Grids MX

The ultimate solution for game and editor grid snapping.

Thank you for choosing Grids MX! If you have any questions, issues, or would like to suggest improvements, please send me an email at tools@maxwellplayed.com. I'd love to hear from you!

This documentation applies to Grids MX version 1.0.0.

Contents

Key Features

- Easy position and rotation **snapping** both **in editor and in-game**!
- Grid size customizable **per axis** with **over 10 measurement units**!
- Full support for **rotated grids**!
- Extremely customizable **scene view grid display**!
- **Selection reference lines** make it easier than ever to snap objects right where you want them!
- Visibility, size, snapping, and more customizable **per axis**!
- Support for grid **wrapping, clamping**, and infinite snapping!
- Setup **multiple grids** and quickly switch between them!
- Massive scene view grids supported with **no performance loss**!
- Quickly move grids around the scene with **grid move, rotate, and follow** actions!
- **Customizable toolbar** with Horizontal and Vertical layouts, and button section hiding!
- **Customizable hotkeys** for common actions!

Getting Started

Want to get started right away? There are two things to know:

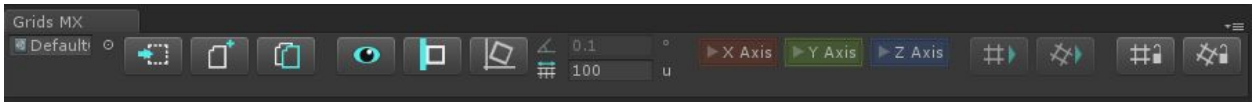
1. Toolbar
2. Settings

The Toolbar is the main interface of Grids MX; open it via the **Window -> Grids MX Toolbar** menu option. These are your moment-to-moment usage tools.

The Settings file can be accessed via the **Edit -> Preferences** menu option, then choose the **Grids MX** tab. These settings allow you to customize many parts of Grids MX to your personal preferences.

It's that easy! From one of these two windows you can change the vast majority of things you'll need to use Grids MX.

The Toolbar



You can open the toolbar via the **Window -> Grids MX Toolbar** option.

Note: *The in-scene grid will only display while the toolbar is open!*

The toolbar has the primary options you'll be using to interact with Grids in the editor. Note that you can choose between vertical and horizontal layouts, as well as hide sections of buttons in the Grid Settings.

You can hover over any button or field to get additional info and hotkeys via **tooltips**.

The first few buttons relate to Using Multiple Grids, see that section for more information.

Global Options

Left to Right:

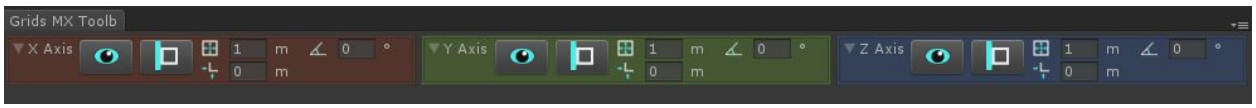
- Toggle scene view grid visibility
- Toggle position snapping
- Toggle rotation snapping

- Rotation snap angle
- Grid size / display distance

These are fairly straightforward. They override axis-specific options. E.g. if global visibility is **off**, no grid will be displayed even if the axis visibility is set to **on**. Additionally, position snapping requires the grid to be **visible**.

The **grid size/display distance** field determines how many cells will be displayed per axis. If the grid's **edge type** is set to **Infinite**, it's effectively the display distance. Otherwise, it will determine where the edge of the grid exists for the important features of **edge type** (See more on edge type here).

Axis Options



Each axis represents the plane of the grid that is perpendicular to that axis. i.e. the 'X Axis' refers to the plane that lies across the Y and Z axes.

Left to Right (per axis):

- Toggle plane visibility
- Toggle plane position snapping
- Cell size
- Grid offset
- Plane rotation

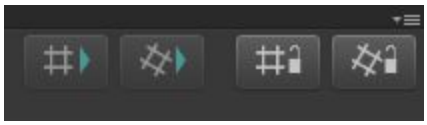
Cell size is the size of cells *along* the given axis, meaning that changing the X cell size will only show up on the Y and Z planes.

Grid offset is the distance that the grid Origin Point is offset along the given axis. Using these you can move the entire grid anywhere in the world. You can use the Grid Move and Follow buttons to quickly position the grid.

Both cell size and grid offset use the global **measurement unit** set in the Grid Settings.

Plane rotation will rotate the entire grid along the given axis in world space. All position snapping *obeys the grid rotation*, allowing you to set up grids at whatever angle you need, and still have everything perfectly aligned! You can use the Grid Move and Follow buttons to quickly rotate the grid.

Grid Move and Follow



The left two of these buttons will snap the entire grid to the position or rotation**,** respectively, of the active selected transform. This changes the **grid offset** and **plane rotation** of each axis.

The right two buttons are similar, except that they are **toggles** that lock the grid to follow the current selection.

These options are particularly helpful if you're exclusively using Grids MX for the in-editor snapping capabilities and don't care so much where the grids are positioned.

Additional Grid Options

There are additional grid options on the GridData object itself that are changed so infrequently they didn't warrant a place on the main toolbar. You can find these by selecting the GridData object in the project view. See Using Multiple Grids for more info on where to find them.

Edge Type

Edge type determines what happens when moving an object beyond the edge of the grid. There are four options:

- **Infinite**: Grid snapping continues after edge of visible grid.
- **Free**: Grid snapping stops after edge of grid.
- **Clamp**: Movement is clamped at grid edge, objects cannot be moved outside of it.
- **Wrap**: Objects moved outside of the grid wrap around to the other side of the grid.

If you only use Grids MX for the snapping features I suggest you leave this on **Infinite**.

Origin Point

All grid coordinates and gridspace positions are calculated from this position, except for the location of the origin point itself. The origin point is measured with the assumptions that **0** is the **negative-most** grid coordinate along an axis, and the **positive-most** axis coordinate is the size of the grid. This means that for an origin point of **(0, 0, 0)**, every coordinate that is on the grid will be positive. If you want the origin in the center of a grid, it should be at **gridsize/2** for each axis.

I recommend you set this once and leave it alone, as changing the origin point can have wide ranging effects since everything else is based on it.

Additionally it only really matters if you're using grid coordinates in-game or for some other purpose. If not, I recommend setting it to the center of the grid so the visuals behave comfortably.

[Extra info] The scene grid display uses the origin point to determine which section of the grid should be displayed in their full glory. If each plane is always displayed and the origin point is not at an extreme, it gets visually confusing very fast! My solution is to have each section of the grid 'focusable' based on where your scene camera is, preventing multiple sections from displaying over top of each other.

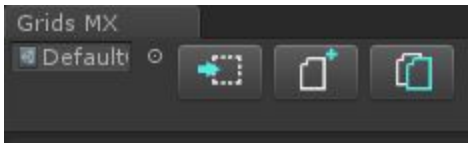
If you want to see the full extents of the grid despite the section-focusing, you can turn on the 'outline' unfocused style option in the Settings. This is a nice alternative which shows the edges of the grid without all the visual clutter of the grid overlapping itself!

Using Multiple Grids

In Grids MX, each grid is a bit of data managed by a 'GridData' scriptable object. This means that it's a special asset file that exists in your project. You can have as many grids as you want. Changes in the toolbar are saved on the GridData object itself. You can even move them between projects as desired.

This feature allows you to setup multiple different Grids for different purposes. As an example suppose you wanted all 'buildings' to snap a 10x10 unit grid, but wanted 'people' to move on a 1x1 grid. No problem!

On the Toolbar there is a set of quick options for common data management actions.



Left to Right:

- Selected Grid
- Select Grid in Project View
- Create New Grid
- Duplicate Grid

The **Selected Grid** field shows the currently selected GridData. You can select a different Grid simply by selecting its GridData object in the project view. The first button opens a path to the current GridData in the project view.

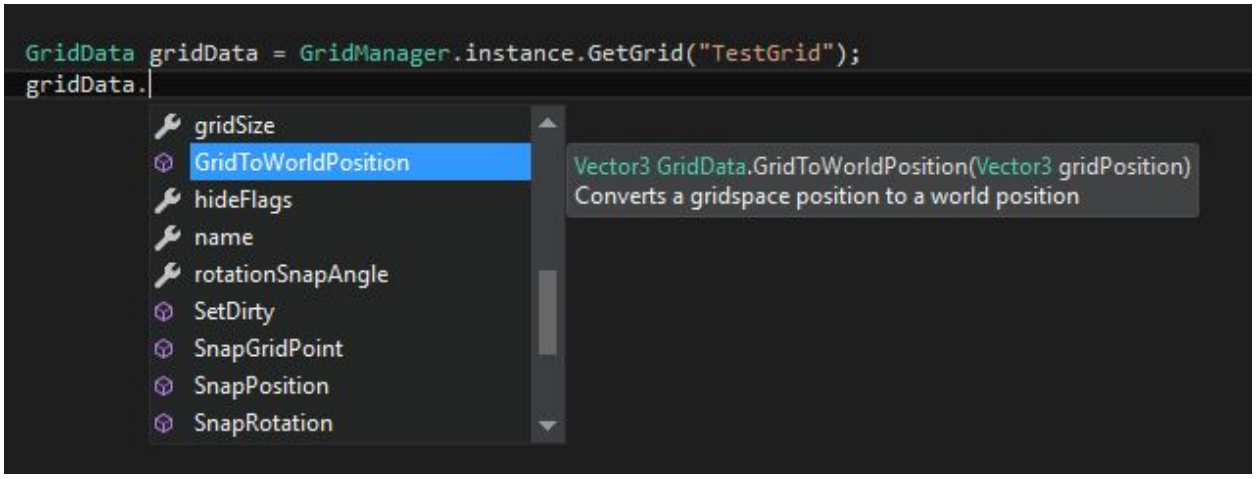
The right two buttons create a new GridData with default settings, and duplicate the currently selected GridData, respectively. Make sure to change the name to something useful afterwards!

Resources Folder

GridData files must be stored in a folder with the resources path: **Resources/GridsMXData**. It is unimportant where these folders are located, just that they have this path. You'll see in the asset package that the default grid data is stored in the Grids MX/Resources/GridsMXData folder, but there are also more GridData objects in their own folders in the examples.

Using Grids In-game

All Grids are accessible in-game using the **GridManager** class, and retain the ability to easily snap to grid coordinates. You can do all of your gameplay logic using the integer-based 'GridPoint' class and only need to convert to unity position at the end of your calculations!



To get started, call **GridManager.instance.GetGrid(id)** where **id** is the name of the GridData object in your project. This will create a runtime GridManager object and load in the GridData from Resources as necessary. There are additional functions on GridManager to handle loading/unloading in a more granular manner.

The **GridData** class has all the snapping and position-manipulating functions you'll need to work with Grids in-game.

The "**In-game Example**" in the Examples folder shows a simple example of using two different grids to move the same object with snapping at different intervals.

Gridspace Position and GridPoints

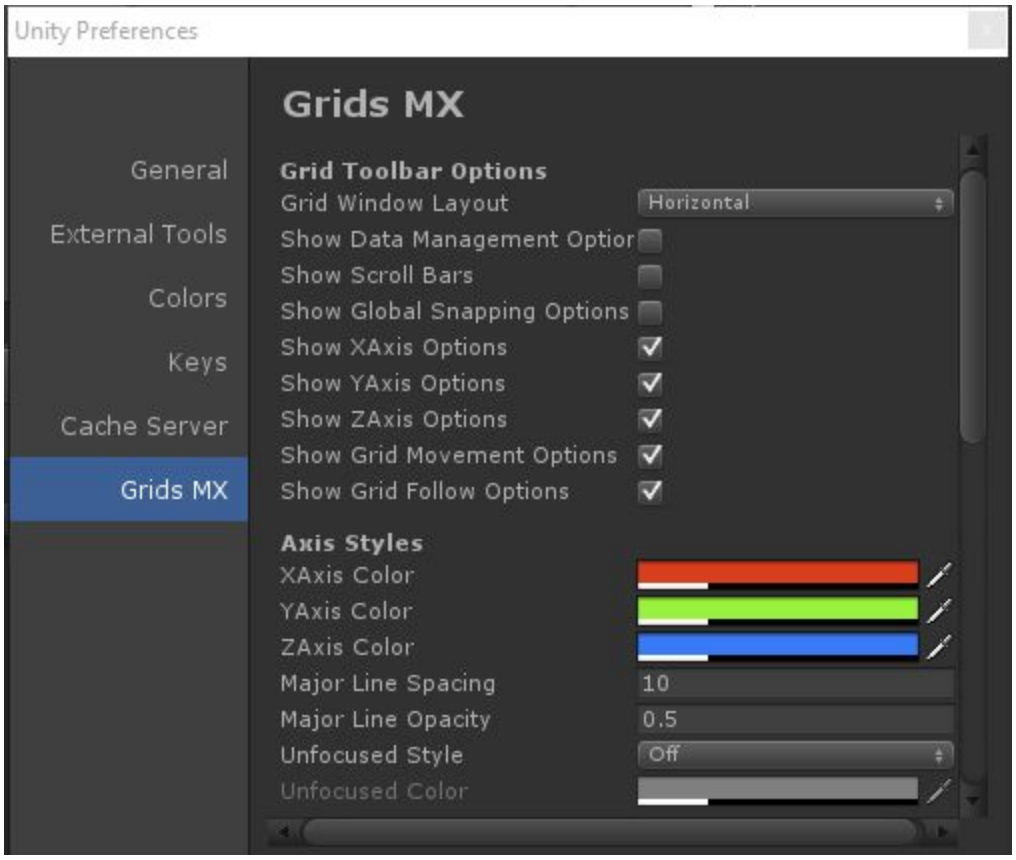
The most important thing to note is that the Grid has its own view of the world, referred to as 'gridspace'. This is equivalent to any other hierarchy-based system, including Unity itself. When using

GridPoints or positions in gridspace, keep in mind it is all **relative to the grid's origin, rotation, and offset.**

GridPoint is an integer-based struct similar to Unity's Vector3. It allows precision logic around Grid coordinates and for this reason many of the GridData functions use this as well as more granular Vector3 for gridspace positions.

Settings and Customization

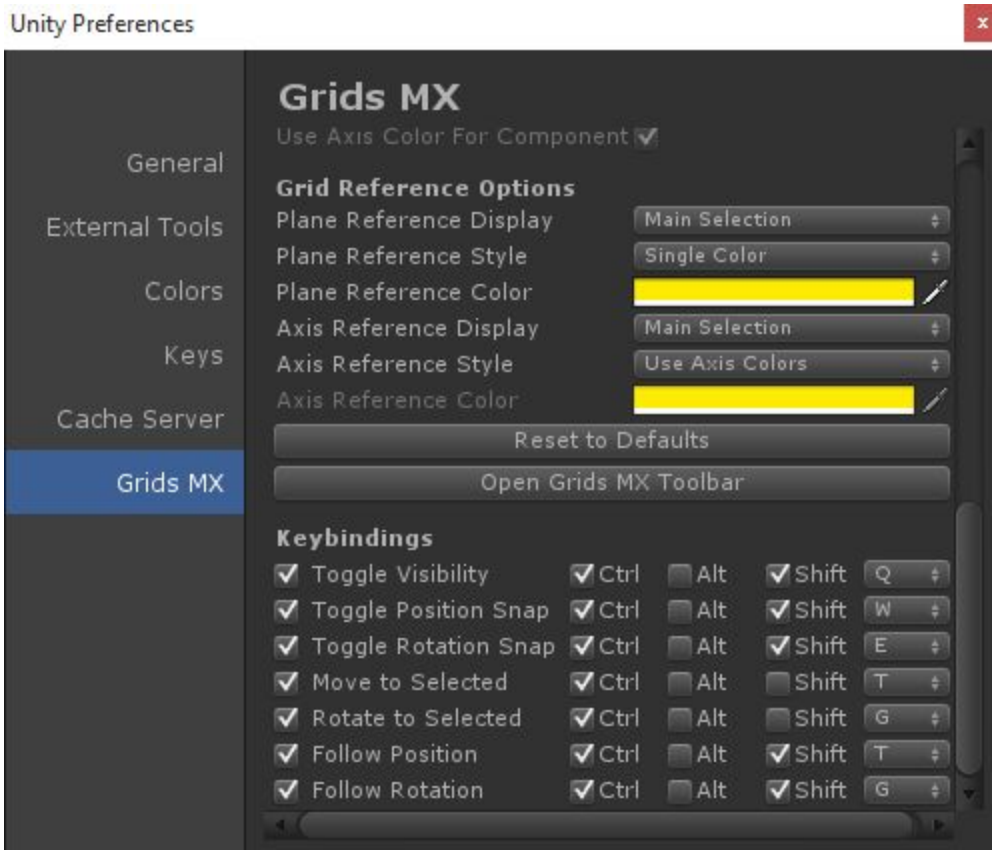
There are a wide array of settings and customization options for you in Grids MX. You can access them through the **Edit -> Preferences** menu, and then choose the **Grids MX** tab.



There are quite a few options here! Most should be fairly self explanatory, and there are additional tooltips on most options for additional information.

Keybindings

Keybindings can be accessed through the **Edit -> Preferences** menu, then choose the **Grids MX** tab and scroll to the bottom.



These keybindings relate directly to the buttons on the the Toolbar. **You may need to re-open the Toolbar after changing keybindings for them to take effect**. The button tooltips will also be updated to display your new settings.

Moving Grids and Settings Between Projects

You can find your settings in a file called **GridSettings,** and your keybindings in a file called **Keybindings**, both of which are located in the folder **Grids MX/Resources/GridsMXData**.

It is perfectly safe to move, GridSettings, Keybindings, and GridData files across Unity Versions and to new projects. Just make sure Grids MX is imported in your new project before doing so! You can overwrite the default files with the ones you bring over.

Support

Thank you for choosing Grids MX!
If you have any questions, issues, or suggestions, please send me an email at tools@maxwellplayed.com. I'd love to hear from you!

You can find this information and more at http://maxwellplayed.com/tools/gridsmx