Global

Methods

`addMarker(point)`

add waypoint/marker to list

Parameters:

Name	Type	Description
`point`	Object	

Source: draw.js, line 232

`addObstacle(point)`

add single pinged obstacle to list

Parameters:

Name	Type	Description
`point`	Object	

Source: draw.js, line 213

`addVectors(vectorA, vectorB) → {Object}`

add two vectors

Parameters:

Name	Type	Description
`vectorA`	Object	
`vectorB`	Object	

Source: draw.js, line 343

Home

Global

Returns:

Type

Object

`amax(pointarray) → {Object}`

find xmax and ymax in a list of points

Parameters:

Name	Type	Description
`pointarray`	Array.<Object>	array of points

Source: draw.js, line 1104

Returns:

Type

Object

`amin(pointarray) → {Object}`

find xmin and ymin in a list of points

Parameters:

Name	Type	Description
`pointarray`	Array.<Object>	array of points

Source: draw.js, line 1121

Returns:

Type

Object

`clearMarkers()`

clear all markers from the map

Source: draw.js, line 240

load
log
log_clear
logAbsatz
minmax
minmaxAll
moveRobot
moveRobotTurn
multiplyScalar
normalize
POST_request
rad
reset
robot_rotate
robot_translate
robotAnimation
rotate_around_origin
save
setCanvasSize
setGridSize
setScale
vector
vector2
vectorAngleBetween
vectorLen
vectorRound
vectorScalarMultiply

createNewObstacleSet()

creates new set of obstacles (one each per PING sweep)

Source: draw.js, line 350

deg(rad) → {number}

convert radians to degrees

Parameters:

Name	Type	Description
rad	number	

Source: draw.js, line 416

Returns:

Type

number

delayThis(callback)

Delays communication by delay milliseconds

Parameters:

Name	Type	Description
callback		do this after

Source: bot.js, line 175

deleteObstacles(center, radius)

delete all obstacles within a certain radius around a point

Parameters:

Name	Type	Description
center	Object	
radius	number	

Source: draw.js, line 379

draw()

draw everything except the robot (->animation) on the canvas (calls all other draw functions)

Source: draw.js, line 1008

draw_plus_robot()

do draw() and draw the robot

Source: draw.js, line 1037

drawBot()

draw the robot on its position

Source: draw.js, line 863

drawCircle(center, radius)

draw a circle

Parameters:

Name	Type	Description
`center`	Object	
`radius`	number	

Source: draw.js, line 823

drawConnectLines(points, color, skipPoints)

connect a set of points on the map with a line

Parameters:

Name	Type	Description
`points`	Array.<Object>	array of points

Name	Type	Description
`color`		string
`skipPoints`		boolean if to skip points if more than bot diameter apart

Source: draw.js, line 798

`drawGrid(size, width)`

draw the grid

Parameters:

Name	Type	Description
`size`	number	
`width`	number	

Source: draw.js, line 894

`drawLine(pointA, pointB)`

draw a line between two points

Parameters:

Name	Type	Description
`pointA`	Object	
`pointB`	Object	

Source: draw.js, line 993

`drawMarkers()`

draw all markers in list on the map

Source: draw.js, line 394

`drawMovementPossibilities()`

draw possible movement angles reachable with ticks

> Source: draw.js, line 835

`drawObstacles()`

draw all obstacles on map

> Source: draw.js, line 357

`drawPoint(point, size)`

plot a point on the map

Parameters:

Name	Type	Description
`point`	Object	
`size`	number	

> Source: draw.js, line 980

`drive_command(point, relative_angle, distance)`

send drive command to Bot

Parameters:

Name	Type	Description
`point`	Object	
`relative_angle`		in degrees. if < 0: turn left this much. if >= 0 turn right this much
`distance`		to travel in cm

> Source: bot.js, line 7

`driveBackToStart()`

move robot back to starting position and delete path

> Source: draw.js, line 748

fitCanvasSize()

fit canvas to range of points

> Source: draw.js, line 1045

get_radar_points()

request radar sweep data

> Source: bot.js, line 46

GET_request(varname, callback)

send a GET request via HTTP

Parameters:

Name	Type	Description
varname		awaiting response on subpage /varname
callback		after receiving data function(responseText)

> Source: bot.js, line 107

getMousePos(canvas, evt) → {Object}

get mouse coordinates

Parameters:

Name	Type	Description
canvas	Object	canvas object
evt	Object	event

> Source: draw.js, line 1167

Returns:

point

Type

Object

getPoint(pos) → {Object}

converts canvas position to point on map

Parameters:

Name	Type	Description
pos	Object	

Source: draw.js, line 971

Returns:

Type

Object

getPos(point) → {Object}

converts points to canvas positions and enlarges the canvas if necessary

Parameters:

Name	Type	Description
point	Object	

Source: draw.js, line 930

Returns:

Type

Object

graph_rotate(angle)

rotate everything else than in robot_rotate()

Parameters:

Name	Type	Description
angle	number	

Source: draw.js, line 636

graph_translate(vec)

translate everything else than in robot_translate()

Parameters:

Name	Type	Description
vec	Object	

> Source: draw.js, line 652

httpGet(path, callback)

Parameters:

Name	Type	Description
path		listened to
callback		function(responseText)

> Source: bot.js, line 147

httpPost(path, param)

Parameters:

Name	Type	Description
path		
param		

> Source: bot.js, line 118

init()

initialize on window.onload: - all event listeners, - the canvas - draw grid / bot / other lines

> Source: draw.js, line 66

isCollidingWithObstacle(fromPoint, toPoint, length, atAngle, radius) → {boolean}

checks if there are obstacles within a radius on a vector

Parameters:

Name	Type	Description
`fromPoint`	Object	
`toPoint`	Object	
`length`	number	
`atAngle`	number	
`radius`	number	

> Source: draw.js, line 712

Returns:

Type

boolean

`load()`

load from cookie (modern browsers)

> Source: draw.js, line 158

`log(str)`

log to screen and console

Parameters:

Name	Type	Description
`str`		string

> Source: draw.js, line 774

`log_clear()`

clears the log below the canvas

> Source: draw.js, line 766

`logAbsatz(str)`

log to screen and console and print a spacer afterwards

Parameters:

Name	Type	Description
`str`		string

Source: draw.js, line 787

minmax(pointarr2d) → {Object}

find minimums and maximums in a list of lists of points

Parameters:

Name	Type	Description
`pointarr2d`	Array.<Array.<Object>>	2D array of points

Source: draw.js, line 1139

Returns:

Type

Object

minmaxAll() → {Object}

find minimum and maximum of ALL points drawn

Source: draw.js, line 1079

Returns:

Type

Object

moveRobot(point)

move robot to point. Moves robot only to reachable angles (no half ticks possible!)

Parameters:

Name	Type	Description
`point`	Object	

Source: draw.js, line 433

moveRobotTurn(degrees)

turn robot on the spot, angle normalizes to TICK_ANGLE

Parameters:

Name	Type	Description
`degrees`		

Source: draw.js, line 568

multiplyScalar(vector, scalar) → {Object}

multiply vector with scalar

Parameters:

Name	Type	Description
`vector`	Object	
`scalar`	number	

Source: draw.js, line 333

Returns:

Type

Object

normalize(degree) → {number}

make degrees positive and mod 360

Parameters:

Name	Type	Description
`degree`	number	

> Source: draw.js, line 425

Returns:

Type
: number

`POST_request(varname, value)`

send a POST request via HTTP to /post

Parameters:

Name	Type	Description
`varname`		name of variable to send
`value`		command to send

> Source: bot.js, line 97

`rad(deg) → {number}`

convert degrees to radians

Parameters:

Name	Type	Description
`deg`	number	

> Source: draw.js, line 407

Returns:

Type
: number

`reset()`

reset everything to start again

> Source: draw.js, line 142

robot_rotate(angle)

rotate robot and the last obstacle set around itself

Parameters:

Name	Type	Description
angle	number	

Source: draw.js, line 592

robot_translate(vec)

translate robot and last obstacle set along a vector

Parameters:

Name	Type	Description
vec	Object	

Source: draw.js, line 620

robotAnimation(cm_per_sec)

rudimentary robot animation

Parameters:

Name	Type	Description
cm_per_sec	number	

Source: draw.js, line 665

rotate_around_origin(point, angle) → {Object}

rotate a point around the origin (x=0,y=0) to rotate around other points translate before and after

Parameters:

Name	Type	Description
point	Object	
angle	number	

> Source: draw.js, line 581

Returns:

Type
Object

`save()`

save to cookie (modern browsers)

> Source: draw.js, line 183

`setCanvasSize(width, height)`

resize canvas to this size

Parameters:

Name	Type	Description
`width`	number	
`height`	number	

> Source: draw.js, line 202

`setGridSize(togridsize)`

change the width of the grid

Parameters:

Name	Type	Description
`togridsize`	number	

> Source: draw.js, line 259

`setScale(toscale)`

change the zoom/scale of the map

Parameters:

Name	Type	Description
`toscale`	number	

Source: draw.js, line 249

`vector(pointFrom, pointTo) → {Object}`

get vector between two points

Parameters:

Name	Type	Description
`pointFrom`	Object	
`pointTo`	Object	

Source: draw.js, line 270

Returns:

Type

Object

`vector2(length, degreefromx) → {Object}`

get vector from length and angle relative to x axis

Parameters:

Name	Type	Description
`length`	number	
`degreefromx`	number	

Source: draw.js, line 280

Returns:

Type

Object

vectorAngleBetween(vector1, vector2) → {number}

Angle between two vectors (absolute)

Parameters:

Name	Type	Description
vector1	Object	
vector2	Object	

Source: draw.js, line 323

Returns:

Type
number

vectorLen(vector) → {number}

get length of vector

Parameters:

Name	Type	Description
vector	Object	

Source: draw.js, line 292

Returns:

length

Type
number

vectorRound(vector) → {Object}

round vector values x,y to zero decimals

Parameters:

Name	Type	Description
vector	Object	

Source: draw.js, line 302

Returns:

Type
    Object

vectorScalarMultiply(vector1, vector2) → {number}

Scalar multiply two vectors

Parameters:

Name	Type	Description
`vector1`	Object	
`vector2`	Object	

Source: draw.js, line 313

Returns:

Type
    number

Documentation generated by JSDoc 3.5.5 on Wed Jan 30 2019 10:31:14 GMT+0100 (GMT+01:00)