

EvoDeMo

Generated by Doxygen 1.6.1

Fri May 14 00:51:03 2010

Contents

1	Class Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	CArray2d Class Reference	5
3.1.1	Detailed Description	6
3.1.2	Constructor & Destructor Documentation	6
3.1.2.1	CArray2d	6
3.1.2.2	~CArray2d	6
3.1.3	Member Function Documentation	7
3.1.3.1	AllocMemory	7
3.1.3.2	ClearArray	7
3.1.3.3	ClearArray	7
3.1.3.4	DeleteMemory	7
3.1.3.5	GetErrorFlag	7
3.1.3.6	GetHeight	7
3.1.3.7	GetHeight	7
3.1.3.8	GetWidth	7
3.1.3.9	GetWidth	7
3.1.3.10	InitArray	8
3.1.3.11	operator()	8
3.1.3.12	operator()	8
3.1.3.13	operator=	8
3.2	CCellularAutomata Class Reference	9
3.2.1	Detailed Description	10
3.2.2	Constructor & Destructor Documentation	10

3.2.2.1	CCellularAutomata	10
3.2.2.2	~CCellularAutomata	10
3.2.3	Member Function Documentation	10
3.2.3.1	DeleteSpace	10
3.2.3.2	DeleteSpaceInit	10
3.2.3.3	GetErrorFlag	10
3.2.3.4	GetInitSpace	11
3.2.3.5	GetRulesTable	11
3.2.3.6	GetSpace	11
3.2.3.7	GetStepsDone	11
3.2.3.8	GoL	11
3.2.3.9	InitMemory	11
3.2.3.10	InitSpace	11
3.2.3.11	IsInitDone	11
3.2.3.12	ReInit	11
3.2.3.13	SetConfigCore	11
3.2.3.14	Step	12
3.2.3.15	StepGoL	12
3.3	CConfigCore Class Reference	13
3.3.1	Constructor & Destructor Documentation	15
3.3.1.1	CConfigCore	15
3.3.2	Member Function Documentation	15
3.3.2.1	GetCrossoverCount	15
3.3.2.2	GetCrossoverProbability	15
3.3.2.3	GetDefaultState	16
3.3.2.4	GetEvolutionRepetitionsCount	16
3.3.2.5	GetExportFileModeCa	16
3.3.2.6	GetExportFileModeGa	16
3.3.2.7	GetExportFilePath	16
3.3.2.8	GetExportLogCore	16
3.3.2.9	GetGenerationsCount	16
3.3.2.10	GetGenomeType	16
3.3.2.11	GetGuiDisplayModeCA	16
3.3.2.12	GetGuiDisplayModeCATimeout	16
3.3.2.13	GetImportGenomeFile	16
3.3.2.14	GetMoveDirection	17

3.3.2.15	GetMoveDistance	17
3.3.2.16	GetMutationCount	17
3.3.2.17	GetMutationProbability	17
3.3.2.18	GetPopulationSize	17
3.3.2.19	GetSpaceSizeX	17
3.3.2.20	GetSpaceSizeY	17
3.3.2.21	GetSpaceType	17
3.3.2.22	GetStatesCount	17
3.3.2.23	GetStepsCountCA	17
3.3.2.24	IsImportGenomeEnabledReevolve	17
3.3.2.25	IsImportGenomeEnabledSimulation	18
3.3.2.26	SetCrossoverCount	18
3.3.2.27	SetCrossoverProbability	18
3.3.2.28	SetDefaultState	18
3.3.2.29	SetEvolutionRepetitionsCount	18
3.3.2.30	SetExportFileModeCa	18
3.3.2.31	SetExportFileModeGa	18
3.3.2.32	SetExportFilePath	19
3.3.2.33	SetExportLogCore	19
3.3.2.34	SetGenerationsCount	19
3.3.2.35	SetGenomeType	19
3.3.2.36	SetGuiDisplayModeCA	19
3.3.2.37	SetGuiDisplayModeCATimeout	19
3.3.2.38	SetImportGenomeEnabledReevolve	20
3.3.2.39	SetImportGenomeEnabledSimulation	20
3.3.2.40	SetImportGenomeFile	20
3.3.2.41	SetMoveDirection	20
3.3.2.42	SetMoveDistance	20
3.3.2.43	SetMutationCount	20
3.3.2.44	SetMutationProbability	21
3.3.2.45	SetPopulationSize	21
3.3.2.46	SetSpaceSizeX	21
3.3.2.47	SetSpaceSizeY	21
3.3.2.48	SetSpaceType	21
3.3.2.49	SetStatesCount	21
3.3.2.50	SetStepsCountCA	22

3.4	CCore Class Reference	23
3.4.1	Detailed Description	24
3.4.2	Constructor & Destructor Documentation	24
3.4.2.1	CCore	24
3.4.2.2	~CCore	25
3.4.3	Member Function Documentation	25
3.4.3.1	GetConfigCore	25
3.4.3.2	GetErrorFlag	25
3.4.3.3	GetSimCoreAncestorsCount	25
3.4.3.4	GetSimCoreChromosomeID	25
3.4.3.5	GetSimCoreDataStruct	25
3.4.3.6	GetSimCoreDifferentiationMin	25
3.4.3.7	GetSimCoreFitMaxStepCA	25
3.4.3.8	GetSimCoreFitnessMax	25
3.4.3.9	GetSimCoreFitnessMaxNorm	25
3.4.3.10	GetSimCoreGenerationID	25
3.4.3.11	GetSimCoreSameChromosomesCount	26
3.4.3.12	GetSimCoreSameChromosomesGenerationAvarage	26
3.4.3.13	GetSpace	26
3.4.3.14	InitGuiCore	26
3.4.3.15	IsInitDone	26
3.4.3.16	SetCaRunTermination	26
3.4.3.17	SetCoreDataExpiration	26
3.4.3.18	SetSimulationRunning	26
3.4.3.19	ShowDebugConfigCore	26
3.4.3.20	StartThreadCore	27
3.4.3.21	ThreadCoreCheckCoreInitDone	27
3.4.3.22	ThreadCoreErrorSlot	27
3.4.3.23	ThreadCoreNewDataAvailable	27
3.4.3.24	ThreadCoreState	27
3.5	CCrossover Class Reference	28
3.5.1	Detailed Description	28
3.5.2	Constructor & Destructor Documentation	28
3.5.2.1	CCrossover	28
3.5.3	Member Function Documentation	28
3.5.3.1	Crossover	28

3.5.3.2	GetErrorFlag	28
3.5.3.3	SetConfigCore	29
3.6	CExportCA Class Reference	30
3.6.1	Detailed Description	30
3.6.2	Constructor & Destructor Documentation	30
3.6.2.1	CExportCA	30
3.6.3	Member Function Documentation	30
3.6.3.1	ExportCellularAutomaton	30
3.6.3.2	ExportCellularAutomatonInput	31
3.6.3.3	GetErrorFlag	31
3.6.3.4	SetConfigCore	31
3.7	CExportConfig Class Reference	32
3.7.1	Detailed Description	32
3.7.2	Constructor & Destructor Documentation	32
3.7.2.1	CExportConfig	32
3.7.2.2	~CExportConfig	32
3.7.3	Member Function Documentation	33
3.7.3.1	ExportConfig	33
3.7.3.2	GetErrorFlag	33
3.7.3.3	SetIODevice	33
3.7.3.4	WriteDocEnd	33
3.7.3.5	WriteDocStart	33
3.7.3.6	WriteSettingsEvolution	33
3.7.3.7	WriteSettingsExport	33
3.7.3.8	WriteSettingsInput	34
3.7.3.9	WriteSettingsInputCASpace	34
3.7.3.10	WriteSettingsOutput	34
3.8	CExportGA Class Reference	35
3.8.1	Detailed Description	35
3.8.2	Constructor & Destructor Documentation	35
3.8.2.1	CExportGA	35
3.8.3	Member Function Documentation	36
3.8.3.1	ExportChromosome	36
3.8.3.2	ExportGeneration	36
3.8.3.3	GetErrorFlag	36
3.8.3.4	SetConfigCore	36

3.8.3.5	SetIODevice	36
3.8.3.6	WriteDocEnd	36
3.8.3.7	WriteDocStart	36
3.8.3.8	WriteGenome	37
3.9	CExportLog Class Reference	38
3.9.1	Detailed Description	38
3.9.2	Constructor & Destructor Documentation	39
3.9.2.1	CExportLog	39
3.9.2.2	~CExportLog	39
3.9.3	Member Function Documentation	39
3.9.3.1	CloseLogFile	39
3.9.3.2	OpenLogFile	39
3.9.3.3	SetFileName	39
3.9.3.4	SetFilePath	39
3.9.3.5	ShowMessageBox	39
3.9.3.6	ShowMessageBox	40
3.9.3.7	WriteErrorLog	40
3.9.3.8	WriteErrorLog	40
3.9.3.9	WriteInfoLog	40
3.10	CFitness Class Reference	41
3.10.1	Detailed Description	42
3.10.2	Constructor & Destructor Documentation	42
3.10.2.1	CFitness	42
3.10.3	Member Function Documentation	42
3.10.3.1	_Fitness_01	42
3.10.3.2	_Fitness_02	42
3.10.3.3	_MoveA	42
3.10.3.4	Fitness	43
3.10.3.5	GetCornerPosX	43
3.10.3.6	GetCornerPosY	43
3.10.3.7	GetErrorFlag	43
3.10.3.8	IdentifyCorners	43
3.10.3.9	IsLiveOrganismInSpace	44
3.10.3.10	SetConfigCore	44
3.11	CGeneticAlgorithm Class Reference	45
3.11.1	Detailed Description	46

3.11.2	Constructor & Destructor Documentation	47
3.11.2.1	CGeneticAlgorithm	47
3.11.2.2	~CGeneticAlgorithm	47
3.11.3	Member Function Documentation	47
3.11.3.1	ClearVectors	47
3.11.3.2	CreateNextGeneration	47
3.11.3.3	CreateNextGeneration	47
3.11.3.4	Crossover	47
3.11.3.5	ExistsGenomeId	47
3.11.3.6	Fitness	48
3.11.3.7	GetActualPopulation	48
3.11.3.8	GetActualPopulationSize	48
3.11.3.9	GetBestGenome	48
3.11.3.10	GetCornerPosX	48
3.11.3.11	GetCornerPosY	48
3.11.3.12	GetErrorFlag	48
3.11.3.13	GetGenome	48
3.11.3.14	IdentifyCorners	49
3.11.3.15	ImportGenomeToPopulation	49
3.11.3.16	InitGenonetypeStandard	49
3.11.3.17	InitGenotype	49
3.11.3.18	InitGenotypeInstruction	49
3.11.3.19	InitGenotypeReevolve	49
3.11.3.20	Mutation	49
3.11.3.21	RefillPopulation	49
3.11.3.22	RenumberActualGeneration	50
3.11.3.23	Selection	50
3.11.3.24	SetConfigCore	50
3.12	CGenome Class Reference	51
3.12.1	Detailed Description	53
3.12.2	Constructor & Destructor Documentation	53
3.12.2.1	CGenome	53
3.12.2.2	CGenome	54
3.12.2.3	~CGenome	54
3.12.3	Member Function Documentation	54
3.12.3.1	GetAncestorsCount	54

3.12.3.2	GetDifferentiationMin	54
3.12.3.3	GetErrorFlag	54
3.12.3.4	GetFitness	54
3.12.3.5	GetFitnessMax	54
3.12.3.6	GetFitnessMaxNorm	54
3.12.3.7	GetFitnessMaxStepCA	54
3.12.3.8	GetFitnessNorm	55
3.12.3.9	GetGene	55
3.12.3.10	GetGenomeType	55
3.12.3.11	GetInstructionGenome	55
3.12.3.12	GetMutatedGenes	55
3.12.3.13	GetMutatedGenesAllAncestors	55
3.12.3.14	GetMutatedGenesTotal	55
3.12.3.15	GetParentGenerationId	55
3.12.3.16	GetParentGenomeId	55
3.12.3.17	GetThisGenerationId	55
3.12.3.18	GetThisGenomeId	55
3.12.3.19	GetThisInitGenerationId	56
3.12.3.20	GetThisInitGenomeId	56
3.12.3.21	SetAncestorsCount	56
3.12.3.22	SetFitness	56
3.12.3.23	SetFitness	56
3.12.3.24	SetFitnessNorm	56
3.12.3.25	SetGene	56
3.12.3.26	SetMutatedGenes	57
3.12.3.27	SetMutatedGenesTotal	57
3.12.3.28	SetParentGenomeId	57
3.12.3.29	SetThisGenomeId	57
3.12.3.30	SetThisInitGenomeId	57
3.13	CGenomeType2 Class Reference	58
3.13.1	Detailed Description	58
3.13.2	Constructor & Destructor Documentation	58
3.13.2.1	CGenomeType2	58
3.13.2.2	CGenomeType2	58
3.13.3	Member Function Documentation	58
3.13.3.1	GetGene	58

3.13.3.2	GetGenome	59
3.13.3.3	SetGene	59
3.14	CGenomeType2_Ins Class Reference	60
3.14.1	Detailed Description	60
3.14.2	Constructor & Destructor Documentation	60
3.14.2.1	CGenomeType2_Ins	60
3.14.2.2	CGenomeType2_Ins	61
3.14.3	Member Function Documentation	61
3.14.3.1	GetBitValue	61
3.14.3.2	GetGene	61
3.14.3.3	GetInstruction	61
3.14.3.4	GetPostCondition	61
3.14.3.5	GetPreCondition	61
3.14.3.6	GetPreConditionBit	62
3.14.3.7	GetPreConditionLogic	62
3.14.3.8	GetPreConditionLogicBit	62
3.14.3.9	SetBitValue	62
3.14.3.10	SetInstruction	62
3.14.3.11	SetPostCondition	63
3.14.3.12	SetPreCondition	63
3.14.3.13	SetPreConditionBit	63
3.14.3.14	SetPreConditionLogic	63
3.14.3.15	SetPreConditionLogicBit	63
3.15	CGenomeType2_Nbh5 Class Reference	64
3.15.1	Detailed Description	64
3.15.2	Constructor & Destructor Documentation	64
3.15.2.1	CGenomeType2_Nbh5	64
3.15.2.2	CGenomeType2_Nbh5	64
3.15.3	Member Function Documentation	64
3.15.3.1	GetGene	64
3.15.3.2	GetGenome	65
3.15.3.3	SetGene	65
3.16	CGenomeType3 Class Reference	66
3.16.1	Detailed Description	66
3.16.2	Constructor & Destructor Documentation	66
3.16.2.1	CGenomeType3	66

3.16.2.2	CGenomeType3	66
3.16.3	Member Function Documentation	66
3.16.3.1	GetGene	66
3.16.3.2	SetGene	67
3.17	CGenomeType3_Nbh5 Class Reference	68
3.17.1	Detailed Description	68
3.17.2	Constructor & Destructor Documentation	68
3.17.2.1	CGenomeType3_Nbh5	68
3.17.2.2	CGenomeType3_Nbh5	68
3.17.3	Member Function Documentation	68
3.17.3.1	GetGene	68
3.17.3.2	GetGenome	69
3.17.3.3	SetGene	69
3.18	CGenomeType4 Class Reference	70
3.18.1	Detailed Description	70
3.18.2	Constructor & Destructor Documentation	70
3.18.2.1	CGenomeType4	70
3.18.2.2	CGenomeType4	70
3.18.3	Member Function Documentation	70
3.18.3.1	GetGene	70
3.18.3.2	SetGene	71
3.19	CGenomeType4_Nbh5 Class Reference	72
3.19.1	Detailed Description	72
3.19.2	Constructor & Destructor Documentation	72
3.19.2.1	CGenomeType4_Nbh5	72
3.19.2.2	CGenomeType4_Nbh5	72
3.19.3	Member Function Documentation	72
3.19.3.1	GetGene	72
3.19.3.2	GetGenome	73
3.19.3.3	SetGene	73
3.20	CGraphicsItem Class Reference	74
3.20.1	Detailed Description	75
3.20.2	Constructor & Destructor Documentation	75
3.20.2.1	CGraphicsItem	75
3.20.2.2	CGraphicsItem	75
3.20.2.3	CGraphicsItem	75

3.20.3	Member Function Documentation	75
3.20.3.1	boundingRect	75
3.20.3.2	GetPosX	75
3.20.3.3	GetPosY	75
3.20.3.4	GetState	75
3.20.3.5	GetStatesCount	76
3.20.3.6	IsEditable	76
3.20.3.7	mouseMoveEvent	76
3.20.3.8	mousePressEvent	76
3.20.3.9	mouseReleaseEvent	76
3.20.3.10	paint	76
3.20.3.11	SetEditable	77
3.20.3.12	SetState	77
3.20.3.13	SetStatesCount	77
3.21	CGraphicsItemConfig Class Reference	78
3.21.1	Detailed Description	78
3.21.2	Constructor & Destructor Documentation	78
3.21.2.1	CGraphicsItemConfig	78
3.21.3	Member Function Documentation	79
3.21.3.1	GetCellActState	79
3.21.3.2	GetDefState	79
3.21.3.3	GetEditable	79
3.21.3.4	GetStatesCount	79
3.21.3.5	SetCellActState	79
3.21.3.6	SetDefState	79
3.21.3.7	SetEditable	79
3.21.3.8	SetStatesCount	79
3.22	CGraphicsScene Class Reference	81
3.22.1	Detailed Description	82
3.22.2	Constructor & Destructor Documentation	82
3.22.2.1	CGraphicsScene	82
3.22.2.2	~CGraphicsScene	82
3.22.3	Member Function Documentation	82
3.22.3.1	BuildScene	82
3.22.3.2	DeleteScene	82
3.22.3.3	GetCell	83

3.22.3.4	GetCellConfig	83
3.22.3.5	GetHeight	83
3.22.3.6	GetScene	83
3.22.3.7	GetWidth	83
3.22.3.8	SetConfigDefState	83
3.22.3.9	SetConfigEditable	83
3.22.3.10	SetConfigStatesCount	83
3.22.3.11	SetHeight	84
3.22.3.12	SetWidth	84
3.23	CGraphicsView Class Reference	85
3.23.1	Detailed Description	85
3.23.2	Constructor & Destructor Documentation	85
3.23.2.1	CGraphicsView	85
3.23.3	Member Function Documentation	85
3.23.3.1	minimumSizeHint	85
3.23.3.2	setupMatrix	85
3.23.3.3	view	86
3.23.3.4	zoomIn	86
3.23.3.5	zoomOut	86
3.24	CImportConfig Class Reference	87
3.24.1	Detailed Description	87
3.24.2	Constructor & Destructor Documentation	87
3.24.2.1	CImportConfig	87
3.24.3	Member Function Documentation	87
3.24.3.1	GetErrorFlag	87
3.24.3.2	ImportConfig	88
3.24.3.3	ReadDocEnd	88
3.24.3.4	ReadDocStart	88
3.24.3.5	ReadSettingsEvolution	88
3.24.3.6	ReadSettingsExport	88
3.24.3.7	ReadSettingsInput	88
3.24.3.8	ReadSettingsInputCASpace	88
3.24.3.9	ReadSettingsOutput	89
3.24.3.10	SetIODevice	89
3.25	CImportGA Class Reference	90
3.25.1	Detailed Description	90

3.25.2	Constructor & Destructor Documentation	90
3.25.2.1	CImportGA	90
3.25.3	Member Function Documentation	90
3.25.3.1	GetErrorFlag	90
3.25.3.2	ImportChromosome	91
3.25.3.3	ReadDocStart	91
3.25.3.4	ReadGenome	91
3.25.3.5	SetConfigCore	91
3.25.3.6	SetIODevice	91
3.25.3.7	WriteGenesIntoGenome	91
3.26	CInputCA Class Reference	92
3.26.1	Detailed Description	92
3.26.2	Constructor & Destructor Documentation	92
3.26.2.1	CInputCA	92
3.26.3	Member Function Documentation	92
3.26.3.1	ExportCA	92
3.26.3.2	GetErrorFlag	93
3.26.3.3	ImportCA	93
3.26.3.4	SetConfigCore	93
3.27	CMutation Class Reference	94
3.27.1	Detailed Description	94
3.27.2	Constructor & Destructor Documentation	94
3.27.2.1	CMutation	94
3.27.3	Member Function Documentation	94
3.27.3.1	_Mutation_01	94
3.27.3.2	_Mutation_02	95
3.27.3.3	_Mutation_Ins_01	95
3.27.3.4	GetErrorFlag	95
3.27.3.5	Mutation	95
3.27.3.6	SetConfigCore	95
3.28	CRandom Class Reference	96
3.28.1	Detailed Description	96
3.28.2	Constructor & Destructor Documentation	96
3.28.2.1	CRandom	96
3.28.3	Member Function Documentation	96
3.28.3.1	GetErrorFlag	96

3.28.3.2	LCG	97
3.28.3.3	Random	97
3.28.3.4	Uniform	97
3.28.3.5	Uniform	97
3.28.3.6	UniformStdLib	97
3.29	CRulesTable Class Reference	98
3.29.1	Detailed Description	98
3.29.2	Constructor & Destructor Documentation	98
3.29.2.1	CRulesTable	98
3.29.3	Member Function Documentation	98
3.29.3.1	at	98
3.29.3.2	GetGenome	98
3.29.3.3	SetConfigCore	99
3.29.3.4	SetGenome	99
3.30	CSelection Class Reference	100
3.30.1	Detailed Description	100
3.30.2	Constructor & Destructor Documentation	100
3.30.2.1	CSelection	100
3.30.3	Member Function Documentation	100
3.30.3.1	_Selection_01	100
3.30.3.2	_Selection_02	101
3.30.3.3	Selection	101
3.30.3.4	SetConfigCore	101
3.31	CSpace Class Reference	102
3.31.1	Detailed Description	102
3.31.2	Constructor & Destructor Documentation	103
3.31.2.1	CSpace	103
3.31.2.2	~CSpace	103
3.31.3	Member Function Documentation	103
3.31.3.1	at	103
3.31.3.2	at	103
3.31.3.3	atGrid	103
3.31.3.4	atGrid	104
3.31.3.5	atTorus	104
3.31.3.6	atTorus	104
3.31.3.7	GetErrorFlag	104

3.31.3.8	GetHeight	104
3.31.3.9	GetSpaceType	104
3.31.3.10	GetWidth	105
3.32	CTFunction Class Reference	106
3.32.1	Detailed Description	106
3.32.2	Constructor & Destructor Documentation	106
3.32.2.1	CTFunction	106
3.32.3	Member Function Documentation	106
3.32.3.1	CalculateIndexGrid	106
3.32.3.2	CalculateIndexTorus	107
3.32.3.3	NextSpace	107
3.32.3.4	NextSpaceGenomeInstruction	107
3.32.3.5	NextSpaceGenomeStandard	107
3.32.3.6	SetConfigCore	107
3.32.3.7	SetRulesTable	108
3.33	CThreadCore Class Reference	109
3.33.1	Detailed Description	112
3.33.2	Constructor & Destructor Documentation	112
3.33.2.1	CThreadCore	112
3.33.2.2	~CThreadCore	112
3.33.3	Member Function Documentation	112
3.33.3.1	CheckCaRunTermination	112
3.33.3.2	CheckCorePointers	112
3.33.3.3	CheckThreadLoopTermination	112
3.33.3.4	ClearTmpCAs	112
3.33.3.5	FileExportCaInit	112
3.33.3.6	FileExportCaSteps	112
3.33.3.7	FileExportGa	113
3.33.3.8	GetErrorFlag	113
3.33.3.9	GetInitSpace	113
3.33.3.10	GetSpace	113
3.33.3.11	InitCore	113
3.33.3.12	InitCoreCAMemory	113
3.33.3.13	InitCoreCellularAutomata	113
3.33.3.14	InitCoreGeneticAlgorithm	113
3.33.3.15	InitExport	113

3.33.3.16	InitTmpCAs	113
3.33.3.17	IsInitDone	113
3.33.3.18	ReinitCore	114
3.33.3.19	run	114
3.33.3.20	RunGenomeCaSimulation	114
3.33.3.21	RunGuiMode	114
3.33.3.22	SetConfigCore	114
3.33.3.23	SetCoreDataExpiration	114
3.33.3.24	SetCoreDataGA	114
3.33.3.25	SetCoreDataValidity	115
3.33.3.26	SetCoreSpace	115
3.33.3.27	SetMutex	115
3.33.3.28	SetSimulationRunning	115
3.33.3.29	SetWaitCondition	115
3.33.3.30	StoreGenomeDataForGui	115
3.33.3.31	SyncDataWithCore	116
3.33.3.32	TerminateCaRun	116
3.33.3.33	TerminateThreadLoop	116
3.33.3.34	WriteCAspaceFromCoreIntoCA	116
3.33.3.35	WriteDataCAToCore	116
3.33.3.36	WriteDataGAToCore	116
3.34	CWidgetEvolution Class Reference	117
3.34.1	Detailed Description	119
3.34.2	Constructor & Destructor Documentation	119
3.34.2.1	CWidgetEvolution	119
3.34.2.2	~CWidgetEvolution	119
3.34.3	Member Function Documentation	119
3.34.3.1	ApplyPressed	119
3.34.3.2	CancelPressed	120
3.34.3.3	changeEvent	120
3.34.3.4	CrossoverCountChanged	120
3.34.3.5	CrossoverProbChanged	120
3.34.3.6	DirectionChanged	120
3.34.3.7	DistanceChanged	120
3.34.3.8	GenerationsChanged	120
3.34.3.9	GenomeTypeChanged	121

3.34.3.10	GetCrossoverCount	121
3.34.3.11	GetCrossoverProbability	121
3.34.3.12	GetGenerationsCount	121
3.34.3.13	GetGenomeType	121
3.34.3.14	GetImportGenomeFile	121
3.34.3.15	GetMoveDirection	121
3.34.3.16	GetMoveDistance	121
3.34.3.17	GetMutationCount	121
3.34.3.18	GetMutationProbability	121
3.34.3.19	GetPopulationSize	121
3.34.3.20	GetRepetitionsCount	122
3.34.3.21	GetStepsCount	122
3.34.3.22	ImportFileButtonPressed	122
3.34.3.23	ImportFileEnableReevolve	122
3.34.3.24	ImportFileEnableSimulation	122
3.34.3.25	IsApplied	122
3.34.3.26	IsImportGenomeEnabledReevolve	122
3.34.3.27	IsImportGenomeEnabledSimulation	122
3.34.3.28	MutationCountChanged	122
3.34.3.29	MutationProbChanged	123
3.34.3.30	PopulationChanged	123
3.34.3.31	RepetitionsChanged	123
3.34.3.32	SetCrossoverCount	123
3.34.3.33	SetCrossoverProbability	123
3.34.3.34	SetGenerationsCount	123
3.34.3.35	SetGenomeType	124
3.34.3.36	SetInitDone	124
3.34.3.37	SetMoveDirection	124
3.34.3.38	SetMoveDistance	124
3.34.3.39	SetMutationCount	124
3.34.3.40	SetMutationProbability	124
3.34.3.41	SetPopulationSize	124
3.34.3.42	SetRepetitionsCount	125
3.34.3.43	SetStepsCount	125
3.34.3.44	StepsChanged	125
3.35	CWidgetExport Class Reference	126

3.35.1	Detailed Description	127
3.35.2	Constructor & Destructor Documentation	127
3.35.2.1	CWidgetExport	127
3.35.2.2	~CWidgetExport	127
3.35.3	Member Function Documentation	127
3.35.3.1	ApplyPressed	127
3.35.3.2	CancelPressed	127
3.35.3.3	changeEvent	128
3.35.3.4	FileExportCACHanged	128
3.35.3.5	FileExportGACHanged	128
3.35.3.6	FileExportPathChanged	128
3.35.3.7	GetFileExportModeCA	128
3.35.3.8	GetFileExportModeGA	128
3.35.3.9	GetFileExportPath	128
3.35.3.10	GetGuiDataDisplayModeCA	128
3.35.3.11	GetGuiDataDisplayModeTimeout	128
3.35.3.12	GuiModeCADisplayModeChanged	129
3.35.3.13	GuiModeCATimeoutChanged	129
3.35.3.14	IsApplied	129
3.35.3.15	SetFileExportModeCA	129
3.35.3.16	SetFileExportModeGA	129
3.35.3.17	SetGuiDataDisplayModeCA	129
3.35.3.18	SetGuiDataDisplayModeTimeout	129
3.35.3.19	SetInitDone	130
3.36	CWidgetInput Class Reference	131
3.36.1	Detailed Description	133
3.36.2	Constructor & Destructor Documentation	133
3.36.2.1	CWidgetInput	133
3.36.2.2	~CWidgetInput	133
3.36.3	Member Function Documentation	133
3.36.3.1	ApplyPressed	133
3.36.3.2	CancelPressed	133
3.36.3.3	changeEvent	133
3.36.3.4	CheckPos	133
3.36.3.5	CheckSize	134
3.36.3.6	DeleteInputGui	134

3.36.3.7	GetGrid	134
3.36.3.8	ImportPressed	134
3.36.3.9	IsApplied	134
3.36.3.10	ReinitInputView	134
3.36.3.11	SetInitDone	134
3.36.3.12	SetInputGridSize	134
3.36.3.13	SetInputGridSizeX	134
3.36.3.14	SetInputGridSizeY	135
3.36.3.15	SetOutputApplied	135
3.36.3.16	SetOutputMapPosX	135
3.36.3.17	SetOutputMapPosY	135
3.36.3.18	SetOutputSizeX	135
3.36.3.19	SetOutputSizeY	135
3.36.3.20	SetStatesCount	136
3.36.3.21	SetupInputView	136
3.36.3.22	SizeChangedX	136
3.36.3.23	SizeChangedY	136
3.36.3.24	SizeChangeUsed	136
3.36.3.25	StatesChandeUsed	136
3.36.3.26	StatesChanged	136
3.37	CWidgetOutput Class Reference	137
3.37.1	Detailed Description	138
3.37.2	Constructor & Destructor Documentation	138
3.37.2.1	CWidgetOutput	138
3.37.2.2	~CWidgetOutput	139
3.37.3	Member Function Documentation	139
3.37.3.1	ApplyPressed	139
3.37.3.2	ArrayTypeChanged	139
3.37.3.3	CancelPressed	139
3.37.3.4	changeEvent	139
3.37.3.5	CheckPos	139
3.37.3.6	CheckSize	139
3.37.3.7	GetOutputArrayType	140
3.37.3.8	GetOutputGridSizeX	140
3.37.3.9	GetOutputGridSizeY	140
3.37.3.10	GetOutputMapPosX	140

3.37.3.11	GetOutputMapPosY	140
3.37.3.12	IsApplied	140
3.37.3.13	PosChangedX	140
3.37.3.14	PosChangedY	140
3.37.3.15	SetInitDone	140
3.37.3.16	SetInputApplied	141
3.37.3.17	SetInputSizeX	141
3.37.3.18	SetInputSizeY	141
3.37.3.19	SetOutputArrayType	141
3.37.3.20	SetOutputGridSizeX	141
3.37.3.21	SetOutputGridSizeY	141
3.37.3.22	SetOutputMapPosX	142
3.37.3.23	SetOutputMapPosY	142
3.37.3.24	SizeChangedX	142
3.37.3.25	SizeChangedY	142
3.38	CWidgetRunCA Class Reference	143
3.38.1	Detailed Description	143
3.38.2	Constructor & Destructor Documentation	144
3.38.2.1	CWidgetRunCA	144
3.38.2.2	~CWidgetRunCA	144
3.38.3	Member Function Documentation	144
3.38.3.1	changeEvent	144
3.38.3.2	DeleteRun	144
3.38.3.3	DeleteRunGridView	144
3.38.3.4	GetGrid	144
3.38.3.5	InitRunGrid	144
3.38.3.6	SetRunGridSizeX	145
3.38.3.7	SetRunGridSizeY	145
3.38.3.8	SetRunGridStates	145
3.38.3.9	SetupRunGridView	145
3.39	CWidgetRunEvo Class Reference	146
3.39.1	Detailed Description	146
3.39.2	Constructor & Destructor Documentation	146
3.39.2.1	CWidgetRunEvo	146
3.39.2.2	~CWidgetRunEvo	146
3.39.3	Member Function Documentation	147

3.39.3.1	changeEvent	147
3.39.3.2	DeleteRun	147
3.39.3.3	InitRun	147
3.39.3.4	SetAncestorsCount	147
3.39.3.5	SetChromosomeID	147
3.39.3.6	SetDifferentiationMin	147
3.39.3.7	SetFitMaxStepCA	147
3.39.3.8	SetFitnessMax	148
3.39.3.9	SetGenerationID	148
3.39.3.10	SetSameChromosomesCount	148
3.39.3.11	SetSameChromosomesGenerationAvarage	148
3.40	CWindowMain Class Reference	149
3.40.1	Detailed Description	151
3.40.2	Constructor & Destructor Documentation	151
3.40.2.1	CWindowMain	151
3.40.2.2	~CWindowMain	151
3.40.3	Member Function Documentation	151
3.40.3.1	CellularAutomatonRunning	151
3.40.3.2	CellularAutomatonStep	151
3.40.3.3	changeEvent	152
3.40.3.4	CheckSettingApplyValidity	152
3.40.3.5	CheckUiLayouts	152
3.40.3.6	CoreCheckInitDone	152
3.40.3.7	CoreDataMapToGui	152
3.40.3.8	DeleteCore	152
3.40.3.9	DeletePressed	152
3.40.3.10	EvolutionInit	152
3.40.3.11	EvolutionRunning	152
3.40.3.12	EvoStartPressed	152
3.40.3.13	GuiDisplayCaMove	153
3.40.3.14	InitCore	153
3.40.3.15	InitPressed	153
3.40.3.16	MenuActionAbout	153
3.40.3.17	MenuActionExportRunConfig	153
3.40.3.18	MenuActionImportRunConfig	153
3.40.3.19	RunPressed	153

3.40.3.20 SimulationDelete	153
3.40.3.21 StepPressed	153
3.40.3.22 StopPressed	153
3.40.3.23 TerminatePressed	153
3.40.3.24 WidgetEvolutionApplied	154
3.40.3.25 WidgetExportApplied	154
3.40.3.26 WidgetInputApplied	154
3.40.3.27 WidgetOutputApplied	154
3.41 stArray Struct Reference	155
3.41.1 Detailed Description	155
3.42 stGeneInstruction Struct Reference	156
3.42.1 Detailed Description	156
3.43 stThreadCoreDataGA Struct Reference	157
3.43.1 Detailed Description	157

Chapter 1

Class Index

1.1 Class Hierarchy

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

CArray2d	5
CSpace	102
CCellularAutomata	9
CConfigCore	13
CCore	23
CCrossover	28
CExportCA	30
CExportConfig	32
CExportGA	35
CExportLog	38
CFitness	41
CGeneticAlgorithm	45
CGenome	51
CGenomeType2	58
CGenomeType2_Ins	60
CGenomeType2_Nbh5	64
CGenomeType3	66
CGenomeType3_Nbh5	68
CGenomeType4	70
CGenomeType4_Nbh5	72
CGraphicsItem	74
CGraphicsItemConfig	78
CGraphicsScene	81
CGraphicsView	85
CImportConfig	87
CImportGA	90
CInputCA	92
CMutation	94
CRandom	96
CRulesTable	98
CSelection	100
CTFunction	106
CThreadCore	109

CWidgetEvolution	117
CWidgetExport	126
CWidgetInput	131
CWidgetOutput	137
CWidgetRunCA	143
CWidgetRunEvo	146
CWindowMain	149
stArray	155
stGeneInstruction	156
stThreadCoreDataGA	157

Chapter 2

Class Index

2.1 Class List

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

CArray2d	5
CCellularAutomata	9
CConfigCore	13
CCore	23
CCrossover	28
CExportCA	30
CExportConfig	32
CExportGA	35
CExportLog	38
CFitness	41
CGeneticAlgorithm	45
CGenome	51
CGenomeType2	58
CGenomeType2_Ins	60
CGenomeType2_Nbh5	64
CGenomeType3	66
CGenomeType3_Nbh5	68
CGenomeType4	70
CGenomeType4_Nbh5	72
CGraphicsItem	74
CGraphicsItemConfig	78
CGraphicsScene	81
CGraphicsView	85
CImportConfig	87
CImportGA	90
CInputCA	92
CMutation	94
CRandom	96
CRulesTable	98
CSelection	100
CSpace	102
CTFunction	106
CThreadCore	109

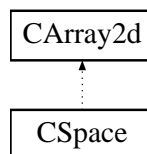
CWidgetEvolution	117
CWidgetExport	126
CWidgetInput	131
CWidgetOutput	137
CWidgetRunCA	143
CWidgetRunEvo	146
CWindowMain	149
stArray	155
stGeneInstruction	156
stThreadCoreDataGA	157

Chapter 3

Class Documentation

3.1 CArray2d Class Reference

`#include <Array2d.h>`Inheritance diagram for CArray2d::



Public Member Functions

- [CArray2d](#) (int sizeX=SPACE_SIZE_X_DEFAULT, int sizeY=SPACE_SIZE_Y_DEFAULT, BYTE stateDef=CELL_STATE_EMPTY, [CConfigCore](#) *pCC=NULL)
- [~CArray2d](#) ()
- BYTE [operator\(\)](#) (int posX, int posY) const
- BYTE & [operator\(\)](#) (int posX, int posY)
- [CArray2d](#) & [operator=](#) (const [CArray2d](#) &rSide)
- int [GetWidth](#) ()
- int [GetHeight](#) ()
- int [GetWidth](#) () const
- int [GetHeight](#) () const
- void [ClearArray](#) ()
- void [ClearArray](#) (BYTE newDefState)
- int [GetErrorFlag](#) ()

Private Member Functions

- int [AllocMemory](#) ()
- int [DeleteMemory](#) ()
- int [InitArray](#) (int sizeX, int sizeY)

Private Attributes

- int `iSizeX`
array width
- int `iSizeY`
array height
- BYTE `byDefCellState`
cell's default state
- BYTE `byErrCellState`
error state
- struct `stArray` * `array`
pointer to dyn.allocated 2d array
- CConfigCore * `pConfigCoreArray`
pointer to config class
- int `iErrFlag`
error flag

3.1.1 Detailed Description

contains 2d array, which is used as cellular space for CA

3.1.2 Constructor & Destructor Documentation

3.1.2.1 CArray2d::CArray2d (int *sizeX* = SPACE_SIZE_X_DEFAULT, int *sizeY* = SPACE_SIZE_Y_DEFAULT, BYTE *stateDef* = CELL_STATE_EMPTY, CConfigCore * *pCC* = NULL)

class constructor

Parameters:

sizeX x-dimension of array
sizeY y-dimension of array
stateDef default state of cell
**pCC* pointer to config class

3.1.2.2 CArray2d::~~CArray2d ()

class destructor

3.1.3 Member Function Documentation

3.1.3.1 int CArray2d::AllocMemory () [private]

allocates memory for 2d array

3.1.3.2 void CArray2d::ClearArray (BYTE *newDefState*)

clears array - sets all cell to given value

Parameters:

newDefState new default state of cells

3.1.3.3 void CArray2d::ClearArray ()

clears array - sets all cell to default value

3.1.3.4 int CArray2d::DeleteMemory () [private]

deletes memory of 2d array

3.1.3.5 int CArray2d::GetErrorFlag ()

returns error flag

Reimplemented in [CSpace](#).

3.1.3.6 int CArray2d::GetHeight () const

returns height of array

3.1.3.7 int CArray2d::GetHeight ()

returns height of array

Reimplemented in [CSpace](#).

3.1.3.8 int CArray2d::GetWidth () const

returns width of array

3.1.3.9 int CArray2d::GetWidth ()

returns width of array

Reimplemented in [CSpace](#).

3.1.3.10 `int CArray2d::InitArray (int sizeX, int sizeY) [private]`

array init - checking sizes of axes

3.1.3.11 `BYTE & CArray2d::operator() (int posX, int posY)`

returns reference to element at given coordinates

Parameters:

posX position on x-axes

posY position on y-axes

3.1.3.12 `BYTE CArray2d::operator() (int posX, int posY) const`

returns element from given coordinates

Parameters:

posX position on x-axes

posY position on y-axes

3.1.3.13 `CArray2d & CArray2d::operator= (const CArray2d & rSide)`

copies one array into another

Parameters:

&rSide reference to class on right side of '='

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

- Array2d.h
- Array2d.cpp

3.2 CCellularAutomata Class Reference

```
#include <CellularAutomata.h>
```

Public Member Functions

- [CCellularAutomata](#) ([CConfigCore](#) *configCore)
- [~CCellularAutomata](#) ()
- void [Step](#) ()
- void [StepGoL](#) ()
- void [InitMemory](#) ()
- void [InitSpace](#) ()
- void [ReInit](#) ()
- [CRulesTable](#) * [GetRulesTable](#) ()

=====

- void [SetConfigCore](#) ([CConfigCore](#) *ccc)
- [CSpace](#) * [GetSpace](#) ()
- [CSpace](#) * [GetInitSpace](#) ()
- int [GetStepsDone](#) ()
- int [GetErrorFlag](#) ()
- bool [IsInitDone](#) ()

Private Member Functions

- void [DeleteSpace](#) ()
- void [DeleteSpaceInit](#) ()
- void [GoL](#) ()

Private Attributes

- [CRulesTable](#) rules
instance of rules table
- [CTFunction](#) tfunction
instance of transition function
- [CConfigCore](#) * pConfigCore
pointer to config class
- [CSpace](#) * spaceInit
init space - this class contains init ca space definated by user
- [CSpace](#) * spaceAct
sa space in actual step of computation
- [CSpace](#) * spaceTmp
tmp space is used for saving next step result

- [CSpace * spaceTmpX](#)
pointer used for switching pointers between act and tmp instances
- unsigned int [iStepsDone](#)
number of steps done
- bool [bInitDone](#)
is init done?
- int [iErrFlag](#)
error flag

3.2.1 Detailed Description

main class of cellular automaton this class creates instances of ca space and manages ca computations for computations itself is used class [CTFunction](#), which calculates index to genome (at this index is gene of BYTE data type which represents value of cell in next step), genome itself is mapped into instance of TRulesTable

3.2.2 Constructor & Destructor Documentation

3.2.2.1 CCellularAutomata::CCellularAutomata (CConfigCore * *configCore*)

class constructor

Parameters:

**configCore* pointer to config class

3.2.2.2 CCellularAutomata::~~CCellularAutomata ()

class destructor

3.2.3 Member Function Documentation

3.2.3.1 void CCellularAutomata::DeleteSpace () [private]

deletes actual and tmp ca spaces

3.2.3.2 void CCellularAutomata::DeleteSpaceInit () [private]

deletes init ca space

3.2.3.3 int CCellularAutomata::GetErrorFlag ()

return error flag

3.2.3.4 CSpace * CCellularAutomata::GetInitSpace ()

returns pointer to init ca space

3.2.3.5 CRulesTable * CCellularAutomata::GetRulesTable ()

===== returns pointer to
rules table function

3.2.3.6 CSpace * CCellularAutomata::GetSpace ()

returns pointer to actual CA space

3.2.3.7 int CCellularAutomata::GetStepsDone ()

return number of done steps

3.2.3.8 void CCellularAutomata::GoL () [private]

special funtion for "Game of Life" computations this function does not using chromosomes mapped into transition function

3.2.3.9 void CCellularAutomata::InitMemory ()

inits memory needed for computation

3.2.3.10 void CCellularAutomata::InitSpace ()

init spaces - copies init space into actual space

3.2.3.11 bool CCellularAutomata::IsInitDone ()

checking if initialization is done

3.2.3.12 void CCellularAutomata::ReInit ()

reinitialization of CA

3.2.3.13 void CCellularAutomata::SetConfigCore (CConfigCore * ccc)

sets pointer to config class into this class and transition function class

Parameters:

*ccc pointer to config core class

3.2.3.14 void CCellularAutomata::Step ()

makes one step of ca computation

3.2.3.15 void CCellularAutomata::StepGoL ()

makes one step with special function [GoL\(\)](#), which does not using transition function class

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

- CellularAutomata.h
- CellularAutomata.cpp

3.3 CConfigCore Class Reference

Public Member Functions

- [CConfigCore](#) ()
- void [SetSpaceType](#) (int st)
- int [GetSpaceType](#) ()
- void [SetSpaceSizeX](#) (int x)
- int [GetSpaceSizeX](#) ()
- void [SetSpaceSizeY](#) (int y)
- int [GetSpaceSizeY](#) ()
- void [SetStatesCount](#) (int sc)
- int [GetStatesCount](#) ()
- void [SetDefaultState](#) (int ds)
- BYTE [GetDefaultState](#) ()
- void [SetEvolutionRepetitionsCount](#) (int rc)
- int [GetEvolutionRepetitionsCount](#) ()
- void [SetGenerationsCount](#) (int gc)
- int [GetGenerationsCount](#) ()
- void [SetPopulationSize](#) (int ips)
- int [GetPopulationSize](#) ()
- void [SetMoveDirection](#) (int md)
- int [GetMoveDirection](#) ()
- void [SetStepsCountCA](#) (int sc)
- int [GetStepsCountCA](#) ()
- void [SetMoveDistance](#) (int md)
- int [GetMoveDistance](#) ()
- void [SetCrossoverProbability](#) (int cp)
- int [GetCrossoverProbability](#) ()
- void [SetCrossoverCount](#) (int cc)
- int [GetCrossoverCount](#) ()
- void [SetMutationProbability](#) (int mp)
- int [GetMutationProbability](#) ()
- void [SetMutationCount](#) (int mc)
- int [GetMutationCount](#) ()
- void [SetGenomeType](#) (int gt)
- int [GetGenomeType](#) ()
- void [SetGuiDisplayModeCA](#) (int dm)
- int [GetGuiDisplayModeCA](#) ()
- void [SetGuiDisplayModeCATimeout](#) (int to)
- int [GetGuiDisplayModeCATimeout](#) ()
- void [SetExportFilePath](#) (std::string path)
- std::string [GetExportFilePath](#) ()
- void [SetExportFileModeCa](#) (int mode)
- int [GetExportFileModeCa](#) ()
- void [SetExportFileModeGa](#) (int mode)
- int [GetExportFileModeGa](#) ()
- void [SetImportGenomeFile](#) (std::string file)
- std::string [GetImportGenomeFile](#) ()
- void [SetImportGenomeEnabledSimulation](#) (bool e)

- bool [IsImportGenomeEnabledSimulation](#) ()
- void [SetImportGenomeEnabledReevolve](#) (bool e)
- bool [IsImportGenomeEnabledReevolve](#) ()
- void [SetExportLogCore](#) (CExportLog *ex)
- CExportLog * [GetExportLogCore](#) ()

Private Attributes

- unsigned int [iSpaceType](#)
ca space type - lattice or torus
- unsigned int [iSizeRunArrayX](#)
ca space width
- unsigned int [iSizeRunArrayY](#)
ca space height
- unsigned int [iStatesCount](#)
number of states in ca
- BYTE [byDefCellState](#)
default cell state
- unsigned int [iRepetitions](#)
number of independet runs of evolution
- unsigned int [iGenCount](#)
generations count
- unsigned int [iPopulationSize](#)
population size
- unsigned int [iMoveDir](#)
direction of movement
- unsigned int [iStepsCountCA](#)
steps needed to move at distance
- unsigned int [iDistance](#)
distance of movement
- unsigned int [iCrossoverProb](#)
crossover probability
- unsigned int [iCrossoverCount](#)
number of crossovers
- unsigned int [iMutProb](#)
mutation probability

- unsigned int [iMutCount](#)
number of genes which should be mutated
- unsigned int [iGenomeType](#)
genome type: 9-neighborhood + 2 states, etc.
- unsigned int [iGuiDisplayModeCa](#)
gui diaplay mode
- unsigned int [iGuiDisplayModeCaTimeout](#)
gui ca animation timeout (between two steps of ca in "Run" mode)
- std::string [sgExportFilePath](#)
path for files for exported
- int [iExportFileModeCa](#)
mode for exporting ca spaces
- int [iExportFileModeGa](#)
mode for exporting ga genomes
- std::string [sgImportFileGenome](#)
imported genome file path/name
- bool [bImportFileEnabledSimulation](#)
use imported genome just for ca simulation running
- bool [bImportFileEnabledReevolve](#)
re-evolve imported genome
- [CExportLog](#) * [pExportLogCore](#)
pointer to instance of export log file class created in [CThreadCore](#)

3.3.1 Constructor & Destructor Documentation

3.3.1.1 CConfigCore::CConfigCore ()

class constructor initializes vars to default values

3.3.2 Member Function Documentation

3.3.2.1 int CConfigCore::GetCrossoverCount ()

returns count of crossovers between two genomes

3.3.2.2 int CConfigCore::GetCrossoverProbability ()

returns crossover probability

3.3.2.3 BYTE CConfigCore::GetDefaultState ()

returns ca default state

3.3.2.4 int CConfigCore::GetEvolutionRepetitionsCount ()

returns count of independent runs of ca

3.3.2.5 int CConfigCore::GetExportFileModeCa ()

returns ca export mode

3.3.2.6 int CConfigCore::GetExportFileModeGa ()

returns ga export mode

3.3.2.7 std::string CConfigCore::GetExportFilePath ()

returns path to export folder

3.3.2.8 CExportLog * CConfigCore::GetExportLogCore ()

returns pointer to one and only instance of error log export class

3.3.2.9 int CConfigCore::GetGenerationsCount ()

returns generations count

3.3.2.10 int CConfigCore::GetGenomeType ()

returns genome tupe

3.3.2.11 int CConfigCore::GetGuiDisplayModeCA ()

returns gui display mode for ca

3.3.2.12 int CConfigCore::GetGuiDisplayModeCATimeout ()

returns gui ca animation timeout (between two steps of ca in "Run" mode)

3.3.2.13 std::string CConfigCore::GetImportGenomeFile ()

returns imported genome file path/name

3.3.2.14 int CConfigCore::GetMoveDirection ()

returns movement direction

3.3.2.15 int CConfigCore::GetMoveDistance ()

returns distance (in cells) to which object should be moved

3.3.2.16 int CConfigCore::GetMutationCount ()

returns number of genes which should be mutated

3.3.2.17 int CConfigCore::GetMutationProbability ()

returns mutation probability

3.3.2.18 int CConfigCore::GetPopulationSize ()

returns population size

3.3.2.19 int CConfigCore::GetSpaceSizeX ()

returns ca space width

3.3.2.20 int CConfigCore::GetSpaceSizeY ()

returns ca space height

3.3.2.21 int CConfigCore::GetSpaceType ()

returns ca space type

3.3.2.22 int CConfigCore::GetStatesCount ()

returns ca states count

3.3.2.23 int CConfigCore::GetStepsCountCA ()

returns number of ca steps needed to move object on given distance

3.3.2.24 bool CConfigCore::IsImportGenomeEnabledReevolve ()

returns if imported genome should be re-evolved

3.3.2.25 bool CConfigCore::IsImportGenomeEnabledSimulation ()

returns if genome should be used in ca simulator

3.3.2.26 void CConfigCore::SetCrossoverCount (int *cc*)

sets count of crossovers between two genomes

Parameters:

cc crossovers count

3.3.2.27 void CConfigCore::SetCrossoverProbability (int *cp*)

sets crossover probability

Parameters:

cp crossover probability

3.3.2.28 void CConfigCore::SetDefaultState (int *ds*)

sets ca default state

Parameters:

ds ca default state

3.3.2.29 void CConfigCore::SetEvolutionRepetitionsCount (int *rc*)

sets independent runs of evolution

Parameters:

rc independent runs count

3.3.2.30 void CConfigCore::SetExportFileModeCa (int *mode*)

sets ca export mode

Parameters:

mode ca export mode

3.3.2.31 void CConfigCore::SetExportFileModeGa (int *mode*)

sets ga export mode

Parameters:

mode ga export mode

3.3.2.32 void CConfigCore::SetExportFilePath (std::string *path*)

sets path to export folder

Parameters:

path path to export folder

3.3.2.33 void CConfigCore::SetExportLogCore (CExportLog * *ex*)

sets pointer to one and only instance of error log export class

Parameters:

**ex* pointer to log class

3.3.2.34 void CConfigCore::SetGenerationsCount (int *gc*)

sets generations count

Parameters:

gc generations count

3.3.2.35 void CConfigCore::SetGenomeType (int *gt*)

sets genome type

Parameters:

gt genome type

3.3.2.36 void CConfigCore::SetGuiDisplayModeCA (int *dm*)

sets gui display mode for ca

Parameters:

dm display mode

3.3.2.37 void CConfigCore::SetGuiDisplayModeCATimeout (int *to*)

sets gui ca animation timeout (between two steps of ca in "Run" mode)

Parameters:

to timeout

3.3.2.38 void CConfigCore::SetImportGenomeEnabledReevolve (bool *e*)

sets if imported genome should be re-evolved

Parameters:

e enable re-evolution

3.3.2.39 void CConfigCore::SetImportGenomeEnabledSimulation (bool *e*)

sets if genome should be used in ca simulator

Parameters:

e enable usage in ca simulator

3.3.2.40 void CConfigCore::SetImportGenomeFile (std::string *file*)

sets imported genome file path/name

Parameters:

file imported genome path/name

3.3.2.41 void CConfigCore::SetMoveDirection (int *md*)

sets direction of movement

Parameters:

md movement direction

3.3.2.42 void CConfigCore::SetMoveDistance (int *md*)

sets distance (in cells) to which object should be moved

Parameters:

md movement distance

3.3.2.43 void CConfigCore::SetMutationCount (int *mc*)

sets number of genes which should be mutated

Parameters:

mc mutated genes count

3.3.2.44 void CConfigCore::SetMutationProbability (int *mp*)

sets mutation probability

Parameters:

mp mutation probability

3.3.2.45 void CConfigCore::SetPopulationSize (int *ips*)

sets population size

Parameters:

ips population size

3.3.2.46 void CConfigCore::SetSpaceSizeX (int *x*)

sets ca space width

Parameters:

x ca space width

3.3.2.47 void CConfigCore::SetSpaceSizeY (int *y*)

sets ca space height

Parameters:

y ca space height

3.3.2.48 void CConfigCore::SetSpaceType (int *st*)

sets ca space type

Parameters:

st space type

3.3.2.49 void CConfigCore::SetStatesCount (int *sc*)

sets ca states count

Parameters:

sc ca states count

3.3.2.50 void CConfigCore::SetStepsCountCA (int *sc*)

sets number of ca steps needed to move object on given distance

Parameters:

sc ca steps count

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

- ConfigCore.h
- ConfigCore.cpp

3.4 CCore Class Reference

```
#include <Core.h>
```

Signals

- void **SignalThreadCoreErr** (int)
- void **SignalThreadCoreState** (int)
- void **SignalNewDataAvailable** ()

Public Member Functions

- [CCore](#) ()
- [~CCore](#) ()
- void [InitGuiCore](#) ()
- void [SetCoreDataExpiration](#) (bool e)
- void [SetSimulationRunning](#) (bool sr)
- void [SetCaRunTermination](#) ()
- void [StartThreadCore](#) ()
- bool [IsInitDone](#) ()
- [CConfigCore](#) * [GetConfigCore](#) ()
- [CSpace](#) * [GetSpace](#) ()
- int [GetSimCoreGenerationID](#) ()
- int [GetSimCoreChromosomeID](#) ()
- int [GetSimCoreAncestorsCount](#) ()
- int [GetSimCoreFitnessMax](#) ()
- double [GetSimCoreFitnessMaxNorm](#) ()
- int [GetSimCoreDifferentiationMin](#) ()
- int [GetSimCoreFitMaxStepCA](#) ()
- int [GetSimCoreSameChromosomesCount](#) ()
- double [GetSimCoreSameChromosomesGenerationAvarage](#) ()
- struct [stThreadCoreDataGA](#) * [GetSimCoreDataStruct](#) ()
- int [GetErrorFlag](#) ()
- void [ShowDebugConfigCore](#) ()

Private Slots

- void [ThreadCoreErrorSlot](#) (int err)
- void [ThreadCoreCheckCoreInitDone](#) ()
- void [ThreadCoreNewDataAvailable](#) ()
- void [ThreadCoreState](#) (int s)

Private Attributes

- [CConfigCore configCore](#)
config class
- bool [bCoreDataExpired](#)
are data from this class already displayed in gui?
- bool [bSimRunning](#)
should be evolution running?
- bool [bInitDone](#)
is init done?
- [CSpace * space](#)
pointer to ca space class
- [CGenome * genome](#)
pointer to genome class
- struct [stThreadCoreDataGA dataGa](#)
ga data struct used for propagating data from thread core into gui
- [QMutex mutexCore](#)
mutex used for sync gui thread (in which this class runs) with [CThreadCore](#) thread
- [QWaitCondition waitCCore](#)
wait cond used for sync gui thread (in which this class runs) with [CThreadCore](#) thread
- [CThreadCore * threadCore](#)
pointer to instance of thread core class
- int [iErrFlag](#)
error flag

3.4.1 Detailed Description

this is special class - is used as mediator between gui classes and [CThreadCore](#) - computation class this class solves thread sync between gui and core core send data from ga and ca to this class, this class then tell gui, that new data arrived and gui then copy this data and displays them, last step is, that gui tells this class, that data are displayed, this class send this info to core, so core will know, that can send new data

this class is created when "Init" button is pressed and deleted when "Delete" button is pressed when init button is pressed, this class creates [CThreadCore](#) and propagate sync pointers to it when delete is pressed this class kills compute core thread

3.4.2 Constructor & Destructor Documentation

3.4.2.1 CCore::CCore ()

class constructor

3.4.2.2 CCore::~~CCore ()

class destructor

3.4.3 Member Function Documentation

3.4.3.1 CConfigCore * CCore::GetConfigCore ()

return pointer to config class

3.4.3.2 int CCore::GetErrorFlag ()

returns erro flag

3.4.3.3 int CCore::GetSimCoreAncestorsCount ()

return ga data - genome's ancestor count

3.4.3.4 int CCore::GetSimCoreChromosomeID ()

return ga data - genome's id

3.4.3.5 struct stThreadCoreDataGA * CCore::GetSimCoreDataStruct () [read]

returns pointer to ga data struct

3.4.3.6 int CCore::GetSimCoreDifferentiationMin ()

return ga data - genome's minimum differentiation

3.4.3.7 int CCore::GetSimCoreFitMaxStepCA ()

return ga data - ca step with max fitness

3.4.3.8 int CCore::GetSimCoreFitnessMax ()

return ga data - genome's maximum fitness

3.4.3.9 double CCore::GetSimCoreFitnessMaxNorm ()

return ga data - genome's maximum fitness

3.4.3.10 int CCore::GetSimCoreGenerationID ()

return ga data - genome's generation id

3.4.3.11 int CCore::GetSimCoreSameChromosomesCount ()

return ga data - count of genomes with same fitness

3.4.3.12 double CCore::GetSimCoreSameChromosomesGenerationAvarage ()

return ga data - avarage generation of "same" genomes

3.4.3.13 CSpace * CCore::GetSpace ()

returns pointer to ca space class

3.4.3.14 void CCore::InitGuiCore ()

inits CCore and core thread CThreadCore propagates dat from this class into CThreadCore data from gui are propagated to this class from gui functions

3.4.3.15 bool CCore::IsInitDone ()

returns if init is done and ok

3.4.3.16 void CCore::SetCaRunTermination ()

sets state var which tells core, that user pressed button "Terminate" for terminating ca simulator

3.4.3.17 void CCore::SetCoreDataExpiration (bool e)

sets state var, which tells core, that data from this class was already displayed in gui, so new data can be propagete to this class

Parameters:

e are core's data expere? (already showed in gui)

3.4.3.18 void CCore::SetSimulationRunning (bool sr)

sets state var which tells core, that user pressed button "Evolve" so evolution can start

Parameters:

sr set ThreadCore thread running state

3.4.3.19 void CCore::ShowDebugConfigCore ()

debug INFO

3.4.3.20 void CCore::StartThreadCore ()

starts [CThreadCore](#) thread, just init part is compute, then thread waits for "Evolve" button to be pressed

3.4.3.21 void CCore::ThreadCoreCheckCoreInitDone () [private, slot]

propagates info about thread core init done into gui

3.4.3.22 void CCore::ThreadCoreErrorSlot (int *err*) [private, slot]

propagates error from core thread into gui

Parameters:

err error code from signal from core thread

3.4.3.23 void CCore::ThreadCoreNewDataAvailable () [private, slot]

tells gui, that new data from thread core are in this class

3.4.3.24 void CCore::ThreadCoreState (int *s*) [private, slot]

propagates state of thread core into gui

Parameters:

s core thread state

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

- Core.h
- Core.cpp

3.5 CCrossover Class Reference

```
#include <Crossover.h>
```

Public Member Functions

- [CCrossover](#) ()
- void [Crossover](#) ([CGenome](#) *g1, [CGenome](#) *g2)
- void [SetConfigCore](#) ([CConfigCore](#) *cs)
- int [GetErrorFlag](#) ()

Private Attributes

- [CConfigCore](#) * [pConfigCore](#)
pointer to config class
- [CRandom](#) [random](#)
instance of pseudo-random number generator class
- int [iErrFlag](#)
error flag

3.5.1 Detailed Description

crossover class for creating child genomes using crossover NOT IMPLEMENTED

3.5.2 Constructor & Destructor Documentation

3.5.2.1 CCrossover::CCrossover ()

class constructor

3.5.3 Member Function Documentation

3.5.3.1 void CCrossover::Crossover ([CGenome](#) * *g1*, [CGenome](#) * *g2*)

performs crossover of given genomes - NOT IMPLEMENTED

Parameters:

- **g1* pointer to 1st genome
- **g2* pointer to 2nd genome

3.5.3.2 int CCrossover::GetErrorFlag ()

returns error flag

3.5.3.3 void CCrossover::SetConfigCore (CConfigCore * cc)

sets config class

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

- Crossover.h
- Crossover.cpp

3.6 CExportCA Class Reference

```
#include <ExportCA.h>
```

Public Member Functions

- [CExportCA](#) ()
- void [ExportCellularAutomaton](#) (QString fileName, [CSpace](#) *space)
- void [ExportCellularAutomatonInput](#) (QString fileName, [CSpace](#) *space)
- void [SetConfigCore](#) ([CConfigCore](#) *cCore)
- int [GetErrorFlag](#) ()

Private Attributes

- [CConfigCore](#) * [config](#)
pointer to config class
- int [iErrFlag](#)
error flag

3.6.1 Detailed Description

this class exports ca space it has two modes - standard, which makes 10x zoom on exported images and input, which is used for exporting input ca space from gui settings tab, this mode does not used zoom, because input png should be imported into app later

3.6.2 Constructor & Destructor Documentation

3.6.2.1 CExportCA::CExportCA ()

class constructor

3.6.3 Member Function Documentation

3.6.3.1 void CExportCA::ExportCellularAutomaton (QString *fileName*, [CSpace](#) * *space*)

exports ca space, using 10x zoom effect, into png

Parameters:

fileName file path/name

**space* pointer to ca space class

3.6.3.2 void CExportCA::ExportCellularAutomatonInput (QString *fileName*, CSpace * *space*)

this function is used for export input ca space from settings input ca tab does not using zoom effect, because these images can be later imported into app

Parameters:

fileName file path/name

**space* pointer to ca space class

3.6.3.3 int CExportCA::GetErrorFlag ()

returns erro flag

3.6.3.4 void CExportCA::SetConfigCore (CConfigCore * *cCore*)

sets config class

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

- ExportCA.h
- ExportCA.cpp

3.7 CExportConfig Class Reference

```
#include <ExportConfig.h>
```

Public Member Functions

- [CExportConfig](#) ()
- [~CExportConfig](#) ()
- bool [ExportConfig](#) (QString filePathName, [CWidgetInput](#) *widIn, [CWidgetOutput](#) *widOut, [CWidgetEvolution](#) *widEvo, [CWidgetExport](#) *widExp)
- int [GetErrorFlag](#) ()

Private Member Functions

- void [WriteDocStart](#) ()
- void [WriteDocEnd](#) ()
- void [WriteSettingsInput](#) ([CWidgetInput](#) *widIn)
- void [WriteSettingsOutput](#) ([CWidgetOutput](#) *widOut)
- void [WriteSettingsEvolution](#) ([CWidgetEvolution](#) *widEvo)
- void [WriteSettingsExport](#) ([CWidgetExport](#) *widExp)
- void [WriteSettingsInputCASpace](#) ([CWidgetInput](#) *widIn)
- void [SetIODevice](#) ([QIODevice](#) *device)

Private Attributes

- [QXmlStreamWriter](#) [xml](#)
xml writer
- [QIODevice](#) * [device](#)
io device - opened file descriptor - for xml class
- int [iErrFlag](#)
error flag

3.7.1 Detailed Description

exports run config

3.7.2 Constructor & Destructor Documentation

3.7.2.1 CExportConfig::CExportConfig ()

class constructor

3.7.2.2 CExportConfig::~CExportConfig ()

class deconstructor

3.7.3 Member Function Documentation

3.7.3.1 **bool CExportConfig::ExportConfig (QString *filePathName*, CWidgetInput * *widIn*, CWidgetOutput * *widOut*, CWidgetEvolution * *widEvo*, CWidgetExport * *widExp*)**

exports actual app config

Parameters:

filePathName file's path/name
**widIn* pointer to settings widget input
**widOut* pointer to settings widget output
**widEvo* pointer to settings widget evolution
**widExp* pointer to settings widget export

3.7.3.2 **int CExportConfig::GetErrorFlag ()**

returns error flag

3.7.3.3 **void CExportConfig::SetIODevice (QIODevice * *device*) [private]**

sets IO device (opened file descriptor) into xml class

Parameters:

**device* opened file descriptor

3.7.3.4 **void CExportConfig::WriteDocEnd () [private]**

writes doc end (closing elements)

3.7.3.5 **void CExportConfig::WriteDocStart () [private]**

writes doc start doc type genome type

3.7.3.6 **void CExportConfig::WriteSettingsEvolution (CWidgetEvolution * *widEvo*) [private]**

writes evolution settings

Parameters:

**widEvo* pointer to evolution settings widget

3.7.3.7 **void CExportConfig::WriteSettingsExport (CWidgetExport * *widExp*) [private]**

writes export settings

Parameters:

**widExp* pointer to export settings widget

3.7.3.8 void CExportConfig::WriteSettingsInput (CWidgetInput * *widIn*) [private]

writes input settings

Parameters:

**widIn* pointer to input settings widget

3.7.3.9 void CExportConfig::WriteSettingsInputCASpace (CWidgetInput * *widIn*) [private]

writes input ca space settings

3.7.3.10 void CExportConfig::WriteSettingsOutput (CWidgetOutput * *widOut*) [private]

writes output settings

Parameters:

**widOut* pointer to output settings widget

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

- ExportConfig.h
- ExportConfig.cpp

3.8 CExportGA Class Reference

```
#include <ExportGA.h>
```

Public Member Functions

- [CExportGA](#) ()
- void [ExportGeneration](#) (QString fileName, std::vector< [CGenome](#) * > *gen)
- void [ExportChromosome](#) (QString fileName, [CGenome](#) *gen)
- void [SetConfigCore](#) (CConfigCore *cCore)
- int [GetErrorFlag](#) ()

Private Member Functions

- void [WriteDocStart](#) ()
- void [WriteDocEnd](#) ()
- void [WriteGenome](#) ([CGenome](#) *genome)
- void [SetIODevice](#) (QIODevice *device)

Private Attributes

- QDomStreamWriter [xml](#)
xml writer
- QIODevice * [device](#)
io device - opened file descriptor - for xml class
- CConfigCore * [config](#)
pointer to config class
- int [iErrFlag](#)
error flag

3.8.1 Detailed Description

this class exports ga genomes or whole population of genomes into xml files, which can be imported later into app by [CImportGA](#) class

3.8.2 Constructor & Destructor Documentation

3.8.2.1 CExportGA::CExportGA ()

class constructor

3.8.3 Member Function Documentation

3.8.3.1 void CExportGA::ExportChromosome (QString *fileName*, CGenome * *gen*)

exports one genome

Parameters:

fileName file path/name

**gen* pointer to genome

3.8.3.2 void CExportGA::ExportGeneration (QString *fileName*, std::vector< CGenome * > * *vecGen*)

exports one generation of whole population

Parameters:

fileName file path/name

**vecGen* pointer to population of genomes

3.8.3.3 int CExportGA::GetErrorFlag ()

returns error flag

3.8.3.4 void CExportGA::SetConfigCore (CConfigCore * *cCore*)

sets config class

3.8.3.5 void CExportGA::SetIODevice (QIODevice * *device*) [private]

sets IO device (opened file descriptor) into xml class

Parameters:

**device* opened file descriptor

3.8.3.6 void CExportGA::WriteDocEnd () [private]

writes doc end (closing elements)

3.8.3.7 void CExportGA::WriteDocStart () [private]

writes doc start doc type genome type

3.8.3.8 void CExportGA::WriteGenome (CGenome * *genome*) [private]

writes one genome into file

Parameters:

**genome* pointer to genome class to be exported

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

- ExportGA.h
- ExportGA.cpp

3.9 CExportLog Class Reference

```
#include <ExportLog.h>
```

Public Member Functions

- [CExportLog](#) ()
- [~CExportLog](#) ()
- void [SetFilePath](#) (QString path)
- void [SetFileName](#) (QString name)
- bool [OpenLogFile](#) ()
- void [CloseLogFile](#) ()
- void [WriteInfoLog](#) (QString info)
- void [WriteErrorLog](#) (int errId)
- void [WriteErrorLog](#) (QString error)
- void [ShowMessageBox](#) (int errId)
- void [ShowMessageBox](#) (QString error)

Private Attributes

- QString [filePath](#)
export path
- QString [fileName](#)
export file name
- QString [filePathName](#)
combination of previous two
- QFile [fileDevice](#)
file descriptor
- QTextStream [fileStream](#)
file writer
- bool [bFileOpened](#)
is file opened?

3.9.1 Detailed Description

this class exports logs from CORE part of application into txt file it is used just as ONE instance, created in [CThreadCore](#) (i hope there) and pointer to this instance is send to all core classes

is used for exporting error and warning strings into file or messagebox

3.9.2 Constructor & Destructor Documentation

3.9.2.1 CExportLog::CExportLog ()

class constructor

3.9.2.2 CExportLog::~~CExportLog ()

class destructor

3.9.3 Member Function Documentation

3.9.3.1 void CExportLog::CloseLogFile ()

closes file

3.9.3.2 bool CExportLog::OpenLogFile ()

opens log file

3.9.3.3 void CExportLog::SetFileName (QString *name*)

sets log file name

Parameters:

name log file name

3.9.3.4 void CExportLog::SetFilePath (QString *path*)

sets path of log file

Parameters:

path relative or absolute path

3.9.3.5 void CExportLog::ShowMessageBox (QString *error*)

shows message box with error

Parameters:

error string to be showed

3.9.3.6 void CExportLog::ShowMessageBox (int *errId*)

shows message box with error

Parameters:

errId error id (described in [Defines.h](#))

3.9.3.7 void CExportLog::WriteErrorLog (QString *error*)

writes ERROR into log file

Parameters:

error string to be write

3.9.3.8 void CExportLog::WriteErrorLog (int *errId*)

writes ERROR into log file

Parameters:

errId error id (described in [Defines.h](#))

3.9.3.9 void CExportLog::WriteInfoLog (QString *info*)

writes INFORMATION into log file

Parameters:

info string to be write

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

- ExportLog.h
- ExportLog.cpp

3.10 CFitness Class Reference

```
#include <Fitness.h>
```

Public Member Functions

- [CFitness](#) ()
- int [Fitness](#) ([CSpace](#) *spaceInit, [CSpace](#) *spaceAct)
- bool [IsLiveOrganismInSpace](#) ([CSpace](#) *s)
- void [IdentifyCorners](#) ([CSpace](#) *s)
- int [GetCornerPosX](#) (int corn)
- int [GetCornerPosY](#) (int corn)
- void [SetConfigCore](#) ([CConfigCore](#) *cc)
- int [GetErrorFlag](#) ()

Private Member Functions

- void [_MoveA](#) (int dir, int distance, int *iMoveDir_X, int *iMoveDir_Y)
- int [_Fitness_01](#) ([CSpace](#) *spaceInit, [CSpace](#) *spaceAct)
- int [_Fitness_02](#) ([CSpace](#) *spaceInit, [CSpace](#) *spaceAct)

Private Attributes

- [CConfigCore](#) * [pConfigCore](#)
pointer to config class
- int [iX_1](#)
x-coord upper left corner of object's rectangle envelope
- int [iY_1](#)
y-coord upper left corner of object's rectangle envelope
- int [iX_2](#)
x-coord lower right corner of object's rectangle envelope
- int [iY_2](#)
y-coord lower right corner of object's rectangle envelope
- bool [bNonEmptyCellExists](#)
tmp var used for object corner detection
- int [iErrFlag](#)
error flag

3.10.1 Detailed Description

this class computes base of fitness value of genome which is actually used for ca computations fitness is maximalized in [CGeneticAlgorithm](#) class in function Fitness, see comment in GeneticAlgorithm.cpp on line 250 (this line in time, when i am writing this comment...)

fitness computation is simple: this class just looks to INIT space, shifts object in this space in (user definated) direction to (user definated) distances, then it looks to ACT space and checks, which cells have right value. if cell have right value, it will give fitness +1 point, else 0

3.10.2 Constructor & Destructor Documentation

3.10.2.1 CFitness::CFitness ()

class constructor

3.10.3 Member Function Documentation

3.10.3.1 int CFitness::_Fitness_01 (CSPACE * *spaceInit*, CSPACE * *spaceAct*) [private]

this function calculates fitness - compares every cell in INIT space with ACT space of course, is expected that object from INIT is shifted in ACT in given direction to given distance (direction and distance given from GUI) function uses _MoveA to determine position of object in ACT relative to INIT maximum fitness is returned, when all cells in ACT have expected values - so, more cells have expected values, higher the fitness will be

Parameters:

- **spaceInit* pointer to init spce of ca
- **spaceAct* pointer to actual space of ca

3.10.3.2 int CFitness::_Fitness_02 (CSPACE * *spaceInit*, CSPACE * *spaceAct*) [private]

provides same functionality as _Fitness_01, but is able to cumpute with direction "ANY" and calculates with variable distances (from 1 to distance given as param in GUI)

Parameters:

- **spaceInit* pointer to init spce of ca
- **spaceAct* pointer to actual space of ca

3.10.3.3 void CFitness::_MoveA (int *dir*, int *distance*, int * *iMoveDir_X*, int * *iMoveDir_Y*) [private]

this function calculates relative position of cell, on which is cell expected to be, result is written into "pointer params" result is added to each cell position from INIT space, and then is checked, if cell in ACTUAL space aton newly computed position have expected value so, it calculates position in ACT relative to INIT

Parameters:

- dir* direction of movement

distance distance in "cell units" which object should go through

**iMoveDir_X* pointer to var which determines x coordinate, on which should cell be in actual step - result of this fc

**iMoveDir_Y* pointer to var which determines y coordinate, on which should cell be in actual step - result of this fc

3.10.3.4 int CFitness::Fitness (CSpace * *spaceInit*, CSpace * *spaceAct*)

calculates fitness of actual step of ca

Parameters:

**spaceInit* init ca space - stores original ca config

**spaceAct* actual ca space

3.10.3.5 int CFitness::GetCornerPosX (int *corn*)

returns coordinate of object corner on x-axes

Parameters:

corn specifies which corner should be returned

3.10.3.6 int CFitness::GetCornerPosY (int *corn*)

returns coordinate of object corner on y-axes

Parameters:

corn specifies which corner should be returned

3.10.3.7 int CFitness::GetErrorFlag ()

return error flag

3.10.3.8 void CFitness::IdentifyCorners (CSpace * *s*)

identifies corners of envelope of object in ca space - used for determinate if any object is in space object is covered by rectangle:

```
---*--- | *** | ** ** | ** * ** | ** ** | *** | ---*---
```

object is created from asterisks, white space are "dead cells", rectangle is created from lines

Parameters:

**s* pointer to ca space

3.10.3.9 bool CFitness::IsLiveOrganismInSpace (CSpace * *s*)

checks, if there is any live cell in ca space

Parameters:

**s* pointer to ca space

3.10.3.10 void CFitness::SetConfigCore (CConfigCore * *cc*)

sets config class

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

- Fitness.h
- Fitness.cpp

3.11 CGeneticAlgorithm Class Reference

```
#include <GeneticAlgorithm.h>
```

Public Member Functions

- [CGeneticAlgorithm](#) ([CConfigCore](#) *configCore)
- [~CGeneticAlgorithm](#) ()
- void [InitGenotype](#) ()
- void [InitGenotypeReevolve](#) ([CGenome](#) *genomeInit)
- void [ImportGenomeToPopulation](#) ([CGenome](#) *genomeImport)
- void [Fitness](#) ([CGenome](#) *genome, [CSpace](#) *spaceInit, [CSpace](#) *spaceAct, int caStepAct)
- void [CreateNextGeneration](#) ()
- void [CreateNextGeneration](#) (int genID)
- void [Selection](#) ()
- void [Crossover](#) ()
- void [Mutation](#) ()
- void [IdentifyCorners](#) ([CSpace](#) *s)
- int [GetCornerPosX](#) (int corn)
- int [GetCornerPosY](#) (int corn)
- void [SetConfigCore](#) ([CConfigCore](#) *ccc)
- [CGenome](#) * [GetGenome](#) (int id)
- [CGenome](#) * [GetBestGenome](#) ()
- std::vector< [CGenome](#) * > * [GetActualPopulation](#) ()
- int [GetActualPopulationSize](#) ()
- bool [ExistsGenomeId](#) (int id)
- int [GetErrorFlag](#) ()

Private Member Functions

- void [ClearVectors](#) ()
- void [RefillPopulation](#) ()
- void [RenumberActualGeneration](#) ()
- void [InitGenonetypeStandard](#) ()
- void [InitGenotypeInstruction](#) ()

Private Attributes

- [CConfigCore](#) * [pConfigCore](#)
pointer to config class
- [CFitness](#) [fitness](#)
instance of fitness class
- [CSelection](#) [select](#)
instance of selection class
- [CCrossover](#) [crossover](#)

instance of crossover class

- [CMutation](#) `mutate`

instance of mutation class

- [CRandom](#) `random`

instance of pseudo-random number generator class

- [CGenome](#) * `genomeTmp`

tmp pointer used for creating new genomes

- [CGenome](#) * `bestGenome`

pointer to best genome

- [CGenome](#) * `genomeErr`

pointer to error genome

- `std::vector< CGenome * >` `vecGenerationX`

vector which contains pointer for genomes

- `std::vector< CGenome * >` `vecGenerationY`

vector which contains pointer for genomes

- `std::vector< CGenome * > *` `vecGenerationAct`

act population vector - contains pointers to genomes from act generation

- `std::vector< CGenome * > *` `vecGenerationTmp`

vector used for saving pointer of genomes for next gen

- `std::vector< CGenome * > *` `vecGenerationTmpX`

just tmp var using for exchange pointers between act and tmp vectors

- `int` `iGenerationActID`

actual generation id

- `int` `iGenerationActSize`

actual generation size

- `int` `iErrFlag`

error flag

3.11.1 Detailed Description

main class of genetic algorithm - manages all ga computation using specialized classes for partial tasks - [CFitness](#), [CMutation](#), [CSelection](#), [CGenome](#)

3.11.2 Constructor & Destructor Documentation

3.11.2.1 CGeneticAlgorithm::CGeneticAlgorithm (CConfigCore * *configCore*)

class constructor

Parameters:

**configCore* pointer config class

3.11.2.2 CGeneticAlgorithm::~~CGeneticAlgorithm ()

class destructor

3.11.3 Member Function Documentation

3.11.3.1 void CGeneticAlgorithm::ClearVectors () [private]

clears vectors which carrying actual generation and tmp (next) gen

3.11.3.2 void CGeneticAlgorithm::CreateNextGeneration (int *genID*)

creates new generation of population of genomes SELECTs genomes to next gen RE-FILLS population from selected genomes MUTATE all genomes (with exception of best one, explained in Mutation fc) RE-NUMBER population - sets new generation id and genome id

CROSSOVER is NOT USED

Parameters:

genID sets generation id to given value

3.11.3.3 void CGeneticAlgorithm::CreateNextGeneration ()

same is previous fc

3.11.3.4 void CGeneticAlgorithm::Crossover ()

performs crossover - NOT IMPLEMENTED

3.11.3.5 bool CGeneticAlgorithm::ExistsGenomeId (int *id*)

checks if genome with given id exists

Parameters:

id id of genome in actual population

3.11.3.6 void CGeneticAlgorithm::Fitness (CGenome * *genome*, CSpace * *spaceInit*, CSpace * *spaceAct*, int *caStepAct*)

computes fitness using [CFitness](#) class, this function checks if actual step of ca is final step in which object shifting should be done if this is final step, this function maximalizes fitness of given genome in case, that this genome is "good" (has relative fitness higher than 0.9) if this is not final step, but fitness is "good", it also maximalizes fitness but the maximalization is smaller in comparison to final step case

Parameters:

- **genome* pointer to actually used genome
- **spaceInit* pointer to init space of ca needed by [CFitness](#) class
- **spaceAct* pointer to actual space of ca
- caStepAct* actual step of ca

3.11.3.7 std::vector< CGenome * > * CGeneticAlgorithm::GetActualPopulation ()

returns pointer to vector with actual generation

3.11.3.8 int CGeneticAlgorithm::GetActualPopulationSize ()

returns size of actual generation

3.11.3.9 CGenome * CGeneticAlgorithm::GetBestGenome ()

returns pointer to best genome

3.11.3.10 int CGeneticAlgorithm::GetCornerPosX (int *corn*)

returns object's x-position of selected corner

3.11.3.11 int CGeneticAlgorithm::GetCornerPosY (int *corn*)

returns object's y-position of selected corner

3.11.3.12 int CGeneticAlgorithm::GetErrorFlag ()

returns error flag

3.11.3.13 CGenome * CGeneticAlgorithm::GetGenome (int *id*)

returns pointer to genome with given id

Parameters:

- id* id of genome from actual population

3.11.3.14 void CGeneticAlgorithm::IdentifyCorners (CSpace * s)

finds corners of object in ca space using [CFitness](#) class

Parameters:

*s pointer to ca space

3.11.3.15 void CGeneticAlgorithm::ImportGenomeToPopulation (CGenome * genomeImport)

imports given genome in actual population

Parameters:

*genomeImport genome which will be imported into actual gen

3.11.3.16 void CGeneticAlgorithm::InitGenonetypeStandard () [private]

creates init population of genomes

3.11.3.17 void CGeneticAlgorithm::InitGenotype ()

creates init population of genomes

3.11.3.18 void CGeneticAlgorithm::InitGenotypeInstruction () [private]

inits population with instruction genomes - NOT IMPLEMENTED

3.11.3.19 void CGeneticAlgorithm::InitGenotypeReevolve (CGenome * genomeInit)

creates init population of genomes from given genome

Parameters:

*genomeInit init genome which should be re-evolved

3.11.3.20 void CGeneticAlgorithm::Mutation ()

performs mutation using CMutate class best genome from population is selected twice - one copy which is on 1st position of population vector is NOT MUTATED - ALGORITHM USING ELITISM !!!

3.11.3.21 void CGeneticAlgorithm::RefillPopulation () [private]

refills population - after selection is important to create new genomes for refilling population to init size, because no croosover to creation child genomes is used population is refilled from genomes selected into next gen

3.11.3.22 void CGeneticAlgorithm::RenumberActualGeneration () [private]

renumbers genomes in population - sets actual generation id and genome id, which depends in index of genome in vector

3.11.3.23 void CGeneticAlgorithm::Selection ()

performs selection using [CSelection](#) class

3.11.3.24 void CGeneticAlgorithm::SetConfigCore (CConfigCore * ccc)

sets config class

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

- GeneticAlgorithm.h
- GeneticAlgorithm.cpp

3.12 CGenome Class Reference

```
#include <Genome.h>
```

Public Member Functions

- [CGenome](#) (int typeG=GENOME_TYPE_2_9N, int initGenerationId=-1, int initGenomeId=-1, [CConfigCore](#) *pCC=NULL)
- [CGenome](#) (const [CGenome](#) &rSide, int mode=GENOME_CONSTRUCT_COPY_MODE_NEXT_GEN, [CConfigCore](#) *pCC=NULL)
- [~CGenome](#) ()
- void [SetGene](#) (int index, BYTE gene)
- BYTE [GetGene](#) (int index)
- [CGenomeType2_Ins](#) * [GetInstructionGenome](#) ()
- void [SetFitness](#) (int f, int d, int caStep)
- void [SetFitness](#) (int f)
- int [GetFitness](#) ()
- int [GetFitnessMax](#) ()
- int [GetDifferentiationMin](#) ()
- int [GetFitnessMaxStepCA](#) ()
- void [SetFitnessNorm](#) (double fn)
- double [GetFitnessNorm](#) ()
- double [GetFitnessMaxNorm](#) ()
- int [GetGenomeType](#) ()
- void [SetAncestorsCount](#) (int anc)
- int [GetAncestorsCount](#) ()
- void [SetMutatedGenes](#) (int mgc)
- int [GetMutatedGenes](#) ()
- void [SetMutatedGenesTotal](#) (int thisG, int ancestorG)
- int [GetMutatedGenesTotal](#) ()
- int [GetMutatedGenesAllAncestors](#) ()
- void [SetThisGenomeId](#) (int generationId, int genomeId)
- int [GetThisGenerationId](#) ()
- int [GetThisGenomeId](#) ()
- void [SetThisInitGenomeId](#) (int generationInitId, int genomeInitId)
- int [GetThisInitGenerationId](#) ()
- int [GetThisInitGenomeId](#) ()
- void [SetParentGenomeId](#) (int generationId, int genomeId)
- int [GetParentGenerationId](#) ()
- int [GetParentGenomeId](#) ()
- int [GetErrorFlag](#) ()

Private Attributes

- [CGenomeType2](#) * [genomeT2](#)
pointer to instance of 9-neighborhood 2 state genome class
- [CGenomeType3](#) * [genomeT3](#)
pointer to instance of 9-neighborhood 3 state genome class

- [CGenomeType4 * genomeT4](#)
pointer to instance of 9-neighborhood 4 state genome class
- [CGenomeType2_Nbh5 * genomeT2_N5](#)
pointer to instance of 5-neighborhood 2 state genome class
- [CGenomeType3_Nbh5 * genomeT3_N5](#)
pointer to instance of 5-neighborhood 3 state genome class
- [CGenomeType4_Nbh5 * genomeT4_N5](#)
pointer to instance of 5-neighborhood 4 state genome class
- [CGenomeType2_Ins * genomeT2_Ins](#)
pointer to instance of instruction 2 state genome class
- [CConfigCore * pConfigCore](#)
pointer to config class
- [int iGenomeType](#)
type of genome
- [int iFitness](#)
fitness in actual step of ca
- [int iDifferentiation](#)
difference between perfect solution and actual solution
- [int iFitnessMax](#)
maximum fitness which this genome was able to obtain in act ga generation
- [int iDifferentiationMin](#)
maximum fitness = minimum differentiation
- [int iFitMaxStepCA](#)
step of ca in which max fitness was obtained
- [double dFitnessNormalized](#)
normalized fitness
- [double dFitnessMaxNormalized](#)
normalized max fitness
- [int iGenesMutated](#)
mutated genes of this genome in act generation
- [int iGenesMutatedTotal](#)
mutated genes of this genome during it's life time
- [int iGenesMutatedAllAncestors](#)

mutated genes of all ancestors of this genome

- int **iAncestorsCount**
ancestor count
- int **iThisGenerationId**
id of actual generation
- int **iThisGenomeId**
id of this genome in actual generation
- int **iThisInitGenerationId**
id of generation in which this genome was created
- int **iThisInitGenomeId**
id of this genome in it's init generation
- int **iParentGenerationId**
id of parent genome generation
- int **iParentGenomeId**
id of parent genome in it's generation
- int **iErrFlag**

3.12.1 Detailed Description

this class contains genome of GA using classes from ./genome/ folder - this classes contain genome itself, this class just make up-level interface for them

3.12.2 Constructor & Destructor Documentation

3.12.2.1 CGenome::CGenome (int *typeG* = GENOME_TYPE_2_9N, int *initGenerationId* = -1, int *initGenomeId* = -1, CConfigCore * *pCC* = NULL)

class constructor

Parameters:

- typeG* genome type
- initGenerationId* id of generation, in which this genome was created
- initGenomeId* id of genome
- *pCC* pointer to config class

3.12.2.2 CGenome::CGenome (const CGenome & *rSide*, int *mode* = GENOME_CONSTRUCT_COPY_MODE_NEXT_GEN, CConfigCore **pCC* = NULL)

class constructor

Parameters:

&rSide reference to genome class at right side of "="

mode class copy mode - DEEP or NEXT-GEN

**pCC* pointer to config class

3.12.2.3 CGenome::~~CGenome ()

class destructor

3.12.3 Member Function Documentation

3.12.3.1 int CGenome::GetAncestorsCount ()

return count of ancestor genomes

3.12.3.2 int CGenome::GetDifferentiationMin ()

returns minimum differentiation, which was reached during all steps of ca

3.12.3.3 int CGenome::GetErrorFlag ()

return error flag

3.12.3.4 int CGenome::GetFitness ()

return fitness of actual step of ca

3.12.3.5 int CGenome::GetFitnessMax ()

return maximum fitness, which was reached during all steps of ca

3.12.3.6 double CGenome::GetFitnessMaxNorm ()

returns maximum mormalized fitness which was reached during all steps of ca

3.12.3.7 int CGenome::GetFitnessMaxStepCA ()

return step of ca, in which max fitness was reached

3.12.3.8 double CGenome::GetFitnessNorm ()

returns formalized fitness

3.12.3.9 BYTE CGenome::GetGene (int *index*)

return value of gene at given index, if index is out of range default value will be returned

Parameters:

index index of gene

3.12.3.10 int CGenome::GetGenomeType ()

return type of genome

3.12.3.11 CGenomeType2_Ins * CGenome::GetInstructionGenome ()

returns whole instruction genome

3.12.3.12 int CGenome::GetMutatedGenes ()

return count of mutated genes from actual generation of ga

3.12.3.13 int CGenome::GetMutatedGenesAllAncestors ()

return total mutated genes of all ancestor genomes

3.12.3.14 int CGenome::GetMutatedGenesTotal ()

returns total mutated genes of this genome

3.12.3.15 int CGenome::GetParentGenerationId ()

returns parent genome generation id

3.12.3.16 int CGenome::GetParentGenomeId ()

returns parent id

3.12.3.17 int CGenome::GetThisGenerationId ()

return this genome actual generation id

3.12.3.18 int CGenome::GetThisGenomeId ()

returns this genome id in actual generation

3.12.3.19 int CGenome::GetThisInitGenerationId ()

return this genome init generation id

3.12.3.20 int CGenome::GetThisInitGenomeId ()

returns this genome id in init generation

3.12.3.21 void CGenome::SetAncestorsCount (int *anc*)

sets count of ancestor genomes

Parameters:

anc ancestors count

3.12.3.22 void CGenome::SetFitness (int *f*)

sets fitness

Parameters:

f fitness

3.12.3.23 void CGenome::SetFitness (int *f*, int *d*, int *caStep*)

sets fitness of this genome

Parameters:

f fitness

caStep step of ca, in which this fitness was reached

3.12.3.24 void CGenome::SetFitnessNorm (double *fn*)

sets normalized fitness

Parameters:

fn normalized fitness

3.12.3.25 void CGenome::SetGene (int *index*, BYTE *gene*)

sets gene of genome to given value

Parameters:

index index of gene in genome

gene value of gene

3.12.3.26 void CGenome::SetMutatedGenes (int *mgc*)

sets count of mutated genes in actual generation of ga

Parameters:

mgc mutated genes count

3.12.3.27 void CGenome::SetMutatedGenesTotal (int *thisG*, int *ancestorG*)

sets total count of mutated genes - this genome + ancestors

Parameters:

thisG mutated genes of this genome

ancestorsG mutated genes of all ancestors

3.12.3.28 void CGenome::SetParentGenomeId (int *generationId*, int *genomeId*)

sets parent genome id and generation id - generation in which parent of this genome was created

Parameters:

generationId id of PARENT generation

genomeId id of genome in PARENT generation

3.12.3.29 void CGenome::SetThisGenomeId (int *generationId*, int *genomeId*)

sets this genome actual id and actual generation id - generation in which this genome was created

Parameters:

generationId id of ACTUAL generation

genomeId id of genome in ACTUAL generation

3.12.3.30 void CGenome::SetThisInitGenomeId (int *generationInitId*, int *genomeInitId*)

sets this genome init id and init generation id - generation in which this genome was created

Parameters:

generationInitId id of INIT generation

genomeInitId id of genome in INIT generation

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

- Genome.h
- Genome.cpp

3.13 CGenomeType2 Class Reference

```
#include <GenomeType2.h>
```

Public Member Functions

- [CGenomeType2](#) ()
- [CGenomeType2](#) (const [CGenomeType2](#) &rSide)
- void [SetGene](#) (int index, BYTE gene)
- BYTE [GetGene](#) (int index)
- BYTE * [GetGenome](#) ()

Private Attributes

- BYTE [genome](#) [GENOME_SIZE_TYPE_2_9N]
linear chromosome

3.13.1 Detailed Description

this class contains genome for 9-neighborhood 2 states ca genome is linear array of BYTES, every element of array represents one gene index to genome is calculated from neighbor cells, see class [CTFunction](#), function CalculateIndexTorus in ../ca/TFunction.cpp file, line 280 when index to genome is calculated, gene from this index is returned to CA - this gene represents new value of cell in next step of ca computation so this genome is built from values CELL_EMPTY and CELL_LIVE_1, which are only allowed values of ca space

3.13.2 Constructor & Destructor Documentation

3.13.2.1 CGenomeType2::CGenomeType2 ()

class constructor

3.13.2.2 CGenomeType2::CGenomeType2 (const CGenomeType2 & rSide)

class constructor

Parameters:

&rSide reference to class on right side of "="

3.13.3 Member Function Documentation

3.13.3.1 BYTE CGenomeType2::GetGene (int *index*)

return value of gene from given index

Parameters:

index index to genome

3.13.3.2 BYTE * CGenomeType2::GetGenome ()

return pointer to whole genome

3.13.3.3 void CGenomeType2::SetGene (int *index*, BYTE *gene*)

sets gene at given index of genome

Parameters:

index index to genome

gene value of gene

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

- GenomeType2.h
- GenomeType2.cpp

3.14 CGenomeType2_Ins Class Reference

```
#include <GenomeType2_Ins.h>
```

Public Member Functions

- [CGenomeType2_Ins](#) ()
- [CGenomeType2_Ins](#) (const [CGenomeType2_Ins](#) &rSide)
- struct [stGeneInstruction](#) * [GetGene](#) (int index)
- void [SetInstruction](#) (int index, BYTE instruction)
- BYTE [GetInstruction](#) (int index)
- void [SetPreCondition](#) (int index, BYTE preCon)
- BYTE [GetPreCondition](#) (int index)
- void [SetPreConditionBit](#) (int index, int bitIndex, bool preCon)
- bool [GetPreConditionBit](#) (int index, int bitIndex)
- void [SetPreConditionLogic](#) (int index, BYTE preConLog)
- BYTE [GetPreConditionLogic](#) (int index)
- void [SetPreConditionLogicBit](#) (int index, int bitIndex, bool preConLog)
- bool [GetPreConditionLogicBit](#) (int index, int bitIndex)
- void [SetPostCondition](#) (int index, BYTE postCon)
- BYTE [GetPostCondition](#) (int index)

Private Member Functions

- void [SetBitValue](#) (BYTE *field, int index, bool value)
- BYTE [GetBitValue](#) (BYTE field, int index)

Private Attributes

- struct [stGeneInstruction](#) [genome](#) [GENOME_INS__COUNT]
chromosome
- struct [stGeneInstruction](#) [errGene](#)
error gene

3.14.1 Detailed Description

this class carry instruction-based genome for 2-state ca chromosome (genome) is created from 4-BYTE (char) structs struct contains instruction type (NOP or IF), precondition - precondition defines combination of cells in neighborhood, which should be active, one precon bit can define more cells preConLogic defines clutches (???) AND or OR between preCon bits postcon defines value of cell, if instruction (gene) can be implemented on this cell

3.14.2 Constructor & Destructor Documentation

3.14.2.1 CGenomeType2_Ins::CGenomeType2_Ins ()

class constructor

3.14.2.2 CGenomeType2_Ins::CGenomeType2_Ins (const CGenomeType2_Ins & *rSide*)

class constructor

Parameters:

&rSide reference to class on right side of "="

3.14.3 Member Function Documentation

3.14.3.1 BYTE CGenomeType2_Ins::GetBitValue (BYTE *field*, int *index*) [private]

return bit of given BYTE field at given bitIndex

Parameters:

field BYTE field of some gene from genome

index index to bit of field

3.14.3.2 struct stGeneInstruction * CGenomeType2_Ins::GetGene (int *index*) [read]

returns gene from given position of genome

Parameters:

index index to gene from genome

3.14.3.3 BYTE CGenomeType2_Ins::GetInstruction (int *index*)

returns instruction type of gene at given index of genome

Parameters:

index index to gene from genome

3.14.3.4 BYTE CGenomeType2_Ins::GetPostCondition (int *index*)

returns postcondition of gene at given index

Parameters:

index index to gene from genome

3.14.3.5 BYTE CGenomeType2_Ins::GetPreCondition (int *index*)

returns instruction type of gene at given index of genome

Parameters:

index index to gene from genome

3.14.3.6 **bool CGenomeType2_Ins::GetPreConditionBit (int *index*, int *bitIndex*)**

returns one bit of precondition of gene at given index

Parameters:

index index to gene from genome
bitIndex index of bit of precondition

3.14.3.7 **BYTE CGenomeType2_Ins::GetPreConditionLogic (int *index*)**

returns precondition logic - and OR or between bits of precondition

Parameters:

index index to gene from genome

3.14.3.8 **bool CGenomeType2_Ins::GetPreConditionLogicBit (int *index*, int *bitIndex*)**

returns one bit of precondition logic of gene at given index

Parameters:

index index to gene from genome
bitIndex index of bit of precondition logic

3.14.3.9 **void CGenomeType2_Ins::SetBitValue (BYTE **field*, int *index*, bool *value*)** **[private]**

sets bit of given BYTE field at given bitIndex to given value

Parameters:

**field* pointer to BYTE field of some gene from genome
index index to bit of field
value value of bit

3.14.3.10 **void CGenomeType2_Ins::SetInstruction (int *index*, BYTE *instruction*)**

sets instruction type of gene at given index of genome

Parameters:

index index to gene from genome
instruction instruction type

3.14.3.11 void CGenomeType2_Ins::SetPostCondition (int *index*, BYTE *postCon*)

sets postcondition of gene at given index

Parameters:

index index to gene from genome
postCon postcondition

3.14.3.12 void CGenomeType2_Ins::SetPreCondition (int *index*, BYTE *preCon*)

sets precondition type of gene at given index of genome

Parameters:

index index to gene from genome
preCon precondition

3.14.3.13 void CGenomeType2_Ins::SetPreConditionBit (int *index*, int *bitIndex*, bool *preCon*)

sets one bit of precondition of gene at given index

Parameters:

index index to gene from genome
bitIndex index of bit of precondition
preCon value of bit

3.14.3.14 void CGenomeType2_Ins::SetPreConditionLogic (int *index*, BYTE *preConLog*)

sets precondition logic - and OR or between bits of precondition

Parameters:

index index to gene from genome
preConLog precondition logic

3.14.3.15 void CGenomeType2_Ins::SetPreConditionLogicBit (int *index*, int *bitIndex*, bool *preConLog*)

sets one bit of precondition logic of gene at given index

Parameters:

index index to gene from genome
bitIndex index of bit of precondition logic
preConLog value of bit

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

- GenomeType2_Ins.h
- GenomeType2_Ins.cpp

3.15 CGenomeType2_Nbh5 Class Reference

```
#include <GenomeType2_Nbh5.h>
```

Public Member Functions

- [CGenomeType2_Nbh5](#) ()
- [CGenomeType2_Nbh5](#) (const [CGenomeType2_Nbh5](#) &rSide)
- void [SetGene](#) (int index, BYTE gene)
- BYTE [GetGene](#) (int index)
- BYTE * [GetGenome](#) ()

Private Attributes

- BYTE [genome](#) [GENOME_SIZE_TYPE_2_5N]
linear chromosome

3.15.1 Detailed Description

this class contains genome for 5-neighborhood 2 states ca genome is linear array of BYTES, every element of array represents one gene index to genome is calculated from neighbor cells, see class [CTFunction](#), function CalculateIndexTorus in ../ca/TFunction.cpp file, line 280 when index to genome is calculated, gene from this index is returned to CA - this gene represents new value of cell in next step of ca computation so this genome is built from values CELL_EMPTY and CELL_LIVE_1, which are only allowed values of ca space

3.15.2 Constructor & Destructor Documentation

3.15.2.1 CGenomeType2_Nbh5::CGenomeType2_Nbh5 ()

class constructor

3.15.2.2 CGenomeType2_Nbh5::CGenomeType2_Nbh5 (const CGenomeType2_Nbh5 &rSide)

class constructor

Parameters:

&rSide reference to class on right side of "="

3.15.3 Member Function Documentation

3.15.3.1 BYTE CGenomeType2_Nbh5::GetGene (int *index*)

return value of gene from given index

Parameters:

index index to genome

3.15.3.2 `BYTE * CGenomeType2_Nbh5::GetGenome ()`

return pointer to whole genome

3.15.3.3 `void CGenomeType2_Nbh5::SetGene (int index, BYTE gene)`

sets gene at given index of genome

Parameters:

index index to genome

gene value of gene

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

- GenomeType2_Nbh5.h
- GenomeType2_Nbh5.cpp

3.16 CGenomeType3 Class Reference

```
#include <GenomeType3.h>
```

Public Member Functions

- [CGenomeType3](#) ()
- [CGenomeType3](#) (const [CGenomeType3](#) &rSide)
- void [SetGene](#) (int index, BYTE gene)
- BYTE [GetGene](#) (int index)

Private Attributes

- std::bitset< GENOME_SIZE_BITS_TYPE_3_9N > [genome](#)
linear chromosome

3.16.1 Detailed Description

this class contains genome for 5-neighborhood 3 states ca genome is created from bitset template class, every 2 elements of bitset represents one gene (2 bits are needed for int value "2") index to genome is calculated from neighbor cells, see class [CTFunction](#), function CalculateIndexTorus in ../../ca/TFunction.cpp file, line 280 when index to genome is calculated, gene from this index is returned to CA - this gene represents new value of cell in next step of ca computation so this genome is built from values CELL_EMPTY and CELL_LIVE_1, CELL_LIVE_2 which are only allowed values of ca space

3.16.2 Constructor & Destructor Documentation

3.16.2.1 CGenomeType3::CGenomeType3 ()

class constructor

3.16.2.2 CGenomeType3::CGenomeType3 (const CGenomeType3 & rSide)

class constructor

Parameters:

&rSide reference to class on right side of "="

3.16.3 Member Function Documentation

3.16.3.1 BYTE CGenomeType3::GetGene (int index)

return value of gene from given index

Parameters:

index index to genome

3.16.3.2 void CGenomeType3::SetGene (int *index*, BYTE *gene*)

sets gene at given index of genome

Parameters:

index index to genome

gene value of gene

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

- GenomeType3.h
- GenomeType3.cpp

3.17 CGenomeType3_Nbh5 Class Reference

```
#include <GenomeType3_Nbh5.h>
```

Public Member Functions

- [CGenomeType3_Nbh5](#) ()
- [CGenomeType3_Nbh5](#) (const [CGenomeType3_Nbh5](#) &rSide)
- void [SetGene](#) (int index, BYTE gene)
- BYTE [GetGene](#) (int index)
- BYTE * [GetGenome](#) ()

Private Attributes

- BYTE [genome](#) [GENOME_SIZE_TYPE_3_5N]
linear chromosome

3.17.1 Detailed Description

this class contains genome for 5-neighborhood 3 states ca genome is linear array of BYTES, every element of array represents one gene index to genome is calculated from neighbor cells, see class [CTFunction](#), function CalculateIndexTorus in ../ca/TFunction.cpp file, line 280 when index to genome is calculated, gene from this index is returned to CA - this gene represents new value of cell in next step of ca computation so this genome is built from values CELL_EMPTY and CELL_LIVE_1, CELL_LIVE_2 which are only allowed values of ca space

3.17.2 Constructor & Destructor Documentation

3.17.2.1 CGenomeType3_Nbh5::CGenomeType3_Nbh5 ()

class constructor

3.17.2.2 CGenomeType3_Nbh5::CGenomeType3_Nbh5 (const CGenomeType3_Nbh5 &rSide)

class constructor

Parameters:

&rSide reference to class on right side of "="

3.17.3 Member Function Documentation

3.17.3.1 BYTE CGenomeType3_Nbh5::GetGene (int *index*)

return value of gene from given index

Parameters:

index index to genome

3.17.3.2 `BYTE * CGenomeType3_Nbh5::GetGenome ()`

return pointer to whole genome

3.17.3.3 `void CGenomeType3_Nbh5::SetGene (int index, BYTE gene)`

sets gene at given index of genome

Parameters:

index index to genome

gene value of gene

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

- GenomeType3_Nbh5.h
- GenomeType3_Nbh5.cpp

3.18 CGenomeType4 Class Reference

```
#include <GenomeType4.h>
```

Public Member Functions

- [CGenomeType4](#) ()
- [CGenomeType4](#) (const [CGenomeType4](#) &rSide)
- void [SetGene](#) (int index, BYTE gene)
- BYTE [GetGene](#) (int index)

Private Attributes

- std::bitset< GENOME_SIZE_BITS_TYPE_4_9N > [genome](#)
linear chromosome

3.18.1 Detailed Description

this class contains genome for 5-neighborhood 4 states ca genome is created from bitset template class, every 2 elements of bitset represents one gene (2 bits are needed for int value "2" and "3") index to genome is calculated from neighbor cells, see class [CTFunction](#), function CalculateIndexTorus in `../ca/TFunction.cpp` file, line 280 when index to genome is calculated, gene from this index is returned to CA - this gene represents new value of cell in next step of ca computation so this genome is built from values CELL_EMPTY and CELL_LIVE_1, CELL_LIVE_2 and CELL_LIVE_3 which are only allowed values of ca space

3.18.2 Constructor & Destructor Documentation

3.18.2.1 CGenomeType4::CGenomeType4 ()

class constructor

3.18.2.2 CGenomeType4::CGenomeType4 (const CGenomeType4 & rSide)

class constructor

Parameters:

&rSide reference to class on right side of "="

3.18.3 Member Function Documentation

3.18.3.1 BYTE CGenomeType4::GetGene (int *index*)

return value of gene from given index

Parameters:

index index to genome

3.18.3.2 void CGenomeType4::SetGene (int *index*, BYTE *gene*)

sets gene at given index of genome

Parameters:

index index to genome

gene value of gene

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

- GenomeType4.h
- GenomeType4.cpp

3.19 CGenomeType4_Nbh5 Class Reference

```
#include <GenomeType4_Nbh5.h>
```

Public Member Functions

- [CGenomeType4_Nbh5](#) ()
- [CGenomeType4_Nbh5](#) (const [CGenomeType4_Nbh5](#) &rSide)
- void [SetGene](#) (int index, BYTE gene)
- BYTE [GetGene](#) (int index)
- BYTE * [GetGenome](#) ()

Private Attributes

- BYTE [genome](#) [GENOME_SIZE_TYPE_4_5N]
linear chromosome

3.19.1 Detailed Description

this class contains genome for 5-neighborhood 4 states ca genome is linear array of BYTES, every element of array represents one gene index to genome is calculated from neighbor cells, see class [CTFunction](#), function CalculateIndexTorus in ../ca/TFunction.cpp file, line 280 when index to genome is calculated, gene from this index is returned to CA - this gene represents new value of cell in next step of ca computation so this genome is built from values CELL_EMPTY and CELL_LIVE_1, CELL_LIVE_2 and CELL_LIVE_3 which are only allowed values of ca space

3.19.2 Constructor & Destructor Documentation

3.19.2.1 CGenomeType4_Nbh5::CGenomeType4_Nbh5 ()

class constructor

3.19.2.2 CGenomeType4_Nbh5::CGenomeType4_Nbh5 (const CGenomeType4_Nbh5 & rSide)

class constructor

Parameters:

&rSide reference to class on right side of "="

3.19.3 Member Function Documentation

3.19.3.1 BYTE CGenomeType4_Nbh5::GetGene (int *index*)

return value of gene from given index

Parameters:

index index to genome

3.19.3.2 `BYTE * CGenomeType4_Nbh5::GetGenome ()`

return pointer to whole genome

3.19.3.3 `void CGenomeType4_Nbh5::SetGene (int index, BYTE gene)`

sets gene at given index of genome

Parameters:

index index to genome

gene value of gene

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

- GenomeType4_Nbh5.h
- GenomeType4_Nbh5.cpp

3.20 CGraphicsItem Class Reference

```
#include <GraphicsItem.h>
```

Public Member Functions

- [CGraphicsItem](#) ()
- [CGraphicsItem](#) (int posX, int posY, int states, BYTE defState)
- [CGraphicsItem](#) (int posX, int posY, [CGraphicsItemConfig](#) *c)
- [QRectF](#) [boundingRect](#) () const
- void [paint](#) (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)
- void [SetState](#) (BYTE s)
- BYTE [GetState](#) ()
- void [SetEditable](#) (bool b)
- bool [IsEditable](#) ()
- int [GetPosX](#) ()
- int [GetPosY](#) ()
- int [GetStatesCount](#) ()
- void [SetStatesCount](#) (int sc)

Protected Member Functions

- void [mouseMoveEvent](#) (QGraphicsSceneMouseEvent *event)
- void [mousePressEvent](#) (QGraphicsSceneMouseEvent *event)
- void [mouseReleaseEvent](#) (QGraphicsSceneMouseEvent *event)

Private Attributes

- unsigned int [iGridPosX](#)
x coord of cell in space
- unsigned int [iGridPosY](#)
y coord of cell in space
- BYTE [byState](#)
current state of cell
- bool [bEditable](#)
is cell editable/clickable?
- int [iStatesCount](#)
states count
- [CGraphicsItemConfig](#) * [config](#)
pointer to cells' config class
- QColor [color](#)
color of cell in gui

3.20.1 Detailed Description

class represents one cell in gui space

3.20.2 Constructor & Destructor Documentation

3.20.2.1 CGraphicsItem::CGraphicsItem ()

class constructor

3.20.2.2 CGraphicsItem::CGraphicsItem (int *posX*, int *posY*, int *states*, BYTE *defState*)

class constructor

Parameters:

posX x coord of cell

posY y coord of cell

states states count

defState default state

3.20.2.3 CGraphicsItem::CGraphicsItem (int *posX*, int *posY*, CGraphicsItemConfig * *c*)

class constructor

Parameters:

posX x coord of cell

posY y coord of cell

**c* pointer to [CGraphicsItemConfig](#) class

3.20.3 Member Function Documentation

3.20.3.1 QRectF CGraphicsItem::boundingRect () const

returns rectangle which represents a cell

3.20.3.2 int CGraphicsItem::GetPosX ()

returns cell's x coord

3.20.3.3 int CGraphicsItem::GetPosY ()

returns cell's y coord

3.20.3.4 BYTE CGraphicsItem::GetState ()

returns cell state

3.20.3.5 int CGraphicsItem::GetStatesCount ()

returns states count

3.20.3.6 bool CGraphicsItem::IsEditable ()

returns if cell is editable

**3.20.3.7 void CGraphicsItem::mouseMoveEvent (QGraphicsSceneMouseEvent * *event*)
[protected]**

reaction on mouse move

Parameters:

event mouse event

**3.20.3.8 void CGraphicsItem::mousePressEvent (QGraphicsSceneMouseEvent * *event*)
[protected]**

reaction on mouse button press event

Parameters:

event mouse event

**3.20.3.9 void CGraphicsItem::mouseReleaseEvent (QGraphicsSceneMouseEvent * *event*)
[protected]**

reaction on mouse button release event

Parameters:

event mouse event

**3.20.3.10 void CGraphicsItem::paint (QPainter * *painter*, const QStyleOptionGraphicsItem *
option, QWidget * *widget*)**

draws cell

Parameters:

**painter* pointer to QPainter class

**option* pointet to QStyleOptionGraphicsItem class

**widget* pointer to parent widget class

3.20.3.11 void CGraphicsItem::SetEditable (bool *b*)

sets if cell is editable - clickable

Parameters:

b is editable?

3.20.3.12 void CGraphicsItem::SetState (BYTE *s*)

sets cell state

Parameters:

s new state

3.20.3.13 void CGraphicsItem::SetStatesCount (int *sc*)

sets states count

Parameters:

sc states count

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

- GraphicsItem.h
- GraphicsItem.cpp

3.21 CGraphicsItemConfig Class Reference

```
#include <GraphicsItemConfig.h>
```

Public Member Functions

- [CGraphicsItemConfig](#) (int statesC=STATES_COUNT_DEFAULT, BYTE defState=CELL_STATE_EMPTY, bool edit=false)
- void [SetStatesCount](#) (int sc)
- int [GetStatesCount](#) ()
- void [SetDefState](#) (BYTE ds)
- BYTE [GetDefState](#) ()
- void [SetEditable](#) (bool e)
- bool [GetEditable](#) ()
- void [SetCellActState](#) (BYTE as)
- BYTE [GetCellActState](#) ()

Private Attributes

- unsigned int [iCellStatesCount](#)
states count
- BYTE [byCellDefState](#)
default stace
- bool [bCellEditable](#)
are cells ediable?
- BYTE [byCellActState](#)
actual state on which cell will be set after click

3.21.1 Detailed Description

contains some of cells' config settings

3.21.2 Constructor & Destructor Documentation

3.21.2.1 CGraphicsItemConfig::CGraphicsItemConfig (int statesC = STATES_COUNT_DEFAULT, BYTE defState = CELL_STATE_EMPTY, bool edit = false)

class constructor

Parameters:

- statesC* states count
defState cell's default state
edit is cell editable ?

3.21.3 Member Function Documentation

3.21.3.1 BYTE CGraphicsItemConfig::GetCellActState ()

returns actual state to set

3.21.3.2 BYTE CGraphicsItemConfig::GetDefState ()

returns cell's default state

3.21.3.3 bool CGraphicsItemConfig::GetEditable ()

returns if cell is editable

3.21.3.4 int CGraphicsItemConfig::GetStatesCount ()

returns states count

3.21.3.5 void CGraphicsItemConfig::SetCellActState (BYTE *as*)

sets cell's "actual" state *as* is used to set concrete state to which is cell set after clicking

Parameters:

as actual state to set

3.21.3.6 void CGraphicsItemConfig::SetDefState (BYTE *ds*)

sets cell's default state

Parameters:

ds default state

3.21.3.7 void CGraphicsItemConfig::SetEditable (bool *e*)

sets if cell is editable

Parameters:

e is editable ?

3.21.3.8 void CGraphicsItemConfig::SetStatesCount (int *sc*)

sets cell's states count

Parameters:

sc states count

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

- GraphicsItemConfig.h
- GraphicsItemConfig.cpp

3.22 CGraphicsScene Class Reference

```
#include <GraphicsScene.h>
```

Public Member Functions

- [CGraphicsScene](#) (int sizeX=SPACE_SIZE_X_DEFAULT, int sizeY=SPACE_SIZE_Y_DEFAULT, int states=STATES_COUNT_DEFAULT, BYTE defState=STATE_DEFAULT, bool edit=false)
- [~CGraphicsScene](#) ()
- QGraphicsScene * [GetScene](#) ()
- int [GetWidth](#) ()
- int [GetHeight](#) ()
- [CGraphicsItem](#) * [GetCell](#) (int posX, int posY)
- [CGraphicsItemConfig](#) * [GetCellConfig](#) ()
- void [SetConfigStatesCount](#) (int sc)
- void [SetConfigDefState](#) (BYTE s)
- void [SetConfigEditable](#) (bool e)

Private Member Functions

- void [SetWidth](#) (int w)
- void [SetHeight](#) (int h)
- void [BuildScene](#) (int sizeX, int sizeY)
- void [DeleteScene](#) ()

Private Attributes

- QGraphicsScene * [scene](#)
pointer to scene
- std::vector< [CGraphicsItem](#) * > [vecItem](#)
vector with pointers to cells
- [CGraphicsItem](#) * [errCell](#)
error cell
- [CGraphicsItemConfig](#) [config](#)
cells' config class
- unsigned int [iSizeX](#)
scene's width
- unsigned int [iSizeY](#)
scene's height
- unsigned int [iStatesCount](#)
states count
- BYTE [byDefState](#)

default cell state

- bool [bEditable](#)

are cells editable?

3.22.1 Detailed Description

contains graphics scene - scene with cells

3.22.2 Constructor & Destructor Documentation

3.22.2.1 CGraphicsScene::CGraphicsScene (int *sizeX* = SPACE_SIZE_X_DEFAULT, int *sizeY* = SPACE_SIZE_Y_DEFAULT, int *states* = STATES_COUNT_DEFAULT, BYTE *defState* = STATE_DEFAULT, bool *edit* = false)

class constructor

Parameters:

sizeX count of cells in x axes

sizeY count of cells in y axes

states count of states (for cell config class)

defState default state of cell (for cell config class)

edit cell editable (for cell config class)

3.22.2.2 CGraphicsScene::~~CGraphicsScene ()

class destructor

3.22.3 Member Function Documentation

3.22.3.1 void CGraphicsScene::BuildScene (int *sizeX*, int *sizeY*) [private]

builds scene with cells

Parameters:

sizeX count of cells in x axes

sizeY count of cells in y axes

3.22.3.2 void CGraphicsScene::DeleteScene () [private]

deletes scene with cells

3.22.3.3 CGraphicsItem * CGraphicsScene::GetCell (int *posX*, int *posY*)

returns pointer to cell in given coords

Parameters:

posX x coord of cell

posY y coord of cell

3.22.3.4 CGraphicsItemConfig * CGraphicsScene::GetCellConfig ()

returns pointer to cell config class

3.22.3.5 int CGraphicsScene::GetHeight ()

returns scene "height" - count of cells in y axes

3.22.3.6 QGraphicsScene * CGraphicsScene::GetScene ()

returns pointer to graphics scene which contains cells

3.22.3.7 int CGraphicsScene::GetWidth ()

returns scene "width" - count of cells in x axes

3.22.3.8 void CGraphicsScene::SetConfigDefState (BYTE *ds*)

sets cell's default state

Parameters:

ds default state

3.22.3.9 void CGraphicsScene::SetConfigEditable (bool *e*)

sets if cells are editable

Parameters:

e are editable?

3.22.3.10 void CGraphicsScene::SetConfigStatesCount (int *sc*)

sets states count

Parameters:

sc states count

3.22.3.11 void CGraphicsScene::SetHeight (int *h*) [private]

sets scene height

Parameters:

h count of cell's in y axes

3.22.3.12 void CGraphicsScene::SetWidth (int *w*) [private]

sets scene "width"

Parameters:

w count of cell's in x axes

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

- GraphicsScene.h
- GraphicsScene.cpp

3.23 CGraphicsView Class Reference

```
#include <GraphicsView.h>
```

Public Member Functions

- [CGraphicsView](#) (QWidget *parent=0)
- QGraphicsView * [view](#) () const
- QSize [minimumSizeHint](#) () const

Private Slots

- void [setupMatrix](#) ()
- void [zoomIn](#) ()
- void [zoomOut](#) ()

Private Attributes

- QGraphicsView * [graphicsView](#)
pointer to graphics view class
- QSlider * [sliderZoom](#)
zoom slider

3.23.1 Detailed Description

class creates view to scene with cells

3.23.2 Constructor & Destructor Documentation

3.23.2.1 CGraphicsView::CGraphicsView (QWidget * *parent* = 0)

class constructor

Parameters:

**parent* parent widget

3.23.3 Member Function Documentation

3.23.3.1 QSize CGraphicsView::minimumSizeHint () const

sets minimum size of view widget

3.23.3.2 void CGraphicsView::setupMatrix () [private, slot]

scene zooming

3.23.3.3 QGraphicsView * CGraphicsView::view () const

returns pointer to graphics view

3.23.3.4 void CGraphicsView::zoomIn () [private, slot]

sets zoom slider when zoom in button is pressed

3.23.3.5 void CGraphicsView::zoomOut () [private, slot]

sets zoom slider when zoom out button is pressed

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

- GraphicsView.h
- GraphicsView.cpp

3.24 CImportConfig Class Reference

```
#include <ImportConfig.h>
```

Public Member Functions

- [CImportConfig](#) ()
- void [ImportConfig](#) (QString fileName, [CWidgetInput](#) *widIn, [CWidgetOutput](#) *widOut, [CWidgetEvolution](#) *widEvo, [CWidgetExport](#) *widExp)
- int [GetErrorFlag](#) ()

Private Member Functions

- bool [ReadDocStart](#) ()
- void [ReadDocEnd](#) ()
- void [ReadSettingsInput](#) ([CWidgetInput](#) *widIn)
- void [ReadSettingsOutput](#) ([CWidgetOutput](#) *widOut)
- void [ReadSettingsEvolution](#) ([CWidgetEvolution](#) *widEvo)
- void [ReadSettingsExport](#) ([CWidgetExport](#) *widExp)
- void [ReadSettingsInputCAspace](#) ([CWidgetInput](#) *widIn)
- void [SetIODevice](#) (QIODevice *device)

Private Attributes

- [QXmlStreamReader](#) [xml](#)
xml reader
- [QIODevice](#) * [device](#)
io device - opened file descriptor - for xml class
- int [iErrFlag](#)
error flag

3.24.1 Detailed Description

imports app's configuration from extern file

3.24.2 Constructor & Destructor Documentation

3.24.2.1 CImportConfig::CImportConfig ()

class constructor

3.24.3 Member Function Documentation

3.24.3.1 int CImportConfig::GetErrorFlag ()

returns error flag

3.24.3.2 void CImportConfig::ImportConfig (QString *fileName*, CWidgetInput * *widIn*, CWidgetOutput * *widOut*, CWidgetEvolution * *widEvo*, CWidgetExport * *widExp*)

function for importing genome from xml file created by [CExportGA](#) class

Parameters:

fileName path/name of file

**gen* pointer to genome class (must exist !!) into which data will be written

3.24.3.3 void CImportConfig::ReadDocEnd () [private]

read document's last (closing) tags

3.24.3.4 bool CImportConfig::ReadDocStart () [private]

read document header for determining if file is compatible and if genome is compatible with user settings

3.24.3.5 void CImportConfig::ReadSettingsEvolution (CWidgetEvolution * *widEvo*) [private]

reads evolution settings

Parameters:

**widEvo* pointer to widget with evolution settings

3.24.3.6 void CImportConfig::ReadSettingsExport (CWidgetExport * *widExp*) [private]

reads export settings

Parameters:

**widExp* pointer to widget with export settings

3.24.3.7 void CImportConfig::ReadSettingsInput (CWidgetInput * *widIn*) [private]

reads input settings

Parameters:

**widIn* pointer to widget with input settings

3.24.3.8 void CImportConfig::ReadSettingsInputCASpace (CWidgetInput * *widIn*) [private]

reads input ca space

Parameters:

**widIn* pointer to widget with input settings

3.24.3.9 void CImportConfig::ReadSettingsOutput (CWidgetOutput * *widOut*) [private]

reads output settings

Parameters:

**widIn* pointer to widget with settings

3.24.3.10 void CImportConfig::SetIODevice (QIODevice * *device*) [private]

sets IO device (opened file) in xml class

Parameters:

**device* pointer to opened file descriptor

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

- ImportConfig.h
- ImportConfig.cpp

3.25 CImportGA Class Reference

```
#include <ImportGA.h>
```

Public Member Functions

- [CImportGA](#) ()
- void [ImportChromosome](#) (QString fileName, [CGenome](#) *gen)
- void [SetConfigCore](#) ([CConfigCore](#) *cCore)
- int [GetErrorFlag](#) ()

Private Member Functions

- bool [ReadDocStart](#) ()
- void [ReadGenome](#) ([CGenome](#) *gen)
- void [WriteGenesIntoGenome](#) (QString *genes, [CGenome](#) *gen)
- void [SetIODevice](#) (QIODevice *device)

Private Attributes

- [QXmlStreamReader](#) [xml](#)
xml reader
- [QIODevice](#) * [device](#)
io device - opened file descriptor - for xml class
- [CConfigCore](#) * [config](#)
pointer to config class
- int [iErrFlag](#)
error flag

3.25.1 Detailed Description

imports genome from xml file (created by [CExportGA](#) class) into genome ([CGenome](#) class)

3.25.2 Constructor & Destructor Documentation

3.25.2.1 CImportGA::CImportGA ()

class constructor

3.25.3 Member Function Documentation

3.25.3.1 int CImportGA::GetErrorFlag ()

returns error flag

3.25.3.2 void CImportGA::ImportChromosome (QString *fileName*, CGenome * *gen*)

function for importing genome from xml file created by CExportGA class

Parameters:

fileName path/name of file

**gen* pointer to genome class (must exist !!) into which data will be written

3.25.3.3 bool CImportGA::ReadDocStart () [private]

read document header for determining if file is compatible and if genome is compatible with user settings

3.25.3.4 void CImportGA::ReadGenome (CGenome * *gen*) [private]

this function reads genome data from xml

Parameters:

**gen* pointer to genome class in which data will be stored

3.25.3.5 void CImportGA::SetConfigCore (CConfigCore * *cCore*)

sets config class

Parameters:

**cCore* pointer to config class

3.25.3.6 void CImportGA::SetIODevice (QIODevice * *device*) [private]

sets IO device (opened file) in xml class

Parameters:

**device* pointer to opened file descriptor

3.25.3.7 void CImportGA::WriteGenesIntoGenome (QString * *genes*, CGenome * *genom*) [private]

this function reads genome data - genes - from file and stores them into genome class

Parameters:

**genes* string with genes read in previous function

**genom* pointer to genome class for writing data

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

- ImportGA.h
- ImportGA.cpp

3.26 CInputCA Class Reference

```
#include <InputCA.h>
```

Public Member Functions

- [CInputCA](#) ()
- bool [ImportCA](#) (QString fileName)
- bool [ExportCA](#) (QString fileName)
- void [SetConfigCore](#) (CConfigCore *cCore)
- int [GetErrorFlag](#) ()

Private Attributes

- QDomStreamWriter [xmlW](#)
instance of xml writer class
- QDomStreamReader [xmlR](#)
instance of xml reader class
- QIODevice * [device](#)
io device - opened file descriptor - for xml class
- CConfigCore * [config](#)
pointer to config class
- int [iErrFlag](#)
error flag

3.26.1 Detailed Description

class will be (maybe) used for export/import init CA space to xml file
this functionality is actually cover by [CWindowMain](#) and png files

3.26.2 Constructor & Destructor Documentation

3.26.2.1 CInputCA::CInputCA ()

class constructor

3.26.3 Member Function Documentation

3.26.3.1 bool CInputCA::ExportCA (QString fileName)

export init CA space config to xml - NOT IMPLEMENTED

Parameters:

fileName path/name of file

3.26.3.2 int CInputCA::GetErrorFlag ()

returns error flag

3.26.3.3 bool CInputCA::ImportCA (QString *fileName*)

import init CA space config from xml - NOT IMPLEMENTED

Parameters:

fileName path/name of file

3.26.3.4 void CInputCA::SetConfigCore (CConfigCore * *cCore*)

sets config class

Parameters:

**cCore* pointer to config class

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

- InputCA.h
- InputCA.cpp

3.27 CMutation Class Reference

```
#include <Mutation.h>
```

Public Member Functions

- [CMutation](#) ()
- void [Mutation](#) ([CGenome](#) *g)
- void [SetConfigCore](#) ([CConfigCore](#) *cc)
- int [GetErrorFlag](#) ()

Private Member Functions

- void [_Mutation_01](#) ([CGenome](#) *g)
- void [_Mutation_02](#) ([CGenome](#) *g)
- void [_Mutation_Ins_01](#) ([CGenome](#) *g)

Private Attributes

- [CConfigCore](#) * [pConfigCore](#)
pointer to config class
- [CRandom](#) [random](#)
instance of pseudo-random numbers generator
- int [iErrFlag](#)
error flag

3.27.1 Detailed Description

this class performs mutation of genome

3.27.2 Constructor & Destructor Documentation

3.27.2.1 CMutation::CMutation ()

class constructor

3.27.3 Member Function Documentation

3.27.3.1 void CMutation::_Mutation_01 (CGenome * g) [private]

performs mutation of given genome

Parameters:

*g pointer to genome which should be mutated

3.27.3.2 void CMutation::_Mutation_02 (CGenome * *g*) [private]

new mutation function - NOT IMPLEMENTED

Parameters:

**g* pointer to genome which should be mutated

3.27.3.3 void CMutation::_Mutation_Ins_01 (CGenome * *g*) [private]

performs mutation of instruction genome - NOT IMPLEMENTED

Parameters:

**g* pointer to genome which should be mutated

3.27.3.4 int CMutation::GetErrorFlag ()

return error flag

3.27.3.5 void CMutation::Mutation (CGenome * *g*)

performs mutation of given genome

Parameters:

**g* pointer to genome which should be mutated

3.27.3.6 void CMutation::SetConfigCore (CConfigCore * *cc*)

sets config class

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

- Mutation.h
- Mutation.cpp

3.28 CRandom Class Reference

```
#include <Random.h>
```

Public Member Functions

- [CRandom](#) ()
- double [Random](#) ()
- double [Uniform](#) (double l, double h)
- unsigned long [Uniform](#) (unsigned long l, unsigned long h)
- int [UniformStdLib](#) (int low, int high)
- int [GetErrorFlag](#) ()

Private Member Functions

- double [LCG](#) ()

Private Attributes

- unsigned long [seed](#)
seed for LCG
- unsigned long [ix](#)
actually generated value
- int [iErrFlag](#)
error flag

3.28.1 Detailed Description

this class implements functions for pseudo-random number generation using own LCG

3.28.2 Constructor & Destructor Documentation

3.28.2.1 CRandom::CRandom ()

class constructor

3.28.3 Member Function Documentation

3.28.3.1 int CRandom::GetErrorFlag ()

return error flag

3.28.3.2 double CRandom::LCG () [private]

returns pseudo-random floating point number $<0,1$) using custom Linear Congruent Generator

3.28.3.3 double CRandom::Random ()

returns pseudo-random number $<0,1$)

3.28.3.4 unsigned long CRandom::Uniform (unsigned long *l*, unsigned long *h*)

returns pseudo-random int number $<l,h$)

3.28.3.5 double CRandom::Uniform (double *l*, double *h*)

returns pseudo-random floating point number $<l,h$)

3.28.3.6 int CRandom::UniformStdLib (int *low*, int *high*)

returns pseudo-random int number $<low,high$) using standard generator

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

- Random.h
- Random.cpp

3.29 CRulesTable Class Reference

```
#include <RulesTable.h>
```

Public Member Functions

- [CRulesTable](#) ([CGenome](#) *gen=0)
- void [SetGenome](#) ([CGenome](#) *gen)
- [CGenome](#) * [GetGenome](#) ()
- [BYTE](#) [at](#) (int index)
- void [SetConfigCore](#) ([CConfigCore](#) *pCC)

Private Attributes

- [CGenome](#) * [genome](#)
pointer to genome
- [CConfigCore](#) * [pConfigCore](#)
pointer to config class

3.29.1 Detailed Description

rules table is touching point between CA and GA, contains pointer to genome and return genes from it

3.29.2 Constructor & Destructor Documentation

3.29.2.1 CRulesTable::CRulesTable ([CGenome](#) * *gen* = 0)

class constructor

Parameters:

**gen* pointer to genome from genetic algorithm

3.29.3 Member Function Documentation

3.29.3.1 [BYTE](#) CRulesTable::at (int *index*)

return gene from genome at given index

Parameters:

index index to genome

3.29.3.2 [CGenome](#) * CRulesTable::GetGenome ()

returns pointer to actually used genome

3.29.3.3 void CRulesTable::SetConfigCore (CConfigCore * *pCC*)

sets config class

3.29.3.4 void CRulesTable::SetGenome (CGenome * *gen*)

sets given genome for ca computations

Parameters:

**gen* pointer to genome from genetic algorithm

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

- RulesTable.h
- RulesTable.cpp

3.30 CSelection Class Reference

```
#include <Selection.h>
```

Public Member Functions

- [CSelection](#) ()
- void [Selection](#) (std::vector< [CGenome](#) * > *act, std::vector< [CGenome](#) * > *next)
- void [SetConfigCore](#) ([CConfigCore](#) *cc)

Private Member Functions

- void [_Selection_01](#) ()
- void [_Selection_02](#) ()

Private Attributes

- [CConfigCore](#) * [pConfigCore](#)
pointer to config class
- [CGenome](#) * [genome](#)
pointer to actually used genome
- [CRandom](#) [random](#)
instance of number generating class
- std::vector< [CGenome](#) * > * [vecGenerationAct](#)
pointer to vector which contains actual generation of population
- std::vector< [CGenome](#) * > * [vecGenerationTmp](#)
pointer to vector for next generation

3.30.1 Detailed Description

performs selection, actually by using tournamen selection implements elitism

3.30.2 Constructor & Destructor Documentation

3.30.2.1 CSelection::CSelection ()

class constructor

3.30.3 Member Function Documentation

3.30.3.1 void CSelection::_Selection_01 () [private]

performs rulete selection - OLD one, actually NOT USED

3.30.3.2 void CSelection::_Selection_02 () [private]

performs tournament selection of genomes ELITISM is used - first of all, best genome from whole population is found, then DEEP COPY of this one is performed, and this copy is inserted into population vector at 1st position best genome is then selected 2nd time, this 2nd copy will be later mutated, first copy wont be

3.30.3.3 void CSelection::Selection (std::vector< CGenome * > * *act*, std::vector< CGenome * > * *next*)

performs selection of genomes from actual generation into next

Parameters:

- **act* pointer to vector with pointers to genomes of actual generation
- **next* pointer to vector with pointers to genomes of next generation

3.30.3.4 void CSelection::SetConfigCore (CConfigCore * *cc*)

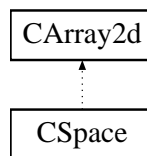
set config class

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

- Selection.h
- Selection.cpp

3.31 CSpace Class Reference

`#include <Space.h>`Inheritance diagram for CSpace::



Public Member Functions

- [CSpace](#) (int type=SPACE_TYPE_GRID, int sizeX=SPACE_SIZE_X_DEFAULT, int sizeY=SPACE_SIZE_Y_DEFAULT, BYTE defState=CELL_STATE_EMPTY, [CConfigCore](#) *pCC=NULL)
- [~CSpace](#) ()
- BYTE [at](#) (int posX, int posY) const
- BYTE & [at](#) (int posX, int posY)
- BYTE [atGrid](#) (int posX, int posY) const
- BYTE & [atGrid](#) (int posX, int posY)
- BYTE [atTorus](#) (int posX, int posY) const
- BYTE & [atTorus](#) (int posX, int posY)
- int [GetWidth](#) ()
- int [GetHeight](#) ()
- int [GetSpaceType](#) ()
- int [GetErrorFlag](#) ()

Private Attributes

- int [iSpaceType](#)
space type
- BYTE [errCell](#)
error cell
- BYTE [byDefCell](#)
default value cell

3.31.1 Detailed Description

inherits [CArray2d](#), which is carrying 2d array, this class implements behaviour of ca space - grid or torus

3.31.2 Constructor & Destructor Documentation

3.31.2.1 `CSpace::CSpace (int type = SPACE_TYPE_GRID, int sizeX = SPACE_SIZE_X_DEFAULT, int sizeY = SPACE_SIZE_Y_DEFAULT, BYTE defState = CELL_STATE_EMPTY, CConfigCore * pConfigCore = NULL)`

class constructor

Parameters:

type torus or lattice
sizeX array width
sizeY array height
defState default state
**pConfigCore* pointer to config class
[CArray2d CSpace](#) inherits this class

3.31.2.2 `CSpace::~~CSpace ()`

class destructor

3.31.3 Member Function Documentation

3.31.3.1 `BYTE & CSpace::at (int posX, int posY)`

returns reference to element at given coordinates

Parameters:

posX position on x-axes
posY position on y-axes

3.31.3.2 `BYTE CSpace::at (int posX, int posY) const`

returns element at given coordinates

Parameters:

posX position on x-axes
posY position on y-axes

3.31.3.3 `BYTE & CSpace::atGrid (int posX, int posY)`

returns reference to element at given coordinates with respect to cellular space type "grid"

Parameters:

posX position on x-axes
posY position on y-axes

3.31.3.4 BYTE CSpace::atGrid (int *posX*, int *posY*) const

returns element at given coordinates with respect to cellular space type "grid"

Parameters:

posX position on x-axes

posY position on y-axes

3.31.3.5 BYTE & CSpace::atTorus (int *posX*, int *posY*)

returns reference to element at given coordinates with respect to cellular space type "torus" - quasi-infinite space

Parameters:

posX position on x-axes

posY position on y-axes

3.31.3.6 BYTE CSpace::atTorus (int *posX*, int *posY*) const

returns element at given coordinates with respect to cellular space type "torus" - quasi-infinite space

Parameters:

posX position on x-axes

posY position on y-axes

3.31.3.7 int CSpace::GetErrorFlag ()

return error flag

Reimplemented from [CArray2d](#).

3.31.3.8 int CSpace::GetHeight ()

returns array height

Reimplemented from [CArray2d](#).

3.31.3.9 int CSpace::GetSpaceType ()

returns space type

3.31.3.10 int CSpace::GetWidth ()

returns array width

Reimplemented from [CArray2d](#).

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

- Space.h
- Space.cpp

3.32 CTFUNCTION Class Reference

```
#include <TFunction.h>
```

Public Member Functions

- [CTFunction](#) ()
- void [SetRulesTable](#) ([CRulesTable](#) *rt)
- void [SetConfigCore](#) ([CConfigCore](#) *cc)
- void [NextSpace](#) ([CSpace](#) *sa, [CSpace](#) *sn)
- int [GetErrorFlag](#) ()

Private Member Functions

- void [NextSpaceGenomeStandard](#) ([CSpace](#) *sa, [CSpace](#) *sn)
- void [NextSpaceGenomeInstruction](#) ([CSpace](#) *sa, [CSpace](#) *sn)
- int [CalculateIndexGrid](#) ([CSpace](#) *s, int x, int y)
- int [CalculateIndexTorus](#) ([CSpace](#) *s, int x, int y)

Private Attributes

- [CRulesTable](#) * [rules](#)
pointer to rules table
- [CConfigCore](#) * [pConfigCore](#)
pointer to config class
- int [iErrFlag](#)
error flag

3.32.1 Detailed Description

transition function performs all calculations of CA for all cell in space it calculates index to genome, gets gene using [CRulesTable](#) class and writes new value of cell into next-gen space

3.32.2 Constructor & Destructor Documentation

3.32.2.1 CTFUNCTION::CTFUNCTION ()

class constructor

3.32.3 Member Function Documentation

3.32.3.1 int CTFUNCTION::CalculateIndexGrid ([CSpace](#) *s, int x, int y) [private]

calculates index into genome using "grid" rules

Parameters:

**s* pointer to actual ca space
x x coordinate of actual cell
y y coordinate of actual cell

3.32.3.2 int CTFFunction::CalculateIndexTorus (CSpace * *s*, int *x*, int *y*) [private]

calculates index into genome using "torus" rules

Parameters:

**s* pointer to actual ca space
x x coordinate of actual cell
y y coordinate of actual cell

3.32.3.3 void CTFFunction::NextSpace (CSpace * *sa*, CSpace * *sn*)

computes new ca array from old one with rules table given

Parameters:

**sa* ca space actual
**sn* ca space next

3.32.3.4 void CTFFunction::NextSpaceGenomeInstruction (CSpace * *sa*, CSpace * *sn*) [private]

computes new ca space with instruction genome, NOT IMPLEMENTED

Parameters:

**sa* ca space actual
**sn* ca space next

3.32.3.5 void CTFFunction::NextSpaceGenomeStandard (CSpace * *sa*, CSpace * *sn*) [private]

computes new ca space with standard, non-instruction genome

Parameters:

**sa* ca space actual
**sn* ca space next

3.32.3.6 void CTFFunction::SetConfigCore (CConfigCore * *cc*)

sets config class

3.32.3.7 void CTFUNCTION::SetRulesTable (CRulesTable * *rt*)

sets actual rules table, which mapping genome from ga

Parameters:

**rt* pointer to rules table class

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

- TFunction.h
- TFunction.cpp

3.33 CThreadCore Class Reference

```
#include <ThreadCore.h>
```

Signals

- void **SignalErrCore** (int)
- void **SignalInitCoreDone** ()
- void **SignalNewDataAvailable** ()
- void **SignalThreadState** (int)

Public Member Functions

- [CThreadCore](#) ()
- [~CThreadCore](#) ()
- void [run](#) ()
- void [SetConfigCore](#) (CConfigCore *cc)
- void [SetCoreSpace](#) (CSpace *cs)
- void [SetCoreDataGA](#) (struct stThreadCoreDataGA *tcdga)
- void [SetMutex](#) (QMutex *mc)
- void [SetWaitCondition](#) (QWaitCondition *wcc)
- void [SetCoreDataExpiration](#) (bool *be)
- void [SetSimulationRunning](#) (bool *sr)
- CSpace * [GetSpace](#) ()
- CSpace * [GetInitSpace](#) ()
- bool [IsInitDone](#) ()
- void [TerminateThreadLoop](#) ()
- bool [CheckThreadLoopTermination](#) ()
- void [TerminateCaRun](#) ()
- bool [CheckCaRunTermination](#) ()
- int [GetErrorFlag](#) ()

Private Member Functions

- void [InitCore](#) ()
- void [ReinitCore](#) ()
- void [InitCoreGeneticAlgorithm](#) ()
- void [InitCoreCellularAutomata](#) ()
- void [InitCoreCAMemory](#) ()
- void [InitExport](#) ()
- void [InitTmpCAs](#) ()
- void [ClearTmpCAs](#) ()
- bool [CheckCorePointers](#) ()
- void [WriteCASpaceFromCoreIntoCA](#) ()
- void [WriteDataCAToCore](#) ()
- void [WriteDataGAToCore](#) ()
- void [SyncDataWithCore](#) ()
- void [SetCoreDataValidity](#) (bool dve)
- void [RunGuiMode](#) (int call_pos)

- void [FileExportGa](#) (int call_pos)
- void [FileExportCaInit](#) ()
- void [FileExportCaSteps](#) (CGenome *gen)
- void [RunGenomeCaSimulation](#) (CGenome *gen)
- void [StoreGenomeDataForGui](#) (int mode)

Private Attributes

- bool [bInitDone](#)
init done var
- bool [bEvoExit](#)
evolution (thread) termination var
- bool [bCaRunTermination](#)
ca simulator terminated var
- CGeneticAlgorithm * [ga](#)
pointer to instance of main GA class
- CCellularAutomata * [ca](#)
pointer to instance of main CA class
- CConfigCore * [pConfigCore](#)
pointer to config class
- CSpace * [pCoreSpace](#)
pointer to CCore ca space used for sending data into gui
- CExportGA [exportGa](#)
instance of export ga class
- CExportCA [exportCa](#)
instance of export ca class
- CImportGA [importGa](#)
instance of import genome class
- CExportLog [exportLog](#)
instance of export error log class
- CGenome * [bestGenome](#)
best genome in actual evolution run
- CGenome * [bestGenomeTotal](#)
pointer to total best genome class (best genome of all independet evolution runs)
- CGenome * [actGenome](#)
pointer to actual genome class

- [CGenome * importGenome](#)
pointer to import genome class
- [QMutex * pMutexCore](#)
pointer to [CCore](#) var used for threads sync
- [QWaitCondition * pWaitCCore](#)
pointer to [CCore](#) var used for threads sync
- [bool * bCoreDataExpired](#)
pointer to [CCore](#) var defining if data send into [CCore](#) are displayed in gui
- [bool * bCoreSimRunning](#)
pointer to [CCore](#) var defining if this thread should start compute
- [std::string gol](#)
i am not sure .. probably from very early version for GoL genome
- [int iErrFlag](#)
error flag
- [struct stThreadCoreDataGA dataGaAct](#)
actually used genome
- [struct stThreadCoreDataGA dataGaMax](#)
best genome of this run
- [struct stThreadCoreDataGA dataGaTot](#)
ga data of total best genome (best genome of all independent runs)
- [struct stThreadCoreDataGA * dataGaToGui](#)
pointer to data struct from which data to gui will be send
- [struct stThreadCoreDataGA * pCoreDataGa](#)
pointer to [CCore](#) ga data struct
- [QString exportFileGa](#)
name of ga export file
- [QString exportFileCa](#)
name of ca export file
- [CCellularAutomata * tmpVecCA](#)
pointer to CA class
- [std::vector< CCellularAutomata * > vecTmpCAs](#)
vector with cellular automaton
- [CConfigCore configTmpCAs](#)
config core for tmp CAs

3.33.1 Detailed Description

main compute class of evolution this class is using classes of GA and CA

3.33.2 Constructor & Destructor Documentation

3.33.2.1 CThreadCore::CThreadCore ()

class constructor

3.33.2.2 CThreadCore::~~CThreadCore ()

class destructor

3.33.3 Member Function Documentation

3.33.3.1 bool CThreadCore::CheckCaRunTermination ()

checks, if ca simulator was abord from gui

3.33.3.2 bool CThreadCore::CheckCorePointers () [private]

checks if all pointers to [CCore](#) var are correctly set

3.33.3.3 bool CThreadCore::CheckThreadLoopTermination ()

checks if app/thread core was terminated

3.33.3.4 void CThreadCore::ClearTmpCAs () [private]

clears tmp CAs which are used for evolving vseo chromosomes

3.33.3.5 void CThreadCore::FileExportCaInit () [private]

exports init ca configuration into png (using [CExportCA](#) class)

3.33.3.6 void CThreadCore::FileExportCaSteps (CGenome * *gen*) [private]

exports all steps (from period) of ca run of object with selected genome for export space into png is used [CExportCA](#) class example: exports 4 png-s of glider with game of life rules

Parameters:

**gen* pointer to genome

3.33.3.7 void CThreadCore::FileExportGa (int *call_pos*) [private]

exports genome from actual run, *call_pos* determines, which data will be exported - whole actual population, best genome from actual population or best genome from all evolution runs

3.33.3.8 int CThreadCore::GetErrorFlag ()

returns error flag

3.33.3.9 CSpace * CThreadCore::GetInitSpace ()

returns pointer to init ca space

3.33.3.10 CSpace * CThreadCore::GetSpace ()

returns pointer to actual ca space

3.33.3.11 void CThreadCore::InitCore () [private]

this function inits thread core init must be done before evolution starts

3.33.3.12 void CThreadCore::InitCoreCAMemory () [private]

inits ca classes

3.33.3.13 void CThreadCore::InitCoreCellularAutomata () [private]

creates instance of CA used in evolution

3.33.3.14 void CThreadCore::InitCoreGeneticAlgorithm () [private]

creates new GA instance used in evolution

3.33.3.15 void CThreadCore::InitExport () [private]

inits export classes

3.33.3.16 void CThreadCore::InitTmpCAs () [private]

inits tmp CAs which are used for evolving vseo chromosomes

3.33.3.17 bool CThreadCore::IsInitDone ()

returns core init state

3.33.3.18 void CThreadCore::ReinitCore () [private]

reinit thread - used when evolution repetition occurs evolution repetitions = independent runs of evolution GA needs to be reinit

3.33.3.19 void CThreadCore::run ()

main compute function, runs in own thread 1st of all, init part of thread is done if any error occurs, this error is send into gui and thread is then abort after successful init thread wait to "Evolve" button in gui is pressed then evolution will begin

3.33.3.20 void CThreadCore::RunGenomeCaSimulation (CGenome * *gen*) [private]

this function is ca run simulator given genome is used for for this simulation simulation data (ca space0 are send into gui

Parameters:

**gen* genome used in simulation

3.33.3.21 void CThreadCore::RunGuiMode (int *call_pos*) [private]

this function is used for sending data from this thread class into gui

Parameters:

call_pos determines which data will be send

3.33.3.22 void CThreadCore::SetConfigCore (CConfigCore * *cc*)

sets pointer to config class

Parameters:

**cc* pointer to config class

3.33.3.23 void CThreadCore::SetCoreDataExpiration (bool * *be*)

sets pointer to [CCore](#) var which determining, if data in [CCore](#) was already written into gui

Parameters:

**be* pointer to [CCore](#) bool var

3.33.3.24 void CThreadCore::SetCoreDataGA (struct stThreadCoreDataGA * *tcdga*)

sets pointer to data struct in [CCore](#) which is used to send GA data from this thread class into gui

Parameters:

**tcdga* pointer to ga data struct

3.33.3.25 void CThreadCore::SetCoreDataValidity (bool *dve*) [private]

sets if data in gui (CCore class) are valid or not

Parameters:

dve data validity

3.33.3.26 void CThreadCore::SetCoreSpace (CSpace * *cs*)

sets pointer to ca space in CCore - this ca space is used to write ca data from this thread class into gui

Parameters:

**ca* pointer to ca space in CCore

3.33.3.27 void CThreadCore::SetMutex (QMutex * *mc*)

sets pointer to mutex from CCore class

Parameters:

**mc* pointer to mutex class

3.33.3.28 void CThreadCore::SetSimulationRunning (bool * *sr*)

sets if thread core would start simulation/evolution thread core is paused after init part of thread is done

Parameters:

**sr* pointer to CCore bool var

3.33.3.29 void CThreadCore::SetWaitCondition (QWaitCondition * *wcc*)

sets pointer to wait condition class from CCore

Parameters:

**wcc* pointer to wait condition class

3.33.3.30 void CThreadCore::StoreGenomeDataForGui (int *mode*) [private]

this function is used for initializing data struct "dataGaAct" which holds information about ga computations which will be send into gui also this fc sets pointer to that data struct, from which ga data will be send into gui

Parameters:

mode sets mode (way of if condition)

3.33.3.31 void CThreadCore::SyncDataWithCore () [private]

sends data into [CCore](#) class from which they are send into gui

3.33.3.32 void CThreadCore::TerminateCaRun ()

sets var for ca simulator termination

3.33.3.33 void CThreadCore::TerminateThreadLoop ()

sets var for thread core termination

3.33.3.34 void CThreadCore::WriteCAspaceFromCoreIntoCA () [private]

this function reads data from [CCore](#) class (ca space init config) and propagates them into ca class

3.33.3.35 void CThreadCore::WriteDataCAToCore () [private]

sends ca space into [CCore](#)

3.33.3.36 void CThreadCore::WriteDataGAToCore () [private]

sends ga data struct into [CCore](#)

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

- ThreadCore.h
- ThreadCore.cpp

3.34 CWidgetEvolution Class Reference

```
#include <WidgetEvolution.h>
```

Signals

- void **SignalWidgetEvoApplayed** (bool)

Public Member Functions

- [CWidgetEvolution](#) (QWidget *parent=0)
- [~CWidgetEvolution](#) ()
- int [GetRepetitionsCount](#) ()
- int [GetGenerationsCount](#) ()
- int [GetPopulationSize](#) ()
- int [GetGenomeType](#) ()
- int [GetCrossoverProbability](#) ()
- int [GetCrossoverCount](#) ()
- int [GetMutationProbability](#) ()
- int [GetMutationCount](#) ()
- int [GetMoveDirection](#) ()
- int [GetStepsCount](#) ()
- int [GetMoveDistance](#) ()
- bool [IsImportGenomeEnabledSimulation](#) ()
- bool [IsImportGenomeEnabledReevolve](#) ()
- QString [GetImportGenomeFile](#) ()
- void [SetRepetitionsCount](#) (int rc)
- void [SetGenerationsCount](#) (int gc)
- void [SetPopulationSize](#) (int ps)
- void [SetGenomeType](#) (int gt)
- void [SetCrossoverProbability](#) (int cp)
- void [SetCrossoverCount](#) (int cc)
- void [SetMutationProbability](#) (int mp)
- void [SetMutationCount](#) (int mc)
- void [SetMoveDirection](#) (int md)
- void [SetStepsCount](#) (int sc)
- void [SetMoveDistance](#) (int md)
- bool [IsApplied](#) ()
- void [SetInitDone](#) (bool id)

Protected Member Functions

- void [changeEvent](#) (QEvent *e)

Private Slots

- void [RepetitionsChanged](#) (int r)
- void [GenerationsChanged](#) (int g)
- void [PopulationChanged](#) (int p)
- void [GenomeTypeChanged](#) (int g)
- void [DirectionChanged](#) (int d)
- void [StepsChanged](#) (int s)
- void [DistanceChanged](#) (int d)
- void [CrossoverProbChanged](#) (int cp)
- void [CrossoverCountChanged](#) (int cc)
- void [MutationProbChanged](#) (int mp)
- void [MutationCountChanged](#) (int mc)
- void [ImportFileButtonPressed](#) ()
- void [ImportFileEnableSimulation](#) (int ifes)
- void [ImportFileEnableReevolve](#) (int ifer)
- void [ApplyPressed](#) ()
- void [CancelPressed](#) ()

Private Attributes

- [Ui::CWidgetEvolution * ui](#)
pointer to widget (gui of this class)
- int [iRepetitions](#)
repetitions - independent runs of evolution
- int [iGenerations](#)
generations count
- int [iPopulation](#)
population size
- int [iStepsCount](#)
ca steps count
- int [iCrossoverProb](#)
crossover probability
- int [iCrossoverCount](#)
crossovers count
- int [iMutationProb](#)
mutation probability
- int [iMutationCount](#)
mutations count
- int [iMoveDir](#)
movement direction

- int [iMoveDis](#)
movement distance
- int [iGenomeType](#)
genome type - 5nbh || 9nbh
- bool [bApplied](#)
is this widget applied?
- bool [bInitDone](#)
is app core init?
- QString [importFile](#)
path/name to ga import file
- bool [bImportFileEnabledSimulation](#)
should be imported genome used for simulation?
- bool [bImportFileEnabledReevolve](#)
should be imported genome used for reevolution?

3.34.1 Detailed Description

settings of evolution and movement of object

3.34.2 Constructor & Destructor Documentation

3.34.2.1 CWidgetEvolution::CWidgetEvolution (QWidget * *parent* = 0)

class constructor

Parameters:

**parent* pointer to parent widget

3.34.2.2 CWidgetEvolution::~~CWidgetEvolution ()

class destructor

3.34.3 Member Function Documentation

3.34.3.1 void CWidgetEvolution::ApplyPressed () [private, slot]

when apply is pressed disable all functional tools in widget

3.34.3.2 void CWidgetEvolution::CancelPressed () [private, slot]

when cancel is pressed, enable all functional tools in widget

3.34.3.3 void CWidgetEvolution::changeEvent (QEvent * *e*) [protected]

change event happened

Parameters:

e event

3.34.3.4 void CWidgetEvolution::CrossoverCountChanged (int *cc*) [private, slot]

crossover count spinbox was changed - set new crossover count

Parameters:

cc new value from spinbox

3.34.3.5 void CWidgetEvolution::CrossoverProbChanged (int *cp*) [private, slot]

crossover probability spinbox was changed - set new crossover prob

Parameters:

cp new value from spinbox

3.34.3.6 void CWidgetEvolution::DirectionChanged (int *d*) [private, slot]

direction combobox was changed - set new direction

Parameters:

d new value from combobox

3.34.3.7 void CWidgetEvolution::DistanceChanged (int *d*) [private, slot]

distance spinbox was changed - set new distance

Parameters:

d new value from spinbox

3.34.3.8 void CWidgetEvolution::GenerationsChanged (int *g*) [private, slot]

generation count spinbox was changed - set new generation count

Parameters:

g new value from spinbox

3.34.3.9 void CWidgetEvolution::GenomeTypeChanged (int *g*) [private, slot]

genome type spinbox was changed - set new genome type

Parameters:

g new value from combobox

3.34.3.10 int CWidgetEvolution::GetCrossoverCount ()

returns crossover count

3.34.3.11 int CWidgetEvolution::GetCrossoverProbability ()

returns crossover probability

3.34.3.12 int CWidgetEvolution::GetGenerationsCount ()

returns generations count

3.34.3.13 int CWidgetEvolution::GetGenomeType ()

returns genome type

3.34.3.14 QString CWidgetEvolution::GetImportGenomeFile ()

returns path/name of imported file

3.34.3.15 int CWidgetEvolution::GetMoveDirection ()

returns movement direction

3.34.3.16 int CWidgetEvolution::GetMoveDistance ()

returns movement distance

3.34.3.17 int CWidgetEvolution::GetMutationCount ()

returns mutation count

3.34.3.18 int CWidgetEvolution::GetMutationProbability ()

returns mutation probability

3.34.3.19 int CWidgetEvolution::GetPopulationSize ()

returns population size

3.34.3.20 int CWidgetEvolution::GetRepetitionsCount ()

returns repetitions count

3.34.3.21 int CWidgetEvolution::GetStepsCount ()

returns steps count

3.34.3.22 void CWidgetEvolution::ImportFileButtonPressed () [private, slot]

import genome button is pressed

3.34.3.23 void CWidgetEvolution::ImportFileEnableReevolve (int *ifer*) [private, slot]

re-evolve checkbox changed it's state

Parameters:

ifer current check state of reevolve checkbox

3.34.3.24 void CWidgetEvolution::ImportFileEnableSimulation (int *ifes*) [private, slot]

simulation checkbox changed it's state

Parameters:

ifes current check state of simulation checkbox

3.34.3.25 bool CWidgetEvolution::IsApplied ()

are settings from this widget applied

3.34.3.26 bool CWidgetEvolution::IsImportGenomeEnabledReevolve ()

should be imported genome used for reevolution?

3.34.3.27 bool CWidgetEvolution::IsImportGenomeEnabledSimulation ()

should be imported genome used in simulation?

3.34.3.28 void CWidgetEvolution::MutationCountChanged (int *mc*) [private, slot]

mutation count spinbox was changed - set new mutation count

Parameters:

mc new value from spinbox

3.34.3.29 void CWidgetEvolution::MutationProbChanged (int *mp*) [private, slot]

mutation probability spinbox was changed - set new mutation prob

Parameters:

mp new value from spinbox

3.34.3.30 void CWidgetEvolution::PopulationChanged (int *p*) [private, slot]

population size spinbox was changed - set new popluation size

Parameters:

p new value from spinbox

3.34.3.31 void CWidgetEvolution::RepetitionsChanged (int *r*) [private, slot]

repetitions spinbox was changed - set new repetitions

Parameters:

r new value from spinbox

3.34.3.32 void CWidgetEvolution::SetCrossoverCount (int *cc*)

sets crossover count - used as extern call - when settings are loaded from file

Parameters:

cc new crossover count

3.34.3.33 void CWidgetEvolution::SetCrossoverProbability (int *cp*)

sets crossover probability - used as extern call - when settings are loaded from file

Parameters:

cp new crossover probability

3.34.3.34 void CWidgetEvolution::SetGenerationsCount (int *gc*)

sets generations count - used as extern call - when settings are loaded from file

Parameters:

gc new generation count

3.34.3.35 void CWidgetEvolution::SetGenomeType (int *gt*)

sets genome type - used as extern call - when settings are loaded from file

Parameters:

gt new genome type

3.34.3.36 void CWidgetEvolution::SetInitDone (bool *id*)

button init changed state - enable/disable any changes in widget

3.34.3.37 void CWidgetEvolution::SetMoveDirection (int *md*)

sets direction - used as extern call - when settings are loaded from file

Parameters:

md new direction

3.34.3.38 void CWidgetEvolution::SetMoveDistance (int *md*)

sets distance - used as extern call - when settings are loaded from file

Parameters:

md new distance

3.34.3.39 void CWidgetEvolution::SetMutationCount (int *mc*)

sets mutation count - used as extern call - when settings are loaded from file

Parameters:

mc new mutation count

3.34.3.40 void CWidgetEvolution::SetMutationProbability (int *mp*)

sets mutation probability - used as extern call - when settings are loaded from file

Parameters:

mp new mutation probability

3.34.3.41 void CWidgetEvolution::SetPopulationSize (int *ps*)

sets population size - used as extern call - when settings are loaded from file

Parameters:

ps new population size

3.34.3.42 void CWidgetEvolution::SetRepetitionsCount (int *rc*)

sets repetitions - used as extern call - when settings are loaded from file

Parameters:

rc new repetitions count

3.34.3.43 void CWidgetEvolution::SetStepsCount (int *sc*)

sets steps count - used as extern call - when settings are loaded from file

Parameters:

sc new steps count

3.34.3.44 void CWidgetEvolution::StepsChanged (int *s*) [private, slot]

steps count spinbox was changed - set new steps count

Parameters:

s new value from spinbox

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

- WidgetEvolution.h
- WidgetEvolution.cpp

3.35 CWidgetExport Class Reference

```
#include <WidgetExport.h>
```

Signals

- void **SignalWidgetExportApplied** (bool)

Public Member Functions

- [CWidgetExport](#) (QWidget *parent=0)
- [~CWidgetExport](#) ()
- int [GetGuiDataDisplayModeTimeout](#) ()
- int [GetGuiDataDisplayModeCA](#) ()
- QString [GetFileExportPath](#) ()
- int [GetFileExportModeCA](#) ()
- int [GetFileExportModeGA](#) ()
- void [SetGuiDataDisplayModeTimeout](#) (int to)
- void [SetGuiDataDisplayModeCA](#) (int dm)
- void [SetFileExportModeCA](#) (int mca)
- void [SetFileExportModeGA](#) (int mga)
- bool [IsApplied](#) ()
- void [SetInitDone](#) (bool id)

Protected Member Functions

- void [changeEvent](#) (QEvent *e)

Private Slots

- void [GuiModeCATimeoutChanged](#) (int t)
- void [GuiModeCADisplayModeChanged](#) (int ca)
- void [FileExportPathChanged](#) ()
- void [FileExportCAChanged](#) (int index)
- void [FileExportGACHanged](#) (int index)
- void [ApplyPressed](#) ()
- void [CancelPressed](#) ()

Private Attributes

- Ui::CWidgetExport * [ui](#)
pointer to widget (gui of this class)
- unsigned int [iGuiModeCA_Timeout](#)
simulator's animation timeout
- int [iGuiModeCA_DisplayMode](#)
simulator display mode

- QString [actPath](#)
actual export path
- QString [tmpPath](#)
tmp export path
- int [iExportModeCA](#)
ca export mode
- int [iExportModeGA](#)
ga export mode
- bool [bApplied](#)
is this widget applied?
- bool [bInitDone](#)
is app core init?

3.35.1 Detailed Description

export settings

3.35.2 Constructor & Destructor Documentation

3.35.2.1 CWidgetExport::CWidgetExport (QWidget * *parent* = 0)

class constructor

Parameters:

**parent* pointer to parent widget

3.35.2.2 CWidgetExport::~~CWidgetExport ()

class destructor

3.35.3 Member Function Documentation

3.35.3.1 void CWidgetExport::ApplyPressed () [**private**, **slot**]

when apply is pressed disable all functional tools in widget

3.35.3.2 void CWidgetExport::CancelPressed () [**private**, **slot**]

when cancel is pressed, enable all functional tools in widget

3.35.3.3 void CWidgetExport::changeEvent (QEvent * *e*) [protected]

change event happened

Parameters:

e event

3.35.3.4 void CWidgetExport::FileExportCAChanged (int *index*) [private, slot]

ca export mode was changed

Parameters:

index new mode from combobox

3.35.3.5 void CWidgetExport::FileExportGAChecked (int *index*) [private, slot]

ga export mode was changed

Parameters:

index new mode from combobox

3.35.3.6 void CWidgetExport::FileExportPathChanged () [private, slot]

export path button was pressed - get new export path

3.35.3.7 int CWidgetExport::GetFileExportModeCA ()

returns ca export mode

3.35.3.8 int CWidgetExport::GetFileExportModeGA ()

returns ga export mode

3.35.3.9 QString CWidgetExport::GetFileExportPath ()

returns path to export folder

3.35.3.10 int CWidgetExport::GetGuiDataDisplayModeCA ()

returns simulation gddm

3.35.3.11 int CWidgetExport::GetGuiDataDisplayModeTimeout ()

returns animation timeout

3.35.3.12 void CWidgetExport::GuiModeCADisplayModeChanged (int *ca*) [private, slot]

simulation display mode was changed

Parameters:

ca new simulator display mode from combobox

3.35.3.13 void CWidgetExport::GuiModeCATimeoutChanged (int *t*) [private, slot]

simulation animation timeout changed

Parameters:

t new timeout

3.35.3.14 bool CWidgetExport::IsApplied ()

was apply button pressed?

3.35.3.15 void CWidgetExport::SetFileExportModeCA (int *mca*)

sets ca export mode - used as extern call - when settings are loaded from file

Parameters:

mca new ca export mode

3.35.3.16 void CWidgetExport::SetFileExportModeGA (int *mga*)

sets ga export mode - used as extern call - when settings are loaded from file

Parameters:

mga new ga export mode

3.35.3.17 void CWidgetExport::SetGuiDataDisplayModeCA (int *dm*)

sets simulator gddm - used as extern call - when settings are loaded from file

Parameters:

dm new gddm ca

3.35.3.18 void CWidgetExport::SetGuiDataDisplayModeTimeout (int *to*)

sets animation timeout - used as extern call - when settings are loaded from file

Parameters:

to new animation timeout

3.35.3.19 void CWidgetExport::SetInitDone (bool *id*)

button init changed state - enable/disable any changes in widget

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

- WidgetExport.h
- WidgetExport.cpp

3.36 CWidgetInput Class Reference

```
#include <WidgetInput.h>
```

Signals

- void **SignalWidgetInputApplied** (bool)

Public Member Functions

- [CWidgetInput](#) (QWidget *parent=0)
- [~CWidgetInput](#) ()
- [CGraphicsScene](#) * [GetGrid](#) ()
- bool [IsApplied](#) ()
- void [SetInitDone](#) (bool id)
- void [SetOutputSizeX](#) (int x)
- void [SetOutputSizeY](#) (int y)
- void [SetOutputMapPosX](#) (int x)
- void [SetOutputMapPosY](#) (int y)
- void [SetOutputApplied](#) (bool a)
- void [SetInputGridSizeX](#) (int x)
- void [SetInputGridSizeY](#) (int y)
- void [SetInputGridSize](#) (int x, int y)
- void [SetStatesCount](#) (int s)

Protected Member Functions

- void [changeEvent](#) (QEvent *e)

Private Slots

- void [SizeChangedX](#) (int x)
- void [SizeChangedY](#) (int y)
- void [SizeChangeUsed](#) ()
- void [StatesChanged](#) (int s)
- void [StatesChandeUsed](#) ()
- void [ImportPressed](#) ()
- void [ApplyPressed](#) ()
- void [CancelPressed](#) ()

Private Member Functions

- void [SetupInputView](#) ()
- void [ReinitInputView](#) ()
- void [DeleteInputGui](#) ()
- bool [CheckSize](#) (int x, int y)
- bool [CheckPos](#) (int x, int y)

Private Attributes

- `Ui::CWidgetInput * ui`
pointer to widget (gui of this class)
- `CGraphicsScene * grid`
pointer to scene class
- `CGraphicsView * view`
pointer to view class
- `unsigned int iSizeX`
space width
- `unsigned int iSizeY`
space height
- `unsigned int iStates`
states count
- `int iTmpSizeX`
tmp space width
- `int iTmpSizeY`
tmp space height
- `int iTmpStates`
tmp states count
- `bool bApplied`
is this widget applied?
- `bool bInitDone`
is app core init?
- `unsigned int iOutputSizeX`
output space width
- `unsigned int iOutputSizeY`
output space height
- `unsigned int iOutputMapPosX`
output space map coord x
- `unsigned int iOutputMapPosY`
output space map coord y
- `bool bOutputApplied`
is output space widget applied?

3.36.1 Detailed Description

input ca space settings

3.36.2 Constructor & Destructor Documentation

3.36.2.1 CWidgetInput::CWidgetInput (QWidget * *parent* = 0)

class constructor

Parameters:

**parent* pointer to parent widget

3.36.2.2 CWidgetInput::~CWidgetInput ()

class destructor

3.36.3 Member Function Documentation

3.36.3.1 void CWidgetInput::ApplyPressed () [private, slot]

when apply is pressed disable all functional tools in widget

3.36.3.2 void CWidgetInput::CancelPressed () [private, slot]

when cancel is pressed, enable all functional tools in widget

3.36.3.3 void CWidgetInput::changeEvent (QEvent * *e*) [protected]

change event happened

Parameters:

e event

3.36.3.4 bool CWidgetInput::CheckPos (int *x*, int *y*) [private]

checks if is possible to map input grid to output from given position output widget was applied before input

Parameters:

x x coord of mapping

y y coord of mapping

3.36.3.5 bool CWidgetInput::CheckSize (int *x*, int *y*) [private]

checks if is input grid is smaller or same as output output widget was applied before input

Parameters:

x width of input grid

y height of input grid

3.36.3.6 void CWidgetInput::DeleteInputGui () [private]

deletes input space scene

3.36.3.7 CGraphicsScene * CWidgetInput::GetGrid ()

returns pointer to input space scene

3.36.3.8 void CWidgetInput::ImportPressed () [private, slot]

imports spce from png file

3.36.3.9 bool CWidgetInput::IsApplied ()

is input widget applied?

3.36.3.10 void CWidgetInput::ReinitInputView () [private]

reinitis input space scene used when size of scene was changed

3.36.3.11 void CWidgetInput::SetInitDone (bool *id*)

button init changed state - enable/disable any changes in widget

3.36.3.12 void CWidgetInput::SetInputGridSize (int *x*, int *y*)

sets input space size - used as extern call - when settings are loaded from file

Parameters:

x new input space width

y new input space height

3.36.3.13 void CWidgetInput::SetInputGridSizeX (int *x*)

sets input space width - used as extern call - when settings are loaded from file

Parameters:

x new input space width

3.36.3.14 void CWidgetInput::SetInputGridSizeY (int *y*)

sets input space height - used as extern call - when settings are loaded from file

Parameters:

y new input space height

3.36.3.15 void CWidgetInput::SetOutputApplied (bool *a*)

sets, when output widget apply state changed

Parameters:

a output widget apply state

3.36.3.16 void CWidgetInput::SetOutputMapPosX (int *x*)

sets output space x mapping position

Parameters:

x x mapping coord

3.36.3.17 void CWidgetInput::SetOutputMapPosY (int *y*)

sets output space y mapping position

Parameters:

y y mapping coord

3.36.3.18 void CWidgetInput::SetOutputSizeX (int *x*)

sets output space width

Parameters:

x output space width

3.36.3.19 void CWidgetInput::SetOutputSizeY (int *y*)

sets output space height

Parameters:

y output space height

3.36.3.20 void CWidgetInput::SetStatesCount (int *s*)

sets states count - used as extern call - when settings are loaded from file

Parameters:

s new states count

3.36.3.21 void CWidgetInput::SetupInputView () [private]

creates input space scene used when application starts

3.36.3.22 void CWidgetInput::SizeChangedX (int *x*) [private, slot]

spinbox with space width changed value

Parameters:

x new space width

3.36.3.23 void CWidgetInput::SizeChangedY (int *y*) [private, slot]

spinbox with space height changed value

Parameters:

y new space height

3.36.3.24 void CWidgetInput::SizeChangeUsed () [private, slot]

space size change was applied - set button was pressed

3.36.3.25 void CWidgetInput::StatesChandeUsed () [private, slot]

count of space was changed - set button was pressed

3.36.3.26 void CWidgetInput::StatesChanged (int *s*) [private, slot]

spinbox with states count changed value

Parameters:

s new states count

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

- WidgetInput.h
- WidgetInput.cpp

3.37 CWidgetOutput Class Reference

```
#include <WidgetOutput.h>
```

Signals

- void **SignalWidgetOutputApplied** (bool)

Public Member Functions

- [CWidgetOutput](#) (QWidget *parent=0)
- [~CWidgetOutput](#) ()
- int [GetOutputGridSizeX](#) ()
- int [GetOutputGridSizeY](#) ()
- int [GetOutputMapPosX](#) ()
- int [GetOutputMapPosY](#) ()
- int [GetOutputArrayType](#) ()
- bool [IsApplied](#) ()
- void [SetInitDone](#) (bool id)
- void [SetInputSizeX](#) (int x)
- void [SetInputSizeY](#) (int y)
- void [SetInputApplied](#) (bool a)
- void [SetOutputGridSizeX](#) (int x)
- void [SetOutputGridSizeY](#) (int y)
- void [SetOutputMapPosX](#) (int x)
- void [SetOutputMapPosY](#) (int y)
- void [SetOutputArrayType](#) (int st)

Protected Member Functions

- void [changeEvent](#) (QEvent *e)

Private Slots

- void [SizeChangedX](#) (int x)
- void [SizeChangedY](#) (int y)
- void [PosChangedX](#) (int x)
- void [PosChangedY](#) (int y)
- void [ArrayTypeChanged](#) (int at)
- void [ApplyPressed](#) ()
- void [CancelPressed](#) ()

Private Member Functions

- bool [CheckSize](#) (int x, int y)
- bool [CheckPos](#) (int x, int y)

Private Attributes

- `Ui::CWidgetOutput * ui`
pointer to widget (gui of this class)
- unsigned int `iSizeX`
output space width
- unsigned int `iSizeY`
output space height
- unsigned int `iMapPosX`
mapping coord x
- unsigned int `iMapPosY`
mapping coord y
- int `iOutputArrayType`
output space type - 2d grid || torus
- unsigned int `iInputSizeX`
input space width
- unsigned int `iInputSizeY`
input space height
- bool `bInputApplied`
is input widget applied?
- bool `bApplied`
is this widget applied?
- bool `bInitDone`
is app core init?

3.37.1 Detailed Description

output sapce settings

3.37.2 Constructor & Destructor Documentation

3.37.2.1 CWidgetOutput::CWidgetOutput (QWidget * *parent* = 0)

class constructor

Parameters:

**parent* pointer to parent wodget

3.37.2.2 CWidgetOutput::~~CWidgetOutput ()

class destructor

3.37.3 Member Function Documentation

3.37.3.1 void CWidgetOutput::ApplyPressed () [private, slot]

when apply is pressed disable all functional tools in widget

3.37.3.2 void CWidgetOutput::ArrayTypeChanged (int *at*) [private, slot]

ca space type was changed

Parameters:

at new array type from combobox

3.37.3.3 void CWidgetOutput::CancelPressed () [private, slot]

when cancel is pressed, enable all functional tools in widget

3.37.3.4 void CWidgetOutput::changeEvent (QEvent * *e*) [protected]

change event happened

Parameters:

e event

3.37.3.5 bool CWidgetOutput::CheckPos (int *x*, int *y*) [private]

checks if is possible to map input grid to output from given position input widget was applied before output

Parameters:

x x coord of mapping

y y coord of mapping

3.37.3.6 bool CWidgetOutput::CheckSize (int *x*, int *y*) [private]

checks if is input grid is smaller or same as output input widget was applied before output

Parameters:

x width of input grid

y height of input grid

3.37.3.7 int CWidgetOutput::GetOutputArrayType ()

returns output space type

3.37.3.8 int CWidgetOutput::GetOutputGridSizeX ()

returns space width

3.37.3.9 int CWidgetOutput::GetOutputGridSizeY ()

returns space height

3.37.3.10 int CWidgetOutput::GetOutputMapPosX ()

returns x mapping coord

3.37.3.11 int CWidgetOutput::GetOutputMapPosY ()

returns y mapping coord

3.37.3.12 bool CWidgetOutput::IsApplied ()

is outpu widget applied?

3.37.3.13 void CWidgetOutput::PosChangedX (int *x*) [private, slot]

x mapping position was changed

Parameters:

x x mapping position from spinbox

3.37.3.14 void CWidgetOutput::PosChangedY (int *y*) [private, slot]

y mapping coord was changed

Parameters:

y y mapping coord from spinbox

3.37.3.15 void CWidgetOutput::SetInitDone (bool *id*)

button init changed state - enable/disable any changes in widget

3.37.3.16 void CWidgetOutput::SetInputApplied (bool *a*)

sets input widget apply state

Parameters:

a input widget apply

3.37.3.17 void CWidgetOutput::SetInputSizeX (int *x*)

sets input space width

Parameters:

x input space width

3.37.3.18 void CWidgetOutput::SetInputSizeY (int *y*)

sets input space height

Parameters:

y input space height

3.37.3.19 void CWidgetOutput::SetOutputArrayType (int *st*)

sets output space type - used as extern call - when settings are loaded from file

Parameters:

st new output space type

3.37.3.20 void CWidgetOutput::SetOutputGridSizeX (int *x*)

sets output space width - used as extern call - when settings are loaded from file

Parameters:

x new output space width

3.37.3.21 void CWidgetOutput::SetOutputGridSizeY (int *y*)

sets output space height - used as extern call - when settings are loaded from file

Parameters:

y new output space height

3.37.3.22 void CWidgetOutput::SetOutputMapPosX (int x)

sets x mapping coord - used as extern call - when settings are loaded from file

Parameters:

x new x mapping coord

3.37.3.23 void CWidgetOutput::SetOutputMapPosY (int y)

sets y mapping coord - used as extern call - when settings are loaded from file

Parameters:

y new y mapping coord

3.37.3.24 void CWidgetOutput::SizeChangedX (int x) [private, slot]

output sapce width was changed

Parameters:

x output space width from spinbox

3.37.3.25 void CWidgetOutput::SizeChangedY (int y) [private, slot]

output space height was changed

Parameters:

y output space height from spinbox

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

- WidgetOutput.h
- WidgetOutput.cpp

3.38 CWidgetRunCA Class Reference

```
#include <WidgetRunCA.h>
```

Public Member Functions

- [CWidgetRunCA](#) (QWidget *parent=0)
- [~CWidgetRunCA](#) ()
- [CGraphicsScene](#) * [GetGrid](#) ()
- void [InitRunGrid](#) (int sizeX, int sizeY, int states)
- void [DeleteRun](#) ()

Protected Member Functions

- void [changeEvent](#) (QEvent *e)

Private Member Functions

- void [SetRunGridSizeX](#) (int x)
- void [SetRunGridSizeY](#) (int y)
- void [SetRunGridStates](#) (int s)
- void [SetupRunGridView](#) ()
- void [DeleteRunGridView](#) ()

Private Attributes

- Ui::CWidgetRunCA * [ui](#)
pointer to widget (gui of this class)
- [CGraphicsScene](#) * [grid](#)
pointer to scene class
- [CGraphicsView](#) * [view](#)
pointer to view class
- unsigned int [iSizeX](#)
space width
- unsigned int [iSizeY](#)
space height
- unsigned int [iStates](#)
ca states count

3.38.1 Detailed Description

simulator's ca space widget - gui

3.38.2 Constructor & Destructor Documentation

3.38.2.1 CWidgetRunCA::CWidgetRunCA (QWidget * *parent* = 0)

class constructor

Parameters:

**parent* pointer to parent widget

3.38.2.2 CWidgetRunCA::~~CWidgetRunCA ()

class destructor

3.38.3 Member Function Documentation

3.38.3.1 void CWidgetRunCA::changeEvent (QEvent * *e*) [protected]

change event happened

Parameters:

e event

3.38.3.2 void CWidgetRunCA::DeleteRun ()

public fc, which calls fc for deleting space

3.38.3.3 void CWidgetRunCA::DeleteRunGridView () [private]

deletes simulator space

3.38.3.4 CGraphicsScene * CWidgetRunCA::GetGrid ()

returns pointer tp ca space scene

3.38.3.5 void CWidgetRunCA::InitRunGrid (int *sizeX*, int *sizeY*, int *states*)

inits simulator space

Parameters:

sizeX space width

sizeY space height

states count of states

3.38.3.6 void CWidgetRunCA::SetRunGridSizeX (int *x*) [private]

sets simulator space width

Parameters:

x space width

3.38.3.7 void CWidgetRunCA::SetRunGridSizeY (int *y*) [private]

sets simulator space height

Parameters:

y space height

3.38.3.8 void CWidgetRunCA::SetRunGridStates (int *s*) [private]

sets simulator sapce states count

Parameters:

s states count

3.38.3.9 void CWidgetRunCA::SetupRunGridView () [private]

creates simulator space

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

- WidgetRunCA.h
- WidgetRunCA.cpp

3.39 CWidgetRunEvo Class Reference

```
#include <WidgetRunEvo.h>
```

Public Member Functions

- [CWidgetRunEvo](#) (QWidget *parent=0)
- [~CWidgetRunEvo](#) ()
- void [SetGenerationID](#) (int gen)
- void [SetChromosomeID](#) (int chrom)
- void [SetAncestorsCount](#) (int anc)
- void [SetFitnessMax](#) (double fit)
- void [SetDifferentiationMin](#) (int diff)
- void [SetFitMaxStepCA](#) (int step)
- void [SetSameChromosomesCount](#) (int chroms)
- void [SetSameChromosomesGenerationAvarage](#) (double avarage)
- void [InitRun](#) ()
- void [DeleteRun](#) ()

Protected Member Functions

- void [changeEvent](#) (QEvent *e)

Private Attributes

- Ui::CWidgetRunEvo * [ui](#)
pointer to widget (gui of this class)

3.39.1 Detailed Description

simulator current chromosome widget

3.39.2 Constructor & Destructor Documentation

3.39.2.1 CWidgetRunEvo::CWidgetRunEvo (QWidget * *parent* = 0)

class constructor

Parameters:

**parent* pointer to parent widget

3.39.2.2 CWidgetRunEvo::~~CWidgetRunEvo ()

class destructor

3.39.3 Member Function Documentation

3.39.3.1 void CWidgetRunEvo::changeEvent (QEvent * *e*) [protected]

change event happened

Parameters:

e event

3.39.3.2 void CWidgetRunEvo::DeleteRun ()

clears lineEdits

3.39.3.3 void CWidgetRunEvo::InitRun ()

inits lineEdits

3.39.3.4 void CWidgetRunEvo::SetAncestorsCount (int *anc*)

sets ancestord count into lineEdit in gui

Parameters:

anc ancestors count

3.39.3.5 void CWidgetRunEvo::SetChromosomeID (int *chrom*)

sets chromosome id into lineEdit in gui

Parameters:

chrom chromosome id

3.39.3.6 void CWidgetRunEvo::SetDifferentionMin (int *diff*)

sets differentiation min into lineEdit in gui

Parameters:

diff differentiation min

3.39.3.7 void CWidgetRunEvo::SetFitMaxStepCA (int *step*)

sets steps ca into lineEdit in gui

Parameters:

step steps ca

3.39.3.8 void CWidgetRunEvo::SetFitnessMax (double *fit*)

sets fitness max into lineEdit in gui

Parameters:

fit fitness max

3.39.3.9 void CWidgetRunEvo::SetGenerationID (int *gen*)

sets generation into lineEdit in gui

Parameters:

gen generations

3.39.3.10 void CWidgetRunEvo::SetSameChromosomesCount (int *chroms*)

sets same chromosomes count into lineEdit in gui

Parameters:

chroms same chromosomes count

3.39.3.11 void CWidgetRunEvo::SetSameChromosomesGenerationAvarage (double *avarage*)

sets same chromosomes generation avarage into lineEdit in gui

Parameters:

avarage same chromosomes generation avarage

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

- WidgetRunEvo.h
- WidgetRunEvo.cpp

3.40 CWindowMain Class Reference

```
#include <WindowMain.h>
```

Public Member Functions

- [CWindowMain](#) (QWidget *parent=0)
- [~CWindowMain](#) ()

Protected Member Functions

- void [changeEvent](#) (QEvent *e)

Private Slots

- void [WidgetInputApplied](#) (bool a)
- void [WidgetOutputApplied](#) (bool a)
- void [WidgetEvolutionApplied](#) (bool a)
- void [WidgetExportApplied](#) (bool a)
- void [InitPressed](#) ()
- void [DeletePressed](#) ()
- void [EvoStartPressed](#) ()
- void [RunPressed](#) ()
- void [StepPressed](#) ()
- void [StopPressed](#) ()
- void [TerminatePressed](#) ()
- void [EvolutionInit](#) ()
- void [EvolutionRunning](#) (bool r)
- void [CellularAutomatonRunning](#) (bool r)
- void [CellularAutomatonStep](#) ()
- void [SimulationDelete](#) ()
- void [CoreCheckInitDone](#) (int ccid)
- void [CoreDataMapToGui](#) ()
- void [MenuActionAbout](#) ()
- void [MenuActionExportRunConfig](#) ()
- void [MenuActionImportRunConfig](#) ()
- void [GuiDisplayCaMove](#) ()

Private Member Functions

- bool [InitCore](#) ()
- void [DeleteCore](#) ()
- void [CheckSettingApplyValidity](#) ()
- bool [CheckUiLayouts](#) ()

Private Attributes

- `Ui::CWindowMain * ui`
pointer to widget (gui of this class)
- `QTimer * timer`
pointer to timer class
- `CWidgetInput * widgetI`
pointer to input widget
- `CWidgetOutput * widgetO`
pointer to output widget
- `CWidgetEvolution * widgetE`
pointer to evolution widget
- `CWidgetExport * widgetEx`
pointer to export widget
- `CWidgetRunCA * widgetRCA`
pointer to run-ca widget
- `CWidgetRunEvo * widgetREvo`
pointer to run-evo widget
- `CCore * core`
pointer to core class
- `CExportConfig exportConfig`
class for exporting all settings
- `CImportConfig importConfig`
class for importing all settings
- `bool bWidgetInputApplied`
is input widget applied?
- `bool bWidgetOutputApplied`
is output widget applied?
- `bool bWidgetEvolutionApplied`
is evolution wodget applied?
- `bool bWidgetExportApplied`
is export widget applied?
- `bool bSimInitDone`
is core init done?

- bool [bEvoRunning](#)
is evolution running?
- bool [bCaRunning](#)
is simulator running?
- bool [bOutputDone](#)
is some output done?
- unsigned int [iGuiDisplayModeCaTimeout](#)
simulator animation timeout

3.40.1 Detailed Description

main window off app, this class use all other gui classes

3.40.2 Constructor & Destructor Documentation

3.40.2.1 CWindowMain::CWindowMain (QWidget * *parent* = 0)

class constructor

Parameters:

**parent* pointer to parent widget

3.40.2.2 CWindowMain::~~CWindowMain ()

class destructor

3.40.3 Member Function Documentation

3.40.3.1 void CWindowMain::CellularAutomatonRunning (bool *r*) [private, slot]

reaction to run/stop button - starts animation in simualtor

Parameters:

r is animation running?

3.40.3.2 void CWindowMain::CellularAutomatonStep () [private, slot]

reaction to step button

3.40.3.3 void CWindowMain::changeEvent (QEvent * *e*) [protected]

change event happened

Parameters:

e event

3.40.3.4 void CWindowMain::CheckSettingApplyValidity () [private]

checks, if all widgets are applied

3.40.3.5 bool CWindowMain::CheckUiLayouts () [private]

checks if all widget layout was correctly allocated

3.40.3.6 void CWindowMain::CoreCheckInitDone (int *ccid*) [private, slot]

checks core initialization state

3.40.3.7 void CWindowMain::CoreDataMapToGui () [private, slot]

maps data from [CCore](#) class into widgets/gui

3.40.3.8 void CWindowMain::DeleteCore () [private]

deletes [CCore](#) class

3.40.3.9 void CWindowMain::DeletePressed () [private, slot]

delete button is pressed

3.40.3.10 void CWindowMain::EvolutionInit () [private, slot]

reaction to init button - inits evolution

3.40.3.11 void CWindowMain::EvolutionRunning (bool *r*) [private, slot]

reaction to evolve/delete button - starts evolution

Parameters:

r is evolution running?

3.40.3.12 void CWindowMain::EvoStartPressed () [private, slot]

evolve button is pressed

3.40.3.13 void CWindowMain::GuiDisplayCaMove () [private, slot]

i am not sure, but i think, that this is called when data from [CCore](#) are written into widgets, so it's possible to write new data into [CCore](#) from app kernel

3.40.3.14 bool CWindowMain::InitCore () [private]

inits app core - creates [CCore](#) class, writes all settings into it [CCore](#) then creates app kernel class - [CThread-Core](#)

3.40.3.15 void CWindowMain::InitPressed () [private, slot]

init button os pressed

3.40.3.16 void CWindowMain::MenuActionAbout () [private, slot]

about from menu is pressed - show about dialog

3.40.3.17 void CWindowMain::MenuActionExportRunConfig () [private, slot]

export from menu is pressed - export all settings

3.40.3.18 void CWindowMain::MenuActionImportRunConfig () [private, slot]

import from menu is pressed - import all settings

3.40.3.19 void CWindowMain::RunPressed () [private, slot]

run button is pressed start animation in simulator

3.40.3.20 void CWindowMain::SimulationDelete () [private, slot]

reaction to delete button

3.40.3.21 void CWindowMain::StepPressed () [private, slot]

step button is pressed do 1 step of ca in simulator

3.40.3.22 void CWindowMain::StopPressed () [private, slot]

stop button is pressed interrupt animation in simulator

3.40.3.23 void CWindowMain::TerminatePressed () [private, slot]

terminate button is pressed terminate simulation

3.40.3.24 void CWindowMain::WidgetEvolutionApplied (bool *a*) [private, slot]

sets evolution widget apply state

Parameters:

a evolution widget apply state

3.40.3.25 void CWindowMain::WidgetExportApplied (bool *a*) [private, slot]

sets export widget apply state

Parameters:

a export widget apply state

3.40.3.26 void CWindowMain::WidgetInputApplied (bool *a*) [private, slot]

sets input widget apply state

Parameters:

a input widget apply state

3.40.3.27 void CWindowMain::WidgetOutputApplied (bool *a*) [private, slot]

sets output widget apply state

Parameters:

a output widget apply state

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

- WindowMain.h
- WindowMain.cpp

3.41 stArray Struct Reference

```
#include <Array2d.h>
```

Public Attributes

- unsigned int [sizeX](#)
array width
- unsigned int [sizeY](#)
array height
- BYTE ** [grid](#)
pointer to 2d array

3.41.1 Detailed Description

2d array struct

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

- Array2d.h

3.42 stGeneInstruction Struct Reference

```
#include <GenomeType2_Ins.h>
```

Public Attributes

- BYTE [instruction](#)
instruction type
- BYTE [preCon](#)
precondition
- BYTE [preConLogic](#)
precondition logic
- BYTE [postCon](#)
postcondition

3.42.1 Detailed Description

this struct contain instruction based gene

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

- GenomeType2_Ins.h

3.43 stThreadCoreDataGA Struct Reference

```
#include <ThreadCore.h>
```

Public Attributes

- int [iGeneration](#)
generation id of send genome
- int [iChromosome](#)
id of send genome in population
- int [iFitnessMax](#)
maximum fitness
- double [dFitnessMaxNorm](#)
normalized max fitness
- int [iDifferentiationMin](#)
minimum differentiation
- int [iFitMaxStepCA](#)
step of ca with maximum fitness
- int [iAncestorsCount](#)
number of ancestors
- int [iSameGenomesCount](#)
count of genomes with same fitness
- double [dSameGenomesGenerationAvarage](#)
avarage generation of "same" genomes

3.43.1 Detailed Description

this struct is used for sending GA data from this compute class into gui
The documentation for this struct was generated from the following file:

- ThreadCore.h

Index

- ~CArray2d
 - CArray2d, [6](#)
- ~CCellularAutomata
 - CCellularAutomata, [10](#)
- ~CCore
 - CCore, [24](#)
- ~CExportConfig
 - CExportConfig, [32](#)
- ~CExportLog
 - CExportLog, [39](#)
- ~CGeneticAlgorithm
 - CGeneticAlgorithm, [47](#)
- ~CGenome
 - CGenome, [54](#)
- ~CGraphicsScene
 - CGraphicsScene, [82](#)
- ~CSpace
 - CSpace, [103](#)
- ~CThreadCore
 - CThreadCore, [112](#)
- ~CWidgetEvolution
 - CWidgetEvolution, [119](#)
- ~CWidgetExport
 - CWidgetExport, [127](#)
- ~CWidgetInput
 - CWidgetInput, [133](#)
- ~CWidgetOutput
 - CWidgetOutput, [138](#)
- ~CWidgetRunCA
 - CWidgetRunCA, [144](#)
- ~CWidgetRunEvo
 - CWidgetRunEvo, [146](#)
- ~CWindowMain
 - CWindowMain, [151](#)
- _Fitness_01
 - CFitness, [42](#)
- _Fitness_02
 - CFitness, [42](#)
- _MoveA
 - CFitness, [42](#)
- _Mutation_01
 - CMutation, [94](#)
- _Mutation_02
 - CMutation, [94](#)
- _Mutation_Ins_01
 - CMutation, [95](#)
- _Selection_01
 - CSelection, [100](#)
- _Selection_02
 - CSelection, [100](#)
- AllocMemory
 - CArray2d, [7](#)
- ApplyPressed
 - CWidgetEvolution, [119](#)
 - CWidgetExport, [127](#)
 - CWidgetInput, [133](#)
 - CWidgetOutput, [139](#)
- ArrayTypeChanged
 - CWidgetOutput, [139](#)
- at
 - CRulesTable, [98](#)
 - CSpace, [103](#)
- atGrid
 - CSpace, [103](#)
- atTorus
 - CSpace, [104](#)
- boundingRect
 - CGraphicsItem, [75](#)
- BuildScene
 - CGraphicsScene, [82](#)
- CalculateIndexGrid
 - CTFunction, [106](#)
- CalculateIndexTorus
 - CTFunction, [107](#)
- CancelPressed
 - CWidgetEvolution, [119](#)
 - CWidgetExport, [127](#)
 - CWidgetInput, [133](#)
 - CWidgetOutput, [139](#)
- CArray2d, [5](#)
 - ~CArray2d, [6](#)
 - AllocMemory, [7](#)
 - CArray2d, [6](#)
 - ClearArray, [7](#)
 - DeleteMemory, [7](#)
 - GetErrorFlag, [7](#)
 - GetHeight, [7](#)

- GetWidth, 7
- InitArray, 7
- operator(), 8
- operator=, 8
- CCellularAutomata, 9
 - ~CCellularAutomata, 10
 - CCellularAutomata, 10
 - DeleteSpace, 10
 - DeleteSpaceInit, 10
 - GetErrorFlag, 10
 - GetInitSpace, 10
 - GetRulesTable, 11
 - GetSpace, 11
 - GetStepsDone, 11
 - GoL, 11
 - InitMemory, 11
 - InitSpace, 11
 - IsInitDone, 11
 - ReInit, 11
 - SetConfigCore, 11
 - Step, 11
 - StepGoL, 12
- CConfigCore, 13
 - CConfigCore, 15
 - GetCrossoverCount, 15
 - GetCrossoverProbability, 15
 - GetDefaultState, 15
 - GetEvolutionRepetitionsCount, 16
 - GetExportFileModeCa, 16
 - GetExportFileModeGa, 16
 - GetExportFilePath, 16
 - GetExportLogCore, 16
 - GetGenerationsCount, 16
 - GetGenomeType, 16
 - GetGuiDisplayModeCA, 16
 - GetGuiDisplayModeCATimeout, 16
 - GetImportGenomeFile, 16
 - GetMoveDirection, 16
 - GetMoveDistance, 17
 - GetMutationCount, 17
 - GetMutationProbability, 17
 - GetPopulationSize, 17
 - GetSpaceSizeX, 17
 - GetSpaceSizeY, 17
 - GetSpaceType, 17
 - GetStatesCount, 17
 - GetStepsCountCA, 17
 - IsImportGenomeEnabledReevolve, 17
 - IsImportGenomeEnabledSimulation, 17
 - SetCrossoverCount, 18
 - SetCrossoverProbability, 18
 - SetDefaultState, 18
 - SetEvolutionRepetitionsCount, 18
 - SetExportFileModeCa, 18
 - SetExportFileModeGa, 18
 - SetExportFilePath, 18
 - SetExportLogCore, 19
 - SetGenerationsCount, 19
 - SetGenomeType, 19
 - SetGuiDisplayModeCA, 19
 - SetGuiDisplayModeCATimeout, 19
 - SetImportGenomeEnabledReevolve, 19
 - SetImportGenomeEnabledSimulation, 20
 - SetImportGenomeFile, 20
 - SetMoveDirection, 20
 - SetMoveDistance, 20
 - SetMutationCount, 20
 - SetMutationProbability, 20
 - SetPopulationSize, 21
 - SetSpaceSizeX, 21
 - SetSpaceSizeY, 21
 - SetSpaceType, 21
 - SetStatesCount, 21
 - SetStepsCountCA, 21
- CCore, 23
 - ~CCore, 24
 - CCore, 24
 - GetConfigCore, 25
 - GetErrorFlag, 25
 - GetSimCoreAncestorsCount, 25
 - GetSimCoreChromosomeID, 25
 - GetSimCoreDataStruct, 25
 - GetSimCoreDifferentiationMin, 25
 - GetSimCoreFitMaxStepCA, 25
 - GetSimCoreFitnessMax, 25
 - GetSimCoreFitnessMaxNorm, 25
 - GetSimCoreGenerationID, 25
 - GetSimCoreSameChromosomesCount, 25
 - GetSimCoreSameChromosomesGenerationAvarage, 26
 - GetSpace, 26
 - InitGuiCore, 26
 - IsInitDone, 26
 - SetCaRunTermination, 26
 - SetCoreDataExpiration, 26
 - SetSimulationRunning, 26
 - ShowDebugConfigCore, 26
 - StartThreadCore, 26
 - ThreadCoreCheckCoreInitDone, 27
 - ThreadCoreErrorSlot, 27
 - ThreadCoreNewDataAvailable, 27
 - ThreadCoreState, 27
- CCrossover, 28
 - CCrossover, 28
 - Crossover, 28
 - GetErrorFlag, 28
 - SetConfigCore, 28
- CellularAutomatonRunning

- CWindowMain, 151
- CellularAutomatonStep
 - CWindowMain, 151
- CExportCA, 30
 - CExportCA, 30
 - ExportCellularAutomaton, 30
 - ExportCellularAutomatonInput, 30
 - GetErrorFlag, 31
 - SetConfigCore, 31
- CExportConfig, 32
 - ~CExportConfig, 32
 - CExportConfig, 32
 - ExportConfig, 33
 - GetErrorFlag, 33
 - SetIODevice, 33
 - WriteDocEnd, 33
 - WriteDocStart, 33
 - WriteSettingsEvolution, 33
 - WriteSettingsExport, 33
 - WriteSettingsInput, 33
 - WriteSettingsInputCASpace, 34
 - WriteSettingsOutput, 34
- CExportGA, 35
 - CExportGA, 35
 - ExportChromosome, 36
 - ExportGeneration, 36
 - GetErrorFlag, 36
 - SetConfigCore, 36
 - SetIODevice, 36
 - WriteDocEnd, 36
 - WriteDocStart, 36
 - WriteGenome, 36
- CExportLog, 38
 - ~CExportLog, 39
 - CExportLog, 39
 - CloseLogFile, 39
 - OpenLogFile, 39
 - SetFileName, 39
 - SetFilePath, 39
 - ShowMessageBox, 39
 - WriteErrorLog, 40
 - WriteInfoLog, 40
- CFitness, 41
 - _Fitness_01, 42
 - _Fitness_02, 42
 - _MoveA, 42
 - CFitness, 42
 - Fitness, 43
 - GetCornerPosX, 43
 - GetCornerPosY, 43
 - GetErrorFlag, 43
 - IdentifyCorners, 43
 - IsLiveOrganismInSpace, 43
 - SetConfigCore, 44
- CGeneticAlgorithm, 45
 - ~CGeneticAlgorithm, 47
 - CGeneticAlgorithm, 47
 - ClearVectors, 47
 - CreateNextGeneration, 47
 - Crossover, 47
 - ExistsGenomeId, 47
 - Fitness, 47
 - GetActualPopulation, 48
 - GetActualPopulationSize, 48
 - GetBestGenome, 48
 - GetCornerPosX, 48
 - GetCornerPosY, 48
 - GetErrorFlag, 48
 - GetGenome, 48
 - IdentifyCorners, 48
 - ImportGenomeToPopulation, 49
 - InitGenonetypeStandard, 49
 - InitGenotype, 49
 - InitGenotypeInstruction, 49
 - InitGenotypeReevolve, 49
 - Mutation, 49
 - RefillPopulation, 49
 - RenumberActualGeneration, 49
 - Selection, 50
 - SetConfigCore, 50
- CGenome, 51
 - ~CGenome, 54
 - CGenome, 53
 - GetAncestorsCount, 54
 - GetDifferentiationMin, 54
 - GetErrorFlag, 54
 - GetFitness, 54
 - GetFitnessMax, 54
 - GetFitnessMaxNorm, 54
 - GetFitnessMaxStepCA, 54
 - GetFitnessNorm, 54
 - GetGene, 55
 - GetGenomeType, 55
 - GetInstructionGenome, 55
 - GetMutatedGenes, 55
 - GetMutatedGenesAllAncestors, 55
 - GetMutatedGenesTotal, 55
 - GetParentGenerationId, 55
 - GetParentGenomeId, 55
 - GetThisGenerationId, 55
 - GetThisGenomeId, 55
 - GetThisInitGenerationId, 55
 - GetThisInitGenomeId, 56
 - SetAncestorsCount, 56
 - SetFitness, 56
 - SetFitnessNorm, 56
 - SetGene, 56
 - SetMutatedGenes, 56

- SetMutatedGenesTotal, 57
- SetParentGenomeId, 57
- SetThisGenomeId, 57
- SetThisInitGenomeId, 57
- CGenomeType2, 58
 - CGenomeType2, 58
 - GetGene, 58
 - GetGenome, 58
 - SetGene, 59
- CGenomeType2_Ins, 60
 - CGenomeType2_Ins, 60
 - CGenomeType2_Ins, 60
 - GetBitValue, 61
 - GetGene, 61
 - GetInstruction, 61
 - GetPostCondition, 61
 - GetPreCondition, 61
 - GetPreConditionBit, 61
 - GetPreConditionLogic, 62
 - GetPreConditionLogicBit, 62
 - SetBitValue, 62
 - SetInstruction, 62
 - SetPostCondition, 62
 - SetPreCondition, 63
 - SetPreConditionBit, 63
 - SetPreConditionLogic, 63
 - SetPreConditionLogicBit, 63
- CGenomeType2_Nbh5, 64
 - CGenomeType2_Nbh5, 64
 - CGenomeType2_Nbh5, 64
 - GetGene, 64
 - GetGenome, 64
 - SetGene, 65
- CGenomeType3, 66
 - CGenomeType3, 66
 - GetGene, 66
 - SetGene, 66
- CGenomeType3_Nbh5, 68
 - CGenomeType3_Nbh5, 68
 - CGenomeType3_Nbh5, 68
 - GetGene, 68
 - GetGenome, 68
 - SetGene, 69
- CGenomeType4, 70
 - CGenomeType4, 70
 - GetGene, 70
 - SetGene, 70
- CGenomeType4_Nbh5, 72
 - CGenomeType4_Nbh5, 72
 - CGenomeType4_Nbh5, 72
 - GetGene, 72
 - GetGenome, 72
 - SetGene, 73
- CGraphicsItem, 74
 - boundingRect, 75
 - CGraphicsItem, 75
 - GetPosX, 75
 - GetPosY, 75
 - GetState, 75
 - GetStatesCount, 75
 - IsEditable, 76
 - mouseMoveEvent, 76
 - mousePressEvent, 76
 - mouseReleaseEvent, 76
 - paint, 76
 - SetEditable, 76
 - SetState, 77
 - SetStatesCount, 77
- CGraphicsItemConfig, 78
 - CGraphicsItemConfig, 78
 - GetCellActState, 79
 - GetDefState, 79
 - GetEditable, 79
 - GetStatesCount, 79
 - SetCellActState, 79
 - SetDefState, 79
 - SetEditable, 79
 - SetStatesCount, 79
- CGraphicsScene, 81
 - ~CGraphicsScene, 82
 - BuildScene, 82
 - CGraphicsScene, 82
 - DeleteScene, 82
 - GetCell, 82
 - GetCellConfig, 83
 - GetHeight, 83
 - GetScene, 83
 - GetWidth, 83
 - SetConfigDefState, 83
 - SetConfigEditable, 83
 - SetConfigStatesCount, 83
 - SetHeight, 83
 - SetWidth, 84
- CGraphicsView, 85
 - CGraphicsView, 85
 - minimumSizeHint, 85
 - setupMatrix, 85
 - view, 85
 - zoomIn, 86
 - zoomOut, 86
- changeEvent
 - CWidgetEvolution, 120
 - CWidgetExport, 127
 - CWidgetInput, 133
 - CWidgetOutput, 139
 - CWidgetRunCA, 144
 - CWidgetRunEvo, 147
 - CWindowMain, 151

- CheckCaRunTermination
 - CThreadCore, 112
- CheckCorePointers
 - CThreadCore, 112
- CheckPos
 - CWidgetInput, 133
 - CWidgetOutput, 139
- CheckSettingApplyValidity
 - CWindowMain, 152
- CheckSize
 - CWidgetInput, 133
 - CWidgetOutput, 139
- CheckThreadLoopTermination
 - CThreadCore, 112
- CheckUiLayouts
 - CWindowMain, 152
- CImportConfig, 87
 - CImportConfig, 87
 - GetErrorFlag, 87
 - ImportConfig, 87
 - ReadDocEnd, 88
 - ReadDocStart, 88
 - ReadSettingsEvolution, 88
 - ReadSettingsExport, 88
 - ReadSettingsInput, 88
 - ReadSettingsInputCAspace, 88
 - ReadSettingsOutput, 88
 - SetIODevice, 89
- CImportGA, 90
 - CImportGA, 90
 - GetErrorFlag, 90
 - ImportChromosome, 90
 - ReadDocStart, 91
 - ReadGenome, 91
 - SetConfigCore, 91
 - SetIODevice, 91
 - WriteGenesIntoGenome, 91
- CInputCA, 92
 - CInputCA, 92
 - ExportCA, 92
 - GetErrorFlag, 93
 - ImportCA, 93
 - SetConfigCore, 93
- ClearArray
 - CArray2d, 7
- ClearTmpCAs
 - CThreadCore, 112
- ClearVectors
 - CGeneticAlgorithm, 47
- CloseLogFile
 - CExportLog, 39
- CMutation, 94
 - _Mutation_01, 94
 - _Mutation_02, 94
 - _Mutation_Ins_01, 95
- CMutation, 94
 - GetErrorFlag, 95
 - Mutation, 95
 - SetConfigCore, 95
- CoreCheckInitDone
 - CWindowMain, 152
- CoreDataMapToGui
 - CWindowMain, 152
- CRandom, 96
 - CRandom, 96
 - GetErrorFlag, 96
 - LCG, 96
 - Random, 97
 - Uniform, 97
 - UniformStdLib, 97
- CreateNextGeneration
 - CGeneticAlgorithm, 47
- Crossover
 - CCrossover, 28
 - CGeneticAlgorithm, 47
- CrossoverCountChanged
 - CWidgetEvolution, 120
- CrossoverProbChanged
 - CWidgetEvolution, 120
- CRulesTable, 98
 - at, 98
 - CRulesTable, 98
 - GetGenome, 98
 - SetConfigCore, 98
 - SetGenome, 99
- CSelection, 100
 - _Selection_01, 100
 - _Selection_02, 100
 - CSelection, 100
 - Selection, 101
 - SetConfigCore, 101
- CSPACE, 102
 - ~CSPACE, 103
 - at, 103
 - atGrid, 103
 - atTorus, 104
 - CSPACE, 103
 - GetErrorFlag, 104
 - GetHeight, 104
 - GetSpaceType, 104
 - GetWidth, 104
- CTFunction, 106
 - CalculateIndexGrid, 106
 - CalculateIndexTorus, 107
 - CTFunction, 106
 - NextSpace, 107
 - NextSpaceGenomeInstruction, 107
 - NextSpaceGenomeStandard, 107

- SetConfigCore, 107
- SetRulesTable, 107
- CThreadCore, 109
 - ~CThreadCore, 112
 - CheckCaRunTermination, 112
 - CheckCorePointers, 112
 - CheckThreadLoopTermination, 112
 - ClearTmpCAs, 112
 - CThreadCore, 112
 - FileExportCaInit, 112
 - FileExportCaSteps, 112
 - FileExportGa, 112
 - GetErrorFlag, 113
 - GetInitSpace, 113
 - GetSpace, 113
 - InitCore, 113
 - InitCoreCAMemory, 113
 - InitCoreCellularAutomata, 113
 - InitCoreGeneticAlgorithm, 113
 - InitExport, 113
 - InitTmpCAs, 113
 - IsInitDone, 113
 - ReinitCore, 113
 - run, 114
 - RunGenomeCaSimulation, 114
 - RunGuiMode, 114
 - SetConfigCore, 114
 - SetCoreDataExpiration, 114
 - SetCoreDataGA, 114
 - SetCoreDataValidity, 114
 - SetCoreSpace, 115
 - SetMutex, 115
 - SetSimulationRunning, 115
 - SetWaitCondition, 115
 - StoreGenomeDataForGui, 115
 - SyncDataWithCore, 115
 - TerminateCaRun, 116
 - TerminateThreadLoop, 116
 - WriteCAspaceFromCoreIntoCA, 116
 - WriteDataCAToCore, 116
 - WriteDataGAToCore, 116
- CWidgetEvolution, 117
 - ~CWidgetEvolution, 119
 - ApplyPressed, 119
 - CancelPressed, 119
 - changeEvent, 120
 - CrossoverCountChanged, 120
 - CrossoverProbChanged, 120
 - CWidgetEvolution, 119
 - DirectionChanged, 120
 - DistanceChanged, 120
 - GenerationsChanged, 120
 - GenomeTypeChanged, 120
 - GetCrossoverCount, 121
 - GetCrossoverProbability, 121
 - GetGenerationsCount, 121
 - GetGenomeType, 121
 - GetImportGenomeFile, 121
 - GetMoveDirection, 121
 - GetMoveDistance, 121
 - GetMutationCount, 121
 - GetMutationProbability, 121
 - GetPopulationSize, 121
 - GetRepetitionsCount, 121
 - GetStepsCount, 122
 - ImportFileButtonPressed, 122
 - ImportFileEnableReevolve, 122
 - ImportFileEnableSimulation, 122
 - IsApplied, 122
 - IsImportGenomeEnabledReevolve, 122
 - IsImportGenomeEnabledSimulation, 122
 - MutationCountChanged, 122
 - MutationProbChanged, 122
 - PopulationChanged, 123
 - RepetitionsChanged, 123
 - SetCrossoverCount, 123
 - SetCrossoverProbability, 123
 - SetGenerationsCount, 123
 - SetGenomeType, 123
 - SetInitDone, 124
 - SetMoveDirection, 124
 - SetMoveDistance, 124
 - SetMutationCount, 124
 - SetMutationProbability, 124
 - SetPopulationSize, 124
 - SetRepetitionsCount, 124
 - SetStepsCount, 125
 - StepsChanged, 125
- CWidgetExport, 126
 - ~CWidgetExport, 127
 - ApplyPressed, 127
 - CancelPressed, 127
 - changeEvent, 127
 - CWidgetExport, 127
 - FileExportCACHanged, 128
 - FileExportGACHanged, 128
 - FileExportPathChanged, 128
 - GetFileExportModeCA, 128
 - GetFileExportModeGA, 128
 - GetFileExportPath, 128
 - GetGuiDataDisplayModeCA, 128
 - GetGuiDataDisplayModeTimeout, 128
 - GuiModeCADisplayModeChanged, 128
 - GuiModeCATimeoutChanged, 129
 - IsApplied, 129
 - SetFileExportModeCA, 129
 - SetFileExportModeGA, 129
 - SetGuiDataDisplayModeCA, 129

- SetGuiDataDisplayModeTimeout, 129
- SetInitDone, 129
- CWidgetInput, 131
 - ~CWidgetInput, 133
 - ApplyPressed, 133
 - CancelPressed, 133
 - changeEvent, 133
 - CheckPos, 133
 - CheckSize, 133
 - CWidgetInput, 133
 - DeleteInputGui, 134
 - GetGrid, 134
 - ImportPressed, 134
 - IsApplied, 134
 - ReinitInputView, 134
 - SetInitDone, 134
 - SetInputGridSize, 134
 - SetInputGridSizeX, 134
 - SetInputGridSizeY, 134
 - SetOutputApplied, 135
 - SetOutputMapPosX, 135
 - SetOutputMapPosY, 135
 - SetOutputSizeX, 135
 - SetOutputSizeY, 135
 - SetStatesCount, 135
 - SetupInputView, 136
 - SizeChangedX, 136
 - SizeChangedY, 136
 - SizeChangeUsed, 136
 - StatesChandeUsed, 136
 - StatesChanged, 136
- CWidgetOutput, 137
 - ~CWidgetOutput, 138
 - ApplyPressed, 139
 - ArrayTypeChanged, 139
 - CancelPressed, 139
 - changeEvent, 139
 - CheckPos, 139
 - CheckSize, 139
 - CWidgetOutput, 138
 - GetOutputArrayType, 139
 - GetOutputGridSizeX, 140
 - GetOutputGridSizeY, 140
 - GetOutputMapPosX, 140
 - GetOutputMapPosY, 140
 - IsApplied, 140
 - PosChangedX, 140
 - PosChangedY, 140
 - SetInitDone, 140
 - SetInputApplied, 140
 - SetInputSizeX, 141
 - SetInputSizeY, 141
 - SetOutputArrayType, 141
 - SetOutputGridSizeX, 141
 - SetOutputGridSizeY, 141
 - SetOutputMapPosX, 141
 - SetOutputMapPosY, 141
 - SizeChangedX, 142
 - SizeChangedY, 142
- CWidgetRunCA, 143
 - ~CWidgetRunCA, 144
 - changeEvent, 144
 - CWidgetRunCA, 144
 - DeleteRun, 144
 - DeleteRunGridView, 144
 - GetGrid, 144
 - InitRunGrid, 144
 - SetRunGridSizeX, 144
 - SetRunGridSizeY, 145
 - SetRunGridStates, 145
 - SetupRunGridView, 145
- CWidgetRunEvo, 146
 - ~CWidgetRunEvo, 146
 - changeEvent, 147
 - CWidgetRunEvo, 146
 - DeleteRun, 147
 - InitRun, 147
 - SetAncestorsCount, 147
 - SetChromosomeID, 147
 - SetDifferentiationMin, 147
 - SetFitMaxStepCA, 147
 - SetFitnessMax, 147
 - SetGenerationID, 148
 - SetSameChromosomesCount, 148
 - SetSameChromosomesGenerationAvarage, 148
- CWindowMain, 149
 - ~CWindowMain, 151
 - CellularAutomatonRunning, 151
 - CellularAutomatonStep, 151
 - changeEvent, 151
 - CheckSettingApplyValidity, 152
 - CheckUiLayouts, 152
 - CoreCheckInitDone, 152
 - CoreDataMapToGui, 152
 - CWindowMain, 151
 - DeleteCore, 152
 - DeletePressed, 152
 - EvolutionInit, 152
 - EvolutionRunning, 152
 - EvoStartPressed, 152
 - GuiDisplayCaMove, 152
 - InitCore, 153
 - InitPressed, 153
 - MenuActionAbout, 153
 - MenuActionExportRunConfig, 153
 - MenuActionImportRunConfig, 153
 - RunPressed, 153

- SimulationDelete, [153](#)
- StepPressed, [153](#)
- StopPressed, [153](#)
- TerminatePressed, [153](#)
- WidgetEvolutionApplied, [153](#)
- WidgetExportApplied, [154](#)
- WidgetInputApplied, [154](#)
- WidgetOutputApplied, [154](#)
- DeleteCore
 - CWindowMain, [152](#)
- DeleteInputGui
 - CWidgetInput, [134](#)
- DeleteMemory
 - CArray2d, [7](#)
- DeletePressed
 - CWindowMain, [152](#)
- DeleteRun
 - CWidgetRunCA, [144](#)
 - CWidgetRunEvo, [147](#)
- DeleteRunGridView
 - CWidgetRunCA, [144](#)
- DeleteScene
 - CGraphicsScene, [82](#)
- DeleteSpace
 - CCellularAutomata, [10](#)
- DeleteSpaceInit
 - CCellularAutomata, [10](#)
- DirectionChanged
 - CWidgetEvolution, [120](#)
- DistanceChanged
 - CWidgetEvolution, [120](#)
- EvolutionInit
 - CWindowMain, [152](#)
- EvolutionRunning
 - CWindowMain, [152](#)
- EvoStartPressed
 - CWindowMain, [152](#)
- ExistsGenomeId
 - CGeneticAlgorithm, [47](#)
- ExportCA
 - CInputCA, [92](#)
- ExportCellularAutomaton
 - CExportCA, [30](#)
- ExportCellularAutomatonInput
 - CExportCA, [30](#)
- ExportChromosome
 - CExportGA, [36](#)
- ExportConfig
 - CExportConfig, [33](#)
- ExportGeneration
 - CExportGA, [36](#)
- FileExportCACHanged
 - CWidgetExport, [128](#)
- FileExportCAInit
 - CThreadCore, [112](#)
- FileExportCaSteps
 - CThreadCore, [112](#)
- FileExportGa
 - CThreadCore, [112](#)
- FileExportGACHanged
 - CWidgetExport, [128](#)
- FileExportPathChanged
 - CWidgetExport, [128](#)
- Fitness
 - CFitness, [43](#)
 - CGeneticAlgorithm, [47](#)
- GenerationsChanged
 - CWidgetEvolution, [120](#)
- GenomeTypeChanged
 - CWidgetEvolution, [120](#)
- GetActualPopulation
 - CGeneticAlgorithm, [48](#)
- GetActualPopulationSize
 - CGeneticAlgorithm, [48](#)
- GetAncestorsCount
 - CGenome, [54](#)
- GetBestGenome
 - CGeneticAlgorithm, [48](#)
- GetBitValue
 - CGenomeType2_Ins, [61](#)
- GetCell
 - CGraphicsScene, [82](#)
- GetCellActState
 - CGraphicsItemConfig, [79](#)
- GetCellConfig
 - CGraphicsScene, [83](#)
- GetConfigCore
 - CCore, [25](#)
- GetCornerPosX
 - CFitness, [43](#)
 - CGeneticAlgorithm, [48](#)
- GetCornerPosY
 - CFitness, [43](#)
 - CGeneticAlgorithm, [48](#)
- GetCrossoverCount
 - CConfigCore, [15](#)
 - CWidgetEvolution, [121](#)
- GetCrossoverProbability
 - CConfigCore, [15](#)
 - CWidgetEvolution, [121](#)
- GetDefaultState
 - CConfigCore, [15](#)
- GetDefState
 - CGraphicsItemConfig, [79](#)

- GetDifferentiationMin
 - CGenome, 54
- GetEditable
 - CGraphicsItemConfig, 79
- GetErrorFlag
 - CArray2d, 7
 - CCellularAutomata, 10
 - CCore, 25
 - CCrossover, 28
 - CExportCA, 31
 - CExportConfig, 33
 - CExportGA, 36
 - CFitness, 43
 - CGeneticAlgorithm, 48
 - CGenome, 54
 - CImportConfig, 87
 - CImportGA, 90
 - CInputCA, 93
 - CMutation, 95
 - CRandom, 96
 - CSPACE, 104
 - CThreadCore, 113
- GetEvolutionRepetitionsCount
 - CConfigCore, 16
- GetExportFileModeCa
 - CConfigCore, 16
- GetExportFileModeGa
 - CConfigCore, 16
- GetExportFilePath
 - CConfigCore, 16
- GetExportLogCore
 - CConfigCore, 16
- GetFileExportModeCA
 - CWidgetExport, 128
- GetFileExportModeGA
 - CWidgetExport, 128
- GetFileExportPath
 - CWidgetExport, 128
- GetFitness
 - CGenome, 54
- GetFitnessMax
 - CGenome, 54
- GetFitnessMaxNorm
 - CGenome, 54
- GetFitnessMaxStepCA
 - CGenome, 54
- GetFitnessNorm
 - CGenome, 54
- GetGene
 - CGenome, 55
 - CGenomeType2, 58
 - CGenomeType2_Ins, 61
 - CGenomeType2_Nbh5, 64
 - CGenomeType3, 66
 - CGenomeType3_Nbh5, 68
 - CGenomeType4, 70
 - CGenomeType4_Nbh5, 72
- GetGenerationsCount
 - CConfigCore, 16
 - CWidgetEvolution, 121
- GetGenome
 - CGeneticAlgorithm, 48
 - CGenomeType2, 58
 - CGenomeType2_Nbh5, 64
 - CGenomeType3_Nbh5, 68
 - CGenomeType4_Nbh5, 72
 - CRulesTable, 98
- GetGenomeType
 - CConfigCore, 16
 - CGenome, 55
 - CWidgetEvolution, 121
- GetGrid
 - CWidgetInput, 134
 - CWidgetRunCA, 144
- GetGuiDataDisplayModeCA
 - CWidgetExport, 128
- GetGuiDataDisplayModeTimeout
 - CWidgetExport, 128
- GetGuiDisplayModeCA
 - CConfigCore, 16
- GetGuiDisplayModeCATimeout
 - CConfigCore, 16
- GetHeight
 - CArray2d, 7
 - CGraphicsScene, 83
 - CSPACE, 104
- GetImportGenomeFile
 - CConfigCore, 16
 - CWidgetEvolution, 121
- GetInitSpace
 - CCellularAutomata, 10
 - CThreadCore, 113
- GetInstruction
 - CGenomeType2_Ins, 61
- GetInstructionGenome
 - CGenome, 55
- GetMoveDirection
 - CConfigCore, 16
 - CWidgetEvolution, 121
- GetMoveDistance
 - CConfigCore, 17
 - CWidgetEvolution, 121
- GetMutatedGenes
 - CGenome, 55
- GetMutatedGenesAllAncestors
 - CGenome, 55
- GetMutatedGenesTotal
 - CGenome, 55

- GetMutationCount
 - CConfigCore, 17
 - CWidgetEvolution, 121
- GetMutationProbability
 - CConfigCore, 17
 - CWidgetEvolution, 121
- GetOutputArrayType
 - CWidgetOutput, 139
- GetOutputGridSizeX
 - CWidgetOutput, 140
- GetOutputGridSizeY
 - CWidgetOutput, 140
- GetOutputMapPosX
 - CWidgetOutput, 140
- GetOutputMapPosY
 - CWidgetOutput, 140
- GetParentGenerationId
 - CGenome, 55
- GetParentGenomeId
 - CGenome, 55
- GetPopulationSize
 - CConfigCore, 17
 - CWidgetEvolution, 121
- GetPostCondition
 - CGenomeType2_Ins, 61
- GetPosX
 - CGraphicsItem, 75
- GetPosY
 - CGraphicsItem, 75
- GetPreCondition
 - CGenomeType2_Ins, 61
- GetPreConditionBit
 - CGenomeType2_Ins, 61
- GetPreConditionLogic
 - CGenomeType2_Ins, 62
- GetPreConditionLogicBit
 - CGenomeType2_Ins, 62
- GetRepetitionsCount
 - CWidgetEvolution, 121
- GetRulesTable
 - CCellularAutomata, 11
- GetScene
 - CGraphicsScene, 83
- GetSimCoreAncestorsCount
 - CCore, 25
- GetSimCoreChromosomeID
 - CCore, 25
- GetSimCoreDataStruct
 - CCore, 25
- GetSimCoreDifferentiationMin
 - CCore, 25
- GetSimCoreFitMaxStepCA
 - CCore, 25
- GetSimCoreFitnessMax
 - CCore, 25
- GetSimCoreFitnessMaxNorm
 - CCore, 25
- GetSimCoreGenerationID
 - CCore, 25
- GetSimCoreSameChromosomesCount
 - CCore, 25
- GetSimCoreSameChromosomesGenerationAvarage
 - CCore, 26
- GetSpace
 - CCellularAutomata, 11
 - CCore, 26
 - CThreadCore, 113
- GetSpaceSizeX
 - CConfigCore, 17
- GetSpaceSizeY
 - CConfigCore, 17
- GetSpaceType
 - CConfigCore, 17
 - CSpace, 104
- GetState
 - CGraphicsItem, 75
- GetStatesCount
 - CConfigCore, 17
 - CGraphicsItem, 75
 - CGraphicsItemConfig, 79
- GetStepsCount
 - CWidgetEvolution, 122
- GetStepsCountCA
 - CConfigCore, 17
- GetStepsDone
 - CCellularAutomata, 11
- GetThisGenerationId
 - CGenome, 55
- GetThisGenomeId
 - CGenome, 55
- GetThisInitGenerationId
 - CGenome, 55
- GetThisInitGenomeId
 - CGenome, 56
- GetWidth
 - CArray2d, 7
 - CGraphicsScene, 83
 - CSpace, 104
- GoL
 - CCellularAutomata, 11
- GuiDisplayCaMove
 - CWindowMain, 152
- GuiModeCADisplayModeChanged
 - CWidgetExport, 128
- GuiModeCATimeoutChanged
 - CWidgetExport, 129
- IdentifyCorners

- CFitness, [43](#)
- CGeneticAlgorithm, [48](#)
- ImportCA
 - CInputCA, [93](#)
- ImportChromosome
 - CImportGA, [90](#)
- ImportConfig
 - CImportConfig, [87](#)
- ImportFileButtonPressed
 - CWidgetEvolution, [122](#)
- ImportFileEnableReevolve
 - CWidgetEvolution, [122](#)
- ImportFileEnableSimulation
 - CWidgetEvolution, [122](#)
- ImportGenomeToPopulation
 - CGeneticAlgorithm, [49](#)
- ImportPressed
 - CWidgetInput, [134](#)
- InitArray
 - CArray2d, [7](#)
- InitCore
 - CThreadCore, [113](#)
 - CWindowMain, [153](#)
- InitCoreCAMemory
 - CThreadCore, [113](#)
- InitCoreCellularAutomata
 - CThreadCore, [113](#)
- InitCoreGeneticAlgorithm
 - CThreadCore, [113](#)
- InitExport
 - CThreadCore, [113](#)
- InitGenonetypeStandard
 - CGeneticAlgorithm, [49](#)
- InitGenotype
 - CGeneticAlgorithm, [49](#)
- InitGenotypeInstruction
 - CGeneticAlgorithm, [49](#)
- InitGenotypeReevolve
 - CGeneticAlgorithm, [49](#)
- InitGuiCore
 - CCore, [26](#)
- InitMemory
 - CCellularAutomata, [11](#)
- InitPressed
 - CWindowMain, [153](#)
- InitRun
 - CWidgetRunEvo, [147](#)
- InitRunGrid
 - CWidgetRunCA, [144](#)
- InitSpace
 - CCellularAutomata, [11](#)
- InitTmpCAs
 - CThreadCore, [113](#)
- IsApplied
 - CWidgetEvolution, [122](#)
 - CWidgetExport, [129](#)
 - CWidgetInput, [134](#)
 - CWidgetOutput, [140](#)
- IsEditable
 - CGraphicsItem, [76](#)
- IsImportGenomeEnabledReevolve
 - CConfigCore, [17](#)
 - CWidgetEvolution, [122](#)
- IsImportGenomeEnabledSimulation
 - CConfigCore, [17](#)
 - CWidgetEvolution, [122](#)
- IsInitDone
 - CCellularAutomata, [11](#)
 - CCore, [26](#)
 - CThreadCore, [113](#)
- IsLiveOrganismInSpace
 - CFitness, [43](#)
- LCG
 - CRandom, [96](#)
- MenuActionAbout
 - CWindowMain, [153](#)
- MenuActionExportRunConfig
 - CWindowMain, [153](#)
- MenuActionImportRunConfig
 - CWindowMain, [153](#)
- minimumSizeHint
 - CGraphicsView, [85](#)
- mouseMoveEvent
 - CGraphicsItem, [76](#)
- mousePressEvent
 - CGraphicsItem, [76](#)
- mouseReleaseEvent
 - CGraphicsItem, [76](#)
- Mutation
 - CGeneticAlgorithm, [49](#)
 - CMutation, [95](#)
- MutationCountChanged
 - CWidgetEvolution, [122](#)
- MutationProbChanged
 - CWidgetEvolution, [122](#)
- NextSpace
 - CTFunction, [107](#)
- NextSpaceGenomeInstruction
 - CTFunction, [107](#)
- NextSpaceGenomeStandard
 - CTFunction, [107](#)
- OpenLogFile
 - CExportLog, [39](#)
- operator()

- CArray2d, 8
- operator=
 - CArray2d, 8
- paint
 - CGraphicsItem, 76
- PopulationChanged
 - CWidgetEvolution, 123
- PosChangedX
 - CWidgetOutput, 140
- PosChangedY
 - CWidgetOutput, 140
- Random
 - CRandom, 97
- ReadDocEnd
 - CImportConfig, 88
- ReadDocStart
 - CImportConfig, 88
 - CImportGA, 91
- ReadGenome
 - CImportGA, 91
- ReadSettingsEvolution
 - CImportConfig, 88
- ReadSettingsExport
 - CImportConfig, 88
- ReadSettingsInput
 - CImportConfig, 88
- ReadSettingsInputCAspace
 - CImportConfig, 88
- ReadSettingsOutput
 - CImportConfig, 88
- RefillPopulation
 - CGeneticAlgorithm, 49
- ReInit
 - CCellularAutomata, 11
- ReinitCore
 - CThreadCore, 113
- ReinitInputView
 - CWidgetInput, 134
- RenumberActualGeneration
 - CGeneticAlgorithm, 49
- RepetitionsChanged
 - CWidgetEvolution, 123
- run
 - CThreadCore, 114
- RunGenomeCaSimulation
 - CThreadCore, 114
- RunGuiMode
 - CThreadCore, 114
- RunPressed
 - CWindowMain, 153
- Selection
 - CGeneticAlgorithm, 50
 - CSelection, 101
- SetAncestorsCount
 - CGenome, 56
 - CWidgetRunEvo, 147
- SetBitValue
 - CGenomeType2_Ins, 62
- SetCaRunTermination
 - CCore, 26
- SetCellActState
 - CGraphicsItemConfig, 79
- SetChromosomeID
 - CWidgetRunEvo, 147
- SetConfigCore
 - CCellularAutomata, 11
 - CCrossover, 28
 - CExportCA, 31
 - CExportGA, 36
 - CFitness, 44
 - CGeneticAlgorithm, 50
 - CImportGA, 91
 - CInputCA, 93
 - CMutation, 95
 - CRulesTable, 98
 - CSelection, 101
 - CTFunction, 107
 - CThreadCore, 114
- SetConfigDefState
 - CGraphicsScene, 83
- SetConfigEditable
 - CGraphicsScene, 83
- SetConfigStatesCount
 - CGraphicsScene, 83
- SetCoreDataExpiration
 - CCore, 26
 - CThreadCore, 114
- SetCoreDataGA
 - CThreadCore, 114
- SetCoreDataValidity
 - CThreadCore, 114
- SetCoreSpace
 - CThreadCore, 115
- SetCrossoverCount
 - CConfigCore, 18
 - CWidgetEvolution, 123
- SetCrossoverProbability
 - CConfigCore, 18
 - CWidgetEvolution, 123
- SetDefaultState
 - CConfigCore, 18
- SetDefState
 - CGraphicsItemConfig, 79
- SetDifferentiationMin
 - CWidgetRunEvo, 147

- SetEditable
 - CGraphicsItem, 76
 - CGraphicsItemConfig, 79
- SetEvolutionRepetitionsCount
 - CConfigCore, 18
- SetExportFileModeCa
 - CConfigCore, 18
- SetExportFileModeGa
 - CConfigCore, 18
- SetExportFilePath
 - CConfigCore, 18
- SetExportLogCore
 - CConfigCore, 19
- SetFileExportModeCA
 - CWidgetExport, 129
- SetFileExportModeGA
 - CWidgetExport, 129
- SetFileName
 - CExportLog, 39
- SetFilePath
 - CExportLog, 39
- SetFitMaxStepCA
 - CWidgetRunEvo, 147
- SetFitness
 - CGenome, 56
- SetFitnessMax
 - CWidgetRunEvo, 147
- SetFitnessNorm
 - CGenome, 56
- SetGene
 - CGenome, 56
 - CGenomeType2, 59
 - CGenomeType2_Nbh5, 65
 - CGenomeType3, 66
 - CGenomeType3_Nbh5, 69
 - CGenomeType4, 70
 - CGenomeType4_Nbh5, 73
- SetGenerationID
 - CWidgetRunEvo, 148
- SetGenerationsCount
 - CConfigCore, 19
 - CWidgetEvolution, 123
- SetGenome
 - CRulesTable, 99
- SetGenomeType
 - CConfigCore, 19
 - CWidgetEvolution, 123
- SetGuiDataDisplayModeCA
 - CWidgetExport, 129
- SetGuiDataDisplayModeTimeout
 - CWidgetExport, 129
- SetGuiDisplayModeCA
 - CConfigCore, 19
- SetGuiDisplayModeCATimeout
 - CConfigCore, 19
- SetHeight
 - CGraphicsScene, 83
- SetImportGenomeEnabledReevolve
 - CConfigCore, 19
- SetImportGenomeEnabledSimulation
 - CConfigCore, 20
- SetImportGenomeFile
 - CConfigCore, 20
- SetInitDone
 - CWidgetEvolution, 124
 - CWidgetExport, 129
 - CWidgetInput, 134
 - CWidgetOutput, 140
- SetInputApplied
 - CWidgetOutput, 140
- SetInputGridSize
 - CWidgetInput, 134
- SetInputGridSizeX
 - CWidgetInput, 134
- SetInputGridSizeY
 - CWidgetInput, 134
- SetInputSizeX
 - CWidgetOutput, 141
- SetInputSizeY
 - CWidgetOutput, 141
- SetInstruction
 - CGenomeType2_Ins, 62
- SetIODevice
 - CExportConfig, 33
 - CExportGA, 36
 - CImportConfig, 89
 - CImportGA, 91
- SetMoveDirection
 - CConfigCore, 20
 - CWidgetEvolution, 124
- SetMoveDistance
 - CConfigCore, 20
 - CWidgetEvolution, 124
- SetMutatedGenes
 - CGenome, 56
- SetMutatedGenesTotal
 - CGenome, 57
- SetMutationCount
 - CConfigCore, 20
 - CWidgetEvolution, 124
- SetMutationProbability
 - CConfigCore, 20
 - CWidgetEvolution, 124
- SetMutex
 - CThreadCore, 115
- SetOutputApplied
 - CWidgetInput, 135
- SetOutputArrayType

- CWidgetOutput, [141](#)
- SetOutputGridSizeX
 - CWidgetOutput, [141](#)
- SetOutputGridSizeY
 - CWidgetOutput, [141](#)
- SetOutputMapPosX
 - CWidgetInput, [135](#)
 - CWidgetOutput, [141](#)
- SetOutputMapPosY
 - CWidgetInput, [135](#)
 - CWidgetOutput, [142](#)
- SetOutputSizeX
 - CWidgetInput, [135](#)
- SetOutputSizeY
 - CWidgetInput, [135](#)
- SetParentGenomeId
 - CGenome, [57](#)
- SetPopulationSize
 - CConfigCore, [21](#)
 - CWidgetEvolution, [124](#)
- SetPostCondition
 - CGenomeType2_Ins, [62](#)
- SetPreCondition
 - CGenomeType2_Ins, [63](#)
- SetPreConditionBit
 - CGenomeType2_Ins, [63](#)
- SetPreConditionLogic
 - CGenomeType2_Ins, [63](#)
- SetPreConditionLogicBit
 - CGenomeType2_Ins, [63](#)
- SetRepetitionsCount
 - CWidgetEvolution, [124](#)
- SetRulesTable
 - CTFunction, [107](#)
- SetRunGridSizeX
 - CWidgetRunCA, [144](#)
- SetRunGridSizeY
 - CWidgetRunCA, [145](#)
- SetRunGridStates
 - CWidgetRunCA, [145](#)
- SetSameChromosomesCount
 - CWidgetRunEvo, [148](#)
- SetSameChromosomesGenerationAvarage
 - CWidgetRunEvo, [148](#)
- SetSimulationRunning
 - CCore, [26](#)
 - CThreadCore, [115](#)
- SetSpaceSizeX
 - CConfigCore, [21](#)
- SetSpaceSizeY
 - CConfigCore, [21](#)
- SetSpaceType
 - CConfigCore, [21](#)
- SetState
 - CGraphicsItem, [77](#)
- SetStatesCount
 - CConfigCore, [21](#)
 - CGraphicsItem, [77](#)
 - CGraphicsItemConfig, [79](#)
 - CWidgetInput, [135](#)
- SetStepsCount
 - CWidgetEvolution, [125](#)
- SetStepsCountCA
 - CConfigCore, [21](#)
- SetThisGenomeId
 - CGenome, [57](#)
- SetThisInitGenomeId
 - CGenome, [57](#)
- SetupInputView
 - CWidgetInput, [136](#)
- setupMatrix
 - CGraphicsView, [85](#)
- SetupRunGridView
 - CWidgetRunCA, [145](#)
- SetWaitCondition
 - CThreadCore, [115](#)
- SetWidth
 - CGraphicsScene, [84](#)
- ShowDebugConfigCore
 - CCore, [26](#)
- ShowMessageBox
 - CExportLog, [39](#)
- SimulationDelete
 - CWindowMain, [153](#)
- SizeChangedX
 - CWidgetInput, [136](#)
 - CWidgetOutput, [142](#)
- SizeChangedY
 - CWidgetInput, [136](#)
 - CWidgetOutput, [142](#)
- SizeChangeUsed
 - CWidgetInput, [136](#)
- stArray, [155](#)
- StartThreadCore
 - CCore, [26](#)
- StatesChandeUsed
 - CWidgetInput, [136](#)
- StatesChanged
 - CWidgetInput, [136](#)
- Step
 - CCellularAutomata, [11](#)
- StepGoL
 - CCellularAutomata, [12](#)
- StepPressed
 - CWindowMain, [153](#)
- StepsChanged
 - CWidgetEvolution, [125](#)
- stGeneInstruction, [156](#)

- StopPressed
 - CWindowMain, [153](#)
- StoreGenomeDataForGui
 - CThreadCore, [115](#)
- stThreadCoreDataGA, [157](#)
- SyncDataWithCore
 - CThreadCore, [115](#)
- TerminateCaRun
 - CThreadCore, [116](#)
- TerminatePressed
 - CWindowMain, [153](#)
- TerminateThreadLoop
 - CThreadCore, [116](#)
- ThreadCoreCheckCoreInitDone
 - CCore, [27](#)
- ThreadCoreErrorSlot
 - CCore, [27](#)
- ThreadCoreNewDataAvailable
 - CCore, [27](#)
- ThreadCoreState
 - CCore, [27](#)
- Uniform
 - CRandom, [97](#)
- UniformStdLib
 - CRandom, [97](#)
- view
 - CGraphicsView, [85](#)
- WidgetEvolutionApplied
 - CWindowMain, [153](#)
- WidgetExportApplied
 - CWindowMain, [154](#)
- WidgetInputApplied
 - CWindowMain, [154](#)
- WidgetOutputApplied
 - CWindowMain, [154](#)
- WriteCASpaceFromCoreIntoCA
 - CThreadCore, [116](#)
- WriteDataCAToCore
 - CThreadCore, [116](#)
- WriteDataGAToCore
 - CThreadCore, [116](#)
- WriteDocEnd
 - CExportConfig, [33](#)
 - CExportGA, [36](#)
- WriteDocStart
 - CExportConfig, [33](#)
 - CExportGA, [36](#)
- WriteErrorLog
 - CExportLog, [40](#)
- WriteGenesIntoGenome
 - CImportGA, [91](#)
- WriteGenome
 - CExportGA, [36](#)
- WriteInfoLog
 - CExportLog, [40](#)
- WriteSettingsEvolution
 - CExportConfig, [33](#)
- WriteSettingsExport
 - CExportConfig, [33](#)
- WriteSettingsInput
 - CExportConfig, [33](#)
- WriteSettingsInputCASpace
 - CExportConfig, [34](#)
- WriteSettingsOutput
 - CExportConfig, [34](#)
- zoomIn
 - CGraphicsView, [86](#)
- zoomOut
 - CGraphicsView, [86](#)