



OpenSceneGraph Version 2.9.6

**osgGA::**

**Reference Manual**



# Contents

---

<b>1</b>	<b>Main Page</b>	<b>1</b>
<b>2</b>	<b>Directory Documentation</b>	<b>3</b>
2.1	include/ Directory Reference . . . . .	3
2.2	src/osgGA/ Directory Reference . . . . .	4
2.3	include/osgGA/ Directory Reference . . . . .	5
2.4	src/ Directory Reference . . . . .	6
<b>3</b>	<b>Namespace Documentation</b>	<b>7</b>
3.1	osgGA Namespace Reference . . . . .	7
3.1.1	Detailed Description . . . . .	8
<b>4</b>	<b>Class Documentation</b>	<b>9</b>
4.1	AnimationPathManipulator Class Reference . . . . .	9
4.1.1	Constructor & Destructor Documentation . . . . .	11
4.1.1.1	AnimationPathManipulator . . . . .	11
4.1.1.2	AnimationPathManipulator . . . . .	11
4.1.2	Member Function Documentation . . . . .	11
4.1.2.1	className . . . . .	11
4.1.2.2	getAnimationPath . . . . .	11
4.1.2.3	getAnimationPath . . . . .	11
4.1.2.4	getInverseMatrix . . . . .	11
4.1.2.5	getMatrix . . . . .	11
4.1.2.6	getPrintOutTimingInfo . . . . .	11
4.1.2.7	getUsage . . . . .	11
4.1.2.8	handle . . . . .	11
4.1.2.9	handleFrame . . . . .	11
4.1.2.10	home . . . . .	11
4.1.2.11	home . . . . .	11
4.1.2.12	init . . . . .	11
4.1.2.13	setAnimationPath . . . . .	11
4.1.2.14	setByInverseMatrix . . . . .	11
4.1.2.15	setByMatrix . . . . .	11
4.1.2.16	setPrintOutTimingInfo . . . . .	12
4.1.2.17	valid . . . . .	12
4.1.3	Member Data Documentation . . . . .	12
4.1.3.1	_animationPath . . . . .	12
4.1.3.2	_animStartOfTimedPeriod . . . . .	12

4.1.3.3	_isPaused . . . . .	12
4.1.3.4	_matrix . . . . .	12
4.1.3.5	_numOfFramesSinceStartOfTimedPeriod . . . . .	12
4.1.3.6	_pauseTime . . . . .	12
4.1.3.7	_printOutTimingInfo . . . . .	12
4.1.3.8	_realStartOfTimedPeriod . . . . .	12
4.1.3.9	_timeOffset . . . . .	12
4.1.3.10	_timeScale . . . . .	12
4.1.3.11	_valid . . . . .	12
4.2	CameraViewSwitchManipulator Class Reference . . . . .	13
4.2.1	Member Typedef Documentation . . . . .	14
4.2.1.1	CameraViewList . . . . .	14
4.2.2	Constructor & Destructor Documentation . . . . .	14
4.2.2.1	CameraViewSwitchManipulator . . . . .	14
4.2.2.2	~CameraViewSwitchManipulator . . . . .	14
4.2.3	Member Function Documentation . . . . .	14
4.2.3.1	className . . . . .	14
4.2.3.2	getInverseMatrix . . . . .	14
4.2.3.3	getMatrix . . . . .	14
4.2.3.4	getNode . . . . .	14
4.2.3.5	getNode . . . . .	15
4.2.3.6	getUsage . . . . .	15
4.2.3.7	handle . . . . .	15
4.2.3.8	init . . . . .	15
4.2.3.9	setByInverseMatrix . . . . .	15
4.2.3.10	setByMatrix . . . . .	15
4.2.3.11	setNode . . . . .	15
4.2.4	Member Data Documentation . . . . .	15
4.2.4.1	_cameraViews . . . . .	15
4.2.4.2	_currentView . . . . .	15
4.2.4.3	_node . . . . .	15
4.3	CollectCameraViewsNodeVisitor Class Reference . . . . .	16
4.3.1	Constructor & Destructor Documentation . . . . .	16
4.3.1.1	CollectCameraViewsNodeVisitor . . . . .	16
4.3.2	Member Function Documentation . . . . .	16
4.3.2.1	apply . . . . .	16
4.3.3	Member Data Documentation . . . . .	16
4.3.3.1	_cameraViews . . . . .	16
4.4	CoordinateFrameCallback Class Reference . . . . .	17
4.4.1	Detailed Description . . . . .	17
4.4.2	Constructor & Destructor Documentation . . . . .	17

4.4.2.1	~CoordinateFrameCallback . . . . .	17
4.4.3	Member Function Documentation . . . . .	17
4.4.3.1	getCoordinateFrame . . . . .	17
4.5	DriveManipulator Class Reference . . . . .	18
4.5.1	Detailed Description . . . . .	20
4.5.2	Member Enumeration Documentation . . . . .	20
4.5.2.1	SpeedControlMode . . . . .	20
4.5.3	Constructor & Destructor Documentation . . . . .	20
4.5.3.1	DriveManipulator . . . . .	20
4.5.3.2	~DriveManipulator . . . . .	20
4.5.4	Member Function Documentation . . . . .	20
4.5.4.1	addMouseEvent . . . . .	20
4.5.4.2	calcMovement . . . . .	20
4.5.4.3	className . . . . .	20
4.5.4.4	computeHomePosition . . . . .	20
4.5.4.5	computePosition . . . . .	20
4.5.4.6	flushMouseEventStack . . . . .	20
4.5.4.7	getHeight . . . . .	21
4.5.4.8	getInverseMatrix . . . . .	21
4.5.4.9	getMatrix . . . . .	21
4.5.4.10	getModelScale . . . . .	21
4.5.4.11	getNode . . . . .	21
4.5.4.12	getNode . . . . .	21
4.5.4.13	getUsage . . . . .	21
4.5.4.14	getVelocity . . . . .	21
4.5.4.15	handle . . . . .	21
4.5.4.16	home . . . . .	21
4.5.4.17	init . . . . .	21
4.5.4.18	intersect . . . . .	21
4.5.4.19	setByInverseMatrix . . . . .	21
4.5.4.20	setByMatrix . . . . .	21
4.5.4.21	setHeight . . . . .	22
4.5.4.22	setModelScale . . . . .	22
4.5.4.23	setNode . . . . .	22
4.5.4.24	setVelocity . . . . .	22
4.5.5	Member Data Documentation . . . . .	22
4.5.5.1	_buffer . . . . .	22
4.5.5.2	_distance . . . . .	22
4.5.5.3	_eye . . . . .	22
4.5.5.4	_ga_t0 . . . . .	22
4.5.5.5	_ga_t1 . . . . .	22

4.5.5.6	_height . . . . .	22
4.5.5.7	_modelScale . . . . .	22
4.5.5.8	_node . . . . .	22
4.5.5.9	_pitch . . . . .	22
4.5.5.10	_pitchDownKeyPressed . . . . .	22
4.5.5.11	_pitchUpKeyPressed . . . . .	22
4.5.5.12	_rotation . . . . .	22
4.5.5.13	_speedMode . . . . .	22
4.5.5.14	_velocity . . . . .	22
4.6	EventQueue Class Reference . . . . .	23
4.6.1	Detailed Description . . . . .	26
4.6.2	Member Typedef Documentation . . . . .	26
4.6.2.1	Events . . . . .	26
4.6.3	Constructor & Destructor Documentation . . . . .	26
4.6.3.1	EventQueue . . . . .	26
4.6.3.2	~EventQueue . . . . .	26
4.6.4	Member Function Documentation . . . . .	26
4.6.4.1	addEvent . . . . .	26
4.6.4.2	appendEvents . . . . .	26
4.6.4.3	closeWindow . . . . .	26
4.6.4.4	closeWindow . . . . .	26
4.6.4.5	copyEvents . . . . .	26
4.6.4.6	createEvent . . . . .	26
4.6.4.7	frame . . . . .	26
4.6.4.8	getCurrentEventState . . . . .	26
4.6.4.9	getCurrentEventState . . . . .	26
4.6.4.10	getStartTick . . . . .	26
4.6.4.11	getTime . . . . .	26
4.6.4.12	getUseFixedMouseInputRange . . . . .	26
4.6.4.13	keyPress . . . . .	26
4.6.4.14	keyPress . . . . .	27
4.6.4.15	keyRelease . . . . .	27
4.6.4.16	keyRelease . . . . .	27
4.6.4.17	mouseButtonPress . . . . .	27
4.6.4.18	mouseButtonPress . . . . .	27
4.6.4.19	mouseButtonRelease . . . . .	27
4.6.4.20	mouseButtonRelease . . . . .	27
4.6.4.21	mouseDoubleButtonPress . . . . .	27
4.6.4.22	mouseDoubleButtonPress . . . . .	27
4.6.4.23	mouseMotion . . . . .	27
4.6.4.24	mouseMotion . . . . .	27

4.6.4.25	mouseScroll	27
4.6.4.26	mouseScroll	27
4.6.4.27	mouseScroll2D	27
4.6.4.28	mouseScroll2D	27
4.6.4.29	mouseWarped	28
4.6.4.30	operator=	28
4.6.4.31	penOrientation	28
4.6.4.32	penOrientation	28
4.6.4.33	penPressure	28
4.6.4.34	penPressure	28
4.6.4.35	penProximity	28
4.6.4.36	penProximity	28
4.6.4.37	quitApplication	28
4.6.4.38	quitApplication	28
4.6.4.39	setCurrentEventState	28
4.6.4.40	setEvents	28
4.6.4.41	setGraphicsContext	28
4.6.4.42	setMouseInputRange	28
4.6.4.43	setStartTick	28
4.6.4.44	setUseFixedMouseInputRange	28
4.6.4.45	takeEvents	28
4.6.4.46	userEvent	28
4.6.4.47	userEvent	28
4.6.4.48	windowResize	29
4.6.4.49	windowResize	29
4.6.5	Member Data Documentation	29
4.6.5.1	_accumulateEventState	29
4.6.5.2	_eventQueue	29
4.6.5.3	_eventQueueMutex	29
4.6.5.4	_startTick	29
4.6.5.5	_useFixedMouseInputRange	29
4.7	EventVisitor Class Reference	30
4.7.1	Detailed Description	31
4.7.2	Member Typedef Documentation	31
4.7.2.1	EventList	31
4.7.3	Constructor & Destructor Documentation	31
4.7.3.1	EventVisitor	31
4.7.3.2	~EventVisitor	31
4.7.4	Member Function Documentation	31
4.7.4.1	addEvent	31
4.7.4.2	apply	31

4.7.4.3	apply	31
4.7.4.4	apply	31
4.7.4.5	apply	31
4.7.4.6	apply	31
4.7.4.7	apply	31
4.7.4.8	apply	31
4.7.4.9	apply	31
4.7.4.10	apply	31
4.7.4.11	apply	31
4.7.4.12	getActionAdapter	31
4.7.4.13	getActionAdapter	31
4.7.4.14	getEventHandled	31
4.7.4.15	getEvents	31
4.7.4.16	getEvents	31
4.7.4.17	handle_callbacks	31
4.7.4.18	handle_callbacks_and_traverse	31
4.7.4.19	handle_geode_callbacks	31
4.7.4.20	META_NodeVisitor	31
4.7.4.21	operator=	31
4.7.4.22	removeEvent	32
4.7.4.23	reset	32
4.7.4.24	setEventHandled	32
4.7.4.25	setEvents	32
4.7.4.26	traverseGeode	32
4.7.5	Member Data Documentation	32
4.7.5.1	_accumulateEventState	32
4.7.5.2	_actionAdapter	32
4.7.5.3	_events	32
4.7.5.4	_handled	32
4.8	FlightManipulator Class Reference	33
4.8.1	Detailed Description	35
4.8.2	Member Enumeration Documentation	35
4.8.2.1	YawControlMode	35
4.8.3	Constructor & Destructor Documentation	35
4.8.3.1	FlightManipulator	35
4.8.3.2	~FlightManipulator	35
4.8.4	Member Function Documentation	35
4.8.4.1	addMouseEvent	35
4.8.4.2	calcMovement	35
4.8.4.3	className	35
4.8.4.4	computePosition	35

4.8.4.5	flushMouseEventStack . . . . .	35
4.8.4.6	getAcceleration . . . . .	35
4.8.4.7	getInverseMatrix . . . . .	35
4.8.4.8	getMatrix . . . . .	35
4.8.4.9	getModelScale . . . . .	35
4.8.4.10	getNode . . . . .	35
4.8.4.11	getNode . . . . .	36
4.8.4.12	getUsage . . . . .	36
4.8.4.13	getVelocity . . . . .	36
4.8.4.14	handle . . . . .	36
4.8.4.15	home . . . . .	36
4.8.4.16	init . . . . .	36
4.8.4.17	setAcceleration . . . . .	36
4.8.4.18	setByInverseMatrix . . . . .	36
4.8.4.19	setByMatrix . . . . .	36
4.8.4.20	setModelScale . . . . .	36
4.8.4.21	setNode . . . . .	36
4.8.4.22	setVelocity . . . . .	36
4.8.4.23	setYawControlMode . . . . .	36
4.8.5	Member Data Documentation . . . . .	37
4.8.5.1	_acceleration . . . . .	37
4.8.5.2	_distance . . . . .	37
4.8.5.3	_eye . . . . .	37
4.8.5.4	_ga_t0 . . . . .	37
4.8.5.5	_ga_t1 . . . . .	37
4.8.5.6	_modelScale . . . . .	37
4.8.5.7	_node . . . . .	37
4.8.5.8	_rotation . . . . .	37
4.8.5.9	_velocity . . . . .	37
4.8.5.10	_yawMode . . . . .	37
4.9	GUIActionAdapter Class Reference . . . . .	38
4.9.1	Detailed Description . . . . .	38
4.9.2	Constructor & Destructor Documentation . . . . .	38
4.9.2.1	~GUIActionAdapter . . . . .	38
4.9.3	Member Function Documentation . . . . .	38
4.9.3.1	asView . . . . .	38
4.9.3.2	requestContinuousUpdate . . . . .	39
4.9.3.3	requestRedraw . . . . .	39
4.9.3.4	requestWarpPointer . . . . .	39
4.10	GUIEventAdapter Class Reference . . . . .	40
4.10.1	Detailed Description . . . . .	45

4.10.2	Member Enumeration Documentation . . . . .	45
4.10.2.1	EventType . . . . .	45
4.10.2.2	KeySymbol . . . . .	45
4.10.2.3	ModKeyMask . . . . .	48
4.10.2.4	MouseButtonMask . . . . .	48
4.10.2.5	MouseYOrientation . . . . .	49
4.10.2.6	ScrollingMotion . . . . .	49
4.10.2.7	TabletPointerType . . . . .	49
4.10.3	Constructor & Destructor Documentation . . . . .	49
4.10.3.1	GUIEventAdapter . . . . .	49
4.10.3.2	GUIEventAdapter . . . . .	49
4.10.3.3	~GUIEventAdapter . . . . .	49
4.10.4	Member Function Documentation . . . . .	49
4.10.4.1	getAccumulatedEventState . . . . .	49
4.10.4.2	getButton . . . . .	49
4.10.4.3	getButtonMask . . . . .	49
4.10.4.4	getEventType . . . . .	49
4.10.4.5	getGraphicsContext . . . . .	49
4.10.4.6	getHandled . . . . .	49
4.10.4.7	getKey . . . . .	49
4.10.4.8	getModKeyMask . . . . .	50
4.10.4.9	getMouseYOrientation . . . . .	50
4.10.4.10	getPenOrientation . . . . .	50
4.10.4.11	getPenPressure . . . . .	50
4.10.4.12	getPenRotation . . . . .	50
4.10.4.13	getPenTiltX . . . . .	50
4.10.4.14	getPenTiltY . . . . .	50
4.10.4.15	getScrollingDeltaX . . . . .	50
4.10.4.16	getScrollingDeltaY . . . . .	50
4.10.4.17	getScrollingMotion . . . . .	50
4.10.4.18	getTabletPointerType . . . . .	50
4.10.4.19	getTime . . . . .	50
4.10.4.20	getWindowHeight . . . . .	50
4.10.4.21	getWindowWidth . . . . .	50
4.10.4.22	getWindowX . . . . .	50
4.10.4.23	getWindowY . . . . .	50
4.10.4.24	getX . . . . .	50
4.10.4.25	getXmax . . . . .	50
4.10.4.26	getXmin . . . . .	50
4.10.4.27	getXnormalized . . . . .	51
4.10.4.28	getY . . . . .	51

4.10.4.29	getYmax	51
4.10.4.30	getYmin	51
4.10.4.31	getYnormalized	51
4.10.4.32	META_Object	51
4.10.4.33	setButton	51
4.10.4.34	setButtonMask	51
4.10.4.35	setEventType	51
4.10.4.36	setGraphicsContext	51
4.10.4.37	setHandled	51
4.10.4.38	setInputRange	51
4.10.4.39	setKey	51
4.10.4.40	setModKeyMask	51
4.10.4.41	setMouseYOrientation	51
4.10.4.42	setPenPressure	51
4.10.4.43	setPenRotation	51
4.10.4.44	setPenTiltX	51
4.10.4.45	setPenTiltY	51
4.10.4.46	setScrollingMotion	52
4.10.4.47	setScrollingMotionDelta	52
4.10.4.48	setTabletPointerType	52
4.10.4.49	setTime	52
4.10.4.50	setWindowRectangle	52
4.10.4.51	setX	52
4.10.4.52	setY	52
4.10.4.53	time	52
4.10.5	Member Data Documentation	52
4.10.5.1	_button	52
4.10.5.2	_buttonMask	52
4.10.5.3	_context	52
4.10.5.4	_eventType	52
4.10.5.5	_handled	52
4.10.5.6	_key	52
4.10.5.7	_modKeyMask	52
4.10.5.8	_mouseYOrientation	52
4.10.5.9	_mx	52
4.10.5.10	_my	52
4.10.5.11	_scrolling	52
4.10.5.12	_tabletPen	52
4.10.5.13	_time	52
4.10.5.14	_windowHeight	52
4.10.5.15	_windowWidth	52

4.10.5.16	_windowX . . . . .	52
4.10.5.17	_windowY . . . . .	52
4.10.5.18	_Xmax . . . . .	52
4.10.5.19	_Xmin . . . . .	52
4.10.5.20	_Ymax . . . . .	52
4.10.5.21	_Ymin . . . . .	52
4.11	GUIEventHandler Class Reference . . . . .	54
4.11.1	Detailed Description . . . . .	55
4.11.2	Constructor & Destructor Documentation . . . . .	55
4.11.2.1	GUIEventHandler . . . . .	55
4.11.2.2	GUIEventHandler . . . . .	55
4.11.3	Member Function Documentation . . . . .	55
4.11.3.1	event . . . . .	55
4.11.3.2	getIgnoreHandledEventsMask . . . . .	55
4.11.3.3	getUsage . . . . .	55
4.11.3.4	handle . . . . .	55
4.11.3.5	handle . . . . .	55
4.11.3.6	handleWithCheckAgainstIgnoreHandledEventsMask . . . . .	55
4.11.3.7	handleWithCheckAgainstIgnoreHandledEventsMask . . . . .	56
4.11.3.8	META_Object . . . . .	56
4.11.3.9	operator() . . . . .	56
4.11.3.10	setIgnoreHandledEventsMask . . . . .	56
4.11.4	Member Data Documentation . . . . .	56
4.11.4.1	_ignoreHandledEventsMask . . . . .	56
4.12	KeySwitchMatrixManipulator Class Reference . . . . .	57
4.12.1	Detailed Description . . . . .	59
4.12.2	Member Typedef Documentation . . . . .	59
4.12.2.1	KeyManipMap . . . . .	59
4.12.2.2	NamedManipulator . . . . .	59
4.12.3	Member Function Documentation . . . . .	59
4.12.3.1	addMatrixManipulator . . . . .	59
4.12.3.2	addNumberedMatrixManipulator . . . . .	59
4.12.3.3	className . . . . .	59
4.12.3.4	computeHomePosition . . . . .	59
4.12.3.5	getCurrentMatrixManipulator . . . . .	59
4.12.3.6	getCurrentMatrixManipulator . . . . .	59
4.12.3.7	getFusionDistanceMode . . . . .	60
4.12.3.8	getFusionDistanceValue . . . . .	60
4.12.3.9	getInverseMatrix . . . . .	60
4.12.3.10	getKeyManipMap . . . . .	60
4.12.3.11	getKeyManipMap . . . . .	60

4.12.3.12	getMatrix	60
4.12.3.13	getMatrixManipulatorWithIndex	60
4.12.3.14	getMatrixManipulatorWithIndex	60
4.12.3.15	getMatrixManipulatorWithKey	60
4.12.3.16	getMatrixManipulatorWithKey	60
4.12.3.17	getNode	60
4.12.3.18	getNode	60
4.12.3.19	getNumMatrixManipulators	60
4.12.3.20	getUsage	60
4.12.3.21	handle	60
4.12.3.22	home	61
4.12.3.23	init	61
4.12.3.24	selectMatrixManipulator	61
4.12.3.25	setAutoComputeHomePosition	61
4.12.3.26	setByInverseMatrix	61
4.12.3.27	setByMatrix	61
4.12.3.28	setCoordinateFrameCallback	61
4.12.3.29	setHomePosition	61
4.12.3.30	setMinimumDistance	61
4.12.3.31	setNode	61
4.13	MatrixManipulator Class Reference	62
4.13.1	Detailed Description	64
4.13.2	Constructor & Destructor Documentation	64
4.13.2.1	MatrixManipulator	64
4.13.2.2	~MatrixManipulator	64
4.13.3	Member Function Documentation	64
4.13.3.1	className	64
4.13.3.2	computeHomePosition	64
4.13.3.3	getAutoComputeHomePosition	64
4.13.3.4	getCoordinateFrame	64
4.13.3.5	getCoordinateFrameCallback	64
4.13.3.6	getCoordinateFrameCallback	64
4.13.3.7	getFrontVector	64
4.13.3.8	getFusionDistanceMode	64
4.13.3.9	getFusionDistanceValue	65
4.13.3.10	getHomePosition	65
4.13.3.11	getIntersectTraversalMask	65
4.13.3.12	getInverseMatrix	65
4.13.3.13	getMatrix	65
4.13.3.14	getMinimumDistance	65
4.13.3.15	getNode	65

4.13.3.16	getNode . . . . .	65
4.13.3.17	getSideVector . . . . .	65
4.13.3.18	getUpVector . . . . .	65
4.13.3.19	handle . . . . .	65
4.13.3.20	home . . . . .	66
4.13.3.21	home . . . . .	66
4.13.3.22	init . . . . .	66
4.13.3.23	setAutoComputeHomePosition . . . . .	66
4.13.3.24	setByInverseMatrix . . . . .	66
4.13.3.25	setByMatrix . . . . .	66
4.13.3.26	setCoordinateFrameCallback . . . . .	66
4.13.3.27	setHomePosition . . . . .	66
4.13.3.28	setIntersectTraversalMask . . . . .	66
4.13.3.29	setMinimumDistance . . . . .	67
4.13.3.30	setNode . . . . .	67
4.13.4	Member Data Documentation . . . . .	67
4.13.4.1	_autoComputeHomePosition . . . . .	67
4.13.4.2	_coordinateFrameCallback . . . . .	67
4.13.4.3	_homeCenter . . . . .	67
4.13.4.4	_homeEye . . . . .	67
4.13.4.5	_homeUp . . . . .	67
4.13.4.6	_intersectTraversalMask . . . . .	67
4.13.4.7	_minimumDistance . . . . .	67
4.14	NodeTrackerManipulator Class Reference . . . . .	68
4.14.1	Member Typedef Documentation . . . . .	70
4.14.1.1	ObserverNodePath . . . . .	70
4.14.2	Member Enumeration Documentation . . . . .	70
4.14.2.1	RotationMode . . . . .	70
4.14.2.2	TrackerMode . . . . .	70
4.14.3	Constructor & Destructor Documentation . . . . .	70
4.14.3.1	NodeTrackerManipulator . . . . .	70
4.14.3.2	~NodeTrackerManipulator . . . . .	70
4.14.4	Member Function Documentation . . . . .	70
4.14.4.1	addMouseEvent . . . . .	70
4.14.4.2	calcMovement . . . . .	71
4.14.4.3	clampOrientation . . . . .	71
4.14.4.4	className . . . . .	71
4.14.4.5	computeHomePosition . . . . .	71
4.14.4.6	computeNodeCenterAndRotation . . . . .	71
4.14.4.7	computeNodeLocalToWorld . . . . .	71
4.14.4.8	computeNodeWorldToLocal . . . . .	71

4.14.4.9	computePosition	71
4.14.4.10	flushMouseEventStack	71
4.14.4.11	getFusionDistanceMode	71
4.14.4.12	getFusionDistanceValue	71
4.14.4.13	getInverseMatrix	71
4.14.4.14	getMatrix	71
4.14.4.15	getNode	71
4.14.4.16	getNode	71
4.14.4.17	getNodePath	72
4.14.4.18	getRotationMode	72
4.14.4.19	getTrackerMode	72
4.14.4.20	getTrackNode	72
4.14.4.21	getTrackNode	72
4.14.4.22	getTrackNodePath	72
4.14.4.23	getUsage	72
4.14.4.24	handle	72
4.14.4.25	home	72
4.14.4.26	init	72
4.14.4.27	isMouseMoving	72
4.14.4.28	setByInverseMatrix	72
4.14.4.29	setByMatrix	72
4.14.4.30	setNode	72
4.14.4.31	setRotationMode	73
4.14.4.32	setTrackerMode	73
4.14.4.33	setTrackNode	73
4.14.4.34	setTrackNodePath	73
4.14.4.35	setTrackNodePath	73
4.14.4.36	tb_project_to_sphere	73
4.14.4.37	trackball	73
4.14.4.38	validateNodePath	73
4.14.5	Member Data Documentation	73
4.14.5.1	_distance	73
4.14.5.2	_ga_t0	73
4.14.5.3	_ga_t1	73
4.14.5.4	_node	73
4.14.5.5	_nodeRotation	73
4.14.5.6	_rotation	73
4.14.5.7	_rotationMode	73
4.14.5.8	_thrown	73
4.14.5.9	_trackerMode	73
4.14.5.10	_trackNodePath	73

4.15	Scrolling Struct Reference . . . . .	74
4.15.1	Constructor & Destructor Documentation . . . . .	74
4.15.1.1	Scrolling . . . . .	74
4.15.1.2	Scrolling . . . . .	74
4.15.2	Member Data Documentation . . . . .	74
4.15.2.1	deltaX . . . . .	74
4.15.2.2	deltaY . . . . .	74
4.15.2.3	motion . . . . .	74
4.16	SphericalManipulator Class Reference . . . . .	75
4.16.1	Member Enumeration Documentation . . . . .	78
4.16.1.1	RotationMode . . . . .	78
4.16.2	Constructor & Destructor Documentation . . . . .	78
4.16.2.1	SphericalManipulator . . . . .	78
4.16.2.2	~SphericalManipulator . . . . .	78
4.16.3	Member Function Documentation . . . . .	78
4.16.3.1	addMouseEvent . . . . .	78
4.16.3.2	calcMovement . . . . .	78
4.16.3.3	className . . . . .	78
4.16.3.4	computeHomePosition . . . . .	78
4.16.3.5	computeViewPosition . . . . .	78
4.16.3.6	flushMouseEventStack . . . . .	78
4.16.3.7	getAllowThrow . . . . .	78
4.16.3.8	getCenter . . . . .	78
4.16.3.9	getDistance . . . . .	78
4.16.3.10	getElevation . . . . .	78
4.16.3.11	getFusionDistanceMode . . . . .	78
4.16.3.12	getFusionDistanceValue . . . . .	78
4.16.3.13	getHeading . . . . .	79
4.16.3.14	getHomeDistance . . . . .	79
4.16.3.15	getInverseMatrix . . . . .	79
4.16.3.16	getMatrix . . . . .	79
4.16.3.17	getMinimumZoomScale . . . . .	79
4.16.3.18	getNode . . . . .	79
4.16.3.19	getNode . . . . .	79
4.16.3.20	getRotationMode . . . . .	79
4.16.3.21	getScroolWheelZoomDelta . . . . .	79
4.16.3.22	getUsage . . . . .	79
4.16.3.23	handle . . . . .	79
4.16.3.24	home . . . . .	79
4.16.3.25	home . . . . .	79
4.16.3.26	init . . . . .	79

4.16.3.27	isMouseMoving . . . . .	79
4.16.3.28	setAllowThrow . . . . .	80
4.16.3.29	setByInverseMatrix . . . . .	80
4.16.3.30	setByMatrix . . . . .	80
4.16.3.31	setCenter . . . . .	80
4.16.3.32	setDistance . . . . .	80
4.16.3.33	setElevation . . . . .	80
4.16.3.34	setHeading . . . . .	80
4.16.3.35	setMinimumZoomScale . . . . .	80
4.16.3.36	setNode . . . . .	80
4.16.3.37	setRotationMode . . . . .	80
4.16.3.38	setScroolWheelZoomDelta . . . . .	80
4.16.3.39	zoomOn . . . . .	80
4.16.4	Member Data Documentation . . . . .	80
4.16.4.1	_allowThrow . . . . .	80
4.16.4.2	_center . . . . .	80
4.16.4.3	_delta_frame_time . . . . .	80
4.16.4.4	_distance . . . . .	80
4.16.4.5	_elevation . . . . .	80
4.16.4.6	_ga_t0 . . . . .	80
4.16.4.7	_ga_t1 . . . . .	80
4.16.4.8	_heading . . . . .	80
4.16.4.9	_homeDistance . . . . .	80
4.16.4.10	_last_frame_time . . . . .	80
4.16.4.11	_minimumZoomScale . . . . .	81
4.16.4.12	_modelScale . . . . .	81
4.16.4.13	_node . . . . .	81
4.16.4.14	_rotationMode . . . . .	81
4.16.4.15	_thrown . . . . .	81
4.16.4.16	_zoomDelta . . . . .	81
4.17	StateSetManipulator Class Reference . . . . .	82
4.17.1	Detailed Description . . . . .	83
4.17.2	Constructor & Destructor Documentation . . . . .	84
4.17.2.1	StateSetManipulator . . . . .	84
4.17.2.2	~StateSetManipulator . . . . .	84
4.17.3	Member Function Documentation . . . . .	84
4.17.3.1	className . . . . .	84
4.17.3.2	clone . . . . .	84
4.17.3.3	cyclePolygonMode . . . . .	84
4.17.3.4	getBackfaceEnabled . . . . .	84
4.17.3.5	getKeyEventCyclePolygonMode . . . . .	84

4.17.3.6	getKeyEventToggleBackfaceCulling	84
4.17.3.7	getKeyEventToggleLighting	84
4.17.3.8	getKeyEventToggleTexturing	84
4.17.3.9	getLightingEnabled	84
4.17.3.10	getMaximumNumOfTextureUnits	84
4.17.3.11	getOrCreatePolygonMode	84
4.17.3.12	getPolygonMode	84
4.17.3.13	getStateSet	84
4.17.3.14	getStateSet	84
4.17.3.15	getTextureEnabled	84
4.17.3.16	getUsage	84
4.17.3.17	handle	84
4.17.3.18	setBackfaceEnabled	84
4.17.3.19	setKeyEventCyclePolygonMode	84
4.17.3.20	setKeyEventToggleBackfaceCulling	84
4.17.3.21	setKeyEventToggleLighting	84
4.17.3.22	setKeyEventToggleTexturing	84
4.17.3.23	setLightingEnabled	84
4.17.3.24	setMaximumNumOfTextureUnits	84
4.17.3.25	setPolygonMode	84
4.17.3.26	setStateSet	84
4.17.3.27	setTextureEnabled	85
4.17.4	Member Data Documentation	85
4.17.4.1	_backface	85
4.17.4.2	_initialized	85
4.17.4.3	_keyEventCyclePolygonMode	85
4.17.4.4	_keyEventToggleBackfaceCulling	85
4.17.4.5	_keyEventToggleLighting	85
4.17.4.6	_keyEventToggleTexturing	85
4.17.4.7	_lighting	85
4.17.4.8	_maxNumOfTextureUnits	85
4.17.4.9	_stateset	85
4.17.4.10	_texture	85
4.18	TabletPen Struct Reference	86
4.18.1	Constructor & Destructor Documentation	86
4.18.1.1	TabletPen	86
4.18.1.2	TabletPen	86
4.18.2	Member Data Documentation	86
4.18.2.1	pressure	86
4.18.2.2	rotation	86
4.18.2.3	tabletPointerType	86

4.18.2.4	tiltX	86
4.18.2.5	tiltY	86
4.19	TerrainManipulator Class Reference	87
4.19.1	Member Enumeration Documentation	89
4.19.1.1	RotationMode	89
4.19.2	Constructor & Destructor Documentation	89
4.19.2.1	TerrainManipulator	89
4.19.2.2	~TerrainManipulator	89
4.19.3	Member Function Documentation	89
4.19.3.1	addMouseEvent	89
4.19.3.2	calcMovement	89
4.19.3.3	clampOrientation	89
4.19.3.4	className	89
4.19.3.5	computePosition	89
4.19.3.6	flushMouseEventStack	89
4.19.3.7	getFusionDistanceMode	89
4.19.3.8	getFusionDistanceValue	89
4.19.3.9	getInverseMatrix	89
4.19.3.10	getMatrix	90
4.19.3.11	getNode	90
4.19.3.12	getNode	90
4.19.3.13	getRotationMode	90
4.19.3.14	getUsage	90
4.19.3.15	handle	90
4.19.3.16	home	90
4.19.3.17	init	90
4.19.3.18	intersect	90
4.19.3.19	isMouseMoving	90
4.19.3.20	setByInverseMatrix	90
4.19.3.21	setByMatrix	90
4.19.3.22	setNode	90
4.19.3.23	setRotationMode	91
4.19.3.24	tb_project_to_sphere	91
4.19.3.25	trackball	91
4.19.4	Member Data Documentation	91
4.19.4.1	_center	91
4.19.4.2	_distance	91
4.19.4.3	_ga_t0	91
4.19.4.4	_ga_t1	91
4.19.4.5	_node	91
4.19.4.6	_previousUp	91

4.19.4.7	_rotation . . . . .	91
4.19.4.8	_rotationMode . . . . .	91
4.19.4.9	_thrown . . . . .	91
4.20	TrackballManipulator Class Reference . . . . .	92
4.20.1	Constructor & Destructor Documentation . . . . .	95
4.20.1.1	TrackballManipulator . . . . .	95
4.20.1.2	~TrackballManipulator . . . . .	95
4.20.2	Member Function Documentation . . . . .	95
4.20.2.1	addMouseEvent . . . . .	95
4.20.2.2	calcMovement . . . . .	95
4.20.2.3	className . . . . .	95
4.20.2.4	computePosition . . . . .	95
4.20.2.5	flushMouseEventStack . . . . .	95
4.20.2.6	getAllowThrow . . . . .	95
4.20.2.7	getCenter . . . . .	95
4.20.2.8	getDistance . . . . .	95
4.20.2.9	getFusionDistanceMode . . . . .	95
4.20.2.10	getFusionDistanceValue . . . . .	95
4.20.2.11	getInverseMatrix . . . . .	95
4.20.2.12	getMatrix . . . . .	95
4.20.2.13	getMinimumZoomScale . . . . .	95
4.20.2.14	getNode . . . . .	95
4.20.2.15	getNode . . . . .	96
4.20.2.16	getRotation . . . . .	96
4.20.2.17	getScroolWheelZoomDelta . . . . .	96
4.20.2.18	getTrackballSize . . . . .	96
4.20.2.19	getUsage . . . . .	96
4.20.2.20	handle . . . . .	96
4.20.2.21	home . . . . .	96
4.20.2.22	home . . . . .	96
4.20.2.23	init . . . . .	96
4.20.2.24	isMouseMoving . . . . .	96
4.20.2.25	setAllowThrow . . . . .	96
4.20.2.26	setByInverseMatrix . . . . .	96
4.20.2.27	setByMatrix . . . . .	96
4.20.2.28	setCenter . . . . .	96
4.20.2.29	setDistance . . . . .	97
4.20.2.30	setMinimumZoomScale . . . . .	97
4.20.2.31	setNode . . . . .	97
4.20.2.32	setRotation . . . . .	97
4.20.2.33	setScroolWheelZoomDelta . . . . .	97

4.20.2.34	setTrackballSize	97
4.20.2.35	tb_project_to_sphere	97
4.20.2.36	trackball	97
4.20.3	Member Data Documentation	97
4.20.3.1	_allowThrow	97
4.20.3.2	_center	97
4.20.3.3	_delta_frame_time	97
4.20.3.4	_distance	97
4.20.3.5	_ga_t0	97
4.20.3.6	_ga_t1	97
4.20.3.7	_last_frame_time	97
4.20.3.8	_minimumZoomScale	97
4.20.3.9	_modelScale	97
4.20.3.10	_node	97
4.20.3.11	_rotation	97
4.20.3.12	_thrown	97
4.20.3.13	_trackballSize	97
4.20.3.14	_zoomDelta	97
4.21	UFOManipulator Class Reference	99
4.21.1	Detailed Description	101
4.21.2	Constructor & Destructor Documentation	102
4.21.2.1	UFOManipulator	102
4.21.2.2	~UFOManipulator	102
4.21.3	Member Function Documentation	102
4.21.3.1	_adjustPosition	102
4.21.3.2	_frame	102
4.21.3.3	_keyDown	102
4.21.3.4	_keyUp	102
4.21.3.5	_stop	102
4.21.3.6	className	102
4.21.3.7	computeHomePosition	102
4.21.3.8	getCurrentPositionAsLookAt	102
4.21.3.9	getForwardSpeed	102
4.21.3.10	getInverseMatrix	102
4.21.3.11	getMatrix	102
4.21.3.12	getMinDistance	102
4.21.3.13	getMinHeight	102
4.21.3.14	getNode	102
4.21.3.15	getNode	103
4.21.3.16	getRotationSpeed	103
4.21.3.17	getSideSpeed	103

4.21.3.18	getUsage . . . . .	103
4.21.3.19	handle . . . . .	103
4.21.3.20	home . . . . .	103
4.21.3.21	home . . . . .	103
4.21.3.22	init . . . . .	103
4.21.3.23	intersect . . . . .	103
4.21.3.24	setByInverseMatrix . . . . .	103
4.21.3.25	setByMatrix . . . . .	104
4.21.3.26	setForwardSpeed . . . . .	104
4.21.3.27	setMinDistance . . . . .	104
4.21.3.28	setMinHeight . . . . .	104
4.21.3.29	setNode . . . . .	104
4.21.3.30	setRotationSpeed . . . . .	105
4.21.3.31	setSideSpeed . . . . .	105
4.21.4	Member Data Documentation . . . . .	105
4.21.4.1	_ctrl . . . . .	105
4.21.4.2	_decelerateOffsetRate . . . . .	105
4.21.4.3	_decelerateUpSideRate . . . . .	105
4.21.4.4	_direction . . . . .	105
4.21.4.5	_directionRotationAcceleration . . . . .	105
4.21.4.6	_directionRotationDeceleration . . . . .	105
4.21.4.7	_directionRotationEpsilon . . . . .	105
4.21.4.8	_directionRotationRate . . . . .	105
4.21.4.9	_dt . . . . .	105
4.21.4.10	_forwardSpeed . . . . .	105
4.21.4.11	_inverseMatrix . . . . .	105
4.21.4.12	_matrix . . . . .	105
4.21.4.13	_minDistanceInFront . . . . .	105
4.21.4.14	_minHeightAboveGround . . . . .	105
4.21.4.15	_node . . . . .	105
4.21.4.16	_offset . . . . .	105
4.21.4.17	_pitchOffset . . . . .	105
4.21.4.18	_pitchOffsetRate . . . . .	105
4.21.4.19	_position . . . . .	105
4.21.4.20	_shift . . . . .	105
4.21.4.21	_sideSpeed . . . . .	105
4.21.4.22	_speedAccelerationFactor . . . . .	105
4.21.4.23	_speedDecelerationFactor . . . . .	105
4.21.4.24	_speedEpsilon . . . . .	105
4.21.4.25	_straightenOffset . . . . .	105
4.21.4.26	_t0 . . . . .	105

4.21.4.27	_upSpeed . . . . .	105
4.21.4.28	_viewAngle . . . . .	105
4.21.4.29	_viewOffsetDelta . . . . .	105
4.21.4.30	_yawOffset . . . . .	105
4.21.4.31	_yawOffsetRate . . . . .	105
<b>5</b>	<b>File Documentation</b>	<b>107</b>
5.1	AnimationPathManipulator File Reference . . . . .	107
5.1.1	Define Documentation . . . . .	107
5.1.1.1	OSGGA_ANIMATION_PATH_MANIPULATOR . . . . .	107
5.2	AnimationPathManipulator.cpp File Reference . . . . .	108
5.3	CameraViewSwitchManipulator File Reference . . . . .	109
5.3.1	Define Documentation . . . . .	109
5.3.1.1	OSGGA_VIEWLISTMANIPULATOR . . . . .	109
5.4	CameraViewSwitchManipulator.cpp File Reference . . . . .	110
5.5	DriveManipulator File Reference . . . . .	111
5.5.1	Define Documentation . . . . .	111
5.5.1.1	OSGGA_DRIVEMANIPULATOR . . . . .	111
5.6	DriveManipulator.cpp File Reference . . . . .	112
5.6.1	Define Documentation . . . . .	112
5.6.1.1	DRIVER_HEIGHT . . . . .	112
5.6.1.2	KEYBOARD_PITCH . . . . .	112
5.6.2	Function Documentation . . . . .	112
5.6.2.1	getHeightOfDriver . . . . .	112
5.7	EventQueue File Reference . . . . .	113
5.7.1	Define Documentation . . . . .	113
5.7.1.1	OSGGA_EVENTQUEUE . . . . .	113
5.8	EventQueue.cpp File Reference . . . . .	114
5.9	EventVisitor File Reference . . . . .	115
5.9.1	Define Documentation . . . . .	115
5.9.1.1	OSGGA_EVENTVISITOR . . . . .	115
5.10	EventVisitor.cpp File Reference . . . . .	116
5.11	Export File Reference . . . . .	117
5.11.1	Define Documentation . . . . .	117
5.11.1.1	OSGGA_EXPORT . . . . .	117
5.11.1.2	OSGGA_EXPORT_ . . . . .	117
5.12	FlightManipulator File Reference . . . . .	118
5.12.1	Define Documentation . . . . .	118
5.12.1.1	OSGGA_FLIGHTMANIPULATOR . . . . .	118
5.13	FlightManipulator.cpp File Reference . . . . .	119
5.14	GUIActionAdapter File Reference . . . . .	120

5.14.1	Define Documentation . . . . .	120
5.14.1.1	OSGGA_GUIACTIONADAPTER . . . . .	120
5.15	GUIEventAdapter File Reference . . . . .	121
5.15.1	Define Documentation . . . . .	121
5.15.1.1	OSGGA_EVENT . . . . .	121
5.16	GUIEventAdapter.cpp File Reference . . . . .	122
5.17	GUIEventHandler File Reference . . . . .	123
5.17.1	Define Documentation . . . . .	123
5.17.1.1	OSGGA_GUIEVENTHANDLER . . . . .	123
5.18	GUIEventHandler.cpp File Reference . . . . .	124
5.19	KeySwitchMatrixManipulator File Reference . . . . .	125
5.19.1	Define Documentation . . . . .	125
5.19.1.1	OSGUTIL_KEYSWITCMATRIXMANIPULATOR . . . . .	125
5.20	KeySwitchMatrixManipulator.cpp File Reference . . . . .	126
5.21	mainpage.h File Reference . . . . .	127
5.21.1	Detailed Description . . . . .	127
5.22	MatrixManipulator File Reference . . . . .	128
5.22.1	Define Documentation . . . . .	128
5.22.1.1	NEW_HOME_POSITION . . . . .	128
5.22.1.2	OSGGA_MatrixManipulator . . . . .	128
5.23	MatrixManipulator.cpp File Reference . . . . .	129
5.24	NodeTrackerManipulator File Reference . . . . .	130
5.24.1	Define Documentation . . . . .	130
5.24.1.1	OSGGA_NODETRACKERMANIPULATOR . . . . .	130
5.25	NodeTrackerManipulator.cpp File Reference . . . . .	131
5.25.1	Variable Documentation . . . . .	131
5.25.1.1	TRACKBALLSIZE . . . . .	131
5.26	SphericalManipulator File Reference . . . . .	132
5.27	SphericalManipulator.cpp File Reference . . . . .	133
5.28	StateSetManipulator File Reference . . . . .	134
5.28.1	Define Documentation . . . . .	134
5.28.1.1	OSGGA_STATESET_MANIPULATOR . . . . .	134
5.29	StateSetManipulator.cpp File Reference . . . . .	135
5.30	TerrainManipulator File Reference . . . . .	136
5.30.1	Define Documentation . . . . .	136
5.30.1.1	OSGGA_TERRAINMANIPULATOR . . . . .	136
5.31	TerrainManipulator.cpp File Reference . . . . .	137
5.31.1	Variable Documentation . . . . .	137
5.31.1.1	TRACKBALLSIZE . . . . .	137
5.32	TrackballManipulator File Reference . . . . .	138
5.32.1	Define Documentation . . . . .	138

5.32.1.1	OSGGA_TRACKBALLMANIPULATOR . . . . .	138
5.33	TrackballManipulator.cpp File Reference . . . . .	139
5.34	UFOManipulator File Reference . . . . .	140
5.34.1	Define Documentation . . . . .	140
5.34.1.1	OSGGA_UFO_MANIPULATOR_DEF . . . . .	140
5.35	UFOManipulator.cpp File Reference . . . . .	141
5.35.1	Define Documentation . . . . .	141
5.35.1.1	M_PI . . . . .	141
5.36	Version File Reference . . . . .	142
5.36.1	Define Documentation . . . . .	142
5.36.1.1	OSGGA_VERSION . . . . .	142
5.36.2	Function Documentation . . . . .	142
5.36.2.1	osgGAGetLibraryName . . . . .	142
5.36.2.2	osgGAGetVersion . . . . .	142
5.37	Version.cpp File Reference . . . . .	143
5.37.1	Function Documentation . . . . .	143
5.37.1.1	osgGAGetLibraryName . . . . .	143
5.37.1.2	osgGAGetVersion . . . . .	143



## Main Page

---

The OpenSceneGraph exists as a number of modules, each sitting in its own library, enclosed within its own namespace. At the very core lies the osg library. This contains the OpenSceneGraph's central classes and, at the bare minimum, it is all users need to write an OpenSceneGraph program in C++.

Around and alongside this sit other supporting libraries, such as osgUtil (containing visitors for app traversals, cull traversals, scene graph optimizers and so on), osgDB (for handling plug-ins, shared library loading, database reading and writing and the like), osgText, osgParticle, etc.

Extensive online documentation is available from the OSG Support section to help in using Open Scene Graph.

The project's original reference guides generated by Doxygen from the source code may be downloaded as a single file from the OSG Reference Guides section.

To download source code, binaries, dependencies and sample datasets visit the OSG Download page.

For more about dependencies see the OSG Dependencies page.

The documentation you are looking at can be downloaded from [www.3draum.ch](http://www.3draum.ch).

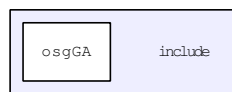
Enjoy!



# Directory Documentation

---

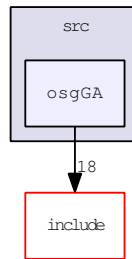
## 2.1 include/ Directory Reference



### Directories

- directory **osgGA**

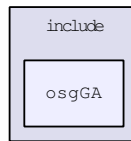
## 2.2 src/osgGA/ Directory Reference



### Files

- file **AnimationPathManipulator.cpp**
- file **CameraViewSwitchManipulator.cpp**
- file **DriveManipulator.cpp**
- file **EventQueue.cpp**
- file **EventVisitor.cpp**
- file **FlightManipulator.cpp**
- file **GUIEventAdapter.cpp**
- file **GUIEventHandler.cpp**
- file **KeySwitchMatrixManipulator.cpp**
- file **MatrixManipulator.cpp**
- file **NodeTrackerManipulator.cpp**
- file **SphericalManipulator.cpp**
- file **StateSetManipulator.cpp**
- file **TerrainManipulator.cpp**
- file **TrackballManipulator.cpp**
- file **UFOManipulator.cpp**
- file **Version.cpp**

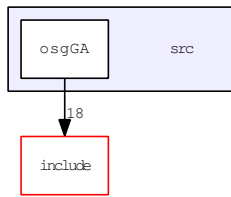
## 2.3 include/osgGA/ Directory Reference



### Files

- file **AnimationPathManipulator**
- file **CameraViewSwitchManipulator**
- file **DriveManipulator**
- file **EventQueue**
- file **EventVisitor**
- file **Export**
- file **FlightManipulator**
- file **GUIActionAdapter**
- file **GUIEventAdapter**
- file **GUIEventHandler**
- file **KeySwitchMatrixManipulator**
- file **mainpage.h**
- file **MatrixManipulator**
- file **NodeTrackerManipulator**
- file **SphericalManipulator**
- file **StateSetManipulator**
- file **TerrainManipulator**
- file **TrackballManipulator**
- file **UFOManipulator**
- file **Version**

## 2.4 src/ Directory Reference



### Directories

- directory **osgGA**

## Namespace Documentation

---

### 3.1 osgGA Namespace Reference

The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.

#### Classes

- class **AnimationPathManipulator**
- class **CameraViewSwitchManipulator**
- class **DriveManipulator**  
*DriveManipulator* (p. 18) is a camera manipulator which provides drive-like functionality.
- class **EventQueue**  
*EventQueue* (p. 23) implementation for collecting and adapting windowing events.
- class **EventVisitor**  
*Basic EventVisitor* (p. 30) implementation for animating a scene.
- class **FlightManipulator**  
*FlightManipulator* (p. 33) is a **MatrixManipulator** (p. 62) which provides flight simulator-like updating of the camera position & orientation.
- class **GUIActionAdapter**  
*Abstract base class defining the interface by which GUIEventHandlers may request actions of the GUI system in use.*
- class **GUIEventAdapter**  
*Event class for storing Keyboard, mouse and window events.*
- class **GUIEventHandler**  
*GUIEventHandler* (p. 54) provides a basic interface for any class which wants to handle a GUI Events.
- class **KeySwitchMatrixManipulator**  
*KeySwitchMatrixManipulator* (p. 57) is a decorator which allows the type of camera manipulator being used to be switched by pressing a key.
- class **MatrixManipulator**  
*MatrixManipulator* (p. 62) is an abstract base class defining the interface, and a certain amount of default functionality, for classes which wish to control OSG cameras in response to GUI events.
- class **NodeTrackerManipulator**
- class **SphericalManipulator**
- class **StateSetManipulator**  
*Experimental class, not been looked at for a while, but which will be returned to at some point :-\.*

- class **TerrainManipulator**
- class **TrackballManipulator**
- class **UFOManipulator**

*A UFO manipulator driven with keybindings.*

### 3.1.1 Detailed Description

The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems. As a cross-platform, window system-agnostic class library, the OpenSceneGraph has no direct ties to any given windowing environment. Viewers, however, must at some level interact with a window system - where Window system may refer to a windowing API, e.g. GLUT, Qt, FLTK, MFC, ...

There is much commonality in the implementation of Viewers for varying windowing environments. E.g. most Viewers will update a Camera position in response to a mouse event, and may request that a timer be started as a result of a model being 'spun'.

The purpose of the **osgGA** (p. 7) namespace is to centralise the common areas of this functionality. The viewer writer needs then only write a **GUIEventAdapter** (p. 40), a **GUIActionAdapter** (p. 38), and assemble a collection of GUIEventHandlers as appropriate for the viewer.

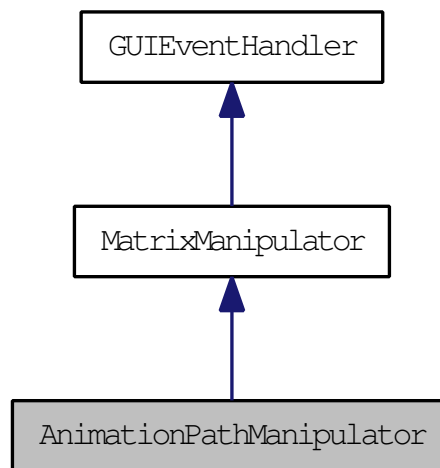
Events from the windowing environment are adapted, and then fed into the GUIEventHandlers. The GUIEventHandlers analyse and take action, and make requests of the windowing environment via the **GUIActionAdapter** (p. 38). The viewer writer should then honour these requests, translating them into calls to the windowing API.

# Class Documentation

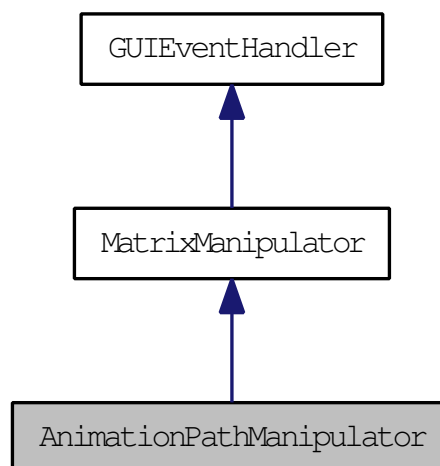
---

## 4.1 AnimationPathManipulator Class Reference

Inheritance diagram for AnimationPathManipulator:



Collaboration diagram for AnimationPathManipulator:



### Public Member Functions

- **AnimationPathManipulator** (const std::string &filename)
- **AnimationPathManipulator** (osg::AnimationPath \*animationPath=0)
- virtual const char \* **className** () const

- const osg::AnimationPath \* **getAnimationPath** () const
- osg::AnimationPath \* **getAnimationPath** ()
- virtual osg::Matrixd **getInverseMatrix** () const  
*get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.*
- virtual osg::Matrixd **getMatrix** () const  
*get the position of the manipulator as 4x4 Matrix.*
- bool **getPrintOutTimingInfo** () const
- virtual void **getUsage** (osg::ApplicationUsage &usage) const  
*Get the keyboard and mouse usage of this manipulator.*
- virtual bool **handle** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Handle events, return true if handled, false otherwise.*
- void **home** (double currentTime)  
*Move the camera to the default position.*
- void **home** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Move the camera to the default position.*
- void **init** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Start/restart the manipulator.*
- void **setAnimationPath** (osg::AnimationPath \*animationPath)
- virtual void **setByInverseMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setByMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- void **setPrintOutTimingInfo** (bool printOutTimingInfo)
- bool **valid** () const

### Protected Member Functions

- void **handleFrame** (double time)

### Protected Attributes

- osg::ref\_ptr< osg::AnimationPath > **\_animationPath**
- double **\_animStartOfTimedPeriod**
- bool **\_isPaused**
- osg::Matrixd **\_matrix**
- int **\_numOfFramesSinceStartOfTimedPeriod**
- double **\_pauseTime**
- bool **\_printOutTimingInfo**
- double **\_realStartOfTimedPeriod**
- double **\_timeOffset**
- double **\_timeScale**
- bool **\_valid**

## 4.1.1 Constructor & Destructor Documentation

4.1.1.1 **AnimationPathManipulator** (osg::AnimationPath \* *animationPath* = 0)

4.1.1.2 **AnimationPathManipulator** (const std::string & *filename*)

## 4.1.2 Member Function Documentation

4.1.2.1 **virtual const char\*** **className** () const [inline, virtual]

Reimplemented from **MatrixManipulator** (p. 64).

4.1.2.2 **const osg::AnimationPath\*** **getAnimationPath** () const [inline]

4.1.2.3 **osg::AnimationPath\*** **getAnimationPath** () [inline]

4.1.2.4 **virtual osg::Matrixd** **getInverseMatrix** () const [inline, virtual]

get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.

Implements **MatrixManipulator** (p. 65).

4.1.2.5 **virtual osg::Matrixd** **getMatrix** () const [inline, virtual]

get the position of the manipulator as 4x4 Matrix.

Implements **MatrixManipulator** (p. 65).

4.1.2.6 **bool** **getPrintOutTimingInfo** () const [inline]

4.1.2.7 **void** **getUsage** (osg::ApplicationUsage & *usage*) const [virtual]

Get the keyboard and mouse usage of this manipulator.

Reimplemented from **GUIEventHandler** (p. 55).

4.1.2.8 **bool** **handle** (const GUIEventAdapter & *ea*, osgGA::GUIActionAdapter & *us*) [virtual]

Handle events, return true if handled, false otherwise.

Reimplemented from **MatrixManipulator** (p. 65).

4.1.2.9 **void** **handleFrame** (double *time*) [protected]

4.1.2.10 **void** **home** (double) [virtual]

Move the camera to the default position. This version does not require **GUIEventAdapter** (p. 40) and **GUIActionAdapter** (p. 38) so may be called from somewhere other than a **handle()** (p. 11) method in **GUIEventHandler** (p. 54). Application must be aware of implications.

Reimplemented from **MatrixManipulator** (p. 66).

4.1.2.11 **void** **home** (const GUIEventAdapter &, GUIActionAdapter &) [virtual]

Move the camera to the default position. May be ignored by manipulators if home functionality is not appropriate.

Reimplemented from **MatrixManipulator** (p. 66).

4.1.2.12 **void** **init** (const GUIEventAdapter &, GUIActionAdapter &) [virtual]

Start/restart the manipulator. FIXME: what does this actually mean? Provide examples.

Reimplemented from **MatrixManipulator** (p. 66).

4.1.2.13 **void** **setAnimationPath** (osg::AnimationPath \* *animationPath*) [inline]

4.1.2.14 **virtual void** **setByInverseMatrix** (const osg::Matrixd & *matrix*) [inline, virtual]

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

4.1.2.15 **virtual void** **setByMatrix** (const osg::Matrixd & *matrix*) [inline, virtual]

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

4.1.2.16 void setPrintOutTimingInfo (bool *printOutTimingInfo*) [inline]

4.1.2.17 bool valid () const [inline]

### 4.1.3 Member Data Documentation

4.1.3.1 osg::ref\_ptr<osg::AnimationPath> \_animationPath [protected]

4.1.3.2 double \_animStartOfTimedPeriod [protected]

4.1.3.3 bool \_isPaused [protected]

4.1.3.4 osg::Matrixd \_matrix [protected]

4.1.3.5 int \_numOfFramesSinceStartOfTimedPeriod [protected]

4.1.3.6 double \_pauseTime [protected]

4.1.3.7 bool \_printOutTimingInfo [protected]

4.1.3.8 double \_realStartOfTimedPeriod [protected]

4.1.3.9 double \_timeOffset [protected]

4.1.3.10 double \_timeScale [protected]

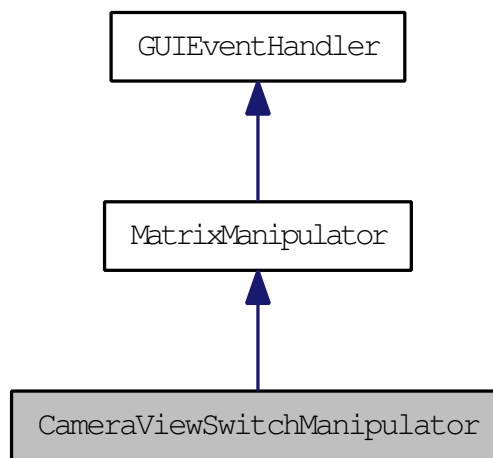
4.1.3.11 bool \_valid [protected]

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

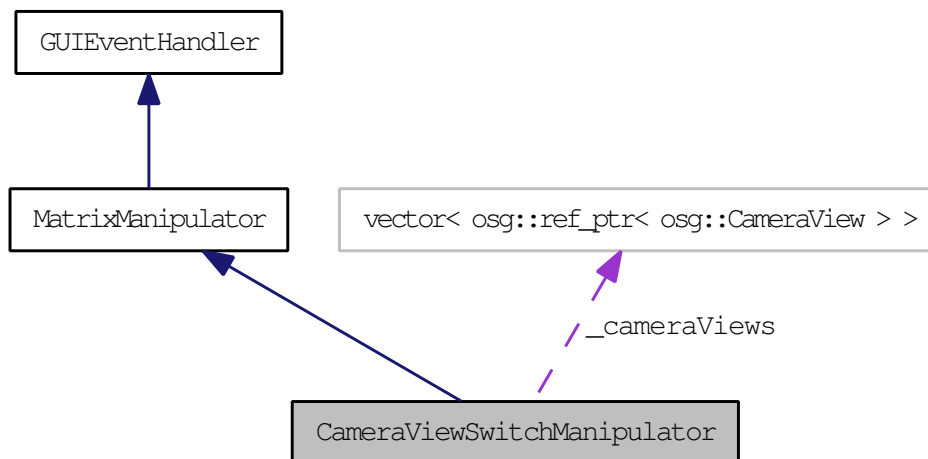
- AnimationPathManipulator
- AnimationPathManipulator.cpp

## 4.2 CameraViewSwitchManipulator Class Reference

Inheritance diagram for CameraViewSwitchManipulator:



Collaboration diagram for CameraViewSwitchManipulator:



### Public Types

- typedef std::vector< osg::ref\_ptr< osg::CameraView > > **CameraViewList**

### Public Member Functions

- **CameraViewSwitchManipulator** ()
- virtual const char \* **className** () const
- virtual osg::Matrixd **getInverseMatrix** () const  
*get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.*
- virtual osg::Matrixd **getMatrix** () const  
*get the position of the manipulator as 4x4 Matrix.*
- virtual osg::Node \* **getNode** ()  
*Return node if attached.*
- virtual const osg::Node \* **getNode** () const

*Return node if attached.*

- virtual void **getUsage** (osg::ApplicationUsage &usage) const  
*Get the keyboard and mouse usage of this manipulator.*
- virtual bool **handle** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*handle events, return true if handled, false otherwise.*
- virtual void **init** (const **GUIEventAdapter** &, **GUIActionAdapter** &)  
*Start/restart the manipulator.*
- virtual void **setByInverseMatrix** (const osg::Matrixd &)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setByMatrix** (const osg::Matrixd &)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setNode** (osg::Node \*)  
*Attach a node to the manipulator.*

## Protected Member Functions

- virtual ~**CameraViewSwitchManipulator** ()

## Protected Attributes

- **CameraViewList \_cameraViews**
- unsigned int **\_currentView**
- osg::ref\_ptr< osg::Node > **\_node**

### 4.2.1 Member Typedef Documentation

4.2.1.1 typedef std::vector< osg::ref\_ptr<osg::CameraView> > **CameraViewList**

### 4.2.2 Constructor & Destructor Documentation

4.2.2.1 **CameraViewSwitchManipulator** () [inline]

4.2.2.2 virtual ~**CameraViewSwitchManipulator** () [inline, protected, virtual]

### 4.2.3 Member Function Documentation

4.2.3.1 virtual const char\* **className** () const [inline, virtual]

Reimplemented from **MatrixManipulator** (p. 64).

4.2.3.2 **osg::Matrixd getInverseMatrix** () const [virtual]

get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.

Implements **MatrixManipulator** (p. 65).

4.2.3.3 **osg::Matrixd getMatrix** () const [virtual]

get the position of the manipulator as 4x4 Matrix.

Implements **MatrixManipulator** (p. 65).

4.2.3.4 virtual **osg::Node\* getNode** () [inline, virtual]

Return node if attached.

Reimplemented from **MatrixManipulator** (p. 65).

**4.2.3.5 virtual const osg::Node\* getNode () const [inline, virtual]**

Return node if attached.

Reimplemented from **MatrixManipulator** (p. 65).

**4.2.3.6 void getUsage (osg::ApplicationUsage & usage) const [virtual]**

Get the keyboard and mouse usage of this manipulator.

Reimplemented from **GUIEventHandler** (p. 55).

**4.2.3.7 bool handle (const GUIEventAdapter & ea, GUIActionAdapter & us) [virtual]**

handle events, return true if handled, false otherwise.

Reimplemented from **MatrixManipulator** (p. 65).

**4.2.3.8 virtual void init (const GUIEventAdapter &, GUIActionAdapter &) [inline, virtual]**

Start/restart the manipulator.

Reimplemented from **MatrixManipulator** (p. 66).

**4.2.3.9 virtual void setByInverseMatrix (const osg::Matrixd &) [inline, virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.2.3.10 virtual void setByMatrix (const osg::Matrixd &) [inline, virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.2.3.11 void setNode (osg::Node \* node) [virtual]**

Attach a node to the manipulator. Automatically detaches previously attached node. setNode(NULL) detaches previously nodes. Is ignored by manipulators which do not require a reference model.

Reimplemented from **MatrixManipulator** (p. 67).

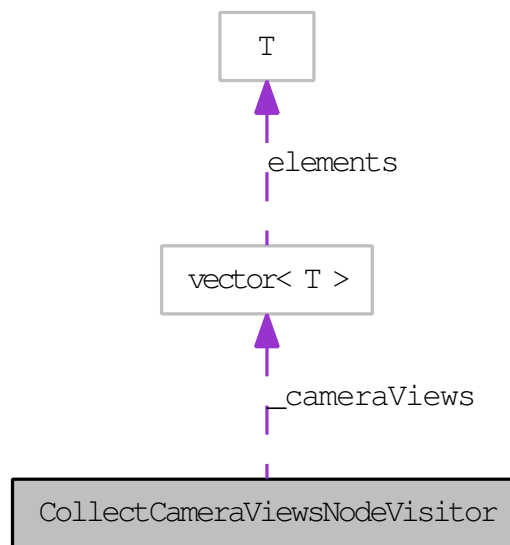
**4.2.4 Member Data Documentation****4.2.4.1 CameraViewList \_cameraViews [protected]****4.2.4.2 unsigned int \_currentView [protected]****4.2.4.3 osg::ref\_ptr<osg::Node> \_node [protected]**

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

- **CameraViewSwitchManipulator**
- **CameraViewSwitchManipulator.cpp**

## 4.3 CollectCameraViewsNodeVisitor Class Reference

Collaboration diagram for CollectCameraViewsNodeVisitor:



### Public Member Functions

- `CollectCameraViewsNodeVisitor (CameraViewSwitchManipulator::CameraViewList *cameraViews)`
- virtual void `apply (CameraView &node)`

### Public Attributes

- `CameraViewSwitchManipulator::CameraViewList * _cameraViews`

### 4.3.1 Constructor & Destructor Documentation

- #### 4.3.1.1 `CollectCameraViewsNodeVisitor (CameraViewSwitchManipulator::CameraViewList * cameraViews) [inline]`

### 4.3.2 Member Function Documentation

- #### 4.3.2.1 `virtual void apply (CameraView & node) [inline, virtual]`

### 4.3.3 Member Data Documentation

- #### 4.3.3.1 `CameraViewSwitchManipulator::CameraViewList* _cameraViews`

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

- `CameraViewSwitchManipulator.cpp`

## 4.4 CoordinateFrameCallback Class Reference

callback class to use to allow matrix manipulators to query the application for the local coordinate frame.

### Public Member Functions

- virtual osg::CoordinateFrame **getCoordinateFrame** (const osg::Vec3d &position) const =0

### Protected Member Functions

- virtual ~CoordinateFrameCallback ()

#### 4.4.1 Detailed Description

callback class to use to allow matrix manipulators to query the application for the local coordinate frame.

#### 4.4.2 Constructor & Destructor Documentation

4.4.2.1 virtual ~CoordinateFrameCallback () [inline, protected, virtual]

#### 4.4.3 Member Function Documentation

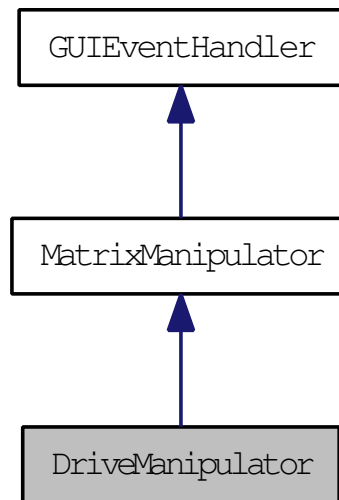
4.4.3.1 virtual osg::CoordinateFrame **getCoordinateFrame** (const osg::Vec3d & *position*) const [pure virtual]

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

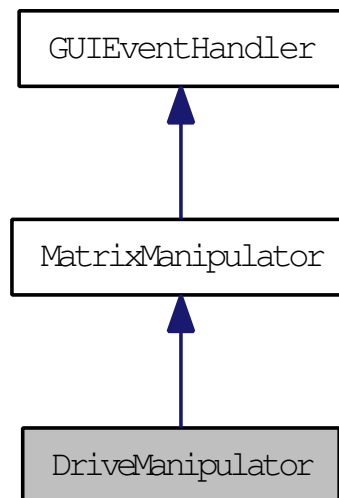
- **MatrixManipulator**

## 4.5 DriveManipulator Class Reference

**DriveManipulator** (p. 18) is a camera manipulator which provides drive-like functionality. Inheritance diagram for DriveManipulator:



Collaboration diagram for DriveManipulator:



### Public Member Functions

- **DriveManipulator** ()
- virtual const char \* **className** () const
- virtual void **computeHomePosition** ()  
*Compute the home position.*
- double **getHeight** () const
- virtual osg::Matrixd **getInverseMatrix** () const  
*get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.*
- virtual osg::Matrixd **getMatrix** () const  
*get the position of the manipulator as 4x4 Matrix.*
- double **getModelScale** () const

- virtual osg::Node \* **getNode** ()  
*Return node if attached.*
- virtual const osg::Node \* **getNode** () const  
*Return const node if attached.*
- virtual void **getUsage** (osg::ApplicationUsage &usage) const  
*Get the keyboard and mouse usage of this manipulator.*
- double **getVelocity** () const
- virtual bool **handle** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Handle events, return true if handled, false otherwise.*
- virtual void **home** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Move the camera to the default position.*
- virtual void **init** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Start/restart the manipulator.*
- virtual void **setByInverseMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setByMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- void **setHeight** (double in\_h)
- void **setModelScale** (double in\_ms)
- virtual void **setNode** (osg::Node \*)  
*Attach a node to the manipulator, automatically detaching any previously attached node.*
- void **setVelocity** (double in\_vel)

### Protected Types

- enum **SpeedControlMode** { **USE\_MOUSE\_Y\_FOR\_SPEED**, **USE\_MOUSE\_BUTTONS\_FOR\_SPEED** }

### Protected Member Functions

- virtual ~**DriveManipulator** ()
- void **addMouseEvent** (const **GUIEventAdapter** &ea)  
*Add the current mouse GUIEvent to internal stack.*
- bool **calcMovement** ()  
*For the give mouse movement calculate the movement of the camera.*
- void **computePosition** (const osg::Vec3d &eye, const osg::Vec3d &lv, const osg::Vec3d &up)
- void **flushMouseEventStack** ()  
*Reset the internal GUIEvent stack.*
- bool **intersect** (const osg::Vec3d &start, const osg::Vec3d &end, osg::Vec3d &intersection, osg::Vec3d &normal) const

## Protected Attributes

- double **\_buffer**
- double **\_distance**
- osg::Vec3d **\_eye**
- osg::ref\_ptr< const **GUIEventAdapter** > **\_ga\_t0**
- osg::ref\_ptr< const **GUIEventAdapter** > **\_ga\_t1**
- double **\_height**
- double **\_modelScale**
- osg::ref\_ptr< osg::Node > **\_node**
- double **\_pitch**
- bool **\_pitchDownKeyPressed**
- bool **\_pitchUpKeyPressed**
- osg::Quat **\_rotation**
- **SpeedControlMode** **\_speedMode**
- double **\_velocity**

### 4.5.1 Detailed Description

**DriveManipulator** (p. 18) is a camera manipulator which provides drive-like functionality. By default, the left mouse button accelerates, the right mouse button decelerates, and the middle mouse button (or left and right simultaneously) stops dead.

### 4.5.2 Member Enumeration Documentation

#### 4.5.2.1 enum SpeedControlMode [protected]

Enumerator:

*USE\_MOUSE\_Y\_FOR\_SPEED*  
*USE\_MOUSE\_BUTTONS\_FOR\_SPEED*

### 4.5.3 Constructor & Destructor Documentation

#### 4.5.3.1 DriveManipulator ()

#### 4.5.3.2 ~DriveManipulator () [protected, virtual]

### 4.5.4 Member Function Documentation

#### 4.5.4.1 void addMouseEvent (const GUIEventAdapter & ea) [protected]

Add the current mouse GUIEvent to internal stack.

#### 4.5.4.2 bool calcMovement () [protected]

For the give mouse movement calculate the movement of the camera. Return true is camera has moved and a redraw is required.

#### 4.5.4.3 virtual const char\* className () const [inline, virtual]

Reimplemented from **MatrixManipulator** (p. 64).

#### 4.5.4.4 void computeHomePosition () [virtual]

Compute the home position.

Reimplemented from **MatrixManipulator** (p. 64).

#### 4.5.4.5 void computePosition (const osg::Vec3d & eye, const osg::Vec3d & lv, const osg::Vec3d & up) [protected]

#### 4.5.4.6 void flushMouseEventStack () [protected]

Reset the internal GUIEvent stack.

**4.5.4.7 double getHeight () const [inline]****4.5.4.8 osg::Matrixd getInverseMatrix () const [virtual]**

get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.  
Implements **MatrixManipulator** (p. 65).

**4.5.4.9 osg::Matrixd getMatrix () const [virtual]**

get the position of the manipulator as 4x4 Matrix.  
Implements **MatrixManipulator** (p. 65).

**4.5.4.10 double getModelScale () const [inline]****4.5.4.11 osg::Node \* getNode () [virtual]**

Return node if attached.

Reimplemented from **MatrixManipulator** (p. 65).

**4.5.4.12 const osg::Node \* getNode () const [virtual]**

Return const node if attached.

Reimplemented from **MatrixManipulator** (p. 65).

**4.5.4.13 void getUsage (osg::ApplicationUsage & usage) const [virtual]**

Get the keyboard and mouse usage of this manipulator.

Reimplemented from **GUIEventHandler** (p. 55).

**4.5.4.14 double getVelocity () const [inline]****4.5.4.15 bool handle (const GUIEventAdapter & ea, GUIActionAdapter & ua) [virtual]**

Handle events, return true if handled, false otherwise.

Reimplemented from **MatrixManipulator** (p. 65).

**4.5.4.16 void home (const GUIEventAdapter &, GUIActionAdapter &) [virtual]**

Move the camera to the default position. May be ignored by manipulators if home functionality is not appropriate.

Reimplemented from **MatrixManipulator** (p. 66).

**4.5.4.17 void init (const GUIEventAdapter &, GUIActionAdapter &) [virtual]**

Start/restart the manipulator. FIXME: what does this actually mean? Provide examples.

Reimplemented from **MatrixManipulator** (p. 66).

**4.5.4.18 bool intersect (const osg::Vec3d & start, const osg::Vec3d & end, osg::Vec3d & intersection, osg::Vec3d & normal) const [protected]****4.5.4.19 virtual void setByInverseMatrix (const osg::Matrixd & matrix) [inline, virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.5.4.20 void setByMatrix (const osg::Matrixd & matrix) [virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

4.5.4.21 void setHeight (double *in\_h*) [inline]

4.5.4.22 void setModelScale (double *in\_ms*) [inline]

4.5.4.23 void setNode (osg::Node \*) [virtual]

Attach a node to the manipulator, automatically detaching any previously attached node. setNode(NULL) detaches previous nodes. May be ignored by manipulators which do not require a reference model.

Reimplemented from **MatrixManipulator** (p. 67).

4.5.4.24 void setVelocity (double *in\_vel*) [inline]

#### 4.5.5 Member Data Documentation

4.5.5.1 double \_buffer [protected]

4.5.5.2 double \_distance [protected]

4.5.5.3 osg::Vec3d \_eye [protected]

4.5.5.4 osg::ref\_ptr<const GUIEventAdapter> \_ga\_t0 [protected]

4.5.5.5 osg::ref\_ptr<const GUIEventAdapter> \_ga\_t1 [protected]

4.5.5.6 double \_height [protected]

4.5.5.7 double \_modelScale [protected]

4.5.5.8 osg::ref\_ptr<osg::Node> \_node [protected]

4.5.5.9 double \_pitch [protected]

4.5.5.10 bool \_pitchDownKeyPressed [protected]

4.5.5.11 bool \_pitchUpKeyPressed [protected]

4.5.5.12 osg::Quat \_rotation [protected]

4.5.5.13 SpeedControlMode \_speedMode [protected]

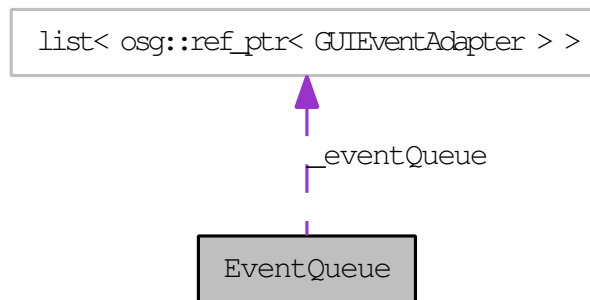
4.5.5.14 double \_velocity [protected]

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

- DriveManipulator
- DriveManipulator.cpp

## 4.6 EventQueue Class Reference

**EventQueue** (p.23) implementation for collecting and adapting windowing events. Collaboration diagram for EventQueue:



### Public Types

- typedef std::list< osg::ref\_ptr< **GUIEventAdapter** > > **Events**

### Public Member Functions

- **EventQueue** (**GUIEventAdapter::MouseYOrientation** mouseYOrientation=**GUIEventAdapter::Y\_INCREASING\_DOWNWARDS**)
- void **addEvent** (**GUIEventAdapter** \*event)  
*Add an event to the end of the event queue.*
- void **appendEvents** (**Events** &events)  
*Add events to end of event queue.*
- void **closeWindow** (double time)  
*Method for adapting close window event with specified event time.*
- void **closeWindow** ()  
*Method for adapting close window events.*
- bool **copyEvents** (**Events** &events) const  
*Take a copy the entire event queue leaving the EventQueue' event queue intact.*
- **GUIEventAdapter** \* **createEvent** ()  
*convenience method for create an event ready to fill in.*
- void **frame** (double time)  
*Method for adapting frame events.*
- const **GUIEventAdapter** \* **getCurrentEventState** () const
- **GUIEventAdapter** \* **getCurrentEventState** ()
- osg::Timer\_t **getStartTick** () const
- double **getTime** () const
- bool **getUseFixedMouseInputRange** ()  
*Get whether the mouse coordinates should be transformed into a pre defined input range.*
- void **keyPress** (int key, double time)  
*Method for adapting keyboard press events.*
- void **keyPress** (int key)

*Method for adapting keyboard press events.*

- void **keyRelease** (int key, double time)  
*Method for adapting keyboard press events.*
- void **keyRelease** (int key)  
*Method for adapting keyboard press events.*
- void **mouseButtonPress** (float x, float y, unsigned int button, double time)  
*Method for adapting mouse button pressed events, placing this event on the back of the event queue, with specified time.*
- void **mouseButtonPress** (float x, float y, unsigned int button)  
*Method for adapting mouse button pressed events, placing this event on the back of the event queue.*
- void **mouseButtonRelease** (float x, float y, unsigned int button, double time)  
*Method for adapting mouse button release events, placing this event on the back of the event queue, with specified time.*
- void **mouseButtonRelease** (float x, float y, unsigned int button)  
*Method for adapting mouse button release events, placing this event on the back of the event queue.*
- void **mouseDoubleButtonPress** (float x, float y, unsigned int button, double time)  
*Method for adapting mouse button pressed events, placing this event on the back of the event queue, with specified time.*
- void **mouseDoubleButtonPress** (float x, float y, unsigned int button)  
*Method for adapting mouse button pressed events, placing this event on the back of the event queue.*
- void **mouseMotion** (float x, float y, double time)  
*Method for adapting mouse motion events, placing this event on the back of the event queue, with specified time.*
- void **mouseMotion** (float x, float y)  
*Method for adapting mouse motion events, placing this event on the back of the event queue.*
- void **mouseScroll** (GUIEventAdapter::ScrollingMotion sm, double time)  
*Method for adapting mouse scroll wheel events, placing this event on the back of the event queue, with specified time.*
- void **mouseScroll** (GUIEventAdapter::ScrollingMotion sm)  
*Method for adapting mouse scroll wheel events, placing this event on the back of the event queue.*
- void **mouseScroll2D** (float x, float y, double time)  
*Method for adapting mouse scroll wheel events, placing this event on the back of the event queue.*
- void **mouseScroll2D** (float x, float y)  
*Method for adapting mouse scroll wheel events, placing this event on the back of the event queue.*
- void **mouseWarped** (float x, float y)  
*Method for updating in response to a mouse warp.*
- void **penOrientation** (float tiltX, float tiltY, float rotation, double time)  
*Method for adapting pen orientation events, placing this event on the back of the event queue, with specified time.*
- void **penOrientation** (float tiltX, float tiltY, float rotation)  
*Method for adapting pen orientation events, placing this event on the back of the event queue.*

- void **penPressure** (float pressure, double time)  
*Method for adapting pen pressure events, placing this event on the back of the event queue, with specified time.*
- void **penPressure** (float pressure)  
*Method for adapting pen pressure events, placing this event on the back of the event queue.*
- void **penProximity** (**GUIEventAdapter::TabletPointerType** pt, bool isEntering, double time)  
*Method for adapting pen proximity events, placing this event on the back of the event queue, with specified time.*
- void **penProximity** (**GUIEventAdapter::TabletPointerType** pt, bool isEntering)  
*Method for adapting pen proximity events, placing this event on the back of the event queue.*
- void **quitApplication** (double time)  
*Method for adapting application quit events with specified event time.*
- void **quitApplication** ()  
*Method for adapting application quit events.*
- void **setCurrentEventState** (**GUIEventAdapter** \*ea)
- void **setEvents** (**Events** &events)  
*Set events.*
- void **setGraphicsContext** (osg::GraphicsContext \*context)  
*Set the graphics context associated with this event queue.*
- void **setMouseInputRange** (float xMin, float yMin, float xMax, float yMax)  
*Set the mouse input range.*
- void **setStartTick** (osg::Timer\_t tick)
- void **setUseFixedMouseInputRange** (bool useFixedMouseInputRange)  
*Specify if mouse coordinates should be transformed into a pre defined input range, or whether they should be simply based on as local coordinates to the window that generated the mouse events.*
- bool **takeEvents** (**Events** &events)  
*Take the entire event queue leaving the EventQueue' event queue empty.*
- void **userEvent** (osg::Referenced \*userEventData, double time)  
*Method for adapting user defined events with specified event time.*
- void **userEvent** (osg::Referenced \*userEventData)  
*Method for adapting user defined events.*
- void **windowResize** (int x, int y, int width, int height, double time)  
*Method for adapting window resize event, placing this event on the back of the event queue, with specified time.*
- void **windowResize** (int x, int y, int width, int height)  
*Method for adapting window resize event, placing this event on the back of the event queue.*

### Protected Member Functions

- virtual **~EventQueue** ()
- **EventQueue** & **operator=** (const **EventQueue** &)  
*Prevent unwanted copy operator.*

## Protected Attributes

- osg::ref\_ptr< **GUIEventAdapter** > **\_accumulateEventState**
- **Events** **\_eventQueue**
- OpenThreads::Mutex **\_eventQueueMutex**
- osg::Timer\_t **\_startTick**
- bool **\_useFixedMouseInputRange**

### 4.6.1 Detailed Description

**EventQueue** (p. 23) implementation for collecting and adapting windowing events.

### 4.6.2 Member Typedef Documentation

4.6.2.1 **typedef std::list< osg::ref\_ptr<GUIEventAdapter> > Events**

### 4.6.3 Constructor & Destructor Documentation

4.6.3.1 **EventQueue (GUIEventAdapter::MouseYOrientation *mouseYOrientation* = GUIEventAdapter::Y\_INCREASING\_DOWNWARDS)**

4.6.3.2 **~EventQueue () [protected, virtual]**

### 4.6.4 Member Function Documentation

4.6.4.1 **void addEvent (GUIEventAdapter \* *event*)**

Add an event to the end of the event queue.

4.6.4.2 **void appendEvents (Events & *events*)**

Add events to end of event queue.

4.6.4.3 **void closeWindow (double *time*)**

Method for adapting close window event with specified event time.

4.6.4.4 **void closeWindow () [inline]**

Method for adapting close window events.

4.6.4.5 **bool copyEvents (Events & *events*) const**

Take a copy the entire event queue leaving the EventQueue' event queue intact.

4.6.4.6 **GUIEventAdapter \* createEvent ()**

convenience method for create an event ready to fill in. Clones the **getCurrentEventState()** (p. 26) to produce a up to date event state.

4.6.4.7 **void frame (double *time*)**

Method for adapting frame events.

4.6.4.8 **const GUIEventAdapter\* getCurrentEventState () const [inline]**

4.6.4.9 **GUIEventAdapter\* getCurrentEventState () [inline]**

4.6.4.10 **osg::Timer\_t getStartTick () const [inline]**

4.6.4.11 **double getTime () const [inline]**

4.6.4.12 **bool getUseFixedMouseInputRange () [inline]**

Get whether the mouse coordinates should be transformed into a pre defined input range.

4.6.4.13 **void keyPress (int *key*, double *time*)**

Method for adapting keyboard press events. Note, special keys such as Ctrl/Function keys should be adapted to **GUIEventAdapter::KeySymbol** (p. 45) mappings, with specified time.

**4.6.4.14 void keyPressed (int key) [inline]**

Method for adapting keyboard press events. Note, special keys such as Ctrl/Function keys should be adapted to **GUIEventAdapter::KeySymbol** (p. 45) mappings.

**4.6.4.15 void keyRelease (int key, double time)**

Method for adapting keyboard press events. Note, special keys such as Ctrl/Function keys should be adapted to **GUIEventAdapter::KeySymbol** (p. 45) mappings, with specified time.

**4.6.4.16 void keyRelease (int key) [inline]**

Method for adapting keyboard press events. Note, special keys such as Ctrl/Function keys should be adapted to **GUIEventAdapter::KeySymbol** (p. 45) mappings.

**4.6.4.17 void mouseButtonPress (float x, float y, unsigned int button, double time)**

Method for adapting mouse button pressed events, placing this event on the back of the event queue, with specified time. Button numbering is 1 for left mouse button, 2 for middle, 3 for right.

**4.6.4.18 void mouseButtonPress (float x, float y, unsigned int button) [inline]**

Method for adapting mouse button pressed events, placing this event on the back of the event queue. Button numbering is 1 for left mouse button, 2 for middle, 3 for right.

**4.6.4.19 void mouseButtonRelease (float x, float y, unsigned int button, double time)**

Method for adapting mouse button release events, placing this event on the back of the event queue, with specified time. Button numbering is 1 for left mouse button, 2 for middle, 3 for right.

**4.6.4.20 void mouseButtonRelease (float x, float y, unsigned int button) [inline]**

Method for adapting mouse button release events, placing this event on the back of the event queue. Button numbering is 1 for left mouse button, 2 for middle, 3 for right.

**4.6.4.21 void mouseDoubleButtonPress (float x, float y, unsigned int button, double time)**

Method for adapting mouse button pressed events, placing this event on the back of the event queue, with specified time. Button numbering is 1 for left mouse button, 2 for middle, 3 for right.

**4.6.4.22 void mouseDoubleButtonPress (float x, float y, unsigned int button) [inline]**

Method for adapting mouse button pressed events, placing this event on the back of the event queue. Button numbering is 1 for left mouse button, 2 for middle, 3 for right.

**4.6.4.23 void mouseMotion (float x, float y, double time)**

Method for adapting mouse motion events, placing this event on the back of the event queue, with specified time.

**4.6.4.24 void mouseMotion (float x, float y) [inline]**

Method for adapting mouse motion events, placing this event on the back of the event queue.

**4.6.4.25 void mouseScroll (GUIEventAdapter::ScrollingMotion sm, double time)**

Method for adapting mouse scroll wheel events, placing this event on the back of the event queue, with specified time.

**4.6.4.26 void mouseScroll (GUIEventAdapter::ScrollingMotion sm) [inline]**

Method for adapting mouse scroll wheel events, placing this event on the back of the event queue.

**4.6.4.27 void mouseScroll2D (float x, float y, double time)**

Method for adapting mouse scroll wheel events, placing this event on the back of the event queue.

**4.6.4.28 void mouseScroll2D (float x, float y) [inline]**

Method for adapting mouse scroll wheel events, placing this event on the back of the event queue.

**4.6.4.29 void mouseWarped (float *x*, float *y*)**

Method for updating in response to a mouse warp. Note, just moves the mouse position without creating a new event for it.

**4.6.4.30 EventQueue& operator= (const EventQueue &) [inline, protected]**

Prevent unwanted copy operator.

**4.6.4.31 void penOrientation (float *tiltX*, float *tiltY*, float *rotation*, double *time*)**

Method for adapting pen orientation events, placing this event on the back of the event queue, with specified time.

**4.6.4.32 void penOrientation (float *tiltX*, float *tiltY*, float *rotation*) [inline]**

Method for adapting pen orientation events, placing this event on the back of the event queue.

**4.6.4.33 void penPressure (float *pressure*, double *time*)**

Method for adapting pen pressure events, placing this event on the back of the event queue, with specified time.

**4.6.4.34 void penPressure (float *pressure*) [inline]**

Method for adapting pen pressure events, placing this event on the back of the event queue.

**4.6.4.35 void penProximity (GUIEventAdapter::TabletPointerType *pt*, bool *isEntering*, double *time*)**

Method for adapting pen proximity events, placing this event on the back of the event queue, with specified time.

**4.6.4.36 void penProximity (GUIEventAdapter::TabletPointerType *pt*, bool *isEntering*) [inline]**

Method for adapting pen proximity events, placing this event on the back of the event queue.

**4.6.4.37 void quitApplication (double *time*)**

Method for adapting application quit events with specified event time.

**4.6.4.38 void quitApplication () [inline]**

Method for adapting application quit events.

**4.6.4.39 void setCurrentEventState (GUIEventAdapter \* *ea*) [inline]****4.6.4.40 void setEvents (Events & *events*)**

Set events.

**4.6.4.41 void setGraphicsContext (osg::GraphicsContext \* *context*) [inline]**

Set the graphics context associated with this event queue.

**4.6.4.42 void setMouseInputRange (float *xMin*, float *yMin*, float *xMax*, float *yMax*) [inline]**

Set the mouse input range.

**4.6.4.43 void setStartTick (osg::Timer\_t *tick*) [inline]****4.6.4.44 void setUseFixedMouseInputRange (bool *useFixedMouseInputRange*) [inline]**

Specify if mouse coordinates should be transformed into a pre defined input range, or whether they should be simply based on as local coordinates to the window that generated the mouse events.

**4.6.4.45 bool takeEvents (Events & *events*)**

Take the entire event queue leaving the EventQueue' event queue empty.

**4.6.4.46 void userEvent (osg::Referenced \* *userEventData*, double *time*)**

Method for adapting user defined events with specified event time.

**4.6.4.47 void userEvent (osg::Referenced \* *userEventData*) [inline]**

Method for adapting user defined events.

**4.6.4.48 void windowResize (int *x*, int *y*, int *width*, int *height*, double *time*)**

Method for adapting window resize event, placing this event on the back of the event queue, with specified time.

**4.6.4.49 void windowResize (int *x*, int *y*, int *width*, int *height*) [inline]**

Method for adapting window resize event, placing this event on the back of the event queue.

**4.6.5 Member Data Documentation****4.6.5.1 osg::ref\_ptr<GUIEventAdapter> \_accumulateEventState [protected]****4.6.5.2 Events\_eventQueue [protected]****4.6.5.3 OpenThreads::Mutex\_eventQueueMutex [mutable, protected]****4.6.5.4 osg::Timer\_t\_startTick [protected]****4.6.5.5 bool \_useFixedMouseInputRange [protected]**

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

- **EventQueue**
- **EventQueue.cpp**

## 4.7 EventVisitor Class Reference

Basic **EventVisitor** (p. 30) implementation for animating a scene. Collaboration diagram for EventVisitor:



### Public Types

- typedef std::list< osg::ref\_ptr< **GUIEventAdapter** > > **EventList**

### Public Member Functions

- **EventVisitor** ()
- virtual ~**EventVisitor** ()
- void **addEvent** (**GUIEventAdapter** \*event)
- virtual void **apply** (osg::OccluderNode &node)
- virtual void **apply** (osg::LOD &node)
- virtual void **apply** (osg::Switch &node)
- virtual void **apply** (osg::Projection &node)
- virtual void **apply** (osg::Transform &node)
- virtual void **apply** (osg::Group &node)
- virtual void **apply** (osg::LightSource &node)
- virtual void **apply** (osg::Billboard &node)
- virtual void **apply** (osg::Geode &node)
- virtual void **apply** (osg::Node &node)

*During traversal each type of node calls its callbacks and its children traversed.*

- const **osgGA::GUIActionAdapter** \* **getActionAdapter** () const
- **osgGA::GUIActionAdapter** \* **getActionAdapter** ()
- bool **getEventHandled** () const
- const **EventQueue::Events** & **getEvents** () const
- **EventQueue::Events** & **getEvents** ()
- **META\_NodeVisitor** ("osgGA", "EventVisitor") void setActionAdapter(osgGA
- void **removeEvent** (**GUIEventAdapter** \*event)
- virtual void **reset** ()
- void **setEventHandled** (bool handled)
- void **setEvents** (const **EventQueue::Events** &events)

### Protected Member Functions

- void **handle\_callbacks** (osg::StateSet \*stateset)
- void **handle\_callbacks\_and\_traverse** (osg::Node &node)
- void **handle\_geode\_callbacks** (osg::Geode &node)
- **EventVisitor** & **operator=** (const **EventVisitor** &)

*Prevent unwanted copy operator.*

- void **traverseGeode** (osg::Geode &geode)

## Protected Attributes

- `osg::ref_ptr< GUIEventAdapter > _accumulateEventState`
- `osgGA::GUIActionAdapter * _actionAdapter`
- `EventQueue::Events _events`
- `bool _handled`

### 4.7.1 Detailed Description

Basic **EventVisitor** (p. 30) implementation for animating a scene. This visitor traverses the scene graph, calling each nodes `appCallback` if it exists.

### 4.7.2 Member Typedef Documentation

4.7.2.1 `typedef std::list< osg::ref_ptr<GUIEventAdapter> > EventList`

### 4.7.3 Constructor & Destructor Documentation

4.7.3.1 `EventVisitor ()`

4.7.3.2 `~EventVisitor () [virtual]`

### 4.7.4 Member Function Documentation

4.7.4.1 `void addEvent (GUIEventAdapter * event)`

4.7.4.2 `virtual void apply (osg::OccluderNode & node) [inline, virtual]`

4.7.4.3 `virtual void apply (osg::LOD & node) [inline, virtual]`

4.7.4.4 `virtual void apply (osg::Switch & node) [inline, virtual]`

4.7.4.5 `virtual void apply (osg::Projection & node) [inline, virtual]`

4.7.4.6 `virtual void apply (osg::Transform & node) [inline, virtual]`

4.7.4.7 `virtual void apply (osg::Group & node) [inline, virtual]`

4.7.4.8 `virtual void apply (osg::LightSource & node) [inline, virtual]`

4.7.4.9 `virtual void apply (osg::Billboard & node) [inline, virtual]`

4.7.4.10 `virtual void apply (osg::Geode & node) [inline, virtual]`

4.7.4.11 `virtual void apply (osg::Node & node) [inline, virtual]`

During traversal each type of node calls its callbacks and its children traversed.

4.7.4.12 `const osgGA::GUIActionAdapter* getActionAdapter () const [inline]`

4.7.4.13 `osgGA::GUIActionAdapter* getActionAdapter () [inline]`

4.7.4.14 `bool getEventHandled () const [inline]`

4.7.4.15 `const EventQueue::Events& getEvents () const [inline]`

4.7.4.16 `EventQueue::Events& getEvents () [inline]`

4.7.4.17 `void handle_callbacks (osg::StateSet * stateset) [inline, protected]`

4.7.4.18 `void handle_callbacks_and_traverse (osg::Node & node) [inline, protected]`

4.7.4.19 `void handle_geode_callbacks (osg::Geode & node) [inline, protected]`

4.7.4.20 `META_NodeVisitor ("osgGA", "EventVisitor") [inline]`

4.7.4.21 `EventVisitor& operator= (const EventVisitor &) [inline, protected]`

Prevent unwanted copy operator.

4.7.4.22 void removeEvent (GUIEventAdapter \* *event*)

4.7.4.23 void reset () [virtual]

4.7.4.24 void setEventHandled (bool *handled*) [inline]

4.7.4.25 void setEvents (const EventQueue::Events & *events*) [inline]

4.7.4.26 void traverseGeode (osg::Geode & *geode*) [inline, protected]

#### 4.7.5 Member Data Documentation

4.7.5.1 osg::ref\_ptr<GUIEventAdapter> \_accumulateEventState [protected]

4.7.5.2 osgGA::GUIActionAdapter\* \_actionAdapter [protected]

4.7.5.3 EventQueue::Events \_events [protected]

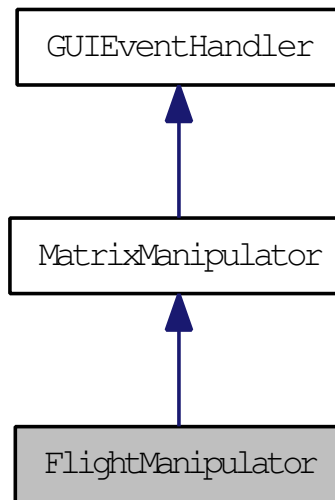
4.7.5.4 bool \_handled [protected]

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

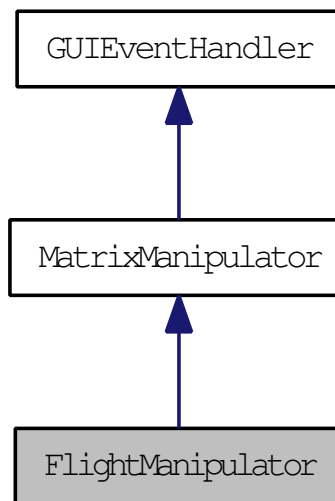
- EventVisitor
- EventVisitor.cpp

## 4.8 FlightManipulator Class Reference

**FlightManipulator** (p. 33) is a **MatrixManipulator** (p. 62) which provides flight simulator-like updating of the camera position & orientation. Inheritance diagram for FlightManipulator:



Collaboration diagram for FlightManipulator:



### Public Types

- enum **YawControlMode** { **YAW\_AUTOMATICALLY\_WHEN\_BANKED**, **NO\_AUTOMATIC\_YAW** }

### Public Member Functions

- **FlightManipulator** ()
- virtual const char \* **className** () const
- double **getAcceleration** () const
- virtual osg::Matrixd **getInverseMatrix** () const  
*get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.*
- virtual osg::Matrixd **getMatrix** () const  
*get the position of the manipulator as 4x4 Matrix.*

- double **getModelScale** () const
- virtual osg::Node \* **getNode** ()  
*Return node if attached.*
- virtual const osg::Node \* **getNode** () const  
*Return const node if attached.*
- virtual void **getUsage** (osg::ApplicationUsage &usage) const  
*Get the keyboard and mouse usage of this manipulator.*
- double **getVelocity** () const
- virtual bool **handle** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Handle events, return true if handled, false otherwise.*
- virtual void **home** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Move the camera to the default position.*
- virtual void **init** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Start/restart the manipulator.*
- void **setAcceleration** (double in\_acc)
- virtual void **setByInverseMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setByMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- void **setModelScale** (double in\_ms)
- virtual void **setNode** (osg::Node \*)  
*Attach a node to the manipulator, automatically detaching any previously attached node.*
- void **setVelocity** (double in\_vel)
- void **setYawControlMode** (**YawControlMode** ycm)  
*Configure the Yaw control for the flight model.*

### Protected Member Functions

- virtual ~**FlightManipulator** ()
- void **addMouseEvent** (const **GUIEventAdapter** &ea)  
*Add the current mouse GUIEvent to internal stack.*
- bool **calcMovement** ()  
*For the give mouse movement calculate the movement of the camera.*
- void **computePosition** (const osg::Vec3 &eye, const osg::Vec3 &lv, const osg::Vec3 &up)
- void **flushMouseEventStack** ()  
*Reset the internal GUIEvent stack.*

### Protected Attributes

- double **\_acceleration**
- double **\_distance**
- osg::Vec3d **\_eye**
- osg::ref\_ptr< const **GUIEventAdapter** > **\_ga\_t0**
- osg::ref\_ptr< const **GUIEventAdapter** > **\_ga\_t1**

- double **\_modelScale**
- osg::ref\_ptr< osg::Node > **\_node**
- osg::Quat **\_rotation**
- double **\_velocity**
- **YawControlMode \_yawMode**

### 4.8.1 Detailed Description

**FlightManipulator** (p. 33) is a **MatrixManipulator** (p. 62) which provides flight simulator-like updating of the camera position & orientation. By default, the left mouse button accelerates, the right mouse button decelerates, and the middle mouse button (or left and right simultaneously) stops dead.

### 4.8.2 Member Enumeration Documentation

#### 4.8.2.1 enum YawControlMode

Enumerator:

*YAW\_AUTOMATICALLY\_WHEN\_BANKED*  
*NO\_AUTOMATIC\_YAW*

### 4.8.3 Constructor & Destructor Documentation

#### 4.8.3.1 FlightManipulator ()

#### 4.8.3.2 ~FlightManipulator () [protected, virtual]

### 4.8.4 Member Function Documentation

#### 4.8.4.1 void addMouseEvent (const GUIEventAdapter & ea) [protected]

Add the current mouse GUIEvent to internal stack.

#### 4.8.4.2 bool calcMovement () [protected]

For the give mouse movement calculate the movement of the camera. Return true is camera has moved and a redraw is required.

#### 4.8.4.3 virtual const char\* className () const [inline, virtual]

Reimplemented from **MatrixManipulator** (p. 64).

#### 4.8.4.4 void computePosition (const osg::Vec3 & eye, const osg::Vec3 & lv, const osg::Vec3 & up) [protected]

#### 4.8.4.5 void flushMouseEventStack () [protected]

Reset the internal GUIEvent stack.

#### 4.8.4.6 double getAcceleration () const [inline]

#### 4.8.4.7 osg::Matrixd getInverseMatrix () const [virtual]

get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.

Implements **MatrixManipulator** (p. 65).

#### 4.8.4.8 osg::Matrixd getMatrix () const [virtual]

get the position of the manipulator as 4x4 Matrix.

Implements **MatrixManipulator** (p. 65).

#### 4.8.4.9 double getModelScale () const [inline]

#### 4.8.4.10 osg::Node \* getNode () [virtual]

Return node if attached.

Reimplemented from **MatrixManipulator** (p. 65).

**4.8.4.11 const osg::Node \* getNode () const [virtual]**

Return const node if attached.

Reimplemented from **MatrixManipulator** (p. 65).

**4.8.4.12 void getUsage (osg::ApplicationUsage & usage) const [virtual]**

Get the keyboard and mouse usage of this manipulator.

Reimplemented from **GUIEventHandler** (p. 55).

**4.8.4.13 double getVelocity () const [inline]****4.8.4.14 bool handle (const GUIEventAdapter & ea, GUIActionAdapter & us) [virtual]**

Handle events, return true if handled, false otherwise.

Reimplemented from **MatrixManipulator** (p. 65).

**4.8.4.15 void home (const GUIEventAdapter &, GUIActionAdapter &) [virtual]**

Move the camera to the default position. May be ignored by manipulators if home functionality is not appropriate.

Reimplemented from **MatrixManipulator** (p. 66).

**4.8.4.16 void init (const GUIEventAdapter &, GUIActionAdapter &) [virtual]**

Start/restart the manipulator. FIXME: what does this actually mean? Provide examples.

Reimplemented from **MatrixManipulator** (p. 66).

**4.8.4.17 void setAcceleration (double in\_acc) [inline]****4.8.4.18 virtual void setByInverseMatrix (const osg::Matrixd & matrix) [inline, virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.8.4.19 void setByMatrix (const osg::Matrixd & matrix) [virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.8.4.20 void setModelScale (double in\_ms) [inline]****4.8.4.21 void setNode (osg::Node \*) [virtual]**

Attach a node to the manipulator, automatically detaching any previously attached node. setNode(NULL) detaches previous nodes. May be ignored by manipulators which do not require a reference model.

Reimplemented from **MatrixManipulator** (p. 67).

**4.8.4.22 void setVelocity (double in\_ve) [inline]****4.8.4.23 void setYawControlMode (YawControlMode ycm) [inline]**

Configure the Yaw control for the flight model.

## 4.8.5 Member Data Documentation

4.8.5.1 `double _acceleration` [protected]

4.8.5.2 `double _distance` [protected]

4.8.5.3 `osg::Vec3d _eye` [protected]

4.8.5.4 `osg::ref_ptr<const GUIEventAdapter> _ga_t0` [protected]

4.8.5.5 `osg::ref_ptr<const GUIEventAdapter> _ga_t1` [protected]

4.8.5.6 `double _modelScale` [protected]

4.8.5.7 `osg::ref_ptr<osg::Node> _node` [protected]

4.8.5.8 `osg::Quat _rotation` [protected]

4.8.5.9 `double _velocity` [protected]

4.8.5.10 `YawControlMode _yawMode` [protected]

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

- `FlightManipulator`
- `FlightManipulator.cpp`

## 4.9 GUIActionAdapter Class Reference

Abstract base class defining the interface by which GUIEventHandlers may request actions of the GUI system in use.

### Public Member Functions

- virtual `~GUIActionAdapter ()`
- virtual `osg::View * asView ()`  
*Provide a mechanism for getting the `osg::View` associated with this **GUIActionAdapter** (p. 38).*
- virtual void `requestContinuousUpdate (bool needed=true)=0`  
*`requestContinuousUpdate(bool)` is for en/disabling a throw or idle callback to be requested by a **GUIEventHandler** (p. 54) (typically a **MatrixManipulator** (p. 62), though other **GUIEventHandler**'s may also provide functionality).*
- virtual void `requestRedraw ()=0`  
*`requestRedraw()` (p. 39) requests a single redraw.*
- virtual void `requestWarpPointer (float x, float y)=0`  
*`requestWarpPointer(int,int)` is requesting a repositioning of the mouse pointer to a specified x,y location on the window.*

### 4.9.1 Detailed Description

Abstract base class defining the interface by which GUIEventHandlers may request actions of the GUI system in use. These requests for actions should then be honored by the GUI toolkit of the user's application.

To provide more detail, when a **GUIEventHandler** (p. 54) (e.g. a **TrackballManipulator** (p. 92)) handles an incoming event, such as a mouse event, it may wish to make a request of the GUI. E.g. if a model is 'thrown', the trackball manipulator may wish to start a timer, and be repeatedly called, to continuously refresh the camera's position and orientation. However, it has no way of doing this, as it knows nothing of the window system in which it's operating. Instead, the **GUIEventHandler** (p. 54) issues it's request via a **GUIActionAdapter** (p. 38), and the viewer in use should honour the request, using the GUI system in play.

There is more than one way of using the **GUIActionAdapter** (p. 38). E.g. it may be inherited into a Viewer class, as is done with `osgGLUT::Viewer`. Alternatively, a simple subclass of **GUIActionAdapter** (p. 38) (e.g. `osgQt::QtActionAdapter`) may be passed to the **GUIEventHandler::handle()** (p. 55) function; once the function has returned, the viewer will then unpack the results and work out what to do to respond to the requests.

Also there are several ways to run your app and handle the updating of the window. `osgGLUT::Viewer` always has a idle callback registered which does a redraw all the time. `osgGLUT::Viewer` can safely ignore both **requestRedraw()** (p. 39) and `requestContinuousUpdate()` as these are happening all the time anyway.

Other apps will probably want to respond to the **requestRedraw()** (p. 39) and `requestContinuousUpdate(bool)` and again there is more than one way to handle it. You can override **requestRedraw()** (p. 39) and implement to call your own window redraw straight away. Or you can implement so that a flag is set and then you then respond the flag being set in your own leisure.

### 4.9.2 Constructor & Destructor Documentation

#### 4.9.2.1 virtual `~GUIActionAdapter ()` [`inline`, `virtual`]

### 4.9.3 Member Function Documentation

#### 4.9.3.1 virtual `osg::View* asView ()` [`inline`, `virtual`]

Provide a mechanism for getting the `osg::View` associated with this **GUIActionAdapter** (p. 38). One would use this to case view to `osgViewer::View(er)` if supported by the subclass.

**4.9.3.2 virtual void requestContinuousUpdate (bool *needed* = true) [pure virtual]**

requestContinuousUpdate(bool) is for en/disabling a throw or idle callback to be requested by a **GUIEventHandler** (p. 54) (typically a **MatrixManipulator** (p. 62), though other GUIEventHandler's may also provide functionality). GUI toolkits can respond to this immediately by registering an idle/timed callback, or can delay setting the callback and update at their own leisure.

**4.9.3.3 virtual void requestRedraw () [pure virtual]**

**requestRedraw()** (p. 39) requests a single redraw.

**4.9.3.4 virtual void requestWarpPointer (float *x*, float *y*) [pure virtual]**

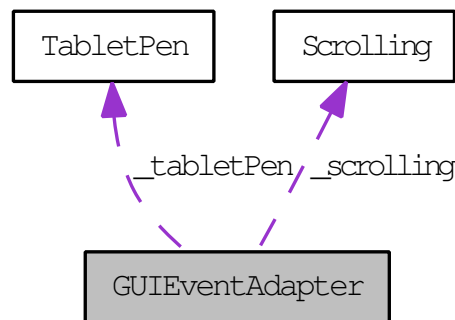
requestWarpPointer(int,int) is requesting a repositioning of the mouse pointer to a specified x,y location on the window. This is used by some camera manipulators to initialise the mouse pointer when mouse position relative to a controls neutral mouse position is required, i.e when mimicking a aircrafts joystick.

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

- **GUIActionAdapter**

## 4.10 GUIEventAdapter Class Reference

Event class for storing Keyboard, mouse and window events. Collaboration diagram for GUIEventAdapter:



### Classes

- struct **Scrolling**
- struct **TabletPen**

### Public Types

- enum **EventType** {
  - NONE** = 0, **PUSH** = 1<<0, **RELEASE** = 1<<1, **DOUBLECLICK** = 1<<2,
  - DRAG** = 1<<3, **MOVE** = 1<<4, **KEYDOWN** = 1<<5, **KEYUP** = 1<<6,
  - FRAME** = 1<<7, **RESIZE** = 1<<8, **SCROLL** = 1<<9, **PEN\_PRESSURE** = 1<<10,
  - PEN\_ORIENTATION** = 1<<11, **PEN\_PROXIMITY\_ENTER** = 1<<12, **PEN\_PROXIMITY\_LEAVE** = 1<<13,
  - CLOSE\_WINDOW** = 1<<14,
  - QUIT\_APPLICATION** = 1<<15, **USER** = 1<<16 }
- enum **KeySymbol** {
  - KEY\_Space** = 0x20, **KEY\_BackSpace** = 0xFF08, **KEY\_Tab** = 0xFF09, **KEY\_Linefeed** = 0xFF0A,
  - KEY\_Clear** = 0xFF0B, **KEY\_Return** = 0xFF0D, **KEY\_Pause** = 0xFF13, **KEY\_Scroll\_Lock** = 0xFF14,
  - KEY\_Sys\_Req** = 0xFF15, **KEY\_Escape** = 0xFF1B, **KEY\_Delete** = 0xFFFF, **KEY\_Home** = 0xFF50,
  - KEY\_Left** = 0xFF51, **KEY\_Up** = 0xFF52, **KEY\_Right** = 0xFF53, **KEY\_Down** = 0xFF54,
  - KEY\_Prior** = 0xFF55, **KEY\_Page\_Up** = 0xFF55, **KEY\_Next** = 0xFF56, **KEY\_Page\_Down** = 0xFF56,
  - KEY\_End** = 0xFF57, **KEY\_Begin** = 0xFF58, **KEY\_Select** = 0xFF60, **KEY\_Print** = 0xFF61,
  - KEY\_Execute** = 0xFF62, **KEY\_Insert** = 0xFF63, **KEY\_Undo** = 0xFF65, **KEY\_Redo** = 0xFF66,
  - KEY\_Menu** = 0xFF67, **KEY\_Find** = 0xFF68, **KEY\_Cancel** = 0xFF69, **KEY\_Help** = 0xFF6A,
  - KEY\_Break** = 0xFF6B, **KEY\_Mode\_switch** = 0xFF7E, **KEY\_Script\_switch** = 0xFF7E, **KEY\_Num\_Lock** = 0xFF7F,
  - KEY\_KP\_Space** = 0xFF80, **KEY\_KP\_Tab** = 0xFF89, **KEY\_KP\_Enter** = 0xFF8D, **KEY\_KP\_F1** = 0xFF91,
  - KEY\_KP\_F2** = 0xFF92, **KEY\_KP\_F3** = 0xFF93, **KEY\_KP\_F4** = 0xFF94, **KEY\_KP\_Home** = 0xFF95,
  - KEY\_KP\_Left** = 0xFF96, **KEY\_KP\_Up** = 0xFF97, **KEY\_KP\_Right** = 0xFF98, **KEY\_KP\_Down** = 0xFF99,
  - KEY\_KP\_Prior** = 0xFF9A, **KEY\_KP\_Page\_Up** = 0xFF9A, **KEY\_KP\_Next** = 0xFF9B, **KEY\_KP\_Page\_Down** = 0xFF9B,
  - KEY\_KP\_End** = 0xFF9C, **KEY\_KP\_Begin** = 0xFF9D, **KEY\_KP\_Insert** = 0xFF9E, **KEY\_KP\_Delete** = 0xFF9F,
  - KEY\_KP\_Equal** = 0xFFBD, **KEY\_KP\_Multiply** = 0xFFAA, **KEY\_KP\_Add** = 0xFFAB, **KEY\_KP\_Separator** = 0xFFAC,
  - KEY\_KP\_Subtract** = 0xFFAD, **KEY\_KP\_Decimal** = 0xFFAE, **KEY\_KP\_Divide** = 0xFFAF, **KEY\_KP\_0** = 0xFFB0,

**KEY\_KP\_1** = 0xFFB1, **KEY\_KP\_2** = 0xFFB2, **KEY\_KP\_3** = 0xFFB3, **KEY\_KP\_4** = 0xFFB4,  
**KEY\_KP\_5** = 0xFFB5, **KEY\_KP\_6** = 0xFFB6, **KEY\_KP\_7** = 0xFFB7, **KEY\_KP\_8** = 0xFFB8,  
**KEY\_KP\_9** = 0xFFB9, **KEY\_F1** = 0xFFBE, **KEY\_F2** = 0xFFBF, **KEY\_F3** = 0xFFC0,  
**KEY\_F4** = 0xFFC1, **KEY\_F5** = 0xFFC2, **KEY\_F6** = 0xFFC3, **KEY\_F7** = 0xFFC4,  
**KEY\_F8** = 0xFFC5, **KEY\_F9** = 0xFFC6, **KEY\_F10** = 0xFFC7, **KEY\_F11** = 0xFFC8,  
**KEY\_F12** = 0xFFC9, **KEY\_F13** = 0xFFCA, **KEY\_F14** = 0xFFCB, **KEY\_F15** = 0xFFCC,  
**KEY\_F16** = 0xFFCD, **KEY\_F17** = 0xFFCE, **KEY\_F18** = 0xFFCF, **KEY\_F19** = 0xFFD0,  
**KEY\_F20** = 0xFFD1, **KEY\_F21** = 0xFFD2, **KEY\_F22** = 0xFFD3, **KEY\_F23** = 0xFFD4,  
**KEY\_F24** = 0xFFD5, **KEY\_F25** = 0xFFD6, **KEY\_F26** = 0xFFD7, **KEY\_F27** = 0xFFD8,  
**KEY\_F28** = 0xFFD9, **KEY\_F29** = 0xFFDA, **KEY\_F30** = 0xFFDB, **KEY\_F31** = 0xFFDC,  
**KEY\_F32** = 0xFFDD, **KEY\_F33** = 0xFFDE, **KEY\_F34** = 0xFFDF, **KEY\_F35** = 0xFFE0,  
**KEY\_Shift\_L** = 0xFFE1, **KEY\_Shift\_R** = 0xFFE2, **KEY\_Control\_L** = 0xFFE3, **KEY\_Control\_R** = 0xFFE4,  
**KEY\_Caps\_Lock** = 0xFFE5, **KEY\_Shift\_Lock** = 0xFFE6, **KEY\_Meta\_L** = 0xFFE7, **KEY\_Meta\_R** =  
0xFFE8,  
**KEY\_Alt\_L** = 0xFFE9, **KEY\_Alt\_R** = 0xFFEA, **KEY\_Super\_L** = 0xFFEB, **KEY\_Super\_R** = 0xFFEC,  
**KEY\_Hyper\_L** = 0xFFED, **KEY\_Hyper\_R** = 0xFFEE }

- enum **ModKeyMask** {
  - MODKEY\_LEFT\_SHIFT** = 0x0001, **MODKEY\_RIGHT\_SHIFT** = 0x0002, **MODKEY\_LEFT\_CTRL** = 0x0004,  
**MODKEY\_RIGHT\_CTRL** = 0x0008,
  - MODKEY\_LEFT\_ALT** = 0x0010, **MODKEY\_RIGHT\_ALT** = 0x0020, **MODKEY\_LEFT\_META** = 0x0040,  
**MODKEY\_RIGHT\_META** = 0x0080,
  - MODKEY\_LEFT\_SUPER** = 0x0100, **MODKEY\_RIGHT\_SUPER** = 0x0200, **MODKEY\_LEFT\_HYPER** =  
0x0400, **MODKEY\_RIGHT\_HYPER** = 0x0800,
  - MODKEY\_NUM\_LOCK** = 0x1000, **MODKEY\_CAPS\_LOCK** = 0x2000, **MODKEY\_CTRL** = (MODKEY\_  
LEFT\_CTRL|MODKEY\_RIGHT\_CTRL), **MODKEY\_SHIFT** = (MODKEY\_LEFT\_SHIFT|MODKEY\_RIGHT\_  
SHIFT),
  - MODKEY\_ALT** = (MODKEY\_LEFT\_ALT|MODKEY\_RIGHT\_ALT), **MODKEY\_META** = (MODKEY\_LEFT\_  
META|MODKEY\_RIGHT\_META), **MODKEY\_SUPER** = (MODKEY\_LEFT\_SUPER|MODKEY\_RIGHT\_  
SUPER), **MODKEY\_HYPER** = (MODKEY\_LEFT\_HYPER|MODKEY\_RIGHT\_HYPER) }
- enum **MouseButtonMask** { **LEFT\_MOUSE\_BUTTON** = 1<<0, **MIDDLE\_MOUSE\_BUTTON** = 1<<1,  
**RIGHT\_MOUSE\_BUTTON** = 1<<2 }
- enum **MouseYOrientation** { **Y\_INCREASING\_UPWARDS**, **Y\_INCREASING\_DOWNWARDS** }
- enum **ScrollingMotion** {
  - SCROLL\_NONE**, **SCROLL\_LEFT**, **SCROLL\_RIGHT**, **SCROLL\_UP**,
  - SCROLL\_DOWN**, **SCROLL\_2D** }
- enum **TabletPointerType** { **UNKNOWN** = 0, **PEN**, **PUCK**, **ERASER** }

## Public Member Functions

- **GUIEventAdapter** (const **GUIEventAdapter** &rhs, const osg::CopyOp  
&copyop=osg::CopyOp::SHALLOW\_COPY)
- **GUIEventAdapter** ()
- int **getButton** () const  
*button pressed/released, return -1 if inappropriate for this **GUIEventAdapter** (p. 40).*
- unsigned int **getButtonMask** () const  
*get current mouse button state.*
- virtual **EventType** **getEventType** () const  
*get the event type.*
- const osg::GraphicsContext \* **getGraphicsContext** () const

- bool **getHandled** () const  
*Get whether this event has been handled by an event handler or not.*
- virtual int **getKey** () const  
*get key pressed, return -1 if inappropriate for this **GUIEventAdapter** (p. 40).*
- unsigned int **getModKeyMask** () const  
*get modifier key mask.*
- **MouseYOrientation** **getMouseYOrientation** () const  
*get mouse-Y orientation (mouse-Y increases upwards or downwards).*
- const osg::Matrix **getPenOrientation** () const  
*set the orientation from a tablet input device as a matrix.*
- float **getPenPressure** () const  
*get the tablet pen pressure (range 0..1).*
- float **getPenRotation** () const  
*get the tablet pen rotation around the Z-axis in degrees.*
- float **getPenTiltX** () const  
*get the tablet pen tiltX in degrees.*
- float **getPenTiltY** () const  
*get the tablet pen tiltY in degrees.*
- float **getScrollingDeltaX** () const  
*get the scrolling x-delta.*
- float **getScrollingDeltaY** () const  
*get the scrolling y-delta.*
- **ScrollingMotion** **getScrollingMotion** () const  
*get scrolling motion (for EventType::SCROLL).*
- **TabletPointerType** **getTabletPointerType** () const  
*get the tablet pointer type.*
- double **getTime** () const  
*get time in seconds of event.*
- int **getWindowHeight** () const  
*get window height.*
- int **getWindowWidth** () const  
*get window width.*
- int **getWindowX** () const  
*get window x origin.*
- int **getWindowY** () const  
*get window y origin.*
- float **getX** () const  
*get current mouse x position.*

- float **getXmax** () const  
*get mouse maximum x.*
- float **getXmin** () const  
*get mouse minimum x.*
- float **getXnormalized** () const  
*return the current mouse x value normalized to the range of -1 to 1.*
- float **getY** () const  
*get current mouse y position.*
- float **getYmax** () const  
*get mouse maximum y.*
- float **getYmin** () const  
*get mouse minimum y.*
- float **getYnormalized** () const  
*return the current mouse y value normalized to the range of -1 to 1.*
- **META\_Object** (osgGA, **GUIEventAdapter**)
- void **setButton** (int button)  
*set button pressed/released.*
- void **setButtonMask** (unsigned int mask)  
*set current mouse button state.*
- void **setEventType** (**EventType** Type)  
*set the event type.*
- void **setGraphicsContext** (osg::GraphicsContext \*context)
- void **setHandled** (bool handled) const  
*Set whether this event has been handled by an event handler or not.*
- void **setInputRange** (float Xmin, float Ymin, float Xmax, float Ymax)  
*set mouse input range.*
- void **setKey** (int key)  
*set key pressed.*
- void **setModKeyMask** (unsigned int mask)  
*set modifier key mask.*
- void **setMouseYOrientation** (**MouseYOrientation** myo)  
*set mouse-Y orientation (mouse-Y increases upwards or downwards).*
- void **setPenPressure** (float pressure)  
*set the tablet pen pressure (range 0..1).*
- void **setPenRotation** (float rotation)  
*set the tablet pen rotation around the Z-axis in degrees.*
- void **setPenTiltX** (float tiltX)  
*set the tablet pen tiltX in degrees.*

- void **setPenTiltY** (float tiltY)  
*set the tablet pen tiltY in degrees.*
- void **setScrollingMotion** (**ScrollingMotion** motion)  
*set scrolling motion (for EventType::SCROLL).*
- void **setScrollingMotionDelta** (float x, float y)  
*set the scrolling delta to x,y and the scrolling motion to SCROLL\_2D.*
- void **setTabletPointerType** (**TabletPointerType** pt)  
*set the tablet pointer type.*
- void **setTime** (double time)  
*set time in seconds of event.*
- void **setWindowRectangle** (int x, int y, int width, int height, bool updateMouseRange=true)  
*set window rectangle.*
- void **setX** (float x)  
*set current mouse x position.*
- void **setY** (float y)  
*set current mouse y position.*
- double **time** () const  
*deprecated function for getting time of event.*

### Static Public Member Functions

- static osg::ref\_ptr< **GUIEventAdapter** > & **getAccumulatedEventState** ()  
*Get the accumulated event state singleton.*

### Protected Member Functions

- virtual ~**GUIEventAdapter** ()  
*Force users to create on heap, so that multiple referencing is safe.*

### Protected Attributes

- int **\_button**
- unsigned int **\_buttonMask**
- osg::observer\_ptr< osg::GraphicsContext > **\_context**
- **EventType** **\_eventType**
- bool **\_handled**
- int **\_key**
- unsigned int **\_modKeyMask**
- **MouseYOrientation** **\_mouseYOrientation**
- float **\_mx**
- float **\_my**
- **Scrolling** **\_scrolling**
- **TabletPen** **\_tabletPen**
- double **\_time**
- int **\_windowHeight**

- int `_windowWidth`
- int `_windowX`
- int `_windowY`
- float `_Xmax`
- float `_Xmin`
- float `_Ymax`
- float `_Ymin`

### 4.10.1 Detailed Description

Event class for storing Keyboard, mouse and window events.

### 4.10.2 Member Enumeration Documentation

#### 4.10.2.1 enum EventType

Enumerator:

*NONE*  
*PUSH*  
*RELEASE*  
*DOUBLECLICK*  
*DRAG*  
*MOVE*  
*KEYDOWN*  
*KEYUP*  
*FRAME*  
*RESIZE*  
*SCROLL*  
*PEN\_PRESSURE*  
*PEN\_ORIENTATION*  
*PEN\_PROXIMITY\_ENTER*  
*PEN\_PROXIMITY\_LEAVE*  
*CLOSE\_WINDOW*  
*QUIT\_APPLICATION*  
*USER*

#### 4.10.2.2 enum KeySymbol

Enumerator:

*KEY\_Space*  
*KEY\_BackSpace*  
*KEY\_Tab*  
*KEY\_Linefeed*  
*KEY\_Clear*  
*KEY\_Return*  
*KEY\_Pause*  
*KEY\_Scroll\_Lock*  
*KEY\_Sys\_Req*  
*KEY\_Escape*  
*KEY\_Delete*  
*KEY\_Home*

*KEY\_Left*  
*KEY\_Up*  
*KEY\_Right*  
*KEY\_Down*  
*KEY\_Prior*  
*KEY\_Page\_Up*  
*KEY\_Next*  
*KEY\_Page\_Down*  
*KEY\_End*  
*KEY\_Begin*  
*KEY\_Select*  
*KEY\_Print*  
*KEY\_Execute*  
*KEY\_Insert*  
*KEY\_Undo*  
*KEY\_Redo*  
*KEY\_Menu*  
*KEY\_Find*  
*KEY\_Cancel*  
*KEY\_Help*  
*KEY\_Break*  
*KEY\_Mode\_switch*  
*KEY\_Script\_switch*  
*KEY\_Num\_Lock*  
*KEY\_KP\_Space*  
*KEY\_KP\_Tab*  
*KEY\_KP\_Enter*  
*KEY\_KP\_F1*  
*KEY\_KP\_F2*  
*KEY\_KP\_F3*  
*KEY\_KP\_F4*  
*KEY\_KP\_Home*  
*KEY\_KP\_Left*  
*KEY\_KP\_Up*  
*KEY\_KP\_Right*  
*KEY\_KP\_Down*  
*KEY\_KP\_Prior*  
*KEY\_KP\_Page\_Up*  
*KEY\_KP\_Next*  
*KEY\_KP\_Page\_Down*  
*KEY\_KP\_End*  
*KEY\_KP\_Begin*  
*KEY\_KP\_Insert*  
*KEY\_KP\_Delete*  
*KEY\_KP\_Equal*  
*KEY\_KP\_Multiply*

*KEY\_KP\_Add*  
*KEY\_KP\_Separator*  
*KEY\_KP\_Subtract*  
*KEY\_KP\_Decimal*  
*KEY\_KP\_Divide*  
*KEY\_KP\_0*  
*KEY\_KP\_1*  
*KEY\_KP\_2*  
*KEY\_KP\_3*  
*KEY\_KP\_4*  
*KEY\_KP\_5*  
*KEY\_KP\_6*  
*KEY\_KP\_7*  
*KEY\_KP\_8*  
*KEY\_KP\_9*  
*KEY\_F1*  
*KEY\_F2*  
*KEY\_F3*  
*KEY\_F4*  
*KEY\_F5*  
*KEY\_F6*  
*KEY\_F7*  
*KEY\_F8*  
*KEY\_F9*  
*KEY\_F10*  
*KEY\_F11*  
*KEY\_F12*  
*KEY\_F13*  
*KEY\_F14*  
*KEY\_F15*  
*KEY\_F16*  
*KEY\_F17*  
*KEY\_F18*  
*KEY\_F19*  
*KEY\_F20*  
*KEY\_F21*  
*KEY\_F22*  
*KEY\_F23*  
*KEY\_F24*  
*KEY\_F25*  
*KEY\_F26*  
*KEY\_F27*  
*KEY\_F28*  
*KEY\_F29*  
*KEY\_F30*  
*KEY\_F31*

*KEY\_F32*  
*KEY\_F33*  
*KEY\_F34*  
*KEY\_F35*  
*KEY\_Shift\_L*  
*KEY\_Shift\_R*  
*KEY\_Control\_L*  
*KEY\_Control\_R*  
*KEY\_Caps\_Lock*  
*KEY\_Shift\_Lock*  
*KEY\_Meta\_L*  
*KEY\_Meta\_R*  
*KEY\_Alt\_L*  
*KEY\_Alt\_R*  
*KEY\_Super\_L*  
*KEY\_Super\_R*  
*KEY\_Hyper\_L*  
*KEY\_Hyper\_R*

#### 4.10.2.3 enum ModKeyMask

Enumerator:

*MODKEY\_LEFT\_SHIFT*  
*MODKEY\_RIGHT\_SHIFT*  
*MODKEY\_LEFT\_CTRL*  
*MODKEY\_RIGHT\_CTRL*  
*MODKEY\_LEFT\_ALT*  
*MODKEY\_RIGHT\_ALT*  
*MODKEY\_LEFT\_META*  
*MODKEY\_RIGHT\_META*  
*MODKEY\_LEFT\_SUPER*  
*MODKEY\_RIGHT\_SUPER*  
*MODKEY\_LEFT\_HYPER*  
*MODKEY\_RIGHT\_HYPER*  
*MODKEY\_NUM\_LOCK*  
*MODKEY\_CAPS\_LOCK*  
*MODKEY\_CTRL*  
*MODKEY\_SHIFT*  
*MODKEY\_ALT*  
*MODKEY\_META*  
*MODKEY\_SUPER*  
*MODKEY\_HYPER*

#### 4.10.2.4 enum MouseButtonMask

Enumerator:

*LEFT\_MOUSE\_BUTTON*  
*MIDDLE\_MOUSE\_BUTTON*  
*RIGHT\_MOUSE\_BUTTON*

## 4.10.2.5 enum MouseYOrientation

Enumerator:

*Y\_INCREASING\_UPWARDS*  
*Y\_INCREASING\_DOWNWARDS*

## 4.10.2.6 enum ScrollingMotion

Enumerator:

*SCROLL\_NONE*  
*SCROLL\_LEFT*  
*SCROLL\_RIGHT*  
*SCROLL\_UP*  
*SCROLL\_DOWN*  
*SCROLL\_2D*

## 4.10.2.7 enum TabletPointerType

Enumerator:

*UNKNOWN*  
*PEN*  
*PUCK*  
*ERASER*

## 4.10.3 Constructor &amp; Destructor Documentation

## 4.10.3.1 GUIEventAdapter ()

## 4.10.3.2 GUIEventAdapter (const GUIEventAdapter &amp; rhs, const osg::CopyOp &amp; copyop = osg::CopyOp::SHALLOW\_COPY)

## 4.10.3.3 ~GUIEventAdapter () [protected, virtual]

Force users to create on heap, so that multiple referencing is safe.

## 4.10.4 Member Function Documentation

## 4.10.4.1 osg::ref\_ptr&lt; GUIEventAdapter &gt; &amp;getAccumulatedEventState () [static]

Get the accumulated event state singleton. Typically all **EventQueue** (p.23) will share this single **GUIEventAdapter** (p.40) object for tracking the mouse position, keyboard and mouse masks.

## 4.10.4.2 int getButton () const [inline]

button pressed/released, return -1 if inappropriate for this **GUIEventAdapter** (p.40).

## 4.10.4.3 unsigned int getButtonMask () const [inline]

get current mouse button state.

## 4.10.4.4 virtual EventType getEventType () const [inline, virtual]

get the event type.

## 4.10.4.5 const osg::GraphicsContext\* getGraphicsContext () const [inline]

## 4.10.4.6 bool getHandled () const [inline]

Get whether this event has been handled by an event handler or not.

## 4.10.4.7 virtual int getKey () const [inline, virtual]

get key pressed, return -1 if inappropriate for this **GUIEventAdapter** (p.40).

**4.10.4.8 unsigned int getModKeyMask () const [inline]**

get modifier key mask.

**4.10.4.9 MouseYOrientation getMouseYOrientation () const [inline]**

get mouse-Y orientation (mouse-Y increases upwards or downwards).

**4.10.4.10 const osg::Matrix getPenOrientation () const**

set the orientation from a tablet input device as a matrix.

**4.10.4.11 float getPenPressure () const [inline]**

get the tablet pen pressure (range 0..1).

**4.10.4.12 float getPenRotation () const [inline]**

get the tablet pen rotation around the Z-axis in degrees.

**4.10.4.13 float getPenTiltX () const [inline]**

get the tablet pen tiltX in degrees.

**4.10.4.14 float getPenTiltY () const [inline]**

get the tablet pen tiltY in degrees.

**4.10.4.15 float getScrollingDeltaX () const [inline]**

get the scrolling x-delta.

**4.10.4.16 float getScrollingDeltaY () const [inline]**

get the scrolling y-delta.

**4.10.4.17 ScrollingMotion getScrollingMotion () const [inline]**

get scrolling motion (for EventType::SCROLL).

**4.10.4.18 TabletPointerType getTabletPointerType () const [inline]**

get the tablet pointer type.

**4.10.4.19 double getTime () const [inline]**

get time in seconds of event.

**4.10.4.20 int getWindowHeight () const [inline]**

get window height.

**4.10.4.21 int getWindowWidth () const [inline]**

get window width.

**4.10.4.22 int getWindowX () const [inline]**

get window x origin.

**4.10.4.23 int getWindowY () const [inline]**

get window y origin.

**4.10.4.24 float getX () const [inline]**

get current mouse x position.

**4.10.4.25 float getXmax () const [inline]**

get mouse maximum x.

**4.10.4.26 float getXmin () const [inline]**

get mouse minimum x.

**4.10.4.27 float getXnormalized () const [inline]**

return the current mouse x value normalized to the range of -1 to 1. -1 would be the left hand side of the window. 0.0 would be the middle of the window. +1 would be the right hand side of the window.

**4.10.4.28 float getY () const [inline]**

get current mouse y position.

**4.10.4.29 float getYmax () const [inline]**

get mouse maximum y.

**4.10.4.30 float getYmin () const [inline]**

get mouse minimum y.

**4.10.4.31 float getYnormalized () const [inline]**

return the current mouse y value normalized to the range of -1 to 1. -1 would be the bottom of the window. 0.0 would be the middle of the window. +1 would be the top of the window.

**4.10.4.32 META\_Object (osgGA, GUIEventAdapter)****4.10.4.33 void setButton (int *button*) [inline]**

set button pressed/released.

**4.10.4.34 void setButtonMask (unsigned int *mask*) [inline]**

set current mouse button state.

**4.10.4.35 void setEventType (EventType *Type*) [inline]**

set the event type.

**4.10.4.36 void setGraphicsContext (osg::GraphicsContext \* *context*) [inline]****4.10.4.37 void setHandled (bool *handled*) const [inline]**

Set whether this event has been handled by an event handler or not.

**4.10.4.38 void setInputRange (float *Xmin*, float *Ymin*, float *Xmax*, float *Ymax*)**

set mouse input range.

**4.10.4.39 void setKey (int *key*) [inline]**

set key pressed.

**4.10.4.40 void setModKeyMask (unsigned int *mask*) [inline]**

set modifier key mask.

**4.10.4.41 void setMouseYOrientation (MouseYOrientation *myo*) [inline]**

set mouse-Y orientation (mouse-Y increases upwards or downwards).

**4.10.4.42 void setPenPressure (float *pressure*) [inline]**

set the tablet pen pressure (range 0..1).

**4.10.4.43 void setPenRotation (float *rotation*) [inline]**

set the tablet pen rotation around the Z-axis in degrees.

**4.10.4.44 void setPenTiltX (float *tiltX*) [inline]**

set the tablet pen tiltX in degrees.

**4.10.4.45 void setPenTiltY (float *tiltY*) [inline]**

set the tablet pen tiltY in degrees.

**4.10.4.46 void setScrollingMotion (ScrollingMotion *motion*) [inline]**

set scrolling motion (for EventType::SCROLL).

**4.10.4.47 void setScrollingMotionDelta (float *x*, float *y*) [inline]**

set the scrolling delta to *x*,*y* and the scrolling motion to SCROLL\_2D.

**4.10.4.48 void setTabletPointerType (TabletPointerType *pt*) [inline]**

set the tablet pointer type.

**4.10.4.49 void setTime (double *time*) [inline]**

set time in seconds of event.

**4.10.4.50 void setWindowRectangle (int *x*, int *y*, int *width*, int *height*, bool *updateMouseRange* = true)**

set window rectangle.

**4.10.4.51 void setX (float *x*) [inline]**

set current mouse *x* position.

**4.10.4.52 void setY (float *y*) [inline]**

set current mouse *y* position.

**4.10.4.53 double time () const [inline]**

deprecated function for getting time of event.

## 4.10.5 Member Data Documentation

**4.10.5.1 int \_button [protected]**

**4.10.5.2 unsigned int \_buttonMask [protected]**

**4.10.5.3 osg::observer\_ptr<osg::GraphicsContext> \_context [protected]**

**4.10.5.4 EventType \_eventType [protected]**

**4.10.5.5 bool \_handled [mutable, protected]**

**4.10.5.6 int \_key [protected]**

**4.10.5.7 unsigned int \_modKeyMask [protected]**

**4.10.5.8 MouseYOrientation \_mouseYOrientation [protected]**

**4.10.5.9 float \_mx [protected]**

**4.10.5.10 float \_my [protected]**

**4.10.5.11 Scrolling \_scrolling [protected]**

**4.10.5.12 TabletPen \_tabletPen [protected]**

**4.10.5.13 double \_time [protected]**

**4.10.5.14 int \_windowHeight [protected]**

**4.10.5.15 int \_windowWidth [protected]**

**4.10.5.16 int \_windowX [protected]**

**4.10.5.17 int \_windowY [protected]**

**4.10.5.18 float \_Xmax [protected]**

**4.10.5.19 float \_Xmin [protected]**

**4.10.5.20 float \_Ymax [protected]**

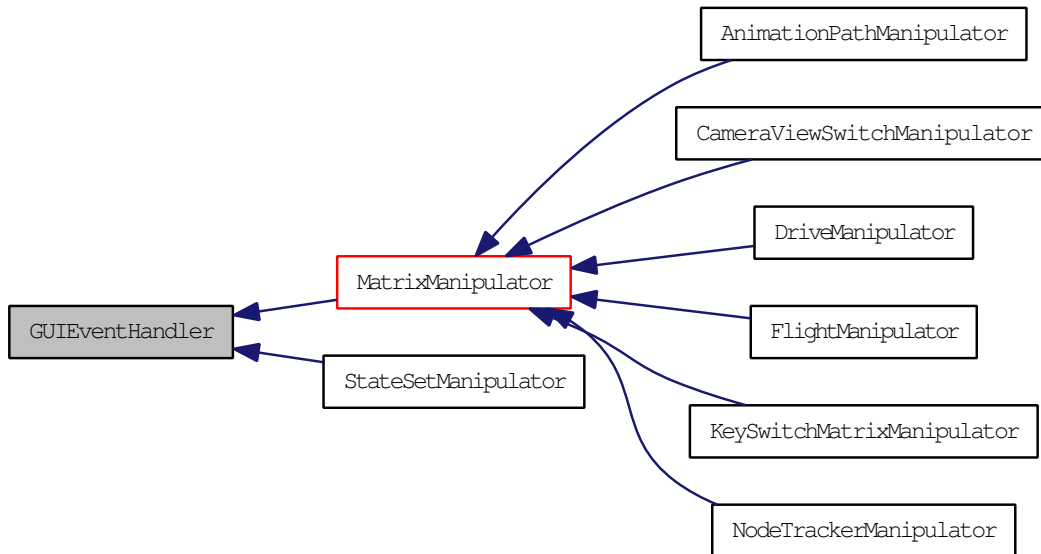
**4.10.5.21 float \_Ymin [protected]**

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

- GUIEventAdapter
- GUIEventAdapter.cpp

## 4.11 GUIEventHandler Class Reference

**GUIEventHandler** (p. 54) provides a basic interface for any class which wants to handle a GUI Events. Inheritance diagram for GUIEventHandler:



### Public Member Functions

- **GUIEventHandler** (const **GUIEventHandler** &eh, const osg::CopyOp &)
- **GUIEventHandler** ()
- virtual void **event** (osg::NodeVisitor \*nv, osg::Drawable \*drawable)  
*Event traversal drawable callback method.*
- unsigned int **getIgnoreHandledEventsMask** () const  
*Get the event mask of the osgGA::GUIEventAdapter::Event to be ignored if marked as handled.*
- virtual void **getUsage** (osg::ApplicationUsage &) const  
*Get the keyboard and mouse usage of this manipulator.*
- virtual bool **handle** (const **GUIEventAdapter** &, **GUIActionAdapter** &)  
*Deprecated, Handle events, return true if handled, false otherwise.*
- virtual bool **handle** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &aa, osg::Object \*, osg::NodeVisitor \*)  
*Handle events, return true if handled, false otherwise.*
- bool **handleWithCheckAgainstIgnoreHandledEventsMask** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &aa)  
*Convenience method that only passes on to the handle(,) method events that either haven't been handled yet, or have been handled but haven't be set to be ignored by the IgnoreHandledEventsMask.*
- bool **handleWithCheckAgainstIgnoreHandledEventsMask** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &aa, osg::Object \*object, osg::NodeVisitor \*nv)  
*Convenience method that only passes on to the handle(,,) method events that either haven't been handled yet, or have been handled but haven't be set to be ignored by the IgnoreHandledEventsMask.*
- **META\_Object** (osgGA, **GUIEventHandler**)
- virtual void **operator()** (osg::Node \*node, osg::NodeVisitor \*nv)  
*Event traversal node callback method.*

- void **setIgnoreHandledEventsMask** (unsigned int mask)  
Set a mask of `osgGA::GUIEventAdapter::Event` to be ignored if marked as handled.

## Protected Attributes

- unsigned int **\_ignoreHandledEventsMask**

### 4.11.1 Detailed Description

**GUIEventHandler** (p. 54) provides a basic interface for any class which wants to handle a GUI Events. The GUIEvent is supplied by a **GUIEventAdapter** (p. 40). Feedback resulting from the handle method is supplied by a **GUIActionAdapter** (p. 38), which allows the **GUIEventHandler** (p. 54) to ask the GUI to take some action in response to an incoming event.

For example, consider a Trackball Viewer class which takes mouse events and manipulates a scene camera in response. The Trackball Viewer is a **GUIEventHandler** (p. 54), and receives the events via the handle method. If the user 'throws' the model, the Trackball Viewer class can detect this via the incoming events, and request that the GUI set up a timer callback to continually redraw the view. This request is made via the **GUIActionAdapter** (p. 38) class.

### 4.11.2 Constructor & Destructor Documentation

4.11.2.1 **GUIEventHandler ()** [inline]

4.11.2.2 **GUIEventHandler (const GUIEventHandler & eh, const osg::CopyOp &)** [inline]

### 4.11.3 Member Function Documentation

4.11.3.1 **void event (osg::NodeVisitor \* nv, osg::Drawable \* drawable)** [virtual]

Event traversal drawable callback method.

4.11.3.2 **unsigned int getIgnoreHandledEventsMask () const** [inline]

Get the event mask of the `osgGA::GUIEventAdapter::Event` to be ignored if marked as handled.

4.11.3.3 **virtual void getUsage (osg::ApplicationUsage &) const** [inline, virtual]

Get the keyboard and mouse usage of this manipulator.

Reimplemented in **AnimationPathManipulator** (p. 11), **CameraViewSwitchManipulator** (p. 15), **DriveManipulator** (p. 21), **FlightManipulator** (p. 36), **KeySwitchMatrixManipulator** (p. 60), **NodeTrackerManipulator** (p. 72), **SphericalManipulator** (p. 79), **StateSetManipulator** (p. 84), **TerrainManipulator** (p. 90), **TrackballManipulator** (p. 96), and **UFOManipulator** (p. 103).

4.11.3.4 **virtual bool handle (const GUIEventAdapter &, GUIActionAdapter &)** [inline, virtual]

Deprecated, Handle events, return true if handled, false otherwise.

Reimplemented in **AnimationPathManipulator** (p. 11), **CameraViewSwitchManipulator** (p. 15), **DriveManipulator** (p. 21), **FlightManipulator** (p. 36), **KeySwitchMatrixManipulator** (p. 60), **MatrixManipulator** (p. 65), **NodeTrackerManipulator** (p. 72), **SphericalManipulator** (p. 79), **StateSetManipulator** (p. 84), **TerrainManipulator** (p. 90), **TrackballManipulator** (p. 96), and **UFOManipulator** (p. 103).

4.11.3.5 **virtual bool handle (const GUIEventAdapter & ea, GUIActionAdapter & aa, osg::Object \*, osg::NodeVisitor \*)** [inline, virtual]

Handle events, return true if handled, false otherwise.

4.11.3.6 **bool handleWithCheckAgainstIgnoreHandledEventsMask (const GUIEventAdapter & ea, GUIActionAdapter & aa)** [inline]

Convenience method that only passes on to the `handle(.)` method events that either haven't been handled yet, or have been handled but haven't be set to be ignored by the `IgnoreHandledEventsMask`. Note, this method is an inline method, and not appropriate for users to override, override the `handle(.)` method instead.

#### 4.11.3.7 **bool handleWithCheckAgainstIgnoreHandledEventsMask (const GUIEventAdapter & *ea*, GUIActionAdapter & *aa*, osg::Object \* *object*, osg::NodeVisitor \* *nv*) [inline]**

Convenience method that only passes on to the handle(,,) method events that either haven't been handled yet, or have been handled but haven't be set to be ignored by the IgnoreHandledEventsMask. Note, this method is an inline method, and not appropriate for users to override, override the handle(,,) method instead.

#### 4.11.3.8 **META\_Object (osgGA, GUIEventHandler)**

#### 4.11.3.9 **void operator() (osg::Node \* *node*, osg::NodeVisitor \* *nv*) [virtual]**

Event traversal node callback method.

#### 4.11.3.10 **void setIgnoreHandledEventsMask (unsigned int *mask*) [inline]**

Set a mask of osgGA::GUIEventAdapter::Event to be ignored if marked as handled.

### 4.11.4 Member Data Documentation

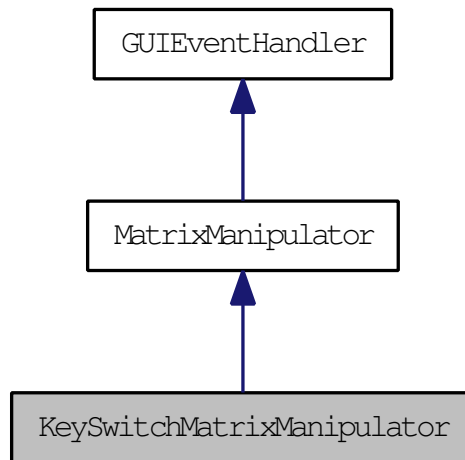
#### 4.11.4.1 **unsigned int \_ignoreHandledEventsMask [protected]**

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

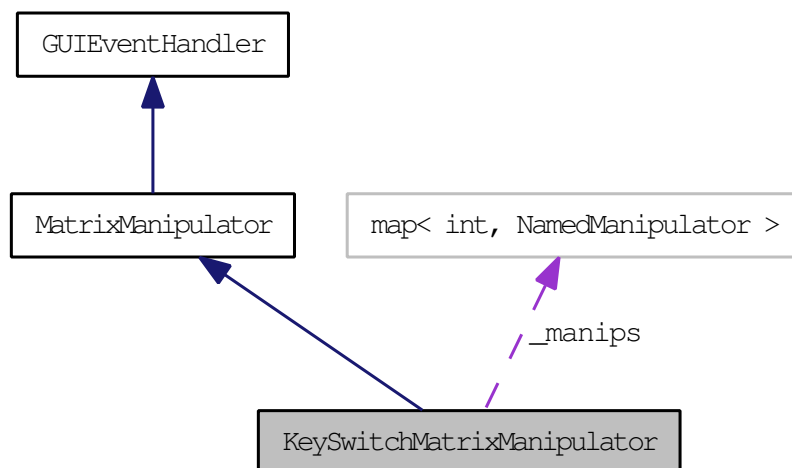
- GUIEventHandler
- GUIEventHandler.cpp

## 4.12 KeySwitchMatrixManipulator Class Reference

**KeySwitchMatrixManipulator** (p. 57) is a decorator which allows the type of camera manipulator being used to be switched by pressing a key. Inheritance diagram for KeySwitchMatrixManipulator:



Collaboration diagram for KeySwitchMatrixManipulator:



### Public Types

- typedef `std::map< int, NamedManipulator >` **KeyManipMap**
- typedef `std::pair< std::string, osg::ref_ptr< MatrixManipulator > >` **NamedManipulator**

### Public Member Functions

- void **addMatrixManipulator** (int key, std::string name, **MatrixManipulator** \*cm)  
*Add a camera manipulator with an associated name, and a key to trigger the switch,.*
- void **addNumberedMatrixManipulator** (**MatrixManipulator** \*cm)  
*Add a camera manipulator with an autogenerated keybinding which is '1' + previous number of camera's registered.*
- virtual const char \* **className** () const
- virtual void **computeHomePosition** ()  
*Compute the home position.*

- const **MatrixManipulator \* getCurrentMatrixManipulator ()** const  
*Get the const current active manipulators.*
- **MatrixManipulator \* getCurrentMatrixManipulator ()**  
*Get the current active manipulators.*
- virtual osgUtil::SceneView::FusionDistanceMode **getFusionDistanceMode ()** const  
*Get the FusionDistanceMode.*
- virtual float **getFusionDistanceValue ()** const  
*Get the FusionDistanceValue.*
- virtual osg::Matrixd **getInverseMatrix ()** const  
*get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.*
- const **KeyManipMap & getKeyManipMap ()** const  
*Get the const complete list of manipulators attached to this keyswitch manipulator.*
- **KeyManipMap & getKeyManipMap ()**  
*Get the complete list of manipulators attached to this keyswitch manipulator.*
- virtual osg::Matrixd **getMatrix ()** const  
*get the position of the manipulator as 4x4 Matrix.*
- const **MatrixManipulator \* getMatrixManipulatorWithIndex** (unsigned int key) const  
*Get const manipulator assigned to a specified index.*
- **MatrixManipulator \* getMatrixManipulatorWithIndex** (unsigned int key)  
*Get manipulator assigned to a specified index.*
- const **MatrixManipulator \* getMatrixManipulatorWithKey** (unsigned int key) const  
*Get const manipulator assigned to a specified key.*
- **MatrixManipulator \* getMatrixManipulatorWithKey** (unsigned int key)  
*Get manipulator assigned to a specified key.*
- virtual osg::Node \* **getNode ()**  
*Return node if attached.*
- virtual const osg::Node \* **getNode ()** const  
*Return const node if attached.*
- unsigned int **getNumMatrixManipulators ()** const
- virtual void **getUsage** (osg::ApplicationUsage &usage) const  
*Get the keyboard and mouse usage of this manipulator.*
- virtual bool **handle** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Handle events, return true if handled, false otherwise.*
- virtual void **home** (const **GUIEventAdapter** &ee, **GUIActionAdapter** &aa)  
*Move the camera to the default position.*
- virtual void **init** (const **GUIEventAdapter** &ee, **GUIActionAdapter** &aa)  
*Start/restart the manipulator.*
- void **selectMatrixManipulator** (unsigned int num)

- virtual void **setAutoComputeHomePosition** (bool flag)  
*Set whether the automatic compute of the home position is enabled.*
- virtual void **setByInverseMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setByMatrix** (const osg::Matrixd &matrix)  
*Set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setCoordinateFrameCallback** (**CoordinateFrameCallback** \*cb)  
*set the coordinate frame which callback tells the manipulator which way is up, east and north.*
- virtual void **setHomePosition** (const osg::Vec3d &eye, const osg::Vec3d &center, const osg::Vec3d &up, bool autoComputeHomePosition=false)  
*Manually set the home position, and set the automatic compute of home position.*
- virtual void **setMinimumDistance** (float minimumDistance)  
*set the minimum distance (as ratio) the eye point can be zoomed in towards the center before the center is pushed forward.*
- virtual void **setNode** (osg::Node \*n)  
*Attach a node to the manipulator, automatically detaching any previously attached node.*

### 4.12.1 Detailed Description

**KeySwitchMatrixManipulator** (p. 57) is a decorator which allows the type of camera manipulator being used to be switched by pressing a key. E.g. '1' for a **TrackballManipulator**, '2' for a **DriveManipulator** (p. 18), '3' for a **FlightManipulator** (p. 33). The manipulators available, and the associated switch keys, can be configured.

### 4.12.2 Member Typedef Documentation

4.12.2.1 **typedef std::map<int, NamedManipulator> KeyManipMap**

4.12.2.2 **typedef std::pair<std::string, osg::ref\_ptr<MatrixManipulator> > NamedManipulator**

### 4.12.3 Member Function Documentation

4.12.3.1 **void addMatrixManipulator (int key, std::string name, MatrixManipulator \* cm)**

Add a camera manipulator with an associated name, and a key to trigger the switch,.

4.12.3.2 **void addNumberedMatrixManipulator (MatrixManipulator \* cm)**

Add a camera manipulator with an autogenerated keybinding which is '1' + previous number of camera's registered.

4.12.3.3 **virtual const char\* className () const [inline, virtual]**

Reimplemented from **MatrixManipulator** (p. 64).

4.12.3.4 **void computeHomePosition () [virtual]**

Compute the home position.

Reimplemented from **MatrixManipulator** (p. 64).

4.12.3.5 **const MatrixManipulator\* getCurrentMatrixManipulator () const [inline]**

Get the const current active manipulators.

4.12.3.6 **MatrixManipulator\* getCurrentMatrixManipulator () [inline]**

Get the current active manipulators.

**4.12.3.7 virtual osgUtil::SceneView::FusionDistanceMode getFusionDistanceMode () const [inline, virtual]**

Get the FusionDistanceMode. Used by SceneView for setting up stereo convergence.

Reimplemented from **MatrixManipulator** (p. 64).

**4.12.3.8 virtual float getFusionDistanceValue () const [inline, virtual]**

Get the FusionDistanceValue. Used by SceneView for setting up stereo convergence.

Reimplemented from **MatrixManipulator** (p. 65).

**4.12.3.9 virtual osg::Matrixd getInverseMatrix () const [inline, virtual]**

get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.

Implements **MatrixManipulator** (p. 65).

**4.12.3.10 const KeyManipMap& getKeyManipMap () const [inline]**

Get the const complete list of manipulators attached to this keyswitch manipulator.

**4.12.3.11 KeyManipMap& getKeyManipMap () [inline]**

Get the complete list of manipulators attached to this keyswitch manipulator.

**4.12.3.12 virtual osg::Matrixd getMatrix () const [inline, virtual]**

get the position of the manipulator as 4x4 Matrix.

Implements **MatrixManipulator** (p. 65).

**4.12.3.13 const MatrixManipulator \* getMatrixManipulatorWithIndex (unsigned int key) const**

Get const manipulator assigned to a specified index.

**4.12.3.14 MatrixManipulator \* getMatrixManipulatorWithIndex (unsigned int key)**

Get manipulator assigned to a specified index.

**4.12.3.15 const MatrixManipulator \* getMatrixManipulatorWithKey (unsigned int key) const**

Get const manipulator assigned to a specified key.

**4.12.3.16 MatrixManipulator \* getMatrixManipulatorWithKey (unsigned int key)**

Get manipulator assigned to a specified key.

**4.12.3.17 virtual osg::Node\* getNode () [inline, virtual]**

Return node if attached.

Reimplemented from **MatrixManipulator** (p. 65).

**4.12.3.18 virtual const osg::Node\* getNode () const [inline, virtual]**

Return const node if attached.

Reimplemented from **MatrixManipulator** (p. 65).

**4.12.3.19 unsigned int getNumMatrixManipulators () const [inline]****4.12.3.20 void getUsage (osg::ApplicationUsage & usage) const [virtual]**

Get the keyboard and mouse usage of this manipulator.

Reimplemented from **GUIEventHandler** (p. 55).

**4.12.3.21 bool handle (const GUIEventAdapter & ea, GUIActionAdapter & ua) [virtual]**

Handle events, return true if handled, false otherwise.

Reimplemented from **MatrixManipulator** (p. 65).

**4.12.3.22 virtual void home (const GUIEventAdapter &, GUIActionAdapter &) [inline, virtual]**

Move the camera to the default position. May be ignored by manipulators if home functionality is not appropriate.

Reimplemented from **MatrixManipulator** (p. 66).

**4.12.3.23 virtual void init (const GUIEventAdapter &, GUIActionAdapter &) [inline, virtual]**

Start/restart the manipulator. FIXME: what does this actually mean? Provide examples.

Reimplemented from **MatrixManipulator** (p. 66).

**4.12.3.24 void selectMatrixManipulator (unsigned int *num*)****4.12.3.25 void setAutoComputeHomePosition (bool *flag*) [virtual]**

Set whether the automatic compute of the home position is enabled.

Reimplemented from **MatrixManipulator** (p. 66).

**4.12.3.26 virtual void setByInverseMatrix (const osg::Matrixd & *matrix*) [inline, virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.12.3.27 virtual void setByMatrix (const osg::Matrixd & *matrix*) [inline, virtual]**

Set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.12.3.28 void setCoordinateFrameCallback (CoordinateFrameCallback \* *cb*) [virtual]**

set the coordinate frame which callback tells the manipulator which way is up, east and north.

Reimplemented from **MatrixManipulator** (p. 66).

**4.12.3.29 void setHomePosition (const osg::Vec3d & *eye*, const osg::Vec3d & *center*, const osg::Vec3d & *up*, bool *autoComputeHomePosition* = false) [virtual]**

Manually set the home position, and set the automatic compute of home position.

Reimplemented from **MatrixManipulator** (p. 66).

**4.12.3.30 void setMinimumDistance (float *minimumDistance*) [virtual]**

set the minimum distance (as ratio) the eye point can be zoomed in towards the center before the center is pushed forward.

Reimplemented from **MatrixManipulator** (p. 67).

**4.12.3.31 void setNode (osg::Node \*) [virtual]**

Attach a node to the manipulator, automatically detaching any previously attached node. setNode(NULL) detaches previous nodes. May be ignored by manipulators which do not require a reference model.

Reimplemented from **MatrixManipulator** (p. 67).

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

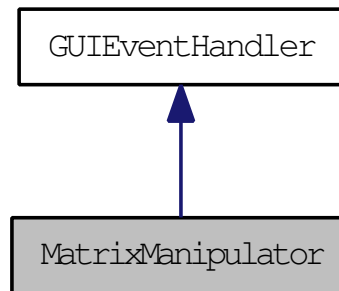
- **KeySwitchMatrixManipulator**
- **KeySwitchMatrixManipulator.cpp**

## 4.13 MatrixManipulator Class Reference

**MatrixManipulator** (p. 62) is an abstract base class defining the interface, and a certain amount of default functionality, for classes which wish to control OSG cameras in response to GUI events.

Inherits **osgGA::GUIEventHandler**.

Inherited by **AnimationPathManipulator**, **CameraViewSwitchManipulator**, **DriveManipulator**, **FlightManipulator**, **KeySwitchMatrixManipulator**, **NodeTrackerManipulator**, **SphericalManipulator**, **TerrainManipulator**, **TrackballManipulator**, and **UFOManipulator**. Collaboration diagram for MatrixManipulator:



### Classes

- class **CoordinateFrameCallback**  
*callback class to use to allow matrix manipulators to query the application for the local coordinate frame.*

### Public Member Functions

- virtual const char \* **className** () const
- virtual void **computeHomePosition** ()  
*Compute the home position.*
- bool **getAutoComputeHomePosition** () const  
*Get whether the automatic compute of the home position is enabled.*
- osg::CoordinateFrame **getCoordinateFrame** (const osg::Vec3d &position) const  
*get the coordinate frame.*
- const **CoordinateFrameCallback** \* **getCoordinateFrameCallback** () const  
*get the coordinate frame callback which tells the manipulator which way is up, east and north.*
- **CoordinateFrameCallback** \* **getCoordinateFrameCallback** ()  
*get the coordinate frame callback which tells the manipulator which way is up, east and north.*
- osg::Vec3d **getFrontVector** (const osg::CoordinateFrame &cf) const
- virtual osgUtil::SceneView::FusionDistanceMode **getFusionDistanceMode** () const  
*Get the FusionDistanceMode.*
- virtual float **getFusionDistanceValue** () const  
*Get the FusionDistanceValue.*
- virtual void **getHomePosition** (osg::Vec3d &eye, osg::Vec3d &center, osg::Vec3d &up) const  
*Get the manually set home position.*
- unsigned int **getIntersectTraversalMask** () const

*Get the mask to use when set up intersection traversal such as used in manipulators that follow terrain or have collision detection.*

- virtual osg::Matrixd **getInverseMatrix** () const =0  
*get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.*
- virtual osg::Matrixd **getMatrix** () const =0  
*get the position of the manipulator as 4x4 Matrix.*
- float **getMinimumDistance** () const  
*get the minimum distance (as ratio) the eye point can be zoomed in*
- virtual osg::Node \* **getNode** ()  
*Return node if attached.*
- virtual const osg::Node \* **getNode** () const  
*Return const node if attached.*
- osg::Vec3d **getSideVector** (const osg::CoordinateFrame &cf) const
- osg::Vec3d **getUpVector** (const osg::CoordinateFrame &cf) const
- virtual bool **handle** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Handle events, return true if handled, false otherwise.*
- virtual void **home** (double)  
*Move the camera to the default position.*
- virtual void **home** (const **GUIEventAdapter** &, **GUIActionAdapter** &)  
*Move the camera to the default position.*
- virtual void **init** (const **GUIEventAdapter** &, **GUIActionAdapter** &)  
*Start/restart the manipulator.*
- virtual void **setAutoComputeHomePosition** (bool flag)  
*Set whether the automatic compute of the home position is enabled.*
- virtual void **setByInverseMatrix** (const osg::Matrixd &matrix)=0  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setByMatrix** (const osg::Matrixd &matrix)=0  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setCoordinateFrameCallback** (**CoordinateFrameCallback** \*cb)  
*set the coordinate frame which callback tells the manipulator which way is up, east and north.*
- virtual void **setHomePosition** (const osg::Vec3d &eye, const osg::Vec3d &center, const osg::Vec3d &up, bool autoComputeHomePosition=false)  
*Manually set the home position, and set the automatic compute of home position.*
- void **setIntersectTraversalMask** (unsigned int mask)  
*Set the mask to use when set up intersection traversal such as used in manipulators that follow terrain or have collision detection.*
- virtual void **setMinimumDistance** (float minimumDistance)  
*set the minimum distance (as ratio) the eye point can be zoomed in towards the center before the center is pushed forward.*
- virtual void **setNode** (osg::Node \*)  
*Attach a node to the manipulator, automatically detaching any previously attached node.*

## Protected Member Functions

- **MatrixManipulator** ()
- virtual **~MatrixManipulator** ()

## Protected Attributes

- bool **\_autoComputeHomePosition**
- osg::ref\_ptr< **CoordinateFrameCallback** > **\_coordinateFrameCallback**
- osg::Vec3d **\_homeCenter**
- osg::Vec3d **\_homeEye**
- osg::Vec3d **\_homeUp**
- unsigned int **\_intersectTraversalMask**
- double **\_minimumDistance**

### 4.13.1 Detailed Description

**MatrixManipulator** (p. 62) is an abstract base class defining the interface, and a certain amount of default functionality, for classes which wish to control OSG cameras in response to GUI events.

### 4.13.2 Constructor & Destructor Documentation

**4.13.2.1 MatrixManipulator** () [protected]

**4.13.2.2 ~MatrixManipulator** () [protected, virtual]

### 4.13.3 Member Function Documentation

**4.13.3.1 virtual const char\* className** () const [inline, virtual]

Reimplemented in **AnimationPathManipulator** (p. 11), **CameraViewSwitchManipulator** (p. 14), **DriveManipulator** (p. 20), **FlightManipulator** (p. 35), **KeySwitchMatrixManipulator** (p. 59), **NodeTrackerManipulator** (p. 71), **SphericalManipulator** (p. 78), **TerrainManipulator** (p. 89), **TrackballManipulator** (p. 95), and **UFOManipulator** (p. 102).

**4.13.3.2 virtual void computeHomePosition** () [inline, virtual]

Compute the home position.

Reimplemented in **DriveManipulator** (p. 20), **KeySwitchMatrixManipulator** (p. 59), **NodeTrackerManipulator** (p. 71), **SphericalManipulator** (p. 78), and **UFOManipulator** (p. 102).

**4.13.3.3 bool getAutoComputeHomePosition** () const [inline]

Get whether the automatic compute of the home position is enabled.

**4.13.3.4 osg::CoordinateFrame getCoordinateFrame** (const osg::Vec3d & *position*) const [inline]

get the coordinate frame.

**4.13.3.5 const CoordinateFrameCallback\* getCoordinateFrameCallback** () const [inline]

get the coordinate frame callback which tells the manipulator which way is up, east and north.

**4.13.3.6 CoordinateFrameCallback\* getCoordinateFrameCallback** () [inline]

get the coordinate frame callback which tells the manipulator which way is up, east and north.

**4.13.3.7 osg::Vec3d getFrontVector** (const osg::CoordinateFrame & *cf*) const [inline]

**4.13.3.8 virtual osgUtil::SceneView::FusionDistanceMode getFusionDistanceMode** () const [inline, virtual]

Get the FusionDistanceMode. Used by SceneView for setting up stereo convergence.

Reimplemented in **KeySwitchMatrixManipulator** (p. 60), **NodeTrackerManipulator** (p. 71), **SphericalManipulator** (p. 78), **TerrainManipulator** (p. 89), and **TrackballManipulator** (p. 95).

**4.13.3.9 virtual float getFusionDistanceValue () const [inline, virtual]**

Get the FusionDistanceValue. Used by SceneView for setting up stereo convergence.

Reimplemented in **KeySwitchMatrixManipulator** (p. 60), **NodeTrackerManipulator** (p. 71), **SphericalManipulator** (p. 78), **TerrainManipulator** (p. 89), and **TrackballManipulator** (p. 95).

**4.13.3.10 virtual void getHomePosition (osg::Vec3d & eye, osg::Vec3d & center, osg::Vec3d & up) const [inline, virtual]**

Get the manually set home position.

**4.13.3.11 unsigned int getIntersectTraversalMask () const [inline]**

Get the mask to use when set up intersection traversal such as used in manipulators that follow terrain or have collision detection.

**4.13.3.12 virtual osg::Matrixd getInverseMatrix () const [pure virtual]**

get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.

Implemented in **AnimationPathManipulator** (p. 11), **CameraViewSwitchManipulator** (p. 14), **DriveManipulator** (p. 21), **FlightManipulator** (p. 35), **KeySwitchMatrixManipulator** (p. 60), **NodeTrackerManipulator** (p. 71), **SphericalManipulator** (p. 79), **TerrainManipulator** (p. 89), **TrackballManipulator** (p. 95), and **UFOManipulator** (p. 102).

**4.13.3.13 virtual osg::Matrixd getMatrix () const [pure virtual]**

get the position of the manipulator as 4x4 Matrix.

Implemented in **AnimationPathManipulator** (p. 11), **CameraViewSwitchManipulator** (p. 14), **DriveManipulator** (p. 21), **FlightManipulator** (p. 35), **KeySwitchMatrixManipulator** (p. 60), **NodeTrackerManipulator** (p. 71), **SphericalManipulator** (p. 79), **TerrainManipulator** (p. 90), **TrackballManipulator** (p. 95), and **UFOManipulator** (p. 102).

**4.13.3.14 float getMinimumDistance () const [inline]**

get the minimum distance (as ratio) the eye point can be zoomed in

**4.13.3.15 virtual osg::Node\* getNode () [inline, virtual]**

Return node if attached.

Reimplemented in **CameraViewSwitchManipulator** (p. 14), **DriveManipulator** (p. 21), **FlightManipulator** (p. 35), **KeySwitchMatrixManipulator** (p. 60), **NodeTrackerManipulator** (p. 71), **SphericalManipulator** (p. 79), **TerrainManipulator** (p. 90), **TrackballManipulator** (p. 95), and **UFOManipulator** (p. 102).

**4.13.3.16 virtual const osg::Node\* getNode () const [inline, virtual]**

Return const node if attached.

Reimplemented in **CameraViewSwitchManipulator** (p. 15), **DriveManipulator** (p. 21), **FlightManipulator** (p. 36), **KeySwitchMatrixManipulator** (p. 60), **NodeTrackerManipulator** (p. 71), **SphericalManipulator** (p. 79), **TerrainManipulator** (p. 90), **TrackballManipulator** (p. 96), and **UFOManipulator** (p. 103).

**4.13.3.17 osg::Vec3d getSideVector (const osg::CoordinateFrame & cf) const [inline]****4.13.3.18 osg::Vec3d getUpVector (const osg::CoordinateFrame & cf) const [inline]****4.13.3.19 bool handle (const GUIEventAdapter & ea, GUIActionAdapter & us) [virtual]**

Handle events, return true if handled, false otherwise.

Reimplemented from **GUIEventHandler** (p. 55).

Reimplemented in **AnimationPathManipulator** (p. 11), **CameraViewSwitchManipulator** (p. 15), **DriveManipulator** (p. 21), **FlightManipulator** (p. 36), **KeySwitchMatrixManipulator** (p. 60), **NodeTrackerManipulator** (p. 72), **SphericalManipulator** (p. 79), **TerrainManipulator** (p. 90), **TrackballManipulator** (p. 96), and **UFOManipulator** (p. 103).

**4.13.3.20 virtual void home (double) [inline, virtual]**

Move the camera to the default position. This version does not require **GUIEventAdapter** (p. 40) and **GUIActionAdapter** (p. 38) so may be called from somewhere other than a **handle()** (p. 65) method in **GUIEventHandler** (p. 54). Application must be aware of implications.

Reimplemented in **AnimationPathManipulator** (p. 11), **SphericalManipulator** (p. 79), **TrackballManipulator** (p. 96), and **UFOManipulator** (p. 103).

**4.13.3.21 virtual void home (const GUIEventAdapter &, GUIActionAdapter &) [inline, virtual]**

Move the camera to the default position. May be ignored by manipulators if home functionality is not appropriate.

Reimplemented in **AnimationPathManipulator** (p. 11), **DriveManipulator** (p. 21), **FlightManipulator** (p. 36), **KeySwitchMatrixManipulator** (p. 61), **NodeTrackerManipulator** (p. 72), **SphericalManipulator** (p. 79), **TerrainManipulator** (p. 90), **TrackballManipulator** (p. 96), and **UFOManipulator** (p. 103).

**4.13.3.22 virtual void init (const GUIEventAdapter &, GUIActionAdapter &) [inline, virtual]**

Start/restart the manipulator. FIXME: what does this actually mean? Provide examples.

Reimplemented in **AnimationPathManipulator** (p. 11), **CameraViewSwitchManipulator** (p. 15), **DriveManipulator** (p. 21), **FlightManipulator** (p. 36), **KeySwitchMatrixManipulator** (p. 61), **NodeTrackerManipulator** (p. 72), **SphericalManipulator** (p. 79), **TerrainManipulator** (p. 90), **TrackballManipulator** (p. 96), and **UFOManipulator** (p. 103).

**4.13.3.23 virtual void setAutoComputeHomePosition (bool flag) [inline, virtual]**

Set whether the automatic compute of the home position is enabled.

Reimplemented in **KeySwitchMatrixManipulator** (p. 61).

**4.13.3.24 virtual void setByInverseMatrix (const osg::Matrixd & matrix) [pure virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implemented in **AnimationPathManipulator** (p. 11), **CameraViewSwitchManipulator** (p. 15), **DriveManipulator** (p. 21), **FlightManipulator** (p. 36), **KeySwitchMatrixManipulator** (p. 61), **NodeTrackerManipulator** (p. 72), **SphericalManipulator** (p. 80), **TerrainManipulator** (p. 90), **TrackballManipulator** (p. 96), and **UFOManipulator** (p. 103).

**4.13.3.25 virtual void setByMatrix (const osg::Matrixd & matrix) [pure virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implemented in **AnimationPathManipulator** (p. 11), **CameraViewSwitchManipulator** (p. 15), **DriveManipulator** (p. 21), **FlightManipulator** (p. 36), **KeySwitchMatrixManipulator** (p. 61), **NodeTrackerManipulator** (p. 72), **SphericalManipulator** (p. 80), **TerrainManipulator** (p. 90), **TrackballManipulator** (p. 96), and **UFOManipulator** (p. 104).

**4.13.3.26 virtual void setCoordinateFrameCallback (CoordinateFrameCallback \* cb) [inline, virtual]**

set the coordinate frame which callback tells the manipulator which way is up, east and north.

Reimplemented in **KeySwitchMatrixManipulator** (p. 61).

**4.13.3.27 virtual void setHomePosition (const osg::Vec3d & eye, const osg::Vec3d & center, const osg::Vec3d & up, bool autoComputeHomePosition = false) [inline, virtual]**

Manually set the home position, and set the automatic compute of home position.

Reimplemented in **KeySwitchMatrixManipulator** (p. 61).

**4.13.3.28 void setIntersectTraversalMask (unsigned int mask) [inline]**

Set the mask to use when set up intersection traversal such as used in manipulators that follow terrain or have collision detection. The intersection traversal mask is useful for controlling what parts of the scene graph should be used for intersection purposes.

**4.13.3.29 virtual void setMinimumDistance (float *minimumDistance*) [inline, virtual]**

set the minimum distance (as ratio) the eye point can be zoomed in towards the center before the center is pushed forward.

Reimplemented in **KeySwitchMatrixManipulator** (p. 61).

**4.13.3.30 virtual void setNode (osg::Node \*) [inline, virtual]**

Attach a node to the manipulator, automatically detaching any previously attached node. setNode(NULL) detaches previous nodes. May be ignored by manipulators which do not require a reference model.

Reimplemented in **CameraViewSwitchManipulator** (p. 15), **DriveManipulator** (p. 22), **FlightManipulator** (p. 36), **KeySwitchMatrixManipulator** (p. 61), **NodeTrackerManipulator** (p. 72), **SphericalManipulator** (p. 80), **TerrainManipulator** (p. 90), **TrackballManipulator** (p. 97), and **UFOManipulator** (p. 104).

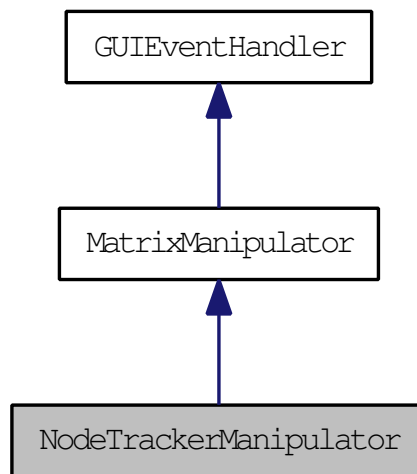
**4.13.4 Member Data Documentation****4.13.4.1 bool \_autoComputeHomePosition [protected]****4.13.4.2 osg::ref\_ptr<CoordinateFrameCallback> \_coordinateFrameCallback [protected]****4.13.4.3 osg::Vec3d \_homeCenter [protected]****4.13.4.4 osg::Vec3d \_homeEye [protected]****4.13.4.5 osg::Vec3d \_homeUp [protected]****4.13.4.6 unsigned int \_intersectTraversalMask [protected]****4.13.4.7 double \_minimumDistance [protected]**

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

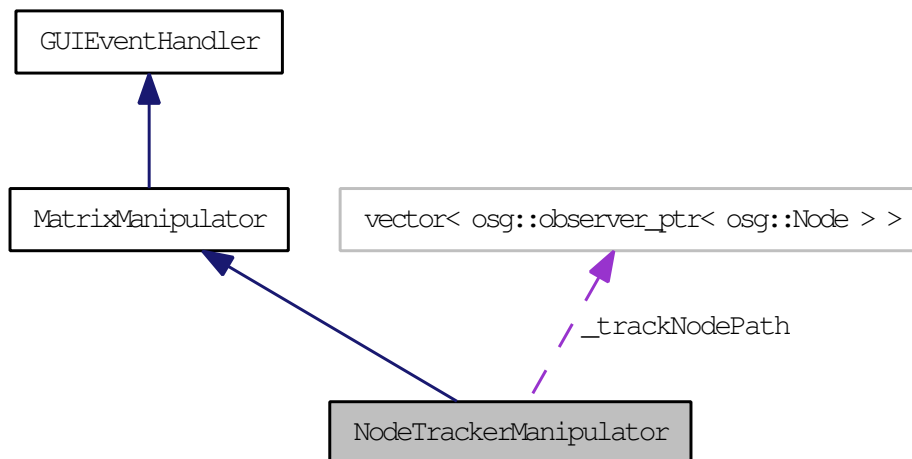
- **MatrixManipulator**
- **MatrixManipulator.cpp**

### 4.14 NodeTrackerManipulator Class Reference

Inheritance diagram for NodeTrackerManipulator:



Collaboration diagram for NodeTrackerManipulator:



#### Public Types

- typedef std::vector< osg::observer\_ptr< osg::Node > > **ObserverNodePath**
- enum **RotationMode** { TRACKBALL, ELEVATION\_AZIM }
- enum **TrackerMode** { NODE\_CENTER, NODE\_CENTER\_AND\_AZIM, NODE\_CENTER\_AND\_ROTATION }

#### Public Member Functions

- **NodeTrackerManipulator** ()
- virtual const char \* **className** () const
- virtual void **computeHomePosition** ()  
*Compute the home position.*
- virtual osgUtil::SceneView::FusionDistanceMode **getFusionDistanceMode** () const  
*Get the FusionDistanceMode.*
- virtual float **getFusionDistanceValue** () const

*Get the FusionDistanceValue.*

- virtual osg::Matrixd **getInverseMatrix** () const  
*get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.*
- virtual osg::Matrixd **getMatrix** () const  
*get the position of the manipulator as 4x4 Matrix.*
- virtual osg::Node \* **getNode** ()  
*Return node if attached.*
- virtual const osg::Node \* **getNode** () const  
*Return node if attached.*
- **RotationMode** **getRotationMode** () const
- **TrackerMode** **getTrackerMode** () const
- const osg::Node \* **getTrackNode** () const
- osg::Node \* **getTrackNode** ()
- **ObserverNodePath** & **getTrackNodePath** ()
- virtual void **getUsage** (osg::ApplicationUsage &usage) const  
*Get the keyboard and mouse usage of this manipulator.*
- virtual bool **handle** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*handle events, return true if handled, false otherwise.*
- virtual void **home** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Move the camera to the default position.*
- virtual void **init** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Start/restart the manipulator.*
- virtual void **setByInverseMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setByMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setNode** (osg::Node \*)  
*Attach a node to the manipulator.*
- void **setRotationMode** (**RotationMode** mode)
- void **setTrackerMode** (**TrackerMode** mode)
- void **setTrackNode** (osg::Node \*node)
- void **setTrackNodePath** (const **ObserverNodePath** &nodePath)
- void **setTrackNodePath** (const osg::NodePath &nodePath)

### Protected Member Functions

- virtual ~**NodeTrackerManipulator** ()
- void **addMouseEvent** (const **GUIEventAdapter** &ea)  
*Add the current mouse GUIEvent to internal stack.*
- bool **calcMovement** ()  
*For the give mouse movement calculate the movement of the camera.*
- void **clampOrientation** ()
- void **computeNodeCenterAndRotation** (osg::Vec3d &center, osg::Quat &rotation) const

- void **computeNodeLocalToWorld** (osg::Matrixd &localToWorld) const
- void **computeNodeWorldToLocal** (osg::Matrixd &worldToLocal) const
- void **computePosition** (const osg::Vec3d &eye, const osg::Vec3d &lv, const osg::Vec3d &up)
- void **flushMouseEventStack** ()  
*Reset the internal GUIEvent stack.*
- osg::NodePath **getNodePath** () const
- bool **isMouseMoving** ()  
*Check the speed at which the mouse is moving.*
- double **tb\_project\_to\_sphere** (double r, double x, double y)
- void **trackball** (osg::Vec3 &axis, double &angle, double p1x, double p1y, double p2x, double p2y)
- bool **validateNodePath** () const

## Protected Attributes

- float **\_distance**
- osg::ref\_ptr< const **GUIEventAdapter** > **\_ga\_t0**
- osg::ref\_ptr< const **GUIEventAdapter** > **\_ga\_t1**
- osg::ref\_ptr< osg::Node > **\_node**
- osg::Quat **\_nodeRotation**
- osg::Quat **\_rotation**
- **RotationMode** **\_rotationMode**
- bool **\_thrown**
- **TrackerMode** **\_trackerMode**
- **ObserverNodePath** **\_trackNodePath**

### 4.14.1 Member Typedef Documentation

4.14.1.1 typedef std::vector< osg::observer\_ptr<osg::Node> > **ObserverNodePath**

### 4.14.2 Member Enumeration Documentation

4.14.2.1 enum **RotationMode**

Enumerator:

**TRACKBALL** Use a trackball style manipulation of the view direction w.r.t the tracked orientation.

**ELEVATION\_AZIM** Allow the elevation and azimuth angles to be adjust w.r.t the tracked orientation.

4.14.2.2 enum **TrackerMode**

Enumerator:

**NODE\_CENTER** Track the center of the node's bounding sphere, but not rotations of the node. For databases which have a `CoordinateSystemNode`, the orientation is kept relative the coordinate frame if the center of the node.

**NODE\_CENTER\_AND\_AZIM** Track the center of the node's bounding sphere, and the azimuth rotation (about the z axis of the current coordinate frame). For databases which have a `CoordinateSystemNode`, the orientation is kept relative the coordinate frame if the center of the node.

**NODE\_CENTER\_AND\_ROTATION** Tack the center of the node's bounding sphere, and the all rotations of the node.

### 4.14.3 Constructor & Destructor Documentation

4.14.3.1 **NodeTrackerManipulator** ()

4.14.3.2 **~NodeTrackerManipulator** () [protected, virtual]

### 4.14.4 Member Function Documentation

4.14.4.1 void **addMouseEvent** (const **GUIEventAdapter** & ea) [protected]

Add the current mouse GUIEvent to internal stack.

**4.14.4.2 bool calcMovement () [protected]**

For the give mouse movement calculate the movement of the camera. Return true is camera has moved and a redraw is required.

**4.14.4.3 void clampOrientation () [protected]****4.14.4.4 virtual const char\* className () const [inline, virtual]**

Reimplemented from **MatrixManipulator** (p. 64).

**4.14.4.5 void computeHomePosition () [virtual]**

Compute the home position.

Reimplemented from **MatrixManipulator** (p. 64).

**4.14.4.6 void computeNodeCenterAndRotation (osg::Vec3d & center, osg::Quat & rotation) const [protected]****4.14.4.7 void computeNodeLocalToWorld (osg::Matrixd & localToWorld) const [protected]****4.14.4.8 void computeNodeWorldToLocal (osg::Matrixd & worldToLocal) const [protected]****4.14.4.9 void computePosition (const osg::Vec3d & eye, const osg::Vec3d & lv, const osg::Vec3d & up) [protected]****4.14.4.10 void flushMouseEventStack () [protected]**

Reset the internal GUIEvent stack.

**4.14.4.11 virtual osgUtil::SceneView::FusionDistanceMode getFusionDistanceMode () const [inline, virtual]**

Get the FusionDistanceMode. Used by SceneView for setting up stereo convergence.

Reimplemented from **MatrixManipulator** (p. 64).

**4.14.4.12 virtual float getFusionDistanceValue () const [inline, virtual]**

Get the FusionDistanceValue. Used by SceneView for setting up stereo convergence.

Reimplemented from **MatrixManipulator** (p. 65).

**4.14.4.13 osg::Matrixd getInverseMatrix () const [virtual]**

get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.

Implements **MatrixManipulator** (p. 65).

**4.14.4.14 osg::Matrixd getMatrix () const [virtual]**

get the position of the manipulator as 4x4 Matrix.

Implements **MatrixManipulator** (p. 65).

**4.14.4.15 osg::Node \* getNode () [virtual]**

Return node if attached.

Reimplemented from **MatrixManipulator** (p. 65).

**4.14.4.16 const osg::Node \* getNode () const [virtual]**

Return node if attached.

Reimplemented from **MatrixManipulator** (p. 65).

**4.14.4.17** `osg::NodePath getNodePath () const` [protected]

**4.14.4.18** `RotationMode getRotationMode () const` [inline]

**4.14.4.19** `TrackerMode getTrackerMode () const` [inline]

**4.14.4.20** `const osg::Node* getTrackNode () const` [inline]

**4.14.4.21** `osg::Node* getTrackNode ()` [inline]

**4.14.4.22** `ObserverNodePath& getTrackNodePath ()` [inline]

**4.14.4.23** `void getUsage (osg::ApplicationUsage & usage) const` [virtual]

Get the keyboard and mouse usage of this manipulator.

Reimplemented from **GUIEventHandler** (p. 55).

**4.14.4.24** `bool handle (const GUIEventAdapter & ea, GUIActionAdapter & us)` [virtual]

handle events, return true if handled, false otherwise.

Reimplemented from **MatrixManipulator** (p. 65).

**4.14.4.25** `void home (const GUIEventAdapter & ea, GUIActionAdapter & us)` [virtual]

Move the camera to the default position. May be ignored by manipulators if home functionality is not appropriate.

Reimplemented from **MatrixManipulator** (p. 66).

**4.14.4.26** `void init (const GUIEventAdapter & ea, GUIActionAdapter & us)` [virtual]

Start/restart the manipulator.

Reimplemented from **MatrixManipulator** (p. 66).

**4.14.4.27** `bool isMouseMoving ()` [protected]

Check the speed at which the mouse is moving. If speed is below a threshold then return false, otherwise return true.

**4.14.4.28** `virtual void setByInverseMatrix (const osg::Matrixd & matrix)` [inline, virtual]

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.14.4.29** `void setByMatrix (const osg::Matrixd & matrix)` [virtual]

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.14.4.30** `void setNode (osg::Node * node)` [virtual]

Attach a node to the manipulator. Automatically detaches previously attached node. setNode(NULL) detaches previously nodes. Is ignored by manipulators which do not require a reference model.

Reimplemented from **MatrixManipulator** (p. 67).

- 4.14.4.31 void setRotationMode (RotationMode *mode*)
- 4.14.4.32 void setTrackerMode (TrackerMode *mode*)
- 4.14.4.33 void setTrackNode (osg::Node \* *node*)
- 4.14.4.34 void setTrackNodePath (const ObserverNodePath & *nodePath*) [inline]
- 4.14.4.35 void setTrackNodePath (const osg::NodePath & *nodePath*) [inline]
- 4.14.4.36 double tb\_project\_to\_sphere (double *r*, double *x*, double *y*) [protected]
- 4.14.4.37 void trackball (osg::Vec3 & *axis*, double & *angle*, double *p1x*, double *p1y*, double *p2x*, double *p2y*) [protected]
- 4.14.4.38 bool validateNodePath () const [protected]

#### 4.14.5 Member Data Documentation

- 4.14.5.1 float *\_distance* [protected]
- 4.14.5.2 osg::ref\_ptr<const GUIEventAdapter> *\_ga\_t0* [protected]
- 4.14.5.3 osg::ref\_ptr<const GUIEventAdapter> *\_ga\_t1* [protected]
- 4.14.5.4 osg::ref\_ptr<osg::Node> *\_node* [protected]
- 4.14.5.5 osg::Quat *\_nodeRotation* [protected]
- 4.14.5.6 osg::Quat *\_rotation* [protected]
- 4.14.5.7 RotationMode *\_rotationMode* [protected]
- 4.14.5.8 bool *\_thrown* [protected]
- 4.14.5.9 TrackerMode *\_trackerMode* [protected]
- 4.14.5.10 ObserverNodePath *\_trackNodePath* [protected]

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

- NodeTrackerManipulator
- NodeTrackerManipulator.cpp

## 4.15 Scrolling Struct Reference

### Public Member Functions

- **Scrolling** (const **Scrolling** &rhs)
- **Scrolling** ()

### Public Attributes

- float **deltaX**
- float **deltaY**
- **ScrollingMotion** motion

### 4.15.1 Constructor & Destructor Documentation

4.15.1.1 **Scrolling** () [inline]

4.15.1.2 **Scrolling** (const **Scrolling** & *rhs*) [inline]

### 4.15.2 Member Data Documentation

4.15.2.1 float **deltaX**

4.15.2.2 float **deltaY**

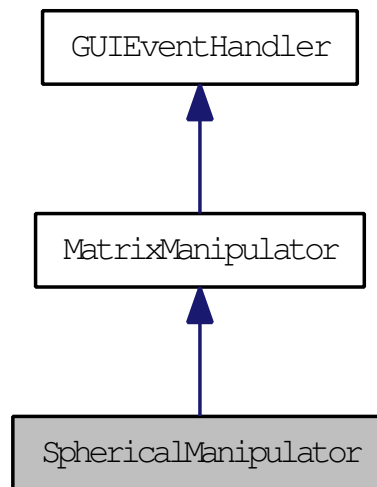
4.15.2.3 **ScrollingMotion** motion

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

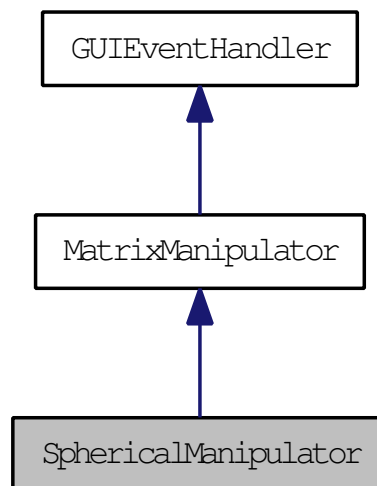
- **GUIEventAdapter**

## 4.16 SphericalManipulator Class Reference

Inheritance diagram for SphericalManipulator:



Collaboration diagram for SphericalManipulator:



### Public Types

- enum **RotationMode** { **ELEVATION\_HEADING** = 0, **HEADING**, **ELEVATION**, **MAP** }

### Public Member Functions

- **SphericalManipulator** ()
- virtual const char \* **className** () const
- virtual void **computeHomePosition** ()  
*Compute the home position.*
- void **computeViewPosition** (const osg::BoundingSphere &bound, double &scale, double &distance, osg::Vec3d &center)
- bool **getAllowThrow** () const  
*Returns true if the camera can be thrown, false otherwise.*
- const osg::Vec3d & **getCenter** () const

- double **getDistance** () const
- double **getElevation** () const
- virtual osgUtil::SceneView::FusionDistanceMode **getFusionDistanceMode** () const  
*Get the FusionDistanceMode.*
- virtual float **getFusionDistanceValue** () const  
*Get the FusionDistanceValue.*
- double **getHeading** () const
- double **getHomeDistance** () const
- virtual osg::Matrixd **getInverseMatrix** () const  
*get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.*
- virtual osg::Matrixd **getMatrix** () const  
*get the position of the manipulator as 4x4 Matrix.*
- double **getMinimumZoomScale** () const  
*get the minimum distance (as ratio) the eye point can be zoomed in*
- virtual osg::Node \* **getNode** ()  
*Return node if attached.*
- virtual const osg::Node \* **getNode** () const  
*Return node if attached.*
- **RotationMode** **getRotationMode** () const
- double **getScrollWheelZoomDelta** () const  
*get the mouse scroll wheel zoom delta.*
- virtual void **getUsage** (osg::ApplicationUsage &usage) const  
*Get the keyboard and mouse usage of this manipulator.*
- virtual bool **handle** (const **osgGA::GUIEventAdapter** &ea, **osgGA::GUIActionAdapter** &us)  
*handle events, return true if handled, false otherwise.*
- virtual void **home** (double)  
*Move the camera to the default position.*
- virtual void **home** (const **osgGA::GUIEventAdapter** &ea, **osgGA::GUIActionAdapter** &us)  
*Move the camera to the default position.*
- virtual void **init** (const **osgGA::GUIEventAdapter** &ea, **osgGA::GUIActionAdapter** &us)  
*Start/restart the manipulator.*
- void **setAllowThrow** (bool allowThrow)  
*Set the 'allow throw' flag.*
- virtual void **setByInverseMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setByMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- void **setCenter** (const osg::Vec3d &center)
- bool **setDistance** (double distance)
- void **setElevation** (double elevation)

- void **setHeading** (double azimuth)
- void **setMinimumZoomScale** (double minimumZoomScale)  
*set the minimum distance (as ratio) the eye point can be zoomed in towards the center before the center is pushed forward.*
- virtual void **setNode** (osg::Node \*)  
*Attach a node to the manipulator.*
- void **setRotationMode** (RotationMode mode)
- void **setScrollWheelZoomDelta** (double zoomDelta)  
*set the mouse scroll wheel zoom delta.*
- void **zoomOn** (const osg::BoundingSphere &bound)

### Protected Member Functions

- virtual ~**SphericalManipulator** ()
- void **addMouseEvent** (const osgGA::GUIEventAdapter &ea)  
*Add the current mouse GUIEvent to internal stack.*
- bool **calcMovement** ()  
*For the give mouse movement calculate the movement of the camera.*
- void **flushMouseEventStack** ()  
*Reset the internal GUIEvent stack.*
- bool **isMouseMoveing** ()  
*Check the speed at which the mouse is moving.*

### Protected Attributes

- bool **\_allowThrow**
- osg::Vec3d **\_center**
- double **\_delta\_frame\_time**  
*The approximate amount of time it is currently taking to draw a frame.*
- double **\_distance**
- double **\_elevation**
- osg::ref\_ptr< const osgGA::GUIEventAdapter > **\_ga\_t0**
- osg::ref\_ptr< const osgGA::GUIEventAdapter > **\_ga\_t1**
- double **\_heading**
- double **\_homeDistance**
- double **\_last\_frame\_time**  
*The time the last frame started.*
- double **\_minimumZoomScale**
- double **\_modelScale**
- osg::ref\_ptr< osg::Node > **\_node**
- **RotationMode \_rotationMode**
- bool **\_thrown**
- double **\_zoomDelta**

## 4.16.1 Member Enumeration Documentation

### 4.16.1.1 enum RotationMode

Enumerator:

***ELEVATION\_HEADING***

***HEADING***

***ELEVATION***

***MAP***

## 4.16.2 Constructor & Destructor Documentation

### 4.16.2.1 SphericalManipulator ()

### 4.16.2.2 ~SphericalManipulator () [protected, virtual]

## 4.16.3 Member Function Documentation

### 4.16.3.1 void addMouseEvent (const osgGA::GUIEventAdapter & ea) [protected]

Add the current mouse GUIEvent to internal stack.

### 4.16.3.2 bool calcMovement () [protected]

For the give mouse movement calculate the movement of the camera. Return true is camera has moved and a redraw is required.

### 4.16.3.3 virtual const char\* className () const [inline, virtual]

Reimplemented from **MatrixManipulator** (p. 64).

### 4.16.3.4 void computeHomePosition () [virtual]

Compute the home position.

Reimplemented from **MatrixManipulator** (p. 64).

### 4.16.3.5 void computeViewPosition (const osg::BoundingSphere & bound, double & scale, double & distance, osg::Vec3d & center)

### 4.16.3.6 void flushMouseEventStack () [protected]

Reset the internal GUIEvent stack.

### 4.16.3.7 bool getAllowThrow () const [inline]

Returns true if the camera can be thrown, false otherwise. This defaults to true.

### 4.16.3.8 const osg::Vec3d& getCenter () const [inline]

### 4.16.3.9 double getDistance () const [inline]

### 4.16.3.10 double getElevation () const [inline]

### 4.16.3.11 virtual osgUtil::SceneView::FusionDistanceMode getFusionDistanceMode () const [inline, virtual]

Get the FusionDistanceMode. Used by SceneView for setting up stereo convergence.

Reimplemented from **MatrixManipulator** (p. 64).

### 4.16.3.12 virtual float getFusionDistanceValue () const [inline, virtual]

Get the FusionDistanceValue. Used by SceneView for setting up stereo convergence.

Reimplemented from **MatrixManipulator** (p. 65).

**4.16.3.13** `double getHeading () const [inline]`

**4.16.3.14** `double getHomeDistance () const [inline]`

**4.16.3.15** `osg::Matrixd getInverseMatrix () const [virtual]`

get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.  
Implements **MatrixManipulator** (p. 65).

**4.16.3.16** `osg::Matrixd getMatrix () const [virtual]`

get the position of the manipulator as 4x4 Matrix.  
Implements **MatrixManipulator** (p. 65).

**4.16.3.17** `double getMinimumZoomScale () const [inline]`

get the minimum distance (as ratio) the eye point can be zoomed in

**4.16.3.18** `osg::Node * getNode () [virtual]`

Return node if attached.  
Reimplemented from **MatrixManipulator** (p. 65).

**4.16.3.19** `const osg::Node * getNode () const [virtual]`

Return node if attached.  
Reimplemented from **MatrixManipulator** (p. 65).

**4.16.3.20** `RotationMode getRotationMode () const [inline]`

**4.16.3.21** `double getScrollWheelZoomDelta () const [inline]`

get the mouse scroll wheel zoom delta.

**4.16.3.22** `void getUsage (osg::ApplicationUsage & usage) const [virtual]`

Get the keyboard and mouse usage of this manipulator.  
Reimplemented from **GUIEventHandler** (p. 55).

**4.16.3.23** `bool handle (const osgGA::GUIEventAdapter & ea, osgGA::GUIActionAdapter & us) [virtual]`

handle events, return true if handled, false otherwise.  
Reimplemented from **MatrixManipulator** (p. 65).

**4.16.3.24** `void home (double) [virtual]`

Move the camera to the default position. This version does not require **GUIEventAdapter** (p. 40) and **GUIActionAdapter** (p. 38) so may be called from somewhere other than a **handle()** (p. 79) method in **GUIEventHandler** (p. 54). Application must be aware of implications.  
Reimplemented from **MatrixManipulator** (p. 66).

**4.16.3.25** `void home (const osgGA::GUIEventAdapter & ea, osgGA::GUIActionAdapter & us) [virtual]`

Move the camera to the default position. May be ignored by manipulators if home functionality is not appropriate.  
Reimplemented from **MatrixManipulator** (p. 66).

**4.16.3.26** `void init (const osgGA::GUIEventAdapter & ea, osgGA::GUIActionAdapter & us) [virtual]`

Start/restart the manipulator.  
Reimplemented from **MatrixManipulator** (p. 66).

**4.16.3.27** `bool isMouseMoving () [protected]`

Check the speed at which the mouse is moving. If speed is below a threshold then return false, otherwise return true.

**4.16.3.28 void setAllowThrow (bool *allowThrow*) [inline]**

Set the 'allow throw' flag. Releasing the mouse button while moving the camera results in a throw.

**4.16.3.29 virtual void setByInverseMatrix (const osg::Matrixd & *matrix*) [inline, virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.16.3.30 void setByMatrix (const osg::Matrixd & *matrix*) [virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.16.3.31 void setCenter (const osg::Vec3d & *center*) [inline]****4.16.3.32 bool setDistance (double *distance*)****4.16.3.33 void setElevation (double *elevation*) [inline]****4.16.3.34 void setHeading (double *azimuth*) [inline]****4.16.3.35 void setMinimumZoomScale (double *minimumZoomScale*) [inline]**

set the minimum distance (as ratio) the eye point can be zoomed in towards the center before the center is pushed forward.

**4.16.3.36 void setNode (osg::Node \* *node*) [virtual]**

Attach a node to the manipulator. Automatically detaches previously attached node. setNode(NULL) detaches previously nodes. Is ignored by manipulators which do not require a reference model.

Reimplemented from **MatrixManipulator** (p. 67).

**4.16.3.37 void setRotationMode (RotationMode *mode*)****4.16.3.38 void setScroolWheelZoomDelta (double *zoomDelta*) [inline]**

set the mouse scroll wheel zoom delta. Range -1.0 to +1.0, -ve value inverts wheel direction and zero switches off scroll wheel.

**4.16.3.39 void zoomOn (const osg::BoundingSphere & *bound*)****4.16.4 Member Data Documentation****4.16.4.1 bool *\_allowThrow* [protected]****4.16.4.2 osg::Vec3d *\_center* [protected]****4.16.4.3 double *\_delta\_frame\_time* [protected]**

The approximate amount of time it is currently taking to draw a frame. This is used to compute the delta in translation/rotation during a thrown display update. It allows us to match an delta in position/rotation independent of the rendering frame rate.

**4.16.4.4 double *\_distance* [protected]****4.16.4.5 double *\_elevation* [protected]****4.16.4.6 osg::ref\_ptr<const osgGA::GUIEventAdapter> *\_ga\_t0* [protected]****4.16.4.7 osg::ref\_ptr<const osgGA::GUIEventAdapter> *\_ga\_t1* [protected]****4.16.4.8 double *\_heading* [protected]****4.16.4.9 double *\_homeDistance* [protected]****4.16.4.10 double *\_last\_frame\_time* [protected]**

The time the last frame started. Used when *\_rate\_sensitive* is true so that we can match display update rate to rotation/translation rate.

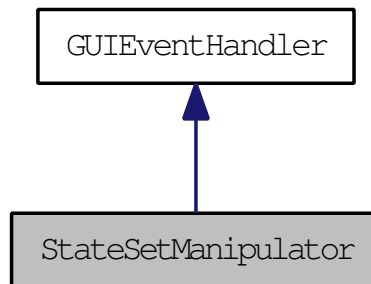
- 4.16.4.11 `double _minimumZoomScale` [protected]
- 4.16.4.12 `double _modelScale` [protected]
- 4.16.4.13 `osg::ref_ptr<osg::Node> _node` [protected]
- 4.16.4.14 `RotationMode _rotationMode` [protected]
- 4.16.4.15 `bool _thrown` [protected]
- 4.16.4.16 `double _zoomDelta` [protected]

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

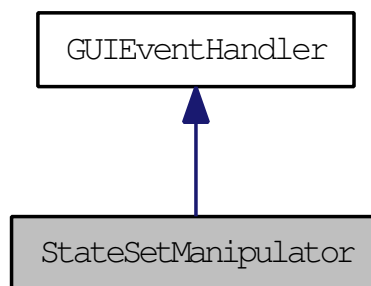
- **SphericalManipulator**
- **SphericalManipulator.cpp**

## 4.17 StateSetManipulator Class Reference

Experimental class, not been looked at for a while, but which will be returned to at some point :-\ . Inheritance diagram for StateSetManipulator:



Collaboration diagram for StateSetManipulator:



### Public Member Functions

- **StateSetManipulator** (osg::StateSet \*stateset=0)
- virtual const char \* **className** () const
- void **cyclePolygonMode** ()
- bool **getBackfaceEnabled** () const
- int **getKeyEventCyclePolygonMode** () const
- int **getKeyEventToggleBackfaceCulling** () const
- int **getKeyEventToggleLighting** () const
- int **getKeyEventToggleTexturing** () const
- bool **getLightingEnabled** () const
- unsigned int **getMaximumNumOfTextureUnits** () const
- osg::PolygonMode::Mode **getPolygonMode** () const
- virtual const osg::StateSet \* **getStateSet** () const  
*get the attached a StateSet.*
- virtual osg::StateSet \* **getStateSet** ()  
*get the attached a StateSet.*
- bool **getTextureEnabled** () const
- virtual void **getUsage** (osg::ApplicationUsage &usage) const  
*Get the keyboard and mouse usage of this manipulator.*
- virtual bool **handle** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Handle events, return true if handled, false otherwise.*
- void **setBackfaceEnabled** (bool newbackface)

- void **setKeyEventCyclePolygonMode** (int key)
- void **setKeyEventToggleBackfaceCulling** (int key)
- void **setKeyEventToggleLighting** (int key)
- void **setKeyEventToggleTexturing** (int key)
- void **setLightingEnabled** (bool newlighting)
- void **setMaximumNumOfTextureUnits** (unsigned int i)
- void **setPolygonMode** (osg::PolygonMode::Mode newpolygonmode)
- virtual void **setStateSet** (osg::StateSet \*)

*attach a StateSet to the manipulator to be used for specifying view.*

- void **setTextureEnabled** (bool newtexture)

### Protected Member Functions

- virtual **~StateSetManipulator** ()
- void **clone** ()
- osg::PolygonMode \* **getOrCreatePolygonMode** ()

### Protected Attributes

- bool **\_backface**
- bool **\_initialized**
- int **\_keyEventCyclePolygonMode**
- int **\_keyEventToggleBackfaceCulling**
- int **\_keyEventToggleLighting**
- int **\_keyEventToggleTexturing**
- bool **\_lighting**
- unsigned int **\_maxNumOfTextureUnits**
- osg::ref\_ptr< osg::StateSet > **\_stateset**
- bool **\_texture**

#### 4.17.1 Detailed Description

Experimental class, not been looked at for a while, but which will be returned to at some point :-\.

## 4.17.2 Constructor & Destructor Documentation

4.17.2.1 **StateSetManipulator** (osg::StateSet \* *stateset* = 0)

4.17.2.2 **~StateSetManipulator** () [protected, virtual]

## 4.17.3 Member Function Documentation

4.17.3.1 **virtual const char\*** className () const [inline, virtual]

4.17.3.2 **void** clone () [protected]

4.17.3.3 **void** cyclePolygonMode ()

4.17.3.4 **bool** getBackfaceEnabled () const [inline]

4.17.3.5 **int** getKeyEventCyclePolygonMode () const [inline]

4.17.3.6 **int** getKeyEventToggleBackfaceCulling () const [inline]

4.17.3.7 **int** getKeyEventToggleLighting () const [inline]

4.17.3.8 **int** getKeyEventToggleTexturing () const [inline]

4.17.3.9 **bool** getLightingEnabled () const [inline]

4.17.3.10 **unsigned int** getMaximumNumOfTextureUnits () const [inline]

4.17.3.11 **osg::PolygonMode \*** getOrCreatePolygonMode () [protected]

4.17.3.12 **osg::PolygonMode::Mode** getPolygonMode () const

4.17.3.13 **const StateSet \*** getStateSet () const [virtual]

get the attached a StateSet.

4.17.3.14 **StateSet \*** getStateSet () [virtual]

get the attached a StateSet.

4.17.3.15 **bool** getTextureEnabled () const [inline]

4.17.3.16 **void** getUsage (osg::ApplicationUsage & *usage*) const [virtual]

Get the keyboard and mouse usage of this manipulator.

Reimplemented from **GUIEventHandler** (p. 55).

4.17.3.17 **bool** handle (const GUIEventAdapter & *ea*, GUIActionAdapter & *us*) [virtual]

Handle events, return true if handled, false otherwise.

Reimplemented from **GUIEventHandler** (p. 55).

4.17.3.18 **void** setBackfaceEnabled (bool *newbackface*)

4.17.3.19 **void** setKeyEventCyclePolygonMode (int *key*) [inline]

4.17.3.20 **void** setKeyEventToggleBackfaceCulling (int *key*) [inline]

4.17.3.21 **void** setKeyEventToggleLighting (int *key*) [inline]

4.17.3.22 **void** setKeyEventToggleTexturing (int *key*) [inline]

4.17.3.23 **void** setLightingEnabled (bool *newlighting*)

4.17.3.24 **void** setMaximumNumOfTextureUnits (unsigned int *i*) [inline]

4.17.3.25 **void** setPolygonMode (osg::PolygonMode::Mode *newpolygonmode*)

4.17.3.26 **virtual void** setStateSet (osg::StateSet \*) [virtual]

attach a StateSet to the manipulator to be used for specifying view.

4.17.3.27 void setTextureEnabled (bool *newtexture*)

#### 4.17.4 Member Data Documentation

4.17.4.1 bool \_backface [protected]

4.17.4.2 bool \_initialized [protected]

4.17.4.3 int \_keyEventCyclePolygonMode [protected]

4.17.4.4 int \_keyEventToggleBackfaceCulling [protected]

4.17.4.5 int \_keyEventToggleLighting [protected]

4.17.4.6 int \_keyEventToggleTexturing [protected]

4.17.4.7 bool \_lighting [protected]

4.17.4.8 unsigned int \_maxNumOfTextureUnits [protected]

4.17.4.9 osg::ref\_ptr<osg::StateSet> \_stateset [protected]

4.17.4.10 bool \_texture [protected]

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

- **StateSetManipulator**
- **StateSetManipulator.cpp**

## 4.18 TabletPen Struct Reference

### Public Member Functions

- **TabletPen** (const **TabletPen** &rhs)
- **TabletPen** ()

### Public Attributes

- float **pressure**
- float **rotation**
- **TabletPointerType** **tabletPointerType**
- float **tiltX**
- float **tiltY**

### 4.18.1 Constructor & Destructor Documentation

4.18.1.1 **TabletPen** () [*inline*]

4.18.1.2 **TabletPen** (const **TabletPen** & *rhs*) [*inline*]

### 4.18.2 Member Data Documentation

4.18.2.1 float **pressure**

4.18.2.2 float **rotation**

4.18.2.3 **TabletPointerType** **tabletPointerType**

4.18.2.4 float **tiltX**

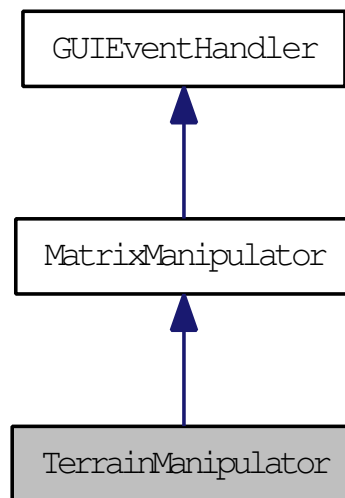
4.18.2.5 float **tiltY**

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

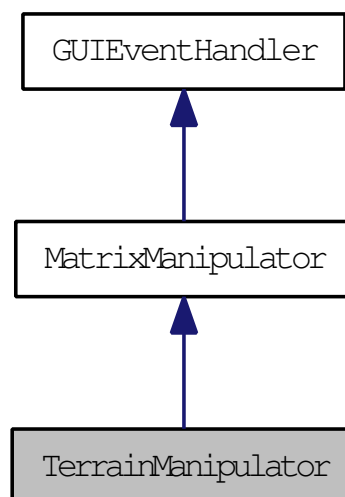
- **GUIEventAdapter**

## 4.19 TerrainManipulator Class Reference

Inheritance diagram for TerrainManipulator:



Collaboration diagram for TerrainManipulator:



### Public Types

- enum **RotationMode** { **ELEVATION\_HEADING\_ROLL**, **ELEVATION\_HEADING** }

### Public Member Functions

- **TerrainManipulator** ()
- virtual const char \* **className** () const
- virtual osgUtil::SceneView::FusionDistanceMode **getFusionDistanceMode** () const  
*Get the FusionDistanceMode.*
- virtual float **getFusionDistanceValue** () const  
*Get the FusionDistanceValue.*
- virtual osg::Matrixd **getInverseMatrix** () const  
*get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.*

- virtual osg::Matrixd **getMatrix** () const  
*get the position of the manipulator as 4x4 Matrix.*
- virtual osg::Node \* **getNode** ()  
*Return node if attached.*
- virtual const osg::Node \* **getNode** () const  
*Return node if attached.*
- **RotationMode getRotationMode** () const
- virtual void **getUsage** (osg::ApplicationUsage &usage) const  
*Get the keyboard and mouse usage of this manipulator.*
- virtual bool **handle** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*handle events, return true if handled, false otherwise.*
- virtual void **home** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Move the camera to the default position.*
- virtual void **init** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Start/restart the manipulator.*
- virtual void **setByInverseMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setByMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setNode** (osg::Node \*)  
*Attach a node to the manipulator.*
- void **setRotationMode** (**RotationMode** mode)

### Protected Member Functions

- virtual ~**TerrainManipulator** ()
- void **addMouseEvent** (const **GUIEventAdapter** &ea)  
*Add the current mouse GUIEvent to internal stack.*
- bool **calcMovement** ()  
*For the give mouse movement calculate the movement of the camera.*
- void **clampOrientation** ()
- void **computePosition** (const osg::Vec3d &eye, const osg::Vec3d &lv, const osg::Vec3d &up)
- void **flushMouseEventStack** ()  
*Reset the internal GUIEvent stack.*
- bool **intersect** (const osg::Vec3d &start, const osg::Vec3d &end, osg::Vec3d &intersection) const
- bool **isMouseMoveing** ()  
*Check the speed at which the mouse is moving.*
- double **tb\_project\_to\_sphere** (double r, double x, double y)
- void **trackball** (osg::Vec3 &axis, double &angle, double p1x, double p1y, double p2x, double p2y)

## Protected Attributes

- osg::Vec3d **\_center**
- double **\_distance**
- osg::ref\_ptr< const **GUIEventAdapter** > **\_ga\_t0**
- osg::ref\_ptr< const **GUIEventAdapter** > **\_ga\_t1**
- osg::ref\_ptr< osg::Node > **\_node**
- osg::Vec3d **\_previousUp**
- osg::Quat **\_rotation**
- **RotationMode** **\_rotationMode**
- bool **\_thrown**

### 4.19.1 Member Enumeration Documentation

#### 4.19.1.1 enum RotationMode

Enumerator:

***ELEVATION\_HEADING\_ROLL***  
***ELEVATION\_HEADING***

### 4.19.2 Constructor & Destructor Documentation

#### 4.19.2.1 TerrainManipulator ()

#### 4.19.2.2 ~TerrainManipulator () [protected, virtual]

### 4.19.3 Member Function Documentation

#### 4.19.3.1 void addMouseEvent (const GUIEventAdapter & ea) [protected]

Add the current mouse GUIEvent to internal stack.

#### 4.19.3.2 bool calcMovement () [protected]

For the give mouse movement calculate the movement of the camera. Return true is camera has moved and a redraw is required.

#### 4.19.3.3 void clampOrientation () [protected]

#### 4.19.3.4 virtual const char\* className () const [inline, virtual]

Reimplemented from **MatrixManipulator** (p. 64).

#### 4.19.3.5 void computePosition (const osg::Vec3d & eye, const osg::Vec3d & lv, const osg::Vec3d & up) [protected]

#### 4.19.3.6 void flushMouseEventStack () [protected]

Reset the internal GUIEvent stack.

#### 4.19.3.7 virtual osgUtil::SceneView::FusionDistanceMode getFusionDistanceMode () const [inline, virtual]

Get the FusionDistanceMode. Used by SceneView for setting up stereo convergence.

Reimplemented from **MatrixManipulator** (p. 64).

#### 4.19.3.8 virtual float getFusionDistanceValue () const [inline, virtual]

Get the FusionDistanceValue. Used by SceneView for setting up stereo convergence.

Reimplemented from **MatrixManipulator** (p. 65).

#### 4.19.3.9 osg::Matrixd getInverseMatrix () const [virtual]

get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.

Implements **MatrixManipulator** (p. 65).

**4.19.3.10 osg::Matrixd getMatrix () const [virtual]**

get the position of the manipulator as 4x4 Matrix.

Implements **MatrixManipulator** (p. 65).

**4.19.3.11 osg::Node \* getNode () [virtual]**

Return node if attached.

Reimplemented from **MatrixManipulator** (p. 65).

**4.19.3.12 const osg::Node \* getNode () const [virtual]**

Return node if attached.

Reimplemented from **MatrixManipulator** (p. 65).

**4.19.3.13 RotationMode getRotationMode () const [inline]****4.19.3.14 void getUsage (osg::ApplicationUsage & usage) const [virtual]**

Get the keyboard and mouse usage of this manipulator.

Reimplemented from **GUIEventHandler** (p. 55).

**4.19.3.15 bool handle (const GUIEventAdapter & ea, GUIActionAdapter & us) [virtual]**

handle events, return true if handled, false otherwise.

Reimplemented from **MatrixManipulator** (p. 65).

**4.19.3.16 void home (const GUIEventAdapter & ea, GUIActionAdapter & us) [virtual]**

Move the camera to the default position. May be ignored by manipulators if home functionality is not appropriate.

Reimplemented from **MatrixManipulator** (p. 66).

**4.19.3.17 void init (const GUIEventAdapter & ea, GUIActionAdapter & us) [virtual]**

Start/restart the manipulator.

Reimplemented from **MatrixManipulator** (p. 66).

**4.19.3.18 bool intersect (const osg::Vec3d & start, const osg::Vec3d & end, osg::Vec3d & intersection) const [protected]****4.19.3.19 bool isMouseMoving () [protected]**

Check the speed at which the mouse is moving. If speed is below a threshold then return false, otherwise return true.

**4.19.3.20 virtual void setByInverseMatrix (const osg::Matrixd & matrix) [inline, virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.19.3.21 void setByMatrix (const osg::Matrixd & matrix) [virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.19.3.22 void setNode (osg::Node \* node) [virtual]**

Attach a node to the manipulator. Automatically detaches previously attached node. setNode(NULL) detaches previously nodes. Is ignored by manipulators which do not require a reference model.

Reimplemented from **MatrixManipulator** (p. 67).

4.19.3.23 void setRotationMode (RotationMode *mode*)

4.19.3.24 double tb\_project\_to\_sphere (double *r*, double *x*, double *y*) [protected]

4.19.3.25 void trackball (osg::Vec3 & *axis*, double & *angle*, double *p1x*, double *p1y*, double *p2x*, double *p2y*) [protected]

#### 4.19.4 Member Data Documentation

4.19.4.1 osg::Vec3d \_center [protected]

4.19.4.2 double \_distance [protected]

4.19.4.3 osg::ref\_ptr<const GUIEventAdapter> \_ga\_t0 [protected]

4.19.4.4 osg::ref\_ptr<const GUIEventAdapter> \_ga\_t1 [protected]

4.19.4.5 osg::ref\_ptr<osg::Node> \_node [protected]

4.19.4.6 osg::Vec3d \_previousUp [protected]

4.19.4.7 osg::Quat \_rotation [protected]

4.19.4.8 RotationMode \_rotationMode [protected]

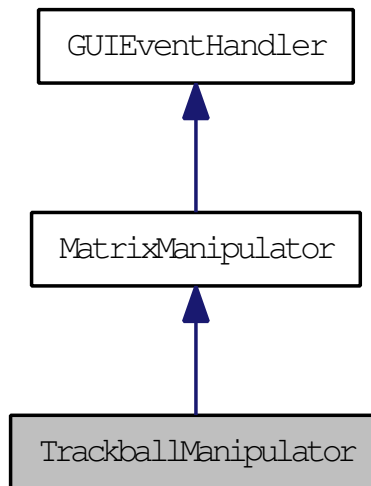
4.19.4.9 bool \_thrown [protected]

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

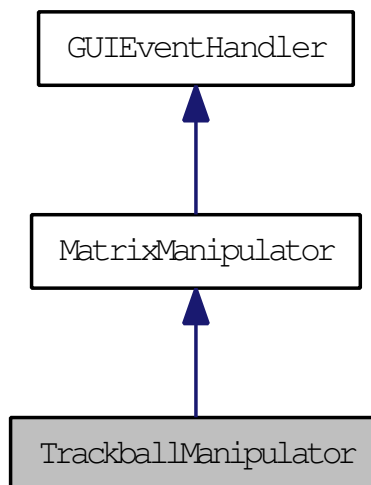
- TerrainManipulator
- TerrainManipulator.cpp

## 4.20 TrackballManipulator Class Reference

Inheritance diagram for TrackballManipulator:



Collaboration diagram for TrackballManipulator:



### Public Member Functions

- **TrackballManipulator** ()
- virtual const char \* **className** () const
- bool **getAllowThrow** () const  
*Returns true if the camera can be thrown, false otherwise.*
- const osg::Vec3d & **getCenter** () const  
*Get the center of the trackball.*
- double **getDistance** () const  
*Get the distance of the trackball.*
- virtual osgUtil::SceneView::FusionDistanceMode **getFusionDistanceMode** () const  
*Get the FusionDistanceMode.*

- virtual float **getFusionDistanceValue** () const  
*Get the FusionDistanceValue.*
- virtual osg::Matrixd **getInverseMatrix** () const  
*get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.*
- virtual osg::Matrixd **getMatrix** () const  
*get the position of the manipulator as 4x4 Matrix.*
- double **getMinimumZoomScale** () const  
*get the minimum distance (as ratio) the eye point can be zoomed in*
- virtual osg::Node \* **getNode** ()  
*Return node if attached.*
- virtual const osg::Node \* **getNode** () const  
*Return node if attached.*
- const osg::Quat & **getRotation** () const  
*Get the rotation of the trackball.*
- double **getScrollWheelZoomDelta** () const  
*get the mouse scroll wheel zoom delta.*
- float **getTrackballSize** () const  
*Get the size of the trackball.*
- virtual void **getUsage** (osg::ApplicationUsage &usage) const  
*Get the keyboard and mouse usage of this manipulator.*
- virtual bool **handle** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*handle events, return true if handled, false otherwise.*
- virtual void **home** (double)  
*Move the camera to the default position.*
- virtual void **home** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Move the camera to the default position.*
- virtual void **init** (const **GUIEventAdapter** &ea, **GUIActionAdapter** &us)  
*Start/restart the manipulator.*
- void **setAllowThrow** (bool allowThrow)  
*Set the 'allow throw' flag.*
- virtual void **setByInverseMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- virtual void **setByMatrix** (const osg::Matrixd &matrix)  
*set the position of the matrix manipulator using a 4x4 Matrix.*
- void **setCenter** (const osg::Vec3d &center)  
*Set the center of the trackball.*
- void **setDistance** (double distance)  
*Set the distance of the trackball.*

- void **setMinimumZoomScale** (double minimumZoomScale)  
*set the minimum distance (as ratio) the eye point can be zoomed in towards the center before the center is pushed forward.*
- virtual void **setNode** (osg::Node \*)  
*Attach a node to the manipulator.*
- void **setRotation** (const osg::Quat &rotation)  
*Set the rotation of the trackball.*
- void **setScrollWheelZoomDelta** (double zoomDelta)  
*set the mouse scroll wheel zoom delta.*
- void **setTrackballSize** (float size)  
*Set the size of the trackball.*

### Protected Member Functions

- virtual ~**TrackballManipulator** ()
- void **addMouseEvent** (const **GUIEventAdapter** &ea)  
*Add the current mouse GUIEvent to internal stack.*
- bool **calcMovement** ()  
*For the give mouse movement calculate the movement of the camera.*
- void **computePosition** (const osg::Vec3 &eye, const osg::Vec3 &lv, const osg::Vec3 &up)
- void **flushMouseEventStack** ()  
*Reset the internal GUIEvent stack.*
- bool **isMouseMoving** ()  
*Check the speed at which the mouse is moving.*
- float **tb\_project\_to\_sphere** (float r, float x, float y)
- void **trackball** (osg::Vec3 &axis, float &angle, float p1x, float p1y, float p2x, float p2y)

### Protected Attributes

- bool **\_allowThrow**
- osg::Vec3d **\_center**
- double **\_delta\_frame\_time**  
*The approximate amount of time it is currently taking to draw a frame.*
- double **\_distance**
- osg::ref\_ptr< const **GUIEventAdapter** > **\_ga\_t0**
- osg::ref\_ptr< const **GUIEventAdapter** > **\_ga\_t1**
- double **\_last\_frame\_time**  
*The time the last frame started.*
- double **\_minimumZoomScale**
- double **\_modelScale**
- osg::ref\_ptr< osg::Node > **\_node**
- osg::Quat **\_rotation**
- bool **\_thrown**
- float **\_trackballSize**
- float **\_zoomDelta**

## 4.20.1 Constructor & Destructor Documentation

### 4.20.1.1 TrackballManipulator ()

### 4.20.1.2 ~TrackballManipulator () [protected, virtual]

## 4.20.2 Member Function Documentation

### 4.20.2.1 void addMouseEvent (const GUIEventAdapter & ea) [protected]

Add the current mouse GUIEvent to internal stack.

### 4.20.2.2 bool calcMovement () [protected]

For the give mouse movement calculate the movement of the camera. Return true is camera has moved and a redraw is required.

### 4.20.2.3 virtual const char\* className () const [inline, virtual]

Reimplemented from **MatrixManipulator** (p. 64).

### 4.20.2.4 void computePosition (const osg::Vec3 & eye, const osg::Vec3 & /v, const osg::Vec3 & up) [protected]

### 4.20.2.5 void flushMouseEventStack () [protected]

Reset the internal GUIEvent stack.

### 4.20.2.6 bool getAllowThrow () const [inline]

Returns true if the camera can be thrown, false otherwise. This defaults to true.

### 4.20.2.7 const osg::Vec3d& getCenter () const [inline]

Get the center of the trackball.

### 4.20.2.8 double getDistance () const [inline]

Get the distance of the trackball.

### 4.20.2.9 virtual osgUtil::SceneView::FusionDistanceMode getFusionDistanceMode () const [inline, virtual]

Get the FusionDistanceMode. Used by SceneView for setting up stereo convergence.

Reimplemented from **MatrixManipulator** (p. 64).

### 4.20.2.10 virtual float getFusionDistanceValue () const [inline, virtual]

Get the FusionDistanceValue. Used by SceneView for setting up stereo convergence.

Reimplemented from **MatrixManipulator** (p. 65).

### 4.20.2.11 osg::Matrixd getInverseMatrix () const [virtual]

get the position of the manipulator as a inverse matrix of the manipulator, typically used as a model view matrix.

Implements **MatrixManipulator** (p. 65).

### 4.20.2.12 osg::Matrixd getMatrix () const [virtual]

get the position of the manipulator as 4x4 Matrix.

Implements **MatrixManipulator** (p. 65).

### 4.20.2.13 double getMinimumZoomScale () const [inline]

get the minimum distance (as ratio) the eye point can be zoomed in

### 4.20.2.14 osg::Node \* getNode () [virtual]

Return node if attached.

Reimplemented from **MatrixManipulator** (p. 65).

**4.20.2.15 const osg::Node \* getNode () const [virtual]**

Return node if attached.

Reimplemented from **MatrixManipulator** (p. 65).

**4.20.2.16 const osg::Quat& getRotation () const [inline]**

Get the rotation of the trackball.

**4.20.2.17 double getScrollWheelZoomDelta () const [inline]**

get the mouse scroll wheel zoom delta.

**4.20.2.18 float getTrackballSize () const [inline]**

Get the size of the trackball.

**4.20.2.19 void getUsage (osg::ApplicationUsage & usage) const [virtual]**

Get the keyboard and mouse usage of this manipulator.

Reimplemented from **GUIEventHandler** (p. 55).

**4.20.2.20 bool handle (const GUIEventAdapter & ea, GUIActionAdapter & us) [virtual]**

handle events, return true if handled, false otherwise.

Reimplemented from **MatrixManipulator** (p. 65).

**4.20.2.21 void home (double) [virtual]**

Move the camera to the default position. This version does not require **GUIEventAdapter** (p. 40) and **GUIActionAdapter** (p. 38) so may be called from somewhere other than a **handle()** (p. 96) method in **GUIEventHandler** (p. 54). Application must be aware of implications.

Reimplemented from **MatrixManipulator** (p. 66).

**4.20.2.22 void home (const GUIEventAdapter & ea, GUIActionAdapter & us) [virtual]**

Move the camera to the default position. May be ignored by manipulators if home functionality is not appropriate.

Reimplemented from **MatrixManipulator** (p. 66).

**4.20.2.23 void init (const GUIEventAdapter & ea, GUIActionAdapter & us) [virtual]**

Start/restart the manipulator.

Reimplemented from **MatrixManipulator** (p. 66).

**4.20.2.24 bool isMouseMoving () [protected]**

Check the speed at which the mouse is moving. If speed is below a threshold then return false, otherwise return true.

**4.20.2.25 void setAllowThrow (bool allowThrow) [inline]**

Set the 'allow throw' flag. Releasing the mouse button while moving the camera results in a throw.

**4.20.2.26 virtual void setByInverseMatrix (const osg::Matrixd & matrix) [inline, virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.20.2.27 void setByMatrix (const osg::Matrixd & matrix) [virtual]**

set the position of the matrix manipulator using a 4x4 Matrix.

Implements **MatrixManipulator** (p. 66).

**4.20.2.28 void setCenter (const osg::Vec3d & center) [inline]**

Set the center of the trackball.

**4.20.2.29 void setDistance (double *distance*) [inline]**

Set the distance of the trackball.

**4.20.2.30 void setMinimumZoomScale (double *minimumZoomScale*) [inline]**

set the minimum distance (as ratio) the eye point can be zoomed in towards the center before the center is pushed forward.

**4.20.2.31 void setNode (osg::Node \* *node*) [virtual]**

Attach a node to the manipulator. Automatically detaches previously attached node. setNode(NULL) detaches previously nodes. Is ignored by manipulators which do not require a reference model.

Reimplemented from **MatrixManipulator** (p. 67).

**4.20.2.32 void setRotation (const osg::Quat & *rotation*) [inline]**

Set the rotation of the trackball.

**4.20.2.33 void setScroolWheelZoomDelta (double *zoomDelta*) [inline]**

set the mouse scroll wheel zoom delta. Range -1.0 to +1.0, -ve value inverts wheel direction and zero switches off scroll wheel.

**4.20.2.34 void setTrackballSize (float *size*)**

Set the size of the trackball.

**4.20.2.35 float tb\_project\_to\_sphere (float *r*, float *x*, float *y*) [protected]****4.20.2.36 void trackball (osg::Vec3 & *axis*, float & *angle*, float *p1x*, float *p1y*, float *p2x*, float *p2y*) [protected]****4.20.3 Member Data Documentation****4.20.3.1 bool *\_allowThrow* [protected]****4.20.3.2 osg::Vec3d *\_center* [protected]****4.20.3.3 double *\_delta\_frame\_time* [protected]**

The approximate amount of time it is currently taking to draw a frame. This is used to compute the delta in translation/rotation during a thrown display update. It allows us to match an delta in position/rotation independent of the rendering frame rate.

**4.20.3.4 double *\_distance* [protected]****4.20.3.5 osg::ref\_ptr<const GUIEventAdapter> *\_ga\_t0* [protected]****4.20.3.6 osg::ref\_ptr<const GUIEventAdapter> *\_ga\_t1* [protected]****4.20.3.7 double *\_last\_frame\_time* [protected]**

The time the last frame started. Used when *\_rate\_sensitive* is true so that we can match display update rate to rotation/translation rate.

**4.20.3.8 double *\_minimumZoomScale* [protected]****4.20.3.9 double *\_modelScale* [protected]****4.20.3.10 osg::ref\_ptr<osg::Node> *\_node* [protected]****4.20.3.11 osg::Quat *\_rotation* [protected]****4.20.3.12 bool *\_thrown* [protected]****4.20.3.13 float *\_trackballSize* [protected]****4.20.3.14 float *\_zoomDelta* [protected]**

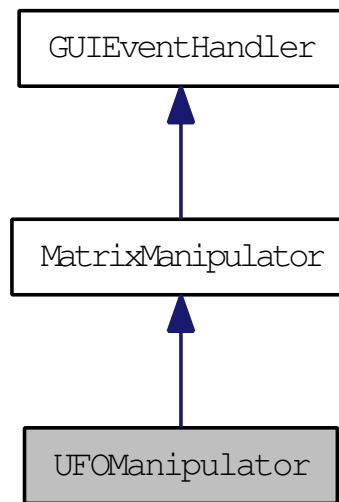
The documentation for this class was generated from the following files:

- **TrackballManipulator**

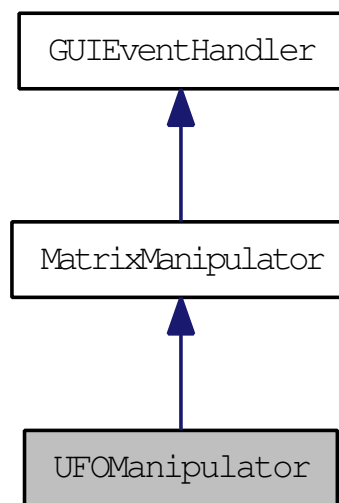
- [TrackballManipulator.cpp](#)

## 4.21 UFOManipulator Class Reference

A UFO manipulator driven with keybindings. Inheritance diagram for UFOManipulator:



Collaboration diagram for UFOManipulator:



### Public Member Functions

- **UFOManipulator** ()  
*Default constructor.*
- virtual const char \* **className** () const  
*return className*
- virtual void **computeHomePosition** ()  
*Computes the home position based on the extents and scale of the scene graph rooted at node.*
- void **getCurrentPositionAsLookAt** (osg::Vec3 &eye, osg::Vec3 &center, osg::Vec3 &up)  
*Report the current position as LookAt vectors.*
- double **getForwardSpeed** () const

- virtual osg::Matrixd **getInverseMatrix** () const  
*Get the current inverse view matrix.*
- virtual osg::Matrixd **getMatrix** () const  
*Get the current viewmatrix.*
- double **getMinDistance** () const
- double **getMinHeight** () const
- virtual osg::Node \* **getNode** ()  
*Get the root node of the subgraph this manipulator is driving the eye through.*
- virtual const osg::Node \* **getNode** () const  
*Get the root node of the subgraph this manipulator is driving the eye through (const).*
- double **getRotationSpeed** () const
- double **getSideSpeed** () const
- void **getUsage** (osg::ApplicationUsage &usage) const  
*Reports Usage parameters to the application.*
- bool **handle** (const osgGA::GUIEventAdapter &ea, osgGA::GUIActionAdapter &aa)  
*Handles incoming **osgGA** (p. 7) events.*
- void **home** (double)  
*Move the camera to the default position.*
- virtual void **home** (const osgGA::GUIEventAdapter &, osgGA::GUIActionAdapter &)  
*Sets the viewpoint matrix to the home position.*
- virtual void **init** (const GUIEventAdapter &, GUIActionAdapter &)  
*Start/restart the manipulator.*
- virtual void **setByInverseMatrix** (const osg::Matrixd &invmat)  
*Set the current position with the inverse matrix.*
- virtual void **setByMatrix** (const osg::Matrixd &matrix)  
*Set the current position with a matrix.*
- void **setForwardSpeed** (double in\_fs)
- void **setMinDistance** (double in\_min\_dist)
- void **setMinHeight** (double in\_min\_height)
- virtual void **setNode** (osg::Node \*node)  
*Set the subgraph this manipulator is driving the eye through.*
- void **setRotationSpeed** (double in\_rot\_speed)
- void **setSideSpeed** (double in\_ss)

### Protected Member Functions

- virtual ~UFOManipulator ()
- void **\_adjustPosition** ()
- void **\_frame** (const osgGA::GUIEventAdapter &ea, osgGA::GUIActionAdapter &)
- void **\_keyDown** (const osgGA::GUIEventAdapter &ea, osgGA::GUIActionAdapter &)
- void **\_keyUp** (const osgGA::GUIEventAdapter &ea, osgGA::GUIActionAdapter &)
- void **\_stop** ()
- bool **intersect** (const osg::Vec3d &start, const osg::Vec3d &end, osg::Vec3d &intersection) const

## Protected Attributes

- bool **\_ctrl**
- bool **\_decelerateOffsetRate**
- bool **\_decelerateUpSideRate**
- osg::Vec3d **\_direction**
- double **\_directionRotationAcceleration**
- double **\_directionRotationDeceleration**
- double **\_directionRotationEpsilon**
- double **\_directionRotationRate**
- double **\_dt**
- double **\_forwardSpeed**
- osg::Matrixd **\_inverseMatrix**
- osg::Matrixd **\_matrix**
- double **\_minDistanceInFront**
- double **\_minHeightAboveGround**
- osg::ref\_ptr< osg::Node > **\_node**
- osg::Matrixd **\_offset**
- double **\_pitchOffset**
- double **\_pitchOffsetRate**
- osg::Vec3d **\_position**
- bool **\_shift**
- double **\_sideSpeed**
- double **\_speedAccelerationFactor**
- double **\_speedDecelerationFactor**
- double **\_speedEpsilon**
- bool **\_straightenOffset**
- double **\_t0**
- double **\_upSpeed**
- float **\_viewAngle**
- double **\_viewOffsetDelta**
- double **\_yawOffset**
- double **\_yawOffsetRate**

### 4.21.1 Detailed Description

A UFO manipulator driven with keybindings. The **UFOManipulator** (p.99) is better suited for applications that employ architectural walk-throughs, or situations where the eyepoint motion model must move slowly, deliberately and well controlled.

The UFO Manipulator allows the following movements with the listed Key combinations: Parameters

**UpArrow** Acceleration forward.

**DownArrow** Acceleration backward (or deceleration forward).

**LeftArrow** Rotate view and direction of travel to the left.

**RightArrow** Rotate view and direction of travel to the right.

**SpaceBar** Brake. Gradually decelerates linear and rotational movement.

**Shift/UpArrow** Accelerate up.

**Shift/DownArrow** Accelerate down.

**Shift/LeftArrow** Accelerate (linearly) left.

**Shift/RightArrow** Accelerate (linearly) right.

**Shift/SpaceBar** Instant brake. Immediately stop all linear and rotational movement.

When the Shift key is released, up, down, linear left and/or linear right movement is decelerated.

Parameters

**Ctrl/UpArrow** Rotate view (but not direction of travel) up.

**Ctrl/DownArrow** Rotate view (but not direction of travel) down.

**Ctrl/LeftArrow** Rotate view (but not direction of travel) left.

**Ctrl/RightArrow** Rotate view (but not direction of travel) right.

**Ctrl/Return** Straightens out the view offset.

## 4.21.2 Constructor & Destructor Documentation

### 4.21.2.1 UFOManipulator ()

Default constructor.

### 4.21.2.2 ~UFOManipulator () [protected, virtual]

## 4.21.3 Member Function Documentation

### 4.21.3.1 void \_adjustPosition () [protected]

### 4.21.3.2 void \_frame (const osgGA::GUIEventAdapter & ea, osgGA::GUIActionAdapter &) [protected]

### 4.21.3.3 void \_keyDown (const osgGA::GUIEventAdapter & ea, osgGA::GUIActionAdapter &) [protected]

### 4.21.3.4 void \_keyUp (const osgGA::GUIEventAdapter & ea, osgGA::GUIActionAdapter &) [protected]

### 4.21.3.5 void \_stop () [protected]

### 4.21.3.6 const char \* className () const [virtual]

return className Returns returns constant "UFO"

Reimplemented from **MatrixManipulator** (p. 64).

### 4.21.3.7 void computeHomePosition () [virtual]

Computes the home position based on the extents and scale of the scene graph rooted at node.

Reimplemented from **MatrixManipulator** (p. 64).

### 4.21.3.8 void getCurrentPositionAsLookAt (osg::Vec3 & eye, osg::Vec3 & center, osg::Vec3 & up)

Report the current position as LookAt vectors.

### 4.21.3.9 double getForwardSpeed () const [inline]

### 4.21.3.10 osg::Matrixd getInverseMatrix () const [virtual]

Get the current inverse view matrix.

Implements **MatrixManipulator** (p. 65).

### 4.21.3.11 osg::Matrixd getMatrix () const [virtual]

Get the current viewmatrix.

Implements **MatrixManipulator** (p. 65).

### 4.21.3.12 double getMinDistance () const [inline]

### 4.21.3.13 double getMinHeight () const [inline]

### 4.21.3.14 osg::Node \* getNode () [virtual]

Get the root node of the subgraph this manipulator is driving the eye through.

Reimplemented from **MatrixManipulator** (p. 65).

**4.21.3.15 const osg::Node \* getNode () const [virtual]**

Get the root node of the subgraph this manipulator is driving the eye through (const).

Reimplemented from **MatrixManipulator** (p. 65).

**4.21.3.16 double getRotationSpeed () const [inline]****4.21.3.17 double getSideSpeed () const [inline]****4.21.3.18 void getUsage (osg::ApplicationUsage & usage) const [virtual]**

Reports Usage parameters to the application.

Way too busy. This needs to wait until we have a scrollable window usage.addKeyboardMouseBinding("UFO Manipulator: <SpaceBar>", "Reset the viewing angle to 0.0"); usage.addKeyboardMouseBinding("UFO Manipulator: <UpArrow>", "Acceleration forward."); usage.addKeyboardMouseBinding("UFO Manipulator: <DownArrow>", "Acceleration backward (or deceleration forward)"); usage.addKeyboardMouseBinding("UFO Manipulator: <LeftArrow>", "Rotate view and direction of travel to the left."); usage.addKeyboardMouseBinding("UFO Manipulator: <RightArrow>", "Rotate view and direction of travel to the right."); usage.addKeyboardMouseBinding("UFO Manipulator: <SpaceBar>", "Brake. Gradually decelerates linear and rotational movement."); usage.addKeyboardMouseBinding("UFO Manipulator: <Shift/UpArrow>", "Accelerate up."); usage.addKeyboardMouseBinding("UFO Manipulator: <Shift/DownArrow>", "Accelerate down."); usage.addKeyboardMouseBinding("UFO Manipulator: <Shift/LeftArrow>", "Accelerate (linearly) left."); usage.addKeyboardMouseBinding("UFO Manipulator: <Shift/RightArrow>", "Accelerate (linearly) right."); usage.addKeyboardMouseBinding("UFO Manipulator: <Shift/SpaceBar>", "Instant brake. Immediately stop all linear and rotational movement."); usage.addKeyboardMouseBinding("UFO Manipulator: <Ctrl/UpArrow>", "Rotate view (but not direction of travel) up."); usage.addKeyboardMouseBinding("UFO Manipulator: <Ctrl/DownArrow>", "Rotate view (but not direction of travel) down."); usage.addKeyboardMouseBinding("UFO Manipulator: <Ctrl/LeftArrow>", "Rotate view (but not direction of travel) left."); usage.addKeyboardMouseBinding("UFO Manipulator: <Ctrl/RightArrow>", "Rotate view (but not direction of travel) right.");

Reimplemented from **GUIEventHandler** (p. 55).

**4.21.3.19 bool handle (const osgGA::GUIEventAdapter & ea, osgGA::GUIActionAdapter & aa) [virtual]**

Handles incoming **osgGA** (p. 7) events.

Reimplemented from **MatrixManipulator** (p. 65).

**4.21.3.20 void home (double) [virtual]**

Move the camera to the default position. This version does not require **GUIEventAdapter** (p. 40) and **GUIActionAdapter** (p. 38) so may be called from somewhere other than a **handle()** (p. 103) method in **GUIEventHandler** (p. 54). Application must be aware of implications.

Reimplemented from **MatrixManipulator** (p. 66).

**4.21.3.21 void home (const osgGA::GUIEventAdapter & ea, osgGA::GUIActionAdapter & ua) [virtual]**

Sets the viewpoint matrix to the home position.

Reimplemented from **MatrixManipulator** (p. 66).

**4.21.3.22 void init (const GUIEventAdapter &, GUIActionAdapter &) [virtual]**

Start/restart the manipulator. FIXME: what does this actually mean? Provide examples.

Reimplemented from **MatrixManipulator** (p. 66).

**4.21.3.23 bool intersect (const osg::Vec3d & start, const osg::Vec3d & end, osg::Vec3d & intersection) const [protected]****4.21.3.24 void setByInverseMatrix (const osg::Matrixd & invmat) [virtual]**

Set the current position with the inverse matrix. Parameters

**invmat** The inverse of a viewpoint matrix

Implements **MatrixManipulator** (p. 66).

#### 4.21.3.25 void setByMatrix (const osg::Matrixd & *matrix*) [virtual]

Set the current position with a matrix. Parameters

***matrix*** A viewpoint matrix.

Implements **MatrixManipulator** (p. 66).

#### 4.21.3.26 void setForwardSpeed (double *in\_fs*) [inline]

#### 4.21.3.27 void setMinDistance (double *in\_min\_dist*) [inline]

#### 4.21.3.28 void setMinHeight (double *in\_min\_height*) [inline]

#### 4.21.3.29 void setNode (osg::Node \* *node*) [virtual]

Set the subgraph this manipulator is driving the eye through. Parameters

***node*** root of subgraph

Reimplemented from **MatrixManipulator** (p. 67).

4.21.3.30 void setRotationSpeed (double *in\_rot\_speed*) [inline]

4.21.3.31 void setSideSpeed (double *in\_ss*) [inline]

#### 4.21.4 Member Data Documentation

4.21.4.1 bool *\_ctrl* [protected]

4.21.4.2 bool *\_decelerateOffsetRate* [protected]

4.21.4.3 bool *\_decelerateUpSideRate* [protected]

4.21.4.4 osg::Vec3d *\_direction* [protected]

4.21.4.5 double *\_directionRotationAcceleration* [protected]

4.21.4.6 double *\_directionRotationDeceleration* [protected]

4.21.4.7 double *\_directionRotationEpsilon* [protected]

4.21.4.8 double *\_directionRotationRate* [protected]

4.21.4.9 double *\_dt* [protected]

4.21.4.10 double *\_forwardSpeed* [protected]

4.21.4.11 osg::Matrixd *\_inverseMatrix* [protected]

4.21.4.12 osg::Matrixd *\_matrix* [protected]

4.21.4.13 double *\_minDistanceInFront* [protected]

4.21.4.14 double *\_minHeightAboveGround* [protected]

4.21.4.15 osg::ref\_ptr<osg::Node> *\_node* [protected]

4.21.4.16 osg::Matrixd *\_offset* [protected]

4.21.4.17 double *\_pitchOffset* [protected]

4.21.4.18 double *\_pitchOffsetRate* [protected]

4.21.4.19 osg::Vec3d *\_position* [protected]

4.21.4.20 bool *\_shift* [protected]

4.21.4.21 double *\_sideSpeed* [protected]

4.21.4.22 double *\_speedAccelerationFactor* [protected]

4.21.4.23 double *\_speedDecelerationFactor* [protected]

4.21.4.24 double *\_speedEpsilon* [protected]

4.21.4.25 bool *\_straightenOffset* [protected]

4.21.4.26 double *\_t0* [protected]

4.21.4.27 double *\_upSpeed* [protected]

4.21.4.28 float *\_viewAngle* [protected]

4.21.4.29 double *\_viewOffsetDelta* [protected]

4.21.4.30 double *\_yawOffset* [protected]

4.21.4.31 double *\_yawOffsetRate* [protected]

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

- UFOManipulator
- UFOManipulator.cpp



# File Documentation

---

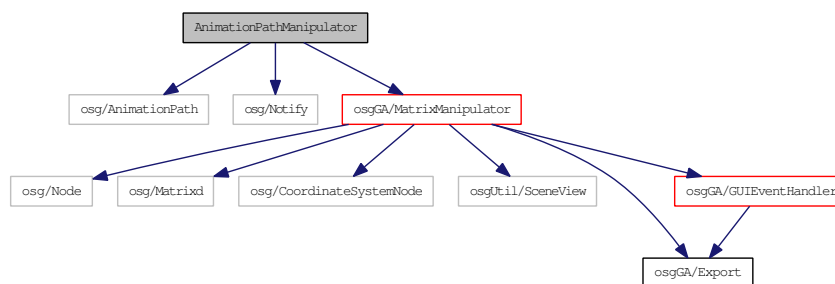
## 5.1 AnimationPathManipulator File Reference

```
#include <osg/AnimationPath>
```

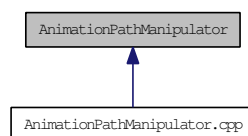
```
#include <osg/Notify>
```

```
#include <osgGA/MatrixManipulator>
```

Include dependency graph for AnimationPathManipulator:



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



### Classes

- class **AnimationPathManipulator**

### Namespaces

- namespace **osgGA**

The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.

### Defines

- #define **OSGGA\_ANIMATION\_PATH\_MANIPULATOR 1**

#### 5.1.1 Define Documentation

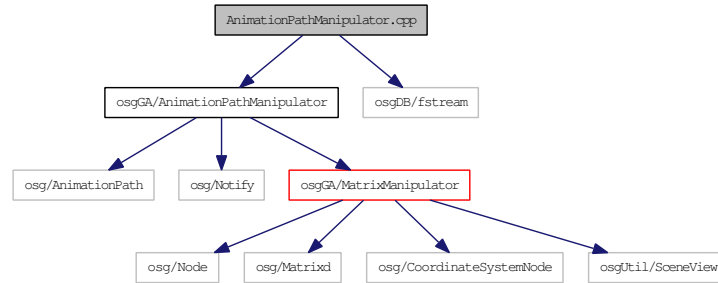
##### 5.1.1.1 #define OSGGA\_ANIMATION\_PATH\_MANIPULATOR 1

## 5.2 AnimationPathManipulator.cpp File Reference

```
#include <osgGA/AnimationPathManipulator>
```

```
#include <osgDB/fstream>
```

Include dependency graph for AnimationPathManipulator.cpp:



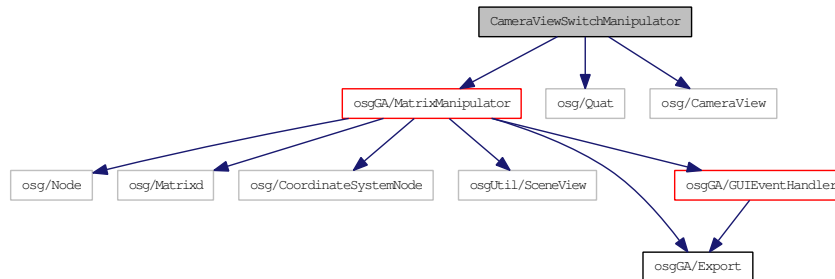
## 5.3 CameraViewSwitchManipulator File Reference

```
#include <osgGA/MatrixManipulator>
```

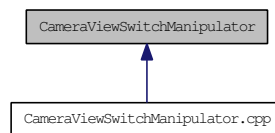
```
#include <osg/Quat>
```

```
#include <osg/CameraView>
```

Include dependency graph for CameraViewSwitchManipulator:



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



### Classes

- class **CameraViewSwitchManipulator**

### Namespaces

- namespace **osgGA**

The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the *osg* to work with varying window systems.

### Defines

- #define **OSGGA\_VIEWLISTMANIPULATOR 1**

#### 5.3.1 Define Documentation

##### 5.3.1.1 #define OSGGA\_VIEWLISTMANIPULATOR 1

## 5.4 CameraViewSwitchManipulator.cpp File Reference

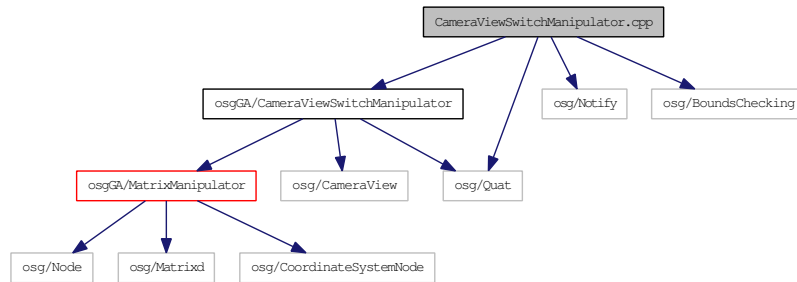
```
#include <osgGA/CameraViewSwitchManipulator>
```

```
#include <osg/Quat>
```

```
#include <osg/Notify>
```

```
#include <osg/BoundsChecking>
```

Include dependency graph for CameraViewSwitchManipulator.cpp:



### Classes

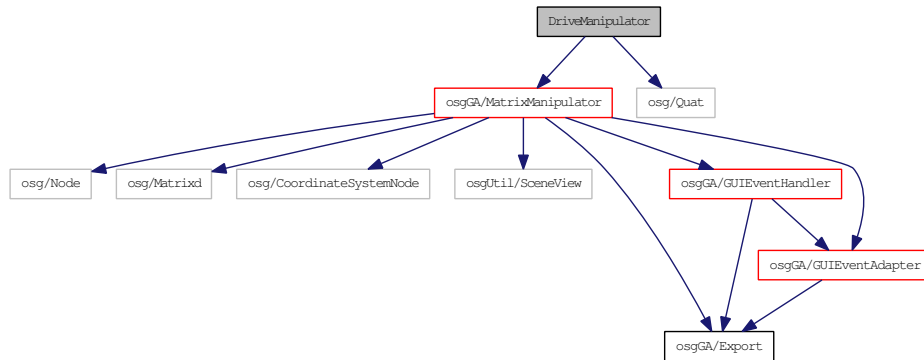
- class **CollectCameraViewsNodeVisitor**

## 5.5 DriveManipulator File Reference

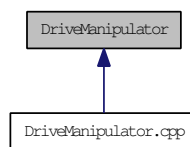
```
#include <osgGA/MatrixManipulator>
```

```
#include <osg/Quat>
```

Include dependency graph for DriveManipulator:



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



### Classes

- class **DriveManipulator**

*DriveManipulator* (p. 18) is a camera manipulator which provides drive-like functionality.

### Namespaces

- namespace **osgGA**

The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.

### Defines

- #define **OSGGA\_DRIVEMANIPULATOR 1**

#### 5.5.1 Define Documentation

##### 5.5.1.1 #define OSGGA\_DRIVEMANIPULATOR 1

## 5.6 DriveManipulator.cpp File Reference

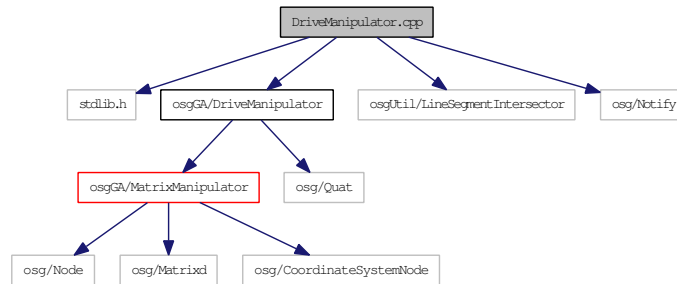
```
#include <stdlib.h>
```

```
#include <osgGA/DriveManipulator>
```

```
#include <osgUtil/LineSegmentIntersector>
```

```
#include <osg/Notify>
```

Include dependency graph for DriveManipulator.cpp:



### Defines

- #define **DRIVER\_HEIGHT** 15
- #define **KEYBOARD\_PITCH** 1

### Functions

- static double **getHeightOfDriver** ()

#### 5.6.1 Define Documentation

5.6.1.1 #define **DRIVER\_HEIGHT** 15

5.6.1.2 #define **KEYBOARD\_PITCH** 1

#### 5.6.2 Function Documentation

5.6.2.1 static double **getHeightOfDriver** () [static]

## 5.7 EventQueue File Reference

```
#include <osgGA/GUIEventAdapter>
```

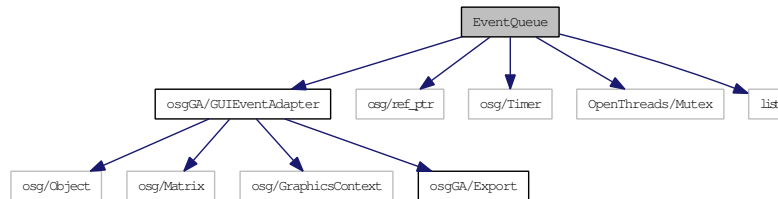
```
#include <osg/ref_ptr>
```

```
#include <osg/Timer>
```

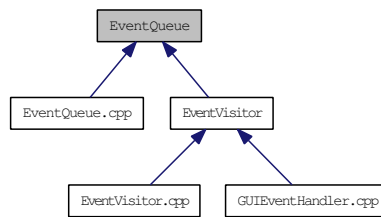
```
#include <OpenThreads/Mutex>
```

```
#include <list>
```

Include dependency graph for EventQueue:



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



### Classes

- class **EventQueue**

*EventQueue* (p. 23) implementation for collecting and adapting windowing events.

### Namespaces

- namespace **osgGA**

The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.

### Defines

- #define **OSGGA\_EVENTQUEUE** 1

#### 5.7.1 Define Documentation

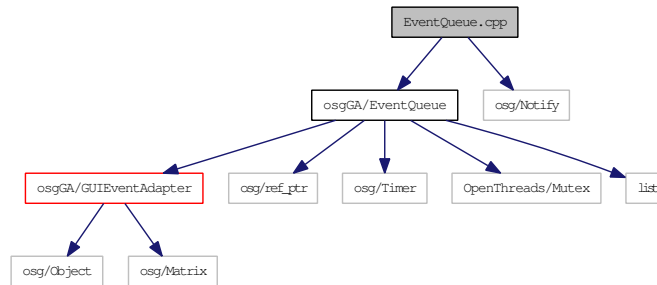
##### 5.7.1.1 #define OSGGA\_EVENTQUEUE 1

## 5.8 EventQueue.cpp File Reference

```
#include <osgGA/EventQueue>
```

```
#include <osg/Notify>
```

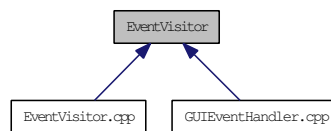
Include dependency graph for EventQueue.cpp:



## 5.9 EventVisitor File Reference

```
#include <osg/NodeVisitor>
#include <osg/Node>
#include <osg/Geode>
#include <osg/Billboard>
#include <osg/LOD>
#include <osg/Switch>
#include <osg/LightSource>
#include <osg/Transform>
#include <osg/Projection>
#include <osg/OccluderNode>
#include <osgGA/GUIEventAdapter>
#include <osgGA/GUIActionAdapter>
#include <osgGA/EventQueue>
```

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



### Classes

- class **EventVisitor**  
*Basic **EventVisitor** (p. 30) implementation for animating a scene.*

### Namespaces

- namespace **osgGA**  
*The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.*

### Defines

- #define **OSGGA\_EVENTVISITOR** 1

#### 5.9.1 Define Documentation

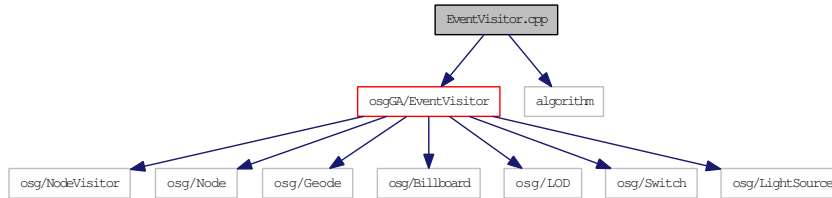
##### 5.9.1.1 #define OSGGA\_EVENTVISITOR 1

## 5.10 EventVisitor.cpp File Reference

```
#include <osgGA/EventVisitor>
```

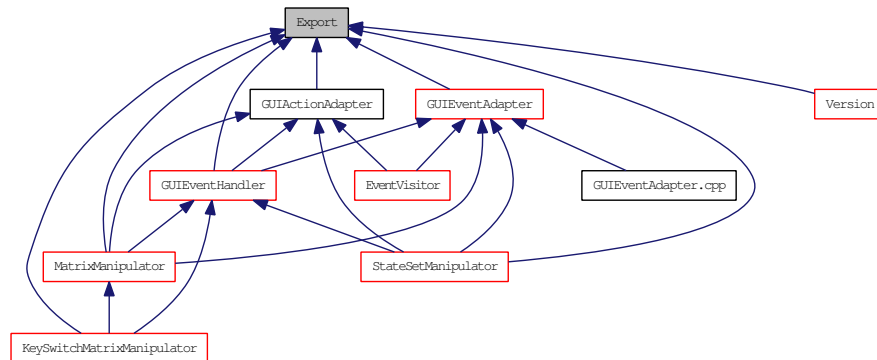
```
#include <algorithm>
```

Include dependency graph for EventVisitor.cpp:



## 5.11 Export File Reference

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



### Namespaces

- namespace **osgGA**

The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.

### Defines

- #define **OSGGA\_EXPORT**
- #define **OSGGA\_EXPORT\_1**

#### 5.11.1 Define Documentation

5.11.1.1 #define **OSGGA\_EXPORT**

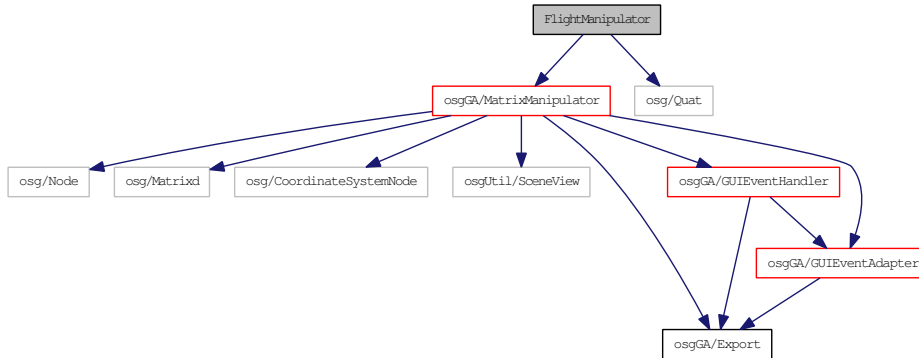
5.11.1.2 #define **OSGGA\_EXPORT\_1**

## 5.12 FlightManipulator File Reference

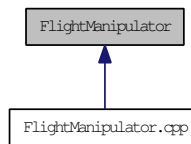
```
#include <osgGA/MatrixManipulator>
```

```
#include <osg/Quat>
```

Include dependency graph for FlightManipulator:



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



### Classes

- class **FlightManipulator**

*FlightManipulator* (p. 33) is a *MatrixManipulator* (p. 62) which provides flight simulator-like updating of the camera position & orientation.

### Namespaces

- namespace **osgGA**

The 'GA' in *osgGA* (p. 7) stands for 'GUI Abstraction'; the *osgGA* (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.

### Defines

- #define **OSGGA\_FLIGHTMANIPULATOR 1**

#### 5.12.1 Define Documentation

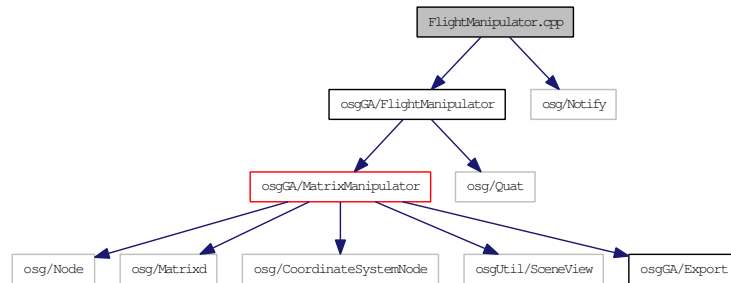
##### 5.12.1.1 #define OSGGA\_FLIGHTMANIPULATOR 1

## 5.13 FlightManipulator.cpp File Reference

```
#include <osgGA/FlightManipulator>
```

```
#include <osg/Notify>
```

Include dependency graph for FlightManipulator.cpp:

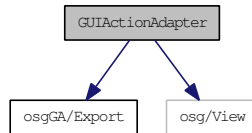


## 5.14 GUIActionAdapter File Reference

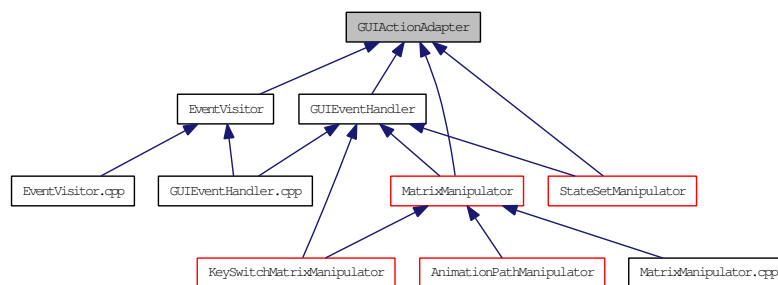
```
#include <osgGA/Export>
```

```
#include <osg/View>
```

Include dependency graph for GUIActionAdapter:



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



### Classes

- class **GUIActionAdapter**

*Abstract base class defining the interface by which GUIEventHandlers may request actions of the GUI system in use.*

### Namespaces

- namespace **osgGA**

*The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.*

### Defines

- #define **OSGGA\_GUIACTIONADAPTER** 1

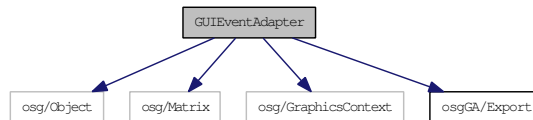
#### 5.14.1 Define Documentation

##### 5.14.1.1 #define OSGGA\_GUIACTIONADAPTER 1

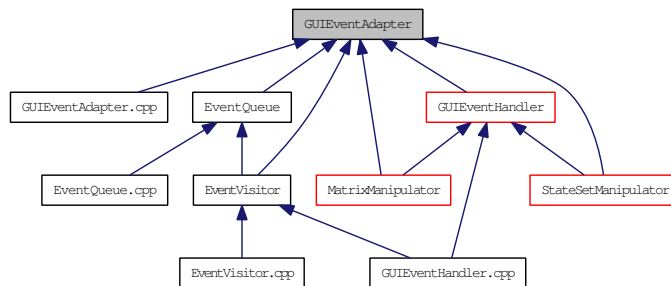
## 5.15 GUIEventAdapter File Reference

```
#include <osg/Object>
#include <osg/Matrix>
#include <osg/GraphicsContext>
#include <osgGA/Export>
```

Include dependency graph for GUIEventAdapter:



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



### Classes

- class **GUIEventAdapter**  
*Event class for storing Keyboard, mouse and window events.*
- struct **Scrolling**
- struct **TabletPen**

### Namespaces

- namespace **osgGA**  
*The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.*

### Defines

- #define **OSGGA\_EVENT 1**

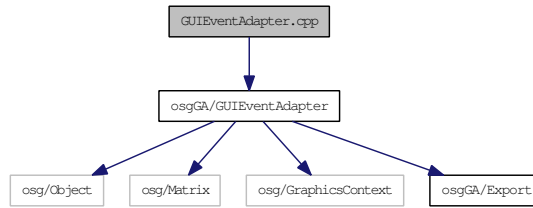
#### 5.15.1 Define Documentation

##### 5.15.1.1 #define OSGGA\_EVENT 1

## 5.16 GUIEventAdapter.cpp File Reference

```
#include <osgGA/GUIEventAdapter>
```

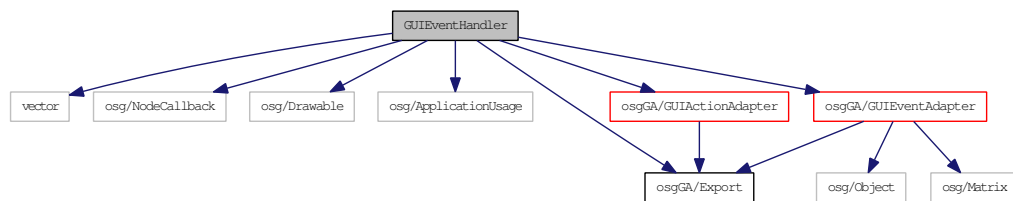
Include dependency graph for GUIEventAdapter.cpp:



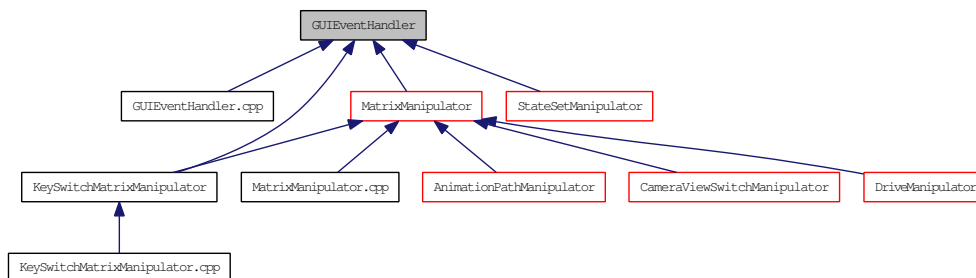
## 5.17 GUIEventHandler File Reference

```
#include <vector>
#include <osg/NodeCallback>
#include <osg/Drawable>
#include <osg/ApplicationUsage>
#include <osgGA/Export>
#include <osgGA/GUIEventAdapter>
#include <osgGA/GUIActionAdapter>
```

Include dependency graph for GUIEventHandler:



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



### Classes

- class **GUIEventHandler**

*GUIEventHandler* (p. 54) provides a basic interface for any class which wants to handle a GUI Events.

### Namespaces

- namespace **osgGA**

The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.

### Defines

- #define **OSGGA\_GUIEVENTHANDLER 1**

#### 5.17.1 Define Documentation

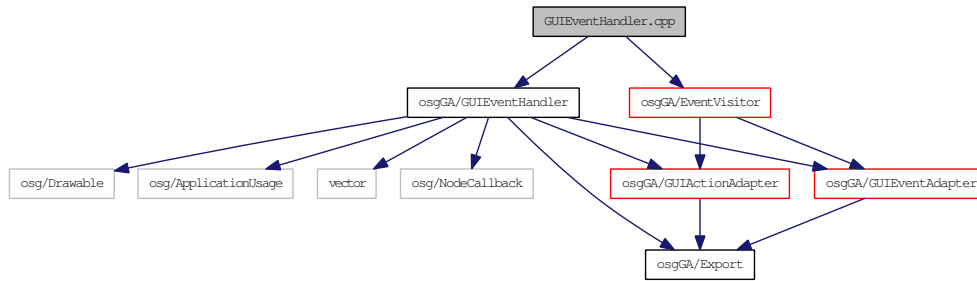
##### 5.17.1.1 #define OSGGA\_GUIEVENTHANDLER 1

## 5.18 GUIEventHandler.cpp File Reference

```
#include <osgGA/GUIEventHandler>
```

```
#include <osgGA/EventVisitor>
```

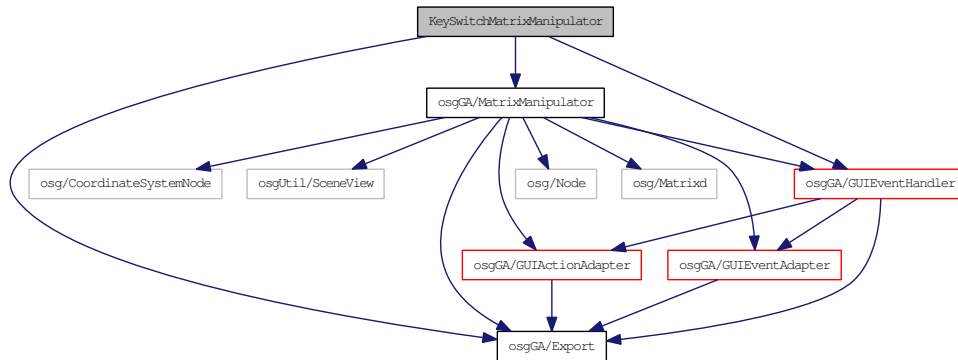
Include dependency graph for GUIEventHandler.cpp:



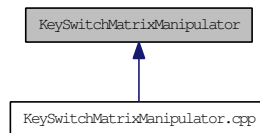
## 5.19 KeySwitchMatrixManipulator File Reference

```
#include <osgGA/Export>
#include <osgGA/MatrixManipulator>
#include <osgGA/GUIEventHandler>
```

Include dependency graph for KeySwitchMatrixManipulator:



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



### Classes

- class **KeySwitchMatrixManipulator**

**KeySwitchMatrixManipulator** (p. 57) is a decorator which allows the type of camera manipulator being used to be switched by pressing a key.

### Namespaces

- namespace **osgGA**

The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.

### Defines

- #define **OSGUTIL\_KEYSWITCMATRIXMANIPULATOR 1**

#### 5.19.1 Define Documentation

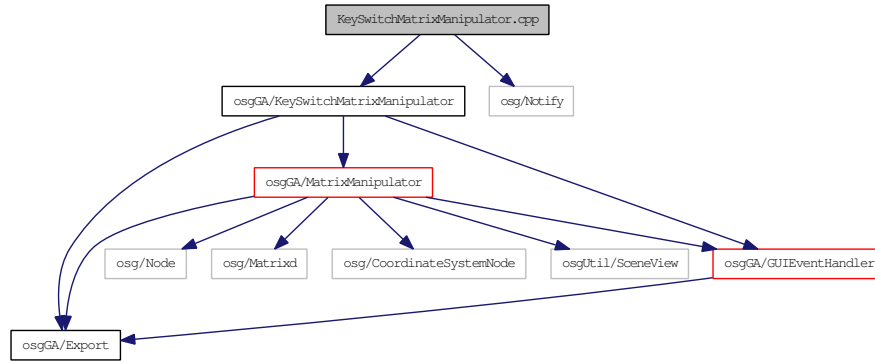
##### 5.19.1.1 #define OSGUTIL\_KEYSWITCMATRIXMANIPULATOR 1

## 5.20 KeySwitchMatrixManipulator.cpp File Reference

```
#include <osgGA/KeySwitchMatrixManipulator>
```

```
#include <osg/Notify>
```

Include dependency graph for KeySwitchMatrixManipulator.cpp:



## 5.21 mainpage.h File Reference

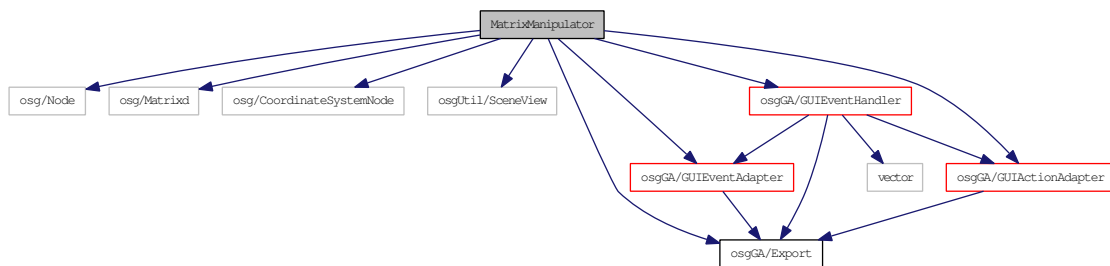
### 5.21.1 Detailed Description

This file contains doxygen special commands and text for the **Main Page** (p. ??) and some other minor aspects of this documentation. It is not part of the OSG.

## 5.22 MatrixManipulator File Reference

```
#include <osg/Node>
#include <osg/Matrixd>
#include <osg/CoordinateSystemNode>
#include <osgUtil/SceneView>
#include <osgGA/Export>
#include <osgGA/GUIEventHandler>
#include <osgGA/GUIEventAdapter>
#include <osgGA/GUIActionAdapter>
```

Include dependency graph for MatrixManipulator:



### Classes

- class **CoordinateFrameCallback**  
callback class to use to allow matrix manipulators to query the application for the local coordinate frame.
- class **MatrixManipulator**  
*MatrixManipulator* (p. 62) is an abstract base class defining the interface, and a certain amount of default functionality, for classes which wish to control OSG cameras in response to GUI events.

### Namespaces

- namespace **osgGA**  
The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.

### Defines

- #define **NEW\_HOME\_POSITION**
- #define **OSGGA\_MatrixManipulator** 1

#### 5.22.1 Define Documentation

5.22.1.1 #define **NEW\_HOME\_POSITION**

5.22.1.2 #define **OSGGA\_MatrixManipulator** 1

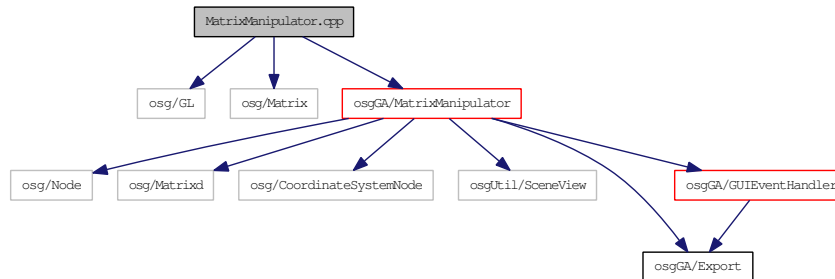
## 5.23 MatrixManipulator.cpp File Reference

```
#include <osg/GL>
```

```
#include <osg/Matrix>
```

```
#include <osgGA/MatrixManipulator>
```

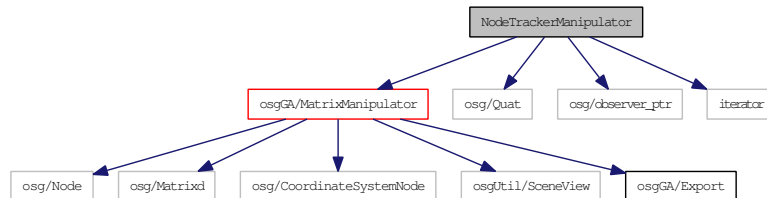
Include dependency graph for MatrixManipulator.cpp:



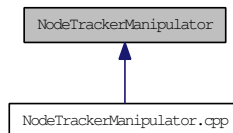
## 5.24 NodeTrackerManipulator File Reference

```
#include <osgGA/MatrixManipulator>
#include <osg/Quat>
#include <osg/observer_ptr>
#include <iterator>
```

Include dependency graph for NodeTrackerManipulator:



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



### Classes

- class **NodeTrackerManipulator**

### Namespaces

- namespace **osgGA**

*The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.*

### Defines

- #define **OSGGA\_NODETRACKERMANIPULATOR 1**

#### 5.24.1 Define Documentation

##### 5.24.1.1 #define OSGGA\_NODETRACKERMANIPULATOR 1

## 5.25 NodeTrackerManipulator.cpp File Reference

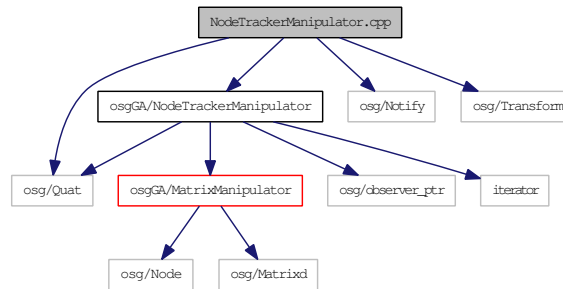
```
#include <osgGA/NodeTrackerManipulator>
```

```
#include <osg/Quat>
```

```
#include <osg/Notify>
```

```
#include <osg/Transform>
```

Include dependency graph for NodeTrackerManipulator.cpp:



### Variables

- const float **TRACKBALLSIZE** = 0.8f

#### 5.25.1 Variable Documentation

##### 5.25.1.1 const float TRACKBALLSIZE = 0.8f

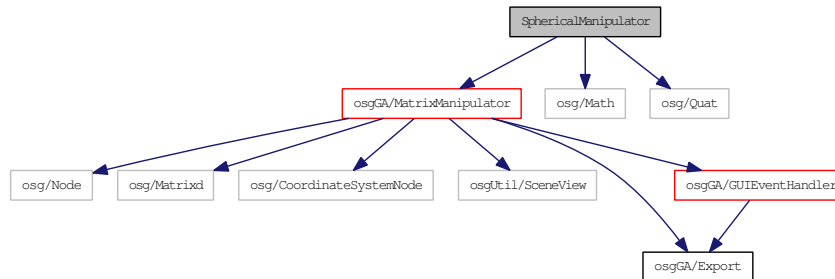
## 5.26 SphericalManipulator File Reference

```
#include <osgGA/MatrixManipulator>
```

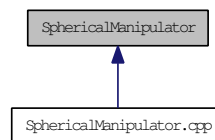
```
#include <osg/Math>
```

```
#include <osg/Quat>
```

Include dependency graph for SphericalManipulator:



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



### Classes

- class **SphericalManipulator**

### Namespaces

- namespace **osgGA**

The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.

## 5.27 SphericalManipulator.cpp File Reference

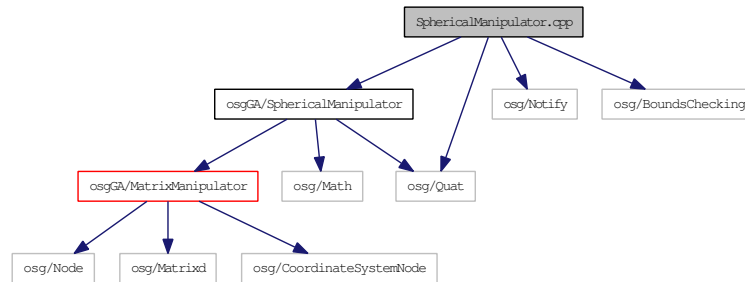
```
#include <osgGA/SphericalManipulator>
```

```
#include <osg/Quat>
```

```
#include <osg/Notify>
```

```
#include <osg/BoundsChecking>
```

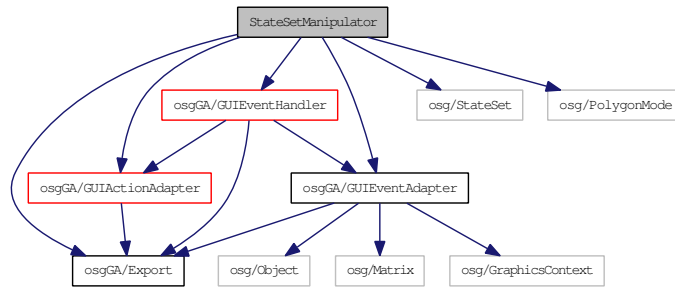
Include dependency graph for SphericalManipulator.cpp:



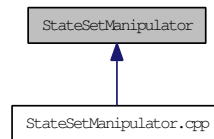
## 5.28 StateSetManipulator File Reference

```
#include <osgGA/Export>
#include <osgGA/GUIEventAdapter>
#include <osgGA/GUIActionAdapter>
#include <osgGA/GUIEventHandler>
#include <osg/StateSet>
#include <osg/PolygonMode>
```

Include dependency graph for StateSetManipulator:



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



### Classes

- class **StateSetManipulator**

*Experimental class, not been looked at for a while, but which will be returned to at some point :-\.*

### Namespaces

- namespace **osgGA**

*The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.*

### Defines

- #define **OSGGA\_STATESET\_MANIPULATOR** 1

#### 5.28.1 Define Documentation

##### 5.28.1.1 #define OSGGA\_STATESET\_MANIPULATOR 1

## 5.29 StateSetManipulator.cpp File Reference

```
#include <osgGA/StateSetManipulator>
```

```
#include <osg/PolygonMode>
```

```
#include <osg/ref_ptr>
```

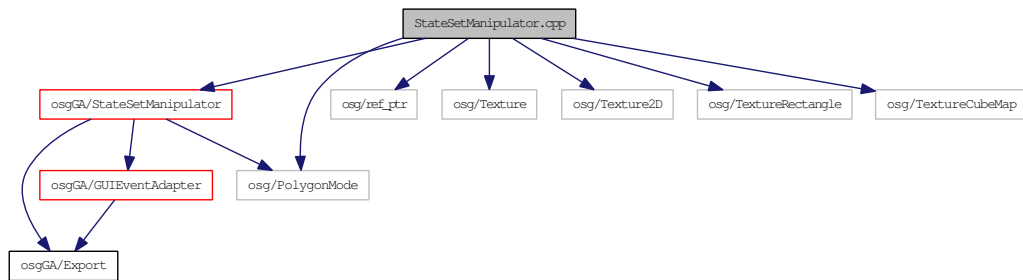
```
#include <osg/Texture>
```

```
#include <osg/Texture2D>
```

```
#include <osg/TextureRectangle>
```

```
#include <osg/TextureCubeMap>
```

Include dependency graph for StateSetManipulator.cpp:

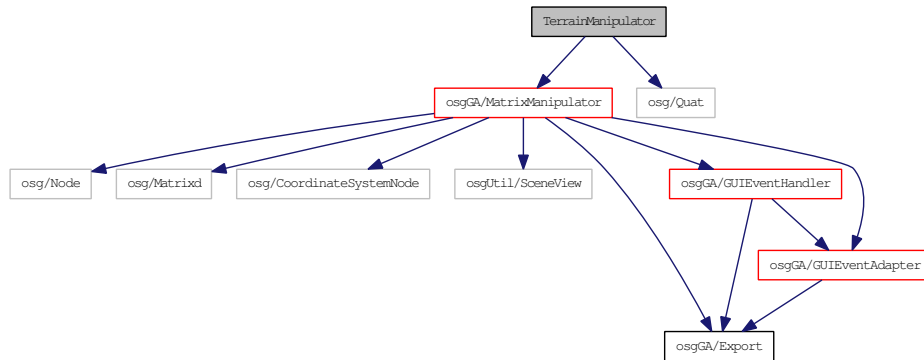


## 5.30 TerrainManipulator File Reference

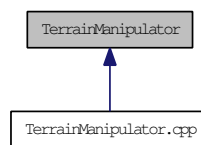
```
#include <osgGA/MatrixManipulator>
```

```
#include <osg/Quat>
```

Include dependency graph for TerrainManipulator:



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



### Classes

- class **TerrainManipulator**

### Namespaces

- namespace **osgGA**

The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.

### Defines

- #define **OSGGA\_TERRAINMANIPULATOR 1**

#### 5.30.1 Define Documentation

##### 5.30.1.1 #define OSGGA\_TERRAINMANIPULATOR 1

## 5.31 TerrainManipulator.cpp File Reference

```
#include <osgGA/TerrainManipulator>
```

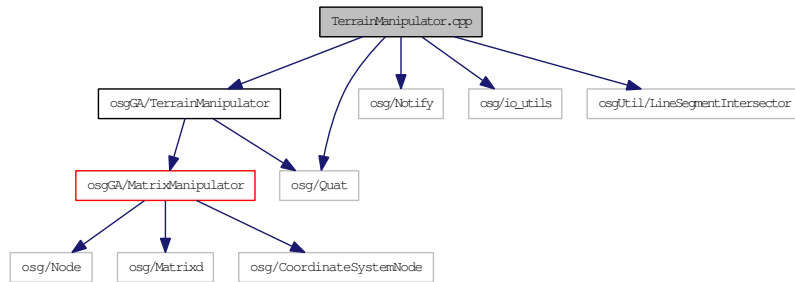
```
#include <osg/Quat>
```

```
#include <osg/Notify>
```

```
#include <osg/io_utils>
```

```
#include <osgUtil/LineSegmentIntersector>
```

Include dependency graph for TerrainManipulator.cpp:



### Variables

- const float **TRACKBALLSIZE** = 0.8f

#### 5.31.1 Variable Documentation

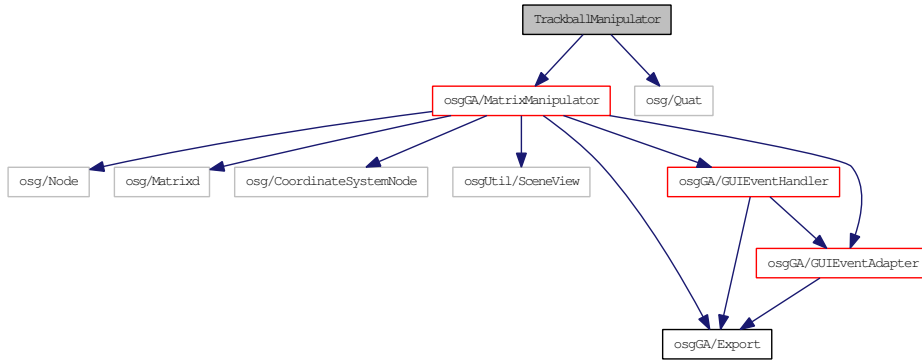
##### 5.31.1.1 const float TRACKBALLSIZE = 0.8f

## 5.32 TrackballManipulator File Reference

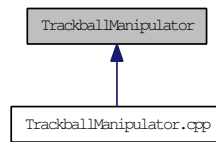
```
#include <osgGA/MatrixManipulator>
```

```
#include <osg/Quat>
```

Include dependency graph for TrackballManipulator:



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



### Classes

- class **TrackballManipulator**

### Namespaces

- namespace **osgGA**

The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.

### Defines

- #define **OSGGA\_TRACKBALLMANIPULATOR 1**

#### 5.32.1 Define Documentation

##### 5.32.1.1 #define OSGGA\_TRACKBALLMANIPULATOR 1

## 5.33 TrackballManipulator.cpp File Reference

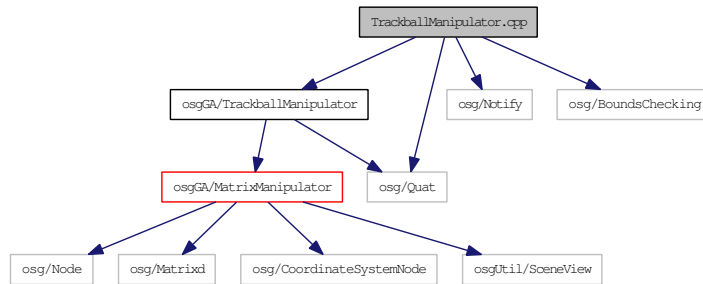
```
#include <osgGA/TrackballManipulator>
```

```
#include <osg/Quat>
```

```
#include <osg/Notify>
```

```
#include <osg/BoundsChecking>
```

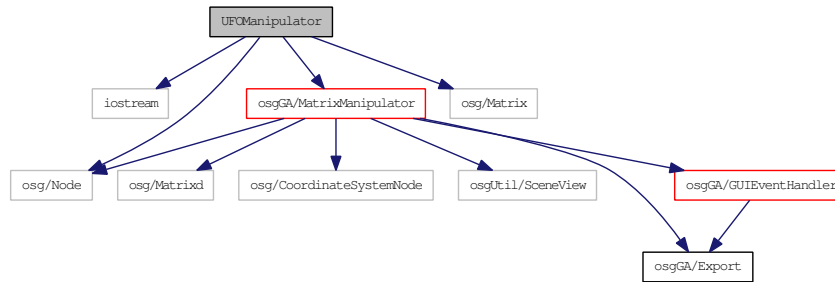
Include dependency graph for TrackballManipulator.cpp:



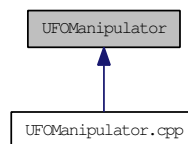
## 5.34 UFOManipulator File Reference

```
#include <iostream>
#include <osgGA/MatrixManipulator>
#include <osg/Node>
#include <osg/Matrix>
```

Include dependency graph for UFOManipulator:



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



### Classes

- class **UFOManipulator**  
A UFO manipulator driven with keybindings.

### Namespaces

- namespace **osgGA**  
The 'GA' in **osgGA** (p. 7) stands for 'GUI Abstraction'; the **osgGA** (p. 7) namespace provides facilities to help developers write the glue to allow the osg to work with varying window systems.

### Defines

- #define **OSGGA\_UFO\_MANIPULATOR\_DEF** 1

#### 5.34.1 Define Documentation

##### 5.34.1.1 #define OSGGA\_UFO\_MANIPULATOR\_DEF 1

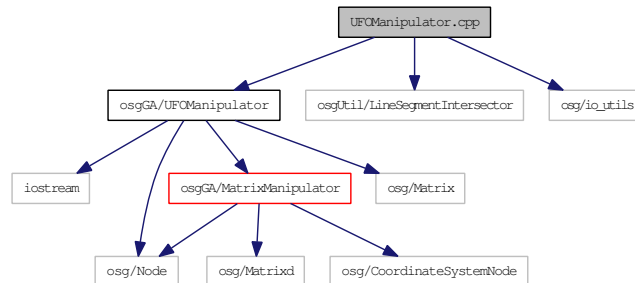
## 5.35 UFOManipulator.cpp File Reference

```
#include <osgGA/UFOManipulator>
```

```
#include <osgUtil/LineSegmentIntersector>
```

```
#include <osg/io_utils>
```

Include dependency graph for UFOManipulator.cpp:



### Defines

- #define **M\_PI** 3.14159265358979323846

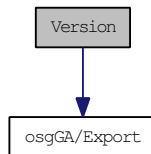
#### 5.35.1 Define Documentation

##### 5.35.1.1 #define **M\_PI** 3.14159265358979323846

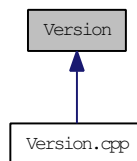
## 5.36 Version File Reference

```
#include <osgGA/Export>
```

Include dependency graph for Version:



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



### Defines

- #define **OSGGA\_VERSION** 1

### Functions

- OSGGA\_EXPORT const char \* **osgGAGetLibraryName** ()  
*getLibraryName\_osgGA()* returns the library name in human friendly form.
- OSGGA\_EXPORT const char \* **osgGAGetVersion** ()  
*osgGAGetVersion()* (p. 143) returns the library version number.

#### 5.36.1 Define Documentation

##### 5.36.1.1 #define OSGGA\_VERSION 1

#### 5.36.2 Function Documentation

##### 5.36.2.1 OSGGA\_EXPORT const char\* osgGAGetLibraryName ()

*getLibraryName\_osgGA()* returns the library name in human friendly form.

##### 5.36.2.2 OSGGA\_EXPORT const char\* osgGAGetVersion ()

**osgGAGetVersion()** (p. 143) returns the library version number. Numbering convention : OpenSceneGraph-1.0 will return 1.0 from *osgGAGetVersion*.

This C function can be also used to check for the existence of the OpenSceneGraph library using autoconf and its m4 macro `AC_CHECK_LIB`.

Here is the code to add to your configure.in:

```

#
# Check for the OpenSceneGraph (OSG) GA library
#
AC_CHECK_LIB(osg, osgGAGetVersion, ,
  [AC_MSG_ERROR(OpenSceneGraph GA library not found. See http://www.openscenegraph.org)],)

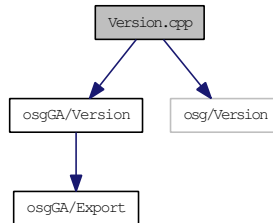
```

## 5.37 Version.cpp File Reference

```
#include <osgGA/Version>
```

```
#include <osg/Version>
```

Include dependency graph for Version.cpp:



### Functions

- `const char * osgGAGetLibraryName ()`  
*getLibraryName\_osgGA()* returns the library name in human friendly form.
- `const char * osgGAGetVersion ()`  
*osgGAGetVersion()* (p. 143) returns the library version number.

#### 5.37.1 Function Documentation

##### 5.37.1.1 `const char* osgGAGetLibraryName ()`

*getLibraryName\_osgGA()* returns the library name in human friendly form.

##### 5.37.1.2 `const char* osgGAGetVersion ()`

**osgGAGetVersion()** (p. 143) returns the library version number. Numbering convention : OpenSceneGraph-1.0 will return 1.0 from *osgGAGetVersion*.

This C function can be also used to check for the existence of the OpenSceneGraph library using *autoconf* and its *m4* macro `AC_CHECK_LIB`.

Here is the code to add to your *configure.in*:

```

#
# Check for the OpenSceneGraph (OSG) GA library
#
AC_CHECK_LIB(osg, osgGAGetVersion, ,
  [AC_MSG_ERROR(OpenSceneGraph GA library not found. See http://www.openscenegraph.org)],)

```

# Index

---

## - Symbols -

- ~CameraViewSwitchManipulator
  - osgGA::CameraViewSwitchManipulator, 14
- ~CoordinateFrameCallback
  - osgGA::MatrixManipulator::CoordinateFrameCallback, 17
- ~DriveManipulator
  - osgGA::DriveManipulator, 20
- ~EventQueue
  - osgGA::EventQueue, 26
- ~EventVisitor
  - osgGA::EventVisitor, 31
- ~FlightManipulator
  - osgGA::FlightManipulator, 35
- ~GUIActionAdapter
  - osgGA::GUIActionAdapter, 38
- ~GUIEventAdapter
  - osgGA::GUIEventAdapter, 49
- ~MatrixManipulator
  - osgGA::MatrixManipulator, 64
- ~NodeTrackerManipulator
  - osgGA::NodeTrackerManipulator, 70
- ~SphericalManipulator
  - osgGA::SphericalManipulator, 78
- ~StateSetManipulator
  - osgGA::StateSetManipulator, 84
- ~TerrainManipulator
  - osgGA::TerrainManipulator, 89
- ~TrackballManipulator
  - osgGA::TrackballManipulator, 95
- ~UFOManipulator
  - osgGA::UFOManipulator, 102
- \_Xmax
  - osgGA::GUIEventAdapter, 52
- \_Xmin
  - osgGA::GUIEventAdapter, 52
- \_Ymax
  - osgGA::GUIEventAdapter, 52
- \_Ymin
  - osgGA::GUIEventAdapter, 52
- \_acceleration
  - osgGA::FlightManipulator, 37
- \_accumulateEventState
  - osgGA::EventQueue, 29
  - osgGA::EventVisitor, 32
- \_actionAdapter
  - osgGA::EventVisitor, 32
- \_adjustPosition
  - osgGA::UFOManipulator, 102
- \_allowThrow
  - osgGA::SphericalManipulator, 80
  - osgGA::TrackballManipulator, 97
- \_animStartOfTimedPeriod
  - osgGA::AnimationPathManipulator, 12
- \_animationPath
  - osgGA::AnimationPathManipulator, 12
- \_autoComputeHomePosition
  - osgGA::MatrixManipulator, 67
- \_backface
  - osgGA::StateSetManipulator, 85
- \_buffer
  - osgGA::DriveManipulator, 22
- \_button
  - osgGA::GUIEventAdapter, 52
- \_buttonMask
  - osgGA::GUIEventAdapter, 52
- \_cameraViews
  - CollectCameraViewsNodeVisitor, 16
  - osgGA::CameraViewSwitchManipulator, 15
- \_center
  - osgGA::SphericalManipulator, 80
  - osgGA::TerrainManipulator, 91
  - osgGA::TrackballManipulator, 97
- \_context
  - osgGA::GUIEventAdapter, 52
- \_coordinateFrameCallback
  - osgGA::MatrixManipulator, 67
- \_ctrl
  - osgGA::UFOManipulator, 105
- \_currentView
  - osgGA::CameraViewSwitchManipulator, 15
- \_decelerateOffsetRate
  - osgGA::UFOManipulator, 105
- \_decelerateUpSideRate
  - osgGA::UFOManipulator, 105
- \_delta\_frame\_time
  - osgGA::SphericalManipulator, 80
  - osgGA::TrackballManipulator, 97
- \_direction
  - osgGA::UFOManipulator, 105
- \_directionRotationAcceleration
  - osgGA::UFOManipulator, 105
- \_directionRotationDeceleration
  - osgGA::UFOManipulator, 105
- \_directionRotationEpsilon
  - osgGA::UFOManipulator, 105
- \_directionRotationRate
  - osgGA::UFOManipulator, 105
- \_distance
  - osgGA::DriveManipulator, 22
  - osgGA::FlightManipulator, 37
  - osgGA::NodeTrackerManipulator, 73
  - osgGA::SphericalManipulator, 80
  - osgGA::TerrainManipulator, 91
  - osgGA::TrackballManipulator, 97
- \_dt
  - osgGA::UFOManipulator, 105
- \_elevation
  - osgGA::SphericalManipulator, 80
- \_eventQueue
  - osgGA::EventQueue, 29
- \_eventQueueMutex
  - osgGA::EventQueue, 29
- \_eventType
  - osgGA::GUIEventAdapter, 52
- \_events
  - osgGA::EventVisitor, 32
- \_eye
  - osgGA::DriveManipulator, 22
  - osgGA::FlightManipulator, 37

- \_forwardSpeed
  - osgGA::UFOManipulator, 105
- \_frame
  - osgGA::UFOManipulator, 102
- \_ga\_t0
  - osgGA::DriveManipulator, 22
  - osgGA::FlightManipulator, 37
  - osgGA::NodeTrackerManipulator, 73
  - osgGA::SphericalManipulator, 80
  - osgGA::TerrainManipulator, 91
  - osgGA::TrackballManipulator, 97
- \_ga\_t1
  - osgGA::DriveManipulator, 22
  - osgGA::FlightManipulator, 37
  - osgGA::NodeTrackerManipulator, 73
  - osgGA::SphericalManipulator, 80
  - osgGA::TerrainManipulator, 91
  - osgGA::TrackballManipulator, 97
- \_handled
  - osgGA::EventVisitor, 32
  - osgGA::GUIEventAdapter, 52
- \_heading
  - osgGA::SphericalManipulator, 80
- \_height
  - osgGA::DriveManipulator, 22
- \_homeCenter
  - osgGA::MatrixManipulator, 67
- \_homeDistance
  - osgGA::SphericalManipulator, 80
- \_homeEye
  - osgGA::MatrixManipulator, 67
- \_homeUp
  - osgGA::MatrixManipulator, 67
- \_ignoreHandledEventsMask
  - osgGA::GUIEventHandler, 56
- \_initialized
  - osgGA::StateSetManipulator, 85
- \_intersectTraversalMask
  - osgGA::MatrixManipulator, 67
- \_inverseMatrix
  - osgGA::UFOManipulator, 105
- \_isPaused
  - osgGA::AnimationPathManipulator, 12
- \_key
  - osgGA::GUIEventAdapter, 52
- \_keyDown
  - osgGA::UFOManipulator, 102
- \_keyEventCyclePolygonMode
  - osgGA::StateSetManipulator, 85
- \_keyEventToggleBackfaceCulling
  - osgGA::StateSetManipulator, 85
- \_keyEventToggleLighting
  - osgGA::StateSetManipulator, 85
- \_keyEventToggleTexturing
  - osgGA::StateSetManipulator, 85
- \_keyUp
  - osgGA::UFOManipulator, 102
- \_last\_frame\_time
  - osgGA::SphericalManipulator, 80
  - osgGA::TrackballManipulator, 97
- \_lighting
  - osgGA::StateSetManipulator, 85
- \_matrix
  - osgGA::AnimationPathManipulator, 12
  - osgGA::UFOManipulator, 105
- \_maxNumOfTextureUnits
  - osgGA::StateSetManipulator, 85
- \_minDistanceInFront
  - osgGA::UFOManipulator, 105
- \_minHeightAboveGround
  - osgGA::UFOManipulator, 105
- \_minimumDistance
  - osgGA::MatrixManipulator, 67
- \_minimumZoomScale
  - osgGA::SphericalManipulator, 80
  - osgGA::TrackballManipulator, 97
- \_modKeyMask
  - osgGA::GUIEventAdapter, 52
- \_modelScale
  - osgGA::DriveManipulator, 22
  - osgGA::FlightManipulator, 37
  - osgGA::SphericalManipulator, 81
  - osgGA::TrackballManipulator, 97
- \_mouseYOrientation
  - osgGA::GUIEventAdapter, 52
- \_mx
  - osgGA::GUIEventAdapter, 52
- \_my
  - osgGA::GUIEventAdapter, 52
- \_node
  - osgGA::CameraViewSwitchManipulator, 15
  - osgGA::DriveManipulator, 22
  - osgGA::FlightManipulator, 37
  - osgGA::NodeTrackerManipulator, 73
  - osgGA::SphericalManipulator, 81
  - osgGA::TerrainManipulator, 91
  - osgGA::TrackballManipulator, 97
  - osgGA::UFOManipulator, 105
- \_nodeRotation
  - osgGA::NodeTrackerManipulator, 73
- \_numOfFramesSinceStartOfTimedPeriod
  - osgGA::AnimationPathManipulator, 12
- \_offset
  - osgGA::UFOManipulator, 105
- \_pauseTime
  - osgGA::AnimationPathManipulator, 12
- \_pitch
  - osgGA::DriveManipulator, 22
- \_pitchDownKeyPressed
  - osgGA::DriveManipulator, 22
- \_pitchOffset
  - osgGA::UFOManipulator, 105
- \_pitchOffsetRate
  - osgGA::UFOManipulator, 105
- \_pitchUpKeyPressed
  - osgGA::DriveManipulator, 22
- \_position
  - osgGA::UFOManipulator, 105
- \_previousUp
  - osgGA::TerrainManipulator, 91
- \_printOutTimingInfo
  - osgGA::AnimationPathManipulator, 12
- \_realStartOfTimedPeriod
  - osgGA::AnimationPathManipulator, 12
- \_rotation
  - osgGA::DriveManipulator, 22
  - osgGA::FlightManipulator, 37
  - osgGA::NodeTrackerManipulator, 73
  - osgGA::TerrainManipulator, 91
  - osgGA::TrackballManipulator, 97
- \_rotationMode
  - osgGA::NodeTrackerManipulator, 73

- osgGA::SphericalManipulator, 81
- osgGA::TerrainManipulator, 91
- \_scrolling
  - osgGA::GUIEventAdapter, 52
- \_shift
  - osgGA::UFOManipulator, 105
- \_sideSpeed
  - osgGA::UFOManipulator, 105
- \_speedAccelerationFactor
  - osgGA::UFOManipulator, 105
- \_speedDecelerationFactor
  - osgGA::UFOManipulator, 105
- \_speedEpsilon
  - osgGA::UFOManipulator, 105
- \_speedMode
  - osgGA::DriveManipulator, 22
- \_startTick
  - osgGA::EventQueue, 29
- \_stateset
  - osgGA::StateSetManipulator, 85
- \_stop
  - osgGA::UFOManipulator, 102
- \_straightenOffset
  - osgGA::UFOManipulator, 105
- \_t0
  - osgGA::UFOManipulator, 105
- \_tabletPen
  - osgGA::GUIEventAdapter, 52
- \_texture
  - osgGA::StateSetManipulator, 85
- \_thrown
  - osgGA::NodeTrackerManipulator, 73
  - osgGA::SphericalManipulator, 81
  - osgGA::TerrainManipulator, 91
  - osgGA::TrackballManipulator, 97
- \_time
  - osgGA::GUIEventAdapter, 52
- \_timeOffset
  - osgGA::AnimationPathManipulator, 12
- \_timeScale
  - osgGA::AnimationPathManipulator, 12
- \_trackNodePath
  - osgGA::NodeTrackerManipulator, 73
- \_trackballSize
  - osgGA::TrackballManipulator, 97
- \_trackerMode
  - osgGA::NodeTrackerManipulator, 73
- \_upSpeed
  - osgGA::UFOManipulator, 105
- \_useFixedMouseInputRange
  - osgGA::EventQueue, 29
- \_valid
  - osgGA::AnimationPathManipulator, 12
- \_velocity
  - osgGA::DriveManipulator, 22
  - osgGA::FlightManipulator, 37
- \_viewAngle
  - osgGA::UFOManipulator, 105
- \_viewOffsetDelta
  - osgGA::UFOManipulator, 105
- \_windowHeight
  - osgGA::GUIEventAdapter, 52
- \_windowWidth
  - osgGA::GUIEventAdapter, 52
- \_windowX
  - osgGA::GUIEventAdapter, 52

- \_windowY
  - osgGA::GUIEventAdapter, 52
- \_yawMode
  - osgGA::FlightManipulator, 37
- \_yawOffset
  - osgGA::UFOManipulator, 105
- \_yawOffsetRate
  - osgGA::UFOManipulator, 105
- \_zoomDelta
  - osgGA::SphericalManipulator, 81
  - osgGA::TrackballManipulator, 97

**- A -**

- addEvent
  - osgGA::EventQueue, 26
  - osgGA::EventVisitor, 31
- addMatrixManipulator
  - osgGA::KeySwitchMatrixManipulator, 59
- addMouseEvent
  - osgGA::DriveManipulator, 20
  - osgGA::FlightManipulator, 35
  - osgGA::NodeTrackerManipulator, 70
  - osgGA::SphericalManipulator, 78
  - osgGA::TerrainManipulator, 89
  - osgGA::TrackballManipulator, 95
- addNumberedMatrixManipulator
  - osgGA::KeySwitchMatrixManipulator, 59
- AnimationPathManipulator, 107
  - osgGA::AnimationPathManipulator, 11
  - OSGGA\_ANIMATION\_PATH\_MANIPULATOR, 107
- AnimationPathManipulator.cpp, 108
- appendEvents
  - osgGA::EventQueue, 26
- apply
  - CollectCameraViewsNodeVisitor, 16
  - osgGA::EventVisitor, 31
- asView
  - osgGA::GUIActionAdapter, 38

**- C -**

- calcMovement
  - osgGA::DriveManipulator, 20
  - osgGA::FlightManipulator, 35
  - osgGA::NodeTrackerManipulator, 70
  - osgGA::SphericalManipulator, 78
  - osgGA::TerrainManipulator, 89
  - osgGA::TrackballManipulator, 95
- CameraViewList
  - osgGA::CameraViewSwitchManipulator, 14
- CameraViewSwitchManipulator, 109
  - osgGA::CameraViewSwitchManipulator, 14
  - OSGGA\_VIEWLISTMANIPULATOR, 109
- CameraViewSwitchManipulator.cpp, 110
- clampOrientation
  - osgGA::NodeTrackerManipulator, 71
  - osgGA::TerrainManipulator, 89
- className
  - osgGA::AnimationPathManipulator, 11
  - osgGA::CameraViewSwitchManipulator, 14
  - osgGA::DriveManipulator, 20
  - osgGA::FlightManipulator, 35
  - osgGA::KeySwitchMatrixManipulator, 59
  - osgGA::MatrixManipulator, 64
  - osgGA::NodeTrackerManipulator, 71
  - osgGA::SphericalManipulator, 78

- osgGA::StateSetManipulator, 84
- osgGA::TerrainManipulator, 89
- osgGA::TrackballManipulator, 95
- osgGA::UFOManipulator, 102
- clone
  - osgGA::StateSetManipulator, 84
- CLOSE\_WINDOW
  - osgGA::GUIEventAdapter, 45
- closeWindow
  - osgGA::EventQueue, 26
- CollectCameraViewsNodeVisitor, 16
  - \_cameraViews, 16
  - apply, 16
  - CollectCameraViewsNodeVisitor, 16
- computeHomePosition
  - osgGA::DriveManipulator, 20
  - osgGA::KeySwitchMatrixManipulator, 59
  - osgGA::MatrixManipulator, 64
  - osgGA::NodeTrackerManipulator, 71
  - osgGA::SphericalManipulator, 78
  - osgGA::UFOManipulator, 102
- computeNodeCenterAndRotation
  - osgGA::NodeTrackerManipulator, 71
- computeNodeLocalToWorld
  - osgGA::NodeTrackerManipulator, 71
- computeNodeWorldToLocal
  - osgGA::NodeTrackerManipulator, 71
- computePosition
  - osgGA::DriveManipulator, 20
  - osgGA::FlightManipulator, 35
  - osgGA::NodeTrackerManipulator, 71
  - osgGA::TerrainManipulator, 89
  - osgGA::TrackballManipulator, 95
- computeViewPosition
  - osgGA::SphericalManipulator, 78
- copyEvents
  - osgGA::EventQueue, 26
- createEvent
  - osgGA::EventQueue, 26
- cyclePolygonMode
  - osgGA::StateSetManipulator, 84
- D -**
- deltaX
  - osgGA::GUIEventAdapter::Scrolling, 74
- deltaY
  - osgGA::GUIEventAdapter::Scrolling, 74
- DOUBLECLICK
  - osgGA::GUIEventAdapter, 45
- DRAG
  - osgGA::GUIEventAdapter, 45
- DriveManipulator, 111
  - osgGA::DriveManipulator, 20
  - OSGGA\_DRIVEMANIPULATOR, 111
- DriveManipulator.cpp, 112
  - DRIVER\_HEIGHT, 112
  - getHeightOfDriver, 112
  - KEYBOARD\_PITCH, 112
- DRIVER\_HEIGHT
  - DriveManipulator.cpp, 112
- E -**
- ELEVATION
  - osgGA::SphericalManipulator, 78
- ELEVATION\_AZIM
  - osgGA::NodeTrackerManipulator, 70
- ELEVATION\_HEADING
  - osgGA::SphericalManipulator, 78
  - osgGA::TerrainManipulator, 89
- ELEVATION\_HEADING\_ROLL
  - osgGA::TerrainManipulator, 89
- ERASER
  - osgGA::GUIEventAdapter, 49
- event
  - osgGA::GUIEventHandler, 55
- EventList
  - osgGA::EventVisitor, 31
- EventQueue, 113
  - osgGA::EventQueue, 26
  - OSGGA\_EVENTQUEUE, 113
- EventQueue.cpp, 114
- Events
  - osgGA::EventQueue, 26
- EventType
  - osgGA::GUIEventAdapter, 45
- EventVisitor, 115
  - osgGA::EventVisitor, 31
  - OSGGA\_EVENTVISITOR, 115
- EventVisitor.cpp, 116
- Export, 117
  - OSGGA\_EXPORT, 117
  - OSGGA\_EXPORT\_, 117
- F -**
- FlightManipulator, 118
  - osgGA::FlightManipulator, 35
  - OSGGA\_FLIGHTMANIPULATOR, 118
- FlightManipulator.cpp, 119
- flushMouseEventStack
  - osgGA::DriveManipulator, 20
  - osgGA::FlightManipulator, 35
  - osgGA::NodeTrackerManipulator, 71
  - osgGA::SphericalManipulator, 78
  - osgGA::TerrainManipulator, 89
  - osgGA::TrackballManipulator, 95
- FRAME
  - osgGA::GUIEventAdapter, 45
- frame
  - osgGA::EventQueue, 26
- G -**
- getAcceleration
  - osgGA::FlightManipulator, 35
- getAccumulatedEventState
  - osgGA::GUIEventAdapter, 49
- getActionAdapter
  - osgGA::EventVisitor, 31
- getAllowThrow
  - osgGA::SphericalManipulator, 78
  - osgGA::TrackballManipulator, 95
- getAnimationPath
  - osgGA::AnimationPathManipulator, 11
- getAutoComputeHomePosition
  - osgGA::MatrixManipulator, 64
- getBackfaceEnabled
  - osgGA::StateSetManipulator, 84
- getButton
  - osgGA::GUIEventAdapter, 49
- getButtonMask
  - osgGA::GUIEventAdapter, 49

- getCenter
  - osgGA::SphericalManipulator, 78
  - osgGA::TrackballManipulator, 95
- getCoordinateFrame
  - osgGA::MatrixManipulator, 64
  - osgGA::MatrixManipulator::CoordinateFrameCallback, 17
- getCoordinateFrameCallback
  - osgGA::MatrixManipulator, 64
- getCurrentEventState
  - osgGA::EventQueue, 26
- getCurrentMatrixManipulator
  - osgGA::KeySwitchMatrixManipulator, 59
- getCurrentPositionAsLookAt
  - osgGA::UFOManipulator, 102
- getDistance
  - osgGA::SphericalManipulator, 78
  - osgGA::TrackballManipulator, 95
- getElevation
  - osgGA::SphericalManipulator, 78
- getEventHandled
  - osgGA::EventVisitor, 31
- getEvents
  - osgGA::EventVisitor, 31
- getEventType
  - osgGA::GUIEventAdapter, 49
- getForwardSpeed
  - osgGA::UFOManipulator, 102
- getFrontVector
  - osgGA::MatrixManipulator, 64
- getFusionDistanceMode
  - osgGA::KeySwitchMatrixManipulator, 59
  - osgGA::MatrixManipulator, 64
  - osgGA::NodeTrackerManipulator, 71
  - osgGA::SphericalManipulator, 78
  - osgGA::TerrainManipulator, 89
  - osgGA::TrackballManipulator, 95
- getFusionDistanceValue
  - osgGA::KeySwitchMatrixManipulator, 60
  - osgGA::MatrixManipulator, 64
  - osgGA::NodeTrackerManipulator, 71
  - osgGA::SphericalManipulator, 78
  - osgGA::TerrainManipulator, 89
  - osgGA::TrackballManipulator, 95
- getGraphicsContext
  - osgGA::GUIEventAdapter, 49
- getHandled
  - osgGA::GUIEventAdapter, 49
- getHeading
  - osgGA::SphericalManipulator, 78
- getHeight
  - osgGA::DriveManipulator, 20
- getHeightOfDriver
  - DriveManipulator.cpp, 112
- getHomeDistance
  - osgGA::SphericalManipulator, 79
- getHomePosition
  - osgGA::MatrixManipulator, 65
- getIgnoreHandledEventsMask
  - osgGA::GUIEventHandler, 55
- getIntersectTraversalMask
  - osgGA::MatrixManipulator, 65
- getInverseMatrix
  - osgGA::AnimationPathManipulator, 11
  - osgGA::CameraViewSwitchManipulator, 14
  - osgGA::DriveManipulator, 21
  - osgGA::FlightManipulator, 35
  - osgGA::KeySwitchMatrixManipulator, 60
  - osgGA::MatrixManipulator, 65
  - osgGA::NodeTrackerManipulator, 71
  - osgGA::SphericalManipulator, 79
  - osgGA::TerrainManipulator, 89
  - osgGA::TrackballManipulator, 95
- osgGA::FlightManipulator, 35
- osgGA::KeySwitchMatrixManipulator, 60
- osgGA::MatrixManipulator, 65
- osgGA::NodeTrackerManipulator, 71
- osgGA::SphericalManipulator, 79
- osgGA::TerrainManipulator, 89
- osgGA::TrackballManipulator, 95
- osgGA::UFOManipulator, 102
- getKey
  - osgGA::GUIEventAdapter, 49
- getKeyEventCyclePolygonMode
  - osgGA::StateSetManipulator, 84
- getKeyEventToggleBackfaceCulling
  - osgGA::StateSetManipulator, 84
- getKeyEventToggleLighting
  - osgGA::StateSetManipulator, 84
- getKeyEventToggleTexturing
  - osgGA::StateSetManipulator, 84
- getKeyManipMap
  - osgGA::KeySwitchMatrixManipulator, 60
- getLightingEnabled
  - osgGA::StateSetManipulator, 84
- getMatrix
  - osgGA::AnimationPathManipulator, 11
  - osgGA::CameraViewSwitchManipulator, 14
  - osgGA::DriveManipulator, 21
  - osgGA::FlightManipulator, 35
  - osgGA::KeySwitchMatrixManipulator, 60
  - osgGA::MatrixManipulator, 65
  - osgGA::NodeTrackerManipulator, 71
  - osgGA::SphericalManipulator, 79
  - osgGA::TerrainManipulator, 89
  - osgGA::TrackballManipulator, 95
  - osgGA::UFOManipulator, 102
- getMatrixManipulatorWithIndex
  - osgGA::KeySwitchMatrixManipulator, 60
- getMatrixManipulatorWithKey
  - osgGA::KeySwitchMatrixManipulator, 60
- getMaximumNumOfTextureUnits
  - osgGA::StateSetManipulator, 84
- getMinDistance
  - osgGA::UFOManipulator, 102
- getMinHeight
  - osgGA::UFOManipulator, 102
- getMinimumDistance
  - osgGA::MatrixManipulator, 65
- getMinimumZoomScale
  - osgGA::SphericalManipulator, 79
  - osgGA::TrackballManipulator, 95
- getModelScale
  - osgGA::DriveManipulator, 21
  - osgGA::FlightManipulator, 35
- getModKeyMask
  - osgGA::GUIEventAdapter, 49
- getMouseYOrientation
  - osgGA::GUIEventAdapter, 50
- getNode
  - osgGA::CameraViewSwitchManipulator, 14
  - osgGA::DriveManipulator, 21
  - osgGA::FlightManipulator, 35
  - osgGA::KeySwitchMatrixManipulator, 60
  - osgGA::MatrixManipulator, 65
  - osgGA::NodeTrackerManipulator, 71
  - osgGA::SphericalManipulator, 79
  - osgGA::TerrainManipulator, 90
  - osgGA::TrackballManipulator, 95

- osgGA::UFOManipulator, 102
  - getNodePath
    - osgGA::NodeTrackerManipulator, 71
  - getNumMatrixManipulators
    - osgGA::KeySwitchMatrixManipulator, 60
  - getOrCreatePolygonMode
    - osgGA::StateSetManipulator, 84
  - getPenOrientation
    - osgGA::GUIEventAdapter, 50
  - getPenPressure
    - osgGA::GUIEventAdapter, 50
  - getPenRotation
    - osgGA::GUIEventAdapter, 50
  - getPenTiltX
    - osgGA::GUIEventAdapter, 50
  - getPenTiltY
    - osgGA::GUIEventAdapter, 50
  - getPolygonMode
    - osgGA::StateSetManipulator, 84
  - getPrintOutTimingInfo
    - osgGA::AnimationPathManipulator, 11
  - getRotation
    - osgGA::TrackballManipulator, 96
  - getRotationMode
    - osgGA::NodeTrackerManipulator, 72
    - osgGA::SphericalManipulator, 79
    - osgGA::TerrainManipulator, 90
  - getRotationSpeed
    - osgGA::UFOManipulator, 103
  - getScrollingDeltaX
    - osgGA::GUIEventAdapter, 50
  - getScrollingDeltaY
    - osgGA::GUIEventAdapter, 50
  - getScrollingMotion
    - osgGA::GUIEventAdapter, 50
  - getScrollWheelZoomDelta
    - osgGA::SphericalManipulator, 79
    - osgGA::TrackballManipulator, 96
  - getSideSpeed
    - osgGA::UFOManipulator, 103
  - getSideVector
    - osgGA::MatrixManipulator, 65
  - getStartTick
    - osgGA::EventQueue, 26
  - getStateSet
    - osgGA::StateSetManipulator, 84
  - getTabletPointerType
    - osgGA::GUIEventAdapter, 50
  - getTextureEnabled
    - osgGA::StateSetManipulator, 84
  - getTime
    - osgGA::EventQueue, 26
    - osgGA::GUIEventAdapter, 50
  - getTrackballSize
    - osgGA::TrackballManipulator, 96
  - getTrackerMode
    - osgGA::NodeTrackerManipulator, 72
  - getTrackNode
    - osgGA::NodeTrackerManipulator, 72
  - getTrackNodePath
    - osgGA::NodeTrackerManipulator, 72
  - getUpVector
    - osgGA::MatrixManipulator, 65
  - getUsage
    - osgGA::AnimationPathManipulator, 11
    - osgGA::CameraViewSwitchManipulator, 15
    - osgGA::DriveManipulator, 21
    - osgGA::FlightManipulator, 36
    - osgGA::GUIEventHandler, 55
    - osgGA::KeySwitchMatrixManipulator, 60
    - osgGA::NodeTrackerManipulator, 72
    - osgGA::SphericalManipulator, 79
    - osgGA::StateSetManipulator, 84
    - osgGA::TerrainManipulator, 90
    - osgGA::TrackballManipulator, 96
    - osgGA::UFOManipulator, 103
  - getUseFixedMouseInputRange
    - osgGA::EventQueue, 26
  - getVelocity
    - osgGA::DriveManipulator, 21
    - osgGA::FlightManipulator, 36
  - getWindowHeight
    - osgGA::GUIEventAdapter, 50
  - getWindowWidth
    - osgGA::GUIEventAdapter, 50
  - getWindowX
    - osgGA::GUIEventAdapter, 50
  - getWindowY
    - osgGA::GUIEventAdapter, 50
  - getX
    - osgGA::GUIEventAdapter, 50
  - getXmax
    - osgGA::GUIEventAdapter, 50
  - getXmin
    - osgGA::GUIEventAdapter, 50
  - getXnormalized
    - osgGA::GUIEventAdapter, 50
  - getY
    - osgGA::GUIEventAdapter, 51
  - getYmax
    - osgGA::GUIEventAdapter, 51
  - getYmin
    - osgGA::GUIEventAdapter, 51
  - getYnormalized
    - osgGA::GUIEventAdapter, 51
  - GUIActionAdapter, 120
    - OSGGA\_GUIACTIONADAPTER, 120
  - GUIEventAdapter, 121
    - osgGA::GUIEventAdapter, 49
    - OSGGA\_EVENT, 121
  - GUIEventAdapter.cpp, 122
  - GUIEventHandler, 123
    - osgGA::GUIEventHandler, 55
    - OSGGA\_GUIEVENTHANDLER, 123
  - GUIEventHandler.cpp, 124
- H -**
- handle
    - osgGA::AnimationPathManipulator, 11
    - osgGA::CameraViewSwitchManipulator, 15
    - osgGA::DriveManipulator, 21
    - osgGA::FlightManipulator, 36
    - osgGA::GUIEventHandler, 55
    - osgGA::KeySwitchMatrixManipulator, 60
    - osgGA::MatrixManipulator, 65
    - osgGA::NodeTrackerManipulator, 72
    - osgGA::SphericalManipulator, 79
    - osgGA::StateSetManipulator, 84
    - osgGA::TerrainManipulator, 90
    - osgGA::TrackballManipulator, 96
    - osgGA::UFOManipulator, 103
  - handle\_callbacks

- osgGA::EventVisitor, 31
- handle\_callbacks\_and\_traverse
  - osgGA::EventVisitor, 31
- handle\_geode\_callbacks
  - osgGA::EventVisitor, 31
- handleFrame
  - osgGA::AnimationPathManipulator, 11
- handleWithCheckAgainstIgnoreHandledEventsMask
  - osgGA::GUIEventHandler, 55
- HEADING
  - osgGA::SphericalManipulator, 78
- home
  - osgGA::AnimationPathManipulator, 11
  - osgGA::DriveManipulator, 21
  - osgGA::FlightManipulator, 36
  - osgGA::KeySwitchMatrixManipulator, 60
  - osgGA::MatrixManipulator, 65, 66
  - osgGA::NodeTrackerManipulator, 72
  - osgGA::SphericalManipulator, 79
  - osgGA::TerrainManipulator, 90
  - osgGA::TrackballManipulator, 96
  - osgGA::UFOManipulator, 103

**- I -**

- include/ Directory Reference, 3
- include/osgGA/ Directory Reference, 5
- init
  - osgGA::AnimationPathManipulator, 11
  - osgGA::CameraViewSwitchManipulator, 15
  - osgGA::DriveManipulator, 21
  - osgGA::FlightManipulator, 36
  - osgGA::KeySwitchMatrixManipulator, 61
  - osgGA::MatrixManipulator, 66
  - osgGA::NodeTrackerManipulator, 72
  - osgGA::SphericalManipulator, 79
  - osgGA::TerrainManipulator, 90
  - osgGA::TrackballManipulator, 96
  - osgGA::UFOManipulator, 103
- intersect
  - osgGA::DriveManipulator, 21
  - osgGA::TerrainManipulator, 90
  - osgGA::UFOManipulator, 103
- isMouseMoveing
  - osgGA::NodeTrackerManipulator, 72
  - osgGA::SphericalManipulator, 79
  - osgGA::TerrainManipulator, 90
  - osgGA::TrackballManipulator, 96

**- K -**

- KEY\_Alt\_L
  - osgGA::GUIEventAdapter, 48
- KEY\_Alt\_R
  - osgGA::GUIEventAdapter, 48
- KEY\_BackSpace
  - osgGA::GUIEventAdapter, 45
- KEY\_Begin
  - osgGA::GUIEventAdapter, 46
- KEY\_Break
  - osgGA::GUIEventAdapter, 46
- KEY\_Cancel
  - osgGA::GUIEventAdapter, 46
- KEY\_Caps\_Lock
  - osgGA::GUIEventAdapter, 48
- KEY\_Clear
  - osgGA::GUIEventAdapter, 45

- KEY\_Control\_L
  - osgGA::GUIEventAdapter, 48
- KEY\_Control\_R
  - osgGA::GUIEventAdapter, 48
- KEY\_Delete
  - osgGA::GUIEventAdapter, 45
- KEY\_Down
  - osgGA::GUIEventAdapter, 46
- KEY\_End
  - osgGA::GUIEventAdapter, 46
- KEY\_Escape
  - osgGA::GUIEventAdapter, 45
- KEY\_Execute
  - osgGA::GUIEventAdapter, 46
- KEY\_F1
  - osgGA::GUIEventAdapter, 47
- KEY\_F10
  - osgGA::GUIEventAdapter, 47
- KEY\_F11
  - osgGA::GUIEventAdapter, 47
- KEY\_F12
  - osgGA::GUIEventAdapter, 47
- KEY\_F13
  - osgGA::GUIEventAdapter, 47
- KEY\_F14
  - osgGA::GUIEventAdapter, 47
- KEY\_F15
  - osgGA::GUIEventAdapter, 47
- KEY\_F16
  - osgGA::GUIEventAdapter, 47
- KEY\_F17
  - osgGA::GUIEventAdapter, 47
- KEY\_F18
  - osgGA::GUIEventAdapter, 47
- KEY\_F19
  - osgGA::GUIEventAdapter, 47
- KEY\_F2
  - osgGA::GUIEventAdapter, 47
- KEY\_F20
  - osgGA::GUIEventAdapter, 47
- KEY\_F21
  - osgGA::GUIEventAdapter, 47
- KEY\_F22
  - osgGA::GUIEventAdapter, 47
- KEY\_F23
  - osgGA::GUIEventAdapter, 47
- KEY\_F24
  - osgGA::GUIEventAdapter, 47
- KEY\_F25
  - osgGA::GUIEventAdapter, 47
- KEY\_F26
  - osgGA::GUIEventAdapter, 47
- KEY\_F27
  - osgGA::GUIEventAdapter, 47
- KEY\_F28
  - osgGA::GUIEventAdapter, 47
- KEY\_F29
  - osgGA::GUIEventAdapter, 47
- KEY\_F3
  - osgGA::GUIEventAdapter, 47
- KEY\_F30
  - osgGA::GUIEventAdapter, 47
- KEY\_F31
  - osgGA::GUIEventAdapter, 47
- KEY\_F32
  - osgGA::GUIEventAdapter, 47

KEY\_F33  
     osgGA::GUIEventAdapter, 48  
 KEY\_F34  
     osgGA::GUIEventAdapter, 48  
 KEY\_F35  
     osgGA::GUIEventAdapter, 48  
 KEY\_F4  
     osgGA::GUIEventAdapter, 47  
 KEY\_F5  
     osgGA::GUIEventAdapter, 47  
 KEY\_F6  
     osgGA::GUIEventAdapter, 47  
 KEY\_F7  
     osgGA::GUIEventAdapter, 47  
 KEY\_F8  
     osgGA::GUIEventAdapter, 47  
 KEY\_F9  
     osgGA::GUIEventAdapter, 47  
 KEY\_Find  
     osgGA::GUIEventAdapter, 46  
 KEY\_Help  
     osgGA::GUIEventAdapter, 46  
 KEY\_Home  
     osgGA::GUIEventAdapter, 45  
 KEY\_Hyper\_L  
     osgGA::GUIEventAdapter, 48  
 KEY\_Hyper\_R  
     osgGA::GUIEventAdapter, 48  
 KEY\_Insert  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_0  
     osgGA::GUIEventAdapter, 47  
 KEY\_KP\_1  
     osgGA::GUIEventAdapter, 47  
 KEY\_KP\_2  
     osgGA::GUIEventAdapter, 47  
 KEY\_KP\_3  
     osgGA::GUIEventAdapter, 47  
 KEY\_KP\_4  
     osgGA::GUIEventAdapter, 47  
 KEY\_KP\_5  
     osgGA::GUIEventAdapter, 47  
 KEY\_KP\_6  
     osgGA::GUIEventAdapter, 47  
 KEY\_KP\_7  
     osgGA::GUIEventAdapter, 47  
 KEY\_KP\_8  
     osgGA::GUIEventAdapter, 47  
 KEY\_KP\_9  
     osgGA::GUIEventAdapter, 47  
 KEY\_KP\_Add  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Begin  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Decimal  
     osgGA::GUIEventAdapter, 47  
 KEY\_KP\_Delete  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Divide  
     osgGA::GUIEventAdapter, 47  
 KEY\_KP\_Down  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_End  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Enter  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Equal  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_F1  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_F2  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_F3  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_F4  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Home  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Insert  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Left  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Multiply  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Next  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Page\_Down  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Page\_Up  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Prior  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Right  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Separator  
     osgGA::GUIEventAdapter, 47  
 KEY\_KP\_Space  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Subtract  
     osgGA::GUIEventAdapter, 47  
 KEY\_KP\_Tab  
     osgGA::GUIEventAdapter, 46  
 KEY\_KP\_Up  
     osgGA::GUIEventAdapter, 46  
 KEY\_Left  
     osgGA::GUIEventAdapter, 45  
 KEY\_Linefeed  
     osgGA::GUIEventAdapter, 45  
 KEY\_Menu  
     osgGA::GUIEventAdapter, 46  
 KEY\_Meta\_L  
     osgGA::GUIEventAdapter, 48  
 KEY\_Meta\_R  
     osgGA::GUIEventAdapter, 48  
 KEY\_Mode\_switch  
     osgGA::GUIEventAdapter, 46  
 KEY\_Next  
     osgGA::GUIEventAdapter, 46  
 KEY\_Num\_Lock  
     osgGA::GUIEventAdapter, 46  
 KEY\_Page\_Down  
     osgGA::GUIEventAdapter, 46  
 KEY\_Page\_Up  
     osgGA::GUIEventAdapter, 46  
 KEY\_Pause  
     osgGA::GUIEventAdapter, 45  
 KEY\_Print  
     osgGA::GUIEventAdapter, 46  
 KEY\_Prior  
     osgGA::GUIEventAdapter, 46  
 KEY\_Redo  
     osgGA::GUIEventAdapter, 46

- KEY\_Return
  - osgGA::GUIEventAdapter, 45
- KEY\_Right
  - osgGA::GUIEventAdapter, 46
- KEY\_Script\_switch
  - osgGA::GUIEventAdapter, 46
- KEY\_Scroll\_Lock
  - osgGA::GUIEventAdapter, 45
- KEY\_Select
  - osgGA::GUIEventAdapter, 46
- KEY\_Shift\_L
  - osgGA::GUIEventAdapter, 48
- KEY\_Shift\_Lock
  - osgGA::GUIEventAdapter, 48
- KEY\_Shift\_R
  - osgGA::GUIEventAdapter, 48
- KEY\_Space
  - osgGA::GUIEventAdapter, 45
- KEY\_Super\_L
  - osgGA::GUIEventAdapter, 48
- KEY\_Super\_R
  - osgGA::GUIEventAdapter, 48
- KEY\_Sys\_Req
  - osgGA::GUIEventAdapter, 45
- KEY\_Tab
  - osgGA::GUIEventAdapter, 45
- KEY\_Undo
  - osgGA::GUIEventAdapter, 46
- KEY\_Up
  - osgGA::GUIEventAdapter, 46
- KEYBOARD\_PITCH
  - DriveManipulator.cpp, 112
- KEYDOWN
  - osgGA::GUIEventAdapter, 45
- KeyManipMap
  - osgGA::KeySwitchMatrixManipulator, 59
- keyPress
  - osgGA::EventQueue, 26
- keyRelease
  - osgGA::EventQueue, 27
- KeySwitchMatrixManipulator, 125
  - OSGUTIL\_KEYSWITCMATRIXMANIPULATOR, 125
- KeySwitchMatrixManipulator.cpp, 126
- KeySymbol
  - osgGA::GUIEventAdapter, 45
- KEYUP
  - osgGA::GUIEventAdapter, 45
- L -**
- LEFT\_MOUSE\_BUTTON
  - osgGA::GUIEventAdapter, 48
- M -**
- M\_PI
  - UFOManipulator.cpp, 141
- mainpage.h, 127
- MAP
  - osgGA::SphericalManipulator, 78
- MatrixManipulator, 128
  - NEW\_HOME\_POSITION, 128
  - osgGA::MatrixManipulator, 64
  - OSGGA\_MatrixManipulator, 128
- MatrixManipulator.cpp, 129
- META\_NodeVisitor
  - osgGA::EventVisitor, 31
- META\_Object
  - osgGA::GUIEventAdapter, 51
  - osgGA::GUIEventHandler, 56
- MIDDLE\_MOUSE\_BUTTON
  - osgGA::GUIEventAdapter, 48
- MODKEY\_ALT
  - osgGA::GUIEventAdapter, 48
- MODKEY\_CAPS\_LOCK
  - osgGA::GUIEventAdapter, 48
- MODKEY\_CTRL
  - osgGA::GUIEventAdapter, 48
- MODKEY\_HYPER
  - osgGA::GUIEventAdapter, 48
- MODKEY\_LEFT\_ALT
  - osgGA::GUIEventAdapter, 48
- MODKEY\_LEFT\_CTRL
  - osgGA::GUIEventAdapter, 48
- MODKEY\_LEFT\_HYPER
  - osgGA::GUIEventAdapter, 48
- MODKEY\_LEFT\_META
  - osgGA::GUIEventAdapter, 48
- MODKEY\_LEFT\_SHIFT
  - osgGA::GUIEventAdapter, 48
- MODKEY\_LEFT\_SUPER
  - osgGA::GUIEventAdapter, 48
- MODKEY\_META
  - osgGA::GUIEventAdapter, 48
- MODKEY\_NUM\_LOCK
  - osgGA::GUIEventAdapter, 48
- MODKEY\_RIGHT\_ALT
  - osgGA::GUIEventAdapter, 48
- MODKEY\_RIGHT\_CTRL
  - osgGA::GUIEventAdapter, 48
- MODKEY\_RIGHT\_HYPER
  - osgGA::GUIEventAdapter, 48
- MODKEY\_RIGHT\_META
  - osgGA::GUIEventAdapter, 48
- MODKEY\_RIGHT\_SHIFT
  - osgGA::GUIEventAdapter, 48
- MODKEY\_RIGHT\_SUPER
  - osgGA::GUIEventAdapter, 48
- MODKEY\_SHIFT
  - osgGA::GUIEventAdapter, 48
- MODKEY\_SUPER
  - osgGA::GUIEventAdapter, 48
- ModKeyMask
  - osgGA::GUIEventAdapter, 48
- motion
  - osgGA::GUIEventAdapter::Scrolling, 74
- MouseButtonMask
  - osgGA::GUIEventAdapter, 48
- mouseButtonPress
  - osgGA::EventQueue, 27
- mouseButtonRelease
  - osgGA::EventQueue, 27
- mouseDoubleButtonPress
  - osgGA::EventQueue, 27
- mouseMotion
  - osgGA::EventQueue, 27
- mouseScroll
  - osgGA::EventQueue, 27
- mouseScroll2D
  - osgGA::EventQueue, 27
- mouseWarped
  - osgGA::EventQueue, 27
- MouseYOrientation

- osgGA::GUIEventAdapter, 48
- MOVE
  - osgGA::GUIEventAdapter, 45
- N -**
- NamedManipulator
  - osgGA::KeySwitchMatrixManipulator, 59
- NEW\_HOME\_POSITION
  - MatrixManipulator, 128
- NO\_AUTOMATIC\_YAW
  - osgGA::FlightManipulator, 35
- NODE\_CENTER
  - osgGA::NodeTrackerManipulator, 70
- NODE\_CENTER\_AND\_AZIM
  - osgGA::NodeTrackerManipulator, 70
- NODE\_CENTER\_AND\_ROTATION
  - osgGA::NodeTrackerManipulator, 70
- NodeTrackerManipulator, 130
  - osgGA::NodeTrackerManipulator, 70
  - OSGGA\_NODETRACKERMANIPULATOR, 130
- NodeTrackerManipulator.cpp, 131
  - TRACKBALLSIZE, 131
- NONE
  - osgGA::GUIEventAdapter, 45
- O -**
- ObserverNodePath
  - osgGA::NodeTrackerManipulator, 70
- operator()
  - osgGA::GUIEventHandler, 56
- operator=
  - osgGA::EventQueue, 28
  - osgGA::EventVisitor, 31
- osgGA, 7
- osgGA::AnimationPathManipulator, 9
  - \_animStartOfTimedPeriod, 12
  - \_animationPath, 12
  - \_isPaused, 12
  - \_matrix, 12
  - \_numOfFramesSinceStartOfTimedPeriod, 12
  - \_pauseTime, 12
  - \_printOutTimingInfo, 12
  - \_realStartOfTimedPeriod, 12
  - \_timeOffset, 12
  - \_timeScale, 12
  - \_valid, 12
  - AnimationPathManipulator, 11
  - className, 11
  - getAnimationPath, 11
  - getInverseMatrix, 11
  - getMatrix, 11
  - getPrintOutTimingInfo, 11
  - getUsage, 11
  - handle, 11
  - handleFrame, 11
  - home, 11
  - init, 11
  - setAnimationPath, 11
  - setByInverseMatrix, 11
  - setByMatrix, 11
  - setPrintOutTimingInfo, 11
  - valid, 12
- osgGA::CameraViewSwitchManipulator, 13
  - ~CameraViewSwitchManipulator, 14
  - \_cameraViews, 15
  - \_currentView, 15
  - \_node, 15
  - CameraViewList, 14
  - CameraViewSwitchManipulator, 14
  - className, 14
  - getInverseMatrix, 14
  - getMatrix, 14
  - getNode, 14
  - getUsage, 15
  - handle, 15
  - init, 15
  - setByInverseMatrix, 15
  - setByMatrix, 15
  - setNode, 15
- osgGA::DriveManipulator, 18
  - ~DriveManipulator, 20
  - \_buffer, 22
  - \_distance, 22
  - \_eye, 22
  - \_ga\_t0, 22
  - \_ga\_t1, 22
  - \_height, 22
  - \_modelScale, 22
  - \_node, 22
  - \_pitch, 22
  - \_pitchDownKeyPressed, 22
  - \_pitchUpKeyPressed, 22
  - \_rotation, 22
  - \_speedMode, 22
  - \_velocity, 22
  - addMouseEvent, 20
  - calcMovement, 20
  - className, 20
  - computeHomePosition, 20
  - computePosition, 20
  - DriveManipulator, 20
  - flushMouseEventStack, 20
  - getHeight, 20
  - getInverseMatrix, 21
  - getMatrix, 21
  - getModelScale, 21
  - getNode, 21
  - getUsage, 21
  - getVelocity, 21
  - handle, 21
  - home, 21
  - init, 21
  - intersect, 21
  - setByInverseMatrix, 21
  - setByMatrix, 21
  - setHeight, 21
  - setModelScale, 22
  - setNode, 22
  - setVelocity, 22
  - SpeedControlMode, 20
  - USE\_MOUSE\_BUTTONS\_FOR\_SPEED, 20
  - USE\_MOUSE\_Y\_FOR\_SPEED, 20
- osgGA::EventQueue, 23
  - ~EventQueue, 26
  - \_accumulateEventState, 29
  - \_eventQueue, 29
  - \_eventQueueMutex, 29
  - \_startTick, 29
  - \_useFixedMouseInputRange, 29
  - addEvent, 26
  - appendEvents, 26

- closeWindow, 26
- copyEvents, 26
- createEvent, 26
- EventQueue, 26
- Events, 26
- frame, 26
- getCurrentEventState, 26
- getStartTick, 26
- getTime, 26
- getUseFixedMouseInputRange, 26
- keyPress, 26
- keyRelease, 27
- mouseButtonPress, 27
- mouseButtonRelease, 27
- mouseDoubleClickPress, 27
- mouseMotion, 27
- mouseScroll, 27
- mouseScroll2D, 27
- mouseWarped, 27
- operator=, 28
- penOrientation, 28
- penPressure, 28
- penProximity, 28
- quitApplication, 28
- setCurrentEventState, 28
- setEvents, 28
- setGraphicsContext, 28
- setMouseInputRange, 28
- setStartTick, 28
- setUseFixedMouseInputRange, 28
- takeEvents, 28
- userEvent, 28
- windowResize, 28, 29
- osgGA::EventVisitor, 30
  - ~EventVisitor, 31
  - \_accumulateEventState, 32
  - \_actionAdapter, 32
  - \_events, 32
  - \_handled, 32
  - addEvent, 31
  - apply, 31
  - EventList, 31
  - EventVisitor, 31
  - getActionAdapter, 31
  - getEventHandled, 31
  - getEvents, 31
  - handle\_callbacks, 31
  - handle\_callbacks\_and\_traverse, 31
  - handle\_geode\_callbacks, 31
  - META\_NodeVisitor, 31
  - operator=, 31
  - removeEvent, 31
  - reset, 32
  - setEventHandled, 32
  - setEvents, 32
  - traverseGeode, 32
- osgGA::FlightManipulator, 33
  - ~FlightManipulator, 35
  - \_acceleration, 37
  - \_distance, 37
  - \_eye, 37
  - \_ga\_t0, 37
  - \_ga\_t1, 37
  - \_modelScale, 37
  - \_node, 37
  - \_rotation, 37
  - \_velocity, 37
  - \_yawMode, 37
- addMouseEvent, 35
- calcMovement, 35
- className, 35
- computePosition, 35
- FlightManipulator, 35
- flushMouseEventStack, 35
- getAcceleration, 35
- getInverseMatrix, 35
- getMatrix, 35
- getModelScale, 35
- getNode, 35
- getUsage, 36
- getVelocity, 36
- handle, 36
- home, 36
- init, 36
- NO\_AUTOMATIC\_YAW, 35
- setAcceleration, 36
- setByInverseMatrix, 36
- setByMatrix, 36
- setModelScale, 36
- setNode, 36
- setVelocity, 36
- setYawControlMode, 36
- YAW\_AUTOMATICALLY\_WHEN\_BANKED, 35
- YawControlMode, 35
- osgGA::GUIActionAdapter, 38
  - ~GUIActionAdapter, 38
  - asView, 38
  - requestContinuousUpdate, 38
  - requestRedraw, 39
  - requestWarpPointer, 39
- osgGA::GUIEventAdapter, 40
  - ~GUIEventAdapter, 49
  - \_Xmax, 52
  - \_Xmin, 52
  - \_Ymax, 52
  - \_Ymin, 52
  - \_button, 52
  - \_buttonMask, 52
  - \_context, 52
  - \_eventType, 52
  - \_handled, 52
  - \_key, 52
  - \_modKeyMask, 52
  - \_mouseYOrientation, 52
  - \_mx, 52
  - \_my, 52
  - \_scrolling, 52
  - \_tabletPen, 52
  - \_time, 52
  - \_windowHeight, 52
  - \_windowWidth, 52
  - \_windowX, 52
  - \_windowY, 52
- CLOSE\_WINDOW, 45
- DOUBLECLICK, 45
- DRAG, 45
- ERASER, 49
- EventType, 45
- FRAME, 45
- getAccumulatedEventState, 49
- getButton, 49
- getButtonMask, 49

getEventType, 49  
getGraphicsContext, 49  
getHandled, 49  
getKey, 49  
getModKeyMask, 49  
getMouseYOrientation, 50  
getPenOrientation, 50  
getPenPressure, 50  
getPenRotation, 50  
getPenTiltX, 50  
getPenTiltY, 50  
getScrollingDeltaX, 50  
getScrollingDeltaY, 50  
getScrollingMotion, 50  
getTabletPointerType, 50  
getTime, 50  
getWindowHeight, 50  
getWindowWidth, 50  
getWindowX, 50  
getWindowY, 50  
getX, 50  
getXmax, 50  
getXmin, 50  
getXnormalized, 50  
getY, 51  
getYmax, 51  
getYmin, 51  
getYnormalized, 51  
GUIEventAdapter, 49  
KEY\_Alt\_L, 48  
KEY\_Alt\_R, 48  
KEY\_BackSpace, 45  
KEY\_Begin, 46  
KEY\_Break, 46  
KEY\_Cancel, 46  
KEY\_Caps\_Lock, 48  
KEY\_Clear, 45  
KEY\_Control\_L, 48  
KEY\_Control\_R, 48  
KEY\_Delete, 45  
KEY\_Down, 46  
KEY\_End, 46  
KEY\_Escape, 45  
KEY\_Execute, 46  
KEY\_F1, 47  
KEY\_F10, 47  
KEY\_F11, 47  
KEY\_F12, 47  
KEY\_F13, 47  
KEY\_F14, 47  
KEY\_F15, 47  
KEY\_F16, 47  
KEY\_F17, 47  
KEY\_F18, 47  
KEY\_F19, 47  
KEY\_F2, 47  
KEY\_F20, 47  
KEY\_F21, 47  
KEY\_F22, 47  
KEY\_F23, 47  
KEY\_F24, 47  
KEY\_F25, 47  
KEY\_F26, 47  
KEY\_F27, 47  
KEY\_F28, 47  
KEY\_F29, 47  
KEY\_F3, 47  
KEY\_F30, 47  
KEY\_F31, 47  
KEY\_F32, 47  
KEY\_F33, 48  
KEY\_F34, 48  
KEY\_F35, 48  
KEY\_F4, 47  
KEY\_F5, 47  
KEY\_F6, 47  
KEY\_F7, 47  
KEY\_F8, 47  
KEY\_F9, 47  
KEY\_Find, 46  
KEY\_Help, 46  
KEY\_Home, 45  
KEY\_Hyper\_L, 48  
KEY\_Hyper\_R, 48  
KEY\_Insert, 46  
KEY\_KP\_0, 47  
KEY\_KP\_1, 47  
KEY\_KP\_2, 47  
KEY\_KP\_3, 47  
KEY\_KP\_4, 47  
KEY\_KP\_5, 47  
KEY\_KP\_6, 47  
KEY\_KP\_7, 47  
KEY\_KP\_8, 47  
KEY\_KP\_9, 47  
KEY\_KP\_Add, 46  
KEY\_KP\_Begin, 46  
KEY\_KP\_Decimal, 47  
KEY\_KP\_Delete, 46  
KEY\_KP\_Divide, 47  
KEY\_KP\_Down, 46  
KEY\_KP\_End, 46  
KEY\_KP\_Enter, 46  
KEY\_KP\_Equal, 46  
KEY\_KP\_F1, 46  
KEY\_KP\_F2, 46  
KEY\_KP\_F3, 46  
KEY\_KP\_F4, 46  
KEY\_KP\_Home, 46  
KEY\_KP\_Insert, 46  
KEY\_KP\_Left, 46  
KEY\_KP\_Multiply, 46  
KEY\_KP\_Next, 46  
KEY\_KP\_Page\_Down, 46  
KEY\_KP\_Page\_Up, 46  
KEY\_KP\_Prior, 46  
KEY\_KP\_Right, 46  
KEY\_KP\_Separator, 47  
KEY\_KP\_Space, 46  
KEY\_KP\_Subtract, 47  
KEY\_KP\_Tab, 46  
KEY\_KP\_Up, 46  
KEY\_Left, 45  
KEY\_Linefeed, 45  
KEY\_Menu, 46  
KEY\_Meta\_L, 48  
KEY\_Meta\_R, 48  
KEY\_Mode\_switch, 46  
KEY\_Next, 46  
KEY\_Num\_Lock, 46  
KEY\_Page\_Down, 46  
KEY\_Page\_Up, 46

KEY\_Pause, 45  
 KEY\_Print, 46  
 KEY\_Prior, 46  
 KEY\_Redo, 46  
 KEY\_Return, 45  
 KEY\_Right, 46  
 KEY\_Script\_switch, 46  
 KEY\_Scroll\_Lock, 45  
 KEY\_Select, 46  
 KEY\_Shift\_L, 48  
 KEY\_Shift\_Lock, 48  
 KEY\_Shift\_R, 48  
 KEY\_Space, 45  
 KEY\_Super\_L, 48  
 KEY\_Super\_R, 48  
 KEY\_Sys\_Req, 45  
 KEY\_Tab, 45  
 KEY\_Undo, 46  
 KEY\_Up, 46  
 KEYDOWN, 45  
 KeySymbol, 45  
 KEYUP, 45  
 LEFT\_MOUSE\_BUTTON, 48  
 META\_Object, 51  
 MIDDLE\_MOUSE\_BUTTON, 48  
 MODKEY\_ALT, 48  
 MODKEY\_CAPS\_LOCK, 48  
 MODKEY\_CTRL, 48  
 MODKEY\_HYPER, 48  
 MODKEY\_LEFT\_ALT, 48  
 MODKEY\_LEFT\_CTRL, 48  
 MODKEY\_LEFT\_HYPER, 48  
 MODKEY\_LEFT\_META, 48  
 MODKEY\_LEFT\_SHIFT, 48  
 MODKEY\_LEFT\_SUPER, 48  
 MODKEY\_META, 48  
 MODKEY\_NUM\_LOCK, 48  
 MODKEY\_RIGHT\_ALT, 48  
 MODKEY\_RIGHT\_CTRL, 48  
 MODKEY\_RIGHT\_HYPER, 48  
 MODKEY\_RIGHT\_META, 48  
 MODKEY\_RIGHT\_SHIFT, 48  
 MODKEY\_RIGHT\_SUPER, 48  
 MODKEY\_SHIFT, 48  
 MODKEY\_SUPER, 48  
 ModKeyMask, 48  
 MouseButtonMask, 48  
 MouseYOrientation, 48  
 MOVE, 45  
 NONE, 45  
 PEN, 49  
 PEN\_ORIENTATION, 45  
 PEN\_PRESSURE, 45  
 PEN\_PROXIMITY\_ENTER, 45  
 PEN\_PROXIMITY\_LEAVE, 45  
 PUCK, 49  
 PUSH, 45  
 QUIT\_APPLICATION, 45  
 RELEASE, 45  
 RESIZE, 45  
 RIGHT\_MOUSE\_BUTTON, 48  
 SCROLL, 45  
 SCROLL\_2D, 49  
 SCROLL\_DOWN, 49  
 SCROLL\_LEFT, 49  
 SCROLL\_NONE, 49  
 SCROLL\_RIGHT, 49  
 SCROLL\_UP, 49  
 ScrollingMotion, 49  
 setButton, 51  
 setButtonMask, 51  
 setEventType, 51  
 setGraphicsContext, 51  
 setHandled, 51  
 setInputRange, 51  
 setKey, 51  
 setModKeyMask, 51  
 setMouseYOrientation, 51  
 setPenPressure, 51  
 setPenRotation, 51  
 setPenTiltX, 51  
 setPenTiltY, 51  
 setScrollingMotion, 51  
 setScrollingMotionDelta, 52  
 setTabletPointerType, 52  
 setTime, 52  
 setWindowRectangle, 52  
 setX, 52  
 setY, 52  
 TabletPointerType, 49  
 time, 52  
 UNKNOWN, 49  
 USER, 45  
 Y\_INCREASING\_DOWNWARDS, 49  
 Y\_INCREASING\_UPWARDS, 49  
 osgGA::GUIEventAdapter::Scrolling, 74  
   deltaX, 74  
   deltaY, 74  
   motion, 74  
   Scrolling, 74  
 osgGA::GUIEventAdapter::TabletPen, 86  
   pressure, 86  
   rotation, 86  
   TabletPen, 86  
   tabletPointerType, 86  
   tiltX, 86  
   tiltY, 86  
 osgGA::GUIEventHandler, 54  
   \_ignoreHandledEventsMask, 56  
   event, 55  
   getIgnoreHandledEventsMask, 55  
   getUsage, 55  
   GUIEventHandler, 55  
   handle, 55  
   handleWithCheckAgainstIgnoreHandledEventsMask, 55  
   META\_Object, 56  
   operator(), 56  
   setIgnoreHandledEventsMask, 56  
 osgGA::KeySwitchMatrixManipulator, 57  
   addMatrixManipulator, 59  
   addNumberedMatrixManipulator, 59  
   className, 59  
   computeHomePosition, 59  
   getCurrentMatrixManipulator, 59  
   getFusionDistanceMode, 59  
   getFusionDistanceValue, 60  
   getInverseMatrix, 60  
   getKeyManipMap, 60  
   getMatrix, 60  
   getMatrixManipulatorWithIndex, 60  
   getMatrixManipulatorWithKey, 60  
   getNode, 60

- getNumMatrixManipulators, 60
- getUsage, 60
- handle, 60
- home, 60
- init, 61
- KeyManipMap, 59
- NamedManipulator, 59
- selectMatrixManipulator, 61
- setAutoComputeHomePosition, 61
- setByInverseMatrix, 61
- setByMatrix, 61
- setCoordinateFrameCallback, 61
- setHomePosition, 61
- setMinimumDistance, 61
- setNode, 61
- osgGA::MatrixManipulator, 62
  - ~MatrixManipulator, 64
  - \_autoComputeHomePosition, 67
  - \_coordinateFrameCallback, 67
  - \_homeCenter, 67
  - \_homeEye, 67
  - \_homeUp, 67
  - \_intersectTraversalMask, 67
  - \_minimumDistance, 67
  - className, 64
  - computeHomePosition, 64
  - getAutoComputeHomePosition, 64
  - getCoordinateFrame, 64
  - getCoordinateFrameCallback, 64
  - getFrontVector, 64
  - getFusionDistanceMode, 64
  - getFusionDistanceValue, 64
  - getHomePosition, 65
  - getIntersectTraversalMask, 65
  - getInverseMatrix, 65
  - getMatrix, 65
  - getMinimumDistance, 65
  - getNode, 65
  - getSideVector, 65
  - getUpVector, 65
  - handle, 65
  - home, 65, 66
  - init, 66
  - MatrixManipulator, 64
  - setAutoComputeHomePosition, 66
  - setByInverseMatrix, 66
  - setByMatrix, 66
  - setCoordinateFrameCallback, 66
  - setHomePosition, 66
  - setIntersectTraversalMask, 66
  - setMinimumDistance, 66
  - setNode, 67
- osgGA::MatrixManipulator::CoordinateFrameCallback, 17
  - ~CoordinateFrameCallback, 17
  - getCoordinateFrame, 17
- osgGA::NodeTrackerManipulator, 68
  - ~NodeTrackerManipulator, 70
  - \_distance, 73
  - \_ga\_t0, 73
  - \_ga\_t1, 73
  - \_node, 73
  - \_nodeRotation, 73
  - \_rotation, 73
  - \_rotationMode, 73
  - \_thrown, 73
  - \_trackNodePath, 73
  - \_trackerMode, 73
  - addMouseEvent, 70
  - calcMovement, 70
  - clampOrientation, 71
  - className, 71
  - computeHomePosition, 71
  - computeNodeCenterAndRotation, 71
  - computeNodeLocalToWorld, 71
  - computeNodeWorldToLocal, 71
  - computePosition, 71
  - ELEVATION\_AZIM, 70
  - flushMouseEventStack, 71
  - getFusionDistanceMode, 71
  - getFusionDistanceValue, 71
  - getInverseMatrix, 71
  - getMatrix, 71
  - getNode, 71
  - getNodePath, 71
  - getRotationMode, 72
  - getTrackerMode, 72
  - getTrackNode, 72
  - getTrackNodePath, 72
  - getUsage, 72
  - handle, 72
  - home, 72
  - init, 72
  - isMouseMoving, 72
  - NODE\_CENTER, 70
  - NODE\_CENTER\_AND\_AZIM, 70
  - NODE\_CENTER\_AND\_ROTATION, 70
  - NodeTrackerManipulator, 70
  - ObserverNodePath, 70
  - RotationMode, 70
  - setByInverseMatrix, 72
  - setByMatrix, 72
  - setNode, 72
  - setRotationMode, 72
  - setTrackerMode, 73
  - setTrackNode, 73
  - setTrackNodePath, 73
  - tb\_project\_to\_sphere, 73
  - TRACKBALL, 70
  - trackball, 73
  - TrackerMode, 70
  - validateNodePath, 73
- osgGA::SphericalManipulator, 75
  - ~SphericalManipulator, 78
  - \_allowThrow, 80
  - \_center, 80
  - \_delta\_frame\_time, 80
  - \_distance, 80
  - \_elevation, 80
  - \_ga\_t0, 80
  - \_ga\_t1, 80
  - \_heading, 80
  - \_homeDistance, 80
  - \_last\_frame\_time, 80
  - \_minimumZoomScale, 80
  - \_modelScale, 81
  - \_node, 81
  - \_rotationMode, 81
  - \_thrown, 81
  - \_zoomDelta, 81
  - addMouseEvent, 78
  - calcMovement, 78
  - className, 78

- computeHomePosition, 78
- computeViewPosition, 78
- ELEVATION, 78
- ELEVATION\_HEADING, 78
- flushMouseEventStack, 78
- getAllowThrow, 78
- getCenter, 78
- getDistance, 78
- getElevation, 78
- getFusionDistanceMode, 78
- getFusionDistanceValue, 78
- getHeading, 78
- getHomeDistance, 79
- getInverseMatrix, 79
- getMatrix, 79
- getMinimumZoomScale, 79
- getNode, 79
- getRotationMode, 79
- getScrollWheelZoomDelta, 79
- getUsage, 79
- handle, 79
- HEADING, 78
- home, 79
- init, 79
- isMouseMoving, 79
- MAP, 78
- RotationMode, 78
- setAllowThrow, 79
- setByInverseMatrix, 80
- setByMatrix, 80
- setCenter, 80
- setDistance, 80
- setElevation, 80
- setHeading, 80
- setMinimumZoomScale, 80
- setNode, 80
- setRotationMode, 80
- setScrollWheelZoomDelta, 80
- SphericalManipulator, 78
- zoomOn, 80
- osgGA::StateSetManipulator, 82
  - ~StateSetManipulator, 84
  - \_backface, 85
  - \_initialized, 85
  - \_keyEventCyclePolygonMode, 85
  - \_keyEventToggleBackfaceCulling, 85
  - \_keyEventToggleLighting, 85
  - \_keyEventToggleTexturing, 85
  - \_lighting, 85
  - \_maxNumOfTextureUnits, 85
  - \_stateset, 85
  - \_texture, 85
  - className, 84
  - clone, 84
  - cyclePolygonMode, 84
  - getBackfaceEnabled, 84
  - getKeyEventCyclePolygonMode, 84
  - getKeyEventToggleBackfaceCulling, 84
  - getKeyEventToggleLighting, 84
  - getKeyEventToggleTexturing, 84
  - getLightingEnabled, 84
  - getMaximumNumOfTextureUnits, 84
  - getOrCreatePolygonMode, 84
  - getPolygonMode, 84
  - getStateSet, 84
  - getTextureEnabled, 84
  - getUsage, 84
  - handle, 84
  - setBackfaceEnabled, 84
  - setKeyEventCyclePolygonMode, 84
  - setKeyEventToggleBackfaceCulling, 84
  - setKeyEventToggleLighting, 84
  - setKeyEventToggleTexturing, 84
  - setLightingEnabled, 84
  - setMaximumNumOfTextureUnits, 84
  - setPolygonMode, 84
  - setStateSet, 84
  - setTextureEnabled, 84
  - StateSetManipulator, 84
- osgGA::TerrainManipulator, 87
  - ~TerrainManipulator, 89
  - \_center, 91
  - \_distance, 91
  - \_ga\_t0, 91
  - \_ga\_t1, 91
  - \_node, 91
  - \_previousUp, 91
  - \_rotation, 91
  - \_rotationMode, 91
  - \_thrown, 91
  - addMouseEvent, 89
  - calcMovement, 89
  - clampOrientation, 89
  - className, 89
  - computePosition, 89
  - ELEVATION\_HEADING, 89
  - ELEVATION\_HEADING\_ROLL, 89
  - flushMouseEventStack, 89
  - getFusionDistanceMode, 89
  - getFusionDistanceValue, 89
  - getInverseMatrix, 89
  - getMatrix, 89
  - getNode, 90
  - getRotationMode, 90
  - getUsage, 90
  - handle, 90
  - home, 90
  - init, 90
  - intersect, 90
  - isMouseMoving, 90
  - RotationMode, 89
  - setByInverseMatrix, 90
  - setByMatrix, 90
  - setNode, 90
  - setRotationMode, 90
  - tb\_project\_to\_sphere, 91
  - TerrainManipulator, 89
  - trackball, 91
- osgGA::TrackballManipulator, 92
  - ~TrackballManipulator, 95
  - \_allowThrow, 97
  - \_center, 97
  - \_delta\_frame\_time, 97
  - \_distance, 97
  - \_ga\_t0, 97
  - \_ga\_t1, 97
  - \_last\_frame\_time, 97
  - \_minimumZoomScale, 97
  - \_modelScale, 97
  - \_node, 97
  - \_rotation, 97
  - \_thrown, 97

- \_trackballSize, 97
- \_zoomDelta, 97
- addMouseEvent, 95
- calcMovement, 95
- className, 95
- computePosition, 95
- flushMouseEventStack, 95
- getAllowThrow, 95
- getCenter, 95
- getDistance, 95
- getFusionDistanceMode, 95
- getFusionDistanceValue, 95
- getInverseMatrix, 95
- getMatrix, 95
- getMinimumZoomScale, 95
- getNode, 95
- getRotation, 96
- getScrollWheelZoomDelta, 96
- getTrackballSize, 96
- getUsage, 96
- handle, 96
- home, 96
- init, 96
- isMouseMoving, 96
- setAllowThrow, 96
- setByInverseMatrix, 96
- setByMatrix, 96
- setCenter, 96
- setDistance, 96
- setMinimumZoomScale, 97
- setNode, 97
- setRotation, 97
- setScrollWheelZoomDelta, 97
- setTrackballSize, 97
- tb\_project\_to\_sphere, 97
- trackball, 97
- TrackballManipulator, 95
- osgGA::UFOManipulator, 99
  - ~UFOManipulator, 102
  - \_adjustPosition, 102
  - \_ctrl, 105
  - \_decelerateOffsetRate, 105
  - \_decelerateUpSideRate, 105
  - \_direction, 105
  - \_directionRotationAcceleration, 105
  - \_directionRotationDeceleration, 105
  - \_directionRotationEpsilon, 105
  - \_directionRotationRate, 105
  - \_dt, 105
  - \_forwardSpeed, 105
  - \_frame, 102
  - \_inverseMatrix, 105
  - \_keyDown, 102
  - \_keyUp, 102
  - \_matrix, 105
  - \_minDistanceInFront, 105
  - \_minHeightAboveGround, 105
  - \_node, 105
  - \_offset, 105
  - \_pitchOffset, 105
  - \_pitchOffsetRate, 105
  - \_position, 105
  - \_shift, 105
  - \_sideSpeed, 105
  - \_speedAccelerationFactor, 105
  - \_speedDecelerationFactor, 105
  - \_speedEpsilon, 105
  - \_stop, 102
  - \_straightenOffset, 105
  - \_t0, 105
  - \_upSpeed, 105
  - \_viewAngle, 105
  - \_viewOffsetDelta, 105
  - \_yawOffset, 105
  - \_yawOffsetRate, 105
  - className, 102
  - computeHomePosition, 102
  - getCurrentPositionAsLookAt, 102
  - getForwardSpeed, 102
  - getInverseMatrix, 102
  - getMatrix, 102
  - getMinDistance, 102
  - getMinHeight, 102
  - getNode, 102
  - getRotationSpeed, 103
  - getSideSpeed, 103
  - getUsage, 103
  - handle, 103
  - home, 103
  - init, 103
  - intersect, 103
  - setByInverseMatrix, 103
  - setByMatrix, 104
  - setForwardSpeed, 104
  - setMinDistance, 104
  - setMinHeight, 104
  - setNode, 104
  - setRotationSpeed, 104
  - setSideSpeed, 105
  - UFOManipulator, 102
- OSGGA\_ANIMATION\_PATH\_MANIPULATOR
  - AnimationPathManipulator, 107
- OSGGA\_DRIVEMANIPULATOR
  - DriveManipulator, 111
- OSGGA\_EVENT
  - GUIEventAdapter, 121
- OSGGA\_EVENTQUEUE
  - EventQueue, 113
- OSGGA\_EVENTVISITOR
  - EventVisitor, 115
- OSGGA\_EXPORT
  - Export, 117
- OSGGA\_EXPORT\_
  - Export, 117
- OSGGA\_FLIGHTMANIPULATOR
  - FlightManipulator, 118
- OSGGA\_GUIACTIONADAPTER
  - GUIActionAdapter, 120
- OSGGA\_GUIEVENTHANDLER
  - GUIEventHandler, 123
- OSGGA\_MatrixManipulator
  - MatrixManipulator, 128
- OSGGA\_NODETRACKERMANIPULATOR
  - NodeTrackerManipulator, 130
- OSGGA\_STATESET\_MANIPULATOR
  - StateSetManipulator, 134
- OSGGA\_TERRAINMANIPULATOR
  - TerrainManipulator, 136
- OSGGA\_TRACKBALLMANIPULATOR
  - TrackballManipulator, 138
- OSGGA\_UFO\_MANIPULATOR\_DEF
  - UFOManipulator, 140

- OSGGA\_VERSION
  - Version, 142
- OSGGA\_VIEWLISTMANIPULATOR
  - CameraViewSwitchManipulator, 109
- osgGAGetLibraryName
  - Version, 142
  - Version.cpp, 143
- osgGAGetVersion
  - Version, 142
  - Version.cpp, 143
- OSGUTIL\_KEYSWITCHMATRIXMANIPULATOR
  - KeySwitchMatrixManipulator, 125
- P -**
- PEN
  - osgGA::GUIEventAdapter, 49
- PEN\_ORIENTATION
  - osgGA::GUIEventAdapter, 45
- PEN\_PRESSURE
  - osgGA::GUIEventAdapter, 45
- PEN\_PROXIMITY\_ENTER
  - osgGA::GUIEventAdapter, 45
- PEN\_PROXIMITY\_LEAVE
  - osgGA::GUIEventAdapter, 45
- penOrientation
  - osgGA::EventQueue, 28
- penPressure
  - osgGA::EventQueue, 28
- penProximity
  - osgGA::EventQueue, 28
- pressure
  - osgGA::GUIEventAdapter::TabletPen, 86
- PUCK
  - osgGA::GUIEventAdapter, 49
- PUSH
  - osgGA::GUIEventAdapter, 45
- Q -**
- QUIT\_APPLICATION
  - osgGA::GUIEventAdapter, 45
- quitApplication
  - osgGA::EventQueue, 28
- R -**
- RELEASE
  - osgGA::GUIEventAdapter, 45
- removeEvent
  - osgGA::EventVisitor, 31
- requestContinuousUpdate
  - osgGA::GUIActionAdapter, 38
- requestRedraw
  - osgGA::GUIActionAdapter, 39
- requestWarpPointer
  - osgGA::GUIActionAdapter, 39
- reset
  - osgGA::EventVisitor, 32
- RESIZE
  - osgGA::GUIEventAdapter, 45
- RIGHT\_MOUSE\_BUTTON
  - osgGA::GUIEventAdapter, 48
- rotation
  - osgGA::GUIEventAdapter::TabletPen, 86
- RotationMode
  - osgGA::NodeTrackerManipulator, 70
  - osgGA::SphericalManipulator, 78
- osgGA::TerrainManipulator, 89
- S -**
- SCROLL
  - osgGA::GUIEventAdapter, 45
- SCROLL\_2D
  - osgGA::GUIEventAdapter, 49
- SCROLL\_DOWN
  - osgGA::GUIEventAdapter, 49
- SCROLL\_LEFT
  - osgGA::GUIEventAdapter, 49
- SCROLL\_NONE
  - osgGA::GUIEventAdapter, 49
- SCROLL\_RIGHT
  - osgGA::GUIEventAdapter, 49
- SCROLL\_UP
  - osgGA::GUIEventAdapter, 49
- Scrolling
  - osgGA::GUIEventAdapter::Scrolling, 74
- ScrollingMotion
  - osgGA::GUIEventAdapter, 49
- selectMatrixManipulator
  - osgGA::KeySwitchMatrixManipulator, 61
- setAcceleration
  - osgGA::FlightManipulator, 36
- setAllowThrow
  - osgGA::SphericalManipulator, 79
  - osgGA::TrackballManipulator, 96
- setAnimationPath
  - osgGA::AnimationPathManipulator, 11
- setAutoComputeHomePosition
  - osgGA::KeySwitchMatrixManipulator, 61
  - osgGA::MatrixManipulator, 66
- setBackfaceEnabled
  - osgGA::StateSetManipulator, 84
- setButton
  - osgGA::GUIEventAdapter, 51
- setButtonMask
  - osgGA::GUIEventAdapter, 51
- setByInverseMatrix
  - osgGA::AnimationPathManipulator, 11
  - osgGA::CameraViewSwitchManipulator, 15
  - osgGA::DriveManipulator, 21
  - osgGA::FlightManipulator, 36
  - osgGA::KeySwitchMatrixManipulator, 61
  - osgGA::MatrixManipulator, 66
  - osgGA::NodeTrackerManipulator, 72
  - osgGA::SphericalManipulator, 80
  - osgGA::TerrainManipulator, 90
  - osgGA::TrackballManipulator, 96
  - osgGA::UFOManipulator, 103
- setByMatrix
  - osgGA::AnimationPathManipulator, 11
  - osgGA::CameraViewSwitchManipulator, 15
  - osgGA::DriveManipulator, 21
  - osgGA::FlightManipulator, 36
  - osgGA::KeySwitchMatrixManipulator, 61
  - osgGA::MatrixManipulator, 66
  - osgGA::NodeTrackerManipulator, 72
  - osgGA::SphericalManipulator, 80
  - osgGA::TerrainManipulator, 90
  - osgGA::TrackballManipulator, 96
  - osgGA::UFOManipulator, 104
- setCenter
  - osgGA::SphericalManipulator, 80
  - osgGA::TrackballManipulator, 96

- setCoordinateFrameCallback
  - osgGA::KeySwitchMatrixManipulator, 61
  - osgGA::MatrixManipulator, 66
- setCurrentEventState
  - osgGA::EventQueue, 28
- setDistance
  - osgGA::SphericalManipulator, 80
  - osgGA::TrackballManipulator, 96
- setElevation
  - osgGA::SphericalManipulator, 80
- setEventHandled
  - osgGA::EventVisitor, 32
- setEvents
  - osgGA::EventQueue, 28
  - osgGA::EventVisitor, 32
- setEventType
  - osgGA::GUIEventAdapter, 51
- setForwardSpeed
  - osgGA::UFOManipulator, 104
- setGraphicsContext
  - osgGA::EventQueue, 28
  - osgGA::GUIEventAdapter, 51
- setHandled
  - osgGA::GUIEventAdapter, 51
- setHeading
  - osgGA::SphericalManipulator, 80
- setHeight
  - osgGA::DriveManipulator, 21
- setHomePosition
  - osgGA::KeySwitchMatrixManipulator, 61
  - osgGA::MatrixManipulator, 66
- setIgnoreHandledEventsMask
  - osgGA::GUIEventHandler, 56
- setInputRange
  - osgGA::GUIEventAdapter, 51
- setIntersectTraversalMask
  - osgGA::MatrixManipulator, 66
- setKey
  - osgGA::GUIEventAdapter, 51
- setKeyEventCyclePolygonMode
  - osgGA::StateSetManipulator, 84
- setKeyEventToggleBackfaceCulling
  - osgGA::StateSetManipulator, 84
- setKeyEventToggleLighting
  - osgGA::StateSetManipulator, 84
- setKeyEventToggleTexturing
  - osgGA::StateSetManipulator, 84
- setLightingEnabled
  - osgGA::StateSetManipulator, 84
- setMaximumNumOfTextureUnits
  - osgGA::StateSetManipulator, 84
- setMinDistance
  - osgGA::UFOManipulator, 104
- setMinHeight
  - osgGA::UFOManipulator, 104
- setMinimumDistance
  - osgGA::KeySwitchMatrixManipulator, 61
  - osgGA::MatrixManipulator, 66
- setMinimumZoomScale
  - osgGA::SphericalManipulator, 80
  - osgGA::TrackballManipulator, 97
- setModelScale
  - osgGA::DriveManipulator, 22
  - osgGA::FlightManipulator, 36
- setModKeyMask
  - osgGA::GUIEventAdapter, 51
- setMouseInputRange
  - osgGA::EventQueue, 28
- setMouseYOrientation
  - osgGA::GUIEventAdapter, 51
- setNode
  - osgGA::CameraViewSwitchManipulator, 15
  - osgGA::DriveManipulator, 22
  - osgGA::FlightManipulator, 36
  - osgGA::KeySwitchMatrixManipulator, 61
  - osgGA::MatrixManipulator, 67
  - osgGA::NodeTrackerManipulator, 72
  - osgGA::SphericalManipulator, 80
  - osgGA::TerrainManipulator, 90
  - osgGA::TrackballManipulator, 97
  - osgGA::UFOManipulator, 104
- setPenPressure
  - osgGA::GUIEventAdapter, 51
- setPenRotation
  - osgGA::GUIEventAdapter, 51
- setPenTiltX
  - osgGA::GUIEventAdapter, 51
- setPenTiltY
  - osgGA::GUIEventAdapter, 51
- setPolygonMode
  - osgGA::StateSetManipulator, 84
- setPrintOutTimingInfo
  - osgGA::AnimationPathManipulator, 11
- setRotation
  - osgGA::TrackballManipulator, 97
- setRotationMode
  - osgGA::NodeTrackerManipulator, 72
  - osgGA::SphericalManipulator, 80
  - osgGA::TerrainManipulator, 90
- setRotationSpeed
  - osgGA::UFOManipulator, 104
- setScrollingMotion
  - osgGA::GUIEventAdapter, 51
- setScrollingMotionDelta
  - osgGA::GUIEventAdapter, 52
- setScrollWheelZoomDelta
  - osgGA::SphericalManipulator, 80
  - osgGA::TrackballManipulator, 97
- setSideSpeed
  - osgGA::UFOManipulator, 105
- setStartTick
  - osgGA::EventQueue, 28
- setStateSet
  - osgGA::StateSetManipulator, 84
- setTabletPointerType
  - osgGA::GUIEventAdapter, 52
- setTextureEnabled
  - osgGA::StateSetManipulator, 84
- setTime
  - osgGA::GUIEventAdapter, 52
- setTrackballSize
  - osgGA::TrackballManipulator, 97
- setTrackerMode
  - osgGA::NodeTrackerManipulator, 73
- setTrackNode
  - osgGA::NodeTrackerManipulator, 73
- setTrackNodePath
  - osgGA::NodeTrackerManipulator, 73
- setUseFixedMouseInputRange
  - osgGA::EventQueue, 28
- setVelocity
  - osgGA::DriveManipulator, 22

- osgGA::FlightManipulator, 36
- setWindowRectangle
  - osgGA::GUIEventAdapter, 52
- setX
  - osgGA::GUIEventAdapter, 52
- setY
  - osgGA::GUIEventAdapter, 52
- setYawControlMode
  - osgGA::FlightManipulator, 36
- SpeedControlMode
  - osgGA::DriveManipulator, 20
- SphericalManipulator, 132
  - osgGA::SphericalManipulator, 78
- SphericalManipulator.cpp, 133
- src/ Directory Reference, 6
- src/osgGA/ Directory Reference, 4
- StateSetManipulator, 134
  - osgGA::StateSetManipulator, 84
  - OSGGA\_STATESSET\_MANIPULATOR, 134
- StateSetManipulator.cpp, 135
- T -**
- TabletPen
  - osgGA::GUIEventAdapter::TabletPen, 86
- TabletPointerType
  - osgGA::GUIEventAdapter, 49
- tabletPointerType
  - osgGA::GUIEventAdapter::TabletPen, 86
- takeEvents
  - osgGA::EventQueue, 28
- tb\_project\_to\_sphere
  - osgGA::NodeTrackerManipulator, 73
  - osgGA::TerrainManipulator, 91
  - osgGA::TrackballManipulator, 97
- TerrainManipulator, 136
  - osgGA::TerrainManipulator, 89
  - OSGGA\_TERRAINMANIPULATOR, 136
- TerrainManipulator.cpp, 137
- TRACKBALLSIZE, 137
- tiltX
  - osgGA::GUIEventAdapter::TabletPen, 86
- tiltY
  - osgGA::GUIEventAdapter::TabletPen, 86
- time
  - osgGA::GUIEventAdapter, 52
- TRACKBALL
  - osgGA::NodeTrackerManipulator, 70
- trackball
  - osgGA::NodeTrackerManipulator, 73
  - osgGA::TerrainManipulator, 91
  - osgGA::TrackballManipulator, 97
- TrackballManipulator, 138
  - osgGA::TrackballManipulator, 95
  - OSGGA\_TRACKBALLMANIPULATOR, 138
- TrackballManipulator.cpp, 139
- TRACKBALLSIZE
  - NodeTrackerManipulator.cpp, 131
  - TerrainManipulator.cpp, 137
- TrackerMode
  - osgGA::NodeTrackerManipulator, 70
- traverseGeode
  - osgGA::EventVisitor, 32
- U -**
- UFOManipulator, 140
  - osgGA::UFOManipulator, 102
  - OSGGA\_UFO\_MANIPULATOR\_DEF, 140
- UFOManipulator.cpp, 141
  - M\_PI, 141
- UNKNOWN
  - osgGA::GUIEventAdapter, 49
- USE\_MOUSE\_BUTTONS\_FOR\_SPEED
  - osgGA::DriveManipulator, 20
- USE\_MOUSE\_Y\_FOR\_SPEED
  - osgGA::DriveManipulator, 20
- USER
  - osgGA::GUIEventAdapter, 45
- userEvent
  - osgGA::EventQueue, 28
- V -**
- valid
  - osgGA::AnimationPathManipulator, 12
- validateNodePath
  - osgGA::NodeTrackerManipulator, 73
- Version, 142
  - OSGGA\_VERSION, 142
  - osgGAGetLibraryName, 142
  - osgGAGetVersion, 142
- Version.cpp, 143
  - osgGAGetLibraryName, 143
  - osgGAGetVersion, 143
- W -**
- windowResize
  - osgGA::EventQueue, 28, 29
- Y -**
- Y\_INCREASING\_DOWNWARDS
  - osgGA::GUIEventAdapter, 49
- Y\_INCREASING\_UPWARDS
  - osgGA::GUIEventAdapter, 49
- YAW\_AUTOMATICALLY\_WHEN\_BANKED
  - osgGA::FlightManipulator, 35
- YawControlMode
  - osgGA::FlightManipulator, 35
- Z -**
- zoomOn
  - osgGA::SphericalManipulator, 80