

Edge Display Scaler Reference Guide

by ThinkBoxly Docs

1. Introduction	3
2. Changelog	3
3. Migrating Versions	7
4. Reference Guide	8
4.1. edgeds_init_scale	9
4.2. edgeds_set_scale	10
4.3. edgeds_set_width	10
4.4. edgeds_set_height	11
4.5. edgeds_set_screenres	11
4.6. edgeds_draw_background	11
4.7. edgeds_draw_sprite	12
4.8. edgeds_draw_view	12
4.9. move_link	13
5. EULA	13

1. Welcome to Edge Display Scaler



Homepage: engine.thinkboxly.com(<http://engine.thinkboxly.com/>)

Support/inquiries: support@thinkboxly.com(<mailto:support@thinkboxly.com>)

Thank you for choosing Display Scaler, the first published member of the Edge Engine family. **Edge Engine** is a fully cross-platform, modular framework built to augment Game Maker Studio with pre-made code and assets that serve as the foundation for a wide variety of game genres. All Edge Engine modules feature creative, human-readable code with helpful notations throughout, making them both powerful and easy to use.

Edge Display Scaler (or Edge DS) is a lightweight scaling solution to run your Game Maker Studio projects in any resolution, at any aspect ratio and respond intelligently to changes in resolution in real time. In today's market of mobile and multiplatform games a good scaling solution is a must. Regardless of what platform you're building on, Edge Display Scaler has you covered...and all it takes to get started is a single object.

In this guide you will learn about the multiple scaling methods Edge Display Scaler offers and how to utilize them in your projects to achieve the desired results. You will also learn concepts such as relative scaling and positioning—core programming techniques required to take a scaled viewport and apply the same scaling to your own sprites and objects. Remember, no amount of scaling will automatically make a project *look good* when scaled—that part is up to you.

2. What's New

v1.6.7

- Updated documentation to new format

v1.6.6

- Fixed a bug causing backgrounds to not scale properly across rooms

v1.6.5

- Improved GUI layer scaling
- Added aspect ratio window resizing for `edgeds_set_scale`
- Improved aspect ratio window resizing for `edgeds_set_screenres`
- Improved performance

v1.6.1

- Fixed a regression from previous versions causing performance to drop as a result of DPI calculations

v1.6.0

- Fixed a bug causing inaccurate scaling on HTML5 platforms when using `edgeds_set_width` or `edgeds_set_height`

v1.5.9

- GUI layer is now automatically scaled along with the application layer

v1.5.8

- Spring cleaning—removed non-view scaling as this feature was deemed useless and unnecessary
- Removed mobile hi-res workaround, as >2K screens are now supported by GameMaker natively (thanks, YoYo!)
- Added aspect ratio window resizing for `edgeds_set_width`, `edgeds_set_height`, and `edgeds_set_screenres`
- Improved, more accurate handling of DPI-based scaling
- Improved, more accurate scale linking. No longer experimental!
- Minor additional improvements and optimizations

v1.5.3

- Added auto mode to view zooming, allowing to minimize the differences in viewport pixel dimensions between resolutions
- Fixed issues in asset scaling scripts causing inaccurate scaling while following an object with the master view

v1.5.1

- Added optional view zooming to all `edgeds_set_*` scripts

v1.5.0

- Added optional minimum resolutions to applicable `edgeds_set_*` scripts
- Added DPI adjustment warning to built-in stats
- Syntactical adjustments for consistency (see 'Migrating Versions')
- Merged `edgeds_get_prevscale` with other scripts; `edgeds_get_prevscale` no longer necessary
- Fixed an issue preventing scaling from carrying over across rooms
- Additional fixes and improvements

v1.4.1

- Minor bug fixes

v1.4.0

- Complete overhaul! Majority of code re-edited, re-written, simplified, and/or improved
- All-new scripts for scaling backgrounds, sprites, and views
- Can now scale any view, or disable view scaling entirely (see documentation)
- Added optional DPI scaling to all `edgeds_set_*` scripts
- Added DPI override setting to force a given DPI on any display

v1.2.8

- Improved support for >2k displays on devices not supporting >2k texture pages
- Improved DPI handling in DPI scaling mode
- Performance improvements
- Minor bug fixes and general improvements

v1.2.6

- Added support for >2k displays on Android and iOS devices (other platforms already supported)
- Added optional DPI mode to `edgeds_set_scale`
- Added `edgeds_scale_width` and `edgeds_scale_height` scripts for single-axis scaling
- General improvements

v1.2.3

- Fixed an issue causing crashes when the game window is minimized

v1.2.2

- Updated `edgeds_view_fit` script to utilize two additional views as letterboxes

v1.2.1

- Added `edgeds_set_screenres` script

v1.2.0

- New naming structure for Edge Display Scaler scripts, for uniformity across Edge Engine modules
- Included 'Legacy Scripts.zip' to ease transition from previous versions
- HTML5 users no longer need to manually edit a script to enable scaling on HTML5 platforms
- Added `edgeds_background_fill` script for scaling backgrounds to fill the view
- Added `edgeds_view_fit` script for scaling additional views to fit within view 0, the 'master view'
- General improvements

v1.1.2

- Fixed an issue with HTML5 scaling

v1.1.1

- Added syntax guides

v1.1

- Rebranded as Edge Engine Display Scaler
- Completely rewritten documentation
- New tools: `image_scale_abs` and optional scale linking in `move_link` (experimental; disabled by default)
- General improvements

v1.0

- Initial release as Simple Display Scaler

3. Migrating to New Versions of Edge Display Scaler

As Edge Display Scaler matures and develops, old features sometimes become incompatible with the latest updates. This does not mean projects built on older versions of Edge DS cannot be upgraded, but it does mean the process is not automatic and will require some effort on the part of the developer.

Note that the latest version of Edge DS should already be installed in your project before following this guide. To install Edge DS on top of older versions, remove Edge Display Scaler from your project's 'Extensions' folder and delete the entire 'Edge Display Scaler' folder from your 'Scripts' folder. Also delete any Edge DS demo objects from your 'Objects' folder, leaving only your own custom objects and code behind. To update these custom project assets, refer to the migration guide below.

- **Scaling scripts will now automatically scale the GUI layer along with the application layer.**

This means it is no longer necessary to run the `display_set_gui_maximize()` command manually, and any existing instances of this function should be removed when using Edge Display Scaler.

- **`edgeds_init_scale` has been reduced to a single argument for view scaling.**

As non-view scaling was determined to be an unnecessary feature it has been removed and as such it is no longer possible to enable or disable view scaling—it is enabled by default. As a result, all instances of `edgeds_init_scale` must be updated to include only one argument setting the view to scale. See documentation for further details on the updated syntax.

- **`edgeds_set_scale`, `edgeds_set_screenres`, `edgeds_set_width`, and `edgeds_set_height` have all been given optional DPI scaling support with optional DPI override, minimum resolution, and zoom support.**

All existing instances of these scripts must be updated to either enable or disable DPI scaling and set either auto or manual DPI, as well as set the minimum scaling resolution and zoom multiplier. See documentation for further details on the updated syntax.

- `move_link` now features both movement and scale linking which can be toggled independently.

This means there are now three parameters to the script and existing instances will need to be updated to include these new values. See documentation for further details on the updated syntax.

4. Overview

Using Edge Display Scaler comes in two stages: 1) Scaling the game itself, and 2) programming your objects to *respond* to that scaling by adjusting themselves accordingly (“responsive design”). Thankfully, all you need to get started is the first step, and with Edge Display Scaler, it’s practically done for you.

First, open your project’s `Global Game Settings` page, navigate to the tab for your desired platform, and under the `Graphics` settings set the `Scaling` option to `Full Scale` or `Stretch`. On platforms that support it, you should also ensure the `Interpolate Colors Between Pixels` box is checked.

Next, create or open the first room of your game project and switch to the `Views` tab. Ensure the `Enable Use of Views` box is checked, and set the parameters for the view of your choice according to the desired base resolution for your game. Whatever *port* width/height are used will act as your base resolution—the point at which the game is operating at a 1:1 scale.

Note: Whichever view you use must be reserved in every room as the ‘master’ scaling view. Edge Display Scaler will enable this view automatically if it is not already activated, but keep in mind that it is not available for normal use. To use views for your own purposes, only use the other seven available views instead.

Note 2: Not sure what the difference between ‘view’ and ‘port’ is? It’s easy: think of the *view* as game pixels and the *port* as screen pixels. The two don’t have to be equal, but in the end, ‘port’ always rules. No matter how much or little of the game view is drawn, it will always be stretched to fit the port. Likewise, if your game is not running in fullscreen, the game window will always be the size of the largest active port.

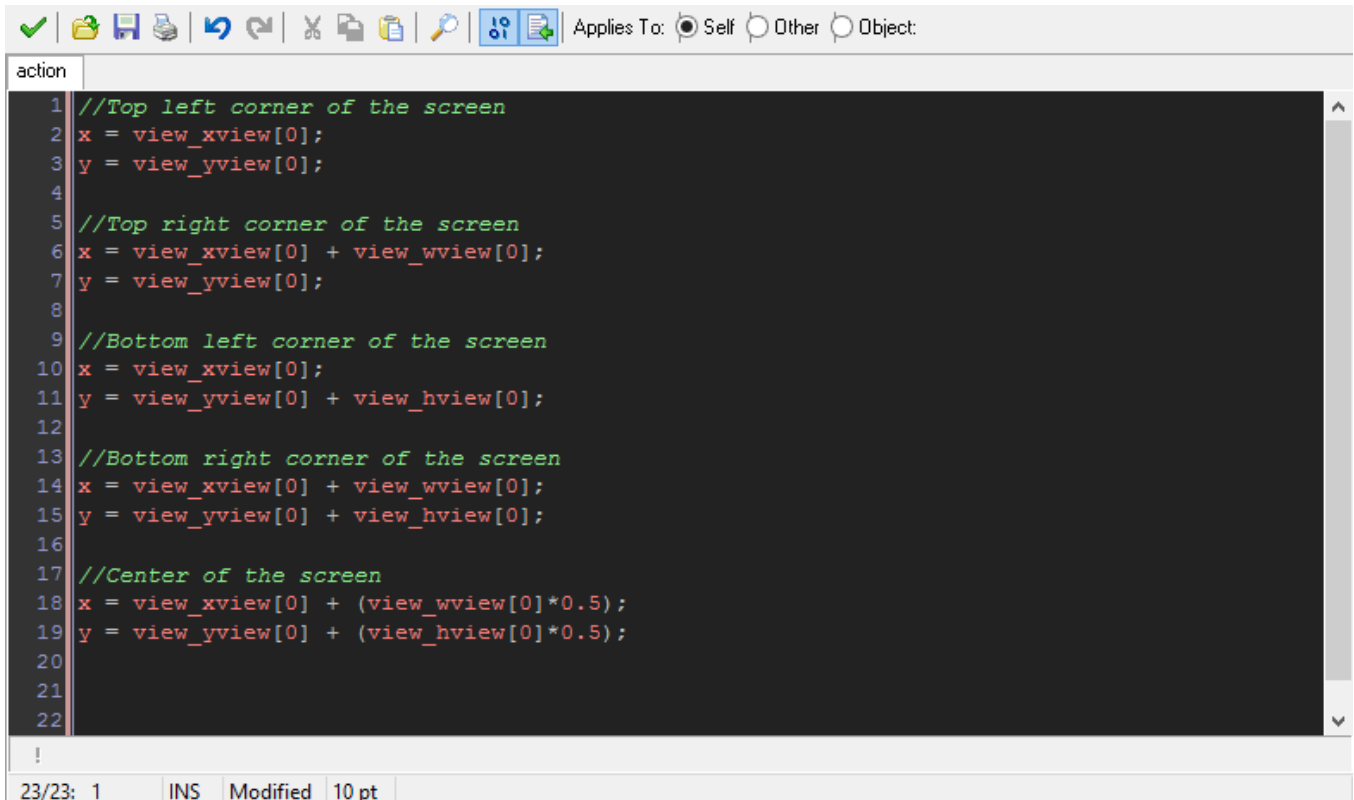
Finally, drop the included `obj_scaler` object into the room. It can go anywhere you like and is not visible by default. This object is also persistent, meaning you only need to include it once at the beginning of your game to achieve perfect full-display scaling entirely throughout.

Note: Enabling visibility of the `obj_scaler` object will draw a variety of debug statistics in the top left corner of the screen, from FPS to display size to DPI to orientation!

And that’s it! The most important step is now done, but now’s when the real work begins. Neither this nor any other guide can tell you precisely how to use the power of Edge Display Scaler. That depends on you and your game. What this guide can do, however, is educate you on the tools at your disposal so you can go about using them effectively for yourself.

The first of these tools is actually not a part of Edge Display Scaler at all, nevertheless it is fundamental in understanding responsive design. That tool is **procedural positioning**. Normally, to position an object you simply drag and drop it into a room editor, but to use display scaling effectively you must learn new, programmatic positioning methods. Some objects, such as menu or GUI elements, must remain on-screen at all times, even if the size and shape of that screen changes either in real-time or just from device to device. To accomplish this, objects must be positioned relative to the view with a little code.

Take a look at the following example:



```

1 //Top left corner of the screen
2 x = view_xview[0];
3 y = view_yview[0];
4
5 //Top right corner of the screen
6 x = view_xview[0] + view_wview[0];
7 y = view_yview[0];
8
9 //Bottom left corner of the screen
10 x = view_xview[0];
11 y = view_yview[0] + view_hview[0];
12
13 //Bottom right corner of the screen
14 x = view_xview[0] + view_wview[0];
15 y = view_yview[0] + view_hview[0];
16
17 //Center of the screen
18 x = view_xview[0] + (view_wview[0]*0.5);
19 y = view_yview[0] + (view_hview[0]*0.5);
20
21
22
!
23/23: 1 INS Modified 10 pt

```

Assuming View 0 is set as your master scaling view, `view_wview[0]` and `view_hview[0]` will always refer to the full width and height in pixels of the current display—be it a window, a fullscreen application, or running in a browser with HTML5. Use these and other measurements to position your objects relatively, rather than absolutely. Think of your coordinates as *percentages* of the game window, rather than absolute pixels.

With this basic setup your project will scale itself according to the default settings for Edge Display Scaler, however you will most likely wish to customize them to best suit your specific project. To learn how, refer to the functions in the rest of this guide.

4.1. `edgeds_init_scale(view)`

```
edgeds_init_scale(0);
```

Run once at the creation of `obj_scaler`. Sets the base resolution for the game and defines all the main variables needed for display scaling. The **view** parameter specifies which view (0-7) to scale and retrieve the base resolution from. Note that base resolution is retrieved from the view's *port* as set in the room settings of the room containing the scaler object.

Note that `edgeds_init_scale` initializes a number of global variables that can be referenced elsewhere for various display scaling purposes. These variables are:

- `global.display_view` (refers to the master scaling view)
- `global.display_wres` (the base horizontal resolution for the game)
- `global.display_hres` (the base vertical resolution for the game)
- `global.display_xprevscale` (refers to the previous scale of the display, relative to the base resolution)
- `global.display_yprevscale` (refers to the previous scale of the display, relative to the base resolution)

resolution)

- `global.display_xscale` (refers to the current scale of the display, relative to the base resolution)
- `global.display_yscale` (refers to the current scale of the display, relative to the base resolution)
- `global.display_orientation` (refers to whether the game is running in portrait or landscape)
- `global.display_zoom` (refers to the current zoom multiplier of the display, where 1 = 100%)
- `global.display_dpi` (refers to the actual DPI or PPI of the display)
- `global.display_dpi_current` (refers to the virtual/override DPI or PPI of the display)

Use these variables in your own code in any object to make the most of what Edge Display Scaler offers.

4.2. `edgeds_set_scale(dpi mode, dpi override, minwidth, minheight, zoom)`

```
edgeds_set_scale(false, 0, 640, 360, 1);
```

Calculates the current display scale and orientation and re-scales the game to match.

As of version 1.2.6, `edgeds_set_scale` also comes with the option to enable a DPI (or PPI) scaling mode that will attempt to adjust the scale of the application based on the *physical* size of the display, not just the actual number of pixels. This is especially important for mobile devices, such as high-end Android smartphones, which can have super high-density displays in very small sizes. While DPI scaling is very useful, be aware that it is also exponentially more complex to work with, and relying on it will require entirely pixel-agnostic programming to position elements where you want. Think of DPI mode as scaling in *percentages* and standard mode as scaling in *pixels*. To enable DPI mode, set the **dpi mode** parameter to 'true'. Otherwise set it to 'false' for standard scaling (recommended).

As of version 1.4.0, Edge Display Scaler also permits DPI override, which will force the display to scale to the DPI value specified in the **dpi override** parameter. This feature should be used carefully, and is generally recommended to leave disabled by setting the **dpi override** parameter to 0.

The **minwidth** and **minheight** parameters set a minimum resolution limit in pixels to prevent the game from scaling below a desirable size. For no minimum, set both of these values to 0.

As of version 1.5.1 scaled views can also be zoomed with the **zoom** parameter. This value is a multiplier where a value of 1 is default, or 100% of the view dimensions plus any DPI adjustments that may be enabled. Larger values equals greater zoom factor. A value of 0 or -1 will enable auto zoom, which will attempt to compensate for differences in resolution so that the view always displays approximately the same in-game area. Note that this area will inevitably still vary somewhat based on aspect ratio.

This script should **not** be run simultaneously with any other `edgeds_set_*` script, and ideally **should** be run in the Step event.

4.3. `edgeds_set_width(dpi mode, dpi override, width, minheight, zoom)`

```
edgeds_set_width(false, 0, 1920, 360, 1);
```

Functions as an alternative to `edgeds_set_scale`.

This script will force an absolute pixel **width** while scaling height dynamically to fill the display by aspect ratio. This can be handy as an easier and more predictable alternative to other scaling methods, though sometimes more limited in its scope.

As usual, the **minheight** parameter still sets a minimum vertical resolution limit in pixels to prevent the game from scaling below a desirable size. For no minimum, set this value to 0.

This script should **not** be run simultaneously with any other `edgeds_set_*` script, and ideally **should** be run in the Step event.

As of version 1.4.0, this script also features optional DPI mode and DPI override settings.

As of version 1.5.1 this script also features optional view zooming. See documentation for `edgeds_set_scale` for usage details.

4.4. `edgeds_set_height(dpi mode, dpi override, minwidth, height, zoom)`

```
edgeds_set_height(false, 0, 640, 1080, 1);
```

Functions as an alternative to `edgeds_set_scale`.

This script will force an absolute pixel **height** while scaling width dynamically to fill the display by aspect ratio. This can be handy as an easier and more predictable alternative to other scaling methods, though sometimes more limited in its scope.

As usual, the **minwidth** parameter still sets a minimum horizontal resolution limit in pixels to prevent the game from scaling below a desirable size. For no minimum, set this value to 0.

This script should **not** be run simultaneously with any other `edgeds_set_*` script, and ideally **should** be run in the Step event.

As of version 1.4.0, this script also features optional DPI mode and DPI override settings.

As of version 1.5.1 this script also features optional view zooming. See `edgeds_set_scale` for usage details.

4.5. `edgeds_set_screenres(dpi mode, dpi override, width, height, zoom)`

```
edgeds_set_screenres(false, 0, 1920, 1080, 1);
```

Functions as an alternative to `edgeds_set_scale`.

Rather than give the user free reign to scale the display, `edgeds_set_screenres` will force the display to conform to the specified **width** and **height** dimensions.

This script should **not** be run simultaneously with any other `edgeds_set_*` script, and ideally **should not** be run in the Step event, but rather triggered just once (e.g. in the Create event).

As of version 1.4.0, this script also features optional DPI mode and DPI override settings.

As of version 1.5.1 this script also features optional view zooming. See `edgeds_set_scale` for usage details.

4.6. `edgeds_draw_background(background, x, y, index, foreground, tile, scale mode)`

```
edgeds_draw_background(my_bg, view_xview[0], view_yview[0], 0, false, false, 0);
```

Assigns the background asset specified in the **background** parameter to the background slot specified in the **index** parameter and scales it to fill the 100% of the display with one of three scaling modes.

By setting the **scale mode** parameter to 0, the background will be scaled on both axes to always fill the display in either direction. By setting it to 1 or 2 it will scale only on the x or y axis, respectively.

Because the background is intended to fill the display, it is recommended to set the **x** and **y** parameters set to 0 or `view_xview[0]` and `view_yview[0]`, where `[0]` refers to the desired scaling view, however the background may still be positioned freely regardless of scaling.

By setting the **foreground** parameter to `true` the background can appear on top of game assets rather than behind, otherwise `false` will display the background normally.

Finally, by setting the **tile** parameter to `true` the background will repeat in both directions, which can be useful when using manual x and y offsets. Otherwise, `false` will draw the background only once.

This script can be run in virtually any event.

4.7. `edgeds_draw_sprite(sprite, index, x, y, rot, color, alpha, scale mode, notme)`

```
edgeds_draw_sprite(my_spr, image_index, view_xview[0], view_yview[0], 0, c_white, 1, 4, true);
```

Draws the sprite asset specified in the **sprite** parameter with the scaling mode specified in the **scale mode** parameter.

As of version 1.4.0, five sprite scaling modes are available in Edge Display Scaler: x and y full scaling (0), x-only scaling (1), y-only scaling (2), fit scaling (3), and proportion scaling (4). Mode 0 scales a sprite essentially as a fullscreen background, with modes 1 and 2 offering single-axis versions of the same style of scaling. Mode 3 only scales a sprite downward as necessary to fit small displays while never exceeding 100% scale on larger ones, and Mode 4 scales proportionately to the display whether larger or smaller than the base resolution.

It is also possible to draw sprites with a degree of customization. The **index** parameter specifies the frame of the sprite to draw (use -1 or `image_index` for the current frame in an animation), and the **x** and **y** parameters specify where to draw the given sprite. The **rot** parameter sets the sprite's rotation in degrees, ranging from 0-360. The **color** parameter allows for blending the sprite with different colors, such as `c_blue` for blue, or `c_white` for default. The **alpha** parameter sets the transparency of the drawn sprite as a fractional range from 0-1.

Lastly, the **notme** parameter sets whether to apply the sprite settings to the running object's assigned sprite. If this value is set to `false`, the sprite specified in this script will be assigned to the running object and any scaling performed on the drawn sprite will be applied directly to the object as well. Conversely, by setting this parameter to `true` only the drawn sprite will be affected (standard behavior as compared to Game Maker Studio's `draw_sprite` scripts).

4.8. `edgeds_draw_view(view, x, y, rot, color, alpha, scale mode)`

```
edgeds_draw_view(1, view_xview[0], view_yview[0], 0, c_white, 1, 0);
```

Draws the contents of the view specified in the **view** index parameter as a surface with optional scaling applied.

The **scale mode** here is either 0 for fullscreen aspect-ratio scaling or any positive number for absolute scaling. For example, a value of 2 will draw the view at twice the original size. This parameter can also be set to -1 for no scaling at all, which can be useful simply to take advantage of the script's other features.

The **x** and **y** parameters set where the view will be drawn, but keep in mind that when the **scale mode** is set to 0 this value will be overridden and the view will be centered first before applying any additional x and y values. Therefore, when full scaling is enabled, **x** and **y** should both be set to 0 or `view_xview[0]` and `view_yview[0]`. The **rot** parameter sets the drawn view's rotation in degrees, ranging from 0-360. The **color** parameter allows for blending the drawn view with different colors, such as `c_blue` for blue, or `c_white` for default. The **alpha** parameter sets the transparency of the drawn view as a fractional range from 0-1.

Note that drawing views this way is only intended for scaling small views *up*, **not** scaling large views *down*, as the latter will result in missing visual data. By scaling small views up, low-resolution games can run on high-resolution displays with different aspect ratios without pixel distortion. However, the practical usefulness of this functionality is limited, and `edgeds_draw_view` primarily exists as a quick and dirty way to convert unscaled games to scaled. For regular usage it is strongly recommended not to rely on this script.

4.9. `move_link(child object, pos link, scale link)`

```
move_link(obj_child, true, true);
```

'Links' the running object as a virtual parent to the specified **child object**.

Contrary to Game Maker Studio's built-in 'parent' system, `move_link` allows for child objects to move and scale independently and merely follow the parent *relatively* whenever the parent is moved or scaled. This can be very useful when scaling objects that are designed to interact with each other, such as a virtual chess board with its various playing pieces on top. In situations like this, the 'parent' object can be scaled with a script such as `edgeds_draw_sprite`, and with `move_link` all 'child' objects will follow suit even without running a scaling script of their own.

Can apply to all instances of an object or just one. For example, `move_link(obj_child, true, true);` would link *all* instances of the object `obj_child` to the running object, while `move_link(1234, true, true);` would link only the *instance* with the ID '1234'.

As of version 1.4.0, Edge Display Scaler also features scale linking, which will match the scale of child objects to the parent object in addition to position. Both position and scale linking can be enabled or disabled individually. To enable or disable position and scale linking, set the **pos link** and **scale link** parameters to `true` or `false`, respectively.

Note: Objects linked with this script should **not** also be linked with Game Maker Studio's own parenting system! Only one or the other can be used, otherwise conflicts will occur and `move_link` will be unable to scale child objects as intended.

5. End-User License Agreement ("EULA")

Use of this product is subject to the terms and agreements of the YoYoGames GameMaker: Marketplace End-User License Agreement ("EULA") as found at <https://marketplace.yoyogames.com/eula>(<https://marketplace.yoyogames.com/eula>)

About this document

This document applies to your use of software products licensed, or services obtained, from the GameMaker: Marketplace (which we'll call "the Marketplace" as shorthand). This document is the default contract between customers of the Marketplace (that's you!) and people or businesses from whom they can license products on the Marketplace. We call those products "Assets" in this EULA (there's more about them explained below). It also applies to the provision of services via the Marketplace ("Services"). The people or businesses providing the Assets or Services are called "the Publisher" in this EULA and they are either: (i) YoYo Games Limited (company number 05260718) of River Court, 5 West Victoria Dock Road, Dundee, United Kingdom ("YoYo Games"); or (ii) any third party business that distributes products from the Marketplace.

1. ALTERNATE EULA

1.1. The Publisher may decide to replace this EULA with its own preferred legal document. If so, that document and not this one will govern your licensing and usage of the relevant Asset or Service. You will be able to see the Publisher's EULA on the page for the relevant Asset or Services. If the Publisher does not use an alternate legal document, this EULA will apply to your licensing and use of the Asset or Services in question.

2. ACCEPTING THIS EULA

2.1. Accepting the EULA. This EULA is a legally binding contract regarding your use of Assets and Services via the Marketplace. Before you can access any Assets or Services, you need to show your acceptance of this EULA as part of your Marketplace account registration process and again when you license Assets or obtain Services. Please review this EULA carefully and contact us at [<http://help.yoyogames.com/>(<http://help.yoyogames.com/>)] if you have any questions or if you wish to propose any amendments, since once it comes into force it is legally binding upon you and us. In any event, your continuing usage of the Marketplace or any Assets or Services from it will be taken as your approval of this EULA.

2.2. Access for adults only or with adult approval. To accept this EULA or to use Assets or Services, you must be at least 18 years of age (or whatever is the age of legal majority in your country, if it's not 18). If you are not an adult, you can accept this EULA and use Assets only with a parent's or a guardian's approval. We reserve the right to refuse to permit minors to obtain access to or to use the Marketplace at our discretion.

3. ACCESSING ASSETS AND SERVICES

3.1. By "Assets" we mean software and other things sold via the Marketplace. This includes: (i) software created to facilitate the development of interactive entertainment products; and (ii) content (like graphics, sounds, music, text, or other assets) that is intended to be integrated with interactive entertainment products.

3.2. By "Services" we mean professional services of all kinds that a Publisher is authorised to provide to you via the Marketplace, including, for example, software development or other technical assistance, graphics creation and audio engineering. You may need to speak or work with the Publisher to establish the exact method of use of the Services, depending on the facts and circumstances of the case (e.g. commissioning artwork).

3.3. By "Publisher" we mean any person, company or other legal entity who has accepted the GameMaker: Marketplace Publisher Agreement to distribute Assets or provide Services on the Marketplace.

3.4. In order to access Assets and Services, you first need to access the Marketplace, which you can do by following the process explained in the Terms of Service, which includes having an activated GameMaker: Studio Professional licence.

3.5. Assets are licensed, not sold. When you "buy" Assets, what happens is that the Publisher grants to you a non-exclusive, worldwide and perpetual right (known legally as a "licence") over the Assets for the purpose of using them and integrating them with interactive entertainment products (which you may choose to distribute for free or for money – that's up to you). You may copy, use and modify Assets for this purpose only and you may not distribute, sublicense, rent, lease or lend the Assets. You are therefore acquiring a licence over the Assets,

not buying the Assets themselves (which the Publisher will still own). Your licence over the Assets is subject to this EULA.

3.6. Provision of Services. When you obtain Services via the Marketplace, the Publisher will be obliged to provide those Services to you and by default you will own what is produced from them, but not the Services themselves.

3.7. Multi-user/location arrangements. Only you are able to use the rights in this EULA regarding the Assets or Services you use – you cannot give rights to other people or share rights with others. If you would like to have a multi-user arrangement regarding an Asset or Service, please contact
[<http://help.yoyogames.com/>(<http://help.yoyogames.com/>)
]

3.8. Payment for Assets. Assets that are licensed for money can be obtained using payment services that we provide within the Marketplace with the help of third party payment services, like PayPal. As part of your licence acquisition process, you normally need to provide standard contact/payment information alongside your licence acquisition, for example your name, address and billing details. You can also provide your VAT number if you are a business (you will not be able to add it to purchases later on). We will charge you for the amount of the Asset price, plus any VAT and if applicable any bank charges, using your chosen payment method.

3.9. Updates to Assets. From time to time, Publishers may make updates (such as bug fixes, patches, new versions or enhancements) to Assets that may become available. You can set in the Marketplace whether to download these updates automatically or manually.

4. OWNERSHIP OF THE MARKETPLACE, ASSETS AND INTELLECTUAL PROPERTY RIGHTS

4.1. Ownership. The Assets, Services and all Intellectual Property Rights in them are owned by the Publisher. This includes the Asset's software, code, graphics, video, audio, music and text. The Marketplace is owned by YoYo Games (this also includes the Marketplace's software, code, graphics, video, audio, music and text).

4.2. Open source software. Some Assets and/or Services materials (whether developed by YoYo Games or by third parties) may contain open source software and so will also be governed by applicable open source software licences. If there is a dispute or discrepancy between this EULA and any such open source licences, the open source software licences will prevail.

5. CONSUMER RIGHTS REGARDING ASSETS

5.1. For consumers resident in the European Union. Under EU "Distance Selling" rules, consumers have the ability to return Assets and Services for a refund within 14 days of their ordering of them in the Marketplace. However, that right is lost as soon as consumers download or use the Assets or Services. Consumers will also benefit from legally required warranties regarding the quality and use of the Assets and Services (depending on their national consumer protection laws), which in appropriate circumstances consumers may use to claim a return or refund.

5.2. For consumers who are resident outside of the European Union (including the USA). All licence acquisitions are final and consumers have no right of return or refund, nor are any warranties given by Publishers or YoYo Games regarding the quality and use of the Assets or Services.

5.3. What happens when a consumer makes a return/refund request? For Assets, YoYo Games reviews that request on behalf of the Publisher and, if a return/refund right is available to the consumer, make a decision regarding whether a return/refund/other remedy should be offered. If a refund/return is offered, then usually you will be required to delete/destroy all copies of the Asset(s) in question and all your rights to it (them) will terminate. For Services, this is a matter for the Publisher.

5.4. What happens if an Asset or Service is taken down from the Marketplace? You will be notified that an Asset or Service is going to be removed from the Marketplace. If you have already licensed the Asset previously then in most situations we will endeavour to give you a reasonable period (normally 30 days) in which to use it a final time. However, this is subject to the reason for which it is being taken down (for example, if it is owing to third party intellectual property infringement we may be required to take it down immediately and not offer a final download period). For Services, this is a matter for the Publisher.

5.5. All other consumer queries. Please contact the Publisher in the first instance (their contact details should be available via the Marketplace). If you still have any concerns or queries after reasonable attempts to resolve them with the Publisher, you can contact YoYo Games at <http://help.yoyogames.com/>(<https://marketplace.yoyogames.com/but%20it%20is%20not%20obliged%20to%20solve%20any%20issues%20and%20can%E2%80%99t%20promise%20that%20it%20will>)

6. WHAT YOU MUST NOT DO WITH THE ASSETS OR SERVICES

6.1. You must not do any of the following regarding Assets, Services or the Marketplace:

6.1.1. Reproduce, duplicate, copy, sell, trade or resell them;

6.1.2. Copy, sell, license, distribute, transfer, modify, adapt, translate, prepare derivative works from, decompile, reverse engineer, disassemble or otherwise attempt to derive source code from them;

6.1.3. Interfere with, circumvent or bypass servers or security or content usage functionality (including any digital rights management) in or regarding them;

6.1.4. Use them to access, copy, transfer, transcode or retransmit content in breach of any laws or third party rights; or

6.1.5. Remove, obscure or alter YoYo Games' or any third party's copyright or trademark or other proprietary notices or documentation regarding them, nor use any YoYo Games or other trademarks for your own purposes.

7. TERM AND TERMINATION

7.1. Start of EULA. This EULA starts when you confirm your agreement to it as explained at the beginning of this EULA and it will end once it is terminated, as we explain further in the rest of this section.

7.2. Your termination rights. You may terminate this EULA by ending use of the Assets or Services.

7.3. Mutual termination rights. You or we may terminate this EULA at any time on written notice to you if: (i) you or we materially breach this EULA; or (ii) you or we become or threaten to become insolvent or bankrupt.

7.4. Effect of termination. If this EULA terminates for any reason then it will cease immediately to have effect (apart from any sections that are necessary for enforcement of any legal rights and remedies against you. Depending on the details of the termination, you may lose your rights over your Assets and/or Services.

7.5. Removal of your Assets or Services. From time to time, there may be serious cases of breach of this EULA by an Asset or Service. In that situation, you may be required to remove such Asset from any computer or other equipment under your control and that you agree to comply promptly with such requirement.

8. LIABILITY

8.1. You agree on demand to indemnify and hold harmless YoYo Group and its officers, directors, employees and agents (and keep them indemnified and held harmless) from and against any and all damages, claims, suits,

actions, judgments and costs (including legal costs) and expenses whatsoever, including reasonable legal fees and costs, arising out of your use of the Assets or Services, including your downloading, installation or use of any Assets, or your material breach of this EULA..

8.2. You understand and agree that your use of Assets and Services is at your own risk and that they are provided “as is” and “as available” without warranty or representation of any kind. All warranties, representations and conditions of any kind relating to them are disclaimed, including without limitation any implied warranties of satisfactory quality, merchantability, fitness for purpose or non-infringement of third party rights. In addition, no warranty or representation is given that your use of them will be uninterrupted or secure or free from bugs or errors.

8.3. The maximum and total aggregate liability of the Publisher and of YoYo Games, its group of companies, employees and affiliates (“YoYo Group”) to you in connection with this EULA shall be equivalent to the sum of \$100 (one hundred US dollars).

8.4. In no event will the Publisher or YoYo Group be liable to you for any loss of profits, charges or expenses, loss of data or any corruption or loss of information, or any loss of business opportunity, or any special, indirect, punitive, exemplary or consequential loss or damage or disruption of any kind, in any case, whether based on breach of contract, tort (including negligence or breach of statutory duty), misrepresentation, restitution or otherwise whether or not the relevant party has been advised of the possibility of such damage. The foregoing limitations will survive and apply even if any limited remedy specified in this EULA is found to have failed of its essential purpose.

8.5. Without prejudice to the generality of the rest of this clause 8, the Publisher and YoYo Group disclaim any liability regarding or arising out of:

8.5.1 Any reliance by you on the existence, completeness or accuracy of any marketing or advertising materials regarding Assets or Services;

8.5.2 Any changes made to any Assets or Services; or

8.5.3 Any damage or harm or deletion or corruption that is in any way attributable to your use of Assets or Services.

8.6. Nothing in this clause 8 purports to limit or exclude any party's liability: (i) for fraud, fraudulent misrepresentation or wilful misconduct; (ii) for death or personal injury caused by that party's negligence; or (iii) to the extent otherwise not permitted by law.

9. GENERAL LEGAL TERMS

9.1. No waiver. No failure or delay by a party to exercise any right or remedy provided under this EULA or by law shall constitute a waiver of that or any other right or remedy, nor shall it preclude or restrict the further exercise of that or any other right or remedy. No single or partial exercise of such right or remedy shall preclude or restrict the further exercise of that or any other right or remedy.

9.2. Impact of enforceability finding. If any court or competent authority finds that any provision of this EULA (or part of any provision) is invalid, illegal or unenforceable, that provision or part-provision shall, to the extent required, be deemed to be deleted, and the validity and enforceability of the other provisions of this EULA shall not be affected. If any invalid, unenforceable or illegal provision of this EULA would be valid, enforceable and legal if some part of it were deleted, the parties shall negotiate in good faith to amend such provision such that, as amended, it is legal, valid and enforceable, and, to the greatest extent possible, achieves the parties' original commercial intention.

9.3. Third parties. No person other than YoYo Group or a party to this EULA shall have any rights (whether under the United Kingdom's Contracts (Rights of Third Parties) Act 1999 or otherwise) to enforce any term of this

EULA.

9.4. No partnership or agency. Nothing in this EULA or in any document referred to in it or in any arrangement contemplated by it shall create a partnership, joint venture, agency or employment between the parties.

9.5. Entire EULA. This EULA contains the whole agreement between the parties relating to the subject matter hereof and supersedes all prior EULAs, arrangements and understandings between the parties relating to that subject matter. Each party acknowledges that, in entering into this EULA, it has not relied on, and shall have no right or remedy in respect of, any statement, representation, assurance or warranty (whether made negligently or innocently) other than as expressly set out in this EULA. Please remember, when reviewing this section, that you had had the opportunity at the start of this EULA to raise any queries and propose any amendments to its terms.

9.6. Export restrictions. It is your responsibility to comply with any import or export laws or regulations that may apply to your Assets on the Marketplace.

9.7. Assignment. The rights granted in this EULA may not be assigned or transferred by you, nor can you sub-contract or delegate your responsibilities under this EULA, without our prior written approval. The rights granted in this EULA may be assigned or transferred by us without your prior approval and we may also delegate or sub-contract our responsibilities or obligations under this EULA without your approval.

10. GOVERNING LAW AND JURISDICTION

You and we agree that this EULA shall be governed by and interpreted according to the laws of England and that any dispute regarding this EULA shall be heard by the courts of England.