

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 | 19 |
| 4.5.3.1 | f() | 19 |
| 4.5.3.2 | g() | 19 |
| 4.5.3.3 | m() | 20 |
| 4.5.4 | Member Data Documentation | 20 |
| 4.5.4.1 | initialState | 20 |
| 4.5.4.2 | presentState | 20 |
| 4.6 | ModelPacman Class Reference | 21 |
| 4.6.1 | Detailed Description | 22 |
| 4.6.2 | Constructor & Destructor Documentation | 22 |
| 4.6.2.1 | ModelPacman() | 22 |
| 4.6.3 | Member Function Documentation | 23 |
| 4.6.3.1 | f() | 23 |
| 4.6.3.2 | g() | 23 |
| 4.6.3.3 | m() | 24 |
| 4.6.4 | Member Data Documentation | 24 |
| 4.6.4.1 | initialState | 24 |
| 4.6.4.2 | presentState | 24 |
| 4.7 | ModelSED Class Reference | 25 |
| 4.7.1 | Detailed Description | 26 |

| | | |
|---------|--|----|
| 4.7.2 | Member Function Documentation | 26 |
| 4.7.2.1 | f() | 26 |
| 4.7.2.2 | g() | 26 |
| 4.7.2.3 | m() | 27 |
| 4.7.3 | Member Data Documentation | 27 |
| 4.7.3.1 | initialState | 27 |
| 4.7.3.2 | presentState | 27 |
| 4.8 | ModelWalls Class Reference | 28 |
| 4.8.1 | Detailed Description | 29 |
| 4.8.2 | Constructor & Destructor Documentation | 29 |
| 4.8.2.1 | ModelWalls() | 29 |
| 4.8.3 | Member Function Documentation | 30 |
| 4.8.3.1 | f() [1/2] | 30 |
| 4.8.3.2 | f() [2/2] | 30 |
| 4.8.3.3 | g() | 30 |
| 4.8.3.4 | m() | 31 |
| 4.8.4 | Member Data Documentation | 31 |
| 4.8.4.1 | i | 31 |
| 4.8.4.2 | initialState | 31 |
| 4.8.4.3 | presentState | 31 |
| 4.8.4.4 | val | 32 |
| 4.9 | StopCondition Class Reference | 32 |
| 4.9.1 | Detailed Description | 33 |
| 4.9.2 | Constructor & Destructor Documentation | 33 |
| 4.9.2.1 | StopCondition() | 33 |
| 4.9.3 | Member Function Documentation | 34 |
| 4.9.3.1 | f() [1/2] | 34 |
| 4.9.3.2 | f() [2/2] | 34 |
| 4.9.3.3 | g() | 35 |
| 4.9.3.4 | m() | 35 |

| | | |
|-----------|--|----|
| 4.9.4 | Member Data Documentation | 36 |
| 4.9.4.1 | initialState | 36 |
| 4.9.4.2 | presentState | 36 |
| 4.10 | Wrapper Class Reference | 36 |
| 4.10.1 | Detailed Description | 37 |
| 4.10.2 | Constructor & Destructor Documentation | 37 |
| 4.10.2.1 | Wrapper() | 38 |
| 4.10.3 | Member Function Documentation | 38 |
| 4.10.3.1 | get_out() | 38 |
| 4.10.3.2 | get_stop() | 38 |
| 4.10.3.3 | init() | 39 |
| 4.10.3.4 | orderer() | 39 |
| 4.10.3.5 | updateConnexion() | 40 |
| 4.10.4 | Member Data Documentation | 40 |
| 4.10.4.1 | commandPacman | 40 |
| 4.10.4.2 | commandWalls | 40 |
| 4.10.4.3 | in | 40 |
| 4.10.4.4 | modellLaby | 41 |
| 4.10.4.5 | out | 41 |
| 4.10.4.6 | pacmanBit | 41 |
| 4.10.4.7 | stop | 41 |
| 4.10.4.8 | stopCondition | 41 |
| 4.10.4.9 | wallsBit | 41 |
| 4.10.4.10 | whoPlay | 41 |

| | |
|---|-----------|
| 5 File Documentation | 43 |
| 5.1 automaton/AutomateGraph.m File Reference | 43 |
| 5.2 automaton/mainLaby.m File Reference | 43 |
| 5.3 automaton/modelGenerator/AutomatonSchedulingCreation.m File Reference | 43 |
| 5.3.1 Function Documentation | 43 |
| 5.3.1.1 function() | 43 |
| 5.4 automaton/modelGenerator/AutomatonStrutureLabyCreation.m File Reference | 43 |
| 5.4.1 Function Documentation | 44 |
| 5.4.1.1 AutomatonStrutureLabyCreation() | 44 |
| 5.5 automaton/modelGenerator/AutomatonWallsConstraintsCreation.m File Reference | 44 |
| 5.5.1 Function Documentation | 44 |
| 5.5.1.1 AutomatonWallsConstraintsCreation() | 44 |
| 5.6 automaton/modelGenerator/generer_lab.m File Reference | 44 |
| 5.6.1 Function Documentation | 44 |
| 5.6.1.1 generer_lab() | 44 |
| 5.7 automaton/modelGenerator/modelGenerator.m File Reference | 45 |
| 5.8 automaton/modelGenerator/Plan_desumaFunctions.m File Reference | 45 |
| 5.8.1 Function Documentation | 45 |
| 5.8.1.1 AutomatonStrutureLabyCreation() | 45 |
| 5.8.1.2 function() | 45 |
| 5.8.1.3 SaveDESUMAFile() | 45 |
| 5.8.1.4 writeStates() | 45 |
| 5.8.1.5 writeTransitions() | 46 |
| 5.9 automaton/modelGenerator/SaveDESUMAFile.m File Reference | 46 |
| 5.9.1 Function Documentation | 46 |
| 5.9.1.1 SaveDESUMAFile() | 46 |
| 5.10 automaton/modelGenerator/writeStates.m File Reference | 46 |
| 5.10.1 Function Documentation | 46 |
| 5.10.1.1 writeStates() | 46 |
| 5.11 automaton/modelGenerator/writeTransitions.m File Reference | 47 |

| | | |
|----------|--|----|
| 5.11.1 | Function Documentation | 47 |
| 5.11.1.1 | writeTransitions() | 47 |
| 5.12 | automaton/optimalCommand/creationMatricetransition.m File Reference | 47 |
| 5.12.1 | Function Documentation | 47 |
| 5.12.1.1 | creationMatricetransition() | 47 |
| 5.13 | automaton/optimalCommand/getStateTransitionFSM.m File Reference | 47 |
| 5.13.1 | Function Documentation | 47 |
| 5.13.1.1 | getStateTransitionFSM() | 48 |
| 5.14 | automaton/optimalCommand/getStateTransitionTXT.m File Reference | 48 |
| 5.14.1 | Function Documentation | 48 |
| 5.14.1.1 | getStateTransitionTXT() | 48 |
| 5.15 | automaton/optimalCommand/main.m File Reference | 48 |
| 5.16 | main.m File Reference | 48 |
| 5.17 | automaton/optimalCommand/optimalCommand.m File Reference | 48 |
| 5.17.1 | Function Documentation | 48 |
| 5.17.1.1 | optimalCommand() | 48 |
| 5.18 | automaton/optimalCommand/ParcourirMatricesTransitions.m File Reference | 49 |
| 5.18.1 | Function Documentation | 49 |
| 5.18.1.1 | ParcourirMatricesTransitions() | 49 |
| 5.19 | automaton/optimalCommand/rafineAutomaton.m File Reference | 49 |
| 5.20 | automaton/optimalCommand/rafineAutomatonClass.m File Reference | 49 |
| 5.20.1 | Function Documentation | 49 |
| 5.20.1.1 | rafineAutomatonClass() | 49 |
| 5.21 | automaton/ParrallelComposition.m File Reference | 49 |
| 5.21.1 | Function Documentation | 50 |
| 5.21.1.1 | ParrallelComposition() | 50 |
| 5.22 | automaton_nd/Automaton.m File Reference | 50 |
| 5.23 | automaton_nd/Labyrinthe.m File Reference | 50 |
| 5.24 | automaton_nd/modi_main.m File Reference | 50 |
| 5.25 | CreatePituresAndVideo.m File Reference | 50 |

| | | |
|-----------|---|----|
| 5.25.1 | Function Documentation | 50 |
| 5.25.1.1 | CreatePituresAndVideo() | 50 |
| 5.26 | CreatePituresAndVideo_textured.m File Reference | 51 |
| 5.26.1 | Function Documentation | 51 |
| 5.26.1.1 | CreatePituresAndVideo_textured() | 51 |
| 5.27 | figure_Laby.m File Reference | 51 |
| 5.27.1 | Function Documentation | 52 |
| 5.27.1.1 | connect_Callback() | 52 |
| 5.27.1.2 | createUIEscape() | 52 |
| 5.27.1.3 | createUIPacman() | 54 |
| 5.27.1.4 | createUIWalls() | 54 |
| 5.27.1.5 | figure_Laby() | 55 |
| 5.27.1.6 | figure_Laby_OpeningFcn() | 55 |
| 5.27.1.7 | figure_Laby_OutputFcn() | 55 |
| 5.27.1.8 | isOne() | 56 |
| 5.27.1.9 | resetUIConnection() | 56 |
| 5.27.1.10 | ui_Callback() | 56 |
| 5.27.1.11 | updatePresenceDetectorDisplay() | 57 |
| 5.27.1.12 | updateUI() | 57 |
| 5.27.1.13 | updateUIActiveCammand() | 58 |
| 5.27.1.14 | updateUIButton() | 58 |
| 5.27.1.15 | updateUIEscape() | 58 |
| 5.27.1.16 | updateUIPlayer() | 59 |
| 5.27.1.17 | updateUIWalls() | 59 |
| 5.27.1.18 | updateUIWallsAround() | 59 |
| 5.28 | ModelLaby.m File Reference | 60 |
| 5.29 | ModelPacman.m File Reference | 60 |
| 5.30 | ModelSED.m File Reference | 60 |
| 5.31 | ModelWalls.m File Reference | 60 |
| 5.31.1 | Detailed Description | 60 |
| 5.32 | setColor.m File Reference | 60 |
| 5.32.1 | Function Documentation | 61 |
| 5.32.1.1 | setColor() | 61 |
| 5.33 | Simulation.m File Reference | 61 |
| 5.34 | StopCondition.m File Reference | 61 |
| 5.35 | visupacman.m File Reference | 61 |
| 5.36 | visupacman2.m File Reference | 61 |
| 5.37 | wallsBorder.m File Reference | 61 |
| 5.37.1 | Function Documentation | 61 |
| 5.37.1.1 | wallsBorder() | 61 |
| 5.38 | Wrapper.m File Reference | 61 |

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 | 25 |
| ModelLaby | 17 |
| ModelPacman | 21 |
| ModelWalls | 28 |
| StopCondition | 32 |
| Labyrinthe | 14 |
| Wrapper | 36 |

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 |
| ModelLaby | |
| Class which contains the "fmg" structure of the labyrinth for 1 player | 17 |
| ModelPacman | |
| Contain ghost Pacman control | 21 |
| ModelSED | |
| Abstract Class who contain the structure of a "fmg" implementation | 25 |
| ModelWalls | |
| Input : No need | |
| Output : [UPwalls , RIGHTwalls] | |
| State : contain the last move (0 = up ; 1 = right) | |
| This command do the sequence walls Right -> walls down | |
| 28 | |
| StopCondition | |
| Class used to manage shutdown conditions | 32 |
| Wrapper | |
| WRAPPER Global organization of the project | 36 |

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

| | |
|---|----|
| CreatePituresAndVideo.m | 50 |
| CreatePituresAndVideo_textured.m | 51 |
| figure_Laby.m | 51 |
| main.m | 48 |
| ModelLaby.m | 60 |
| ModelPacman.m | 60 |
| ModelSED.m | 60 |
| ModelWalls.m | 60 |
| Contain wall movement command | 60 |
| setColor.m | 60 |
| Simulation.m | 61 |
| StopCondition.m | 61 |
| visupacman.m | 61 |
| visupacman2.m | 61 |
| wallsBorder.m | 61 |
| Wrapper.m | 61 |
| automaton/AutomateGraph.m | 43 |
| automaton/mainLaby.m | 43 |
| automaton/ParrallelComposition.m | 49 |
| automaton/modelGenerator/AutomatonSchedulingCreation.m | 43 |
| automaton/modelGenerator/AutomatonStrutureLabyCreation.m | 43 |
| automaton/modelGenerator/AutomatonWallsContraintsCreation.m | 44 |
| automaton/modelGenerator/generer_lab.m | 44 |
| automaton/modelGenerator/modelGenerator.m | 45 |
| automaton/modelGenerator/Plan_desumaFunctions.m | 45 |
| automaton/modelGenerator/SaveDESUMAFFile.m | 46 |
| automaton/modelGenerator/writeStates.m | 46 |
| automaton/modelGenerator/writeTransitions.m | 47 |
| automaton/optimalCommand/creationMatricetransition.m | 47 |
| automaton/optimalCommand/getStateTransitionFSM.m | 47 |
| automaton/optimalCommand/getStateTransitionTXT.m | 48 |
| automaton/optimalCommand/main.m | 48 |
| automaton/optimalCommand/optimalCommand.m | 48 |
| automaton/optimalCommand/ParcourirMatricesTransitions.m | 49 |
| automaton/optimalCommand/rafineAutomaton.m | 49 |

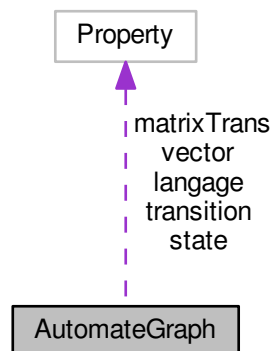
| | |
|--|----|
| automaton/optimalCommand/ rafineAutomatonClass.m | 49 |
| automaton_nd/ Automaton.m | 50 |
| automaton_nd/ Labyrinthe.m | 50 |
| automaton_nd/ modi_main.m | 50 |

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

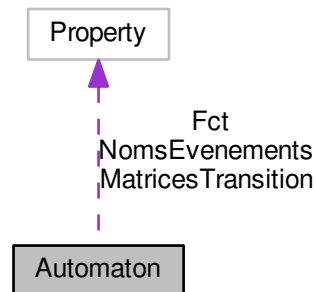
Property vector

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

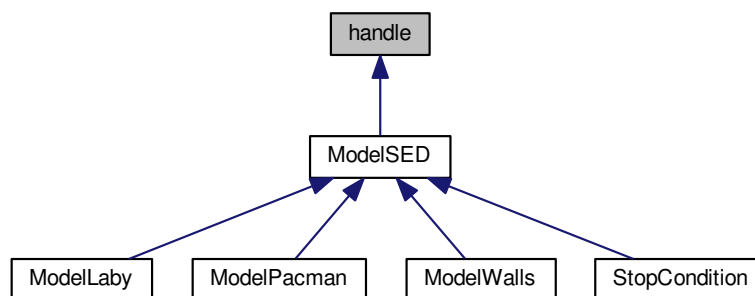
Property NomsEvenements

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

- automaton_nd/[Automaton.m](#)

4.3 handle Class Reference

Inheritance diagram for handle:

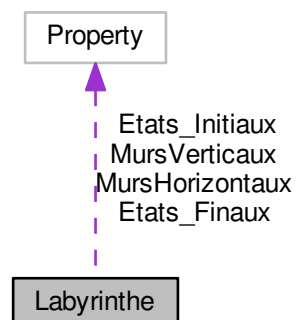


The documentation for this class was generated from the following file:

- [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

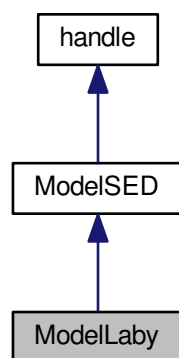
The documentation for this class was generated from the following file:

- automaton_nd/[Labyrinthe.m](#)

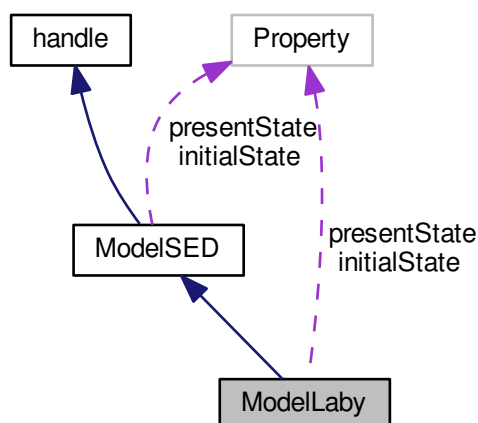
4.5 ModelLaby Class Reference

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 Instance of [ModelLaby](#) Class.
- [function f](#) (in obj, in in)

- *Compute the evolution of the model.*
- **function m** (in obj, in nextState, in init)
Memory method.
- **function g** (in obj)
Create the outputs in a 1x9 cell-array.

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.

You can change here labyrinth's dynamic : how objects and walls are evolving in the labyrinth, not the command of then.

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 Instance of **ModelLaby** Class.

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]
```

Compute the evolution of the model.

Parameters

| | |
|------------|---------------------------------|
| <i>obj</i> | The instance which will evolve. |
| <i>in</i> | Input needed for the computing. |

Returns

Next instance of the [ModelLaby](#) class.

Reimplemented from [ModelSED](#).

4.5.3.2 g()

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

Create the outputs in a 1x9 cell-array.

Parameters

| | |
|------------|-------------------------------------|
| <i>obj</i> | the concerned instance of the class |
|------------|-------------------------------------|

Return values

| | |
|------------|--|
| <i>out</i> | Constructed output 1x9 cell-array of the model |
|------------|--|

Reimplemented from [ModelSED](#).

4.5.3.3 m()

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

Memory method.

Update the state of the command.

Parameters

| | |
|------------------|---|
| <i>obj</i> | The selected instance of the class |
| <i>nextState</i> | The value of the state need to update |
| <i>init</i> | Boolean condition for initialize or reset the command |

Returns

instance of the class updated

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.

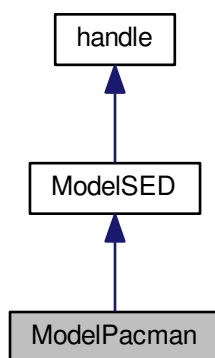
The documentation for this class was generated from the following file:

- [ModelLaby.m](#)

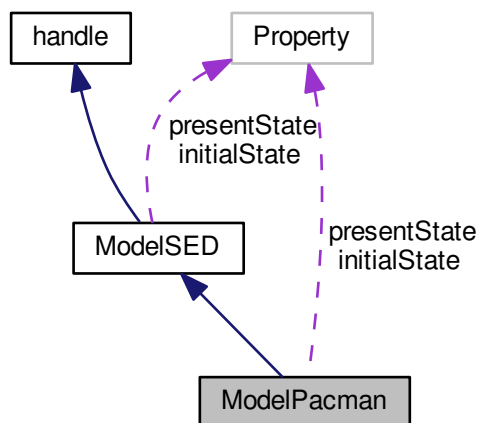
4.6 ModelPacman Class Reference

Contain ghost Pacman control.

Inheritance diagram for ModelPacman:



Collaboration diagram for ModelPacman:



Public Member Functions

- [function ModelPacman](#) (in initialValue)
Class constructor.
- [function f](#) (in obj, in in)

Compute the evolution of the command.

- `function m` (in `obj`, in `nextState`, in `init`)

Memory method update the state of the command.

- `function g` (in `obj`)

Create the outputs.

Public Attributes

- Property `presentState`

This is the state of the command in the present moment.

- Property `initialState`

This is the state of the command in the initialization and when it's reseted.

4.6.1 Detailed Description

Contain ghost Pacman control.

You can change here Pacman's command.

Input : Possible Pacman's moves [Up Down Left Right]

0 = move not possible ; 1 = move possible

(`Wout{7}`)

Output : Pacman's moves 1 : `pacmanLeftBut`, (`Wout(3)`)

2 : `pacmanUpBut`, (`Wout(1)`)

3 : `pacmanRightBut`, (`Wout(4)`)

4 : `pacmanDownBut`, (`Wout(2)`)

(`Win(4:7)` of `wrapper`)

Input : Walls around Pacman

1 up

2 down

3 left

4 right

This command do the sequence $P(D) > P(B) > P(H) > P(G)$

4.6.2 Constructor & Destructor Documentation

4.6.2.1 `ModelPacman()`

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

Class constructor.

Parameters

| | |
|---------------------------|---------------------------|
| <code>initialValue</code> | Contain the initial state |
|---------------------------|---------------------------|

Returns

instance of the [ModelPacman](#) class.

4.6.3 Member Function Documentation

4.6.3.1 f()

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

Compute the evolution of the command.

Parameters

| | |
|------------|------------------------------|
| <i>obj</i> | The instance who evolute |
| <i>in</i> | Input needed for the compute |

Return values

| | |
|------------------|--|
| <i>nextState</i> | The future state of the Pacman command |
|------------------|--|

Reimplemented from [ModelSED](#).

4.6.3.2 g()

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

Create the outputs.

Parameters

| | |
|------------|-------------------------------------|
| <i>obj</i> | the concerned instance of the class |
|------------|-------------------------------------|

Return values

| | |
|------------|--------------------------------|
| <i>out</i> | The output who is the command. |
|------------|--------------------------------|

Reimplemented from [ModelSED](#).

4.6.3.3 m()

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

Memory method update the state of the command.

Parameters

| | |
|------------------|---|
| <i>obj</i> | The selected instance of the class |
| <i>nextState</i> | The value of the state need to update |
| <i>init</i> | Boolean condition for initialize or reset the command |

Returns

instance of the class updated

Reimplemented from [ModelSED](#).

4.6.4 Member Data Documentation

4.6.4.1 initialState

Property `initialState`

This is the state of the command in the initialization and when it's reseted.

4.6.4.2 presentState

Property `presentState`

This is the state of the command in the present moment.

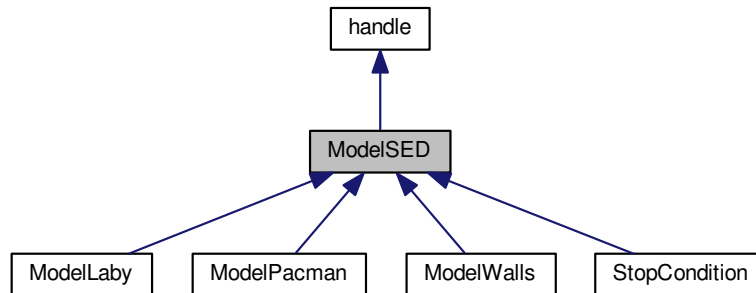
The documentation for this class was generated from the following file:

- [ModelPacman.m](#)

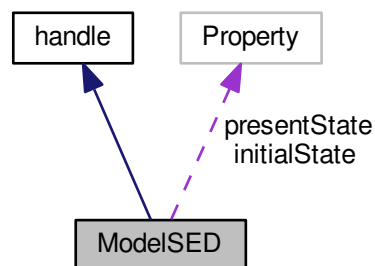
4.7 ModelSED Class Reference

Abstract Class who contain the structure of a "fmg" implementation.

Inheritance diagram for ModelSED:



Collaboration diagram for ModelSED:



Public Member Functions

- virtual **f** (in obj, in in)
Compute the evolution of the model.
- virtual **m** (in obj, in nextState, in init)
Memory method update the state of the command.
- virtual **g** (in obj)
Create the outputs.

Public Attributes

- Property **presentState**
This is the state of the command in the present moment.
- Property **initialState**
This is the state of the command in the initialization and when it's reseted.

4.7.1 Detailed Description

Abstract Class who contain the structure of a "fmg" implementation.

This class is used for give a general definition of Model Class.

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

Output : output's action of the model

State : minimal information necessary who evolve

4.7.2 Member Function Documentation

4.7.2.1 f()

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

Compute the evolution of the model.

Parameters

| | |
|------------|--------------------------------|
| <i>obj</i> | The instance who evolve |
| <i>in</i> | Input needed for the computing |

Return values

| | |
|------------------|-------------------------------|
| <i>nextState</i> | The future state of the model |
|------------------|-------------------------------|

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

4.7.2.2 g()

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

Create the outputs.

Parameters

| | |
|------------|-------------------------------------|
| <i>obj</i> | the concerned instance of the class |
|------------|-------------------------------------|

Return values

| | |
|------------|---------------------------------|
| <i>out</i> | Constructed output of the model |
|------------|---------------------------------|

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

4.7.2.3 m()

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

Memory method update the state of the command.

Parameters

| | |
|------------------|---|
| <i>obj</i> | The selected instance of the class |
| <i>nextState</i> | The value of the state need to update |
| <i>init</i> | Boolean condition for initialize or reset the command |

Returns

instance of the class updated

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

4.7.3 Member Data Documentation

4.7.3.1 initialState

Property `initialState`

This is the state of the command in the initialization and when it's reseted.

4.7.3.2 presentState

Property `presentState`

This is the state of the command in the present moment.

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

- [ModelSED.m](#)

4.8 ModelWalls Class Reference

Input : No need

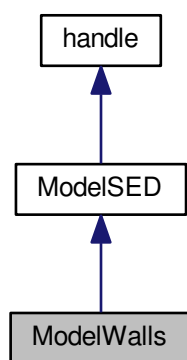
Output : [UPwalls , RIGHTwalls]

State : contain the last move (0 = up ; 1 = right)

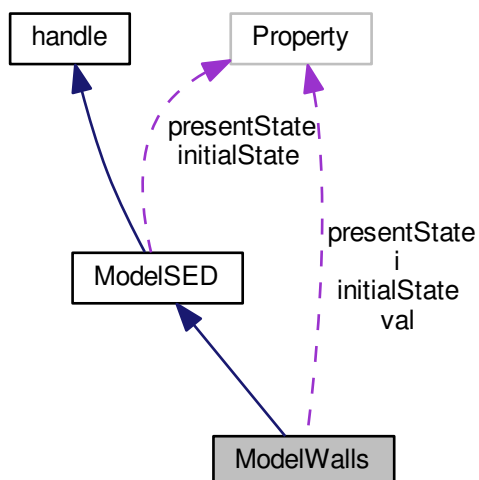
This command do the sequence walls Right -> walls down

.

Inheritance diagram for ModelWalls:



Collaboration diagram for ModelWalls:



Public Member Functions

- `function ModelWalls` (in `initValue`)
Class constructor.
- `function f` (in `obj`)
- `function m` (in `obj`, in `nextState`, in `init`)
Memory method update the state of the command.
- `function g` (in `obj`)
Create the outputs.
- `virtual f` (in `obj`, in `in`)
Compute the evolution of the model.

Public Attributes

- Property `presentState`
- Property `initialState`
- Property `i`
- Property `val`

4.8.1 Detailed Description

Input : No need

Output : [UPwalls , RIGHTwalls]

State : contain the last move (0 = up ; 1 = right)

This command do the sequence walls Right → walls down

.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 ModelWalls()

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

Class constructor.

Parameters

| | |
|------------------|---------------------------|
| <i>initValue</i> | Contain the initial state |
|------------------|---------------------------|

Returns

instance of the `ModelWalls` class.

4.8.3 Member Function Documentation

4.8.3.1 `f()` [1/2]

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

Compute the evolution of the model.

Parameters

| | |
|------------|--------------------------------|
| <i>obj</i> | The instance who evolve |
| <i>in</i> | Input needed for the computing |

Return values

| | |
|------------------|-------------------------------|
| <i>nextState</i> | The future state of the model |
|------------------|-------------------------------|

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

4.8.3.2 `f()` [2/2]

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

4.8.3.3 `g()`

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

Create the outputs.

Parameters

| | |
|------------|-------------------------------------|
| <i>obj</i> | the concerned instance of the class |
|------------|-------------------------------------|

Return values

| | |
|------------|--------------------------------|
| <i>out</i> | The output who is the command. |
|------------|--------------------------------|

Reimplemented from [ModelSED](#).

4.8.3.4 m()

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

Memory method update the state of the command.

Parameters

| | |
|------------------|---|
| <i>obj</i> | The selected instance of the class |
| <i>nextState</i> | The value of the state need to update |
| <i>init</i> | Boolean condition for initialize or reset the command |

Returns

instance of the class updated

Reimplemented from [ModelSED](#).

4.8.4 Member Data Documentation

4.8.4.1 i

Property i

4.8.4.2 initialState

Property initialState

4.8.4.3 presentState

Property presentState

4.8.4.4 val

Property val

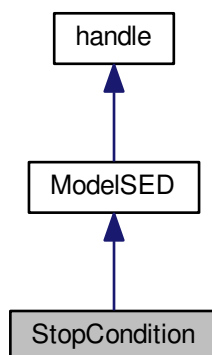
The documentation for this class was generated from the following file:

- [ModelWalls.m](#)

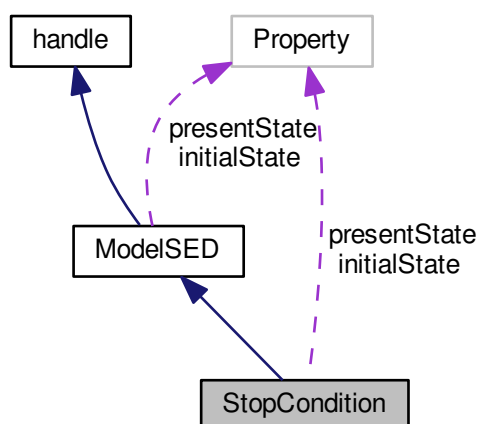
4.9 StopCondition Class Reference

Class used to manage shutdown conditions.

Inheritance diagram for StopCondition:



Collaboration diagram for StopCondition:



Public Member Functions

- [function StopCondition](#) (in `initCondition`)
Class constructor of Instance of [StopCondition](#) Class.
- [function f](#) (in `obj`, in `noEscape`, in `pacmanWallsBreak`)
Compute the evolution of the model.
- [function m](#) (in `obj`, in `nextState`, in `init`)
Memory method.
- [function g](#) (in `obj`)
Create the outputs in a vector with 4 parameters.
- virtual [f](#) (in `obj`, in `in`)
Compute the evolution of the model.

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.

4.9.1 Detailed Description

Class used to manage shutdown conditions.

Labyrinth shutdown conditions model.

You can modify the shutdown conditions here. It is developing in the same way as MODELSED, with "FMG" block. (MODELSED's legacy)

Input : walls of Pacman's
walls of ghost's
escape of Pacman
CaughtBreak

Output : 1 Escape
2 Caught
3 pacmanWallsBreak
4 ghostWallsBreak

4.9.2 Constructor & Destructor Documentation

4.9.2.1 StopCondition()

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

Class constructor of Instance of [StopCondition](#) Class.

Parameters

| | |
|----------------------|--|
| <i>initCondition</i> | Structure for the InitialState. It have to contain : 'escape', 'caught', 'pacman' and 'numberOfPossibleCaught' |
|----------------------|--|

Returns

instance of the [ModelLaby](#) class.

4.9.3 Member Function Documentation

4.9.3.1 `f()` [1/2]

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

Compute the evolution of the model.

Parameters

| | |
|------------|--------------------------------|
| <i>obj</i> | The instance who evolve |
| <i>in</i> | Input needed for the computing |

Return values

| | |
|------------------|-------------------------------|
| <i>nextState</i> | The future state of the model |
|------------------|-------------------------------|

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

4.9.3.2 `f()` [2/2]

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

Compute the evolution of the model.

Parameters

| | |
|------------|---------------------------------|
| <i>obj</i> | The instance which will evolve. |
| <i>in</i> | Input needed for the computing. |

Returns

Next instance of the [StopCondition](#) class.

4.9.3.3 g()

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

Create the outputs in a vector with 4 parameters.

Parameters

| | |
|------------|-------------------------------------|
| <i>obj</i> | the concerned instance of the class |
|------------|-------------------------------------|

Return values

| | |
|------------|--|
| <i>out</i> | Constructed output vector with 4 parameters of the model. In this case, the output contain only escape information and Pacman block information. |
|------------|--|

Reimplemented from [ModelSED](#).

4.9.3.4 m()

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

Memory method.

Update the state of the command.

Parameters

| | |
|------------------|---|
| <i>obj</i> | The selected instance of the class |
| <i>nextState</i> | The value of the state need to update |
| <i>init</i> | Boolean condition for initialize or reset the command |

Returns

instance of the class updated

Reimplemented from [ModelSED](#).

4.9.4 Member Data Documentation

4.9.4.1 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.

4.9.4.2 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.

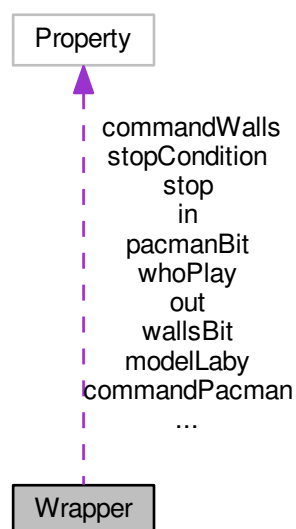
The documentation for this class was generated from the following file:

- [StopCondition.m](#)

4.10 Wrapper Class Reference

WRAPPER Global organization of the project.

Collaboration diagram for Wrapper:



Public Member Functions

- **function** `Wrapper` (`in` inSize, `in` outSize, `in` initLaby, `in` initWalls, `in` initPac, `in` initStop)
Class constructor of Instance of `StopCondition` Class.
- **function** `updateConnexion` (`in` obj, `in` indBit, `in` value)
Update the connection bit for connect automatic mode for Pacman and/or the walls.
- **function** `init` (`in` obj)
Allows to completely reset the labyrinth.
- **function** `orderer` (`in` obj, `in` vectIn)
Ordinate the global execution of Models.
- **function** `get_stop` (`in` obj)
Return the current State of shutdown condition.
- **function** `get_out` (`in` obj)
Return the current State of output cell.

Public Attributes

- Property `wallsBit`
Boolean connection for the walls.
- Property `pacmanBit`
Boolean connection for the Pacman.
- Property `modelLaby`
contain the instance of the model of labyrinth.
- Property `commandWalls`
contain the instance of wall's command.
- Property `commandPacman`
contain the instance of Pacman's command.
- Property `stopCondition`
contain the instance shutdown condition.
- Property `in`
A integer vector who contain the state of input,.
- Property `out`
A cell who contain the state of output,.
- Property `stop`
A cell that contains the output of the stop conditions instance.
- Property `whoPlay`
A increment integer that permit to know which object to play 0 = walls ; 1 = Pacman ;.

4.10.1 Detailed Description

WRAPPER Global organization of the project.

Contain the connection between the different elements and contain all the models (walls, labyrinth, Pacman, escape)

4.10.2 Constructor & Destructor Documentation

4.10.2.1 Wrapper()

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

Class constructor of Instance of [StopCondition](#) Class.

Parameters

| | |
|------------------|---|
| <i>inSize</i> | Integer containing the size of the state of inputs. |
| <i>outSize</i> | Integer containing the size of the state of Outputs. |
| <i>initLaby</i> | Structure containing every fields need to initialize the labyrinth Model. |
| <i>initWalls</i> | Structure containing every fields need to initialize the Walls Model. |
| <i>initPac</i> | Structure containing every fields need to initialize the Pacman Model. |
| <i>initStop</i> | Structure containing every fields need to initialize the Stop Model. |

Returns

instance of the [Wrapper](#) class.

4.10.3 Member Function Documentation

4.10.3.1 get_out()

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

Return the current State of output cell.

Parameters

| | |
|------------|---|
| <i>obj</i> | The instance of Wrapper . |
|------------|---|

Returns

Output cell.

4.10.3.2 get_stop()

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


Return the current State of shutdown condition.

Parameters

| | |
|------------|---|
| <i>obj</i> | The instance of Wrapper . |
|------------|---|

Returns

instance of the Stop class.

4.10.3.3 init()

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

Allows to completely reset the labyrinth.

Parameters

| | |
|------------|---|
| <i>obj</i> | The instance of Wrapper . |
|------------|---|

Returns

instance of the [Wrapper](#) class.

4.10.3.4 orderer()

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

Ordinate the global execution of Models.

Parameters

| | |
|------------|---|
| <i>obj</i> | The instance of Wrapper . |
|------------|---|

Returns

instance of the [Wrapper](#) class. In a first case, it checks which models is connected via the interface. If a model is connected, we execute the 'fmg' structure of the model, else, we are waiting that a player push the desired button to move the labyrinth.

Here a little graphic about the scheduling of the call of each model : murs > Laby > pacman > laby
Every call of 'laby' implies a call of Stop Condition.

4.10.3.5 updateConnexion()

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

Update the connection bit for connect automatic mode for Pacman and/or the walls.

Parameters

| | |
|---------------|--|
| <i>obj</i> | The instance of Wrapper . |
| <i>indBit</i> | Integer pointing the element to be connected : '1' for walls and '3' for Pacman. |
| <i>value</i> | Boolean indicating if the element is connected (True) or not. |

Returns

instance of the [Wrapper](#) class.

4.10.4 Member Data Documentation

4.10.4.1 commandPacman

Property commandPacman

contain the instance of Pacman's command.

4.10.4.2 commandWalls

Property commandWalls

contain the instance of wall's command.

4.10.4.3 in

Property in

A integer vector who contain the state of input,.

incremented by the callback or some action.

4.10.4.4 modelLaby

Property modelLaby

contain the instance of the model of labyrinth.

4.10.4.5 out

Property out

A cell who contain the state of output,.

incremented by the callback or some action.

4.10.4.6 pacmanBit

Property pacmanBit

Boolean connection for the Pacman.

4.10.4.7 stop

Property stop

A cell that contains the output of the stop conditions instance.

incremented by the callback or some action.

4.10.4.8 stopCondition

Property stopCondition

contain the instance shutdown condition.

4.10.4.9 wallsBit

Property wallsBit

Boolean connection for the walls.

4.10.4.10 whoPlay

Property whoPlay

A increment integer that permit to know which object to play 0 = walls ; 1 = Pacman ;.

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`)
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.

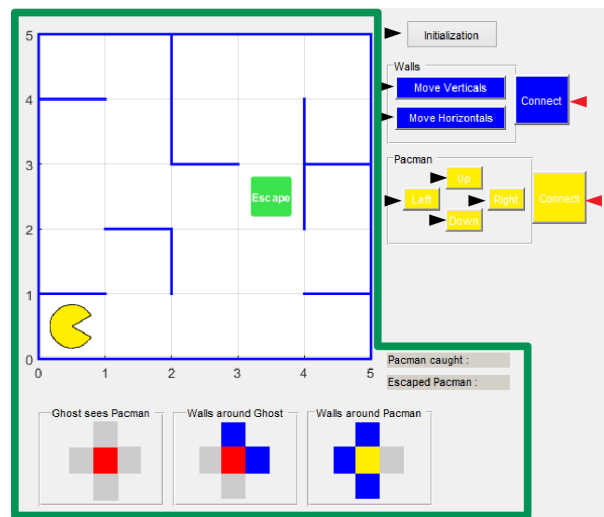


Figure 5.1 button's type of GUI

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.

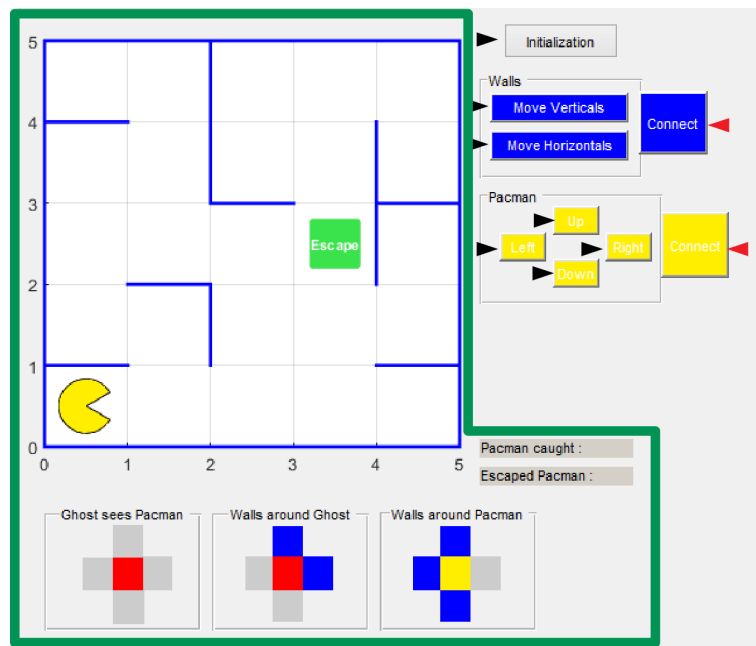


Figure 5.2 button's type of GUI

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 $cv = -11.27357$.

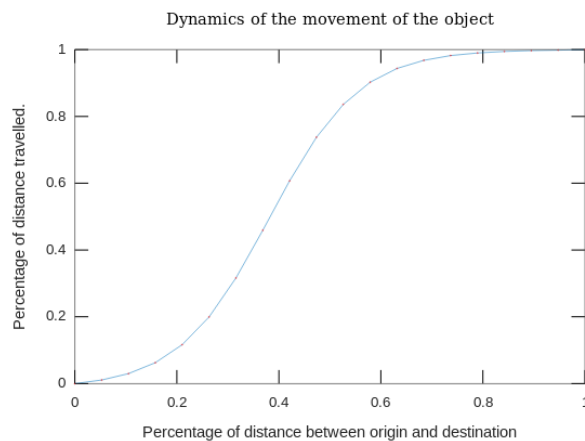


Figure 5.3 Dynamics of movement

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](#)
Contain ghost Pacman control.

5.30 ModelSED.m File Reference

Classes

- class [ModelSED](#)
Abstract Class who contain the structure of a "fmg" implementation.

5.31 ModelWalls.m File Reference

Contain wall movement command.

Classes

- class [ModelWalls](#)
Input : No need

Output : [UPwalls , RIGHTwalls]

State : contain the last move (0 = up ; 1 = right)
This command do the sequence walls Right → walls down
.

5.31.1 Detailed Description

Contain wall movement command.

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](#)
Class used to manage shutdown conditions.

5.35 visupacman.m File Reference

5.36 visupacman2.m File Reference

5.37 wallsBorder.m File Reference

Functions

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

5.37.1 Function Documentation

5.37.1.1 wallsBorder()

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

5.38 Wrapper.m File Reference

Classes

- class [Wrapper](#)
WRAPPER Global organization of the project.

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, 43
- automaton/ParrallelComposition.m, 49
- automaton/mainLaby.m, 43
- automaton/modelGenerator/AutomatonScheduling↔
 - Creation.m, 43
- automaton/modelGenerator/AutomatonStrutureLaby↔
 - Creation.m, 43
- automaton/modelGenerator/AutomatonWallsContraints↔
 - Creation.m, 44
- automaton/modelGenerator/Plan_desumaFunctions.m, 45
- automaton/modelGenerator/SaveDESUMAFFile.m, 46
- automaton/modelGenerator/generer_lab.m, 44
- automaton/modelGenerator/modelGenerator.m, 45
- automaton/modelGenerator/writeStates.m, 46
- automaton/modelGenerator/writeTransitions.m, 47
- automaton/optimalCommand/ParcourirMatrices↔
 - Transitions.m, 49
- automaton/optimalCommand/creationMatricetransition.↔
 - m, 47
- automaton/optimalCommand/getStateTransitionFSM.m, 47
- automaton/optimalCommand/getStateTransitionTXT.m, 48
- automaton/optimalCommand/main.m, 48
- automaton/optimalCommand/optimalCommand.m, 48
- automaton/optimalCommand/rafineAutomaton.m, 49
- automaton/optimalCommand/rafineAutomatonClass.m, 49
- automaton_nd/Automaton.m, 50
- automaton_nd/Labyrinthe.m, 50
- automaton_nd/modi_main.m, 50
- AutomatonSchedulingCreation.m
 - function, 43
- AutomatonStrutureLabyCreation
 - AutomatonStrutureLabyCreation.m, 44
 - Plan_desumaFunctions.m, 45
- AutomatonStrutureLabyCreation.m
 - AutomatonStrutureLabyCreation, 44
- AutomatonWallsContraintsCreation
 - AutomatonWallsContraintsCreation.m, 44
- AutomatonWallsContraintsCreation.m
 - AutomatonWallsContraintsCreation, 44
- commandPacman
 - Wrapper, 40
- commandWalls
 - Wrapper, 40
- connect_Callback
 - figure_Laby.m, 52
- CreatePituresAndVideo
 - CreatePituresAndVideo.m, 50
- CreatePituresAndVideo.m, 50

- CreatePituresAndVideo, 50
- CreatePituresAndVideo_textured
 - CreatePituresAndVideo_textured.m, 51
- CreatePituresAndVideo_textured.m, 51
 - CreatePituresAndVideo_textured, 51
- createUIEscape
 - figure_Laby.m, 52
- createUIPacman
 - figure_Laby.m, 54
- createUIWalls
 - figure_Laby.m, 54
- CreationFonctionTransitionEnsembleDesParties
 - Automaton, 12
- creationMatricetransition
 - creationMatricetransition.m, 47
- creationMatricetransition.m
 - creationMatricetransition, 47
- 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, 19
 - ModelPacman, 23
 - ModelSED, 26
 - ModelWalls, 30
 - StopCondition, 34
- FSM2Automata
 - AutomateGraph, 9
- Fct
 - Automaton, 13
- figure_Laby
 - figure_Laby.m, 54
- figure_Laby.m, 51
 - connect_Callback, 52
 - createUIEscape, 52
 - createUIPacman, 54
 - createUIWalls, 54
 - figure_Laby, 54
 - figure_Laby_OpeningFcn, 55
 - figure_Laby_OutputFcn, 55
 - isOne, 56
 - resetUIConnection, 56
 - ui_Callback, 56
 - updatePresenceDetectorDisplay, 57
 - updateUIActiveCammand, 58
 - updateUIButton, 58
 - updateUIEscape, 58
 - updateUIPlayer, 58
 - updateUIWalls, 59
 - updateUIWallsAround, 59
 - updateUI, 57
- figure_Laby_OpeningFcn
 - figure_Laby.m, 55
- figure_Laby_OutputFcn
 - figure_Laby.m, 55
- function
 - AutomatonSchedulingCreation.m, 43
 - Plan_desumaFunctions.m, 45
- g
 - ModelLaby, 19
 - ModelPacman, 23
 - ModelSED, 26
 - ModelWalls, 30
 - StopCondition, 35
- generer_lab
 - generer_lab.m, 44
- generer_lab.m
 - generer_lab, 44
- get_out
 - Wrapper, 38
- get_stop
 - Wrapper, 38
- getStateTransitionFSM.m
 - getStateTransitionFSM, 47
- getStateTransitionFSM
 - getStateTransitionFSM.m, 47
- getStateTransitionTXT.m
 - getStateTransitionTXT, 48
- getStateTransitionTXT
 - getStateTransitionTXT.m, 48
- handle, 14
- i
 - ModelWalls, 31
- in
 - Wrapper, 40
- incertain
 - Labyrinthe, 15
- init
 - Wrapper, 39
- initialState
 - ModelLaby, 20
 - ModelPacman, 24
 - ModelSED, 27
 - ModelWalls, 31
 - StopCondition, 36
- isOne
 - figure_Laby.m, 56
- 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, 23
 - ModelSED, 27
 - ModelWalls, 31
 - StopCondition, 35
- main.m, 48
- matrices2vector
 - AutomateGraph, 9
- MatricesTransition
 - Automaton, 13
- matrixTrans
 - AutomateGraph, 10
- ModelLaby, 17
 - f, 19
 - g, 19
 - initialState, 20
 - m, 19
 - ModelLaby, 18
 - presentState, 20
- modelLaby
 - Wrapper, 40
- ModelLaby.m, 60
- ModelPacman, 21
 - f, 23
 - g, 23
 - initialState, 24
 - m, 23
 - ModelPacman, 22
 - presentState, 24
- ModelPacman.m, 60
- ModelSED.m, 60
- ModelSED, 25
 - f, 26
 - g, 26
 - initialState, 27
 - m, 27
 - presentState, 27
- ModelWalls, 28
 - f, 30
 - g, 30
 - i, 31
 - initialState, 31
 - m, 31
 - ModelWalls, 29
 - presentState, 31
 - val, 31
- ModelWalls.m, 60
- MursHorizontaux
 - Labyrinthe, 16
- MursVerticaux
 - Labyrinthe, 16
- NomsEvenements
 - Automaton, 13
- optimalCommand
 - optimalCommand.m, 48
- optimalCommand.m
 - optimalCommand, 48
- orderer
 - Wrapper, 39
- out
 - Wrapper, 41
- pacmanBit
 - Wrapper, 41
- ParcourirMatricesTransitions
 - ParcourirMatricesTransitions.m, 49
- ParcourirMatricesTransitions.m
 - ParcourirMatricesTransitions, 49
- ParrallelComposition
 - ParrallelComposition.m, 50
- ParrallelComposition.m
 - ParrallelComposition, 50
- Pas_a_pas
 - Labyrinthe, 16
- PathResearche
 - AutomateGraph, 9
 - Automaton, 13
- Plan_desumaFunctions.m
 - AutomatonStrutureLabyCreation, 45
 - function, 45
 - SaveDESUMAFFile, 45
 - writeStates, 45
 - writeTransitions, 45
- presentState
 - ModelLaby, 20
 - ModelPacman, 24
 - ModelSED, 27
 - ModelWalls, 31
 - StopCondition, 36
- rafineAutomatonClass
 - rafineAutomatonClass.m, 49
- rafineAutomatonClass.m
 - rafineAutomatonClass, 49
- resetUIConnection
 - figure_Laby.m, 56
- SaveDESUMAFFile
 - Plan_desumaFunctions.m, 45
 - SaveDESUMAFFile.m, 46
- SaveDESUMAFFile.m
 - SaveDESUMAFFile, 46
- setColor

- setColor.m, 61
- setColor.m, 60
 - setColor, 61
- Simulation.m, 61
- state
 - AutomateGraph, 10
- stop
 - Wrapper, 41
- StopCondition, 32
 - f, 34
 - g, 35
 - initialState, 36
 - m, 35
 - presentState, 36
 - StopCondition, 33
- stopCondition
 - Wrapper, 41
- StopCondition.m, 61
- structAutomata2vectorAutomata
 - AutomateGraph, 9
- transition
 - AutomateGraph, 10
- ui_Callback
 - figure_Laby.m, 56
- updateConnexion
 - Wrapper, 40
- updatePresenceDetectorDisplay
 - figure_Laby.m, 57
- updateUIActiveCammand
 - figure_Laby.m, 58
- updateUIButton
 - figure_Laby.m, 58
- updateUIEscape
 - figure_Laby.m, 58
- updateUIPlayer
 - figure_Laby.m, 58
- updateUIWalls
 - figure_Laby.m, 59
- updateUIWallsAround
 - figure_Laby.m, 59
- updateUI
 - figure_Laby.m, 57
- val
 - ModelWalls, 31
- vector
 - AutomateGraph, 10
- vector2matrices
 - AutomateGraph, 9
- vector2structAutomata
 - AutomateGraph, 9
- visupacman.m, 61
- visupacman2.m, 61
- wallsBit
 - Wrapper, 41
- wallsBorder
 - wallsBorder.m, 61
- wallsBorder.m, 61
 - wallsBorder, 61
- whoPlay
 - Wrapper, 41
- Wrapper, 36
 - commandPacman, 40
 - commandWalls, 40
 - get_out, 38
 - get_stop, 38
 - in, 40
 - init, 39
 - modelLaby, 40
 - orderer, 39
 - out, 41
 - pacmanBit, 41
 - stop, 41
 - stopCondition, 41
 - updateConnexion, 40
 - wallsBit, 41
 - whoPlay, 41
 - Wrapper, 37
- Wrapper.m, 61
- writeStates
 - Plan_desumaFunctions.m, 45
 - writeStates.m, 46
- writeStates.m
 - writeStates, 46
- writeTransitions
 - Plan_desumaFunctions.m, 45
 - writeTransitions.m, 47
- writeTransitions.m
 - writeTransitions, 47