Drawing functions

CROP\_PICTURE Crop the outer frame of the polygons by a quantity dl\_crop.

CREATE\_COLORS Creates an mx3 matrix of RGB colors randomly.

DRAW\_CONTACT\_POINTS Draw the contact points between the stones.

DRAW\_STONES Draw all the stones when the stones are random polygons.

PLOT\_BRICK Draw a rectangular brick according to its four corners.

ADD\_FUNDATION Summary of this function goes here

SAVE\_PICTURE Summary of this function goes here

WRITE\_NODES Write the number of the nodes on an existing picture.

Fractalization function

FRACTALIZE\_POLYGONS Function that will add a random shape to polygons.

Parameters functions

GET\_PARAMS\_EROSION will return the parameters of the erosion process, if needed according to the position of the stone.

GET\_PARAMS\_SHAKING will return the parameters of the shaking process, if needed according to the position of the stone.

GET\_PARAMS\_STGHT\_PATTERN will return the parameters of the straight pattern process process, if needed according to the position of the stone or the row of the stone.

PARAMETERS\_STONE\_MASONRY This script has to be launched before every calculation as it defines some of the basic parameters of the stone pattern generation algorith, such as wall length, and so on.

Random field Gen. functions

CORRELATION\_FUN returns the correlation matrix (or times a vector).

CREATE\_RANDOM\_FIELD generates a random field given the size of the map.

RANDOMFIELD Generates realizations of a Gaussian random field.

Resampling functions

GET\_PERIMETER computes the perimeter of a polygon given its vertices.

PUT\_CORNERS puts corners to polygons according to the max/min x-y coordinates

RESAMPLE\_POLYGONS resamples the polygons given a target edge length l\_edges.

Shaking functions

IS\_CONCAVE function that determine if a polygon is concave or not.

ISNOTONBORDER function that will tell if a point is on the border or no.

SHAKE function that will move slightly every point of the wall.

TESTNODE function that will assert that a given node is in a correct position (during shaking operation).

Shortest path functions

COMPUTE\_VISIBILITY\_GRAPH computes the visibility graph of an environnement

DIJKSTRA\_MODIFIED computes the shortest path with an adjacency matrix an a xy coordinates tab.

GET\_INTERLOCKING\_PATH gives the interlocking of a path (length/delta\_H)

GET\_POINTS function that allows the user to pick the starting and ending point in the current figure.

MAKE\_ENVIRONNEMENT create a data structure that is needed by the visilibity toolbox.

PLOT\_SHORTEST\_PATH Summary of the function goes here

Sieving

SIEVING sieves the polygons, ie. removes all the smallest stones.

Erosion functions

ACTUALIZE\_SURF actualizes the vector containing the coordinates of the pixels on the surface of the polygons during the erosion process.

ADD\_NEW\_EXPOSURE function that computes the initial exposure value of a pixel

ERODE\_POLYGON function that erodes a polygon.

ERODE\_STONES function that erodes all the stones

FIND\_NEIGHBORS function that will find the neighbors of all stones in the wall.

GET\_CONTACT\_POINTS function that will create the contact points for all the stones in the wall.

GET\_DAMPING\_EROSION function that will compute the damping factor for the erosion with contact points.

GET\_MASK\_EXPOSURE function that compute the mask of the bubble

IS\_ON\_SURFACE function that verifies if a pixel is on the surface of the polygon or no.

REORDER\_BY\_DISTANCE\_POLYGON function that reorders a polygon with the method of the closest neighbor.

UPDATE\_EXPOSURE function that update the value of exposure of a pixel.

Straight Pattern functions

ADDBRICK core function, that will add a brick to the wall.

ADDORFINDPOINT function that will either add a new point to the position matrix or find it if it already exists in this position matrix.

% ADDROW core function that will add a row to the existing wall, based on the topline

ADDUNDERSTONE function that will add an understone when called.

CHECK\_VALIDITY checks the validity of a straight pattern.

CREATE\_BRICK function that will create the aleatory dimensions of a brick, given a medium length and a standard deviation.

FILL\_WALL core function that will initialize all the useful variables and fill the wall adding rows while the wall is not full.

FIND\_XSTART function that will find the first x coordinate on the topline that is not equal to Ly\_wall.

GET\_ALL\_NODES function that will get all nodes corresponding to a stone and order them in a clockwise order.

GET\_NEXT\_CHANGE function that will find the first change in a vector.

GETXNEXTBRICK function that will get the x coordinate of the next brick on the topline.

GETYSTART function that will give the y coordinate of a given x coordinate on the topline.

IS\_TO\_DEL says if a stone is to be deleted or not.

UPDATE\_TOPLINE function that updates the topline when a new brick is added to the wall.

Utilities functions

DISTANZ function that compute the distance between two points

GET\_AIRE\_TRIANGLE function that computes the area of a triangle given its three vertices

GET\_AREA\_POLYONG function that computes the area of a polygon given its vertices

GET\_NUMBER\_VERTICES function that gets the number of vertices in an array of stones

ISONSTONE function that will tell if a point is on a stone or no.

ORDER\_POLYGON function that will order a polygon in clockwise order.

ORDER\_STONE\_NODES function that will order the nodes of a stone in a clockwise order.

REMOVE\_DOUBLE\_NODES will remove any "too close" nodes, ie. less than 1e-6 distance between them.

SAVE\_PARAMETERS Summary of this function goes here

%DELETE\_REDONDANT\_VERTICES This function deletes any vertex that is on a line between its two neighbors (ie. if two consecutive edges are //)