

libpedsim

2.1

Generated by Doxygen 1.7.3

Fri Jan 20 2012 09:58:25

Contents

1	Todo List	1
2	Class Index	2
2.1	Class List	2
3	Class Documentation	2
3.1	Ped::Tagent Class Reference	2
3.1.1	Detailed Description	3
3.1.2	Constructor & Destructor Documentation	3
3.1.3	Member Function Documentation	3
3.2	Ped::Tobstacle Class Reference	7
3.2.1	Detailed Description	7
3.2.2	Constructor & Destructor Documentation	8
3.2.3	Member Function Documentation	8
3.3	Ped::Tscene Class Reference	9
3.3.1	Detailed Description	9
3.3.2	Constructor & Destructor Documentation	10
3.3.3	Member Function Documentation	10
3.4	Ped::Tvector Class Reference	10
3.4.1	Detailed Description	11
3.4.2	Constructor & Destructor Documentation	11
3.4.3	Member Function Documentation	11
3.5	Ped::Twaypoint Class Reference	12
3.5.1	Detailed Description	13
3.5.2	Constructor & Destructor Documentation	13
3.5.3	Member Function Documentation	14

1 Todo List

Member Ped::Tagent::addWaypoint(Twaypoint wp) Add a flag to change the way-point queue behavior of the Tagents.

Member Ped::Tagent::getFollow() Add a method that returns a Tagent*

Member Ped::Tagent::setFollow(int id) Add a method that takes a Tagent* as argument

Member Ped::Tvector::cross(Tvector *a, Tvector *b) This syntax is a bit weird. Will change in future.

Member Ped::Tvector::scalar(Tvector *a, Tvector *b) This syntax is a bit weird. Will change in future.

2 Class Index

2.1 Class List

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

Ped::Tagent	2
Ped::Tobstacle	7
Ped::Tscene	9
Ped::Tvector	10
Ped::Twaypoint	12

3 Class Documentation

3.1 Ped::Tagent Class Reference

```
#include <ped_agent.h>
```

Public Member Functions

- [Tagent](#) ()
- virtual void [move](#) (double h)
- virtual [Tvector](#) [socialForce](#) ()
- virtual [Tvector](#) [obstacleForce](#) ()
- virtual [Tvector](#) [desiredForce](#) ()
- virtual [Tvector](#) [lookaheadForce](#) ([Tvector](#) desired)
- virtual void [print](#) ()
- void [setPosition](#) (double px, double py, double pz)
- void [setType](#) (int t)
- void [setFollow](#) (int id)
- int [getFollow](#) ()
- void [setVmax](#) (double vmax)
- void [addWaypoint](#) ([Twaypoint](#) wp)
- int [getid](#) ()
- int [gettype](#) ()
- double [getx](#) ()
- double [gety](#) ()
- double [getz](#) ()
- double [getax](#) ()
- double [getay](#) ()
- double [getaz](#) ()
- double [getvx](#) ()

- double **getvy** ()
- double **getvz** ()
- void **assignScene** (Tscene *s)

3.1.1 Detailed Description

This is the main class of the library. It contains the [Tagent](#), which eventually will move through the [Tscene](#) and interact with [Tobstacle](#) and other [Tagent](#). You can use it as it is, and access the agent's coordinates using the `getx()` etc methods. Or, if you want to change the way the agent behaves, you can derive a new class from it, and overwrite the methods you want to change. This is also a convenient way to get access to internal variables not available through public methods, like the individual forces that affect the agent.

Author

chglloor

Date

2003-12-26

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Ped::Tagent::Tagent ()

Default Constructor

Date

2003-12-29

3.1.3 Member Function Documentation

3.1.3.1 void Ped::Tagent::addWaypoint (Twaypoint wp)

Adds a TWaypoint to an agent's list of waypoints. Twaypoints are stored in a cyclic queue, the one just visited is pushed to the back again. There will be a flag to change this behavior soon.

Todo

Add a flag to change the waypoint queue behavior of the Tagents.

Author

chglloor

Date

2012-01-19

3.1.3.2 void Ped::Tagent::assignScene (Ped::Tscene * s)

Assigns a [Tscene](#) to the agent. [Tagent](#) uses this to iterate over all obstacles and other agents in a scene. The scene will invoke this function when [Tscene::addAgent\(\)](#) is called.

Date

2012-01-17

Warning

Bad things will happen if the agent is not assigned to a scene. But usually, [Tscene](#) takes care of that.

Parameters

*s A valid Tscene initialized earlier.
--

Referenced by [Ped::Tscene::addAgent\(\)](#).

3.1.3.3 Ped::Tvector Ped::Tagent::desiredForce () [virtual]

Calculates the force between this agent and the next assigned waypoint. If the waypoint has been reached, the next waypoint in the list will be selected. At the moment, a visited waypoint is pushed back to the end of the list, which means that the agents will visit all the waypoints over and over again. In a later release, this behavior can be controlled by a flag.

Date

2012-01-17

Returns

[Tvector](#): the calculated force

3.1.3.4 int Ped::Tagent::getFollow ()

Gets the ID of the agent this agent is following.

Date

2012-01-18

Returns

int, the agent id of the agent

Todo

Add a method that returns a [Tagent*](#)

3.1.3.5 Ped::Tvector Ped::Tagent::lookaheadForce (Ped::Tvector e) [virtual]

Calculates the mental layer force of the strategy "look ahead". It is implemented here in the physical layer because of performance reasons. It iterates over all Tagents in the [Tscene](#), complexity $O(N^2)$.

Date

2012-01-17

Returns

[Tvector](#): the calculated force

Parameters

e	is a vector defining the direction in which the agent should look ahead to. Usually, this is the direction he wants to walk to.
-----	---

3.1.3.6 void Ped::Tagent::move (double h) [virtual]

Does the agent dynamics stuff. Calls the methods to calculate the individual forces, adds them to get the total force affecting the agent. This will then be translated into a velocity difference, which is applied to the agents velocity, and then to its position.

Date

2003-12-29

Parameters

h	This tells the simulation how far the agent should proceed (also known as Tau in literature). $1 = 1$ unit.
-----	---

3.1.3.7 Ped::Tvector Ped::Tagent::obstacleForce () [virtual]

Calculates the force between this agent and the nearest obstacle in this scene. Iterates over all obstacles == $O(N)$.

Date

2012-01-17

Returns

[Tvector](#): the calculated force

3.1.3.8 void Ped::Tagent::setFollow (int *id*)

Sets the agent ID this agent has to follow. If set, the agent will ignore its assigned waypoints and just follow the other agent.

Date

2012-01-08

Parameters

<i>id</i>	is the agent to follow (must exist, obviously)
-----------	--

Todo

Add a method that takes a Tagent* as argument

3.1.3.9 void Ped::Tagent::setPosition (double *px*, double *py*, double *pz*)

Sets the agent's position. This, and other getters returning coordinates, will eventually be changed to returning a [Tvector](#).

Date

2004-02-10

Parameters

<i>px</i>	Position x
<i>py</i>	Position y
<i>pz</i>	Position z

3.1.3.10 void Ped::Tagent::setVmax (double *pvmax*)

Sets the maximum velocity of an agent (vmax). Even if pushed by other agents, it will not move faster than this.

Date

2012-01-08

Parameters

<i>pvmax</i>	The maximum velocity. In scene units per timestep, multiplied by the simulation's precision h.
--------------	--

3.1.3.11 Ped::Tvector Ped::Tagent::socialForce () [virtual]

Calculates the social force between this agent and all the other agents belonging to the same scene. It iterates over all agents inside the scene, has therefore the complexity $O(N^2)$. A better agent storing structure in [Tscene](#) would fix this. But for small (less than 10000 agents) scenarios, this is just fine.

Date

2012-01-17

Returns

[Tvector](#): the calculated force

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

- ped_agent.h
- ped_agent.cpp

3.2 Ped::Tobstacle Class Reference

```
#include <ped_obstacle.h>
```

Public Member Functions

- [Tobstacle](#) ()
- [Tobstacle](#) (double ax, double ay, double bx, double by)
- void [setPosition](#) (double ax, double ay, double bx, double by)
- virtual [Tvector obstacleforce](#) (double p1, double p2)
- void [setType](#) (int t)
- int [getid](#) ()
- int [gettype](#) ()
- double [getax](#) ()
- double [getay](#) ()
- double [getbx](#) ()
- double [getby](#) ()

3.2.1 Detailed Description

Class that defines a [Tobstacle](#) object. An obstacle is, for now, always a wall with start and end coordinate.

Author

chglloor

Date

2012-01-17

3.2.2 Constructor & Destructor Documentation

3.2.2.1 Ped::Tobstacle::Tobstacle ()

Default constructor, places a wall from 0/0 to 1/1

Date

2012-01-07

3.2.2.2 Ped::Tobstacle::Tobstacle (double *pax*, double *pay*, double *pbx*, double *pby*)

Constructor used to set initial values.

Date

2012-01-07

Parameters

<i>pax</i>	x coordinate of the first corner of the obstacle.
<i>pay</i>	y coordinate of the first corner of the obstacle.
<i>pbx</i>	x coordinate of the second corner of the obstacle.
<i>pby</i>	y coordinate of the second corner of the obstacle.

3.2.3 Member Function Documentation

3.2.3.1 Ped::Tvector Ped::Tobstacle::obstacleforce (double *x*, double *y*)
[virtual]

Calculates and returns the forces of the obstacle to a given point x/y. x/y can be the location of an agent, but it can also be anything else, for example a grid coordinate of the user interface, if you want to display the obstacle forces on the map.

Date

2012-01-17

Returns

[Tvector](#) forces

Parameters

<i>double</i>	<i>x</i> : The x coordinate of the point
<i>double</i>	<i>y</i> : The y coordinate of the point

3.2.3.2 void Ped::Tobstacle::setPosition (double *pax*, double *pay*, double *pbx*, double *pby*)

Moves the obstacle to a new position. Can be uses to simulate opening doors etc.

Date

2012-01-07

Parameters

<i>pax</i>	x coordinate of the first corner of the obstacle.
<i>pay</i>	y coordinate of the first corner of the obstacle.
<i>pbx</i>	x coordinate of the second corner of the obstacle.
<i>pby</i>	y coordinate of the second corner of the obstacle.

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

- ped_obstacle.h
- ped_obstacle.cpp

3.3 Ped::Tscene Class Reference

```
#include <ped_scene.h>
```

Public Member Functions

- [Tscene](#) ()
- void [addAgent](#) ([Tagent](#) *a)
- void [addObstacle](#) ([Tobstacle](#) *o)

Friends

- class [Ped::Tagent](#)

3.3.1 Detailed Description

Class that defines a [Tscene](#) object

Author

chglloor

Date

2010-02-12

3.3.2 Constructor & Destructor Documentation

3.3.2.1 Ped::Tscene::Tscene ()

Default constructor.

Date

2012-01-17

3.3.3 Member Function Documentation

3.3.3.1 void Ped::Tscene::addAgent (Ped::Tagent * *a*)

Used to add a [Tagent](#) to the [Tscene](#).

Date

2012-01-17

Warning

[addAgent\(\)](#) does call [Tagent::assignScene\(\)](#) to assign itself to the agent.

Parameters

<i>*a</i>	A pointer to the Tagent to add.
-----------	---

3.3.3.2 void Ped::Tscene::addObstacle (Ped::Tobstacle * *o*)

Used to add a [Tobstacle](#) to the [Tscene](#).

Date

2012-01-17

Parameters

<i>*o</i>	A pointer to the Tobstacle to add.
-----------	--

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

- ped_scene.h
- ped_scene.cpp

3.4 Ped::Tvector Class Reference

```
#include <ped_vector.h>
```

Public Member Functions

- [Tvector](#) ()
- void [cross](#) ([Tvector](#) *a, [Tvector](#) *b)
- double [scalar](#) ([Tvector](#) *a, [Tvector](#) *b)
- void [normalize](#) ()

Public Attributes

- double **x**
- double **y**
- double **z**

3.4.1 Detailed Description

Vector helper class. This is basically a struct with some related functions attached.

Author

chglloor

Date

2010-02-12

3.4.2 Constructor & Destructor Documentation

3.4.2.1 Ped::Tvector::Tvector ()

Default constructor, which makes sure that all the values are set to 0.

Date

2012-01-16

3.4.3 Member Function Documentation

3.4.3.1 void Ped::Tvector::cross (Ped::Tvector * a, Ped::Tvector * b)

Vector cross product helper: calculates the cross product of two vectors.

Date

2010-02-12

[Todo](#)

This syntax is a bit weird. Will change in future.

Warning

The result is assigned to the vector calling the method.

Parameters

<i>*a</i>	The first vector
<i>*b</i>	The second vector

3.4.3.2 void Ped::Tvector::normalize ()

Normalizes the vector to a length of 1.

Date

2010-02-12

3.4.3.3 double Ped::Tvector::scalar (Ped::Tvector * *a*, Ped::Tvector * *b*)

Vector scalar product helper: calculates the scalar product of two vectors.

Date

2012-01-14

Todo

This syntax is a bit weird. Will change in future.

Returns

double The scalar product.

Parameters

<i>*a</i>	The first vector
<i>*b</i>	The second vector

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

- ped_vector.h
- ped_vector.cpp

3.5 Ped::Twaypoint Class Reference

```
#include <ped_waypoint.h>
```

Public Member Functions

- [Twaypoint](#) ()
- [Twaypoint](#) (double x, double y, double r)
- virtual [Tvector](#) [getForce](#) (double myx, double myy, double fromx, double fromy, bool *reached)
- virtual [Tvector](#) [normalpoint](#) (double p1, double p2, double oc11, double oc12, double oc21, double oc22)
- void [setType](#) (int t)
- int [getid](#) ()
- int [gettype](#) ()
- double [getx](#) ()
- double [gety](#) ()
- double [getr](#) ()
- void [setx](#) (double px)
- void [sety](#) (double py)
- void [setr](#) (double pr)
- void [settype](#) (int t)

3.5.1 Detailed Description

The waypoint classs

Author

chglloor

Date

2012-01-07

3.5.2 Constructor & Destructor Documentation**3.5.2.1 Ped::Twaypoint::Twaypoint ()**

Default Constructor - sets the most basic parameters.

Date

2012-01-07

3.5.2.2 Ped::Twaypoint::Twaypoint (double *px*, double *py*, double *pr*)

Constructor: Sets some intial values. The agent has to pass within the give radius.

Date

2012-01-07

Parameters

<i>px</i>	The x coordinate of the waypoint
<i>py</i>	The y coordinate of the waypoint
<i>pr</i>	The radius of the waypoint

3.5.3 Member Function Documentation

3.5.3.1 Ped::Tvector Ped::Twaypoint::getForce (double *myx*, double *myy*, double *fromx*, double *fromy*, bool * *reached*) [virtual]

Returns the force into the direction of the waypoint

Date

2012-01-10

Parameters

<i>myx</i>	The x coordinate of the current position of the agent
<i>myy</i>	The y coordinate of the current position of the agent
<i>fromx</i>	The x coordinate of the last assigned waypoint, i.e. where the agent is coming from
<i>fromy</i>	The y coordinate of the last assigned waypoint, i.e. where the agent is coming from
<i>*reached</i>	Set to true if the agent has reached the waypoint in this call.

Returns

[Tvector](#) The calculated force

3.5.3.2 Ped::Tvector Ped::Twaypoint::normalpoint (double *p1*, double *p2*, double *oc11*, double *oc12*, double *oc21*, double *oc22*) [virtual]

Calculates the point that is on the given line and normal to the give position. If it is not inside the line, the start or end point of the line is returned.

Date

2012-01-10

Parameters

<i>p1</i>	The x coordinate of the point outside the obstacle
<i>p2</i>	The y coordinate of the point outside the obstacle
<i>oc11</i>	The x coordinate of the first corner of the obstacle
<i>oc12</i>	The y coordinate of the first corner of the obstacle
<i>oc21</i>	The x coordinate of the second corner of the obstacle
<i>oc22</i>	The y coordinate of the second corner of the obstacle

Returns

[Tvector](#) The calculated point

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

- ped_waypoint.h
- ped_waypoint.cpp

Index

- addAgent
 - Ped::Tscene, 9
- addObstacle
 - Ped::Tscene, 9
- addWaypoint
 - Ped::Tagent, 3
- assignScene
 - Ped::Tagent, 3
- cross
 - Ped::Tvector, 11
- desiredForce
 - Ped::Tagent, 3
- getFollow
 - Ped::Tagent, 4
- getForce
 - Ped::Twaypoint, 13
- lookaheadForce
 - Ped::Tagent, 4
- move
 - Ped::Tagent, 5
- normalize
 - Ped::Tvector, 11
- normalpoint
 - Ped::Twaypoint, 14
- obstacleForce
 - Ped::Tagent, 5
- obstacleforce
 - Ped::Tobstacle, 8
- Ped::Tagent, 1
 - addWaypoint, 3
 - assignScene, 3
 - desiredForce, 3
 - getFollow, 4
 - lookaheadForce, 4
 - move, 5
 - obstacleForce, 5
 - setFollow, 5
 - setPosition, 5
 - setVmax, 6
 - socialForce, 6
- Tagent, 3
- Ped::Tobstacle, 6
 - obstacleforce, 8
 - setPosition, 8
 - Tobstacle, 7
- Ped::Tscene, 8
 - addAgent, 9
 - addObstacle, 9
 - Tscene, 9
- Ped::Tvector, 10
 - cross, 11
 - normalize, 11
 - scalar, 11
 - Tvector, 11
- Ped::Twaypoint, 12
 - getForce, 13
 - normalpoint, 14
 - Twaypoint, 13
- scalar
 - Ped::Tvector, 11
- setFollow
 - Ped::Tagent, 5
- setPosition
 - Ped::Tagent, 5
 - Ped::Tobstacle, 8
- setVmax
 - Ped::Tagent, 6
- socialForce
 - Ped::Tagent, 6
- Tagent
 - Ped::Tagent, 3
- Tobstacle
 - Ped::Tobstacle, 7
- Tscene
 - Ped::Tscene, 9
- Tvector
 - Ped::Tvector, 11
- Twaypoint
 - Ped::Twaypoint, 13