed on HUD

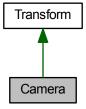
The documentation for this class was generated from the following files:

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

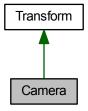
# 7.28 Camera Class Reference

#include <camera.h>

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

### 7.28.1 Detailed Description

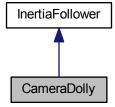
Represents the camera. matrix thus is the view matrix for all other objects

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

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

# 7.29 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**

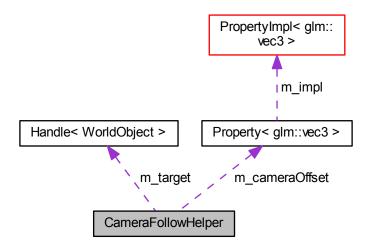
- std::unique\_ptr< CameraHead > m\_cameraHead
- std::unique\_ptrCameraFollowHelper > m\_followHelper

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

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

# 7.30 CameraFollowHelper Class Reference

Collaboration diagram for CameraFollowHelper:



#### **Public Member Functions**

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

#### **Protected Attributes**

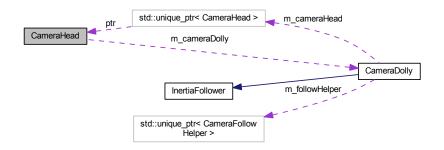
- Handle < WorldObject > m\_target
- Property < glm::vec3 > m\_cameraOffset

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

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

# 7.31 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

# 7.32 Starfield::CameraLocation Struct Reference

#### **Public Attributes**

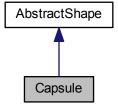
- · float time
- glm::vec3 position
- glm::quat orientation

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

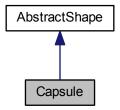
• src/display/rendering/starfield.h

# 7.33 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

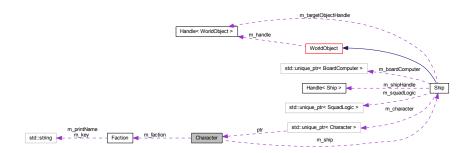
The documentation for this class was generated from the following files:

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

# 7.34 Character Class Reference

#include <character.h>

Collaboration diagram for Character:



## **Public Member Functions**

- Character (Ship &ship, Faction &faction)
- Faction & faction ()
- void setFaction (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 faction
- std::shared\_ptr< AiTask > m\_task

# 7.34.1 Detailed Description

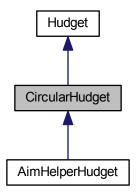
The Character is the Ship's pilot and executes his AiTask. He has a Faction which decides who is a friend or foe to him

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

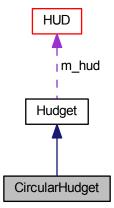
- · src/ai/character.h
- src/ai/character.cpp

# 7.35 Circular Hudget 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**

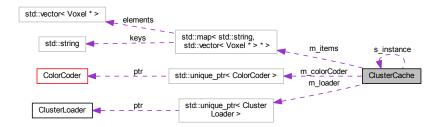
· float m\_radius

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

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

#### 7.36 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

# 7.37 ClusterLoader Class Reference

**Public Member Functions** 

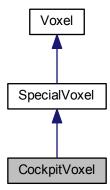
void load (const std::string &filename, std::vector< Voxel \* > \*list)

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

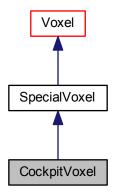
- src/resource/clusterloader.h
- src/resource/clusterloader.cpp

# 7.38 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

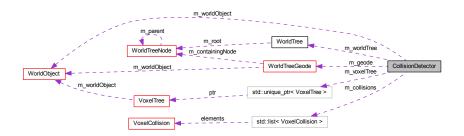
# **Additional Inherited Members**

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

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

# 7.39 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**

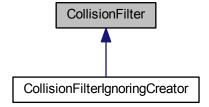
- std::unique\_ptr< VoxelTree > m\_voxelTree
- WorldObject & m\_worldObject
- WorldTreeGeode \* m\_geode
- WorldTree \* m\_worldTree
- std::list< VoxelCollision > m\_collisions

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

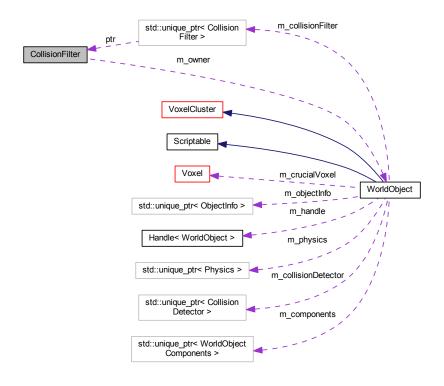
- src/collision/collisiondetector.h
- src/collision/collisiondetector.cpp

## 7.40 CollisionFilter Class Reference

Inheritance diagram for CollisionFilter:



### Collaboration diagram for CollisionFilter:



#### **Public Member Functions**

- CollisionFilter (WorldObject \*owner, uint32\_t collisionMask=0xFFFFFFF)
- 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 speciallsCollideableWith (const CollisionFilter \*other) const

### **Protected Attributes**

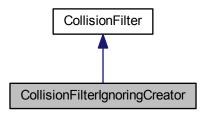
- uint32\_t m\_collisionMask
- WorldObject \* m\_owner

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

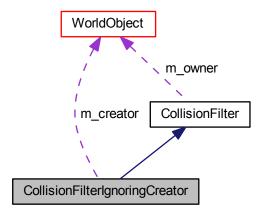
- · src/collision/collisionfilter.h
- · src/collision/collisionfilter.cpp

## 7.41 CollisionFilterIgnoringCreator Class Reference

Inheritance diagram for CollisionFilterIgnoringCreator:



Collaboration diagram for CollisionFilterIgnoringCreator:



## **Public Member Functions**

- CollisionFilterIgnoringCreator (WorldObject \*owner, WorldObject \*creator, uint32\_t collisionMask=0xFFF-
- virtual WorldObject \* creator () const override

#### **Protected Member Functions**

• virtual bool **speciallsCollideableWith** (const CollisionFilter \*other) const override

### **Protected Attributes**

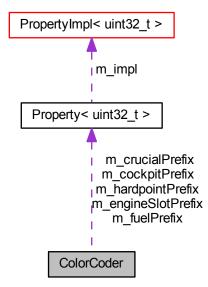
WorldObject \* m\_creator

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

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

## 7.42 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

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

## 7.43 CommandLineParser Class Reference

### **Public Member Functions**

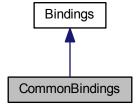
- void parse (int argc, char \*argv[])
- · bool hmd () const
- bool stereoView () const
- bool fullScreen () const

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

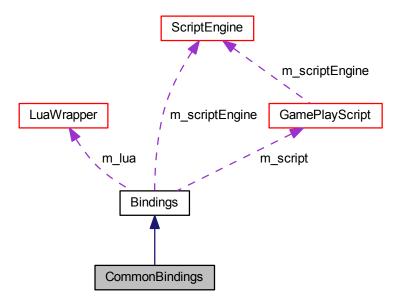
- src/etc/cli/commandlineparser.h
- src/etc/cli/commandlineparser.cpp

## 7.44 CommonBindings Class Reference

Inheritance diagram for CommonBindings:



Collaboration diagram for CommonBindings:



## **Public Member Functions**

• CommonBindings (GamePlayScript &script)

### **Protected Member Functions**

- · virtual void bind () override
- bool apiValid (apikey key)
- int apiShowMessage (const std::string &string)
- int apiPlayVoice (const std::string &soundFile)
- int apiSetEventActive (apikey eventPoll, bool active)
- apikey apiCreateSingleShotTimer (const std::string &callback, float delta)
- apikey apiCreateLoopingTimer (const std::string &callback, float delta)

### **Additional Inherited Members**

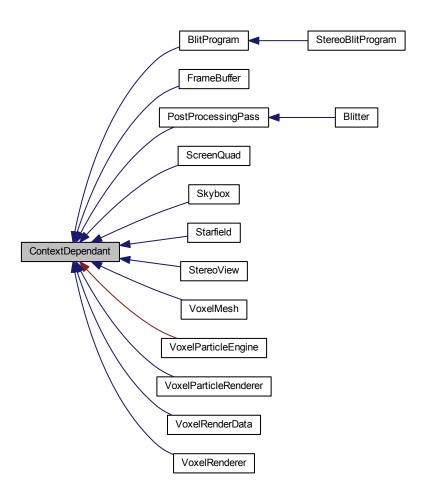
The documentation for this class was generated from the following files:

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

## 7.45 ContextDependant Class Reference

#include <contextdependant.h>

Inheritance diagram for ContextDependant:



### **Protected Member Functions**

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

### **Friends**

class ContextProvider

## 7.45.1 Detailed Description

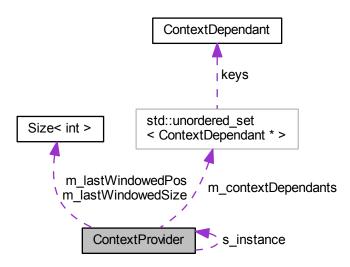
A ContextDependant is a class that has resources bound to the current GL context and must thus be notified before and after the context is rebuilt by the ContextProvider

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

### 7.46 ContextProvider Class Reference

#include <contextprovider.h>

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

## 7.46.1 Detailed Description

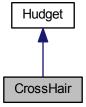
The ContextProvider manages the use of the GL context, especially its recreation when switching fullscreen and windowed mode

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

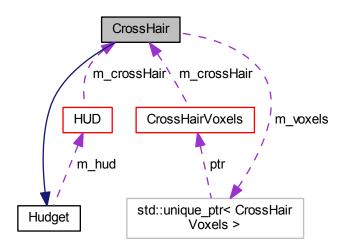
- · src/etc/contextprovider.h
- src/etc/contextprovider.cpp

## 7.47 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

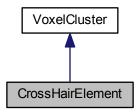
## **Protected Attributes**

- bool m\_actionActive
- $std::unique\_ptr < CrossHairVoxels > m\_voxels$

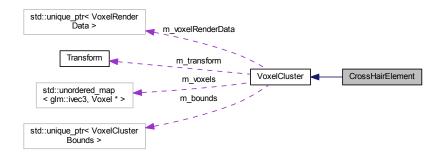
- src/ui/hud/crosshair.h
- src/ui/hud/crosshair.cpp

## 7.48 CrossHairElement Class Reference

Inheritance diagram for CrossHairElement:



Collaboration diagram for CrossHairElement:



## **Public Attributes**

- glm::quat relativeOrientation
- · float zOrientation

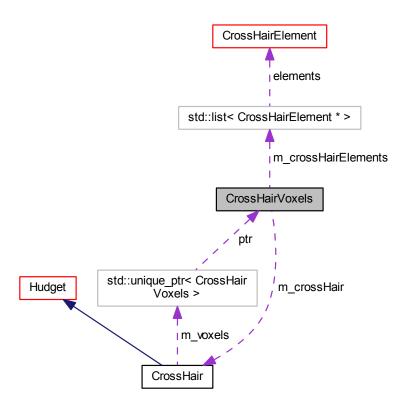
## **Additional Inherited Members**

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

• src/ui/hud/crosshairvoxels.cpp

## 7.49 CrossHairVoxels Class Reference

Collaboration diagram for CrossHairVoxels:



### **Public Member Functions**

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

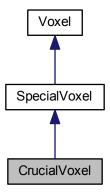
### **Protected Attributes**

- CrossHair \* m\_crossHair
- std::list< CrossHairElement \* > m\_crossHairElements

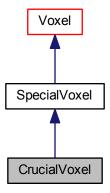
- src/ui/hud/crosshairvoxels.h
- src/ui/hud/crosshairvoxels.cpp

## 7.50 CrucialVoxel Class Reference

Inheritance diagram for CrucialVoxel:



Collaboration diagram for CrucialVoxel:



## **Public Member Functions**

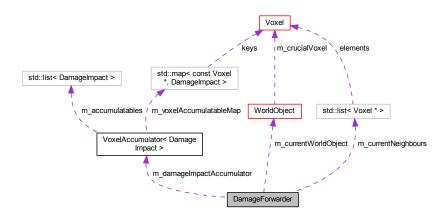
- CrucialVoxel (const glm::ivec3 &gridCell, int index)
- virtual void addToObject (WorldObject \*worldObject)
- virtual void onRemoval ()
- virtual void onDestruction ()

## **Additional Inherited Members**

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

## 7.51 DamageForwarder Class Reference

Collaboration diagram for DamageForwarder:



### **Public Member Functions**

- $\bullet \ \ \mathsf{void} \ \textbf{forwardDamageImpacts} \ (\mathsf{std} :: \mathsf{list} < \mathsf{DamageImpact} > \& \mathsf{dampedDeadlyDamageImpacts}) \\$
- void dontForwardTo (std::list< Voxel \* > &deadVoxels)
- std::list< DamageImpact > forwardedDamageImpacts ()

### **Protected Member Functions**

• float forwardFactor (float dotProduct, float fieldOfDamage, int neighbours)

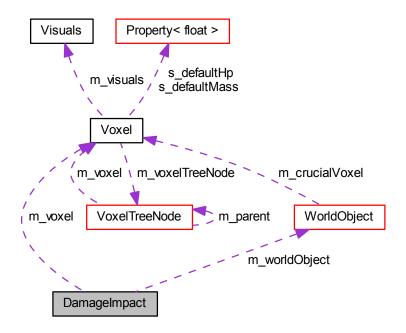
### **Protected Attributes**

- WorldObject \* m\_currentWorldObject
- std::list< Voxel \* > \* m\_currentNeighbours
- glm::ivec3 m\_currentGridCell
- VoxelAccumulator< DamageImpact > m\_damageImpactAccumulator

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

## 7.52 DamageImpact Class Reference

Collaboration diagram for DamageImpact:



### **Public Member Functions**

- DamageImpact (WorldObject \*worldObject, Voxel \*voxel, const glm::vec3 &damageVec, float fieldOf-Damage)
- 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)

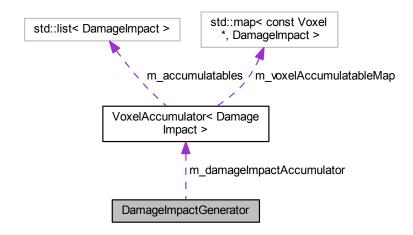
### **Protected Attributes**

- WorldObject \* m\_worldObject
- Voxel \* m\_voxel
- glm::vec3 m\_damageVec
- float m\_fieldOfDamage

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

## 7.53 DamageImpactGenerator Class Reference

Collaboration diagram for DamageImpactGenerator:



## **Public Member Functions**

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

## **Protected Attributes**

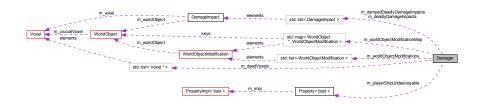
VoxelAccumulator< DamageImpact > m\_damageImpactAccumulator

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

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

## 7.54 Damager Class Reference

Collaboration diagram for Damager:



#### **Public Member Functions**

- void applyDamages (std::list< DamageImpact > &damageImpacts)
- · 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</li>
  - \*, WorldObjectModification > m\_worldObjectModificationMap
- · std::list
  - < WorldObjectModification > m\_worldObjectModifications
- Property < bool > m\_playerShipUndestroyable

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

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

## 7.55 DdsTexture Class Reference

#### Static Public Member Functions

- static bool loadlmage2d (glow::Texture \*texture, std::string path)
- static bool **loadImageCube** (glow::Texture \*texture, std::string pathXp, std::string pathXn, std::string pathYp, std::string pathYp, std::string pathZp, std::string pathZp, std::string pathZp

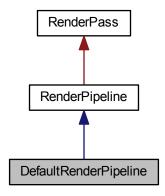
The documentation for this class was generated from the following files:

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

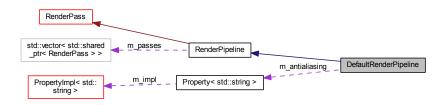
## 7.56 DefaultRenderPipeline Class Reference

#include <defaultrenderpipeline.h>

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
  - $< {\sf PostProcessingPass} > {\sf m\_fxaa}$
- std::shared\_ptr
  - $< {\sf PostProcessingPass} > {\sf m\_finalization}$
- $\bullet \ \, \text{Property}{<} \, \text{std::string} > \textbf{m\_antialiasing}$

### **Additional Inherited Members**

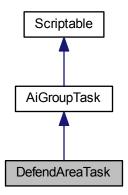
### 7.56.1 Detailed Description

a renderpipeline that applies basic effects to the frame like a bloom effect and creates the final image. The documentation for this class was generated from the following files:

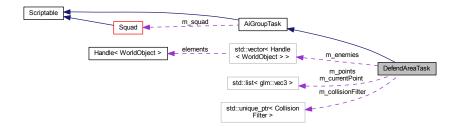
- src/display/rendering/defaultrenderpipeline.h
- src/display/rendering/defaultrenderpipeline.cpp

## 7.57 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</li>WorldObject >> m\_enemies
- float m\_defendRange

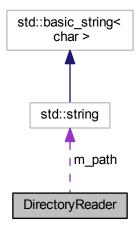
### **Additional Inherited Members**

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

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

## 7.58 DirectoryReader Class Reference

Collaboration diagram for DirectoryReader:



## **Public Member Functions**

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

### **Protected Attributes**

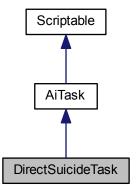
• std::string m\_path

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

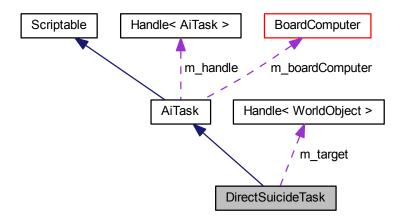
- src/utils/directoryreader.h
- src/utils/directoryreader.cpp

## 7.59 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**

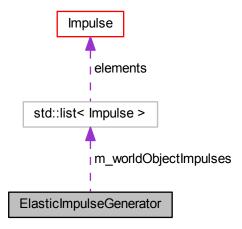
Handle < WorldObject > m\_target

## **Additional Inherited Members**

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

## 7.60 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)

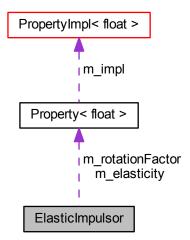
## **Protected Attributes**

std::list< Impulse > m\_worldObjectImpulses

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

## 7.61 ElasticImpulsor Class Reference

Collaboration diagram for ElasticImpulsor:



## **Public Member Functions**

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

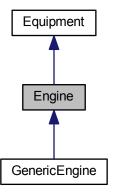
### **Protected Attributes**

- Property< float > m\_rotationFactor
- Property< float > m\_elasticity

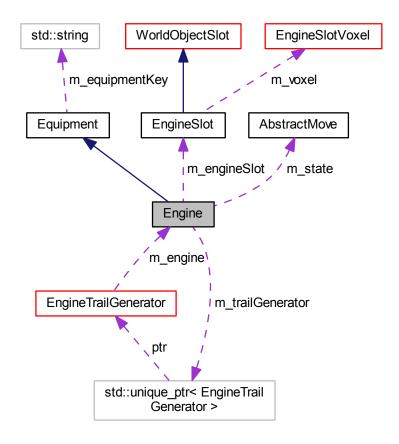
- src/world/handler/elasticimpulsor.h
- src/world/handler/elasticimpulsor.cpp

# 7.62 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**

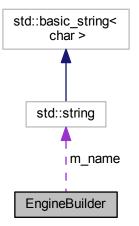
- std::unique\_ptr
   EngineTrailGenerator > m\_trailGenerator
- std::shared\_ptr< Sound > m\_sound
- EngineSlot \* m\_engineSlot
- EngineState m\_state

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

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

## 7.63 EngineBuilder Class Reference

Collaboration diagram for EngineBuilder:



## **Public Member Functions**

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

#### **Protected Attributes**

• std::string m\_name

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

## 7.64 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**

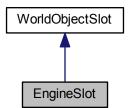
- glm::vec4 m\_directional
- glm::vec3 m\_angular

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

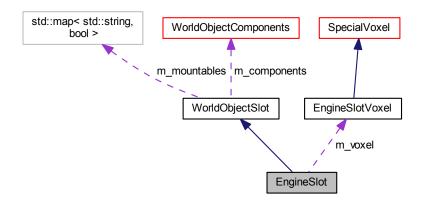
- src/equipment/enginepower.h
- src/equipment/enginepower.cpp

## 7.65 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 ()

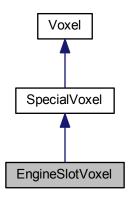
### **Protected Attributes**

- EngineSlotVoxel \* m\_voxel
- std::shared\_ptr< Engine > m\_engine
- glm::vec3 m\_direction

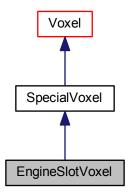
- src/equipment/engineslot.h
- src/equipment/engineslot.cpp

## 7.66 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

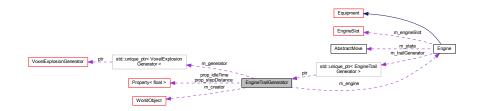
#### **Additional Inherited Members**

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

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

## 7.67 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)

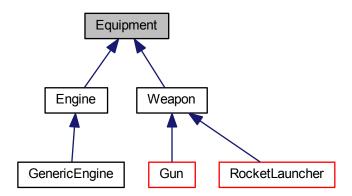
### **Protected Attributes**

- 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

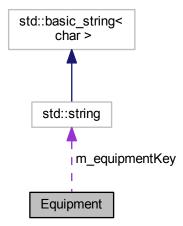
- · src/voxeleffect/enginetrailgenerator.h
- src/voxeleffect/enginetrailgenerator.cpp

## 7.68 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**

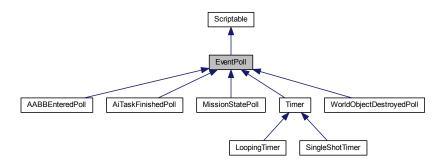
• std::string m\_equipmentKey

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

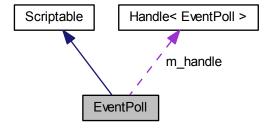
- · src/equipment/equipment.h
- src/equipment/equipment.cpp

## 7.69 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

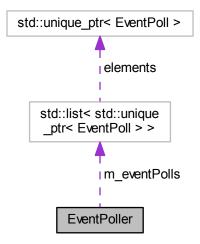
#### **Additional Inherited Members**

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

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

## 7.70 EventPoller Class Reference

Collaboration diagram for EventPoller:



#### **Public Member Functions**

- void addPoll (EventPoll \*eventPoll)
- void removePoll (EventPoll \*eventPoll)
- void **update** (float deltaSec)

### **Protected Attributes**

std::list< std::unique\_ptr</li>
 EventPoll >> m\_eventPolls

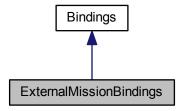
The documentation for this class was generated from the following files:

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

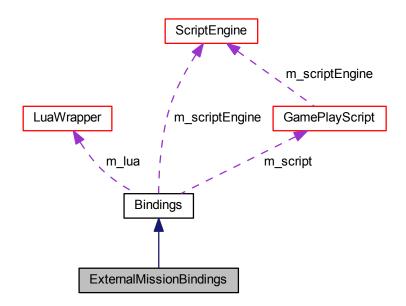
## 7.71 ExternalMissionBindings Class Reference

#include <externalmissionbindings.h>

Inheritance diagram for ExternalMissionBindings:



Collaboration diagram for ExternalMissionBindings:



**Public Member Functions** 

ExternalMissionBindings (GamePlayScript &script)

## **Protected Member Functions**

- · virtual void bind () override
- apikey apiMissionStart (const std::string &name)
- apikey apiOnMissionFailure (apikey missionKey, const std::string &callback)
- apikey apiOnMissionSuccess (apikey missionKey, const std::string &callback)
- apikey createStatePoll (apikey missionKey, MissionState state, const std::string &callback)

## **Additional Inherited Members**

## 7.71.1 Detailed Description

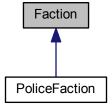
API to manage missions from the outside

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

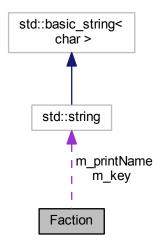
- src/scripting/bindings/externalmissionbindings.h
- src/scripting/bindings/externalmissionbindings.cpp

## 7.72 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

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

## 7.73 FactionMatrix Class Reference

Collaboration diagram for FactionMatrix:



### **Public Member Functions**

- Faction & pirateFaction ()
- Faction & policeFaction ()
- Faction & playerFaction ()
- Faction & unknownFaction ()
- Faction & getFaction (const std::string &factionName)
- void addFaction (std::shared\_ptr< Faction > faction)
- FactionRelation & getRelation (Faction &factionA, Faction &factionB)
- FactionRelation & getRelationToPlayer (Faction &faction)

#### **Protected Member Functions**

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

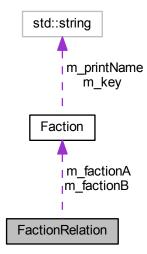
### **Protected Attributes**

```
    std::unordered_map
        < std::string, std::shared_ptr
        < Faction > > m_factions
    std::unordered_map< std::pair
        < Faction *, Faction * >
        , std::shared_ptr
        < FactionRelation > > m_relations
```

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

## 7.74 FactionRelation Class Reference

Collaboration diagram for FactionRelation:



### **Public Member Functions**

- FactionRelation (Faction &factionA, Faction &factionB, float friendliness)
- Faction & factionA ()
- Faction & factionB ()
- float friendliness () const
- · void setFriendliness (float friendliness)
- FactionRelationType type () const

# **Static Public Member Functions**

• static std::string typeName (FactionRelationType type)

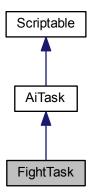
### **Protected Attributes**

- Faction & m\_factionA
- Faction & m\_factionB
- float m\_friendliness
- FactionRelationType m\_type

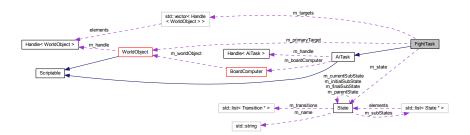
- · src/factions/factionrelation.h
- · src/factions/factionrelation.cpp

# 7.75 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</li>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

# 7.76 FileSystem Class Reference

```
#include <filesystem.h>
```

### **Static Public Member Functions**

- static bool exists (const std::string &path)
- static bool **copyFile** (const std::string &from, const std::string &to)
- static bool createDirectory (const std::string &path)
- static bool removeFile (const std::string &path)
- static bool removeDirectory (const std::string &path)
- static std::string userConfigDir ()

### 7.76.1 Detailed Description

A wrapper for basic filesystem. Can be replaced with QtCore or something like this if someone wants to do this;)

## 7.76.2 Member Function Documentation

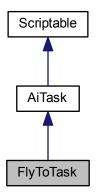
```
7.76.2.1 std::string FileSystem::userConfigDir() [static]
```

Returns the writable user config directory. Ensures that it exists

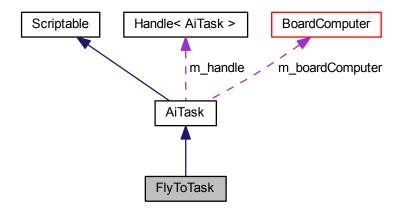
- · src/utils/filesystem.h
- src/utils/filesystem.cpp

# 7.77 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

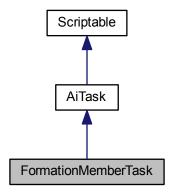
### **Additional Inherited Members**

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

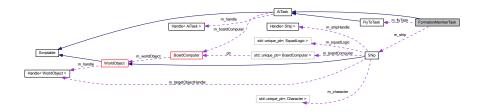
- · src/ai/basictasks/flytotask.h
- src/ai/basictasks/flytotask.cpp

# 7.78 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

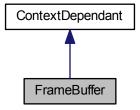
## **Additional Inherited Members**

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

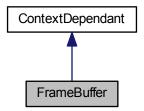
- · src/ai/basictasks/formationmembertask.h
- src/ai/basictasks/formationmembertask.cpp

## 7.79 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**

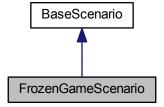
- int m\_colorAttachmentCount
- bool m\_useDepthAttachment
- glm::ivec2 m\_resolution
- glow::ref\_ptrglow::FrameBufferObject > m\_fbo

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

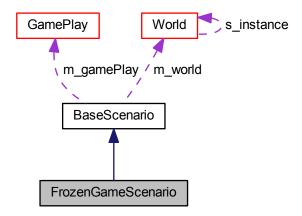
- src/display/rendering/framebuffer.h
- src/display/rendering/framebuffer.cpp

## 7.80 FrozenGameScenario Class Reference

Inheritance diagram for FrozenGameScenario:



Collaboration diagram for FrozenGameScenario:



## **Public Member Functions**

• FrozenGameScenario (GamePlay \*inGame)

### **Protected Member Functions**

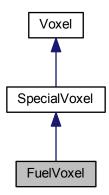
• virtual void populateWorld () override

## **Additional Inherited Members**

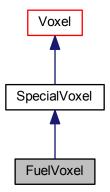
- src/scenarios/frozengamescenario.h
- src/scenarios/frozengamescenario.cpp

# 7.81 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

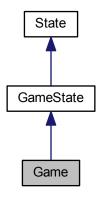
# **Additional Inherited Members**

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

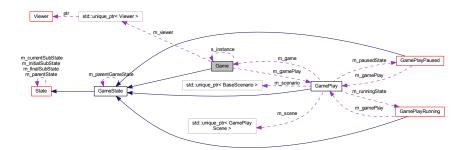
# 7.82 Game Class Reference

#include <game.h>

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 ()

### **Static Public Member Functions**

- static void setup ()
- static Game \* instance ()
- static void tearDown ()

### **Protected Attributes**

- std::shared\_ptr< HMDManager > m\_hmdManager
- $std::unique\_ptr < Viewer > m\_viewer$
- GamePlay \* m\_gamePlay

# **Static Protected Attributes**

static Game \* s\_instance

#### **Additional Inherited Members**

## 7.82.1 Detailed Description

Mainstate of the Game, entered once when libraries and context are setup and left just before they are teared down again

### 7.82.2 Member Function Documentation

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

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

Reimplemented from GameState.

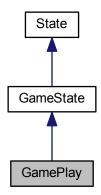
The documentation for this class was generated from the following files:

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

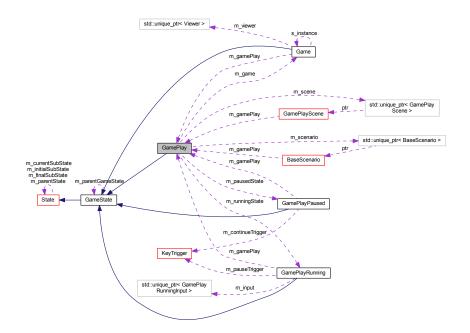
# 7.83 GamePlay Class Reference

```
#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

- 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< GamePlayScene > m\_scene
- std::unique\_ptr< BaseScenario > m\_scenario
- std::shared\_ptr< SoundManager > m\_soundManager
- GamePlayRunning \* m\_runningState
- GamePlayPaused \* m pausedState

### **Additional Inherited Members**

### 7.83.1 Detailed Description

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

### 7.83.2 Member Function Documentation

```
7.83.2.1 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.

```
7.83.2.2 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.

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

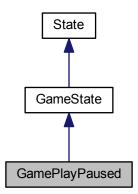
Performs a Transition from the currentSubState, if such isPossible()

Reimplemented from GameState.

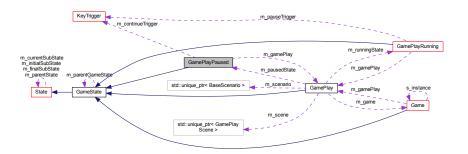
- · src/gamestate/gameplay/gameplay.h
- src/gamestate/gameplay/gameplay.cpp

# 7.84 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**

### 7.84.1 Member Function Documentation

```
7.84.1.1 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.

```
7.84.1.2 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.

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

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

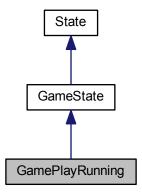
Reimplemented from GameState.

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

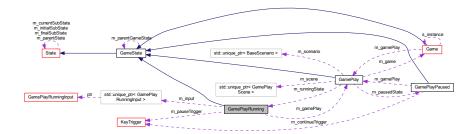
- · src/gamestate/gameplay/paused/gameplaypaused.h
- src/gamestate/gameplay/paused/gameplaypaused.cpp

# 7.85 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**

### 7.85.1 Member Function Documentation

```
7.85.1.1 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.

```
7.85.1.2 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.

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

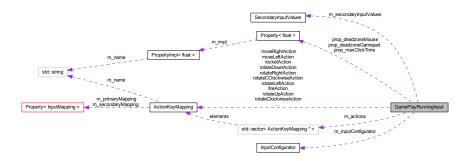
Performs a Transition from the currentSubState, if such isPossible()

Reimplemented from GameState.

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

# 7.86 GamePlayRunningInput Class Reference

Collaboration diagram for GamePlayRunningInput:



### **Public Member Functions**

- 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 addActionsToVector ()
- void setupJoystickControls ()
- void retrieveInputValues ()
- void placeCrossHair (double winX, double winY)

### **Protected Attributes**

- InputConfigurator \* m\_inputConfigurator
- SecondaryInputValues m\_secondaryInputValues
- std::vector< ActionKeyMapping \* > m\_actions
- · bool m centerCrosshair
- glm::vec2 m\_lastMousePos

- 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

#### 7.86.1 Member Function Documentation

7.86.1.1 void GamePlayRunningInput::update ( float deltaSec )

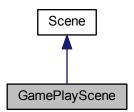
Check here for every-frame events, e.g. view & movement controls

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

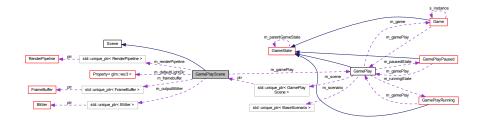
- src/gamestate/gameplay/running/gameplayrunninginput.h
- src/gamestate/gameplay/running/gameplayrunninginput.cpp

# 7.87 GamePlayScene Class Reference

Inheritance diagram for GamePlayScene:



### Collaboration diagram for GamePlayScene:



### **Public Member Functions**

- GamePlayScene (GamePlay &gamePlay)
- virtual void draw (const Camera &camera, glow::FrameBufferObject \*target, const Viewport &destination-Viewport, 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

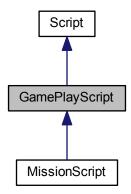
#### **Protected Attributes**

- $\bullet \ \, \text{std::unique\_ptr} < \, \text{Blitter} > \, \text{m\_outputBlitter}$
- 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
- Property < glm::vec3 > m\_defaultLightDir
- int m\_currentOutputBuffer

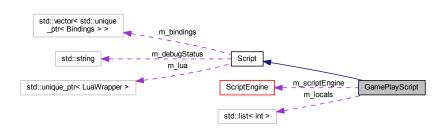
- · src/gamestate/gameplay/gameplayscene.h
- src/gamestate/gameplay/gameplayscene.cpp

# 7.88 GamePlayScript Class Reference

Inheritance diagram for GamePlayScript:



Collaboration diagram for GamePlayScript:



### **Public Member Functions**

- GamePlayScript (ScriptEngine \*scriptEngine)
- ScriptEngine & scriptEngine ()
- LuaWrapper & luaWrapper ()
- void addLocal (int key)

# **Protected Attributes**

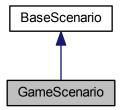
- ScriptEngine \* m\_scriptEngine
- $std::list < int > m_locals$

## **Additional Inherited Members**

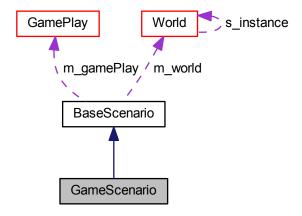
- src/scripting/gameplayscript.h
- src/scripting/gameplayscript.cpp

# 7.89 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 ()
- void spawnPoliceFleet ()

- void spawnPirateFleet ()
- void spawnStuff ()

### **Additional Inherited Members**

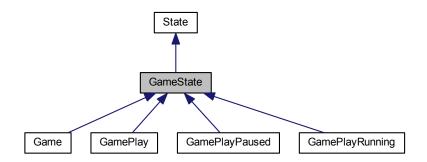
The documentation for this class was generated from the following files:

- src/scenarios/gamescenario.h
- src/scenarios/gamescenario.cpp

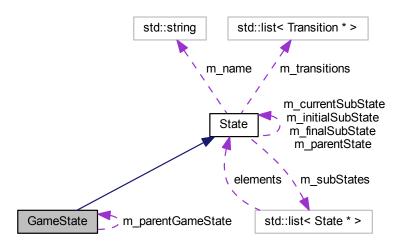
# 7.90 GameState Class Reference

#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**

### 7.90.1 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

#### 7.90.2 Member Function Documentation

```
7.90.2.1 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.

```
7.90.2.2 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.

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

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

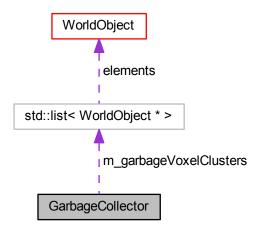
Reimplemented from State.

Reimplemented in GamePlay, Game, GamePlayRunning, and GamePlayPaused.

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

# 7.91 GarbageCollector Class Reference

Collaboration diagram for GarbageCollector:



### **Public Member Functions**

- void check (std::unordered\_set< WorldObject \* > &modifiedVoxelClusters)
- std::list< WorldObject \* > & garbageVoxelClusters ()

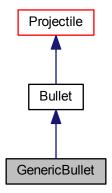
### **Protected Attributes**

std::list< WorldObject \* > m\_garbageVoxelClusters

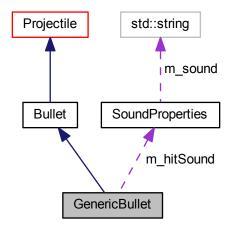
- src/world/handler/garbagecollector.h
- src/world/handler/garbagecollector.cpp

# 7.92 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

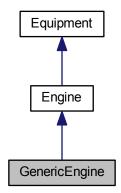
## **Additional Inherited Members**

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

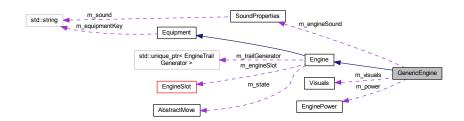
- src/equipment/weapons/genericbullet.h
- src/equipment/weapons/genericbullet.cpp

# 7.93 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

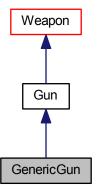
### **Additional Inherited Members**

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

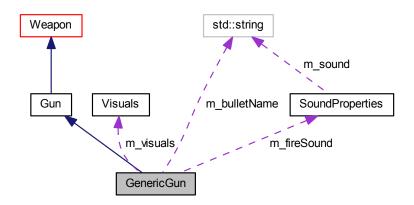
- src/equipment/engines/genericengine.h
- src/equipment/engines/genericengine.cpp

# 7.94 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

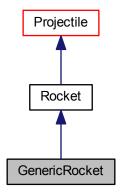
### **Protected Attributes**

- float m\_bulletSpeed
- · float m bulletLifetime
- float m\_cooldownTime
- Visuals m\_visuals
- SoundProperties m fireSound
- std::string m\_bulletName

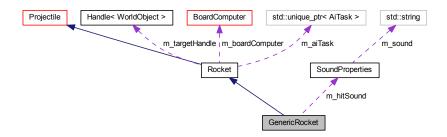
- · src/equipment/weapons/genericgun.h
- src/equipment/weapons/genericgun.cpp

# 7.95 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

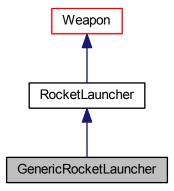
## **Additional Inherited Members**

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

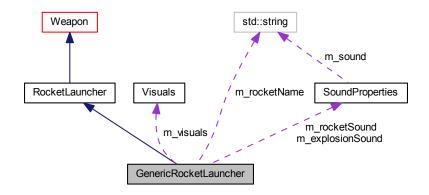
- · src/equipment/weapons/genericrocket.h
- src/equipment/weapons/genericrocket.cpp

# 7.96 GenericRocketLauncher Class Reference

Inheritance diagram for GenericRocketLauncher:



 $Collaboration\ diagram\ for\ Generic Rocket Launcher:$ 



### **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**

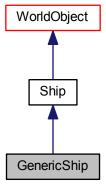
- float m\_cooldownTime
- SoundProperties m\_rocketSound
- SoundProperties m\_explosionSound
- · Visuals m\_visuals
- std::string m\_rocketName

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

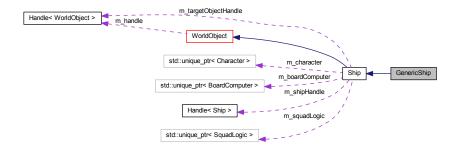
- src/equipment/weapons/genericrocketlauncher.h
- src/equipment/weapons/genericrocketlauncher.cpp

# 7.97 GenericShip Class Reference

Inheritance diagram for GenericShip:



Collaboration diagram for GenericShip:



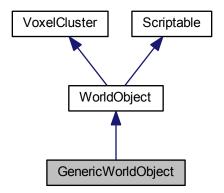
### **Additional Inherited Members**

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

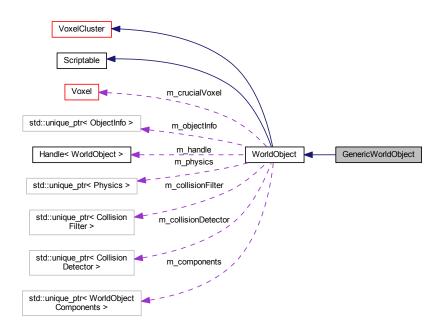
- · src/worldobject/genericship.h
- · src/worldobject/genericship.cpp

# 7.98 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

# 7.99 GeometryHelper Class Reference

### **Static Public Member Functions**

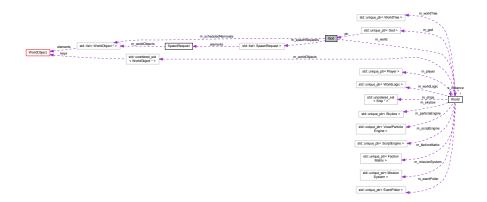
- static bool **intersectRectangle** (const Ray \*ray, const glm::vec3 &p, const glm::vec3 &q, const glm::vec3 &r, const glm::vec3 &s)
- static bool **intersectRectangle** (const Ray \*ray, const glm::vec3 &p, const glm::vec3 &q, const glm::vec3 &r, const glm::vec3 &s, glm::vec3 &intersection)
- static 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)

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

### 7.100 God Class Reference

#include <god.h>

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
- $\bullet \ \, \text{std::list}{<} \, \, \text{WorldObject} * > \text{m\_scheduledRemovals} \\$

## 7.100.1 Detailed Description

Responsible for safely inserting object into the World, preventing overlappings

- to spawn an object, first call scheduleSpawn() and then spawn()
- to remove an object, first call scheduleRemoval and then remove()

The World will be given full ownership of spawned object, rejected objects however will have to be managed by the caller.

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

## 7.101 GridAABB Class Reference

### **Public Member Functions**

- GridAABB (const glm::ivec3 &llf, const glm::ivec3 &urb)
- const glm::ivec3 & IIf () const
- void setLLF (const glm::ivec3 &llf)
- const glm::ivec3 & urb () const
- void setURB (const glm::ivec3 &urb)
- bool contains (const glm::ivec3 &cell) const
- int extent (Axis axis) const
- float diameter () const
- bool operator== (const GridAABB &other) const

### **Protected Attributes**

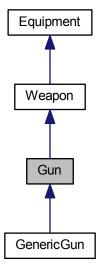
- glm::ivec3 m\_llf
- glm::ivec3 m\_urb

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

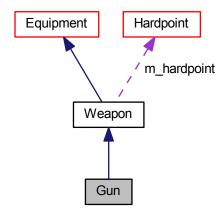
- · src/geometry/gridaabb.h
- · src/geometry/gridaabb.cpp

## 7.102 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

# 7.103 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

# 7.104 HandleImpl < T > Class Template Reference

### **Public Member Functions**

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

#### **Protected Attributes**

• T \* m\_object

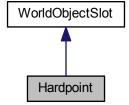
The documentation for this class was generated from the following files:

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

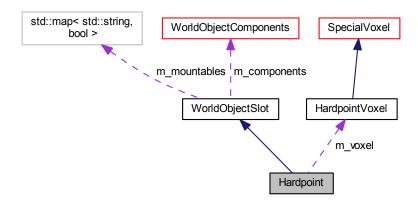
# 7.105 Hardpoint Class Reference

#include <hardpoint.h>

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

### 7.105.1 Detailed Description

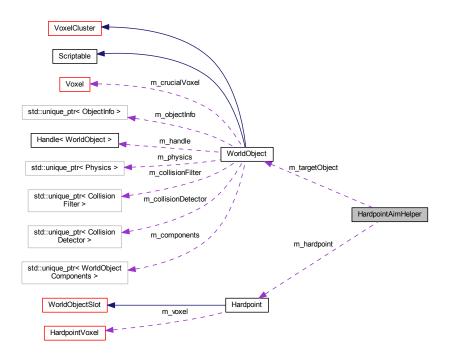
Slot into which interchangeable weapons may be mounted direction and fieldOfAim form a cone in which the hardpoint can fire its weapons

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

# 7.106 HardpointAimHelper Class Reference

#include <hardpointaimhelper.h>

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

# 7.106.1 Detailed Description

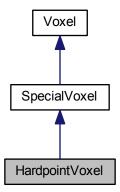
Identifies the direction a hardpoint must shoot to hit a moving WorldObject given the direction and the speed of this other WorldObject doesn't change too much

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

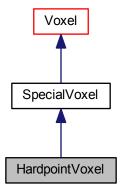
- src/worldobject/helper/hardpointaimhelper.h
- src/worldobject/helper/hardpointaimhelper.cpp

# 7.107 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

# 7.108 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

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

### **Public Member Functions**

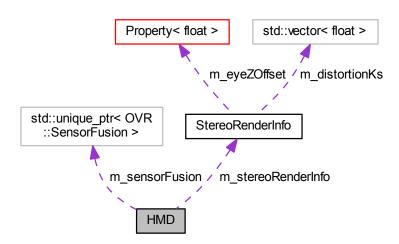
size\_t operator() (const pair< A, B > &arg) const

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

· src/utils/pairhash.h

## 7.110 HMD Class Reference

Collaboration diagram for HMD:



### **Public Member Functions**

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

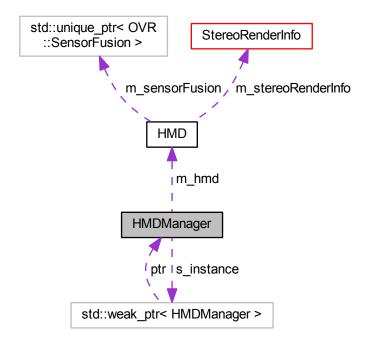
#### **Protected Attributes**

- std::unique\_ptr
  - < OVR::SensorFusion > m\_sensorFusion
- OVR::HMDDevice \* m\_hmdDevice
- OVR::SensorDevice \* m\_sensorDevice
- StereoRenderInfo m\_stereoRenderInfo

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

# 7.111 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

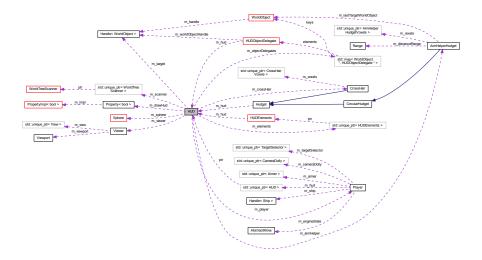
# **Static Protected Attributes**

static std::weak\_ptr< HMDManager > s\_instance

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

## 7.112 HUD Class Reference

### Collaboration diagram for HUD:



#### **Public Member Functions**

- HUD (Player \*player)
- Player \* player ()
- const Sphere & sphere () const
- CrossHair & crossHair ()
- AimHelperHudget & aimHelper ()
- glm::vec3 centerOfView () const
- glm::vec3 position () const
- · glm::quat orientation () const
- void addHudget (Hudget \*hudget)
- void removeHudget (Hudget \*hudget)
- void addObjectDelegate (HUDObjectDelegate \*objectDelegate)
- void removeObjectDelegate (HUDObjectDelegate \*objectDelegate)
- HUDObjectDelegate \* 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
- float fovy () const
- · float fovx () const
- void showMissionInfo (const std::string &title, const std::string &caption)
- void **showMissionMessage** (const std::string &message)
- void showMessage (const std::string &message)

## **Protected Member Functions**

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

#### **Protected Attributes**

- Player \* m\_player
- Viewer \* m\_viewer
- Sphere m sphere
- Handle < WorldObject > m\_target
- Property< bool > m\_drawHud
- float m\_fovy
- float m fovx
- CrossHair \* m\_crossHair
- AimHelperHudget \* m\_aimHelper
- std::unique\_ptr< WorldTreeScanner > m\_scanner
- std::unique ptr< HUDElements > m\_elements
- std::map< WorldObject</li>
  - \*, HUDObjectDelegate \* > m\_objectDelegates

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

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

### 7.113 HUDElements Class Reference

Collaboration diagram for HUDElements:



### **Public Member Functions**

- HUDElements (HUD &hud)
- HUD & hud ()
- void addHudget (Hudget \*hudget)
- void removeHudget (Hudget \*hudget)
- std::list< std::unique\_ptr</li>
  - < Hudget > > & hudgets ()
- void addAnimation (HudgetAnimation \*animation)
- void setTargetName (const std::string &name)
- void setSpeed (const std::string &speed)
- void **showMissionInfo** (const std::string &title, const std::string &caption)
- void **showMissionMessage** (const std::string &message)
- void showMessage (const std::string &message)
- void **update** (float deltaSec)
- · void draw ()

## **Protected Attributes**

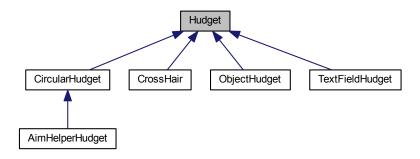
- HUD & m\_hud
- std::list< std::unique\_ptr</li>Hudget >> m\_hudgets
- std::list< std::unique\_ptr</li>HudgetAnimation > > m\_animations
- TextFieldHudget \* m\_speedLabel
- TextFieldHudget \* m\_targetName
- TextFieldHudget \* m\_missionTitle
- HudgetHideAnimation \* m\_missionTitleHider
- TextFieldHudget \* m\_missionCaption
- HudgetHideAnimation \* m\_missionCaptionHider
- TextFieldHudget \* m\_missionMessage
- HudgetHideAnimation \* m\_missionMessageHider

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

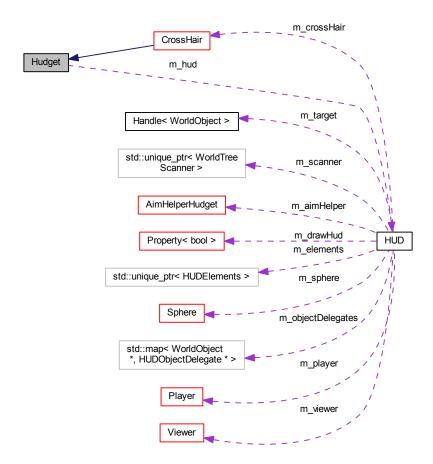
- src/ui/hud/hudelements.h
- src/ui/hud/hudelements.cpp

# 7.114 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**

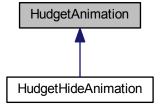
- HUD \* m\_hud
- glm::vec3 m\_direction
- float m\_directionAngle
- float m\_relativeDistance
- bool m\_visible
- · bool m\_pressed
- bool m\_hovered
- bool m\_clicked
- · bool m released
- · bool m\_entered
- bool m\_left

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

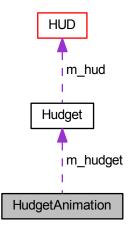
- src/ui/hud/hudget.h
- src/ui/hud/hudget.cpp

# 7.115 HudgetAnimation Class Reference

Inheritance diagram for HudgetAnimation:



Collaboration diagram for HudgetAnimation:



### **Public Member Functions**

- HudgetAnimation (Hudget &hudget)
- Hudget & hudget ()
- · bool active () const
- void setActive (bool active)
- virtual void **update** (float deltaSec)

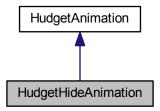
### **Protected Attributes**

- Hudget & m\_hudget
- bool m\_active

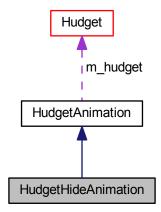
- src/ui/hud/hudgetanimation.h
- src/ui/hud/hudgetanimation.cpp

# 7.116 HudgetHideAnimation Class Reference

Inheritance diagram for HudgetHideAnimation:



Collaboration diagram for HudgetHideAnimation:



## **Public Member Functions**

- HudgetHideAnimation (Hudget &hudget)
- void hideln (float deltaSec)
- void update (float deltaSec)

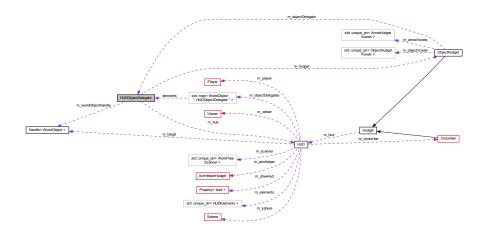
## **Protected Attributes**

float m\_countdown

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

# 7.117 HUDObjectDelegate Class Reference

Collaboration diagram for HUDObjectDelegate:



### **Public Member Functions**

- HUDObjectDelegate (HUD \*hud, WorldObject \*worldObject, ObjectHudget \*hudget)
- HUD \* hud ()
- WorldObject \* worldObject ()
- ObjectHudget \* hudget ()

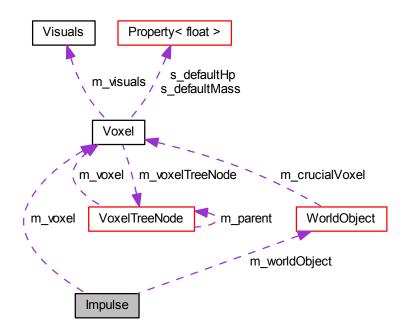
## **Protected Attributes**

- HUD \* m\_hud
- Handle < WorldObject > m\_worldObjectHandle
- ObjectHudget \* m\_hudget

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

# 7.118 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)

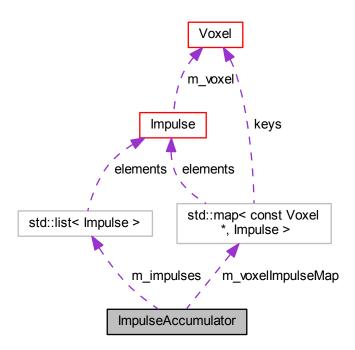
### **Protected Attributes**

- WorldObject \* m\_worldObject
- Voxel \* m\_voxel
- glm::vec3 m\_speed
- float m\_mass
- glm::vec3 m\_normal

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

# 7.119 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)

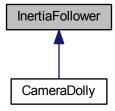
### **Protected Attributes**

- std::map< const Voxel \*, Impulse > m\_voxelImpulseMap
- std::list< Impulse > m\_impulses

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

### 7.120 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

# 7.121 InputConfigurator Class Reference

### **Public Member Functions**

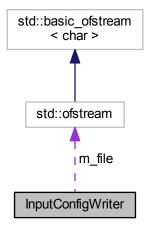
- InputConfigurator (std::vector< ActionKeyMapping \* > \*actions, SecondaryInputValues \*secondaryInputValues, Property< float > \*deadzone, HUD \*hud)
- void startConfiguration (InputClass inputClass)
- bool isConfiguring ()
- · void update ()
- void setActions (std::vector< ActionKeyMapping \* > \*actions)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

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

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

# 7.122 InputConfigWriter Class Reference

Collaboration diagram for InputConfigWriter:



### **Public Member Functions**

- InputConfigWriter (const std::string &file)
- void write (ActionKeyMapping &mapping)

### **Protected Member Functions**

• void write (const std::string &name, const InputMapping &mapping)

### **Protected Attributes**

• std::ofstream m\_file

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

- · src/input/inputconfigwriter.h
- src/input/inputconfigwriter.cpp

# 7.123 InputMapping Class Reference

## **Public Member Functions**

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

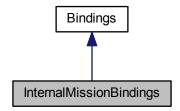
The documentation for this class was generated from the following files:

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

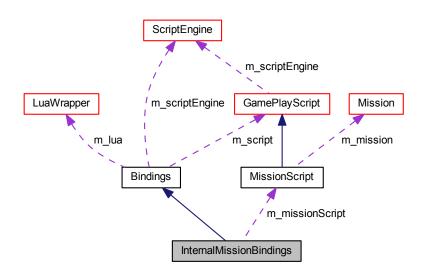
# 7.124 Internal Mission Bindings Class Reference

#include <internalmissionbindings.h>

Inheritance diagram for InternalMissionBindings:



Collaboration diagram for InternalMissionBindings:



# **Public Member Functions**

• InternalMissionBindings (MissionScript &script)

### **Protected Member Functions**

- virtual void bind () override
- int apiMissionSucceed ()
- int apiMissionFail ()
- int apiMissionMessage (const std::string &message)
- int apiMissionFailureMessage (const std::string &message)
- int apiMissionSuccessMessage (const std::string &message)

#### **Protected Attributes**

• MissionScript & m\_missionScript

### 7.124.1 Detailed Description

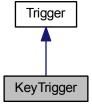
API available to MissionScript

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

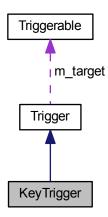
- src/scripting/bindings/internalmissionbindings.h
- src/scripting/bindings/internalmissionbindings.cpp

# 7.125 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

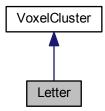
### **Protected Attributes**

- int m\_glfwKey
- int m\_lastState

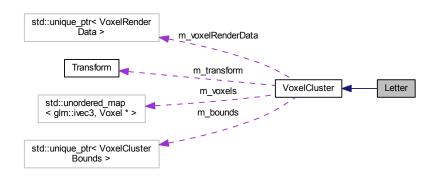
- src/utils/statemachine/keytrigger.h
- src/utils/statemachine/keytrigger.cpp

# 7.126 Letter Class Reference

Inheritance diagram for Letter:



Collaboration diagram for Letter:



## **Public Member Functions**

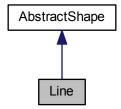
• Letter (float scale=1.0)

## **Additional Inherited Members**

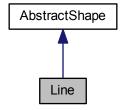
- · src/ui/letter.h
- src/ui/letter.cpp

## 7.127 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

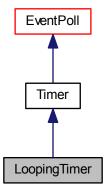
## **Protected Attributes**

- glm::vec3 **m\_a**
- glm::vec3 m\_b

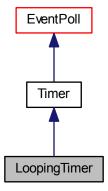
- · src/geometry/line.h
- · src/geometry/line.cpp

# 7.128 LoopingTimer Class Reference

Inheritance diagram for LoopingTimer:



Collaboration diagram for LoopingTimer:



## **Public Member Functions**

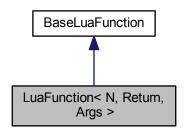
• LoopingTimer (float interval, const std::function< void()> &callback)

## **Additional Inherited Members**

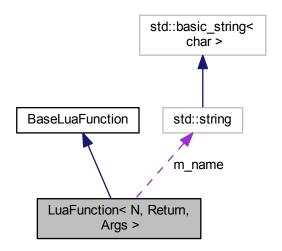
- src/events/loopingtimer.h
- · src/events/loopingtimer.cpp

# 7.129 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::pushcclosure (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

#### 7.129.1 Member Data Documentation

7.129.1.1 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

# 7.130 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 ()
- bool has (const std::string &fun)
- 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

7.131 Math Class Reference 161

## **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

# 7.131 Math Class Reference

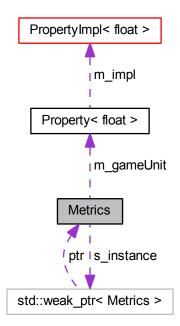
### **Static Public Member Functions**

• static uint32\_t nextPowerOf2 (uint32\_t n)

- · src/utils/math.h
- · src/utils/math.cpp

# 7.132 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<  ${\sf Metrics}>{\sf instance}$  ()

### **Protected Attributes**

Property< float > m\_gameUnit

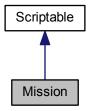
## **Static Protected Attributes**

static std::weak\_ptr< Metrics > s\_instance

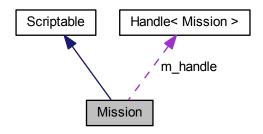
- · src/utils/metrics.h
- · src/utils/metrics.cpp

# 7.133 Mission Class Reference

Inheritance diagram for Mission:



Collaboration diagram for Mission:



## **Public Member Functions**

- Mission (const std::string &path)
- void start ()
- MissionState state () const
- void succeed ()
- void fail ()
- Handle < Mission > & handle ()
- void update (float deltaSec)

## **Protected Attributes**

- std::shared\_ptr< MissionScript > m\_script
- Handle < Mission > m\_handle
- MissionState m\_state

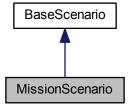
## **Additional Inherited Members**

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

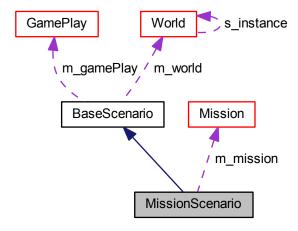
- · src/missions/mission.h
- src/missions/mission.cpp

# 7.134 MissionScenario Class Reference

Inheritance diagram for MissionScenario:



Collaboration diagram for MissionScenario:



# **Public Member Functions**

• MissionScenario (GamePlay \*gamePlay, const std::string &path)

## **Protected Member Functions**

• virtual void populateWorld () override

## **Protected Attributes**

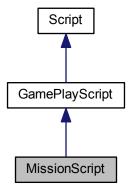
• Mission \* m\_mission

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

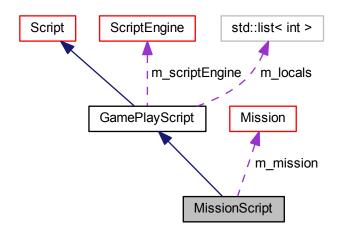
- · src/scenarios/missionscenario.h
- src/scenarios/missionscenario.cpp

# 7.135 MissionScript Class Reference

Inheritance diagram for MissionScript:



Collaboration diagram for MissionScript:



### **Public Member Functions**

- MissionScript (Mission &mission, ScriptEngine \*scriptEngine)
- Mission & mission ()
- · void onSuccess ()
- void onFailure ()

## **Protected Attributes**

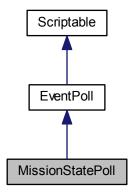
• Mission & m\_mission

### **Additional Inherited Members**

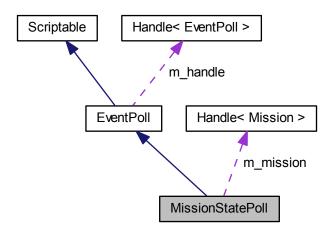
- src/scripting/missionscript.h
- src/scripting/missionscript.cpp

# 7.136 MissionStatePoll Class Reference

Inheritance diagram for MissionStatePoll:



Collaboration diagram for MissionStatePoll:



## **Public Member Functions**

- MissionStatePoll (const Handle < Mission > &mission, MissionState state, const std::function < void() > &callback)
- virtual bool isDead () override

## **Protected Member Functions**

• virtual bool poll () override

• virtual void specialOnCallback () override

#### **Protected Attributes**

- Handle < Mission > m\_mission
- MissionState m\_state
- · bool m\_dead

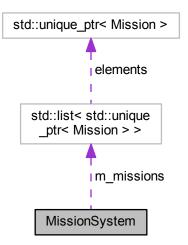
### **Additional Inherited Members**

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

- · src/events/missionstatepoll.h
- src/events/missionstatepoll.cpp

# 7.137 MissionSystem Class Reference

Collaboration diagram for MissionSystem:



## **Public Member Functions**

- void update (float deltaSec)
- void addMission (Mission \*mission)
- void removeMission (Mission \*mission)

### **Protected Attributes**

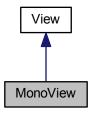
std::list< std::unique\_ptr</li>Mission >> m\_missions

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

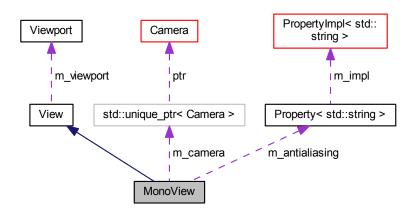
- · src/missions/missionsystem.h
- · src/missions/missionsystem.cpp

# 7.138 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**

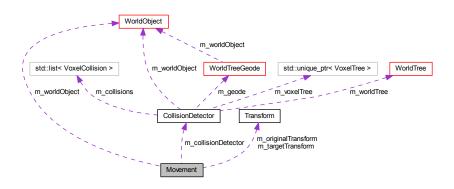
- std::unique\_ptr< Camera > m\_camera
- Property < std::string > m\_antialiasing

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

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

## 7.139 Movement Class Reference

Collaboration diagram for Movement:



### **Public Member Functions**

- Movement (WorldObject &worldObject, const Transform &originalTransform, const Transform &target-Transform)
- bool perform ()

## **Protected Member Functions**

- bool performSplitted ()
- bool performStepped ()
- int calculateStepCount ()
- Transform calculateStep (int s, int stepCount) const

### **Protected Attributes**

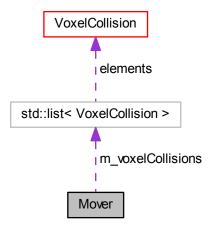
- WorldObject & m\_worldObject
- CollisionDetector & m\_collisionDetector
- Transform m\_originalTransform
- Transform m\_targetTransform
- · float m distance

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

7.140 Mover Class Reference 171

# 7.140 Mover Class Reference

Collaboration diagram for Mover:



### **Public Member Functions**

- void moveWorldObjects (float deltaSec)
- std::list< VoxelCollision > & voxelCollisions ()

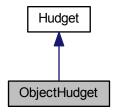
### **Protected Attributes**

•  $std::list < VoxelCollision > m_voxelCollisions$ 

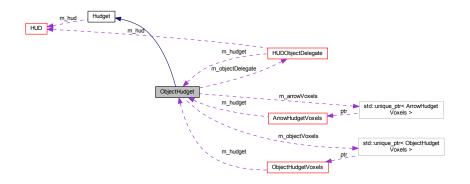
- src/world/handler/mover.h
- src/world/handler/mover.cpp

# 7.141 ObjectHudget Class Reference

Inheritance diagram for ObjectHudget:



Collaboration diagram for ObjectHudget:



### **Public Member Functions**

- ObjectHudget (HUD \*hud)
- 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 ()
- void setObjectDelegate (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

# 7.142 ObjectHudgetCornerVoxels Class Reference

Collaboration diagram for ObjectHudgetCornerVoxels:



### **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)

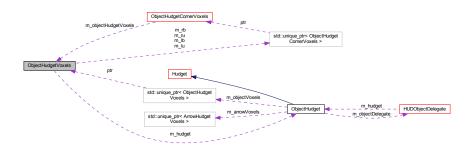
### **Protected Attributes**

- ObjectHudgetVoxels \* m\_objectHudgetVoxels
- glm::ivec3 m baseOffset
- glm::vec3 m\_position
- glm::quat m\_orientation
- std::vector< std::unique\_ptr
  - < VoxelCluster > > m\_voxelClusters

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

# 7.143 ObjectHudgetVoxels Class Reference

Collaboration diagram for ObjectHudgetVoxels:



#### **Public Member Functions**

- ObjectHudgetVoxels (ObjectHudget \*hudget)
- ObjectHudget \* hudget ()
- void setTargetHightlight (bool targetHightlight)
- void **setRelationType** (FactionRelationType relationType)
- float openingAngle () const
- void **setOpeningAngle** (float openingAngle)
- · void update (float deltaSec)
- void draw ()
- · bool isAt (const Ray &ray) const

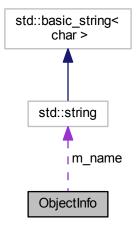
### **Protected Attributes**

- 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\_targetHightlight
- FactionRelationType m\_relationType

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

# 7.144 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)

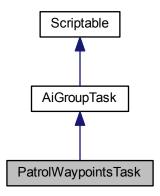
# **Protected Attributes**

- std::string m\_name
- bool m\_showOnHud
- bool m\_canLockOn

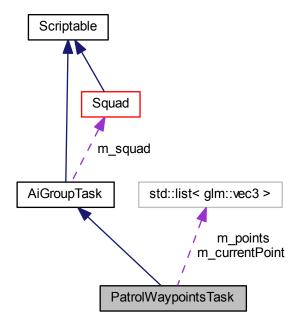
- · src/ui/objectinfo.h
- · src/ui/objectinfo.cpp

# 7.145 PatrolWaypointsTask Class Reference

Inheritance diagram for PatrolWaypointsTask:



Collaboration diagram for PatrolWaypointsTask:



# **Public Member Functions**

- PatrolWaypointsTask (Squad &squad)
- $\bullet \ \, \textbf{PatrolWaypointsTask} \ ( \textbf{Squad} \ \& \textbf{squad}, \ \textbf{const} \ \textbf{std} :: \textbf{list} < \textbf{glm} :: \textbf{vec3} > \& \textbf{points} ) \\$

- 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

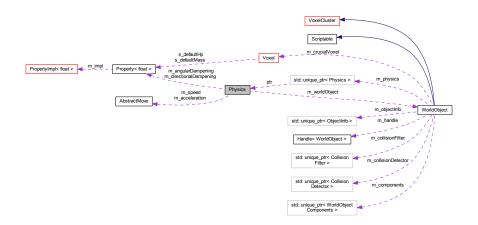
#### **Additional Inherited Members**

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

- · src/ai/grouptasks/patrolwaypointstask.h
- src/ai/grouptasks/patrolwaypointstask.cpp

# 7.146 Physics Class Reference

Collaboration diagram for Physics:



#### **Public Member Functions**

- Physics (WorldObject &worldObject, float scale)
- float directionalDampening () const
- void **setDirectionalDampening** (const Property< float > &directionalDampening)
- float angularDampening () const
- void setAngularDampening (const Property< float > &angularDampening)
- const Speed & speed () const
- void setSpeed (const Speed &speed)

- · const Acceleration & acceleration () const
- · void setAcceleration (const Acceleration &acceleration)
- · float mass () const
- · float maxMass () 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**

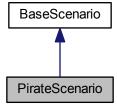
- WorldObject & m\_worldObject
- Speed m\_speed
- Acceleration m\_acceleration
- Property< float > m\_directionalDampening
- Property< float > m\_angularDampening
- float m\_mass
- float m\_maxMass
- glm::vec3 m\_accumulatedMassVec
- float m\_massScaleFactor

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

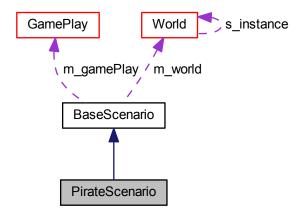
- src/physics/physics.h
- · src/physics/physics.cpp

# 7.147 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 ()

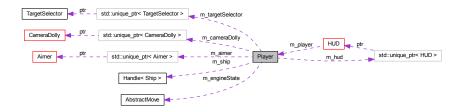
### **Additional Inherited Members**

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

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

# 7.148 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**

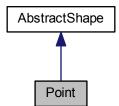
- Handle < Ship > m\_ship
- std::unique\_ptr< CameraDolly > m\_cameraDolly
- std::unique\_ptr< HUD > m\_hud
- $std::unique\_ptr < TargetSelector > m\_targetSelector$
- $std::unique\_ptr < Aimer > m\_aimer$
- EngineState m\_engineState

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

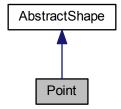
- · src/player.h
- · src/player.cpp

# 7.149 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**

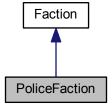
• glm::vec3 m\_position

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

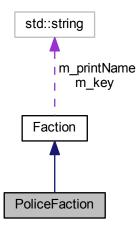
- src/geometry/point.h
- src/geometry/point.cpp

# 7.150 PoliceFaction Class Reference

Inheritance diagram for PoliceFaction:



Collaboration diagram for PoliceFaction:



### **Public Member Functions**

• PoliceFaction (FactionMatrix \*factionMatrix)

## **Additional Inherited Members**

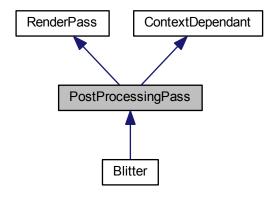
The documentation for this class was generated from the following file:

· src/factions/policefaction.h

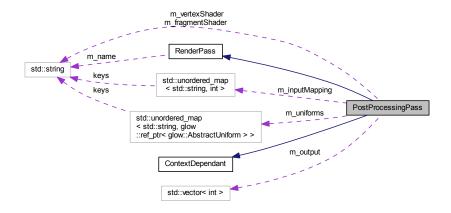
# 7.151 PostProcessingPass Class Reference

#include <postprocessingpass.h>

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\_mapstd::string, int > m\_inputMapping
- std::vector< int > m\_output
- std::string m\_fragmentShader
- std::string m\_vertexShader
- bool m\_enabled

## 7.151.1 Detailed Description

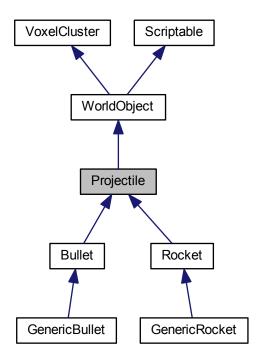
A configurable RenderPass for a shader that reads and writes on a framebuffer and has no further gamelogic The documentation for this class was generated from the following files:

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

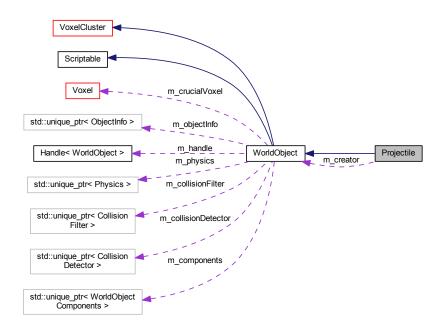
# 7.152 Projectile Class Reference

#include jectile.h>

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**

### 7.152.1 Detailed Description

Base class for everything shot by a weapon, characterised by

- · doesn't collide with creator
- · destroyed after lifetime is over

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

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

# 7.153 Property < T > Class Template Reference

```
#include property.h>
```

#### **Public Member Functions**

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

### **Static Public Member Functions**

- static T get (const std::string &name)
- static T get (const std::string &name, const T &defaultvalue)
- static Property < T > unnamed (const T &value)

### **Protected Attributes**

• PropertyImpl< T > \* m\_impl

## 7.153.1 Detailed Description

template < class T > class Property < T >

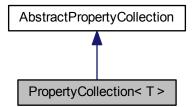
represents a value loaded from an .ini file

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

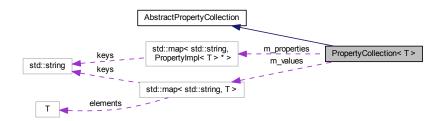
- src/property/property.h
- src/property/property.inl

# 7.154 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)
- PropertyImpl< T > \* getImpl (const std::string &key)
- PropertyImpl < T > \* getImpl (const std::string &key, const T &defaultValue)
- 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

#### **Protected Member Functions**

- PropertyImpl< T > \* getOrCreate (const std::string &key)
- PropertyImpl< T > \* create (const std::string &key)

#### **Protected Attributes**

- std::map< std::string, T > m\_values
- std::map< std::string,</li>
  - PropertyImpl< T > \* > m\_properties
- regexns::regex m\_regex
- std::function< T(const</li>

std::string &)> m\_converter

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

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

# 7.155 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)

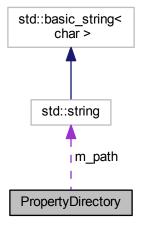
The documentation for this class was generated from the following files:

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

# 7.156 Property Directory Class Reference

#include propertydirectory.h>

Collaboration diagram for PropertyDirectory:



## **Public Member Functions**

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

# **Protected Attributes**

• std::string m\_path

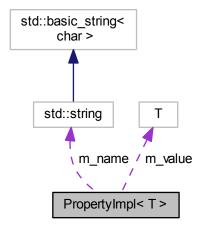
## 7.156.1 Detailed Description

Reads a whole directory of files as property-files Prefixes the data from a file with the files basename, e.g. a property "foo.name" in a file "/some/where/bar.ini" will be stored in "bar.foo.name"

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

# 7.157 PropertyImpl < T > Class Template Reference

Collaboration diagram for PropertyImpl < T >:



#### **Public Member Functions**

- PropertyImpl (const std::string &name)
- PropertyImpI (const std::string &name, const T &value)
- · const std::string & name () const
- · const T & get () const
- void set (const T &value)

### **Protected Attributes**

- std::string m\_name
- T m\_value

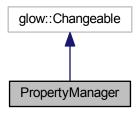
The documentation for this class was generated from the following files:

- src/property/property.h
- src/property/propertyimpl.h
- · src/property/propertyimpl.inl

# 7.158 PropertyManager Class Reference

#include propertymanager.h>

Inheritance diagram for PropertyManager:



### Collaboration diagram for PropertyManager:



### **Public Member Functions**

- void load (const std::string &file, const std::string &prefix="")
- template<typename T >
   PropertyImpl< T > \* getImpl (const std::string &key)
- template<typename T >
   PropertyImpl< T > \* getImpl (const std::string &key, const T &defaultValue)
- 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

## 7.158.1 Detailed Description

Keeps track of properties and loads ini files. Properties will be updated when a new ini file is loaded. Implements glow::Changeable, so glow::ChangeListener can be notified about changes.

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

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

### 7.159 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

### 7.160 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

# 7.161 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

# 7.162 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**

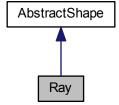
- float m\_min
- float m\_max

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

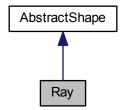
- src/geometry/range.h
- src/geometry/range.cpp

# 7.163 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

#### 7.164 RenderMetaData Class Reference

### **Public Member Functions**

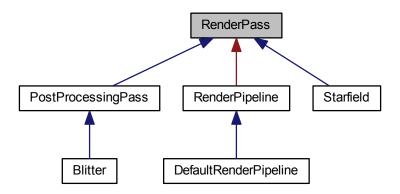
- RenderMetaData (const Camera &camera, EyeSide eyeside)
- const Camera & camera () const
- EyeSide eyeside () const

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

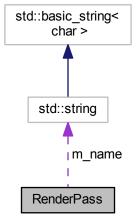
# 7.165 RenderPass Class Reference

#include <renderpass.h>

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

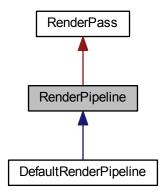
# 7.165.1 Detailed Description

A step that adds something to the current frame can be postprocessing, geometry or any other effect The documentation for this class was generated from the following files:

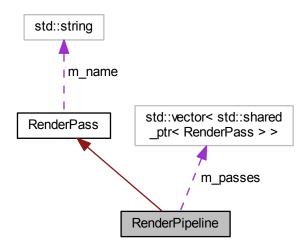
- src/display/rendering/renderpass.h
- src/display/rendering/renderpass.cpp

# 7.166 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</li>RenderPass >> m\_passes
- · bool m\_initialized

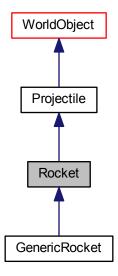
The documentation for this class was generated from the following files:

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

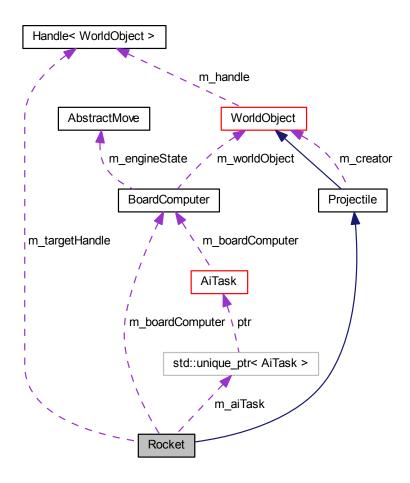
## 7.167 Rocket Class Reference

#include <rocket.h>

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**

# 7.167.1 Detailed Description

Base class for Projectiles that follow a target and attempt to crush into it. Also, currently explodes on any collision

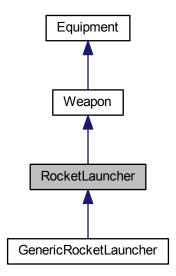
The documentation for this class was generated from the following files:

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

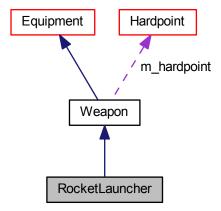
# 7.168 RocketLauncher Class Reference

#include <rocketlauncher.h>

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**

### 7.168.1 Detailed Description

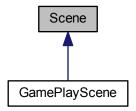
Base class for every Weapon that shoots rockets, i.e. projectiles that follow a target

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

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

# 7.169 Scene Class Reference

Inheritance diagram for Scene:



### **Public Member Functions**

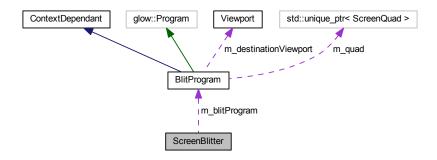
- virtual void draw (const Camera &camera, glow::FrameBufferObject \*destination, const Viewport &destinationViewport, EyeSide side=EyeSide::None) const =0
- virtual void update (float deltaSec)=0

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

· src/display/scene.h

## 7.170 ScreenBlitter Class Reference

Collaboration diagram for ScreenBlitter:



#### **Public Member Functions**

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

### **Protected Attributes**

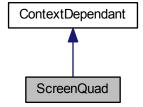
• BlitProgram \* m\_blitProgram

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

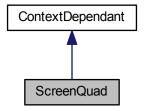
- · src/display/screenblitter.h
- src/display/screenblitter.cpp

# 7.171 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

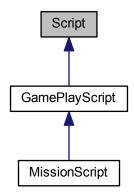
### **Protected Attributes**

- glow::ref\_ptr
   glow::VertexArrayObject > m\_vertexArrayObject
- $\bullet \ \, \mathsf{glow::ref\_ptr} \! < \mathsf{glow::Buffer} > \mathbf{m\_vertexBuffer}$

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

# 7.172 Script Class Reference

Inheritance diagram for Script:



Collaboration diagram for Script:



# **Public Member Functions**

- void start ()
- · bool started () const
- void stop ()
- bool **stopped** () const
- virtual void load (const std::string &path)
- virtual void loadString (const std::string &script)
- void **update** (float deltaSec)
- const std::string & debugStatus ()
- int apiSetDebugStatus (const std::string &string)

### **Protected Member Functions**

• void addBindings (Bindings \*bindings)

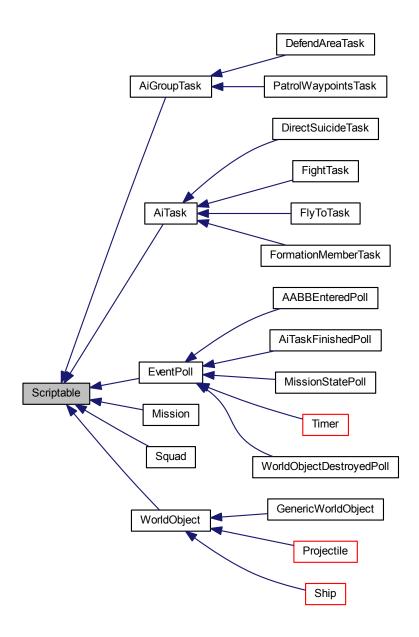
### **Protected Attributes**

- $std::unique\_ptr < LuaWrapper > m_lua$
- bool m\_started
- bool m\_stopped
- std::string m\_debugStatus
- std::vector< std::unique\_ptr</li>Bindings >> m\_bindings

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

## 7.173 Scriptable Class Reference

Inheritance diagram for Scriptable:



#### **Public Member Functions**

- int scriptKey () const
- void setScriptKey (int key)
- bool isScriptLocal () const
- void setScriptLocal (bool local)

## **Static Public Attributes**

• static const int INVALID\_KEY = -1

#### **Protected Attributes**

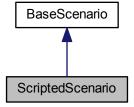
- int m key
- bool m\_local

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

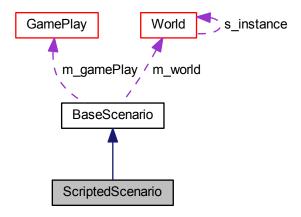
- src/scripting/scriptable.h
- src/scripting/scriptable.cpp

## 7.174 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**

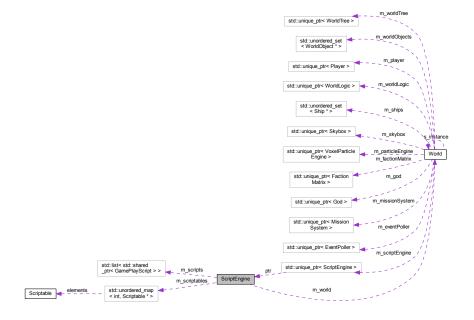
std::shared ptr< GamePlayScript > m\_script

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

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

## 7.175 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)

- · bool keyValid (int key) const
- · void update (float deltaSec)

#### **Protected Member Functions**

- Scriptable \* getScriptable (int key)
- void performRemovals ()
- void removeScriptable (Scriptable \*scriptable)

#### **Protected Attributes**

- World \* m\_world
- std::list< std::shared\_ptr</li>
   GamePlayScript >> m\_scripts
- std::unordered\_map< int,</li>
   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

## 7.176 SecondaryInputValues Struct Reference

#### **Public Attributes**

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

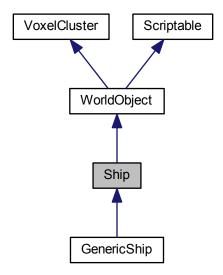
The documentation for this struct was generated from the following files:

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

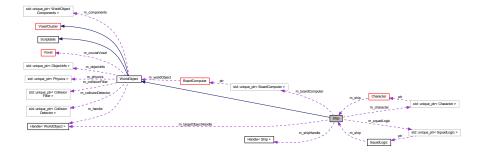
## 7.177 Ship Class Reference

#include <ship.h>

Inheritance diagram for Ship:



#### Collaboration diagram for Ship:



#### **Public Member Functions**

- virtual WorldObjectType  ${\bf objectType}$  () 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**

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**

### 7.177.1 Detailed Description

A Ship is a WorldObject with a Pilot (Character), thus can have an AiTask and a Squad membership

- The documentation for this class was generated from the following files:
  - · src/worldobject/ship.h
  - src/worldobject/ship.cpp

## 7.178 SimpleWayfind Class Reference

#### **Static Public Member Functions**

• static glm::vec3 calculateTravelPoint (WorldObject &object, glm::vec3 targetPoint)

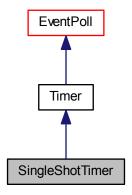
## **Static Protected Member Functions**

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

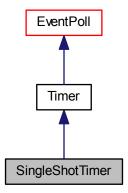
- · src/utils/simplewayfind.h
- · src/utils/simplewayfind.cpp

# 7.179 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\_ticking

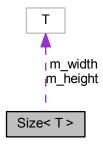
#### **Additional Inherited Members**

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

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

## 7.180 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)

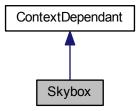
#### **Protected Attributes**

- T m\_width
- T m\_height

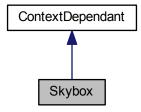
- · src/geometry/size.h
- · src/geometry/size.inl

## 7.181 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

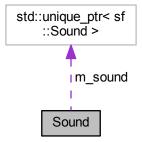
#### **Protected Attributes**

- 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

- · src/skybox.h
- · src/skybox.cpp

### 7.182 Sound Class Reference

Collaboration diagram for Sound:



## **Public Types**

• enum Status { Paused, Playing, Stopped, Null }

#### **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**

•  $std::unique\_ptr < sf::Sound > m\_sound$ 

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

## 7.183 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 ()
- · void stopAll ()

#### **Static Public Member Functions**

static SoundManager \* current ()

#### **Protected Member Functions**

- sf::SoundBuffer \* obtain (std::string soundFile)
- void cleanUp ()
- void forcedCleanup ()
- std::shared\_ptr< Sound > createNullSound ()

#### **Protected Attributes**

- std::unordered\_map
  - < std::string, sf::SoundBuffer \* > m\_buffer
- std::list< std::shared\_ptr
  - < Sound > > m\_sounds
- int m\_nextCleanup

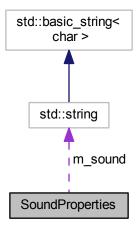
### **Static Protected Attributes**

static SoundManager \* s\_current

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

## 7.184 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

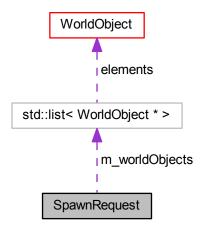
The documentation for this class was generated from the following files:

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

## 7.185 SpawnRequest Class Reference

#include <spawnrequest.h>

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

## 7.185.1 Detailed Description

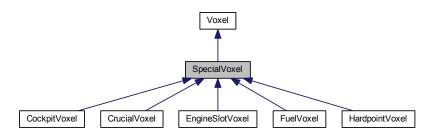
Encapsulates the request to &God to spawn a number of &WorldObject into the &World Currently available metadata is:

· deleteOnRejection: Controls whether the object is deleted when it can't be spawned

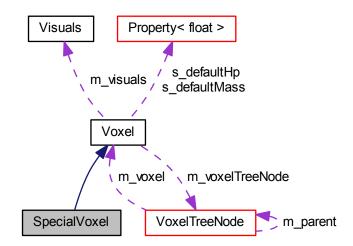
- · src/world/spawnrequest.h
- · src/world/spawnrequest.cpp

## 7.186 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**

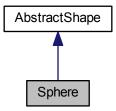
• int m\_index

### **Additional Inherited Members**

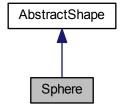
- · src/voxel/specialvoxel.h
- src/voxel/specialvoxel.cpp

## 7.187 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**

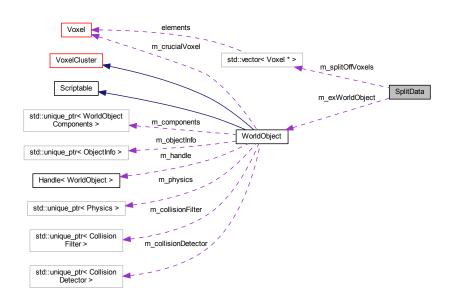
- glm::vec3 m\_position
- float m\_radius

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

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

## 7.188 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 IIf ()

#### **Protected Attributes**

- WorldObject \* m\_exWorldObject
- std::vector<  $Voxel * > m_splitOffVoxels$
- glm::ivec3 m\_llf

- · src/world/helper/splitdata.h
- · src/world/helper/splitdata.cpp

## 7.189 SplitDetector Class Reference

Collaboration diagram for SplitDetector:



#### Classes

struct VoxelGroup

#### **Public Member Functions**

- void searchSplitOffs (std::list< WorldObjectModification > &worldObjectModifications)
- std::vector< std::shared\_ptr</li>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)

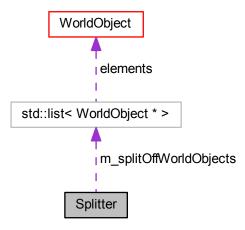
#### **Protected Attributes**

- 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

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

## 7.190 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

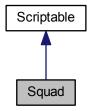
The documentation for this class was generated from the following files:

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

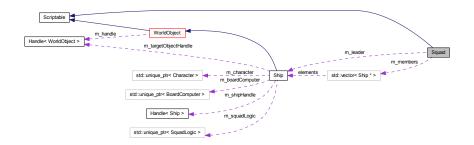
## 7.191 Squad Class Reference

#include <squad.h>

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**

### 7.191.1 Detailed Description

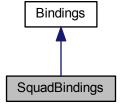
Ships can be joined to Squads to give them AiGroupTasks. The Squad's leader is the one to execute the Squad's task.

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

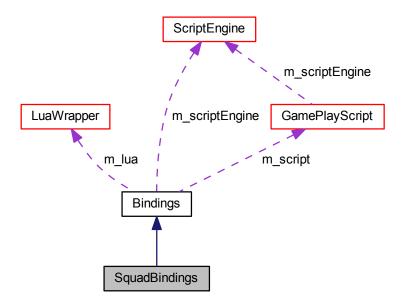
- src/ai/squad.h
- · src/ai/squad.cpp

# 7.192 SquadBindings Class Reference

Inheritance diagram for SquadBindings:



Collaboration diagram for SquadBindings:



### **Public Member Functions**

• SquadBindings (GamePlayScript &script)

#### **Protected Member Functions**

- · virtual void bind () override
- apikey apiCreateSquad (apikey leader)
- int apiJoinSquad (apikey squad, apikey ship)
- int apiCreatePatrolWaypointsTask (apikey squad)
- int apiAddPatrolwWaypointPoint (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**

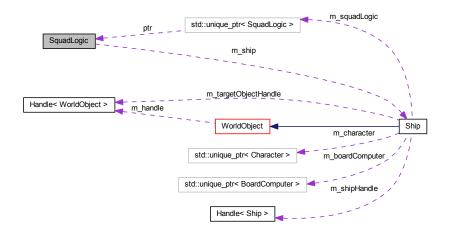
The documentation for this class was generated from the following files:

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

## 7.193 SquadLogic Class Reference

#include <squadlogic.h>

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

## 7.193.1 Detailed Description

The SquadLogic is the single member's endpoint for interaction with a Squad

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

- src/ai/squadlogic.h
- src/ai/squadlogic.cpp

## 7.194 Starfield::StarData Struct Reference

## **Public Attributes**

• glm::vec3 pos

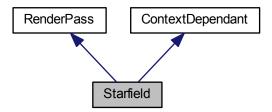
- · float brightness
- float size

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

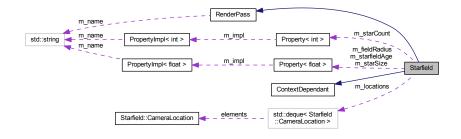
• src/display/rendering/starfield.h

## 7.195 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
- $\bullet \ \, \mathsf{glow::ref\_ptr} < \mathsf{glow::Buffer} > \mathbf{m\_gpuBuffer}$
- glow::Array < StarData > m\_cpuBuffer

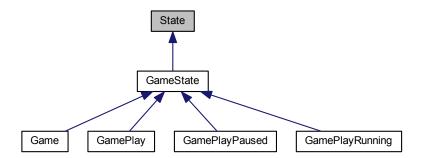
The documentation for this class was generated from the following files:

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

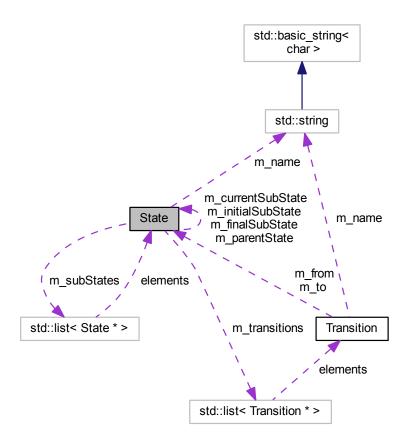
### 7.196 State Class Reference

#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)

7.196 State Class Reference 231

- 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

#### 7.196.1 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)

#### 7.196.2 Member Function Documentation

```
7.196.2.1 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, GamePlayRunning, and GamePlayPaused.

```
7.196.2.2 void State::onLeft() [virtual]
```

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

 $Reimplemented\ in\ Game Play,\ Game State,\ Game Play Running,\ and\ Game Play Paused.$ 

```
7.196.2.3 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

7.196.2.4 void State::transit ( State \* target ) [protected]

Ensures the graph of m\_currentSubState points from the root to targt calls onLeft() on every state left and on-Entered() on every entered

7.196.2.5 void State::update (float deltaSec) [virtual]

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

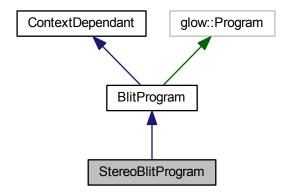
Reimplemented in GamePlay, Game, GameState, GamePlayRunning, and GamePlayPaused.

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

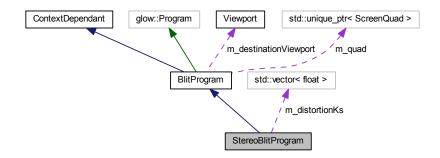
- · src/utils/statemachine/state.h
- · src/utils/statemachine/state.cpp

## 7.197 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

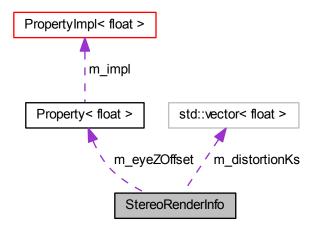
#### **Additional Inherited Members**

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

- · src/programs/stereoblitprogram.h
- src/programs/stereoblitprogram.cpp

#### 7.198 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

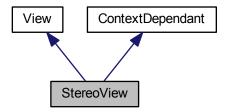
## **Protected Attributes**

- 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

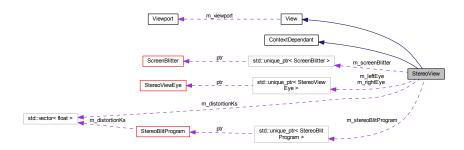
- src/display/stereorenderinfo.h
- · src/display/stereorenderinfo.cpp

## 7.199 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

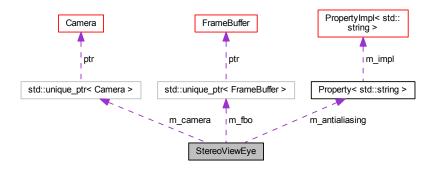
- std::unique\_ptr
  - < StereoBlitProgram > m\_stereoBlitProgram
- glm::vec2 m\_leftEyeLensCenter
- glm::vec2 m\_rightEyeLensCenter
- std::vector< float > m\_distortionKs
- · float m distortionScale

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

- · src/display/stereoview.h
- · src/display/stereoview.cpp

## 7.200 StereoViewEye Class Reference

Collaboration diagram for StereoViewEye:



#### **Public Member Functions**

- StereoViewEye (const glm::ivec2 &viewportResolution, const StereoRenderInfo &stereoRenderInfo, Eye-Side side)
- FrameBuffer & fbo ()
- const Camera & camera () const
- void setViewportResolution (const glm::ivec2 &viewportResolution)
- · void draw (const Scene &scene, const CameraHead &cameraHead)

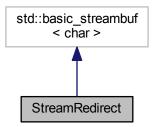
#### **Protected Attributes**

- glm::vec3 m offset
- · EyeSide m\_side
- Property < std::string > m\_antialiasing
- float m\_distortionScale
- glm::ivec2 m textureSize
- std::unique\_ptr< Camera > m\_camera
- std::unique\_ptr< FrameBuffer > m\_fbo

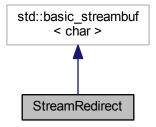
- · src/display/stereovieweye.h
- · src/display/stereovieweye.cpp

## 7.201 StreamRedirect Class Reference

Inheritance diagram for StreamRedirect:



Collaboration diagram for StreamRedirect:



## **Public Member Functions**

• StreamRedirect (std::ostream &stream, HUD \*hud, bool copy=false)

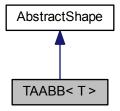
### **Protected Member Functions**

- virtual int\_type **overflow** (int\_type v) override
- virtual std::streamsize xsputn (const char \*p, std::streamsize n) override

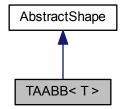
- src/ui/streamredirect.h
- src/ui/streamredirect.cpp

## 7.202 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 > &urb)
- template<typename OtherT >
  - **TAABB** (const TAABB< OtherT > &other)
- const glm::detail::tvec3< T > & IIf () const
- void setLLF (const glm::detail::tvec3< T > &llf)
- const glm::detail::tvec3< T > & urb () const
- void setURB (const glm::detail::tvec3< T > &urb)
- TaxisMin (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\_urb

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

- · src/geometry/aabb.h
- · src/geometry/aabb.inl

## 7.203 TargetSelector Class Reference

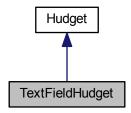
#### **Public Member Functions**

- TargetSelector (Player \*player)
- · void selectTarget (bool next)

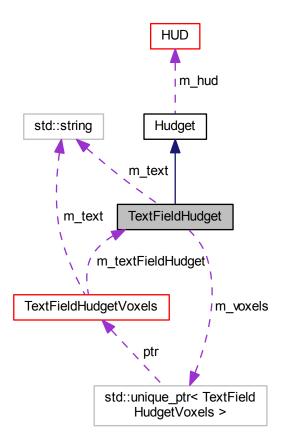
- · src/ui/targetselector.h
- src/ui/targetselector.cpp

# 7.204 TextFieldHudget Class Reference

Inheritance diagram for TextFieldHudget:



Collaboration diagram for TextFieldHudget:



#### **Public Member Functions**

- TextFieldHudget (HUD \*hud, const glm::vec3 &direction, float scale=0.5f, const std::string &content="", FontSize fontSize=FontSize::SIZE5x7)
- void setText (const 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\_text
- · 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

# 7.205 TextFieldHudgetVoxels Class Reference

Collaboration diagram for TextFieldHudgetVoxels:



#### **Public Member Functions**

- TextFieldHudgetVoxels (TextFieldHudget \*textFieldHudget, const glm::vec3 &direction, float scale=0.5f, const std::string &text="", FontSize=FontSize::SIZE5x7)
- void setText (const std::string &text)
- 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 offsetToCenter (bool upper, bool left) const
- 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

## **Protected Attributes**

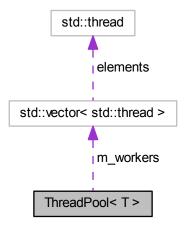
- · FontSize m fontSize
- TextFieldHudget \* m\_textFieldHudget
- std::string m\_text
- VoxelFont \* m\_voxelFont
- glm::vec3 m\_direction
- · float m width
- float m\_height
- float m\_scale
- · float m offset

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

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

# 7.206 ThreadPool < T > Class Template Reference

Collaboration diagram for ThreadPool< T >:



#### **Public Member Functions**

- ThreadPool (int threadcount=4, int chunksize=100)
- void  ${\bf map}$  (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

#### 7.207 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 ()

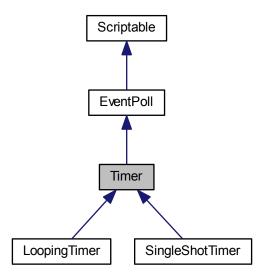
#### **Protected Attributes**

- $std::chrono::duration < float > m_intervalSecs$
- std::chrono::high\_resolution\_clock::time\_point m\_lastExecution

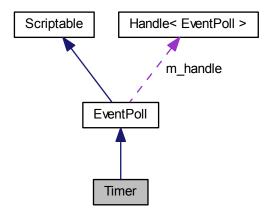
- src/utils/timedtask.h
- · src/utils/timedtask.cpp

# 7.208 Timer Class Reference

Inheritance diagram for Timer:



Collaboration diagram for Timer:



- Timer (float interval, const std::function<  $\mbox{void}(\mbox{\scriptsize })>$  &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

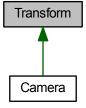
#### **Additional Inherited Members**

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

- · src/events/timer.h
- · src/events/timer.cpp

## 7.209 Transform Class Reference

Inheritance diagram for Transform:



- 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::guat &guat)
- const glm::vec3 & center () const
- void setCenter (const glm::vec3 &center)
- 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

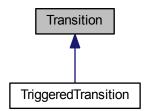
The documentation for this class was generated from the following files:

- src/geometry/transform.h
- src/geometry/transform.cpp

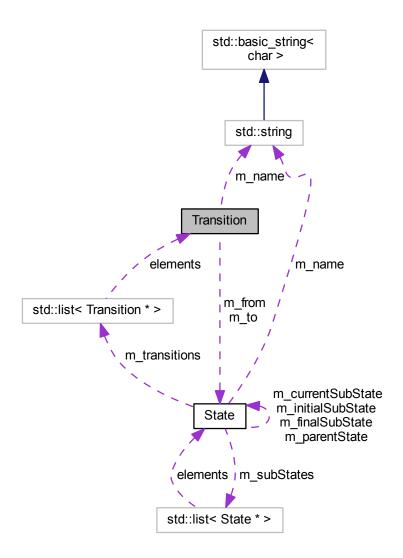
# 7.210 Transition Class Reference

#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

## 7.210.1 Detailed Description

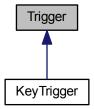
Abstract base class for a transition from one state to another. Derived classes need to implement isPossible(). The actual transition is performed by the update() of the parentstate of from- Transitions register themself at from, no need to call addTransition() on the State. to isn't informed it's "target" 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:

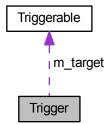
- · src/utils/statemachine/transition.h
- · src/utils/statemachine/transition.cpp

# 7.211 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**

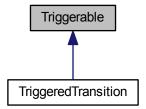
• Triggerable \* m\_target

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

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

# 7.212 Triggerable Class Reference

Inheritance diagram for Triggerable:



## **Public Member Functions**

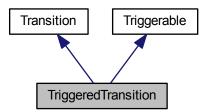
• virtual void trigger ()=0

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

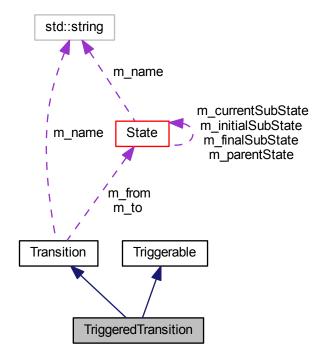
• src/utils/statemachine/triggerable.h

# 7.213 TriggeredTransition Class Reference

Inheritance diagram for TriggeredTransition:



Collaboration diagram for TriggeredTransition:



- 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

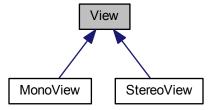
# 7.214 glow::Uniform < T > Class Template Reference

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

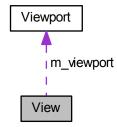
· src/voxel/voxelrenderer.h

# 7.215 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**

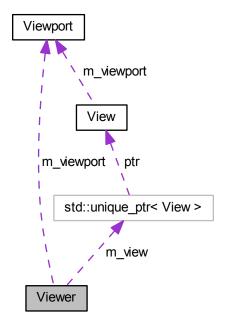
Viewport m\_viewport

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

- · src/display/view.h
- · src/display/view.cpp

## 7.216 Viewer Class Reference

Collaboration diagram for Viewer:



- 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

# 7.217 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

#### 7.218 Visuals Class Reference

#include <visuals.h>

- 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

## 7.218.1 Detailed Description

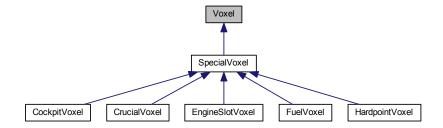
Provides all attributes needed to render something with all currently implemented visual effects

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

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

### 7.219 Voxel Class Reference

Inheritance diagram for Voxel:



Collaboration diagram for Voxel:



- **Voxel** (const glm::ivec3 &gridCell, uint32\_t color=0xFFFFF, 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

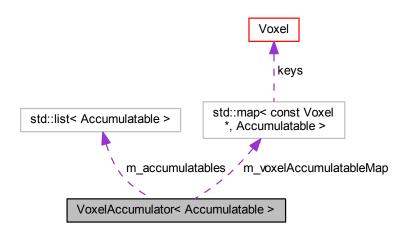
#### **Static Protected Attributes**

- static Property< float > \* s\_defaultMass
- static Property< float > \* s\_defaultHp

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

# 7.220 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 ()

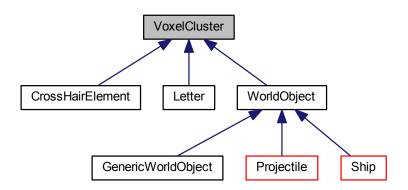
#### **Protected Attributes**

- std::map< const Voxel</li>
  - \*, Accumulatable > m\_voxelAccumulatableMap
- std::list< Accumulatable > m\_accumulatables

- src/world/helper/voxelaccumulator.h
- · src/world/helper/voxelaccumulator.inl

#### 7.221 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,</li>
   Voxel \* > m\_voxels
- std::unique\_ptr< VoxelRenderData > m\_voxelRenderData

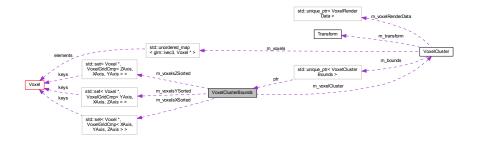
- std::unique\_ptr< VoxelClusterBounds > m\_bounds
- Transform m\_transform

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

- · src/voxel/voxelcluster.h
- · src/voxel/voxelcluster.cpp

# 7.222 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</li>
  - \*, VoxelGridCmp< XAxis, YAxis,

ZAxis > m\_voxelsXSorted

- std::set< Voxel</li>
  - \*, VoxelGridCmp< YAxis, XAxis,

 ${\sf ZAxis} >> {\sf m\_voxelsYSorted}$ 

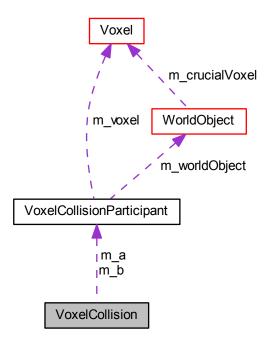
```
    std::set< Voxel</li>
    *, VoxelGridCmp< ZAxis, XAxis,</li>
    YAxis >> m_voxelsZSorted
```

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

- · src/voxel/voxelclusterbounds.h
- src/voxel/voxelclusterbounds.cpp

# 7.223 VoxelCollision Class Reference

Collaboration diagram for VoxelCollision:



# **Public Member Functions**

- VoxelCollision (const VoxelCollisionParticipant &a, const VoxelCollisionParticipant &b)
- VoxelCollisionParticipant & a ()
- VoxelCollisionParticipant & b ()

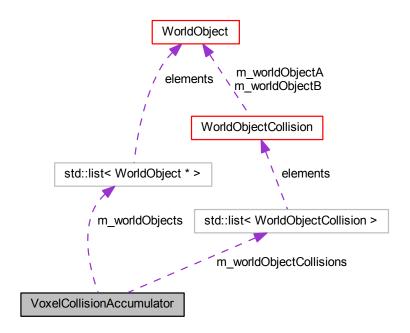
#### **Protected Attributes**

- VoxelCollisionParticipant m\_a
- VoxelCollisionParticipant m\_b

- · src/collision/voxelcollision.h
- src/collision/voxelcollision.cpp

## 7.224 VoxelCollisionAccumulator Class Reference

Collaboration diagram for VoxelCollisionAccumulator:



## **Public Member Functions**

- void parse (std::list< VoxelCollision > &voxelCollisions)
- void applyOnCollsionHooks ()
- std::list< WorldObjectCollision > & worldObjectCollisions ()
- std::list< WorldObject \* > & worldObjects ()

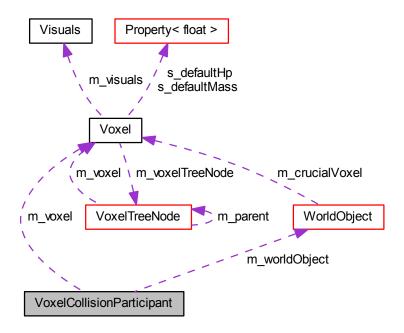
# **Protected Attributes**

- $std::list < WorldObjectCollision > m_worldObjectCollisions$
- std::list< WorldObject \* > m\_worldObjects

- · src/world/handler/voxelcollisionaccumulator.h
- src/world/handler/voxelcollisionaccumulator.cpp

# 7.225 VoxelCollisionParticipant Class Reference

Collaboration diagram for VoxelCollisionParticipant:



# **Public Member Functions**

- $\bullet \ \ \textbf{VoxelCollisionParticipant} \ (\textbf{WorldObject} * \textbf{worldObject}, \ \textbf{Voxel} * \textbf{voxel})$
- WorldObject \* worldObject ()
- Voxel \* voxel ()

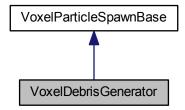
## **Protected Attributes**

- WorldObject \* m\_worldObject
- Voxel \* m\_voxel

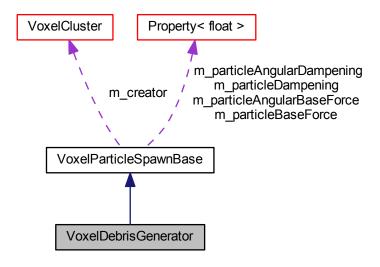
- · src/collision/voxelcollision.h
- src/collision/voxelcollision.cpp

# 7.226 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**

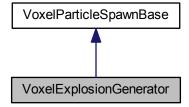
- glm::quat m\_orientation
- int m\_density
- float m\_spawnProbability

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

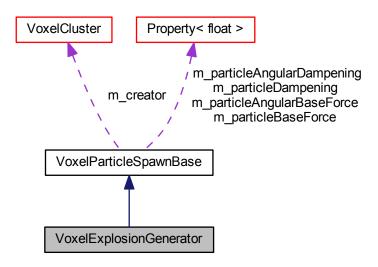
- src/voxeleffect/voxeldebrisgenerator.h
- src/voxeleffect/voxeldebrisgenerator.cpp

# 7.227 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**

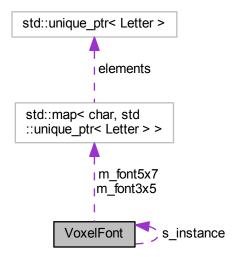
- float m\_radius
- int m\_count

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

- · src/voxeleffect/voxelexplosiongenerator.h
- src/voxeleffect/voxelexplosiongenerator.cpp

#### 7.228 VoxelFont Class Reference

Collaboration diagram for VoxelFont:



- 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

# 7.229 VoxelGridCmp< highPriorityAxis, middlePriorityAxis, lowPriorityAxis > Class Template Reference

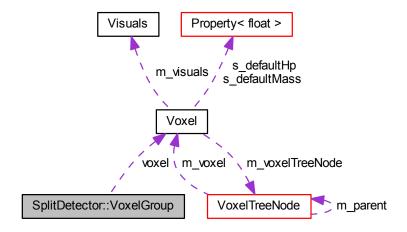
#### **Public Member Functions**

• bool operator() (const Voxel \*voxel1, const Voxel \*voxel2) const

- · src/voxel/voxelgridcmp.h
- · src/voxel/voxelgridcmp.inl

# 7.230 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

# 7.231 VoxelHangman Class Reference

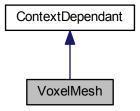
**Public Member Functions** 

- void applyOnDestructionHooks (std::list< DamageImpact > &deadlyDamageImpacts)
- void removeDestroyedVoxels (std::list< DamageImpact > &deadlyDamageImpacts)

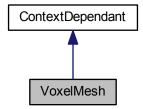
- · src/world/handler/voxelhangman.h
- · src/world/handler/voxelhangman.cpp

## 7.232 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

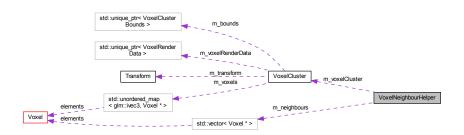
#### **Protected Attributes**

- bool m initialized
- glow::ref\_ptr< glow::Buffer > m\_vertexBuffer

- · src/voxeleffect/voxelmesh.h
- src/voxeleffect/voxelmesh.cpp

# 7.233 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

# 7.234 VoxelParticleData Struct Reference

# **Public Types**

enum Status { Removed, Alive, Dead }

#### **Public Attributes**

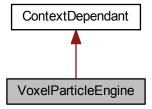
- · Status status
- glm::vec3 creationPosition
- glm::vec3 creationEulers
- glm::vec3 directionalSpeed
- glm::vec3 angularSpeed
- float creationTime
- · float deathTime
- · float scale
- uint32\_t color
- · float emissiveness

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

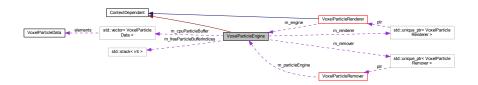
· src/voxeleffect/voxelparticledata.h

# 7.235 VoxelParticleEngine Class Reference

Inheritance diagram for VoxelParticleEngine:



Collaboration diagram for VoxelParticleEngine:



- float time () const
- int particleCount () const
- int particleDataCount () const
- VoxelParticleData \* particleData (int index)
- std::vector < VoxelParticleData > & particleDataVector ()

- 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)
- virtual void beforeContextDestroy ()
- · virtual void afterContextRebuild ()

#### **Protected Attributes**

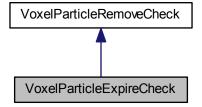
- · float m\_time
- · bool m\_initialized
- · std::unique\_ptr
  - < VoxelParticleRenderer > m\_renderer
- std::unique ptr
  - < VoxelParticleRemover > m\_remover
- std::vector < VoxelParticleData > m\_cpuParticleBuffer
- std::stack< int > m\_freeParticleBufferIndices
- bool m\_gpuParticleBufferInvalid
- int m\_gpuParticleBufferInvalidBegin
- int m\_gpuParticleBufferInvalidEnd

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

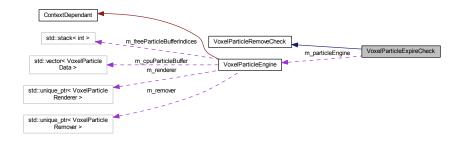
- · src/voxeleffect/voxelparticleengine.h
- src/voxeleffect/voxelparticleengine.cpp

# 7.236 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

## 7.237 VoxelParticleFutureCheck Class Reference

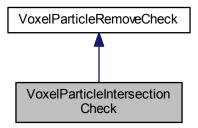
#### **Static Public Member Functions**

• static bool intersectsIn (const VoxelParticleData &particle, float futureSecs, const VoxelCluster &against)

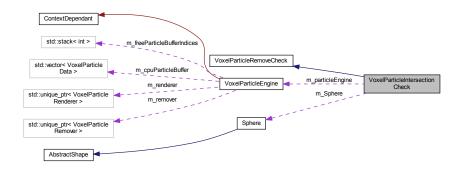
- · src/voxeleffect/particlechecks/voxelparticlefuturecheck.h
- src/voxeleffect/particlechecks/voxelparticlefuturecheck.cpp

## 7.238 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 ()

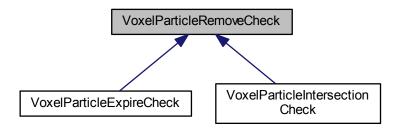
## **Protected Attributes**

- const VoxelParticleEngine & m\_particleEngine
- · Sphere m\_Sphere

- src/voxeleffect/particlechecks/voxelparticleintersectioncheck.h
- src/voxeleffect/particlechecks/voxelparticleintersectioncheck.cpp

#### 7.239 VoxelParticleRemoveCheck Class Reference

Inheritance diagram for VoxelParticleRemoveCheck:



#### **Public Member Functions**

- virtual bool isDead (const VoxelParticleData &particle)=0
- virtual void beforeCheck ()

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

- src/voxeleffect/particlechecks/voxelparticleremovecheck.h
- src/voxeleffect/particlechecks/voxelparticleremovecheck.cpp

## 7.240 VoxelParticleRemover Class Reference

Collaboration diagram for VoxelParticleRemover:



- 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**

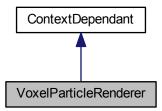
- VoxelParticleEngine \* m\_particleEngine
- std::vector< std::shared\_ptr</li>
   VoxelParticleRemoveCheck >> m\_checker
- std::unique\_ptr< ThreadPool</li>< VoxelParticleData > > m\_threadPool
- Property< float > m\_interval
- Property < bool > m\_multithreaded
- int m\_currentIndex

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

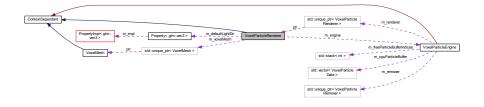
- · src/voxeleffect/particlechecks/voxelparticleremover.h
- src/voxeleffect/particlechecks/voxelparticleremover.cpp

# 7.241 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 setupVertexAttribute (GLint offset, const std::string &name, int numPerVertex, GLenum type, G-Lboolean normalised, int bindingNum)
- void setupVertexAttribDivisors ()
- void setBufferSize (int bufferSize)
- · virtual void beforeContextDestroy () override
- · virtual void afterContextRebuild () override

#### **Protected Attributes**

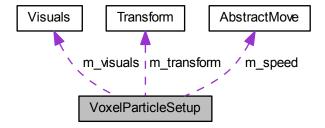
- std::unique\_ptr< VoxelMesh > m\_voxelMesh
- bool m\_initialized
- VoxelParticleEngine \* m\_engine
- int m bufferSize
- Property < glm::vec3 > m\_defaultLightDir
- glow::ref\_ptr< glow::Buffer > m\_gpuParticleBuffer
- glow::ref\_ptr< glow::Program > m\_program
- · glow::ref\_ptr
  - < glow::VertexArrayObject > m\_vertexArrayObject

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

- src/voxeleffect/voxelparticlerenderer.h
- src/voxeleffect/voxelparticlerenderer.cpp

# 7.242 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**

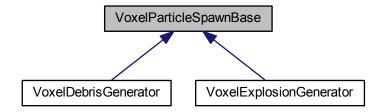
- · Transform m transform
- · Visuals m\_visuals
- Speed m\_speed
- · float m\_lifetime

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

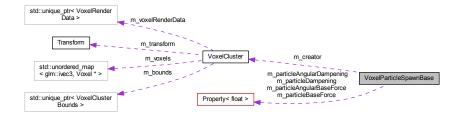
- · src/voxeleffect/voxelparticlesetup.h
- src/voxeleffect/voxelparticlesetup.cpp

# 7.243 VoxelParticleSpawnBase Class Reference

Inheritance diagram for VoxelParticleSpawnBase:



Collaboration diagram for VoxelParticleSpawnBase:



- · 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 \*angularDampening-Name, char \*baseForceName, char \*angularBaseForceName)
- glm::vec3 createDirectionalSpeed ()
- glm::vec3 createAngularSpeed ()
- float createLifetime ()

### **Protected Attributes**

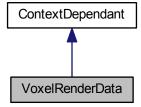
- const VoxelCluster \* m creator
- glm::vec3 m\_position
- float m\_scale
- float m\_scaleRandomization
- · float m force
- float m\_forceRandomization
- float m lifetime
- · float m lifetimeRandomization
- · int m color
- float m\_emissiveness
- glm::vec3 m\_impactVector
- Property < float > m\_particleDampening
- Property < float > m\_particleAngularDampening
- Property < float > m\_particleBaseForce
- Property < float > m\_particleAngularBaseForce

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

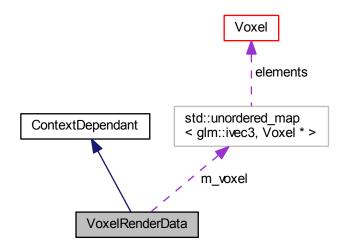
- · src/voxeleffect/voxelparticlespawnbase.h
- src/voxeleffect/voxelparticlespawnbase.cpp

### 7.244 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 setupVertexAttribute (GLint offset, const std::string &name, int numPerVertex, GLenum type, G-Lboolean normalised, int bindingNum)
- virtual void beforeContextDestroy () override
- · virtual void afterContextRebuild () override

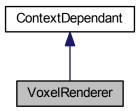
# **Protected Attributes**

- std::unordered\_map< glm::ivec3,</li>
  - $Voxel * > & m_voxel$
- bool m\_isDirty
- int m bufferSize
- glow::ref ptr< glow::Buffer > m\_voxelDataBuffer
- glow::ref\_ptr
  - < glow::VertexArrayObject > m\_vertexArrayObject

- · src/voxel/voxelrenderdata.h
- src/voxel/voxelrenderdata.cpp

# 7.245 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**

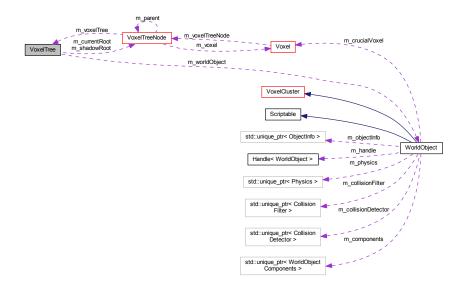
static std::weak\_ptr< VoxelRenderer > s\_instance

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

- · src/voxel/voxelrenderer.h
- src/voxel/voxelrenderer.cpp

# 7.246 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**

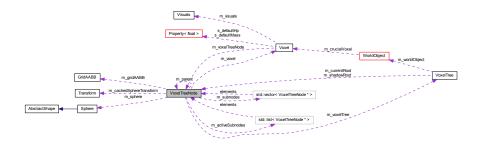
- VoxelTreeNode \* m\_shadowRoot
- VoxelTreeNode \* m\_currentRoot
- WorldObject \* m\_worldObject

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

- · src/voxel/voxeltree.h
- src/voxel/voxeltree.cpp

### 7.247 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**

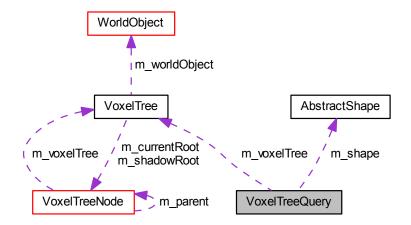
- int m\_octIndex
- VoxelTreeNode \* m parent
- VoxelTree \* m\_voxelTree
- GridAABB m\_gridAABB
- Sphere m\_sphere
- Transform m\_cachedSphereTransform
- bool m\_active
- std::vector< VoxelTreeNode \* > m\_subnodes
- $std::list < VoxelTreeNode * > m_activeSubnodes$
- Voxel \* m\_voxel

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

- src/voxel/voxeltreenode.h
- src/voxel/voxeltreenode.cpp

# 7.248 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

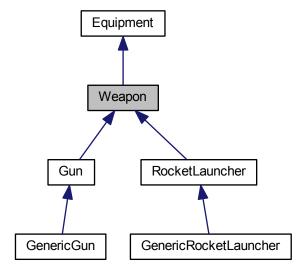
The documentation for this class was generated from the following files:

- src/voxel/voxeltreequery.h
- src/voxel/voxeltreequery.cpp

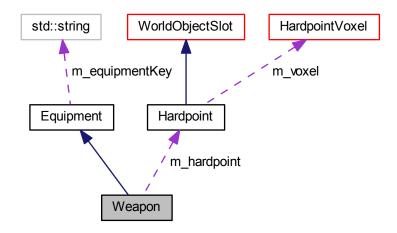
# 7.249 Weapon Class Reference

#include <weapon.h>

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

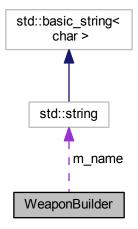
# 7.249.1 Detailed Description

Weapon to be mounted to a HardPoint

- · src/equipment/weapon.h
- · src/equipment/weapon.cpp

# 7.250 WeaponBuilder Class Reference

Collaboration diagram for WeaponBuilder:



# **Public Member Functions**

- WeaponBuilder (const std::string &name)
- Weapon \* build ()
- GenericGun \* buildGenericGun ()
- GenericRocketLauncher \* buildGenericRocketLauncher ()

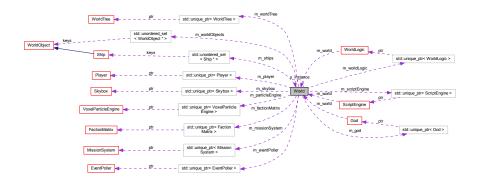
# **Protected Attributes**

• const std::string & m\_name

- · src/resource/weaponbuilder.h
- src/resource/weaponbuilder.cpp

### 7.251 World Class Reference

### Collaboration diagram for World:



### **Public Member Functions**

- Player & player ()
- Skybox & skybox ()
- WorldLogic & worldLogic ()
- God & god ()
- WorldTree & worldTree ()
- ScriptEngine & scriptEngine ()
- VoxelParticleEngine & particleEngine ()
- FactionMatrix & factionMatrix ()
- EventPoller & eventPoller ()
- MissionSystem & missionSystem ()
- std::unordered\_set
  - < WorldObject \* > & worldObjects ()
- std::unordered\_set< Ship \* > & ships ()
- · void printStatus ()
- · 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
- std::unique\_ptr< Player > m\_player
- std::unique\_ptr< ScriptEngine > m\_scriptEngine
- 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
  - < VoxelParticleEngine > m\_particleEngine
- std::unique\_ptr< FactionMatrix > m\_factionMatrix
- std::unique\_ptr< EventPoller > m\_eventPoller
- std::unique\_ptr< MissionSystem > m\_missionSystem
- std::unordered\_set< WorldObject \* > m\_worldObjects
- std::unordered set< Ship \* > m\_ships

### **Static Protected Attributes**

• static World \* s\_instance = nullptr

### **Friends**

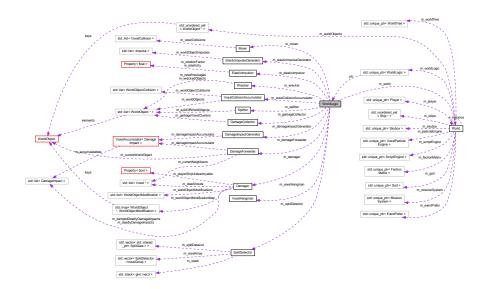
· class God

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

- · src/world/world.h
- · src/world/world.cpp

# 7.252 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< DamageImpact > 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
- · Wrecker m wrecker
- GarbageCollector m\_garbageCollector
- VoxelHangman m\_voxelHangman

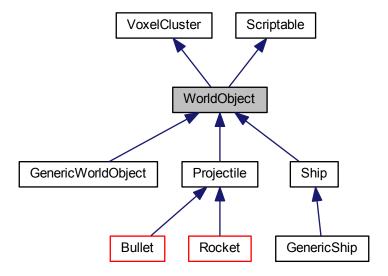
The documentation for this class was generated from the following files:

- src/world/worldlogic.h
- src/world/worldlogic.cpp

# 7.253 WorldObject Class Reference

#include <worldobject.h>

Inheritance diagram for WorldObject:



### Collaboration diagram for WorldObject:



#### **Public Member Functions**

- WorldObject (const Transform &transform)
- 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)
- bool isCrucialVoxelDestroyed ()
- 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)
- virtual bool passiveForCollisionDetection ()

### **Protected Attributes**

- $\bullet \ \, \text{std::unique\_ptr} < \\ \textbf{CollisionFilter} > \\ \textbf{m\_collisionFilter} > \\ \textbf{m\_collisionFil$
- 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
- · bool m crucialVoxelDestroyed
- float m\_collisionFieldOfDamage
- SpawnState m\_spawnState

**Additional Inherited Members** 

# 7.253.1 Detailed Description

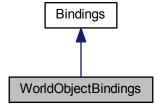
A WorldObject is an Object in our World. Being the second level in the object hierarchy, it adds CollisionDetection, Physics and SpecialVoxels aka WorldObjectComponents

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

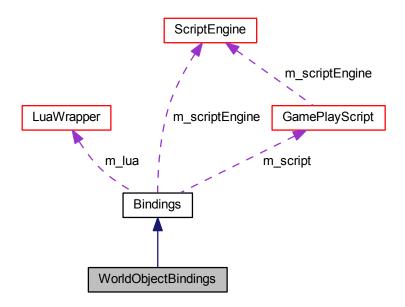
- src/worldobject/worldobject.h
- src/worldobject/worldobject.cpp

# 7.254 WorldObjectBindings Class Reference

Inheritance diagram for WorldObjectBindings:



Collaboration diagram for WorldObjectBindings:



#### **Public Member Functions**

WorldObjectBindings (GamePlayScript &script)

### **Protected Member Functions**

- · virtual void bind () override
- apikey apiPlayerShip ()
- apikey apiCreateShip (const std::string &name)
- apikey apiCreateWorldObject (const std::string &name)
- int apiSpawn (apikey worldObject)
- int apiRemove (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)
- int apiSetShowOnHud (apikey worldObject, bool show)
- int apiSetCanLockOn (apikey worldObject, bool lockon)
- apikey apiOnWorldObjectDestroyed (apikey worldObject, const std::string &callback)
- apikey apiOnAABBEntered (apikey worldObject, const glm::vec3 &llf, const glm::vec3 &urb, const std::string &callback)

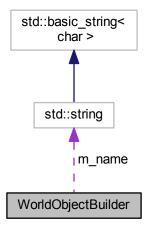
### **Additional Inherited Members**

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

# 7.255 WorldObjectBuilder Class Reference

#include <worldobjectbuilder.h>

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

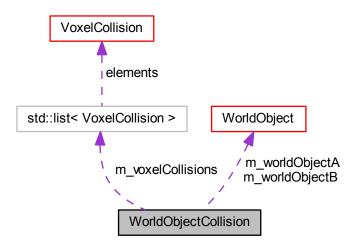
# 7.255.1 Detailed Description

Sets up WorldObject by the settinsg provided by properties

- src/resource/worldobjectbuilder.h
- src/resource/worldobjectbuilder.cpp

# 7.256 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

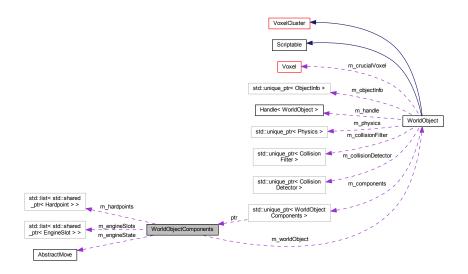
The documentation for this class was generated from the following files:

- · src/collision/worldobjectcollision.h
- · src/collision/worldobjectcollision.cpp

# 7.257 WorldObjectComponents Class Reference

#include <worldobjectcomponents.h>

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</li>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)
- otanonaroa\_ptr < riarapoint > narapoint (int
- std::list< std::shared\_ptr</li>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</li>EngineSlot >> m\_engineSlots
- std::list< std::shared\_ptr</li>Hardpoint >> m\_hardpoints
- EngineState m\_engineState

# 7.257.1 Detailed Description

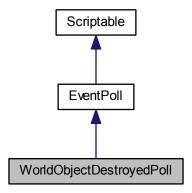
Module of the WorldObject that is responsible for managing all the components of a WorldObject (Engines, Weapons) and their respective slots. Also provides functions to trigger actions or retrieve values from a whole category of components. (like, fire all weapons, set all engines)

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

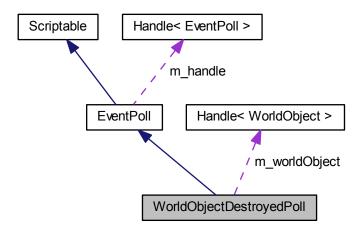
- · src/worldobject/worldobjectcomponents.h
- src/worldobject/worldobjectcomponents.cpp

# 7.258 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

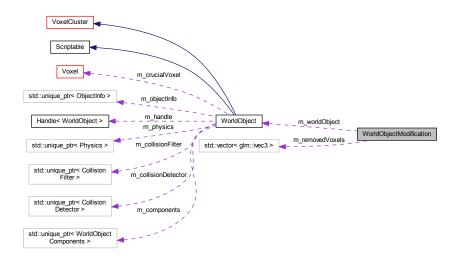
#### **Additional Inherited Members**

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

- · src/events/worldobjectdestroyedpoll.h
- · src/events/worldobjectdestroyedpoll.cpp

# 7.259 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**

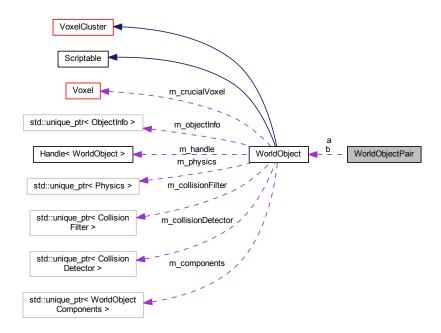
- WorldObject \* m\_worldObject
- std::vector< glm::ivec3 > m\_removedVoxels

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

- src/world/helper/worldobjectmodification.h
- · src/world/helper/worldobjectmodification.cpp

# 7.260 WorldObjectPair Struct Reference

Collaboration diagram for WorldObjectPair:



### **Public Member Functions**

- WorldObjectPair (WorldObject \*a, WorldObject \*b)
- bool operator < (const WorldObjectPair &other) const

### **Public Attributes**

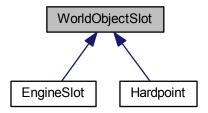
- WorldObject \* a
- WorldObject \* b

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

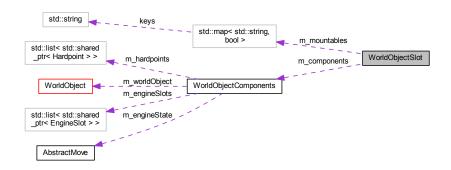
src/world/handler/voxelcollisionaccumulator.cpp

# 7.261 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

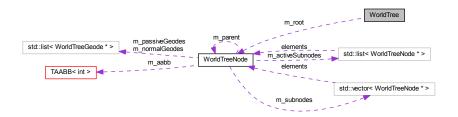
### **Protected Attributes**

- WorldObjectComponents \* m\_components
- std::map< std::string, bool > m\_mountables
- int m\_index

- src/equipment/worldobjectslot.h
- · src/equipment/worldobjectslot.cpp

# 7.262 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**

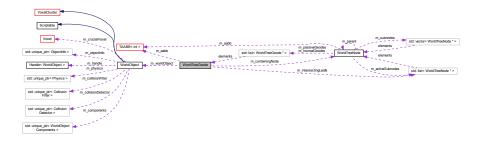
WorldTreeNode \* m\_root

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

- · src/worldtree/worldtree.h
- src/worldtree/worldtree.cpp

# 7.263 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)
- bool isPassive ()

### **Protected Attributes**

- WorldObject \* m\_worldObject
- WorldTreeNode \* m\_containingNode
- · IAABB m aabb
- · bool m\_passive
- std::list< WorldTreeNode \* > m\_intersectingLeafs

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

- · src/worldtree/worldtreegeode.h
- src/worldtree/worldtreegeode.cpp

# 7.264 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 convertToGroup (WorldTreeNode \*initialSubnode=nullptr)
- void moveToSubnode (WorldTreeGeode \*geode, WorldTreeNode \*subnode)
- void subnodeActivated (WorldTreeNode \*subnode)
- void subnodeDeactivated (WorldTreeNode \*subnode)
- std::list< WorldTreeGeode \* > & geodesList (WorldTreeGeode \*geode)

### **Protected Attributes**

- WorldTreeNode \* m\_parent
- IAABB m aabb
- · int m octIndex
- float m\_extent
- bool m\_active
- std::list< WorldTreeGeode \* > m\_normalGeodes
- std::list< WorldTreeGeode \* > m\_passiveGeodes
- std::vector< WorldTreeNode \* > m\_subnodes
- std::list< WorldTreeNode \* > m\_activeSubnodes

### **Static Protected Attributes**

- static const int MIN\_EXTENT = 16
- static const int MAX\_GEODES = 8

#### 7.264.1 Member Function Documentation

```
7.264.1.1 void WorldTreeNode::convertToGroup ( WorldTreeNode * initialSubnode = nullptr ) [protected]
```

convert a leaf to a node with subnodes

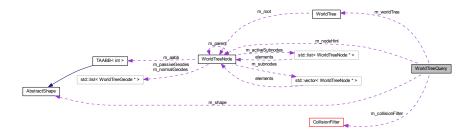
```
7.264.1.2 void WorldTreeNode::moveToSubnode ( WorldTreeGeode * geode, WorldTreeNode * subnode ) [protected]
```

move a geode to the specified subnode if it is contained in its aabb

- · src/worldtree/worldtreenode.h
- src/worldtree/worldtreenode.cpp

# 7.265 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)

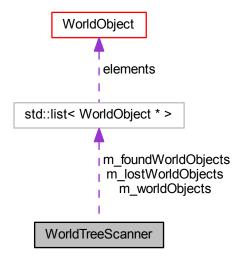
### **Protected Attributes**

- WorldTree \* m\_worldTree
- WorldTreeNode \* m\_nodeHint
- CollisionFilter \* m\_collisionFilter
- const AbstractShape \* m\_shape
- bool m\_queryInterrupted

- · src/worldtree/worldtreequery.h
- · src/worldtree/worldtreequery.cpp

### 7.266 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  $* > m_foundWorldObjects$

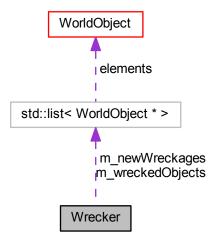
std::list< WorldObject \* > m\_lostWorldObjects

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

- src/worldtree/worldtreescanner.h
- src/worldtree/worldtreescanner.cpp

# 7.267 Wrecker Class Reference

Collaboration diagram for Wrecker:



# **Public Member Functions**

- void detectWreckedObjects (std::list< WorldObjectModification > &worldObjectModifications)
- std::list< WorldObject \* > & wreckedObjects ()
- std::list< WorldObject \* > & newWreckages ()

### **Protected Member Functions**

WorldObject \* wreckFromObject (WorldObject \*object)

### **Protected Attributes**

- std::list< WorldObject \* > m\_wreckedObjects
- std::list< WorldObject \* > m\_newWreckages

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

# Index

AABBEnteredPoll, 25	DamageImpactGenerator, 74
AbstractMove, 26	Damager, 74
AbstractPropertyCollection, 26	DdsTexture, 75
AbstractShape, 27	DefaultRenderPipeline, 75
ActionKeyMapping, 28	DefendAreaTask, 77
AiBindings, 29	DirectSuicideTask, 79
AiGroupTask, 30	DirectoryReader, 78
AiTask, 35	,
AiTaskFinishedPoll, 37	ElasticImpulseGenerator, 81
AimHelperHudget, 33	ElasticImpulsor, 82
AimHelperHudgetVoxels, 34	Engine, 83
Aimer, 32	EngineBuilder, 85
ArrowHudgetVoxels, 38	EnginePower, 86
· ····································	EngineSlot, 86
BaseLuaFunction, 39	EngineSlotVoxel, 88
BaseScenario, 40	EngineTrailGenerator, 89
BattleScenario, 41	Equipment, 90
Bindings, 42	EventPoll, 91
BlitProgram, 43	EventPoller, 92
Blitter, 45	ExternalMissionBindings, 93
BoardComputer, 46	External vilosion Biranigo, 00
Bullet, 47	Faction, 94
Banot, 17	FactionMatrix, 96
Camera, 48	FactionRelation, 97
CameraDolly, 50	FightTask, 98
CameraFollowHelper, 51	FileSystem, 99
CameraHead, 52	userConfigDir, 99
Capsule, 53	FlyToTask, 100
Character, 54	FormationMemberTask, 101
CircularHudget, 55	FrameBuffer, 102
ClusterCache, 56	FrozenGameScenario, 103
ClusterLoader, 57	FuelVoxel, 105
CockpitVoxel, 57	r dervoxer, roo
CollisionDetector, 58	Game, 106
CollisionFilter, 59	update, 107
CollisionFilterIgnoringCreator, 61	GamePlay, 107
ColorCoder, 62	onEntered, 109
CommandLineParser, 63	onLeft, 109
	update, 109
CommonBindings, 63	GamePlayPaused, 110
ContextDependant, 64	onEntered, 111
ContextProvider, 66	onLeft, 111
convertToGroup	
WorldTreeNode, 301	update, 111
CrossHair, 67	GamePlayRunning, 111
CrossHairElement, 69	onEntered, 112
CrossHairVoxels, 70	onLeft, 112
CrucialVoxel, 71	update, 112
Damana Fanyandan 70	GamePlayRunningInput, 113
DamageForwarder, 72	update, 114
DamageImpact, 73	GamePlayScene, 114

306 INDEX

GamePlayScript, 116	Metrics, 162
GameScenario, 117	Mission, 163
GameState, 118	MissionScenario, 164
onEntered, 119	MissionScript, 165
onLeft, 119	MissionStatePoll, 167
update, 119	MissionSystem, 168
GarbageCollector, 120	MonoView, 169
GenericBullet, 121	moveToSubnode
GenericEngine, 122	WorldTreeNode, 301
GenericGun, 123	Movement, 170
GenericRocket, 125	Mover, 171
GenericRocketLauncher, 126	,
GenericShip, 127	ObjectHudget, 172
GenericWorldObject, 128	ObjectHudgetCornerVoxels, 173
GeometryHelper, 129	ObjectHudgetVoxels, 174
glow::Uniform< T >, 251	ObjectInfo, 175
God, 130	onEntered
GridAABB, 131	GamePlay, 109
,	GamePlayPaused, 111
Gun, 131	GamePlayRunning, 112
UMD 120	GameState, 119
HMD, 138	
HMDManager, 139	State, 231
HUD, 140	onLeft
HUDElements, 141	GamePlay, 109
HUDObjectDelegate, 147	GamePlayPaused, 111
Handle < T >, 132	GamePlayRunning, 112
HandleImpl $<$ T $>$ , 133	GameState, 119
Hardpoint, 133	State, 231
HardpointAimHelper, 135	
HardpointVoxel, 136	pathToDescendant
Hudget, 142	State, 231
HudgetAnimation, 144	PatrolWaypointsTask, 176
HudgetHideAnimation, 146	Physics, 177
	PirateScenario, 178
Impulse, 148	Player, 179
ImpulseAccumulator, 149	Point, 180
InertiaFollower, 150	PoliceFaction, 181
InputConfigWriter, 151	PostProcessingPass, 182
InputConfigurator, 150	Projectile, 184
InputMapping, 151	Property< T >, 186
InternalMissionBindings, 152	PropertyCollection< T >, 187
•	PropertyConverter, 188
KeyTrigger, 153	PropertyDirectory, 188
	PropertyImpl< T >, 190
Letter, 155	PropertyManager, 190
Line, 156	Transager, rec
LoopingTimer, 157	RandBool, 192
LuaFunction	RandFloat, 192
state, 159	RandVec3, 192
LuaFunction< N, Return, Args >, 158	Range, 193
LuaWrapper, 159	Ray, 193
LuaWrapper::_pop< 0, Ts>, 24	RenderMetaData, 194
LuaWrapper::_pop< 1, T >, 24	RenderPass, 195
LuaWrapper::_pop< size_t, Ts >, 23	RenderPipeline, 196
Luaw::_indices < ls >, 23	Rocket, 197
Luaw::_indices_builder < 0, ls>, 23	RocketLauncher, 200
Luaw::_indices_builder< N, Is >, 23	Soone 201
Moth 161	Scene, 201
Math, 161	ScreenBlitter, 202

INDEX 307

ScreenQuad, 202	Game, 107
Script, 204	GamePlay, 109
ScriptEngine, 208	GamePlayPaused, 111
Scriptable, 206	GamePlayRunning, 112
ScriptedScenario, 207	GamePlayRunningInput, 114
SecondaryInputValues, 209	GameState, 119
Ship, 209	State, 232
SimpleWayfind, 211	userConfigDir
SingleShotTimer, 212	FileSystem, 99
Size < T >, 213	•
Skybox, 214	View, 251
Sound, 215	Viewer, 252
SoundManager, 216	Viewport, 253
SoundProperties, 217	Visuals, 253
SpawnRequest, 217	Voxel, 254
SpecialVoxel, 219	VoxelAccumulator< Accumulatable >, 256
Sphere, 220	VoxelCluster, 257
SplitData, 221	VoxelClusterBounds, 258
SplitDetector, 222	VoxelCollision, 259
SplitDetector::VoxelGroup, 266	VoxelCollisionAccumulator, 260
Splitter, 223	VoxelCollisionParticipant, 261
Squad, 223	VoxelDebrisGenerator, 262
SquadBindings, 225	VoxelExplosionGenerator, 263
SquadLogic, 226	VoxelFont, 264
•	VoxelGridCmp< highPriorityAxis, middlePriorityAxis,
Starfield, 228 Starfield: Comorel coation, 53	lowPriorityAxis >, 265
Starfield::CameraLocation, 53	VoxelHangman, 266
Starfield::StarData, 227	VoxelMesh, 267
State, 229	VoxelNeighbourHelper, 268
onEntered, 231	VoxelParticleData, 268
onLeft, 231	VoxelParticleEngine, 269
pathToDescendant, 231	VoxelParticleExpireCheck, 270
transit, 231	VoxelParticleFutureCheck, 271
update, 232	VoxelParticleIntersectionCheck, 272
state 450	VoxelParticleRemoveCheck, 273
LuaFunction, 159	VoxelParticleRemover, 273
std::hash< glm::ivec3 >, 137	VoxelParticleRenderer, 274
std::hash< pair< A, B > >, 138	VoxelParticleSetup, 275
StereoBlitProgram, 232	VoxelParticleSpawnBase, 276
StereoRenderInfo, 233	VoxelRenderData, 277
StereoView, 235	VoxelRenderer, 279
StereoViewEye, 236	VoxelTree, 280
StreamRedirect, 237	VoxelTreeNode, 281
TAABB< T >, 238	VoxelTreeQuery, 282
TargetSelector, 239	voxernee Query, 202
TextFieldHudget, 240	Weapon, 283
TextFieldHudgetVoxels, 241	WeaponBuilder, 285
ThreadPool < T >, 242	World, 286
TimedTask, 243	WorldLogic, 287
	WorldObject, 288
Timer, 244	WorldObjectBindings, 290
Transform, 245	WorldObjectBuilder, 292
transit	WorldObjectCollision, 293
State, 231	WorldObjectComponents, 293
Transition, 246	WorldObjectOestroyedPoll, 295
Trigger, 248	WorldObjectModification, 296
Triggerable, 249	WorldObjectPair, 297
TriggeredTransition, 250	WorldObjectSlot, 298
update	WorldColectoid, 250 WorldTree, 299
- April - Apri	

308 INDEX