

Simulation Laby

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	AutomateGraph Class Reference . . . . .	7
4.1.1	Constructor & Destructor Documentation . . . . .	7
4.1.1.1	AutomateGraph() . . . . .	7
4.1.2	Member Function Documentation . . . . .	8
4.1.2.1	accessibilityAutomate() . . . . .	8
4.1.2.2	adaptTourLangage() . . . . .	8
4.1.2.3	addWord2Langage() . . . . .	8
4.1.2.4	export2DESUMA() . . . . .	8
4.1.2.5	FSM2Automata() . . . . .	8
4.1.2.6	matrices2vector() . . . . .	8
4.1.2.7	PathResearche() . . . . .	9
4.1.2.8	structAutomata2vectorAutomata() . . . . .	9
4.1.2.9	vector2matrices() . . . . .	9
4.1.2.10	vector2structAutomata() . . . . .	9
4.1.3	Member Data Documentation . . . . .	9

4.1.3.1	langage	9
4.1.3.2	matrixTrans	9
4.1.3.3	state	10
4.1.3.4	transition	10
4.1.3.5	vector	10
4.2	Automaton Class Reference	10
4.2.1	Constructor & Destructor Documentation	10
4.2.1.1	Automaton()	11
4.2.2	Member Function Documentation	11
4.2.2.1	AjoutTransitionStable()	11
4.2.2.2	AutomateAccessible()	11
4.2.2.3	AutomateAccessibleIncertain()	11
4.2.2.4	CreationFonctionTransitionEnsembleDesParties()	11
4.2.2.5	Ensemble2Etats()	11
4.2.2.6	EnsemblesContenantEtat()	12
4.2.2.7	Etats2Ensemble()	12
4.2.2.8	Evolution()	12
4.2.2.9	PathRecherche()	12
4.2.3	Member Data Documentation	12
4.2.3.1	Fct	12
4.2.3.2	MatricesTransition	12
4.2.3.3	NomsEvenements	13
4.3	handle Class Reference	13
4.4	Labyrinthe Class Reference	13
4.4.1	Constructor & Destructor Documentation	14
4.4.1.1	Labyrinthe()	14
4.4.2	Member Function Documentation	14
4.4.2.1	affichage()	14
4.4.2.2	incertain()	14
4.4.2.3	jusqu_au_mur()	14

4.4.2.4	Pas_a_pas()	14
4.4.3	Member Data Documentation	14
4.4.3.1	Etats_Finaux	15
4.4.3.2	Etats_Initiaux	15
4.4.3.3	MursHorizontaux	15
4.4.3.4	MursVerticaux	15
4.5	ModelLaby Class Reference	15
4.5.1	Detailed Description	16
4.5.2	Constructor & Destructor Documentation	16
4.5.2.1	ModelLaby()	16
4.5.3	Member Function Documentation	17
4.5.3.1	f()	17
4.5.3.2	g() [1/2]	17
4.5.3.3	g() [2/2]	17
4.5.3.4	m()	17
4.5.4	Member Data Documentation	17
4.5.4.1	initialState	18
4.5.4.2	presentState	18
4.6	ModelPacman Class Reference	18
4.6.1	Constructor & Destructor Documentation	19
4.6.1.1	ModelPacman()	19
4.6.2	Member Function Documentation	19
4.6.2.1	f()	19
4.6.2.2	g() [1/2]	19
4.6.2.3	g() [2/2]	19
4.6.2.4	m()	20
4.6.3	Member Data Documentation	20
4.6.3.1	initialState	20
4.6.3.2	presentState	20
4.7	ModelSED Class Reference	20

4.7.1	Member Function Documentation	21
4.7.1.1	f()	21
4.7.1.2	g()	21
4.7.1.3	m()	21
4.7.2	Member Data Documentation	21
4.7.2.1	initialState	22
4.7.2.2	presentState	22
4.8	ModelWalls Class Reference	22
4.8.1	Constructor & Destructor Documentation	23
4.8.1.1	ModelWalls()	23
4.8.2	Member Function Documentation	23
4.8.2.1	f() [1/2]	23
4.8.2.2	f() [2/2]	23
4.8.2.3	g() [1/2]	23
4.8.2.4	g() [2/2]	23
4.8.2.5	m()	24
4.8.3	Member Data Documentation	24
4.8.3.1	i	24
4.8.3.2	initialState	24
4.8.3.3	presentState	24
4.8.3.4	val	24
4.9	StopCondition Class Reference	25
4.9.1	Constructor & Destructor Documentation	25
4.9.1.1	StopCondition()	25
4.9.2	Member Function Documentation	25
4.9.2.1	f() [1/2]	26
4.9.2.2	f() [2/2]	26
4.9.2.3	g() [1/2]	26
4.9.2.4	g() [2/2]	26
4.9.2.5	m()	26

4.9.3	Member Data Documentation . . . . .	26
4.9.3.1	initialState . . . . .	27
4.9.3.2	presentState . . . . .	27
4.10	Wrapper Class Reference . . . . .	27
4.10.1	Constructor & Destructor Documentation . . . . .	27
4.10.1.1	Wrapper() . . . . .	28
4.10.2	Member Function Documentation . . . . .	28
4.10.2.1	get_out() . . . . .	28
4.10.2.2	get_stop() . . . . .	28
4.10.2.3	init() . . . . .	28
4.10.2.4	orderer() . . . . .	28
4.10.2.5	updateConnexion() . . . . .	28
4.10.3	Member Data Documentation . . . . .	29
4.10.3.1	commandPacman . . . . .	29
4.10.3.2	commandWalls . . . . .	29
4.10.3.3	in . . . . .	29
4.10.3.4	modelLaby . . . . .	29
4.10.3.5	out . . . . .	29
4.10.3.6	pacmanBit . . . . .	29
4.10.3.7	stop . . . . .	29
4.10.3.8	stopCondition . . . . .	30
4.10.3.9	wallsBit . . . . .	30
4.10.3.10	whoPlay . . . . .	30

<b>5</b>	<b>File Documentation</b>	<b>31</b>
5.1	automaton/AutomateGraph.m File Reference . . . . .	31
5.2	automaton/mainLaby.m File Reference . . . . .	31
5.3	automaton/modelGenerator/AutomatonSchedulingCreation.m File Reference . . . . .	31
5.3.1	Function Documentation . . . . .	31
5.3.1.1	function() . . . . .	31
5.4	automaton/modelGenerator/AutomatonStrutureLabyCreation.m File Reference . . . . .	31
5.4.1	Function Documentation . . . . .	32
5.4.1.1	AutomatonStrutureLabyCreation() . . . . .	32
5.5	automaton/modelGenerator/AutomatonWallsConstraintsCreation.m File Reference . . . . .	32
5.5.1	Function Documentation . . . . .	32
5.5.1.1	AutomatonWallsConstraintsCreation() . . . . .	32
5.6	automaton/modelGenerator/generer_lab.m File Reference . . . . .	32
5.6.1	Function Documentation . . . . .	32
5.6.1.1	generer_lab() . . . . .	32
5.7	automaton/modelGenerator/modelGenerator.m File Reference . . . . .	33
5.8	automaton/modelGenerator/Plan_desumaFunctions.m File Reference . . . . .	33
5.8.1	Function Documentation . . . . .	33
5.8.1.1	AutomatonStrutureLabyCreation() . . . . .	33
5.8.1.2	function() . . . . .	33
5.8.1.3	SaveDESUMAFile() . . . . .	33
5.8.1.4	writeStates() . . . . .	33
5.8.1.5	writeTransitions() . . . . .	34
5.9	automaton/modelGenerator/SaveDESUMAFile.m File Reference . . . . .	34
5.9.1	Function Documentation . . . . .	34
5.9.1.1	SaveDESUMAFile() . . . . .	34
5.10	automaton/modelGenerator/writeStates.m File Reference . . . . .	34
5.10.1	Function Documentation . . . . .	34
5.10.1.1	writeStates() . . . . .	34
5.11	automaton/modelGenerator/writeTransitions.m File Reference . . . . .	35



5.11.1	Function Documentation	35
5.11.1.1	writeTransitions()	35
5.12	automaton/optimalCommand/creationMatricetransition.m File Reference	35
5.12.1	Function Documentation	35
5.12.1.1	creationMatricetransition()	35
5.13	automaton/optimalCommand/getStateTransitionFSM.m File Reference	35
5.13.1	Function Documentation	35
5.13.1.1	getStateTransitionFSM()	36
5.14	automaton/optimalCommand/getStateTransitionTXT.m File Reference	36
5.14.1	Function Documentation	36
5.14.1.1	getStateTransitionTXT()	36
5.15	automaton/optimalCommand/main.m File Reference	36
5.16	main.m File Reference	36
5.17	automaton/optimalCommand/optimalCommand.m File Reference	36
5.17.1	Function Documentation	36
5.17.1.1	optimalCommand()	36
5.18	automaton/optimalCommand/ParcourirMatricesTransitions.m File Reference	37
5.18.1	Function Documentation	37
5.18.1.1	ParcourirMatricesTransitions()	37
5.19	automaton/optimalCommand/rafineAutomaton.m File Reference	37
5.20	automaton/optimalCommand/rafineAutomatonClass.m File Reference	37
5.20.1	Function Documentation	37
5.20.1.1	rafineAutomatonClass()	37
5.21	automaton/ParrallelComposition.m File Reference	37
5.21.1	Function Documentation	38
5.21.1.1	ParrallelComposition()	38
5.22	automaton_nd/Automaton.m File Reference	38
5.23	automaton_nd/Labyrinthe.m File Reference	38
5.24	automaton_nd/modi_main.m File Reference	38
5.25	CreatePituresAndVideo.m File Reference	38

5.25.1	Function Documentation	38
5.25.1.1	CreatePituresAndVideo()	38
5.26	CreatePituresAndVideo_textured.m File Reference	39
5.26.1	Function Documentation	39
5.26.1.1	CreatePituresAndVideo_textured()	39
5.27	figure_Laby.m File Reference	39
5.27.1	Function Documentation	39
5.27.1.1	connect_Callback()	40
5.27.1.2	createUIEscape()	40
5.27.1.3	createUIPacman()	40
5.27.1.4	createUIWalls()	40
5.27.1.5	figure_Laby()	40
5.27.1.6	figure_Laby_OpeningFcn()	40
5.27.1.7	figure_Laby_OutputFcn()	41
5.27.1.8	isOne()	41
5.27.1.9	resetUIConnection()	41
5.27.1.10	ui_Callback()	41
5.27.1.11	updatePresenceDetectorDisplay()	41
5.27.1.12	updateUI()	41
5.27.1.13	updateUIActiveCammand()	42
5.27.1.14	updateUIButton()	42
5.27.1.15	updateUIEscape()	42
5.27.1.16	updateUIPlayer()	42
5.27.1.17	updateUIWalls()	42
5.27.1.18	updateUIWallsAround()	42
5.28	ModelLaby.m File Reference	43
5.29	ModelPacman.m File Reference	43
5.30	ModelSED.m File Reference	43
5.31	ModelWalls.m File Reference	43
5.32	setColor.m File Reference	43
5.32.1	Function Documentation	43
5.32.1.1	setColor()	43
5.33	Simulation.m File Reference	44
5.34	StopCondition.m File Reference	44
5.35	testUI.m File Reference	44
5.36	THEplan.m File Reference	44
5.37	visupacman.m File Reference	44
5.38	visupacman2.m File Reference	44
5.39	wallsBorder.m File Reference	44
5.39.1	Function Documentation	44
5.39.1.1	wallsBorder()	44
5.40	Wrapper.m File Reference	44





# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

AutomateGraph . . . . .	7
Automaton . . . . .	10
handle . . . . .	13
ModelSED . . . . .	20
ModelLaby . . . . .	15
ModelPacman . . . . .	18
ModelWalls . . . . .	22
StopCondition . . . . .	25
Labyrinthe . . . . .	13
Wrapper . . . . .	27



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">AutomateGraph</a>	7
<a href="#">Automaton</a>	10
<a href="#">handle</a>	13
<a href="#">Labyrinthe</a>	13
<a href="#">ModellLaby</a>	
Class which contains the "fmg" structure of the labyrinth for 1 player	15
<a href="#">ModelPacman</a>	18
<a href="#">ModelSED</a>	20
<a href="#">ModelWalls</a>	22
<a href="#">StopCondition</a>	25
<a href="#">Wrapper</a>	27





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

CreatePituresAndVideo.m	38
CreatePituresAndVideo_textured.m	39
figure_Laby.m	39
main.m	36
ModelLaby.m	43
ModelPacman.m	43
ModelSED.m	43
ModelWalls.m	43
setColor.m	43
Simulation.m	44
StopCondition.m	44
testUI.m	44
THEplan.m	44
visupacman.m	44
visupacman2.m	44
wallsBorder.m	44
Wrapper.m	44
automaton/AutomateGraph.m	31
automaton/mainLaby.m	31
automaton/ParrallelComposition.m	37
automaton/modelGenerator/AutomatonSchedulingCreation.m	31
automaton/modelGenerator/AutomatonStrutureLabyCreation.m	31
automaton/modelGenerator/AutomatonWallsContraintsCreation.m	32
automaton/modelGenerator/generer_lab.m	32
automaton/modelGenerator/modelGenerator.m	33
automaton/modelGenerator/Plan_desumaFunctions.m	33
automaton/modelGenerator/SaveDESUMAFFile.m	34
automaton/modelGenerator/writeStates.m	34
automaton/modelGenerator/writeTransitions.m	35
automaton/optimalCommand/creationMatricetransition.m	35
automaton/optimalCommand/getStateTransitionFSM.m	35
automaton/optimalCommand/getStateTransitionTXT.m	36
automaton/optimalCommand/main.m	36
automaton/optimalCommand/optimalCommand.m	36
automaton/optimalCommand/ParcourirMatricesTransitions.m	37

automaton/optimalCommand/ <a href="#">rafineAutomaton.m</a> . . . . .	37
automaton/optimalCommand/ <a href="#">rafineAutomatonClass.m</a> . . . . .	37
automaton_nd/ <a href="#">Automaton.m</a> . . . . .	38
automaton_nd/ <a href="#">Labyrinthe.m</a> . . . . .	38
automaton_nd/ <a href="#">modi_main.m</a> . . . . .	38

# Chapter 4

## Class Documentation

### 4.1 AutomateGraph Class Reference

#### Public Member Functions

- [function AutomateGraph \(\)](#)
- [function FSM2Automata](#) (in obj, in nameFileFSM)
- [function vector2matrices](#) (in obj)
- [function addWord2Langage](#) (in obj, in word)
- [function adaptTourLangage](#) (in obj)
- [function structAutomata2vectorAutomata](#) (in obj)
- [function matrices2vector](#) (in obj)
- [function PathResearche](#) (in obj, in initialState, in studiedState)
- [function vector2structAutomata](#) (in obj)
- [function export2DESUMA](#) (in obj, in file)
- [function accessibilityAutomate](#) (in obj)

#### Public Attributes

- Property [state](#)
- Property [transition](#)
- Property [matrixTrans](#)
- Property [langage](#)
- Property [vector](#)

#### 4.1.1 Constructor & Destructor Documentation

##### 4.1.1.1 AutomateGraph()

```
function AutomateGraph ( )
```

## 4.1.2 Member Function Documentation

### 4.1.2.1 accessibilityAutomate()

```
function accessibilityAutomate (  
    in obj )
```

### 4.1.2.2 adaptTourLangage()

```
function adaptTourLangage (  
    in obj )
```

### 4.1.2.3 addWord2Langage()

```
function addWord2Langage (  
    in obj,  
    in word )
```

### 4.1.2.4 export2DESUMA()

```
function export2DESUMA (  
    in obj,  
    in file )
```

### 4.1.2.5 FSM2Automata()

```
function FSM2Automata (  
    in obj,  
    in nameFileFSM )
```

### 4.1.2.6 matrices2vector()

```
function matrices2vector (  
    in obj )
```

#### 4.1.2.7 PathResearche()

```
function PathResearche (
    in obj,
    in initialState,
    in studiedState )
```

#### 4.1.2.8 structAutomata2vectorAutomata()

```
function structAutomata2vectorAutomata (
    in obj )
```

#### 4.1.2.9 vector2matrices()

```
function vector2matrices (
    in obj )
```

#### 4.1.2.10 vector2structAutomata()

```
function vector2structAutomata (
    in obj )
```

### 4.1.3 Member Data Documentation

#### 4.1.3.1 langage

Property langage

#### 4.1.3.2 matrixTrans

Property matrixTrans

#### 4.1.3.3 state

Property state

#### 4.1.3.4 transition

Property transition

#### 4.1.3.5 vector

Property vector

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

- automaton/[AutomateGraph.m](#)

## 4.2 Automaton Class Reference

### Public Member Functions

- [function Automaton](#) (in varargin)
- [function CreationFonctionTransitionEnsembleDesParties](#) (in obj)
- [function Evolution](#) (in obj, in Conditions, in Initial)
- [function AjoutTransitionStable](#) (in obj)
- [function PathRecherche](#) (in obj, in initialState, in studiedState)
- [function AutomateAccessible](#) (in obj, in initialState, in studiedState)
- [function AutomateAccessibleIncertain](#) (in obj, in Etat\_initial, in Etat)
- [function Ensemble2Etats](#) (in obj, in Ensemble)
- [function EnsemblesContenantEtat](#) (in obj, in Etats)
- [function Etats2Ensemble](#) (in obj, in Etats)

### Public Attributes

- Property [MatricesTransition](#)
- Property [Fct](#)
- Property [NomsEvenements](#)

#### 4.2.1 Constructor & Destructor Documentation

#### 4.2.1.1 Automaton()

```
function Automaton (  
    in varargin )
```

### 4.2.2 Member Function Documentation

#### 4.2.2.1 AjoutTransitionStable()

```
function AjoutTransitionStable (  
    in obj )
```

#### 4.2.2.2 AutomateAccessible()

```
function AutomateAccessible (  
    in obj,  
    in initialState,  
    in studiedState )
```

#### 4.2.2.3 AutomateAccessibleIncertain()

```
function AutomateAccessibleIncertain (  
    in obj,  
    in Etat_initial,  
    in Etat )
```

#### 4.2.2.4 CreationFonctionTransitionEnsembleDesParties()

```
function CreationFonctionTransitionEnsembleDesParties (  
    in obj )
```

#### 4.2.2.5 Ensemble2Etats()

```
function Ensemble2Etats (  
    in obj,  
    in Ensemble )
```

#### 4.2.2.6 EnsemblesContenantEtat()

```
function EnsemblesContenantEtat (
    in obj,
    in Etats )
```

#### 4.2.2.7 Etats2Ensemble()

```
function Etats2Ensemble (
    in obj,
    in Etats )
```

#### 4.2.2.8 Evolution()

```
function Evolution (
    in obj,
    in Conditions,
    in Initial )
```

#### 4.2.2.9 PathRecherche()

```
function PathRecherche (
    in obj,
    in initialState,
    in studiedState )
```

### 4.2.3 Member Data Documentation

#### 4.2.3.1 Fct

Property Fct

#### 4.2.3.2 MatricesTransition

Property MatricesTransition



## 4.2.3.3 NomsEvenements

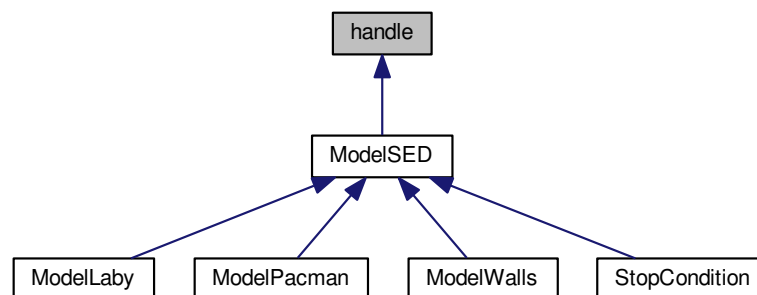
Property NomsEvenements

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

- [automaton\\_nd/Automaton.m](#)

## 4.3 handle Class Reference

Inheritance diagram for handle:



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

- [ModelSED.m](#)

## 4.4 Labyrinthe Class Reference

### Public Member Functions

- [function Labyrinthe](#) (in murs\_Verticaux, in murs\_Horizontaux, in etats\_Initiaux, in etats\_Finaux)
- [function Pas\\_a\\_pas](#) (in obj)
- [function jusqu\\_au\\_mur](#) (in obj)
- [function incertain](#) (in obj)
- [function affichage](#) (in obj)

### Public Attributes

- Property [MursVerticaux](#)
- Property [MursHorizontaux](#)
- Property [Etats\\_Initiaux](#)
- Property [Etats\\_Finaux](#)

## 4.4.1 Constructor & Destructor Documentation

### 4.4.1.1 Labyrinthe()

```
function Labyrinthe (
    in murs_Verticaux,
    in murs_Horizontaux,
    in etats_Initiaux,
    in etats_Finaux )
```

## 4.4.2 Member Function Documentation

### 4.4.2.1 affichage()

```
function affichage (
    in obj )
```

### 4.4.2.2 incertain()

```
function incertain (
    in obj )
```

### 4.4.2.3 jusqu\_au\_mur()

```
function jusqu_au_mur (
    in obj )
```

### 4.4.2.4 Pas\_a\_pas()

```
function Pas_a_pas (
    in obj )
```

## 4.4.3 Member Data Documentation

#### 4.4.3.1 Etats\_Finaux

Property Etats\_Finaux

#### 4.4.3.2 Etats\_Initiaux

Property Etats\_Initiaux

#### 4.4.3.3 MursHorizontaux

Property MursHorizontaux

#### 4.4.3.4 MursVerticaux

Property MursVerticaux

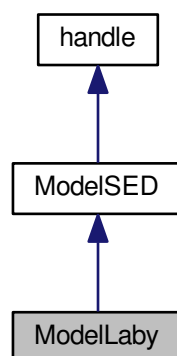
The documentation for this class was generated from the following file:

- automaton\_nd/[Labyrinthe.m](#)

## 4.5 Modellaby Class Reference

Class which contains the "fmg" structure of the labyrinth for 1 player

Inheritance diagram for Modellaby:



## Public Member Functions

- [function ModellLaby](#) (in wallsV\_init, in wallsH\_init, in pacman\_init, in escape\_init)  
*Class constructor of.*
- [function f](#) (in obj, in in)
- [function m](#) (in obj, in nextState, in init)
- [function g](#) (in obj)
- virtual [g](#) (in obj, in in)

## Public Attributes

- Property [presentState](#)  
*Data Structure of the current state of Labyrinth. It contains "wallsV", "wallsH" (2 matrix for the walls), "escape" and "pacman", a Cartesian position of current position of escape and pacman and 'wallsAroundPacman' A vector indicating the presence of a wall around the Pacman for the 4 directions Up Down Left Right.*
- Property [initialState](#)  
*Data Structure of the initial state of Labyrinth. It contains "wallsV", "wallsH" (2 matrix for the walls), "escape" and "pacman", a Cartesian position of current position of escape and pacman and 'wallsAroundPacman' A vector indicating the presence of a wall around the Pacman for the 4 directions Up Down Left Right.*

### 4.5.1 Detailed Description

Class which contains the "fmg" structure of the labyrinth for 1 player

Input : necessary information for compute the next state of the model

Output : output's action of the model State : minimal information necessary who evolve

### 4.5.2 Constructor & Destructor Documentation

#### 4.5.2.1 ModellLaby()

```
function ModellLaby (
    in wallsV_init,
    in wallsH_init,
    in pacman_init,
    in escape_init )
```

Class constructor of.

#### Parameters

<i>wallsV_init</i>	Contain a matrix (N, N-1) of Initial Vertical Walls.
<i>wallsH_init</i>	Contain a matrix (N-1, N) of Initial Horizontal Walls.
<i>pacman_init</i>	Contain a vector (x, y) of Initial Position of Pacman.
<i>escape_init</i>	Contain a vector (x, y) of Escape's Position.

#### Returns

instance of the [ModelLaby](#) class.

### 4.5.3 Member Function Documentation

#### 4.5.3.1 `f()`

```
function f (  
    in obj,  
    in in ) [virtual]
```

Reimplemented from [ModelSED](#).

#### 4.5.3.2 `g()` [1/2]

```
virtual g (  
    in obj,  
    in in ) [virtual], [inherited]
```

#### 4.5.3.3 `g()` [2/2]

```
function g (  
    in obj )
```

#### 4.5.3.4 `m()`

```
function m (  
    in obj,  
    in nextState,  
    in init ) [virtual]
```

Reimplemented from [ModelSED](#).

### 4.5.4 Member Data Documentation

#### 4.5.4.1 initialState

Property initialState

Data Structure of the initial state of Labyrinth. It contains "wallsV", "wallsH" (2 matrix for the walls), "escape" and "pacman", a Cartesian position of current position of escape and pacman and 'wallsAroundPacman' A vector indicating the presence of a wall around the Pacman for the 4 directions Up Down Left Right.

#### 4.5.4.2 presentState

Property presentState

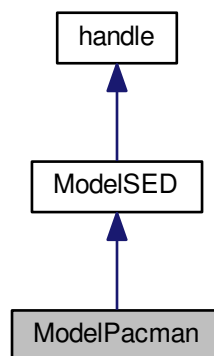
Data Structure of the current state of Labyrinth. It contains "wallsV", "wallsH" (2 matrix for the walls), "escape" and "pacman", a Cartesian position of current position of escape and pacman and 'wallsAroundPacman' A vector indicating the presence of a wall around the Pacman for the 4 directions Up Down Left Right.

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

- [ModelLaby.m](#)

## 4.6 ModelPacman Class Reference

Inheritance diagram for ModelPacman:



### Public Member Functions

- [function ModelPacman](#) (in initialValue)
- [function f](#) (in obj, in in)
- [function m](#) (in obj, in nextState, in init)
- [function g](#) (in obj)
- [virtual g](#) (in obj, in in)

## Public Attributes

- Property [presentState](#)
- Property [initialState](#)

## 4.6.1 Constructor & Destructor Documentation

### 4.6.1.1 ModelPacman()

```
function ModelPacman (  
    in initialValue )
```

## 4.6.2 Member Function Documentation

### 4.6.2.1 f()

```
function f (  
    in obj,  
    in in ) [virtual]
```

Reimplemented from [ModelSED](#).

### 4.6.2.2 g() [1/2]

```
virtual g (  
    in obj,  
    in in ) [virtual], [inherited]
```

### 4.6.2.3 g() [2/2]

```
function g (  
    in obj )
```

#### 4.6.2.4 m()

```
function m (
    in obj,
    in nextState,
    in init ) [virtual]
```

Reimplemented from [ModelSED](#).

### 4.6.3 Member Data Documentation

#### 4.6.3.1 initialState

Property initialState

#### 4.6.3.2 presentState

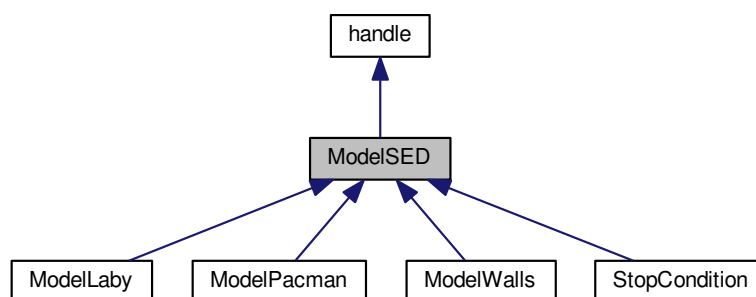
Property presentState

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

- [ModelPacman.m](#)

## 4.7 ModelSED Class Reference

Inheritance diagram for ModelSED:





## Public Member Functions

- virtual [f](#) (in obj, in in)
- virtual [m](#) (in obj, in nextState, in init)
- virtual [g](#) (in obj, in in)

## Public Attributes

- Property [presentState](#)
- Property [initialState](#)

### 4.7.1 Member Function Documentation

#### 4.7.1.1 f()

```
virtual f (  
    in obj,  
    in in ) [virtual]
```

Reimplemented in [ModelLaby](#), and [ModelPacman](#).

#### 4.7.1.2 g()

```
virtual g (  
    in obj,  
    in in ) [virtual]
```

#### 4.7.1.3 m()

```
virtual m (  
    in obj,  
    in nextState,  
    in init ) [virtual]
```

Reimplemented in [ModelPacman](#), [ModelLaby](#), [ModelWalls](#), and [StopCondition](#).

### 4.7.2 Member Data Documentation

#### 4.7.2.1 initialState

Property `initialState`

#### 4.7.2.2 presentState

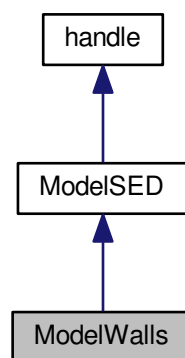
Property `presentState`

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

- [ModelSED.m](#)

## 4.8 ModelWalls Class Reference

Inheritance diagram for ModelWalls:



### Public Member Functions

- [function ModelWalls](#) (in `initValue`)
- [function f](#) (in `obj`)
- [function m](#) (in `obj`, in `nextState`, in `init`)
- [function g](#) (in `obj`)
- virtual [f](#) (in `obj`, in `in`)
- virtual [g](#) (in `obj`, in `in`)

### Public Attributes

- Property [presentState](#)
- Property [initialState](#)
- Property [i](#)
- Property [val](#)

## 4.8.1 Constructor & Destructor Documentation

### 4.8.1.1 ModelWalls()

```
function ModelWalls (  
    in initValue )
```

## 4.8.2 Member Function Documentation

### 4.8.2.1 f() [1/2]

```
virtual f (  
    in obj,  
    in in ) [virtual], [inherited]
```

Reimplemented in [ModelLaby](#), and [ModelPacman](#).

### 4.8.2.2 f() [2/2]

```
function f (  
    in obj )
```

### 4.8.2.3 g() [1/2]

```
virtual g (  
    in obj,  
    in in ) [virtual], [inherited]
```

### 4.8.2.4 g() [2/2]

```
function g (  
    in obj )
```

#### 4.8.2.5 m()

```
function m (
    in obj,
    in nextState,
    in init ) [virtual]
```

Reimplemented from [ModelSED](#).

### 4.8.3 Member Data Documentation

#### 4.8.3.1 i

Property i

#### 4.8.3.2 initialState

Property initialState

#### 4.8.3.3 presentState

Property presentState

#### 4.8.3.4 val

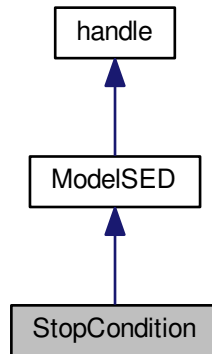
Property val

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

- [ModelWalls.m](#)

## 4.9 StopCondition Class Reference

Inheritance diagram for StopCondition:



### Public Member Functions

- [function StopCondition](#) (in `initCondition`)
- [function f](#) (in `obj`, in `noEscape`, in `pacmanWallsBreak`)
- [function m](#) (in `obj`, in `nextState`, in `init`)
- [function g](#) (in `obj`)
- virtual [f](#) (in `obj`, in `in`)
- virtual [g](#) (in `obj`, in `in`)

### Public Attributes

- Property [presentState](#)
- Property [initialState](#)

### 4.9.1 Constructor & Destructor Documentation

#### 4.9.1.1 StopCondition()

```
function StopCondition (  
    in initCondition )
```

### 4.9.2 Member Function Documentation

#### 4.9.2.1 `f()` [1/2]

```
virtual f (  
    in obj,  
    in in ) [virtual], [inherited]
```

Reimplemented in [ModelLaby](#), and [ModelPacman](#).

#### 4.9.2.2 `f()` [2/2]

```
function f (  
    in obj,  
    in noEscape,  
    in pacmanWallsBreak )
```

#### 4.9.2.3 `g()` [1/2]

```
virtual g (  
    in obj,  
    in in ) [virtual], [inherited]
```

#### 4.9.2.4 `g()` [2/2]

```
function g (  
    in obj )
```

#### 4.9.2.5 `m()`

```
function m (  
    in obj,  
    in nextState,  
    in init ) [virtual]
```

Reimplemented from [ModelSED](#).

### 4.9.3 Member Data Documentation

#### 4.9.3.1 initialState

Property `initialState`

#### 4.9.3.2 presentState

Property `presentState`

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

- [StopCondition.m](#)

## 4.10 Wrapper Class Reference

### Public Member Functions

- [function Wrapper](#) ([in](#) inSize, [in](#) outSize, [in](#) initLaby, [in](#) initWalls, [in](#) initPac, [in](#) initStop)
- [function updateConnexion](#) ([in](#) obj, [in](#) indBit, [in](#) value)
- [function init](#) ([in](#) obj)
- [function orderer](#) ([in](#) obj, [in](#) vectIn)
- [function get\\_stop](#) ([in](#) obj)
- [function get\\_out](#) ([in](#) obj)

### Public Attributes

- Property [wallsBit](#)
- Property [pacmanBit](#)
- Property [modelLaby](#)
- Property [commandWalls](#)
- Property [commandPacman](#)
- Property [stopCondition](#)
- Property [in](#)
- Property [out](#)
- Property [stop](#)
- Property [whoPlay](#)

#### 4.10.1 Constructor & Destructor Documentation

#### 4.10.1.1 Wrapper()

```
function Wrapper (
    in inSize,
    in outSize,
    in initLaby,
    in initWalls,
    in initPac,
    in initStop )
```

### 4.10.2 Member Function Documentation

#### 4.10.2.1 get\_out()

```
function get_out (
    in obj )
```

#### 4.10.2.2 get\_stop()

```
function get_stop (
    in obj )
```

#### 4.10.2.3 init()

```
function init (
    in obj )
```

#### 4.10.2.4 orderer()

```
function orderer (
    in obj,
    in vectIn )
```

#### 4.10.2.5 updateConnexion()

```
function updateConnexion (
    in obj,
    in indBit,
    in value )
```



### 4.10.3 Member Data Documentation

#### 4.10.3.1 `commandPacman`

Property `commandPacman`

#### 4.10.3.2 `commandWalls`

Property `commandWalls`

#### 4.10.3.3 `in`

Property `in`

#### 4.10.3.4 `modelLaby`

Property `modelLaby`

#### 4.10.3.5 `out`

Property `out`

#### 4.10.3.6 `pacmanBit`

Property `pacmanBit`

#### 4.10.3.7 `stop`

Property `stop`

#### 4.10.3.8 stopCondition

Property stopCondition

#### 4.10.3.9 wallsBit

Property wallsBit

#### 4.10.3.10 whoPlay

Property whoPlay

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

- [Wrapper.m](#)

## Chapter 5

# File Documentation

### 5.1 automaton/AutomateGraph.m File Reference

#### Classes

- class [AutomateGraph](#)

### 5.2 automaton/mainLaby.m File Reference

### 5.3 automaton/modelGenerator/AutomatonSchedulingCreation.m File Reference

#### Functions

- [function](#) ()

#### 5.3.1 Function Documentation

##### 5.3.1.1 [function](#)()

`function ( )`

### 5.4 automaton/modelGenerator/AutomatonStrutureLabyCreation.m File Reference

#### Functions

- [function AutomatonStrutureLabyCreation](#) (in labySize, in playerPosition, in escapePosition)

### 5.4.1 Function Documentation

#### 5.4.1.1 AutomatonStrutureLabyCreation()

```
function AutomatonStrutureLabyCreation (
    in labySize,
    in playerPosition,
    in escapePosition )
```

## 5.5 automaton/modelGenerator/AutomatonWallsContraintsCreation.m File Reference

### Functions

- [function AutomatonWallsContraintsCreation](#) (in verticalsWalls, in horizontalsWalls, in FirstWallsMove)

### 5.5.1 Function Documentation

#### 5.5.1.1 AutomatonWallsContraintsCreation()

```
function AutomatonWallsContraintsCreation (
    in verticalsWalls,
    in horizontalsWalls,
    in FirstWallsMove )
```

## 5.6 automaton/modelGenerator/generer\_lab.m File Reference

### Functions

- [function generer\\_lab](#) (in Matrice\_Horizontale, in Matrice\_Verticale)

### 5.6.1 Function Documentation

#### 5.6.1.1 generer\_lab()

```
function generer_lab (
    in Matrice_Horizontale,
    in Matrice_Verticale )
```

## 5.7 automaton/modelGenerator/modelGenerator.m File Reference

## 5.8 automaton/modelGenerator/Plan\_desumaFunctions.m File Reference

### Functions

- [function writeStates](#) (in prefix, in nbrOfStates, in initialIndice, in markedStatesIndices)
- [function writeTransitions](#) (in prefix, in datas)
- [function SaveDESUMAFile](#) (in transitionsString, in statesString, in fileName)
- [function AutomatonStrutureLabyCreation](#) (in labySize, in playerPosition, in escapePosition)
- [function](#) ()

### 5.8.1 Function Documentation

#### 5.8.1.1 AutomatonStrutureLabyCreation()

```
function AutomatonStrutureLabyCreation (
    in labySize,
    in playerPosition,
    in escapePosition )
```

#### 5.8.1.2 function()

```
function ( )
```

#### 5.8.1.3 SaveDESUMAFile()

```
function SaveDESUMAFile (
    in transitionsString,
    in statesString,
    in fileName )
```

#### 5.8.1.4 writeStates()

```
function writeStates (
    in prefix,
    in nbrOfStates,
    in initialIndice,
    in markedStatesIndices )
```

#### 5.8.1.5 writeTransitions()

```
function writeTransitions (
    in prefix,
    in datas )
```

### 5.9 automaton/modelGenerator/SaveDESUMAFFile.m File Reference

#### Functions

- [function SaveDESUMAFFile](#) (in transitionsString, in statesString, in fileName)

#### 5.9.1 Function Documentation

##### 5.9.1.1 SaveDESUMAFFile()

```
function SaveDESUMAFFile (
    in transitionsString,
    in statesString,
    in fileName )
```

### 5.10 automaton/modelGenerator/writeStates.m File Reference

#### Functions

- [function writeStates](#) (in prefix, in nbrOfStates, in initialIndice, in markedStatesIndices)

#### 5.10.1 Function Documentation

##### 5.10.1.1 writeStates()

```
function writeStates (
    in prefix,
    in nbrOfStates,
    in initialIndice,
    in markedStatesIndices )
```

## 5.11 automaton/modelGenerator/writeTransitions.m File Reference

### Functions

- [function writeTransitions](#) (in prefix, in datas)

#### 5.11.1 Function Documentation

##### 5.11.1.1 writeTransitions()

```
function writeTransitions (
    in prefix,
    in datas )
```

## 5.12 automaton/optimalCommand/creationMatricetransition.m File Reference

### Functions

- [function creationMatricetransition](#) (in nameOfFileFSM, in cellOrder)

#### 5.12.1 Function Documentation

##### 5.12.1.1 creationMatricetransition()

```
function creationMatricetransition (
    in nameOfFileFSM,
    in cellOrder )
```

## 5.13 automaton/optimalCommand/getStateTransitionFSM.m File Reference

### Functions

- [function getStateTransitionFSM](#) (in nameOfFileFSM)

#### 5.13.1 Function Documentation

#### 5.13.1.1 `getStateTransitionFSM()`

```
function getStateTransitionFSM (
    in nameOfFileFSM )
```

### 5.14 `automaton/optimalCommand/getStateTransitionTXT.m` File Reference

#### Functions

- [function `getStateTransitionTXT`](#) (in `nameOfFileTXT`, in `ST`, in `SP`)

#### 5.14.1 Function Documentation

##### 5.14.1.1 `getStateTransitionTXT()`

```
function getStateTransitionTXT (
    in nameOfFileTXT,
    in ST,
    in SP )
```

### 5.15 `automaton/optimalCommand/main.m` File Reference

#### 5.16 `main.m` File Reference

### 5.17 `automaton/optimalCommand/optimalCommand.m` File Reference

#### Functions

- [function `optimalCommand`](#) (in `transitionsMatrix`, in `s_init`, in `s_final`)

#### 5.17.1 Function Documentation

##### 5.17.1.1 `optimalCommand()`

```
function optimalCommand (
    in transitionsMatrix,
    in s_init,
    in s_final )
```



## 5.18 automaton/optimalCommand/ParcourirMatricesTransitions.m File Reference

### Functions

- [function ParcourirMatricesTransitions](#) (in MatricesTransition, in Poids)

#### 5.18.1 Function Documentation

##### 5.18.1.1 ParcourirMatricesTransitions()

```
function ParcourirMatricesTransitions (
    in MatricesTransition,
    in Poids )
```

## 5.19 automaton/optimalCommand/rafineAutomaton.m File Reference

## 5.20 automaton/optimalCommand/rafineAutomatonClass.m File Reference

### Functions

- [function rafineAutomatonClass](#) (in A, in paternName)

#### 5.20.1 Function Documentation

##### 5.20.1.1 rafineAutomatonClass()

```
function rafineAutomatonClass (
    in A,
    in paternName )
```

## 5.21 automaton/ParrallelComposition.m File Reference

### Functions

- [function ParrallelComposition](#) (in A1, in A2)

### 5.21.1 Function Documentation

#### 5.21.1.1 ParrallelComposition()

```
function ParrallelComposition (
    in A1,
    in A2 )
```

## 5.22 automaton\_nd/Automaton.m File Reference

### Classes

- class [Automaton](#)

## 5.23 automaton\_nd/Labyrinthe.m File Reference

### Classes

- class [Labyrinthe](#)

## 5.24 automaton\_nd/modi\_main.m File Reference

## 5.25 CreatePituresAndVideo.m File Reference

### Functions

- [function CreatePituresAndVideo](#) (in n, in escape\_i, in labyState)

### 5.25.1 Function Documentation

#### 5.25.1.1 CreatePituresAndVideo()

```
function CreatePituresAndVideo (
    in n,
    in escape_i,
    in labyState )
```

## 5.26 CreatePituresAndVideo\_textured.m File Reference

### Functions

- [function CreatePituresAndVideo\\_textured](#) (in `n`, in `escape_i`, in `labyState`)

### 5.26.1 Function Documentation

#### 5.26.1.1 CreatePituresAndVideo\_textured()

```
function CreatePituresAndVideo_textured (
    in n,
    in escape_i,
    in labyState )
```

## 5.27 figure\_Laby.m File Reference

### Functions

- [function figure\\_Laby](#) (in `varargin`)
- [function figure\\_Laby\\_OpeningFcn](#) (in `hObject`, in `eventdata`, in `handles`, in `varargin`)
- [function figure\\_Laby\\_OutputFcn](#) (in `hObject`, in `eventdata`, in `handles`)
- [function ui\\_Callback](#) (in `hObject`, in `eventdata`, in `handles`)
- [function connect\\_Callback](#) (in `hObject`, in `eventdata`, in `handles`)
- [function createUIPacman](#) (in `handles`)
- [function createUIWalls](#) (in `handles`)
- [function createUIEscape](#) (in `handles`)
- [function updateUI](#) (in `handles`, in `out`)
- [function updateUIActiveCammand](#) (in `handles`)
- [function updateUIButton](#) (in `handles`)
- [function updateUIPlayer](#) (in `handles`, in `strPlayer`, in `position`)
- [function updateUIEscape](#) (in `elementToSet`, in `boolState`)
- [function updateUIWallsAround](#) (in `handles`, in `strElement`, in `wallsAround`)
- [function updateUIWalls](#) (in `wallsUI`, in `vertWalls`, in `horizWalls`)
- [function isOne](#) (in `boolCond`)
- [function updatePresenceDetectorDisplay](#) (in `elementToSet`, in `boolCondition`)
- [function resetUIConnection](#) (in `handles`)

### 5.27.1 Function Documentation

#### 5.27.1.1 connect\_Callback()

```
function connect_Callback (
    in hObject,
    in eventdata,
    in handles )
```

#### 5.27.1.2 createUIEscape()

```
function createUIEscape (
    in handles )
```

#### 5.27.1.3 createUIPacman()

```
function createUIPacman (
    in handles )
```

#### 5.27.1.4 createUIWalls()

```
function createUIWalls (
    in handles )
```

#### 5.27.1.5 figure\_Laby()

```
function figure_Laby (
    in varargin )
```

#### 5.27.1.6 figure\_Laby\_OpeningFcn()

```
function figure_Laby_OpeningFcn (
    in hObject,
    in eventdata,
    in handles,
    in varargin )
```

**5.27.1.7 figure\_Laby\_OutputFcn()**

```
function figure_Laby_OutputFcn (
    in hObject,
    in eventdata,
    in handles )
```

**5.27.1.8 isOne()**

```
function isOne (
    in boolCond )
```

**5.27.1.9 resetUIConnection()**

```
function resetUIConnection (
    in handles )
```

**5.27.1.10 ui\_Callback()**

```
function ui_Callback (
    in hObject,
    in eventdata,
    in handles )
```

**5.27.1.11 updatePresenceDetectorDisplay()**

```
function updatePresenceDetectorDisplay (
    in elementToSet,
    in boolCondition )
```

**5.27.1.12 updateUI()**

```
function updateUI (
    in handles,
    in out )
```

**5.27.1.13 updateUIActiveCammand()**

```
function updateUIActiveCammand (
    in handles )
```

**5.27.1.14 updateUIButton()**

```
function updateUIButton (
    in handles )
```

**5.27.1.15 updateUIEscape()**

```
function updateUIEscape (
    in elementToSet,
    in boolState )
```

**5.27.1.16 updateUIPlayer()**

```
function updateUIPlayer (
    in handles,
    in strPlayer,
    in position )
```

**5.27.1.17 updateUIWalls()**

```
function updateUIWalls (
    in wallsUI,
    in vertWalls,
    in horizWalls )
```

**5.27.1.18 updateUIWallsAround()**

```
function updateUIWallsAround (
    in handles,
    in strElement,
    in wallsAround )
```

## 5.28 ModelLaby.m File Reference

### Classes

- class [ModelLaby](#)  
*Class which contains the "fmg" structure of the labyrinth for 1 player*

## 5.29 ModelPacman.m File Reference

### Classes

- class [ModelPacman](#)

## 5.30 ModelSED.m File Reference

### Classes

- class [ModelSED](#)

## 5.31 ModelWalls.m File Reference

### Classes

- class [ModelWalls](#)

## 5.32 setColor.m File Reference

### Functions

- function [setColor](#) (in img, in imgRef, in colors, in indice)

#### 5.32.1 Function Documentation

##### 5.32.1.1 setColor()

```
function setColor (
    in img,
    in imgRef,
    in colors,
    in indice )
```

### 5.33 Simulation.m File Reference

### 5.34 StopCondition.m File Reference

#### Classes

- class [StopCondition](#)

### 5.35 testUI.m File Reference

### 5.36 THEplan.m File Reference

### 5.37 visupacman.m File Reference

### 5.38 visupacman2.m File Reference

### 5.39 wallsBorder.m File Reference

#### Functions

- function [wallsBorder](#) (in walls)

#### 5.39.1 Function Documentation

##### 5.39.1.1 wallsBorder()

```
function wallsBorder (  
    in walls )
```

### 5.40 Wrapper.m File Reference

#### Classes

- class [Wrapper](#)



# Index

- accessibilityAutomate
  - AutomateGraph, 8
- adaptTourLangage
  - AutomateGraph, 8
- addWord2Langage
  - AutomateGraph, 8
- affichage
  - Labyrinthe, 14
- AjoutTransitionStable
  - Automaton, 11
- AutomateAccessible
  - Automaton, 11
- AutomateAccessibleIncertain
  - Automaton, 11
- AutomateGraph, 7
  - accessibilityAutomate, 8
  - adaptTourLangage, 8
  - addWord2Langage, 8
  - AutomateGraph, 7
  - export2DESUMA, 8
  - FSM2Automata, 8
  - langage, 9
  - matrices2vector, 8
  - matrixTrans, 9
  - PathRecherche, 8
  - state, 9
  - structAutomata2vectorAutomata, 9
  - transition, 10
  - vector, 10
  - vector2matrices, 9
  - vector2structAutomata, 9
- Automaton, 10
  - AjoutTransitionStable, 11
  - AutomateAccessible, 11
  - AutomateAccessibleIncertain, 11
  - Automaton, 10
  - CreationFonctionTransitionEnsembleDesParties, 11
  - Ensemble2Etats, 11
  - EnsemblesContenantEtat, 11
  - Etats2Ensemble, 12
  - Evolution, 12
  - Fct, 12
  - MatricesTransition, 12
  - NomsEvenements, 12
  - PathRecherche, 12
- automaton/AutomateGraph.m, 31
- automaton/ParrallelComposition.m, 37
- automaton/mainLaby.m, 31
- automaton/modelGenerator/AutomatonScheduling↔
  - Creation.m, 31
- automaton/modelGenerator/AutomatonStrutureLaby↔
  - Creation.m, 31
- automaton/modelGenerator/AutomatonWallsContraints↔
  - Creation.m, 32
- automaton/modelGenerator/Plan\_desumaFunctions.m, 33
- automaton/modelGenerator/SaveDESUMAFFile.m, 34
- automaton/modelGenerator/generer\_lab.m, 32
- automaton/modelGenerator/modelGenerator.m, 33
- automaton/modelGenerator/writeStates.m, 34
- automaton/modelGenerator/writeTransitions.m, 35
- automaton/optimalCommand/ParcourirMatrices↔
  - Transitions.m, 37
- automaton/optimalCommand/creationMatricetransition.↔
  - m, 35
- automaton/optimalCommand/getStateTransitionFSM.m, 35
- automaton/optimalCommand/getStateTransitionTXT.m, 36
- automaton/optimalCommand/main.m, 36
- automaton/optimalCommand/optimalCommand.m, 36
- automaton/optimalCommand/rafineAutomaton.m, 37
- automaton/optimalCommand/rafineAutomatonClass.m, 37
- automaton\_nd/Automaton.m, 38
- automaton\_nd/Labyrinthe.m, 38
- automaton\_nd/modi\_main.m, 38
- AutomatonSchedulingCreation.m
  - function, 31
- AutomatonStrutureLabyCreation
  - AutomatonStrutureLabyCreation.m, 32
  - Plan\_desumaFunctions.m, 33
- AutomatonStrutureLabyCreation.m
  - AutomatonStrutureLabyCreation, 32
- AutomatonWallsContraintsCreation
  - AutomatonWallsContraintsCreation.m, 32
- AutomatonWallsContraintsCreation.m
  - AutomatonWallsContraintsCreation, 32
- commandPacman
  - Wrapper, 29
- commandWalls
  - Wrapper, 29
- connect\_Callback
  - figure\_Laby.m, 39
- CreatePituresAndVideo
  - CreatePituresAndVideo.m, 38
- CreatePituresAndVideo.m, 38

- CreatePituresAndVideo, 38
- CreatePituresAndVideo\_textured
  - CreatePituresAndVideo\_textured.m, 39
- CreatePituresAndVideo\_textured.m, 39
  - CreatePituresAndVideo\_textured, 39
- createUIEscape
  - figure\_Laby.m, 40
- createUIPacman
  - figure\_Laby.m, 40
- createUIWalls
  - figure\_Laby.m, 40
- CreationFonctionTransitionEnsembleDesParties
  - Automaton, 11
- creationMatricetransition
  - creationMatricetransition.m, 35
- creationMatricetransition.m
  - creationMatricetransition, 35
- Ensemble2Etats
  - Automaton, 11
- EnsemblesContenantEtat
  - Automaton, 11
- Etats2Ensemble
  - Automaton, 12
- Etats\_Finaux
  - Labyrinthe, 14
- Etats\_Initiaux
  - Labyrinthe, 15
- Evolution
  - Automaton, 12
- export2DESUMA
  - AutomateGraph, 8
- f
  - ModelLaby, 17
  - ModelPacman, 19
  - ModelSED, 21
  - ModelWalls, 23
  - StopCondition, 25, 26
- FSM2Automata
  - AutomateGraph, 8
- Fct
  - Automaton, 12
- figure\_Laby
  - figure\_Laby.m, 40
- figure\_Laby.m, 39
  - connect\_Callback, 39
  - createUIEscape, 40
  - createUIPacman, 40
  - createUIWalls, 40
  - figure\_Laby, 40
  - figure\_Laby\_OpeningFcn, 40
  - figure\_Laby\_OutputFcn, 40
  - isOne, 41
  - resetUIConnection, 41
  - ui\_Callback, 41
  - updatePresenceDetectorDisplay, 41
  - updateUIActiveCammand, 41
  - updateUIButton, 42
  - updateUIEscape, 42
  - updateUIPlayer, 42
  - updateUIWalls, 42
  - updateUIWallsAround, 42
  - updateUI, 41
- figure\_Laby\_OpeningFcn
  - figure\_Laby.m, 40
- figure\_Laby\_OutputFcn
  - figure\_Laby.m, 40
- function
  - AutomatonSchedulingCreation.m, 31
  - Plan\_desumaFunctions.m, 33
- g
  - ModelLaby, 17
  - ModelPacman, 19
  - ModelSED, 21
  - ModelWalls, 23
  - StopCondition, 26
- generer\_lab
  - generer\_lab.m, 32
- generer\_lab.m
  - generer\_lab, 32
- get\_out
  - Wrapper, 28
- get\_stop
  - Wrapper, 28
- getStateTransitionFSM.m
  - getStateTransitionFSM, 35
- getStateTransitionFSM
  - getStateTransitionFSM.m, 35
- getStateTransitionTXT.m
  - getStateTransitionTXT, 36
- getStateTransitionTXT
  - getStateTransitionTXT.m, 36
- handle, 13
- i
  - ModelWalls, 24
- in
  - Wrapper, 29
- incertain
  - Labyrinthe, 14
- init
  - Wrapper, 28
- initialState
  - ModelLaby, 17
  - ModelPacman, 20
  - ModelSED, 21
  - ModelWalls, 24
  - StopCondition, 26
- isOne
  - figure\_Laby.m, 41
- jusqu\_au\_mur
  - Labyrinthe, 14
- Labyrinthe, 13

- affichage, 14
- Etats\_Finaux, 14
- Etats\_Initiaux, 15
- incertain, 14
- jusqu\_au\_mur, 14
- Labyrinthe, 14
- MursHorizontaux, 15
- MursVerticaux, 15
- Pas\_a\_pas, 14
- langage
  - AutomateGraph, 9
- m
  - ModelLaby, 17
  - ModelPacman, 19
  - ModelSED, 21
  - ModelWalls, 23
  - StopCondition, 26
- main.m, 36
- matrices2vector
  - AutomateGraph, 8
- MatricesTransition
  - Automaton, 12
- matrixTrans
  - AutomateGraph, 9
- ModelLaby, 15
  - f, 17
  - g, 17
  - initialState, 17
  - m, 17
  - ModelLaby, 16
  - presentState, 18
- modelLaby
  - Wrapper, 29
- ModelLaby.m, 43
- ModelPacman, 18
  - f, 19
  - g, 19
  - initialState, 20
  - m, 19
  - ModelPacman, 19
  - presentState, 20
- ModelPacman.m, 43
- ModelSED.m, 43
- ModelSED, 20
  - f, 21
  - g, 21
  - initialState, 21
  - m, 21
  - presentState, 22
- ModelWalls, 22
  - f, 23
  - g, 23
  - i, 24
  - initialState, 24
  - m, 23
  - ModelWalls, 23
  - presentState, 24
  - val, 24
- ModelWalls.m, 43
- MursHorizontaux
  - Labyrinthe, 15
- MursVerticaux
  - Labyrinthe, 15
- NomsEvenements
  - Automaton, 12
- optimalCommand
  - optimalCommand.m, 36
- optimalCommand.m
  - optimalCommand, 36
- orderer
  - Wrapper, 28
- out
  - Wrapper, 29
- pacmanBit
  - Wrapper, 29
- ParcourirMatricesTransitions
  - ParcourirMatricesTransitions.m, 37
- ParcourirMatricesTransitions.m
  - ParcourirMatricesTransitions, 37
- ParrallelComposition
  - ParrallelComposition.m, 38
- ParrallelComposition.m
  - ParrallelComposition, 38
- Pas\_a\_pas
  - Labyrinthe, 14
- PathResearche
  - AutomateGraph, 8
  - Automaton, 12
- Plan\_desumaFunctions.m
  - AutomatonStrutureLabyCreation, 33
  - function, 33
  - SaveDESUMAFFile, 33
  - writeStates, 33
  - writeTransitions, 33
- presentState
  - ModelLaby, 18
  - ModelPacman, 20
  - ModelSED, 22
  - ModelWalls, 24
  - StopCondition, 27
- rafineAutomatonClass
  - rafineAutomatonClass.m, 37
- rafineAutomatonClass.m
  - rafineAutomatonClass, 37
- resetUIConnection
  - figure\_Laby.m, 41
- SaveDESUMAFFile
  - Plan\_desumaFunctions.m, 33
  - SaveDESUMAFFile.m, 34
- SaveDESUMAFFile.m
  - SaveDESUMAFFile, 34
- setColor

- setColor.m, 43
- setColor.m, 43
  - setColor, 43
- Simulation.m, 44
- state
  - AutomateGraph, 9
- stop
  - Wrapper, 29
- StopCondition, 25
  - f, 25, 26
  - g, 26
  - initialState, 26
  - m, 26
  - presentState, 27
  - StopCondition, 25
- stopCondition
  - Wrapper, 29
- StopCondition.m, 44
- structAutomata2vectorAutomata
  - AutomateGraph, 9
- THEplan.m, 44
- testUI.m, 44
- transition
  - AutomateGraph, 10
- ui\_Callback
  - figure\_Laby.m, 41
- updateConnexion
  - Wrapper, 28
- updatePresenceDetectorDisplay
  - figure\_Laby.m, 41
- updateUIActiveCammand
  - figure\_Laby.m, 41
- updateUIButton
  - figure\_Laby.m, 42
- updateUIEscape
  - figure\_Laby.m, 42
- updateUIPlayer
  - figure\_Laby.m, 42
- updateUIWalls
  - figure\_Laby.m, 42
- updateUIWallsAround
  - figure\_Laby.m, 42
- updateUI
  - figure\_Laby.m, 41
- val
  - ModelWalls, 24
- vector
  - AutomateGraph, 10
- vector2matrices
  - AutomateGraph, 9
- vector2structAutomata
  - AutomateGraph, 9
- visupacman.m, 44
- visupacman2.m, 44
- wallsBit
  - Wrapper, 30
- wallsBorder
  - wallsBorder.m, 44
- wallsBorder.m, 44
  - wallsBorder, 44
- whoPlay
  - Wrapper, 30
- Wrapper, 27
  - commandPacman, 29
  - commandWalls, 29
  - get\_out, 28
  - get\_stop, 28
  - in, 29
  - init, 28
  - modellaby, 29
  - orderer, 28
  - out, 29
  - pacmanBit, 29
  - stop, 29
  - stopCondition, 29
  - updateConnexion, 28
  - wallsBit, 30
  - whoPlay, 30
  - Wrapper, 27
- Wrapper.m, 44
- writeStates
  - Plan\_desumaFunctions.m, 33
  - writeStates.m, 34
- writeStates.m
  - writeStates, 34
- writeTransitions
  - Plan\_desumaFunctions.m, 33
  - writeTransitions.m, 35
- writeTransitions.m
  - writeTransitions, 35