

Labyrinth - One Player mode

Generated by Doxygen 1.8.13

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	AutomateGraph Class Reference	7
4.1.1	Constructor & Destructor Documentation	8
4.1.1.1	AutomateGraph()	8
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()	9
4.1.2.5	FSM2Automata()	9
4.1.2.6	matrices2vector()	9
4.1.2.7	PathResearche()	9
4.1.2.8	structAutomata2vectorAutomata()	9
4.1.2.9	vector2matrices()	9
4.1.2.10	vector2structAutomata()	10
4.1.3	Member Data Documentation	10

4.1.3.1	langage	10
4.1.3.2	matrixTrans	10
4.1.3.3	state	10
4.1.3.4	transition	10
4.1.3.5	vector	10
4.2	Automaton Class Reference	11
4.2.1	Constructor & Destructor Documentation	11
4.2.1.1	Automaton()	11
4.2.2	Member Function Documentation	12
4.2.2.1	AjoutTransitionStable()	12
4.2.2.2	AutomateAccessible()	12
4.2.2.3	AutomateAccessibleIncertain()	12
4.2.2.4	CreationFonctionTransitionEnsembleDesParties()	12
4.2.2.5	Ensemble2Etats()	12
4.2.2.6	EnsemblesContenantEtat()	13
4.2.2.7	Etats2Ensemble()	13
4.2.2.8	Evolution()	13
4.2.2.9	PathRecherche()	13
4.2.3	Member Data Documentation	13
4.2.3.1	Fct	13
4.2.3.2	MatricesTransition	13
4.2.3.3	NomsEvenements	14
4.3	handle Class Reference	14
4.4	Labyrinthe Class Reference	14
4.4.1	Constructor & Destructor Documentation	15
4.4.1.1	Labyrinthe()	15
4.4.2	Member Function Documentation	15
4.4.2.1	affichage()	15
4.4.2.2	incertain()	15
4.4.2.3	jusqu_au_mur()	16

4.4.2.4	Pas_a_pas()	16
4.4.3	Member Data Documentation	16
4.4.3.1	Etats_Finaux	16
4.4.3.2	Etats_Initiaux	16
4.4.3.3	MursHorizontaux	16
4.4.3.4	MursVerticaux	16
4.5	ModelLaby Class Reference	17
4.5.1	Detailed Description	18
4.5.2	Constructor & Destructor Documentation	18
4.5.2.1	ModelLaby()	18
4.5.3	Member Function Documentation	18
4.5.3.1	f()	19
4.5.3.2	g() [1/2]	19
4.5.3.3	g() [2/2]	19
4.5.3.4	m()	19
4.5.4	Member Data Documentation	19
4.5.4.1	initialState	19
4.5.4.2	presentState	20
4.6	ModelPacman Class Reference	20
4.6.1	Constructor & Destructor Documentation	21
4.6.1.1	ModelPacman()	21
4.6.2	Member Function Documentation	21
4.6.2.1	f()	21
4.6.2.2	g() [1/2]	21
4.6.2.3	g() [2/2]	22
4.6.2.4	m()	22
4.6.3	Member Data Documentation	22
4.6.3.1	initialState	22
4.6.3.2	presentState	22
4.7	ModelSED Class Reference	23

4.7.1	Member Function Documentation	23
4.7.1.1	f()	24
4.7.1.2	g()	24
4.7.1.3	m()	24
4.7.2	Member Data Documentation	24
4.7.2.1	initialState	24
4.7.2.2	presentState	24
4.8	ModelWalls Class Reference	25
4.8.1	Constructor & Destructor Documentation	26
4.8.1.1	ModelWalls()	26
4.8.2	Member Function Documentation	26
4.8.2.1	f() [1/2]	26
4.8.2.2	f() [2/2]	26
4.8.2.3	g() [1/2]	26
4.8.2.4	g() [2/2]	27
4.8.2.5	m()	27
4.8.3	Member Data Documentation	27
4.8.3.1	i	27
4.8.3.2	initialState	27
4.8.3.3	presentState	27
4.8.3.4	val	27
4.9	StopCondition Class Reference	28
4.9.1	Constructor & Destructor Documentation	29
4.9.1.1	StopCondition()	29
4.9.2	Member Function Documentation	29
4.9.2.1	f() [1/2]	29
4.9.2.2	f() [2/2]	29
4.9.2.3	g() [1/2]	29
4.9.2.4	g() [2/2]	30
4.9.2.5	m()	30

4.9.3	Member Data Documentation	30
4.9.3.1	initialState	30
4.9.3.2	presentState	30
4.10	Wrapper Class Reference	31
4.10.1	Constructor & Destructor Documentation	31
4.10.1.1	Wrapper()	32
4.10.2	Member Function Documentation	32
4.10.2.1	get_out()	32
4.10.2.2	get_stop()	32
4.10.2.3	init()	32
4.10.2.4	orderer()	32
4.10.2.5	updateConnexion()	32
4.10.3	Member Data Documentation	33
4.10.3.1	commandPacman	33
4.10.3.2	commandWalls	33
4.10.3.3	in	33
4.10.3.4	modelLaby	33
4.10.3.5	out	33
4.10.3.6	pacmanBit	33
4.10.3.7	stop	33
4.10.3.8	stopCondition	34
4.10.3.9	wallsBit	34
4.10.3.10	whoPlay	34

5 File Documentation	35
5.1 automaton/AutomateGraph.m File Reference	35
5.2 automaton/mainLaby.m File Reference	35
5.3 automaton/modelGenerator/AutomatonSchedulingCreation.m File Reference	35
5.3.1 Function Documentation	35
5.3.1.1 function()	35
5.4 automaton/modelGenerator/AutomatonStrutureLabyCreation.m File Reference	35
5.4.1 Function Documentation	36
5.4.1.1 AutomatonStrutureLabyCreation()	36
5.5 automaton/modelGenerator/AutomatonWallsConstraintsCreation.m File Reference	36
5.5.1 Function Documentation	36
5.5.1.1 AutomatonWallsConstraintsCreation()	36
5.6 automaton/modelGenerator/generer_lab.m File Reference	36
5.6.1 Function Documentation	36
5.6.1.1 generer_lab()	36
5.7 automaton/modelGenerator/modelGenerator.m File Reference	37
5.8 automaton/modelGenerator/Plan_desumaFunctions.m File Reference	37
5.8.1 Function Documentation	37
5.8.1.1 AutomatonStrutureLabyCreation()	37
5.8.1.2 function()	37
5.8.1.3 SaveDESUMAFile()	37
5.8.1.4 writeStates()	37
5.8.1.5 writeTransitions()	38
5.9 automaton/modelGenerator/SaveDESUMAFile.m File Reference	38
5.9.1 Function Documentation	38
5.9.1.1 SaveDESUMAFile()	38
5.10 automaton/modelGenerator/writeStates.m File Reference	38
5.10.1 Function Documentation	38
5.10.1.1 writeStates()	38
5.11 automaton/modelGenerator/writeTransitions.m File Reference	39

5.11.1	Function Documentation	39
5.11.1.1	writeTransitions()	39
5.12	automaton/optimalCommand/creationMatricetransition.m File Reference	39
5.12.1	Function Documentation	39
5.12.1.1	creationMatricetransition()	39
5.13	automaton/optimalCommand/getStateTransitionFSM.m File Reference	39
5.13.1	Function Documentation	39
5.13.1.1	getStateTransitionFSM()	40
5.14	automaton/optimalCommand/getStateTransitionTXT.m File Reference	40
5.14.1	Function Documentation	40
5.14.1.1	getStateTransitionTXT()	40
5.15	automaton/optimalCommand/main.m File Reference	40
5.16	main.m File Reference	40
5.17	automaton/optimalCommand/optimalCommand.m File Reference	40
5.17.1	Function Documentation	40
5.17.1.1	optimalCommand()	40
5.18	automaton/optimalCommand/ParcourirMatricesTransitions.m File Reference	41
5.18.1	Function Documentation	41
5.18.1.1	ParcourirMatricesTransitions()	41
5.19	automaton/optimalCommand/rafineAutomaton.m File Reference	41
5.20	automaton/optimalCommand/rafineAutomatonClass.m File Reference	41
5.20.1	Function Documentation	41
5.20.1.1	rafineAutomatonClass()	41
5.21	automaton/ParrallelComposition.m File Reference	41
5.21.1	Function Documentation	42
5.21.1.1	ParrallelComposition()	42
5.22	automaton_nd/Automaton.m File Reference	42
5.23	automaton_nd/Labyrinthe.m File Reference	42
5.24	automaton_nd/modi_main.m File Reference	42
5.25	CreatePituresAndVideo.m File Reference	42

5.25.1	Function Documentation	42
5.25.1.1	CreatePituresAndVideo()	42
5.26	CreatePituresAndVideo_textured.m File Reference	43
5.26.1	Function Documentation	43
5.26.1.1	CreatePituresAndVideo_textured()	43
5.27	figure_Laby.m File Reference	43
5.27.1	Function Documentation	44
5.27.1.1	connect_Callback()	44
5.27.1.2	createUIEscape()	44
5.27.1.3	createUIPacman()	46
5.27.1.4	createUIWalls()	46
5.27.1.5	figure_Laby()	47
5.27.1.6	figure_Laby_OpeningFcn()	47
5.27.1.7	figure_Laby_OutputFcn()	47
5.27.1.8	isOne()	48
5.27.1.9	resetUIConnection()	48
5.27.1.10	ui_Callback()	48
5.27.1.11	updatePresenceDetectorDisplay()	49
5.27.1.12	updateUI()	49
5.27.1.13	updateUIActiveCammand()	50
5.27.1.14	updateUIButton()	50
5.27.1.15	updateUIEscape()	50
5.27.1.16	updateUIPlayer()	51
5.27.1.17	updateUIWalls()	51
5.27.1.18	updateUIWallsAround()	51
5.28	ModelLaby.m File Reference	52
5.29	ModelPacman.m File Reference	52
5.30	ModelSED.m File Reference	52
5.31	ModelWalls.m File Reference	52
5.32	setColor.m File Reference	52
5.32.1	Function Documentation	52
5.32.1.1	setColor()	52
5.33	Simulation.m File Reference	53
5.34	StopCondition.m File Reference	53
5.35	testUI.m File Reference	53
5.36	THEplan.m File Reference	53
5.37	visupacman.m File Reference	53
5.38	visupacman2.m File Reference	53
5.39	wallsBorder.m File Reference	53
5.39.1	Function Documentation	53
5.39.1.1	wallsBorder()	53
5.40	Wrapper.m File Reference	53

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

AutomateGraph	7
Automaton	11
handle	14
ModelSED	23
ModelLaby	17
ModelPacman	20
ModelWalls	25
StopCondition	28
Labyrinthe	14
Wrapper	31

Chapter 2

Class Index

2.1 Class List

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

AutomateGraph	7
Automaton	11
handle	14
Labyrinthe	14
ModellLaby	
Class which contains the "fmg" structure of the labyrinth for 1 player	17
ModelPacman	20
ModelSED	23
ModelWalls	25
StopCondition	28
Wrapper	31

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

CreatePituresAndVideo.m	42
CreatePituresAndVideo_textured.m	43
figure_Laby.m	43
main.m	40
ModelLaby.m	52
ModelPacman.m	52
ModelSED.m	52
ModelWalls.m	52
setColor.m	52
Simulation.m	53
StopCondition.m	53
testUI.m	53
THEplan.m	53
visupacman.m	53
visupacman2.m	53
wallsBorder.m	53
Wrapper.m	53
automaton/AutomateGraph.m	35
automaton/mainLaby.m	35
automaton/ParrallelComposition.m	41
automaton/modelGenerator/AutomatonSchedulingCreation.m	35
automaton/modelGenerator/AutomatonStrutureLabyCreation.m	35
automaton/modelGenerator/AutomatonWallsContraintsCreation.m	36
automaton/modelGenerator/generer_lab.m	36
automaton/modelGenerator/modelGenerator.m	37
automaton/modelGenerator/Plan_desumaFunctions.m	37
automaton/modelGenerator/SaveDESUMAFFile.m	38
automaton/modelGenerator/writeStates.m	38
automaton/modelGenerator/writeTransitions.m	39
automaton/optimalCommand/creationMatricetransition.m	39
automaton/optimalCommand/getStateTransitionFSM.m	39
automaton/optimalCommand/getStateTransitionTXT.m	40
automaton/optimalCommand/main.m	40
automaton/optimalCommand/optimalCommand.m	40
automaton/optimalCommand/ParcourirMatricesTransitions.m	41

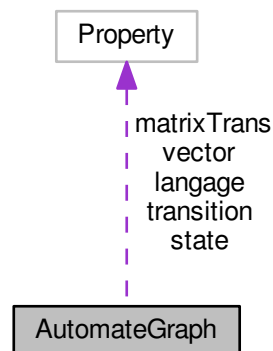
automaton/optimalCommand/ rafineAutomaton.m	41
automaton/optimalCommand/ rafineAutomatonClass.m	41
automaton_nd/ Automaton.m	42
automaton_nd/ Labyrinthe.m	42
automaton_nd/ modi_main.m	42

Chapter 4

Class Documentation

4.1 AutomateGraph Class Reference

Collaboration diagram for AutomateGraph:



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

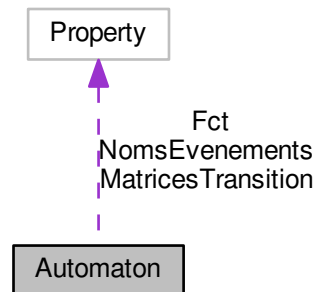
Property vector

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

- automaton/[AutomateGraph.m](#)

4.2 Automaton Class Reference

Collaboration diagram for Automaton:



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

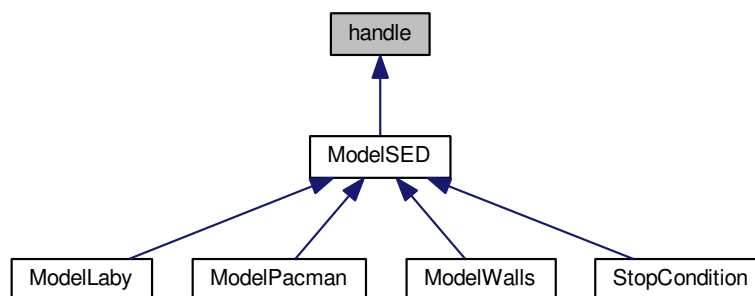
Property NomsEvenements

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

- automaton_nd/[Automaton.m](#)

4.3 handle Class Reference

Inheritance diagram for handle:

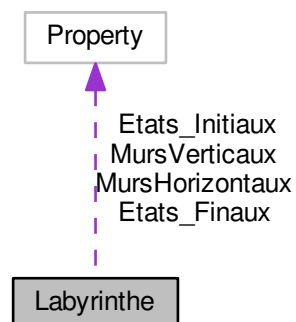


The documentation for this class was generated from the following file:

- [ModelSED.m](#)

4.4 Labyrinthe Class Reference

Collaboration diagram for Labyrinthe:



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

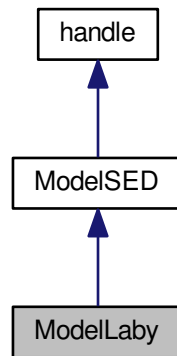
The documentation for this class was generated from the following file:

- automaton_nd/[Labyrinthe.m](#)

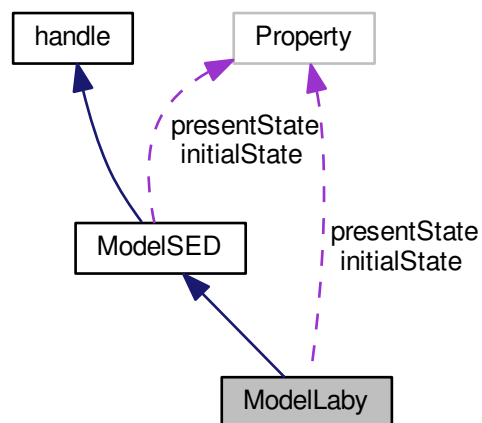
4.5 Modellaby Class Reference

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

Inheritance diagram for Modellaby:



Collaboration diagram for Modellaby:



Public Member Functions

- [function Modellaby](#) (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 Modellaby()

```
function Modellaby (
    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`

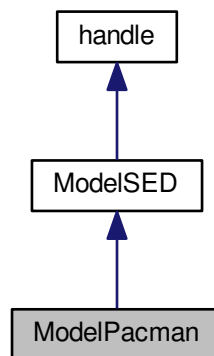
Data Structure of the current state of Labyrinth. It contains "wallsV", "wallsH" (2 matrix for the walls), "escape" and "pacman", a Cartesian position of current position of escape and pacman and 'wallsAroundPacman' A vector indicating the presence of a wall around the Pacman for the 4 directions Up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

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

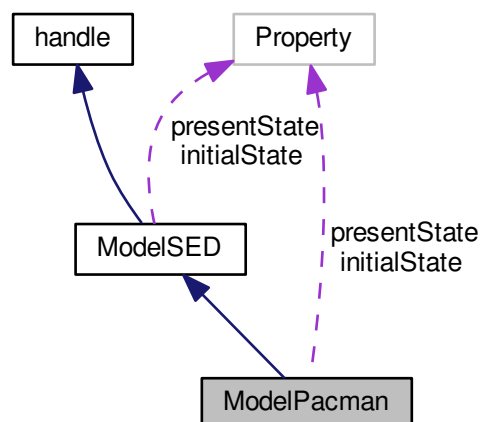
- [ModelLaby.m](#)

4.6 ModelPacman Class Reference

Inheritance diagram for ModelPacman:



Collaboration 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`

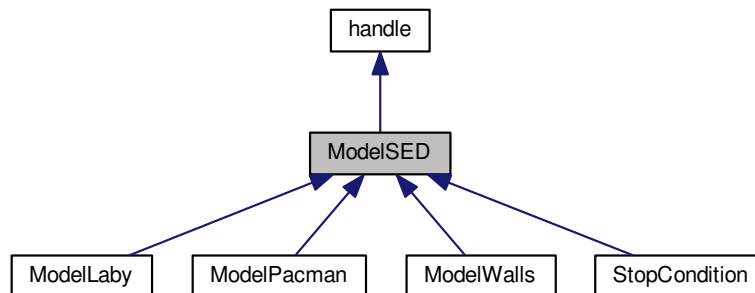
Property `presentState`

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

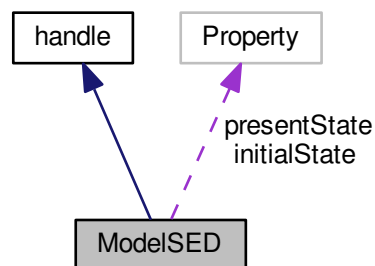
- [ModelPacman.m](#)

4.7 ModelSED Class Reference

Inheritance diagram for ModelSED:



Collaboration 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`

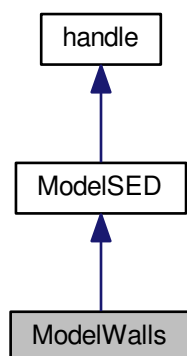
Property `presentState`

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

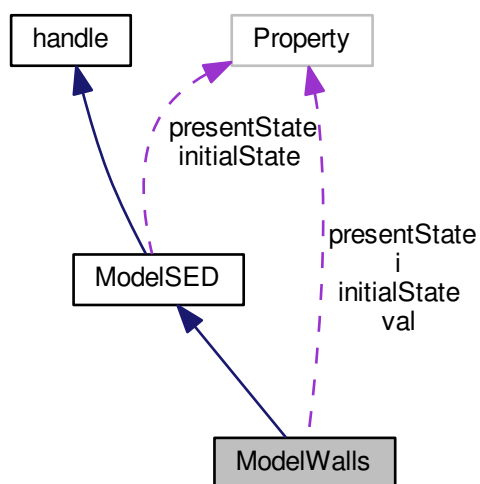
- [ModelSED.m](#)

4.8 ModelWalls Class Reference

Inheritance diagram for ModelWalls:



Collaboration 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`

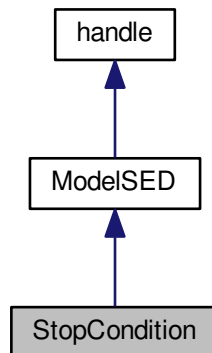
Property `val`

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

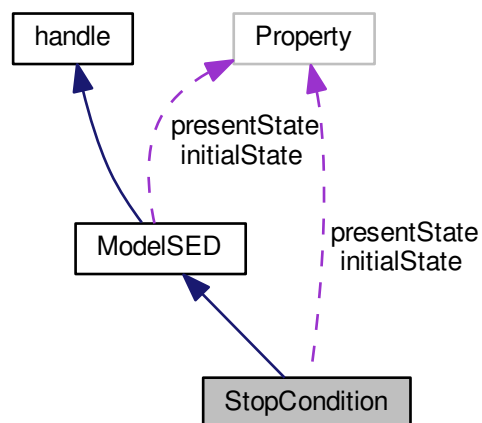
- [ModelWalls.m](#)

4.9 StopCondition Class Reference

Inheritance diagram for StopCondition:



Collaboration 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`

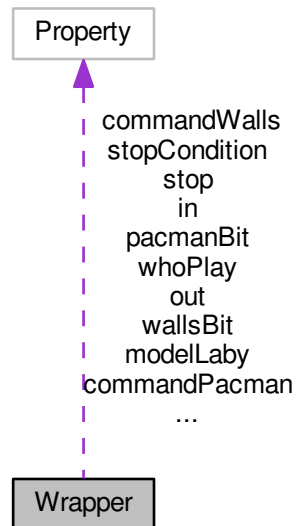
Property `presentState`

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

- [StopCondition.m](#)

4.10 Wrapper Class Reference

Collaboration diagram for Wrapper:



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 SaveDESUMAFFile](#) (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 SaveDESUMAFFile()

```
function SaveDESUMAFFile (
    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`)
figure_Laby.m
- [function figure_Laby_OpeningFcn](#) (in `hObject`, in `eventdata`, in `handles`, in `varargin`)
initialization function.
- [function figure_Laby_OutputFcn](#) (in `hObject`, in `eventdata`, in `handles`)
Automatic generated function by GUI.
- [function ui_Callback](#) (in `hObject`, in `eventdata`, in `handles`)
Callback for all the action's buttons (see detailed explications).
- [function connect_Callback](#) (in `hObject`, in `eventdata`, in `handles`)
Callback for all the connection's buttons (see detailed explications).
- [function createUIPacman](#) (in `handles`)
Creation of the graphical object "pacman".
- [function createUIWalls](#) (in `handles`)
Creation of the graphical objects "walls".
- [function createUIEscape](#) (in `handles`)
Creation of the graphical objects "escape".
- [function updateUI](#) (in `handles`, in `out`)
This function update all graphicals element who can change.
- [function updateUIActiveCammand](#) (in `handles`)
Update visibility of control panel, connection and step button.
- [function updateUIButton](#) (in `handles`)
Show the needed moving buttons.
- [function updateUIPlayer](#) (in `handles`, in `strPlayer`, in `position`)
Update graphical place of a player (only pacman in this case).
- [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 )
```

Callback for all the connection's buttons (see detailed explications).

in the following image, buttons marked with a red arrow lanch this Callback.

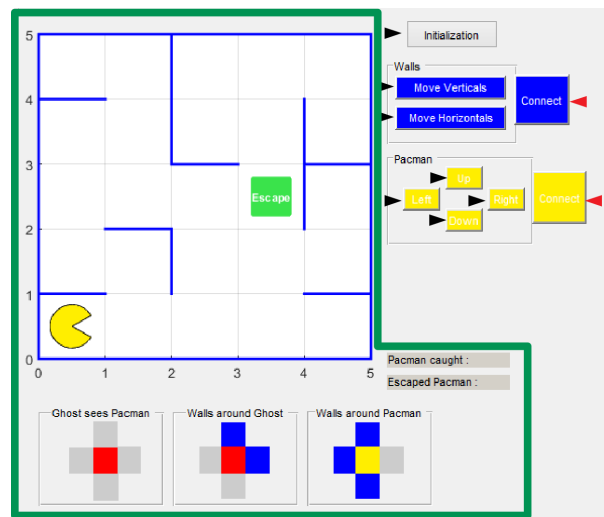


Figure 5.1 button's type of GUI

This callback lanch updateConnexion method of [Wrapper](#) class, which modify what command are automatic.

Parameters

<i>hObject</i>	handle to actived button
<i>eventdata</i>	reserved - to be defined in a future version of MATLAB
<i>handles</i>	structure with handles and user data (see GUIDATA)

5.27.1.2 createUIEscape()

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

Creation of the graphical objects "escape".

The escape is created whit a rectangle and and text box. It's stored into handles in 'escape'.

Parameters

<i>handles</i>	structure with handles and user data (see GUIDATA)
----------------	--

Returns

h the updated structure with handles and user data (see GUIDATA)

5.27.1.3 createUIPacman()

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

Creation of the graphical object "pacman".

The pacman is created by using the patch function and store into the handle in 'pacman'.

Parameters

<i>handles</i>	structure with handles and user data (see GUIDATA)
----------------	--

Returns

h the updated structure with handles and user data (see GUIDATA)

5.27.1.4 createUIWalls()

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

Creation of the graphical objects "walls".

The walls are created as two line elmenents matrix. They are stored into handles in 'walls'.

The first matrix is for the verticals walls and named 'horizontals' and the second called 'verticals' for the verticals walls.

All possible walls are created and it is by making them visible or invisible that they appear or disappear.

Parameters

<i>handles</i>	structure with handles and user data (see GUIDATA)
----------------	--

Returns

h the updated structure with handles and user data (see GUIDATA)

5.27.1.5 figure_Laby()

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

figure_Laby.m

Script linked to the graphical interface which contain all the graphical functions. This file contain also the instance of [Wrapper](#) class. All the handles of graphical elements and instance of class are stored into the "handles" structure. function call when figure_Laby si open. It's initialize the UI.

Parameters

<i>varargin</i>	Several inputs.
-----------------	-----------------

Returns

varargout Several Outputs.

5.27.1.6 figure_Laby_OpeningFcn()

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

initialization function.

It's where is initialize the parameters of the labyrinth and all the commands in the section "INITIAL PARAMETERS OF THE LABYRINTH AND THE COMMANDS".

Parameters

<i>hObject</i>	handle to figure
<i>eventdata</i>	reserved - to be defined in a future version of MATLAB
<i>handles</i>	structure with handles and user data (see GUIDATA)
<i>varargin</i>	command line arguments to figure_Laby (see VARARGIN)

5.27.1.7 figure_Laby_OutputFcn()

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

Automatic generated function by GUI.

Parameters

<i>hObject</i>	handle to figure
<i>eventdata</i>	reserved - to be defined in a future version of MATLAB
<i>handles</i>	structure with handles and user data (see GUIDATA)

Returns

varargout cell array for returning output args (see VARARGOUT);

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 )
```

Callback for all the action's buttons (see detailed explications).

in the following image, buttons marked with a black arrow lanch this Callback.

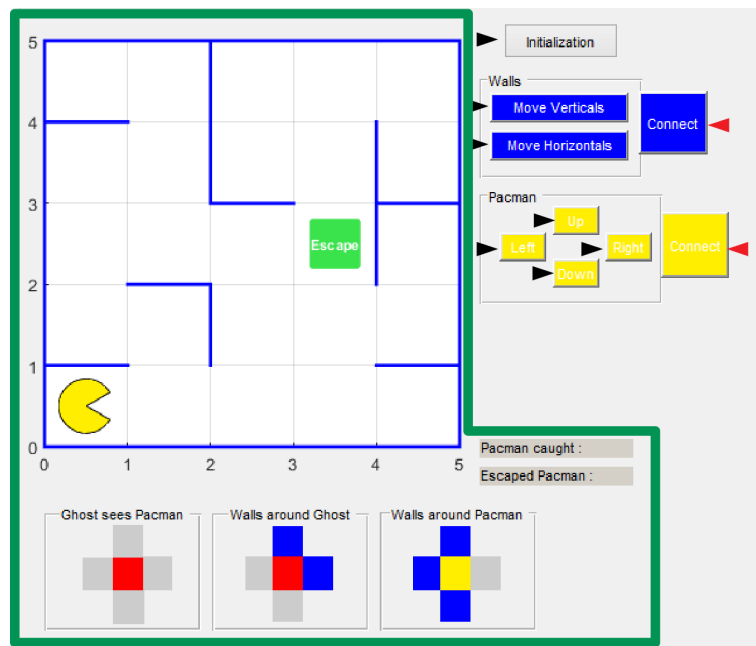


Figure 5.2 button's type of GUI

This callback launch orderer method of [Wrapper](#) class, which allows the simulation to evolve.

Parameters

<i>hObject</i>	handle to actived button
<i>eventdata</i>	reserved - to be defined in a future version of MATLAB
<i>handles</i>	structure with handles and user data (see GUIDATA)

5.27.1.11 updatePresenceDetectorDisplay()

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

5.27.1.12 updateUI()

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

This function update all graphicals element who can change.

With the input called 'out', this function launch all the functions who update a specific graphical element.

Parameters

<i>handles</i>	Structure with handles and user data (see GUIDATA)
<i>out</i>	Cell who contain all informations needed from the wrapper for update the graphical interface.

5.27.1.13 `updateUIActiveCammand()`

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

Update visibility of control panel, connection and step button.

This function show or hide the control's panels and the connection's buttons according whit who will move. It also show step button if a command is connected.

Example : if is pacman time to move and command is not connected, this function hide walls and step element and show pacman one's.

Parameters

<i>handles</i>	structure with handles and user data (see GUIDATA)
----------------	--

5.27.1.14 `updateUIButton()`

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

Show the needed moving buttons.

This function show the direction's buttons allows by the output informations of modelLaby and hide the others one.

Parameters

<i>handles</i>	structure with handles and user data (see GUIDATA)
----------------	--

5.27.1.15 `updateUIEscape()`

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

5.27.1.16 updateUIPlayer()

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

Update graphical place of a player (only pacman in this case).

This function, with the actual position (present in the handles) and the new one as a input, move object.

The dynamics of movement is defined by this fonction $out(t) = \frac{\frac{om+1}{om*e^{cv*t}+1}-1}{om}$ for $t \in [0, 1]$, $om = 72.89105$ and $cv = -11.27357$.

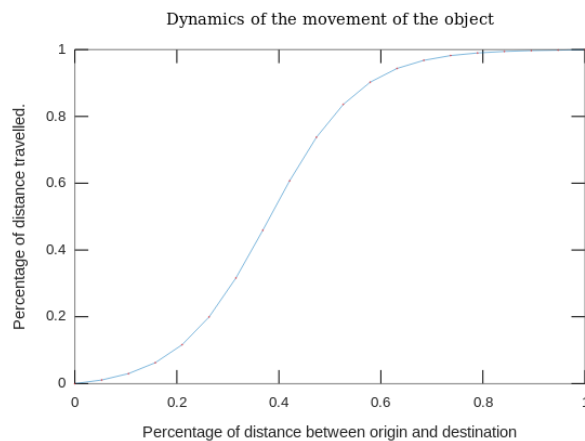


Figure 5.3 Dynamics of movement

Parameters

<i>strPlayer</i>	String contain the exact name of the object to move.
<i>position</i>	new position of the object. format : [x y]
<i>handles</i>	structure with handles and user data (see GUIDATA)

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, 15
- AjoutTransitionStable
 - Automaton, 12
- AutomateAccessible
 - Automaton, 12
- AutomateAccessibleIncertain
 - Automaton, 12
- AutomateGraph, 7
 - accessibilityAutomate, 8
 - adaptTourLangage, 8
 - addWord2Langage, 8
 - AutomateGraph, 8
 - export2DESUMA, 8
 - FSM2Automata, 9
 - langage, 10
 - matrices2vector, 9
 - matrixTrans, 10
 - PathRecherche, 9
 - state, 10
 - structAutomata2vectorAutomata, 9
 - transition, 10
 - vector, 10
 - vector2matrices, 9
 - vector2structAutomata, 9
- Automaton, 11
 - AjoutTransitionStable, 12
 - AutomateAccessible, 12
 - AutomateAccessibleIncertain, 12
 - Automaton, 11
 - CreationFonctionTransitionEnsembleDesParties, 12
 - Ensemble2Etats, 12
 - EnsemblesContenantEtat, 12
 - Etats2Ensemble, 13
 - Evolution, 13
 - Fct, 13
 - MatricesTransition, 13
 - NomsEvenements, 13
 - PathRecherche, 13
- automaton/AutomateGraph.m, 35
- automaton/ParrallelComposition.m, 41
- automaton/mainLaby.m, 35
- automaton/modelGenerator/AutomatonScheduling↔
 - Creation.m, 35
- automaton/modelGenerator/AutomatonStrutureLaby↔
 - Creation.m, 35
- automaton/modelGenerator/AutomatonWallsContraints↔
 - Creation.m, 36
- automaton/modelGenerator/Plan_desumaFunctions.m, 37
- automaton/modelGenerator/SaveDESUMAFFile.m, 38
- automaton/modelGenerator/generer_lab.m, 36
- automaton/modelGenerator/modelGenerator.m, 37
- automaton/modelGenerator/writeStates.m, 38
- automaton/modelGenerator/writeTransitions.m, 39
- automaton/optimalCommand/ParcourirMatrices↔
 - Transitions.m, 41
- automaton/optimalCommand/creationMatricetransition.↔
 - m, 39
- automaton/optimalCommand/getStateTransitionFSM.m, 39
- automaton/optimalCommand/getStateTransitionTXT.m, 40
- automaton/optimalCommand/main.m, 40
- automaton/optimalCommand/optimalCommand.m, 40
- automaton/optimalCommand/rafineAutomaton.m, 41
- automaton/optimalCommand/rafineAutomatonClass.m, 41
- automaton_nd/Automaton.m, 42
- automaton_nd/Labyrinthe.m, 42
- automaton_nd/modi_main.m, 42
- AutomatonSchedulingCreation.m
 - function, 35
- AutomatonStrutureLabyCreation
 - AutomatonStrutureLabyCreation.m, 36
 - Plan_desumaFunctions.m, 37
- AutomatonStrutureLabyCreation.m
 - AutomatonStrutureLabyCreation, 36
- AutomatonWallsContraintsCreation
 - AutomatonWallsContraintsCreation.m, 36
- AutomatonWallsContraintsCreation.m
 - AutomatonWallsContraintsCreation, 36
- commandPacman
 - Wrapper, 33
- commandWalls
 - Wrapper, 33
- connect_Callback
 - figure_Laby.m, 44
- CreatePituresAndVideo
 - CreatePituresAndVideo.m, 42
- CreatePituresAndVideo.m, 42

- CreatePituresAndVideo, 42
- CreatePituresAndVideo_textured
 - CreatePituresAndVideo_textured.m, 43
- CreatePituresAndVideo_textured.m, 43
 - CreatePituresAndVideo_textured, 43
- createUIEscape
 - figure_Laby.m, 44
- createUIPacman
 - figure_Laby.m, 46
- createUIWalls
 - figure_Laby.m, 46
- CreationFonctionTransitionEnsembleDesParties
 - Automaton, 12
- creationMatricetransition
 - creationMatricetransition.m, 39
- creationMatricetransition.m
 - creationMatricetransition, 39
- Ensemble2Etats
 - Automaton, 12
- EnsemblesContenantEtat
 - Automaton, 12
- Etats2Ensemble
 - Automaton, 13
- Etats_Finaux
 - Labyrinthe, 16
- Etats_Initiaux
 - Labyrinthe, 16
- Evolution
 - Automaton, 13
- export2DESUMA
 - AutomateGraph, 8
- f
 - ModelLaby, 18
 - ModelPacman, 21
 - ModelSED, 23
 - ModelWalls, 26
 - StopCondition, 29
- FSM2Automata
 - AutomateGraph, 9
- Fct
 - Automaton, 13
- figure_Laby
 - figure_Laby.m, 46
- figure_Laby.m, 43
 - connect_Callback, 44
 - createUIEscape, 44
 - createUIPacman, 46
 - createUIWalls, 46
 - figure_Laby, 46
 - figure_Laby_OpeningFcn, 47
 - figure_Laby_OutputFcn, 47
 - isOne, 48
 - resetUIConnection, 48
 - ui_Callback, 48
 - updatePresenceDetectorDisplay, 49
 - updateUIActiveCammand, 50
 - updateUIButton, 50
 - updateUIEscape, 50
 - updateUIPlayer, 50
 - updateUIWalls, 51
 - updateUIWallsAround, 51
 - updateUI, 49
- figure_Laby_OpeningFcn
 - figure_Laby.m, 47
- figure_Laby_OutputFcn
 - figure_Laby.m, 47
- function
 - AutomatonSchedulingCreation.m, 35
 - Plan_desumaFunctions.m, 37
- g
 - ModelLaby, 19
 - ModelPacman, 21
 - ModelSED, 24
 - ModelWalls, 26
 - StopCondition, 29
- generer_lab
 - generer_lab.m, 36
- generer_lab.m
 - generer_lab, 36
- get_out
 - Wrapper, 32
- get_stop
 - Wrapper, 32
- getStateTransitionFSM.m
 - getStateTransitionFSM, 39
- getStateTransitionFSM
 - getStateTransitionFSM.m, 39
- getStateTransitionTXT.m
 - getStateTransitionTXT, 40
- getStateTransitionTXT
 - getStateTransitionTXT.m, 40
- handle, 14
- i
 - ModelWalls, 27
- in
 - Wrapper, 33
- incertain
 - Labyrinthe, 15
- init
 - Wrapper, 32
- initialState
 - ModelLaby, 19
 - ModelPacman, 22
 - ModelSED, 24
 - ModelWalls, 27
 - StopCondition, 30
- isOne
 - figure_Laby.m, 48
- jusqu_au_mur
 - Labyrinthe, 15
- Labyrinthe, 14

- affichage, 15
- Etats_Finaux, 16
- Etats_Initiaux, 16
- incertain, 15
- jusqu_au_mur, 15
- Labyrinthe, 15
- MursHorizontaux, 16
- MursVerticaux, 16
- Pas_a_pas, 16
- langage
 - AutomateGraph, 10
- m
 - ModelLaby, 19
 - ModelPacman, 22
 - ModelSED, 24
 - ModelWalls, 27
 - StopCondition, 30
- main.m, 40
- matrices2vector
 - AutomateGraph, 9
- MatricesTransition
 - Automaton, 13
- matrixTrans
 - AutomateGraph, 10
- ModelLaby, 17
 - f, 18
 - g, 19
 - initialState, 19
 - m, 19
 - ModelLaby, 18
 - presentState, 19
- modelLaby
 - Wrapper, 33
- ModelLaby.m, 52
- ModelPacman, 20
 - f, 21
 - g, 21
 - initialState, 22
 - m, 22
 - ModelPacman, 21
 - presentState, 22
- ModelPacman.m, 52
- ModelSED.m, 52
- ModelSED, 23
 - f, 23
 - g, 24
 - initialState, 24
 - m, 24
 - presentState, 24
- ModelWalls, 25
 - f, 26
 - g, 26
 - i, 27
 - initialState, 27
 - m, 27
 - ModelWalls, 26
 - presentState, 27
 - val, 27
- ModelWalls.m, 52
- MursHorizontaux
 - Labyrinthe, 16
- MursVerticaux
 - Labyrinthe, 16
- NomsEvenements
 - Automaton, 13
- optimalCommand
 - optimalCommand.m, 40
- optimalCommand.m
 - optimalCommand, 40
- orderer
 - Wrapper, 32
- out
 - Wrapper, 33
- pacmanBit
 - Wrapper, 33
- ParcourirMatricesTransitions
 - ParcourirMatricesTransitions.m, 41
- ParcourirMatricesTransitions.m
 - ParcourirMatricesTransitions, 41
- ParrallelComposition
 - ParrallelComposition.m, 42
- ParrallelComposition.m
 - ParrallelComposition, 42
- Pas_a_pas
 - Labyrinthe, 16
- PathResearche
 - AutomateGraph, 9
 - Automaton, 13
- Plan_desumaFunctions.m
 - AutomatonStrutureLabyCreation, 37
 - function, 37
 - SaveDESUMAFFile, 37
 - writeStates, 37
 - writeTransitions, 37
- presentState
 - ModelLaby, 19
 - ModelPacman, 22
 - ModelSED, 24
 - ModelWalls, 27
 - StopCondition, 30
- rafineAutomatonClass
 - rafineAutomatonClass.m, 41
- rafineAutomatonClass.m
 - rafineAutomatonClass, 41
- resetUIConnection
 - figure_Laby.m, 48
- SaveDESUMAFFile
 - Plan_desumaFunctions.m, 37
 - SaveDESUMAFFile.m, 38
- SaveDESUMAFFile.m
 - SaveDESUMAFFile, 38
- setColor

- setColor.m, 52
- setColor.m, 52
 - setColor, 52
- Simulation.m, 53
- state
 - AutomateGraph, 10
- stop
 - Wrapper, 33
- StopCondition, 28
 - f, 29
 - g, 29
 - initialState, 30
 - m, 30
 - presentState, 30
 - StopCondition, 29
- stopCondition
 - Wrapper, 33
- StopCondition.m, 53
- structAutomata2vectorAutomata
 - AutomateGraph, 9
- THEplan.m, 53
- testUI.m, 53
- transition
 - AutomateGraph, 10
- ui_Callback
 - figure_Laby.m, 48
- updateConnexion
 - Wrapper, 32
- updatePresenceDetectorDisplay
 - figure_Laby.m, 49
- updateUIActiveCammand
 - figure_Laby.m, 50
- updateUIButton
 - figure_Laby.m, 50
- updateUIEscape
 - figure_Laby.m, 50
- updateUIPlayer
 - figure_Laby.m, 50
- updateUIWalls
 - figure_Laby.m, 51
- updateUIWallsAround
 - figure_Laby.m, 51
- updateUI
 - figure_Laby.m, 49
- val
 - ModelWalls, 27
- vector
 - AutomateGraph, 10
- vector2matrices
 - AutomateGraph, 9
- vector2structAutomata
 - AutomateGraph, 9
- visupacman.m, 53
- visupacman2.m, 53
- wallsBit
 - Wrapper, 34
- wallsBorder
 - wallsBorder.m, 53
- wallsBorder.m, 53
 - wallsBorder, 53
- whoPlay
 - Wrapper, 34
- Wrapper, 31
 - commandPacman, 33
 - commandWalls, 33
 - get_out, 32
 - get_stop, 32
 - in, 33
 - init, 32
 - modellaby, 33
 - orderer, 32
 - out, 33
 - pacmanBit, 33
 - stop, 33
 - stopCondition, 33
 - updateConnexion, 32
 - wallsBit, 34
 - whoPlay, 34
 - Wrapper, 31
- Wrapper.m, 53
- writeStates
 - Plan_desumaFunctions.m, 37
 - writeStates.m, 38
- writeStates.m
 - writeStates, 38
- writeTransitions
 - Plan_desumaFunctions.m, 37
 - writeTransitions.m, 39
- writeTransitions.m
 - writeTransitions, 39