Common Math 1.1

Compilation of essential math functions; by Faguss (ofp-faguss.com)

1. Installation

- Copy `CommonMath.sqf` to the mission directory.
- Write in `Init.sqs`:

```
call loadFile "CommonMath.sqf"
```

2. List of functions

MATH_PLACE2D	find point XY
MATH_PLACE3D	find point XYZ
MATH_DIST2D	distance between two points
MATH_DIST3D	distance between two points (including height)
MATH_POS2DIR	angle XY between two points
MATH_POS2PITCH	angle Z between two points
MATH_ABS2REL	find relative coordinates
MATH_REL2ABS	find absolute coordinates
MATH_POLYGONS_INTERSECT	find if one shape overlaps the other
MATH_CORNERS_RECTANGLE	find corner points of a rectangle
MATH_RADIUS_RECTANGLE	find distance to the perimeter of a rectangle
MATH_INSIDE_RECTANGLE	find if point is inside the rectangle
MATH_RADIUS_ELLIPSE	find distance to the perimeter of an ellipse
MATH_INSIDE_ELLIPSE	find if point is inside the ellipse
MATH_VEL2INFO	find movement speed, yaw and pitch of an object
MATH_ROUND	round number to given precision

3. Description

MATH_PLACE2D

Finds coordinates relative from given position at a specified distance and angle. Position can be in two or three dimensions. Alternatively you can pass one more number to specify horizontal and vertical offset.

```
[<array position XY or XYZ>, <angle>, <distance>] call MATH_PLACE2D
[<array position XY or XYZ>, <angle>, <side>, <forward>] call MATH_PLACE2D
```

MATH_PLACE3D

Finds coordinates relative from given position at a specified distance and angle in three dimensions.

```
[<array position XY or XYZ>, <yaw>, <bank>, <pitch>, <side>, <forward>,
<up>] call MATH_PLACE3D
```

MATH_DIST2D

Finds distance in two dimensions between two points.

```
[<array position XY or XYZ>, <array position XY or XYZ>] call MATH_DIST2D
```

MATH_DIST3D

Finds distance in three dimensions between two points.

```
[<array position XYZ>, <array position XYZ>] call MATH_DIST3D
```

MATH_POS2DIR

Finds horizontal angle between two points

```
[<array position XY or XYZ>, <array position XY or XYZ>] call MATH_POS2DIR
```

MATH_POS2PITCH

Finds vertical angle between two points.

```
[<array position XYZ>, <array position XYZ>] call MATH_POS2PITCH
```

MATH_ABS2REL

Finds relative coordinates between two positions in given direction.

```
[<array position XY or XYZ>, <angle>, <array position XY or XYZ>] call
MATH_ABS2REL
```

MATH_REL2ABS

Finds absolute coordinates from the specified position given relative coordinates and direction.

```
[<array position XY or XYZ>, <angle>, <array relative XY or XYZ>] call
MATH_REL2ABS
```

MATH_POLYGONS_INTERSECT

Finds if one polygon is intersecting the other. Each sub-array must contain at least two positions. Points must be given in a clockwise or counter-clockwise order. Works only with convex shapes.

```
[<array with positions XY or XYZ>, <array with positions XY or XYZ>] call
MATH_POLYGONS_INTERSECT
```

MATH_CORNERS_RECTANGLE

Finds location of the four corners of a rectangle of given size and rotation.

```
[<array position XY or XYZ>, <angle>, <width>, <height>] call
MATH_CORNERS_RECTANGLE
```

MATH_RADIUS_RECTANGLE

Finds distance between center and the edge of a rectangle of given size and rotation.

```
[<angle>, <width>, <height>] call MATH_RADIUS_RECTANGLE
```

MATH_INSIDE_RECTANGLE

Finds if position is in the area of a rectangle of given size and rotation.

```
[<array rectangle position XY or XYZ>, <angle>, <width>, <height>, <array
position XY or XYZ>] call MATH_INSIDE_RECTANGLE
```

MATH_RADIUS_ELLIPSE

Finds distance between center and the edge of an ellipse of given size and rotation.

```
[<angle>, <width>, <height>] call MATH_RADIUS_ELLIPSE
```

MATH_INSIDE_ELLIPSE

Finds if position is in the area of the ellipse of given size and rotation.

```
[<array ellipse position XY or XYZ>, <angle>, <width>, <height>, <array
position XY or XYZ>] call MATH_INSIDE_ELLIPSE
```

MATH_VEL2INFO

Finds movement speed, direction and elevation angle from the velocity of a specified object.

```
<object> call MATH_VEL2INFO
```

MATH_ROUND

Rounds number to a natural number. Optionally you can specify precision.

```
<number> call MATH_ROUND
[<number>, <round to>] call MATH_ROUND
```

See demo mission for example usage. Feel free to modify them to suit your needs.

3. Credits

MATH_PLACE2D

Formula taken from the trigonometry tutorial by THobson

https://www.ofpec.com/editors-depot/index.php?action=details&id=332&game=OFP

MATH_PLACE3D

Formula taken from the following website and modified with the help of Miki45

https://web.archive.org/web/20140122132306/http://thronic.com/articles/3D%20Rotation%20Trig.php

MATH_POS2DIR

Formula taken from the DirToPos.sqf function by snYpir

https://www.ofpec.com/editors-depot/index.php?action=details&id=99&game=OFP

MATH_POLYGONS_INTERSECT

Formula taken from the following website

https://stackoverflow.com/questions/10962379/how-to-check-intersection-between-2-rotated-rectangles

MATH_RADIUS_RECTANGLE

Formula taken from the following website and adapted with the help of Miki45

https://www.physicsforums.com/threads/equation-of-a-rectangles-radius.453778/#post-3054954

MATH_RADIUS_ELLIPSE

Formula taken from the following website

https://math.stackexchange.com/questions/432902

MATH_VEL2INFO

Formula taken from the intercept.sqf function by Raptorsaurus

https://www.ofpec.com/editors-depot/index.php?action=details&id=122&game=OFP

4. Version history

v1.0 (03.02.19)

First release.

v1.1 (05.06.19)

- added: MATH_DIST3D, MATH_POLYGONS_INTERSECT, MATH_CORNERS_RECTANGLE, MATH_INSIDE_RECTANGLE, MATH_INSIDE_ELLIPSE
- MATH_PLACE3D, MATH_ABS2REL, MATH_REL2ABS now accept 2D position input