Al class Project

Generated by Doxygen 1.8.7

Fri Sep 5 2014 13:52:41

Contents

1	Data	Struct	ure Index		1
	1.1	Data S	Structures		1
2	File	Index			3
	2.1	File Lis	st		3
3	Data	Struct	ure Docur	mentation	5
	3.1	_TheB	rain Struct	Reference	5
		3.1.1	Field Do	cumentation	5
			3.1.1.1	pbar	5
			3.1.1.2	percent_comp	5
			3.1.1.3	window	5
	3.2	Node	Struct Refe	erence	5
		3.2.1	Detailed	Description	5
		3.2.2	Field Do	cumentation	6
			3.2.2.1	childern	6
			3.2.2.2	data	6
			3.2.2.3	lock	6
			3.2.2.4	numChildern	6
			3.2.2.5	numParent	6
			3.2.2.6	parent	6
4	File	Docum	entation		7
	4.1			s/Documents/Al-git/implementation/Al/classifier.h File Reference	7
		4.1.1		Documentation	7
			4.1.1.1	classifier_depth	7
				classifier_free	7
			4.1.1.3	classifier_init	7
			4.1.1.4	classifier_make_move	8
			4.1.1.5	classifier_prune	8
	4.2	/homo		s/Documents/Al-git/implementation/Al/heur.h File Reference	8
	4.2	/110111 U		5/2004ments/Argivimplementation/Armedi.HTHe Netetence	c

iv CONTENTS

		4.2.1.1	heur_depth	9
		4.2.1.2	heur_free	10
		4.2.1.3	heur_init	10
		4.2.1.4	heur_make_move	10
		4.2.1.5	heur_prune	10
4.3	/home/	/sweetness	s/Documents/Al-git/implementation/Al/TDNN.h File Reference	11
	4.3.1	Function	Documentation	11
		4.3.1.1	TDNN_depth	11
		4.3.1.2	TDNN_free	11
		4.3.1.3	TDNN_init	11
		4.3.1.4	TDNN_make_move	12
		4.3.1.5	TDNN_prune	12
4.4	/home/	/sweetness	s/Documents/AI-git/implementation/board_state.c File Reference	12
4.5	/home/	/sweetnes	s/Documents/AI-git/implementation/board_state.h File Reference	12
	4.5.1	Function	Documentation	12
		4.5.1.1	get_state	12
4.6	/home/	/sweetness	s/Documents/Al-git/implementation/gui/exit.c File Reference	13
	4.6.1	Function	Documentation	13
		4.6.1.1	delete_event	13
		4.6.1.2	destroy	13
4.7	/home/	/sweetness	s/Documents/AI-git/implementation/gui/exit.h File Reference	13
	4.7.1	Function	Documentation	13
		4.7.1.1	delete_event	13
		4.7.1.2	destroy	13
4.8	/home/	/sweetnes	s/Documents/Al-git/implementation/gui/game.c File Reference	13
	4.8.1	Typedef	Documentation	14
		4.8.1.1	TheBrain	14
	4.8.2	Function	Documentation	14
		4.8.2.1	main	14
		4.8.2.2	progress_timeout	14
		4.8.2.3	set_up_board	14
	4.8.3	Variable	Documentation	14
		4.8.3.1	adj	14
		4.8.3.2	align	14
		4.8.3.3	board_location	14
		4.8.3.4	fixed_main	15
		4.8.3.5	i_location	15
		4.8.3.6	player_turn	15
		4.8.3.7	thinking	15
		4.8.3.8	vbox	15

CONTENTS

		4.8.3.9	window_main	15
4.9	/home/	sweetness	s/Documents/Al-git/implementation/gui/initboard.c File Reference	15
	4.9.1	Function	Documentation	15
		4.9.1.1	clear_board	15
4.10	/home/	sweetness	s/Documents/Al-git/implementation/gui/initboard.h File Reference	15
	4.10.1	Function	Documentation	15
		4.10.1.1	clear_board	15
4.11	/home/	sweetness	s/Documents/Al-git/implementation/util/node.c File Reference	15
	4.11.1	Function	Documentation	16
		4.11.1.1	Node_addChild	16
		4.11.1.2	Node_addParent	16
		4.11.1.3	Node_create	16
		4.11.1.4	Node_delete	17
		4.11.1.5	Node_doneData	18
		4.11.1.6	Node_getData	18
4.12	/home/	sweetness	s/Documents/AI-git/implementation/util/node.h File Reference	18
	4.12.1	Function	Documentation	18
		4.12.1.1	Node_addChild	18
		4.12.1.2	Node_addParent	19
		4.12.1.3	Node_create	20
		4.12.1.4	Node_delete	20
		4.12.1.5	Node_doneData	20
		4.12.1.6	Node_getData	21
4.13	/home/	sweetness	s/Documents/Al-git/implementation/winchecker.c File Reference	21
	4.13.1	Function	Documentation	21
		4.13.1.1	check_win	21
4.14	/home/	sweetness	s/Documents/AI-git/implementation/winchecker.h File Reference	21
	4.14.1	Function	Documentation	21
		4.14.1.1	check_win	21
Index				22

Chapter 1

Data Structure Index

4	4	Data	Ctur	4	
1	1	Data	STri	ICTLI	ıres

Here are the data structures with brief descriptions:	
_TheBrain	
Node	

2 Data Structure Index

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

/home/sweetness/Documents/AI-git/implementation/board_state.c	2
/home/sweetness/Documents/Al-git/implementation/board_state.h	12
/home/sweetness/Documents/Al-git/implementation/winchecker.c	21
/home/sweetness/Documents/AI-git/implementation/winchecker.h	21
/home/sweetness/Documents/AI-git/implementation/AI/classifier.h	7
/home/sweetness/Documents/AI-git/implementation/AI/heur.h	8
/home/sweetness/Documents/AI-git/implementation/AI/TDNN.h	11
/home/sweetness/Documents/AI-git/implementation/gui/exit.c	13
/home/sweetness/Documents/AI-git/implementation/gui/exit.h	13
/home/sweetness/Documents/AI-git/implementation/gui/game.c	13
/home/sweetness/Documents/AI-git/implementation/gui/initboard.c	15
/home/sweetness/Documents/AI-git/implementation/gui/initboard.h	15
/home/sweetness/Documents/AI-git/implementation/util/node.c	5
/home/sweetness/Documents/Al-git/implementation/util/node.h	8

File Index

Chapter 3

Data Structure Documentation

3.1 _TheBrain Struct Reference

Data Fields

- GtkWidget * window
- GtkWidget * pbar
- · int percent_comp

3.1.1 Field Documentation

```
\textbf{3.1.1.1} \quad \textbf{GtkWidget} \\ * \_\textbf{TheBrain::pbar}
```

3.1.1.2 int _TheBrain::percent_comp

3.1.1.3 GtkWidget* _TheBrain::window

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

• /home/sweetness/Documents/Al-git/implementation/gui/game.c

3.2 Node Struct Reference

```
#include <node.h>
```

Data Fields

- pthread_mutex_t lock
- void * data
- struct Node ** childern
- struct Node ** parent
- int numChildern
- int numParent

3.2.1 Detailed Description

Copyright 2015

Author

Kaleb, Jacob, Angelica

This file is part of Artificially Intelligent Polar Tic Tac Toe game.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AI Polar Tic Tac Toe.

If not, see http://www.gnu.org/licenses/.

3.2.2 Field Documentation

3.2.2.1 struct Node** Node::childern

3.2.2.2 void* Node::data

3.2.2.3 pthread_mutex_t Node::lock

3.2.2.4 int Node::numChildern

3.2.2.5 int Node::numParent

3.2.2.6 struct Node** Node::parent

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

· /home/sweetness/Documents/Al-git/implementation/util/node.h

Chapter 4

File Documentation

4.1 /home/sweetness/Documents/Al-git/implementation/Al/classifier.h File Reference

Functions

```
• int classifier_init ()
```

- int classifier_make_move (int *xyarray)
- int classifier_prune (int prune)
- int classifier_free ()
- int classifier_depth (int depth)

4.1.1 Function Documentation

```
4.1.1.1 int classifier_depth ( int depth )
```

Set the depth of search if used

Parameters

```
depth the depth to search
```

Returns

0 for success

```
4.1.1.2 int classifier_free ( )
```

Free up memory allocated for the classifier

Returns

0 for success

```
4.1.1.3 int classifier_init ( )
```

Copyright 2014

Author

Kaleb, Jacob, Angelica

This file is part of Artificially Intelligent Polar Tic Tac Toe game.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AI Polar Tic Tac Toe.

If not, see http://www.gnu.org/licenses/. Initialize the classifier

Returns

returns 0 on success and 1 on fail

4.1.1.4 int classifier_make_move (int * xyarray)

Make a move with the classifier

Parameters

xyarray	the x and y integer for placement
---------	-----------------------------------

Returns

0 for success

4.1.1.5 int classifier_prune (int prune)

Set option for potential prunning

Parameters

prune	a boolean flag to set prunning
-------	--------------------------------

Returns

0 for success

4.2 /home/sweetness/Documents/Al-git/implementation/Al/heur.h File Reference

Functions

- int heur_init ()
- int heur_make_move (int *xyarray)
- int heur_prune (int prune)
- int heur_free ()
- int heur_depth (int depth)

- 4.2.1 Function Documentation
- 4.2.1.1 int heur_depth (int depth)

Set the depth of search if used

Parameters

depth	the depth to search
-------	---------------------

Returns

0 for success

4.2.1.2 int heur_free ()

Free up memory allocated for standard heuristic

Returns

0 for success

4.2.1.3 int heur_init ()

Copyright 2014

Author

Kaleb, Jacob, Angelica

This file is part of Artificially Intelligent Polar Tic Tac Toe game.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AI Polar Tic Tac Toe.

If not, see http://www.gnu.org/licenses/. Initialize the standard heuristic

Returns

returns 0 on success and 1 on fail

4.2.1.4 int heur_make_move (int * xyarray)

Make a move with the standard heuristic

Parameters

xyarray	the x and y integer for placement

Returns

0 for success

4.2.1.5 int heur_prune (int prune)

Set option for potential prunning

Parameters

prune	a boolean flag to set prunning

Returns

0 for success

4.3 /home/sweetness/Documents/Al-git/implementation/Al/TDNN.h File Reference

Functions

- int TDNN init ()
- int TDNN_make_move (int *xyarray)
- int TDNN prune (int prune)
- int TDNN_free ()
- int TDNN_depth (int depth)

4.3.1 Function Documentation

4.3.1.1 int TDNN_depth (int depth)

Set the depth of search if used

Parameters

depth	the depth to search
-------	---------------------

Returns

0 for success

4.3.1.2 int TDNN_free ()

Free up memory allocated for neural network

Returns

0 for success

4.3.1.3 int TDNN_init ()

Copyright 2014

Author

Kaleb, Jacob, Angelica

This file is part of Artificially Intelligent Polar Tic Tac Toe game.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Al Polar Tic Tac Toe.

If not, see http://www.gnu.org/licenses/. Initialize the neural network

Returns

returns 0 on success and 1 on fail

4.3.1.4 int TDNN_make_move (int * xyarray)

Make a move with the neural network

Parameters

xyarray the x and y integer for placement

Returns

0 for success

4.3.1.5 int TDNN_prune (int *prune*)

Set option for potential prunning

Parameters

prune a boolean flag to set prunning

Returns

0 for success

- 4.4 /home/sweetness/Documents/Al-git/implementation/board_state.c File Reference
- 4.5 /home/sweetness/Documents/Al-git/implementation/board_state.h File Reference

Functions

void get_state (int **array, int *height, int *width)

4.5.1 Function Documentation

4.5.1.1 void get_state (int ** array, int * height, int * width)

Gets the current positions of pieces on the board

Parameters

array	a two demensional array with board locations
height	sets the height of array [height][]
width	sets the width of array [][width]

4.6 /home/sweetness/Documents/Al-git/implementation/gui/exit.c File Reference

```
#include <gtk/gtk.h>
#include "exit.h"
```

Functions

- gboolean delete_event (GtkWidget *widget, GdkEvent *event, gpointer data)
- void destroy (GtkWidget *widget, gpointer data)

4.6.1 Function Documentation

```
4.6.1.1 gboolean delete_event ( GtkWidget * widget, GdkEvent * event, gpointer data )4.6.1.2 void destroy ( GtkWidget * widget, gpointer data )
```

4.7 /home/sweetness/Documents/Al-git/implementation/gui/exit.h File Reference

Functions

- gboolean delete_event (GtkWidget *widget, GdkEvent *event, gpointer data)
- void destroy (GtkWidget *widget, gpointer data)

4.7.1 Function Documentation

```
4.7.1.1 gboolean delete_event ( GtkWidget * widget, GdkEvent * event, gpointer data )4.7.1.2 void destroy ( GtkWidget * widget, gpointer data )
```

4.8 /home/sweetness/Documents/Al-git/implementation/gui/game.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdbool.h>
#include <gtk/gtk.h>
#include "exit.h"
#include "../winchecker.h"
#include "initboard.h"
```

Data Structures

struct _TheBrain

Typedefs

· typedef struct TheBrain TheBrain

Functions

```
void set_up_board ()
```

- gint progress_timeout (gpointer data)
- int main (int argc, char *argv[])

Variables

```
• int i_location [2]
```

- char board_location [17][17]
- char player turn = 'X'
- GtkWidget * window_main
- GtkWidget * fixed main
- TheBrain * thinking
- GtkWidget * vbox
- GtkWidget * align
- GtkAdjustment * adj

4.8.1 Typedef Documentation

- 4.8.1.1 typedef struct _TheBrain TheBrain
- 4.8.2 Function Documentation
- 4.8.2.1 int main (int argc, char * argv[])
- 4.8.2.2 gint progress_timeout (gpointer data)
- 4.8.2.3 void set_up_board ()
- 4.8.3 Variable Documentation
- 4.8.3.1 GtkAdjustment* adj
- 4.8.3.2 GtkWidget* align
- 4.8.3.3 char board_location[17][17]

Initial value:

4.9 /home/sweetness/Documents/Al-git/implementation/gui/initboard.c File Reference

```
#include <stdio.h>
#include <stdbool.h>
#include <gtk/gtk.h>
#include "initboard.h"
```

4.8.3.9 GtkWidget* window_main

Functions

void clear_board (char cBoardLoc[][17], char *cTurn, bool *bWon)

4.9.1 Function Documentation

4.9.1.1 void clear_board (char cBoardLoc[][17], char * cTurn, bool * bWon)

4.10 /home/sweetness/Documents/Al-git/implementation/gui/initboard.h File Reference

Functions

• void clear_board (char cBoardLoc[][17], char *cTurn, bool *bWon)

4.10.1 Function Documentation

4.10.1.1 void clear_board (char cBoardLoc[][17], char * cTurn, bool * bWon)

4.11 /home/sweetness/Documents/Al-git/implementation/util/node.c File Reference

```
#include "node.h"
```

Functions

struct Node * Node_create ()

- int Node_delete (struct Node *in)
- int Node_addParent (struct Node *in, struct Node *parent)
- int Node_addChild (struct Node *in, struct Node *child)
- void * Node_getData (struct Node *in)
- void Node_doneData (struct Node *in)

4.11.1 Function Documentation

```
4.11.1.1 int Node_addChild ( struct Node * in, struct Node * child ) [inline]
```

Adds a child to a node

Parameters

in	the address of node to add a child to
child	the address of node to be added as a child

Returns

0 on success

```
4.11.1.2 int Node_addParent ( struct Node * in, struct Node * parent ) [inline]
```

Adds a parent to a node

Parameters

in	the address of node to add a parent to
parent	the address of node to be added as a parent

Returns

0 on success

4.11.1.3 struct Node* Node_create ()

Copyright 2015

Author

Kaleb, Jacob, Angelica

This file is part of Artificially Intelligent Polar Tic Tac Toe game.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Al Polar Tic Tac Toe.

If not, see http://www.gnu.org/licenses/. Creates and allocates memory for a new node.

Returns

the pointer the the memory of the new node

4.11.1.4 int Node_delete (struct Node * in) [inline]

Used to properly delete a node and free up its memory

Parameters

the	addres to the node to be deleted
-----	----------------------------------

Returns

0 on success

```
4.11.1.5 void Node_doneData ( struct Node * in ) [inline]
```

Done using the nodes data. Use this function to make access to memory thread safe. in the address of the node that user is done with

```
4.11.1.6 void* Node_getData ( struct Node * in ) [inline]
```

Gets the data of the node. Use this function to make access to memory thread safe. Be sure to call function Node_doneData when done using the data in the address of the node to access memory from

4.12 /home/sweetness/Documents/Al-git/implementation/util/node.h File Reference

```
#include <stdlib.h>
#include <pthread.h>
```

Data Structures

struct Node

Functions

- struct Node * Node_create ()
- int Node_delete (struct Node *in)
- int Node_addParent (struct Node *in, struct Node *parent)
- int Node_addChild (struct Node *in, struct Node *child)
- void * Node_getData (struct Node *in)
- void Node_doneData (struct Node *in)

4.12.1 Function Documentation

```
4.12.1.1 int Node_addChild ( struct Node * in, struct Node * child ) [inline]
```

Adds a child to a node

Parameters

in	the address of node to add a child to
child	the address of node to be added as a child

Returns

0 on success

4.12.1.2 int Node_addParent (struct Node * in, struct Node * parent) [inline]

Adds a parent to a node

Parameters

in	the address of node to add a parent to
parent	the address of node to be added as a parent

Returns

0 on success

4.12.1.3 struct Node * Node_create ()

Creates and allocates memory for a new node.

Returns

the pointer the the memory of the new node

Copyright 2015

Author

Kaleb, Jacob, Angelica

This file is part of Artificially Intelligent Polar Tic Tac Toe game.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AI Polar Tic Tac Toe.

If not, see http://www.gnu.org/licenses/. Creates and allocates memory for a new node.

Returns

the pointer the the memory of the new node

4.12.1.4 int Node_delete (struct Node * in) [inline]

Used to properly delete a node and free up its memory

Parameters

the	addres to the node to be deleted

Returns

0 on success

4.12.1.5 void Node_doneData (struct Node * in) [inline]

Done using the nodes data. Use this function to make access to memory thread safe. in the address of the node that user is done with

```
4.12.1.6 void* Node_getData ( struct Node * in ) [inline]
```

Gets the data of the node. Use this function to make access to memory thread safe. Be sure to call function Node_doneData when done using the data in the address of the node to access memory from

4.13 /home/sweetness/Documents/Al-git/implementation/winchecker.c File Reference

```
#include <stdio.h>
#include <stdbool.h>
#include "winchecker.h"
```

Functions

• bool check_win (char cPlayerTurn, char cPlayerBoardLoc[][17])

4.13.1 Function Documentation

4.13.1.1 bool check_win (char cPlayerTurn, char cPlayerBoardLoc[][17])

Function to check for a win

Parameters

cPlayerTurn	current player to check
cPlayerBoardLoc	

Returns

a boolean value of win or not

4.14 /home/sweetness/Documents/Al-git/implementation/winchecker.h File Reference

Functions

bool check_win (char cPlayerTurn, char cPlayerBoardLoc[][17])

4.14.1 Function Documentation

4.14.1.1 bool check_win (char cPlayerTurn, char cPlayerBoardLoc[][17])

Function to check for a win

Parameters

cPlayerTurn	current player to check
cPlayerBoardLoc	

Returns

a boolean value of win or not

Index

```
childern
Node, 6

data
Node, 6

lock
Node, 6

Node, 5
childern, 6
data, 6
lock, 6
parent, 6

parent
Node, 6
```