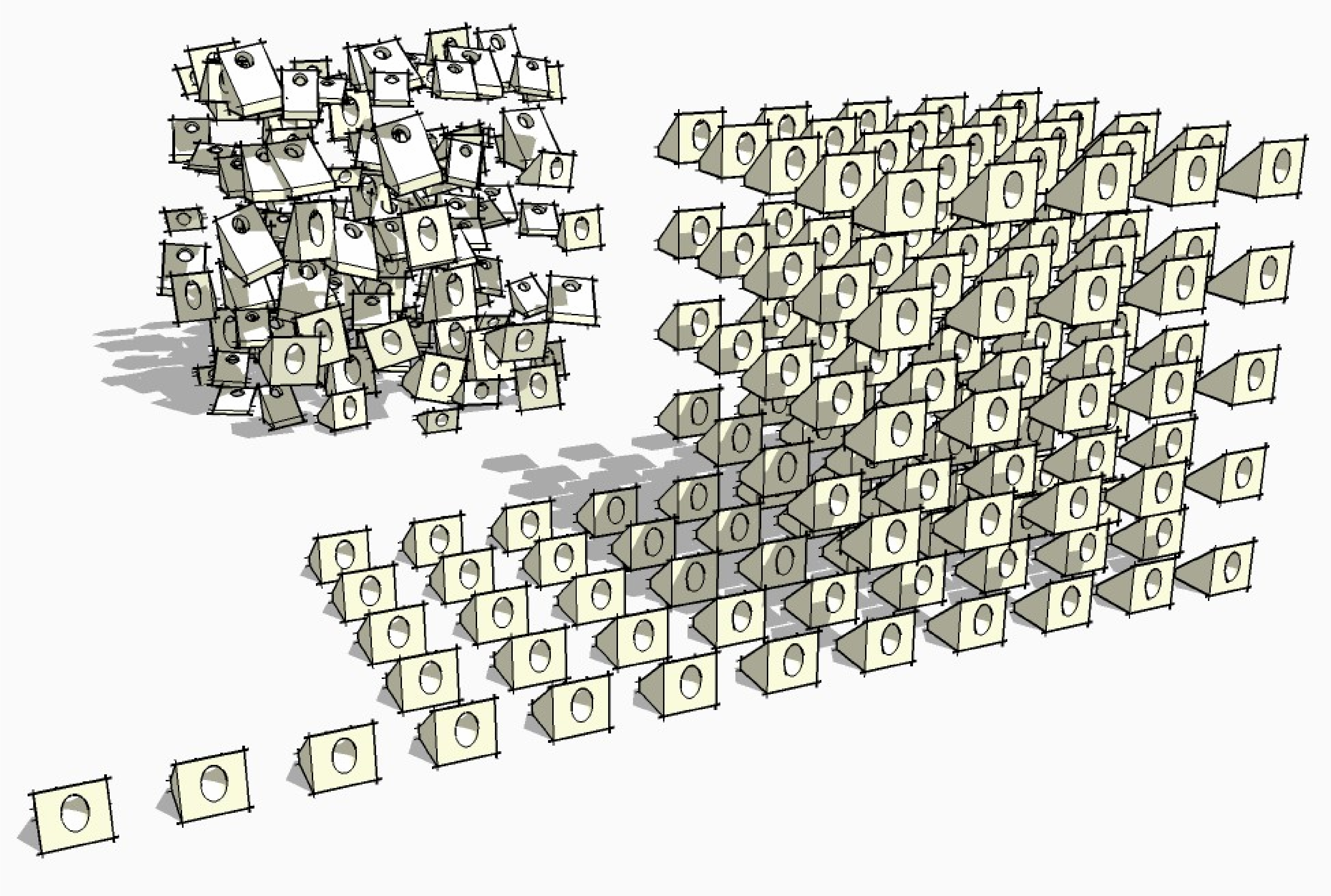
THE MATRIX-PROXIMITY GUIDE...



Contents:

Overview

Loading

Matrix

Proximity

Overview:

What is MatrixProximity ?

MatrixProximity.rb is a ruby script © TIG November 2005.

It is a script add-on to @Last's SketchUp, and works with all its versions that support ruby.

Note that the demo 'skp' files are saved in v4 format.

When MatrixProximity loads it creates two new tools that are then available from the Plugins menu.

They are called Matrix and Proximity.

Their functions are explained in detail on the following pages.

The current version of MatrixProximity is v1.2 - 26 November 2005.

Loading:

Before you start:

If you are reading this you've already managed to unzip the distribution file.

In the resulting main folder called MatrixProximity-v1.1 you will have found and presumably read the README.txt file and, as instructed, you are now reading this.   This main folder can go anywhere but putting it into the SketchUp folder seems logical… I have to assume that you know how to move or copy files on your computer…

What files and folders go where:

The main folder contains the following folders and files:

- MatrixProximity Demos (folder)
- Proximity Components (folder)
- MatrixProximity.rb (the script)
- MatrixProximityGuide.pdf (this !)
- README.txt (tells you to read this !)

Leave the MatrixProximity Demos folder and its contents alone for now – we'll look at it in later sections.
Move or Copy the Proximity Components folder and its contents into the SketchUp Components folder; again we'll need to look at this later.
Move or Copy MatrixProximity.rb into the SketchUp Plugins folder – this is so that it will load automatically when SketchUp starts.
Leave the MatrixProximityGuide.pdf and README.txt files where they are, for future use as reference…

You are done loading.

How to find the Matrix and Proximity tools:

The Matrix and Proximity tools are both available from the SketchUp Plugins pull-down menu.

Matrix:

What does Matrix do ?

Matrix makes multiple copies of a selected component to set rules. The matrix can be either one dimensional – i.e. a line in X or Y or Z, or two dimensional (the default in XY) i.e. a rectangular area of copies, or a three dimensional 'block' of copies. You set the XYZ spacings and number of copies. Additionally you can set XYZ randomized percentage adjustments of spacings, rotation and scaling.

Selecting a component

Firstly you must have in place (or have just placed down) a component for copying in a matrix.   If your intended matrix is in one plane (2D) then place the start one where you want the matrix to begin – e.g. on a flat face. If you want to eventually drop the matrix copies onto faces (e.g. aground surface) then place the first component sufficiently high up (in the Z) so that all matrix copies will be made above the faces onto which they will drop – otherwise there might be some failures – see below for details.

You must select now one instance of the component.

Starting the tool

After making the selection you pick the Plugins menu item called Matrix.

Possible first error dialog

If you have selected nothing at all, or selected something that isn't a component, or selected more than one component you will get an appropriate error message and you must re-select and select again…

First dialog

This main dialog asks a lot of questions, however it's quite straightforward. You can accept the default or previously entered values, or enter new ones as you wish

Note that the current model units are shown in any dimension requests (e.g. cm, m, feet, inches).

The first three items ask for the XYZ spacings – you enter the required spacing dimensions in each axis.
(Plain numbers without a unit suffix will usually suffice for all decimal units – however if you are in 'architectural' units (inches – ' ") and want to enter feet and inches, when entering 12'4" is taken as equivalent to entering 124 (inches)).
A typical matrix is in the XY 2D when Z stays as 0.
Use 2D for groups - trees, bushes, people etc.
Use 1D for lines - trees, queues etc.
Use 3D for blocks – usually special effects or massing.

The next three items ask for the XYZ number of copies – enter the numbers of copies in each axis.
A typical matrix is in the XY when Z stays 0.
Remember that matrix arrays multiply-up so a 9 copy matrix in 1D has 10 items (including the original component), whilst in 2D it'd become 10x10 = 100 or in 3D 10x10x10 = 1000; clearly if the component is complex your model will get very heavy very quickly. Matrix copy simple components or limit the numbers sensibly.
(Note that if either the Z spacing or the Z number of copies is set to 0 then both are taken as 0).

The next three items ask for XYZ randomized spacing %.
This allows some naturalness to creep into the matrix spacings. The range is "plus or minus" from the entered spacing – so for example if you enter 10m and a % range of 30 the spacing will randomize between 7m and 13m. Percentages over 50 might result in some copies 'clashing' – though of course you might want this effect…

The next three items ask for XYZ randomizing rotation %.
This also allows some naturalness in the matrix - rotated copies often look better. Setting the range at 100 gives all possible rotations, lower numbers will give more subtle results. The default is in the Z rotation only – this is best for most 2D matrix copies – any rotation in X or Y can give interesting but perhaps unexpected results.

The last three items ask for XYZ randomizing scaling %.
This also allows some naturalness in the matrix - modestly scaled copies can look better. Setting the range between 33 and 66 usually gives the best natural looking results. The default is for a very small scaling range (1) in X and Y (this also allows 50/50 'handing' of components) and the main range at 33 in the Z (height) most applicable to a 2D matrix. Unscaled axes don't 'hand' copies and there is never handing in the Z axis since it would flip half of the copies upside down below the Z plane – but if you really need this you can edit the script to allow it (advanced users see ### part). Scaling ranges set to over 100 can flip some copies upside down or handed.

When it's all set pick OK to continue or Cancel to abort.

The result

A matrix of copies in produced following the settings you have given.
Note that if there are any XYZ randomized settings that are not zero then these are applied to the original component as well as any matrix copies.

Second dialog

You are now asked if you would like to drop the matrix copies onto the nearest faces- typically a ground surface. Pick – Yes - to it, or – No - to skip this option.

The result

With Yes -the copies in the matrix drop vertically onto the surface faces below them.
With No - the copies stay where they are.

Possible second error dialog

If some copies fail to drop because they can't find a face below them, or perhaps they have been made below the surface of a hillside's faces, then you'll get an error message telling you how many didn't work. Usually the problem is self-evident – you can undo soon and move the start component and do it again to ensure all copies drop as required…

Possible third dialog

The matrix copied component might belong to a Proximity Component 'family' (one of the trio-sets – typically the 'Far' type – see later). If so then Matrix realises this and looks for related family components of the same type (e.g. Far) that are already made within or loaded into the model or are in the Proximity Components folder. If it finds them a dialog tells you how many alternatives are available for use and asks if you want to swap and randomly scatter them within the matrix copies (this is in approximately equal proportions). This is useful for introducing randomness and naturalness into groups of trees etc – use a Proximity Component family 'Far' tree to make the matrix initially (e.g. tree-001-) and if you have say three others available tree-002-, tree-003- and tree-004- then the result would be a randomized mix of the four types.

Answer – Yes - to do it, or No – to skip this option.

Note that if the original component used is not a Proximity Component or if it is one but it has no other family members then you will not get this dialog.

The result

With Yes -the copies in the matrix are swapped in approximately equal proportions with all available family members of the same type.
With No - the copies stay unchanged.

Undoing

The matrix copy undoes in a single undo - unless you have chosen the Proximity Component family swap option in the third dialog – in which case the first undo will revert to the unswapped matrix copies and a second will undo the matrix copying.

Making adjustments

If you are unhappy with some aspect of the matrix copies then simply undo – then either adjust the starting position of the original component and/or re-run Matrix and enter new values for the spacings, numbers and randomizations until the desired result is achieved.

Advanced playing around

Try combinations of 1D, 2D and 3D matrix copies with different spacings, numbers and various randomizations of spacings, rotations and scaling.

Note that if you set all parameters to zero in the dialog then no copies are made or randomized but you can still drop the selected component onto a face (like TBD's 'Drop.rb' script), and if it's from a Proximity Component family you can also take a chance on its randomized swapping.

Also try using several Proximity Component family members with Matrix for scattered, natural effects.

You can get some very interesting and sometimes surprising results with minor adjustments.

The MatrixProximity Demos folder and Proximity Components folder contain some examples to try out.

Publish your results and give feedback.

Proximity:

What does Proximity do ?

Proximity calculates the distance from the current camera position to every component in the model that belongs to a Proximity Component trio-set. It then asks for ranges of distance for Near and Mid field components and substitutes appropriate set-members, depending on how near to or far away they are from the camera. By doing this only complex versions of components are put near the viewer, whilst mid range ones can have less detail and those far away can be simple. This greatly reduces poly-count and heaviness in moving around a large model etc.

Proximity Component trio-sets

Proximity Components consist of three components – a trio-set.
They have the same main name – e.g. tree-001.
They also have a special suffix that tells Proximity it is that type of component and what type it is for substitution:

- Near +
- Mid ~
- Far -

So the trio-set is:

- tree-001+
- tree-001~
- tree-001-

Proximity Components created within a model should follow this convention; those kept for reuse in the Proximity Components folder will too, but of course they will also have the file suffix '.skp' – e.g. tree-001+.skp. Make your own trio-sets and save them – ensure that each version of the component is similar in form, size, origin/axes, colour etc, use simple 2D face-me versions for Far and perhaps Mid ones; remember to purge unused layers, materials and components from any '.skp' files you save into the Proximity Components folder.

Proximity Components can if appropriate also belong to 'families'. For example 'trees':

- tree-001+ tree-001~ tree-001-
- tree-002+ tree-002~ tree-002-
- tree-003+ tree-003~ tree-003-

It is the prefix separated by the "-" that in this case that tells Matrix which could be used in the randomized scattering option. Keep your families logical, e.g. tree-, bush-, pine-, conifer-, people-, car- and so on.

Starting the tool

Pick the Plugins menu item called Proximity.

The dialog

This dialog asks two questions, it's quite straightforward. You can accept the default or previously entered values, or enter new ones as you wish

Note that the current model units are shown in any dimension requests (e.g. cm, m, feet, inches).

The first box asks for the Near (+) distance range – 0 to ???, i.e. the distance over which the most complex of the Proximity Component trio is placed. Note that entering zero will limit the substitution to 'Mid' and 'Far' members of the trio-sets.

The second box asks for the Mid (~) distance range – 'Near' to ???, i.e. the distance over which the simpler of the Proximity Component trio is placed (beyond that the Far (-) version will be placed). Note that entering zero in the first box and this one will limit the substitution to 'Far' members of the trio-sets – this is useful to (re)make the model as simple as possible during the main modelling phase – saving the complex versions until views are to be taken.

Entering the same positive value (nnn) for Near and Mid will exclude the use of 'Mid' components and it'll use only 'Near' and 'Far' ones, within the range 0 to nnn and over.

Pick OK to continue or Cancel to abort.

Possible error dialog

If the entered value for Mid (~) is less than Near (+) you will get an error dialog telling you that Mid has been made the same as Near, however after doing this the processing continues.

The result

Every Proximity Component in the model is substituted with a trio version that suits its location relative to the camera. Near-to components are the most detailed types whilst the Mid and Far ones are increasingly simplified versions.

Undoing

Proximity undoes in one step.

Making adjustments

If you are not happy with the result then simply pick undo and re-run Proximity with different values for the distance ranges, so that more or fewer components swap around in the Near or Mid locations.

Advanced playing around

The MatrixProximity Demos folder and Proximity Components folder contain some examples to try out.

You can get some very interesting and sometimes surprising results with minor adjustments.

Publish your trio-sets, results and give feedback.

TIG