

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	handle Class Reference	7
4.2	ModelLaby Class Reference	7
4.2.1	Detailed Description	8
4.2.2	Constructor & Destructor Documentation	9
4.2.2.1	ModelLaby()	9
4.2.3	Member Function Documentation	9
4.2.3.1	f()	9
4.2.3.2	g()	10
4.2.3.3	m()	10
4.2.4	Member Data Documentation	10
4.2.4.1	initialState	10
4.2.4.2	presentState	11
4.3	ModelPacman Class Reference	11
4.3.1	Detailed Description	12
4.3.2	Constructor & Destructor Documentation	12

4.3.2.1	ModelPacman()	12
4.3.3	Member Function Documentation	12
4.3.3.1	f()	12
4.3.3.2	g()	13
4.3.3.3	m()	13
4.3.4	Member Data Documentation	14
4.3.4.1	initialState	14
4.3.4.2	presentState	14
4.4	ModelSED Class Reference	14
4.4.1	Detailed Description	15
4.4.2	Member Function Documentation	15
4.4.2.1	f()	15
4.4.2.2	g()	16
4.4.2.3	m()	16
4.4.3	Member Data Documentation	16
4.4.3.1	initialState	16
4.4.3.2	presentState	17
4.5	ModelWalls Class Reference	17
4.5.1	Detailed Description	18
4.5.2	Constructor & Destructor Documentation	18
4.5.2.1	ModelWalls()	18
4.5.3	Member Function Documentation	18
4.5.3.1	f() [1/2]	18
4.5.3.2	f() [2/2]	19
4.5.3.3	g()	19
4.5.3.4	m()	19
4.5.4	Member Data Documentation	20
4.5.4.1	i	20
4.5.4.2	initialState	20
4.5.4.3	presentState	20

4.5.4.4	val	20
4.6	StopCondition Class Reference	21
4.6.1	Constructor & Destructor Documentation	21
4.6.1.1	StopCondition()	21
4.6.2	Member Function Documentation	22
4.6.2.1	f() [1/2]	22
4.6.2.2	f() [2/2]	22
4.6.2.3	g()	22
4.6.2.4	m()	23
4.6.3	Member Data Documentation	23
4.6.3.1	initialState	23
4.6.3.2	presentState	23
4.7	Wrapper Class Reference	24
4.7.1	Constructor & Destructor Documentation	24
4.7.1.1	Wrapper()	24
4.7.2	Member Function Documentation	24
4.7.2.1	get_out()	24
4.7.2.2	get_stop()	25
4.7.2.3	init()	25
4.7.2.4	orderer()	25
4.7.2.5	updateConnexion()	25
4.7.3	Member Data Documentation	25
4.7.3.1	commandPacman	25
4.7.3.2	commandWalls	25
4.7.3.3	in	26
4.7.3.4	modelLaby	26
4.7.3.5	out	26
4.7.3.6	pacmanBit	26
4.7.3.7	stop	26
4.7.3.8	stopCondition	26
4.7.3.9	wallsBit	26
4.7.3.10	whoPlay	26

5 File Documentation	27
5.1 CreatePituresAndVideo.m File Reference	27
5.1.1 Function Documentation	27
5.1.1.1 CreatePituresAndVideo()	27
5.2 CreatePituresAndVideo_textured.m File Reference	27
5.2.1 Function Documentation	27
5.2.1.1 CreatePituresAndVideo_textured()	28
5.3 figure_Laby.m File Reference	28
5.3.1 Function Documentation	28
5.3.1.1 connect_Callback()	28
5.3.1.2 createUIEscape()	28
5.3.1.3 createUIPacman()	29
5.3.1.4 createUIWalls()	29
5.3.1.5 figure_Laby()	29
5.3.1.6 figure_Laby_OpeningFcn()	29
5.3.1.7 figure_Laby_OutputFcn()	29
5.3.1.8 isOne()	29
5.3.1.9 resetUIConnection()	30
5.3.1.10 ui_Callback()	30
5.3.1.11 updatePresenceDetectorDisplay()	30
5.3.1.12 updateUI()	30
5.3.1.13 updateUIActiveCammand()	30
5.3.1.14 updateUIButton()	30
5.3.1.15 updateUIEscape()	31
5.3.1.16 updateUIPlayer()	31
5.3.1.17 updateUIWalls()	31
5.3.1.18 updateUIWallsAround()	31
5.4 main.m File Reference	31
5.5 Modellaby.m File Reference	31
5.6 ModelPacman.m File Reference	32

5.6.1 Detailed Description	32
5.7 ModelSED.m File Reference	32
5.7.1 Detailed Description	33
5.8 ModelWalls.m File Reference	33
5.8.1 Detailed Description	33
5.9 setColor.m File Reference	33
5.9.1 Function Documentation	33
5.9.1.1 setColor()	33
5.10 Simulation.m File Reference	34
5.11 StopCondition.m File Reference	34
5.12 testUI.m File Reference	34
5.13 THEplan.m File Reference	34
5.14 visupacman.m File Reference	34
5.15 visupacman2.m File Reference	34
5.16 wallsBorder.m File Reference	34
5.16.1 Function Documentation	34
5.16.1.1 wallsBorder()	34
5.17 Wrapper.m File Reference	34
Index	35

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

handle	7
ModelSED	14
ModelLaby	7
ModelPacman	11
ModelWalls	17
StopCondition	21
Wrapper	24

Chapter 2

Class Index

2.1 Class List

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

handle	7
ModelLaby Class which contains the "fmg" structure of the labyrinth for 1 player	7
ModelPacman 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) 11	
ModelSED State : minimal information necessary who evolute	14
ModelWalls This command do the sequence walls Right → walls down 17	
StopCondition	21
Wrapper	24

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

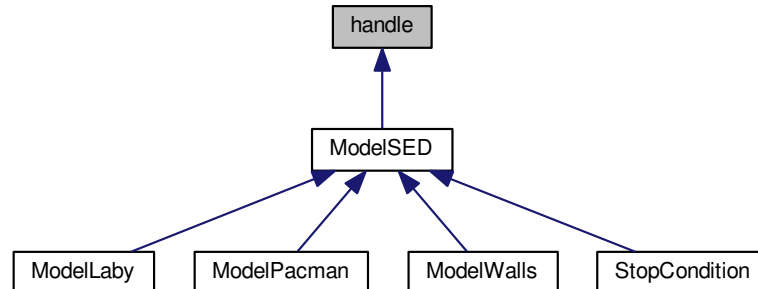
CreatePituresAndVideo.m	27
CreatePituresAndVideo_textured.m	27
figure_Laby.m	28
main.m	31
ModelLaby.m	31
ModelPacman.m	
Command of the Pacman's moves Input : Possible Pacman's moves [Up Down Left Right]	
0 = move not possible ; 1 = move possible	
(Wout{7})	
Output : Pacman's moves 1 : pacmanLeftBut, (Wout(3))	
2 : pacmanUpBut, (Wout(1))	
3 : pacmanRightBut, (Wout(4))	
4 : pacmanDownBut , (Wout(2))	
(Win(4:7) of wrapper)	
32	
ModelSED.m	
abstract Class who contain the structure of a "fmg" implementation Input : necessary information	
for compute the next state of the model	32
ModelWalls.m	
Command of the walls' move Input : No need	
Output : [UPwalls , RIGHTwalls]	33
setColor.m	33
Simulation.m	34
StopCondition.m	34
testUI.m	34
THEplan.m	34
visupacman.m	34
visupacman2.m	34
wallsBorder.m	34
Wrapper.m	34

Chapter 4

Class Documentation

4.1 handle Class Reference

Inheritance diagram for handle:



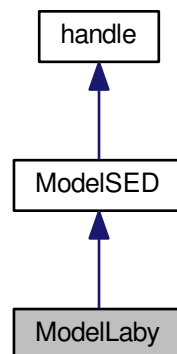
The documentation for this class was generated from the following file:

- [ModelSED.m](#)

4.2 ModelLaby Class Reference

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

Inheritance 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 update the state of the command.
- 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.

4.2.1 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 evolute

4.2.2 Constructor & Destructor Documentation

4.2.2.1 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.

4.2.3 Member Function Documentation

4.2.3.1 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](#).

4.2.3.2 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](#).

4.2.3.3 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](#).

4.2.4 Member Data Documentation

4.2.4.1 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.

4.2.4.2 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](#)

4.3 ModelPacman Class Reference

Input : Walls around Pacman

1 up

2 down

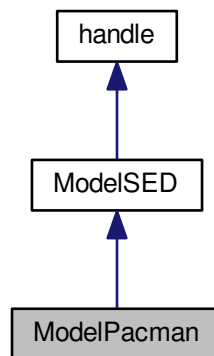
3 left

4 right

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

.

Inheritance 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.

4.3.1 Detailed Description

Input : Walls around Pacman

1 up

2 down

3 left

4 right

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

.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 ModelPacman()

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

Class constructor.

Parameters

<i>initialValue</i>	Contain the initial state
---------------------	---------------------------

Returns

instance of the [ModelPacman](#) class.

4.3.3 Member Function Documentation

4.3.3.1 f()

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

Compute the evolution of the command.

Parameters

<i>obj</i>	The instance who evolve
<i>in</i>	Input needed for the compute

Return values

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

Reimplemented from [ModelSED](#).

4.3.3.2 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](#).

4.3.3.3 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](#).

4.3.4 Member Data Documentation

4.3.4.1 initialState

Property initialState

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

4.3.4.2 presentState

Property presentState

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

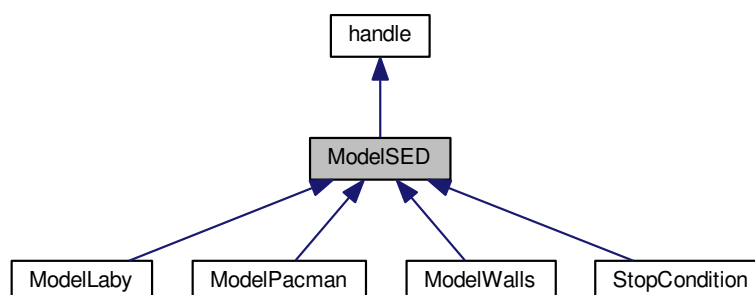
The documentation for this class was generated from the following file:

- [ModelPacman.m](#)

4.4 ModelSED Class Reference

State : minimal information necessary who evolve.

Inheritance 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.

4.4.1 Detailed Description

State : minimal information necessary who evolve.

4.4.2 Member Function Documentation

4.4.2.1 [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](#).

4.4.2.2 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](#), [ModelWalls](#), and [StopCondition](#).

4.4.2.3 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](#), [ModelWalls](#), and [StopCondition](#).

4.4.3 Member Data Documentation

4.4.3.1 initialState

Property `initialState`

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

4.4.3.2 presentState

Property presentState

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

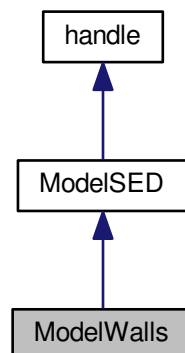
The documentation for this class was generated from the following file:

- [ModelSED.m](#)

4.5 ModelWalls Class Reference

This command do the sequence walls Right → walls down
.

Inheritance 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](#)

4.5.1 Detailed Description

This command do the sequence walls Right → walls down

.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 ModelWalls()

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

Class constructor.

Parameters

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

Returns

instance of the [ModelWalls](#) class.

4.5.3 Member Function Documentation

4.5.3.1 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](#).

4.5.3.2 **f()** [2/2]

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

4.5.3.3 **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](#).

4.5.3.4 **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](#).

4.5.4 Member Data Documentation**4.5.4.1 i**

Property i

4.5.4.2 initialState

Property initialState

4.5.4.3 presentState

Property presentState

4.5.4.4 val

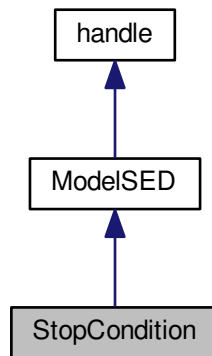
Property val

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

- [ModelWalls.m](#)

4.6 StopCondition Class Reference

Inheritance diagram for StopCondition:



Public Member Functions

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

4.6.1 Constructor & Destructor Documentation

4.6.1.1 StopCondition()

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

4.6.2 Member Function Documentation

4.6.2.1 `f()` [1/2]

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

4.6.2.2 `f()` [2/2]

```
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](#).

4.6.2.3 `g()`

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

Create the outputs.

Parameters

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

Return values

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

Reimplemented from [ModelSED](#).

4.6.2.4 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](#).

4.6.3 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:

- [StopCondition.m](#)

4.7 Wrapper Class Reference

Public Member Functions

- function [Wrapper](#) (in inSize, in outSize, in initLaby, in initWalls, in initPac, in initStop)
- function [updateConnexion](#) (in obj, in indBit, in value)
- function [init](#) (in obj)
- function [orderer](#) (in obj, in vectIn)
- function [get_stop](#) (in obj)
- function [get_out](#) (in obj)

Public Attributes

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

4.7.1 Constructor & Destructor Documentation

4.7.1.1 Wrapper()

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

4.7.2 Member Function Documentation

4.7.2.1 get_out()

```
function get\_out (  
    in obj )
```


4.7.2.2 get_stop()

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

4.7.2.3 init()

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

4.7.2.4 orderer()

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

4.7.2.5 updateConnexion()

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

4.7.3 Member Data Documentation

4.7.3.1 commandPacman

Property commandPacman

4.7.3.2 commandWalls

Property commandWalls

4.7.3.3 in

Property in

4.7.3.4 modelLaby

Property modelLaby

4.7.3.5 out

Property out

4.7.3.6 pacmanBit

Property pacmanBit

4.7.3.7 stop

Property stop

4.7.3.8 stopCondition

Property stopCondition

4.7.3.9 wallsBit

Property wallsBit

4.7.3.10 whoPlay

Property whoPlay

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

- [Wrapper.m](#)

Chapter 5

File Documentation

5.1 CreatePituresAndVideo.m File Reference

Functions

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

5.1.1 Function Documentation

5.1.1.1 CreatePituresAndVideo()

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

5.2 CreatePituresAndVideo_textured.m File Reference

Functions

- function [CreatePituresAndVideo_textured](#) (in n, in escape_i, in labyState)

5.2.1 Function Documentation

5.2.1.1 CreatePituresAndVideo_textured()

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

5.3 figure_Laby.m File Reference

Functions

- function [figure_Laby](#) (in varargin)
- function [figure_Laby_OpeningFcn](#) (in hObject, in eventdata, in handles, in varargin)
- function [figure_Laby_OutputFcn](#) (in hObject, in eventdata, in handles)
- function [ui_Callback](#) (in hObject, in eventdata, in handles)
- function [connect_Callback](#) (in hObject, in eventdata, in handles)
- function [createUIPacman](#) (in handles)
- function [createUIWalls](#) (in handles)
- function [createUIEscape](#) (in handles)
- function [updateUI](#) (in handles, in out)
- function [updateUIActiveCammand](#) (in handles)
- function [updateUIButton](#) (in handles)
- function [updateUIPlayer](#) (in handles, in strPlayer, in position)
- function [updateUIEscape](#) (in elementToSet, in boolState)
- function [updateUIWallsAround](#) (in handles, in strElement, in wallsAround)
- function [updateUIWalls](#) (in wallsUI, in vertWalls, in horizWalls)
- function [isOne](#) (in boolCond)
- function [updatePresenceDetectorDisplay](#) (in elementToSet, in boolCondition)
- function [resetUIConnection](#) (in handles)

5.3.1 Function Documentation

5.3.1.1 connect_Callback()

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

5.3.1.2 createUIEscape()

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

5.3.1.3 createUIPacman()

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

5.3.1.4 createUIWalls()

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

5.3.1.5 figure_Laby()

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

5.3.1.6 figure_Laby_OpeningFcn()

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

5.3.1.7 figure_Laby_OutputFcn()

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

5.3.1.8 isOne()

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

5.3.1.9 resetUIConnection()

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

5.3.1.10 ui_Callback()

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

5.3.1.11 updatePresenceDetectorDisplay()

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

5.3.1.12 updateUI()

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

5.3.1.13 updateUIActiveCammand()

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

5.3.1.14 updateUIButton()

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

5.3.1.15 updateUIEscape()

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

5.3.1.16 updateUIPlayer()

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

5.3.1.17 updateUIWalls()

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

5.3.1.18 updateUIWallsAround()

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

5.4 main.m File Reference

5.5 ModelLaby.m File Reference

Classes

- class [ModelLaby](#)

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

5.6 ModelPacman.m File Reference

Command of the Pacman's moves 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)

Classes

- class [ModelPacman](#)
Input : Walls around Pacman
1 up
2 down
3 left
4 right
This command do the sequence $P(D) > P(B) > P(H) > P(G)$
.

5.6.1 Detailed Description

Command of the Pacman's moves 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)

5.7 ModelSED.m File Reference

abstract Class who contain the structure of a "fmg" implementation Input : necessary information for compute the next state of the model

Classes

- class [ModelSED](#)
State : minimal information necessary who evolve.

5.7.1 Detailed Description

abstract Class who contain the structure of a "fmg" implementation
Input : necessary information for compute the next state of the model

Output : output's action of the model

5.8 ModelWalls.m File Reference

Command of the walls' move Input : No need

Output : [UPwalls , RIGHTwalls]

Classes

- class [ModelWalls](#)

This command do the sequence walls Right → walls down

.

5.8.1 Detailed Description

Command of the walls' move Input : No need

Output : [UPwalls , RIGHTwalls]

5.9 setColor.m File Reference

Functions

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

5.9.1 Function Documentation

5.9.1.1 setColor()

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

5.10 Simulation.m File Reference

5.11 StopCondition.m File Reference

Classes

- class [StopCondition](#)

5.12 testUI.m File Reference

5.13 THEplan.m File Reference

5.14 visupacman.m File Reference

5.15 visupacman2.m File Reference

5.16 wallsBorder.m File Reference

Functions

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

5.16.1 Function Documentation

5.16.1.1 wallsBorder()

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

5.17 Wrapper.m File Reference

Classes

- class [Wrapper](#)

Index

- commandPacman
 - Wrapper, [25](#)
- commandWalls
 - Wrapper, [25](#)
- connect_Callback
 - figure_Laby.m, [28](#)
- CreatePituresAndVideo
 - CreatePituresAndVideo.m, [27](#)
- CreatePituresAndVideo.m, [27](#)
 - CreatePituresAndVideo, [27](#)
- CreatePituresAndVideo_textured
 - CreatePituresAndVideo_textured.m, [27](#)
- CreatePituresAndVideo_textured.m, [27](#)
 - CreatePituresAndVideo_textured, [27](#)
- createUIEscape
 - figure_Laby.m, [28](#)
- createUIPacman
 - figure_Laby.m, [28](#)
- createUIWalls
 - figure_Laby.m, [29](#)
- f
 - ModelLaby, [9](#)
 - ModelPacman, [12](#)
 - ModelSED, [15](#)
 - ModelWalls, [18](#), [19](#)
 - StopCondition, [22](#)
- figure_Laby
 - figure_Laby.m, [29](#)
- figure_Laby.m, [28](#)
 - connect_Callback, [28](#)
 - createUIEscape, [28](#)
 - createUIPacman, [28](#)
 - createUIWalls, [29](#)
 - figure_Laby, [29](#)
 - figure_Laby_OpeningFcn, [29](#)
 - figure_Laby_OutputFcn, [29](#)
 - isOne, [29](#)
 - resetUIConnection, [29](#)
 - ui_Callback, [30](#)
 - updatePresenceDetectorDisplay, [30](#)
 - updateUIActiveCammand, [30](#)
 - updateUIButton, [30](#)
 - updateUIEscape, [30](#)
 - updateUIPlayer, [31](#)
 - updateUIWalls, [31](#)
 - updateUIWallsAround, [31](#)
 - updateUI, [30](#)
- figure_Laby_OpeningFcn
 - figure_Laby.m, [29](#)
- figure_Laby_OutputFcn
 - figure_Laby.m, [29](#)
- g
 - ModelLaby, [9](#)
 - ModelPacman, [13](#)
 - ModelSED, [15](#)
 - ModelWalls, [19](#)
 - StopCondition, [22](#)
- get_out
 - Wrapper, [24](#)
- get_stop
 - Wrapper, [24](#)
- handle, [7](#)
- i
 - ModelWalls, [20](#)
- in
 - Wrapper, [25](#)
- init
 - Wrapper, [25](#)
- initialState
 - ModelLaby, [10](#)
 - ModelPacman, [14](#)
 - ModelSED, [16](#)
 - ModelWalls, [20](#)
 - StopCondition, [23](#)
- isOne
 - figure_Laby.m, [29](#)
- m
 - ModelLaby, [10](#)
 - ModelPacman, [13](#)
 - ModelSED, [16](#)
 - ModelWalls, [19](#)
 - StopCondition, [23](#)
- main.m, [31](#)
- ModelLaby, [7](#)
 - f, [9](#)
 - g, [9](#)
 - initialState, [10](#)
 - m, [10](#)
 - ModelLaby, [9](#)
 - presentState, [10](#)
- modelLaby
 - Wrapper, [26](#)
- ModelLaby.m, [31](#)
- ModelPacman, [11](#)
 - f, [12](#)

- g, [13](#)
- initialState, [14](#)
- m, [13](#)
- ModelPacman, [12](#)
- presentState, [14](#)
- ModelPacman.m, [32](#)
- ModelSED.m, [32](#)
- ModelSED, [14](#)
 - f, [15](#)
 - g, [15](#)
 - initialState, [16](#)
 - m, [16](#)
 - presentState, [16](#)
- ModelWalls, [17](#)
 - f, [18](#), [19](#)
 - g, [19](#)
 - i, [20](#)
 - initialState, [20](#)
 - m, [19](#)
 - ModelWalls, [18](#)
 - presentState, [20](#)
 - val, [20](#)
- ModelWalls.m, [33](#)
- orderer
 - Wrapper, [25](#)
- out
 - Wrapper, [26](#)
- pacmanBit
 - Wrapper, [26](#)
- presentState
 - ModelLaby, [10](#)
 - ModelPacman, [14](#)
 - ModelSED, [16](#)
 - ModelWalls, [20](#)
 - StopCondition, [23](#)
- resetUIConnection
 - figure_Laby.m, [29](#)
- setColor
 - setColor.m, [33](#)
- setColor.m, [33](#)
 - setColor, [33](#)
- Simulation.m, [34](#)
- stop
 - Wrapper, [26](#)
- StopCondition, [21](#)
 - f, [22](#)
 - g, [22](#)
 - initialState, [23](#)
 - m, [23](#)
 - presentState, [23](#)
 - StopCondition, [21](#)
- stopCondition
 - Wrapper, [26](#)
- StopCondition.m, [34](#)
- THEplan.m, [34](#)
- testUI.m, [34](#)
- ui_Callback
 - figure_Laby.m, [30](#)
- updateConnexion
 - Wrapper, [25](#)
- updatePresenceDetectorDisplay
 - figure_Laby.m, [30](#)
- updateUIActiveCammand
 - figure_Laby.m, [30](#)
- updateUIButton
 - figure_Laby.m, [30](#)
- updateUIEscape
 - figure_Laby.m, [30](#)
- updateUIPlayer
 - figure_Laby.m, [31](#)
- updateUIWalls
 - figure_Laby.m, [31](#)
- updateUIWallsAround
 - figure_Laby.m, [31](#)
- updateUI
 - figure_Laby.m, [30](#)
- val
 - ModelWalls, [20](#)
- visupacman.m, [34](#)
- visupacman2.m, [34](#)
- wallsBit
 - Wrapper, [26](#)
- wallsBorder
 - wallsBorder.m, [34](#)
- wallsBorder.m, [34](#)
 - wallsBorder, [34](#)
- whoPlay
 - Wrapper, [26](#)
- Wrapper, [24](#)
 - commandPacman, [25](#)
 - commandWalls, [25](#)
 - get_out, [24](#)
 - get_stop, [24](#)
 - in, [25](#)
 - init, [25](#)
 - modelLaby, [26](#)
 - orderer, [25](#)
 - out, [26](#)
 - pacmanBit, [26](#)
 - stop, [26](#)
 - stopCondition, [26](#)
 - updateConnexion, [25](#)
 - wallsBit, [26](#)
 - whoPlay, [26](#)
 - Wrapper, [24](#)
- Wrapper.m, [34](#)