



A NATIVE PIXEL-ART EDITOR FOR MACOS

Minium

Everything in the app, and how to use it — written for somebody opening it for the first time.

Minium draws sprites, animates them, and paints the rooms they walk around in. It is arranged the way a Mac image editor is arranged and it uses the words you already know: an Image menu holds the size of the picture, a panel is a panel, and the Eyedropper is not called Pick.



Contents

01	The window	What every panel is for	2
02	Your first sprite	From New to a PNG, in ten minutes	3
03	The tools	Every tool, its key, and what the buttons do	4
04	Selecting and transforming	Marquees, moving, rotating, scaling	6
05	Layers, groups and text	The stack, and the things that live in it	8
06	Animation	Frames, tags, onion skin, views and facings	10
07	Painting a room	Masks, perspective, depth and the pivot	12
08	Colour and palettes	Ramps, the swatches panel, importing and saving	14
09	The canvas	Zoom, guides, grid, references, resizing	16
10	Saving, exporting, importing	Documents, sheets, GIFs, recovery, timelapse	17
11	Making it yours	Themes, settings and the workspace	19
A	Keyboard shortcuts	The whole set, by menu	20
B	What it does not do	Known gaps, so you are not hunting for them	22

The window

Five things surround the sprite, and each one answers a different question. Learn where they are once and the rest of the app is findable.



The default arrangement. Tools can be moved to the right-hand side in **Settings ► General**; every panel can be hidden from the **Window** menu.

The Tools panel

Down the left edge, one button per tool, each with a single-key shortcut. Those keys are suspended while a text field has focus, so typing a layer name never puts the eraser in your hand.

The options bar

Above the canvas, and the most important strip in the window: it holds the settings for whichever tool you are holding — brush size, tolerance, fill style, stroke thickness — and it changes as you change tools. Two things live at its right-hand end permanently, because they are wanted whatever tool is in hand: **symmetry** and **perspective**. The bar also swaps its contents wholesale when the current layer is a text layer (type controls) or a mask layer (an area picker).

Swatches and Layers

Docked down the right. Swatches holds the palette, the colour picker and the foreground/background wells; Layers holds the stack — eye, lock, name, and

the commands underneath the rows. Drag the panel's leading edge to make it wider; the edge shows three dots when the pointer is over it.

The Timeline

Along the bottom: frame numbers, the cels, the coloured tag bars, and the transport. Drag the bar above it to make it taller. The layer names run down its left side as a gutter to read the grid by — the Layers panel is where layers are actually worked on.

The preview

A floating window showing the sprite at its real size while you work at 800%. Pixel art is drawn zoomed in and judged zoomed out.

NOTHING IS LOST BETWEEN SESSIONS

Which panels were open, how big they were, your colours, your palette, and the tool you were holding with its brush size all come back next launch. See *Making it yours* at the end of this guide.

Your first sprite

A complete pass through the app — new document, some pixels, a second layer, and a PNG on your desk. Ten minutes.

- 1 **New Sprite** (**⌘N**). Rather than typing a size, take a template: they are grouped into characters, animation, tiles, backgrounds and screens, and each says what it is for. A 32×32 *character* is a good first canvas. The width, height and frame count stay editable, so a template is a starting point rather than a mode.
- 2 **Zoom in.** **⌘9** fits the sprite in the window, snapping to a whole-number zoom so the pixels stay square; **⌘0** is actual size. **⌘**-scroll or pinch zooms about the pointer, plain scroll pans, and holding **Space** drags the canvas from any tool.
- 3 **Pick a colour.** The Swatches panel opens on *Ramps 32*, which is the palette to work in. Click a swatch for the foreground colour, right-click one for the background. **X** swaps them.
- 4 **Draw** with the Pencil (**B**). The left button paints the foreground colour, the right button the background. Fast strokes have no gaps — the pointer samples are joined. **[** and **]** make the brush smaller and bigger.
- 5 **Undo freely** (**⌘Z**). One entry per stroke or command, and the history is bounded by memory rather than by a fixed count, so it is longer than you expect.

- 6 **Add a layer** for the shading: the *new layer* button under the Layers rows, or **⇧N**. A new layer always arrives empty and becomes the current one. Drag rows to reorder, double-click a name for opacity and blend mode.
- 7 **Shade it** with the Shading tool (**S**). It lightens under the brush and darkens with the right button, stepping along the colour's own ramp in the palette — so the shading stays inside the palette instead of inventing colours.
- 8 **Save** (**⌘S**) writes a .minium document: layers, frames, tags, blend modes, opacities, guides and the palette. This is the file you keep working in.
- 9 **Export** (**⌘E**) writes a flat PNG of the current frame. Exporting is not saving — the document stays marked as edited, because a PNG cannot carry the stack.

TWO THINGS WORTH KNOWING ON DAY ONE

Hold **⌘ with any tool to pick a colour off the canvas** — and unlike every other tool switch, this one leaves the tool in your hand. The pencil and the bucket also pick with **⌘**.

Sprites live only in memory until they are saved. Quitting, starting a new sprite or opening another image with unsaved edits asks first, and a recovery copy is written every couple of minutes while there are unsaved changes.

The tools

Fourteen tools, each on a single key. Their settings are in the options bar above the canvas.

TOOL	KEY	WHAT IT DOES
Marquee	M	Select an area. Rectangle, ellipse, lasso or polygon, chosen in the options bar.  adds,  subtracts, a click with no drag deselects.
Layer Selector	A	Click art to select the layer it is on, then drag to move that whole layer.
Move	V	Drags the selection's pixels — or everything on the layer when nothing is selected. Arrow keys nudge one pixel,  ten.
Transform	T	Turns and resizes.  stamps it down,  puts it back. See the next chapter.
Magic wand	W	Selects a region within the tolerance, using the same sampling and contiguity settings as the bucket.
Pencil	B	Freehand. Drags are joined so a fast stroke has no gaps.
Eraser	E	Freehand, writes transparency.
Line	L	Drag for a live preview, release to commit. Snaps to perspective rays when a perspective is set up.
Rectangle	R	Corner to corner. Outline, fill, or both.
Ellipse	O	Corner to corner of its bounding box. Outline, fill, or both.

Shading	S	Lightens under the brush, darkens with the right button, stepping along the colour's palette ramp. Each pixel moves once per stroke.
Paint bucket	G	Flood fill. Judges the region from the flattened image by default — an outline on a layer below still bounds the fill — while the paint lands on the active layer.
Text	Y	Click empty canvas for a new text layer, an existing one to edit it, drag to move it.
Eyedropper	I	Reads the composited image. Also on  from any tool.
Hand	H	Pan. Space -drag and middle-drag do the same from any tool.

The two colours

Left button paints the foreground colour, right button the background.  swaps them, and  and  fill the selection with one or the other — the whole layer when nothing is selected. Fill is Clear's opposite, and it is not the question the bucket asks: the bucket judges a region and stops at its edges.

Which tool you end up holding

Choosing a colour brings the pencil back, but *only* from a tool that cannot lay that colour down — the eraser, the shading tool, the eyedropper and the selection tools. A tool that already paints with the colour keeps working: choosing a fill colour should not take the bucket out of your hand.

Stroke widths

The pencil, eraser and shading tool paint at the **brush size**. The line, rectangle and ellipse have their own **thickness** — a three-pixel outline and a three-pixel pencil are different decisions. [and] narrow and widen whichever is in hand, so the keys mean smaller and bigger without your having to know which number is behind them. Both are remembered between sessions.

Filling and outlining

The shape tools take three styles rather than a fill checkbox:

- **Outline** — the border only, at the stroke thickness.
- **Fill** — solid, in the colour being painted with.
- **Both** — filled with the background colour and outlined with the foreground. The right button swaps that pair.

The inside is painted first and the border over it, so a thick border eats into the fill rather than growing outside the shape you dragged.

Freehand algorithm

How the pencil and eraser trace a drag, set in the options bar or under **Edit ► Freehand Algorithm**:

- **Regular** (default) — join the pointer samples into a continuous stroke.
- **Pixel Perfect** (⇧P) — drop the doubled pixel at L-shaped turns, so a hand-drawn diagonal comes out with even steps and a right-angle turn leaves a single elbow pixel. It runs *during* the stroke, so the preview is already the result. It applies to a 1 px brush only.
- **Dots** — one dab per pointer sample, unjoined.

A stroke keeps the algorithm and brush it started with, and the whole drag is a single undo entry. Painting is blocked on hidden and locked layers, and clipped to the selection when there is one.

Symmetry

Set from the right-hand end of the options bar whatever tool is in hand, or **View ► Symmetry Options**. It mirrors what the drawing tools plot across a vertical axis, a horizontal one, or both. The axes are drawn as pink dashed guides and start centred; *Centre the Axes* re-centres them after a resize. The pencil, eraser and line mirror; the bucket does not.

Selecting and transforming

A selection is a shape drawn on the canvas that says where the next thing applies. Painting is clipped to it, and it is what the Move and Transform tools act on.

Placing a marquee

Rectangle, ellipse and lasso are dragged out. Two modifiers are worth learning immediately:

- **Space** slides the shape while you are **dragging it**. Started a rectangle a shade too far left? Hold space and the whole marquee moves with the pointer; let go and the drag carries on sizing it from where it now is. Nothing is decided until the mouse comes up.
- **⌘** -drag from inside a selection moves a copy of it, leaving the original where it was. Starting the drag outside the selection draws a new marquee as usual.

The polygon marquee

The other three shapes are a drag; a polygon is not. **Click to place each corner** — straight edges at exact corners are the whole point, and a hand dragging a mouse cannot place one. Close it by clicking the first corner, double-clicking, or pressing .  takes back the last corner;  abandons the shape and leaves any existing selection alone. Placed corners are joined solid, the segment following the pointer is dashed, and a fainter seam runs back to the start so the area reads before it is closed.

The Select menu

All () , Deselect () , Reselect () , Inverse () , Color Range..., and Modify ► Border / Expand / Contract by 1, 2, 4 or 8 px. Selections are outlined with animated marching ants.

Transform

Rotate and scale, on a selection or on a whole layer. Reach it with , or Edit ► Transform ► Rotate and Scale... () , which takes the tool and lifts the pixels in one go. Clicking or dragging with the tool **lifts** the

pixels: the selected ones, or every opaque pixel on the layer where nothing is selected.

Drag inside the box	move it
Drag a corner	resize, pinning the opposite corner
Drag the grip	turn it about the pivot — the knob on a stem off the top edge
Drag outside the box	turn it too
Drag the ring	move the pivot, leaving the art where it is
Arrow keys	nudge, ten with 
	stamp it down
	put every pixel back

Nothing is committed until you say so, and the whole session — however many adjustments went into it — is **one undo entry**. Backing out records nothing at all. The original is resampled from the pixels it started as every time, never from the previous result, so going out to an ugly angle and coming back to zero returns the sprite exactly. Reaching for another tool, layer or frame *applies* the transform rather than dropping it.

THE ANGLE

Three ways to set it, because there are three ways of knowing what you want: drag the grip for an angle found by eye; drag the readout in the options bar for one arrived at against the number ( inverts the snap); or step it with the arrows beside the readout, a degree at a time. It reads against zero rather than round a full turn — a small turn to the left is -15° , not 345° .

THE PIVOT

The ring and cross inside the box is what the art turns about. It starts in the middle and drags anywhere, including outside the art. **Pivot ► Middle / Feet / Character Pivot** in the options bar

are the three places worth naming. The pin is held in the art's own coordinates, so a signpost pivoted at its base is still pivoted at its base after being scaled and turned.

WHAT A ROTATION COSTS

A **quarter turn is the only rotation that costs a pixel sprite nothing** — it moves whole pixels and nothing else. **Edit ► Transform ► Turn Left / Turn Right** ( and ) are the one-shot version.

Everything else has to guess: the step is one degree, resampling is nearest-neighbour throughout (anything that blended would invent colours the palette does not have), and the options bar shows a warning triangle whenever the transform in hand is not one of the lossless ones. Art hanging over the canvas edge is trimmed when it is stamped down, and the options bar says *off canvas* while any of it is —  still puts it all back, and **Image ► Canvas Size** makes room.

Layers, groups and text

A sprite is layers × frames. Each layer holds one cel per frame, and rendering composites the stack bottom-up, honouring each layer's visibility, opacity and blend mode.

The Layers panel

Top layer first. Each row carries the eye and the lock, the name, and a badge when opacity or blend mode is not the default. Click to make one current, double-click for its properties ($\uparrow P$), drag one onto another to reorder, right-click for the rest. The commands sit *under* the rows: new layer, new mask layer, group and ungroup, then, past a divider, duplicate, merge down and delete.

- **New Layer is always empty** — no cel on any frame, whatever the layer under it holds — and named *Layer N*.
- **Names count rather than pile up.** A duplicate of "Layer 5" is "Layer 6", not "Layer 5 copy"; "BG" gives "BG 2". The number is counted from the names actually in the list.
- **A duplicate is never locked**, whatever it was copied from. Everything else comes across.
- **Selecting more than one** — $\mathbb{M}$ -click adds a layer, $\uparrow$ -click takes the run between. The current layer keeps an outline the rest do not. **Merge Layers** ($\mathbb{M}E$) collapses them in stacking order with each one's opacity and blend mode folded in; hidden layers are merged rather than dropped.

Picking a layer by its art

The **Layer Selector** (A) answers "which layer is that on?" by clicking it: the topmost visible layer with something opaque under the pointer becomes current, and the same drag moves the whole of that layer. A hidden layer is seen through — you are looking at whatever is below it. A locked one is not: locked art stops the pick, and the click does nothing.

A layer dragged off the canvas keeps what leaves it. Push half a layer over the left edge and the half that went is held off the side; dragging or nudging it back brings it home intact, and it survives saving and reopening. Growing the canvas paints it back in. This applies to moving a *layer*; moving a

selection does not take it, and scaling or rotating the sprite drops what is off the canvas rather than guessing.

Groups

Layer ► Group Layers ($\mathbb{M}G$) puts the selected layers in a folder; $\mathbb{M}G$ takes it apart. A group can hold groups, and the triangle on its row shuts it — a room of thirty layers reads as five things.

A group is not only a folder. Its visibility and lock reach everything inside it, and its **opacity and blend mode apply to what it holds, composited as one**: three props at 70% each show through one another; three props in a group at 70% do not. Deleting a group deletes its contents, duplicating one copies them, New Layer with a group selected adds *into* it, and selecting a group in the timeline makes the Layer Selector drag every layer inside it on one offset. Locked and text members stay behind; hidden ones travel.

Copying layers

Layer ► Copy Layer ($\uparrow \mathbb{M}C$) and **Paste Layer** ($\uparrow \mathbb{M}V$) take the whole layer — every frame's cel, the name, opacity, blend mode, visibility, and a mask's kind, baselines and scaling. It is the app's own clipboard, so copying a layer does not throw away whatever image you had copied from somewhere else; $\mathbb{M}C$ and $\mathbb{M}V$ still work on the selection's pixels and still talk to other apps.

A PASTE GOES BACK WHERE IT CAME FROM

Pixels copied here return to the coordinates they were taken from, whatever is selected at the time — the point of copying something is usually to have it again in the same place, on another frame or another layer. An image from another app has no origin to return to, so it lands at the selection if there is one and in the middle otherwise.

Text layers

Layer ► New Text Layer, or the Text tool () , adds a layer that keeps its *words* rather than only a picture of them: the wording, font, size and colour all stay changeable afterwards. The options bar swaps to the type controls whenever a text layer is current. A dashed box shows the block while the tool is in hand, because where the words end is what you are working against.

Anti-aliasing is off by default — a soft glyph edge is a row of colours the palette does not have. Turn it on for a title card over a photograph, where it is right. Any font on the machine can be used; one a document names but this machine lacks falls back to the system font, and the pixels that were saved are what opens.

A text layer composites, exports, groups, merges and duplicates like any other. What it will not take is paint — a stroke on one would be drawn over the next time the text changed — so painting is refused with a note saying why. **Rasterize Text Layer**, in the Layer menu and on the options bar, turns it into ordinary pixels; that is one-way, apart from undo. A resize or crop carries the words along; a rotation or flip rasterizes the layer rather than pretending.

Mask layers

Layer ► New ► New Mask Layer adds a layer that is a note about the picture rather than part of it — walkable areas, walk-behinds, hotspots and regions for a point-and-click room. Masks never composite, never export with the art, and refuse to be grouped. They have a chapter of their own.

Animation

Frames carry their own duration in milliseconds. The timeline holds the transport, the tags and the grid of cels.

Frames

New Frame		copies the current one
New Empty Frame		
Duplicate Cel		
Delete Frame		
Frame Properties		duration, or a constant frame rate across the sprite
Reverse Frames		tags reverse with them
Previous / next frame		wrapping round the ends
Play / stop		0.25x–3x, loop and rewind-on-stop in the Playback menu

With nothing selected, **the arrow keys walk the timeline** — left and right for frames, up and down for layers, the same two axes the timeline is laid out on. With pixels lifted or a selection on the canvas they nudge those instead.

Tags

Animation ► Tags ► New Tag..., or the tag button in the timeline, names a range of frames. Tags appear as coloured bars above the frames they cover; clicking one plays it, and while one is selected playback and GIF export stay inside its range. Tags follow their frames: inserting a frame inside a tag extends it, inserting one before shifts it along, deleting frames pulls it back.

Onion skin

F3, the *Onion* button, or **View ► Onion Skin**. Earlier frames are tinted red, later ones blue, fading with distance, and the frame numbers in the timeline tint to show which are being ghosted. The **–/+** steppers set the range. It **wraps around the loop**: at the first frame of a walk cycle, the frames "before" are the

last frames of that cycle, which is exactly where the reference is wanted most.

Views and loops

A walk is not one animation but four or eight — one per facing. **Animation ► View ► New View...** lays a whole one down: name it, pick four or eight facings and a length, and it creates the frames and tags each loop. Doing that by hand means adding thirty-two frames and then tagging them in eight goes.

- **A loop is as long as you say** — the length is typed, and any number up to 240 is a length. Four frames reads, eight is smooth, five is what the walk you are drawing turned out to need.
- Loops are ordinary tags with a facing attached, so everything that worked on tags still does.
- Facings are declared in the order engines number their loops — down, left, right, up, then the diagonals.
-  and  step between facings, wrapping round the view, each landing on its own first frame. Each facing gets its own hue in the timeline and the tag shows an arrow.

Drawing eight facings

The difficulty is never one loop — it is the agreement between them. Two features exist for exactly that.

The facing reference (, the *Facing* button, or **View ► Facing Reference**) ghosts the matching frame of another facing behind the one being drawn. Onion skin looks along the timeline; this looks across it. On frame 3 of *Walk left* it shows frame 3 of *Walk down* — the same moment, not that loop's first frame. It is drawn as a flat silhouette, under the art, so what shows is exactly the part of him this drawing does not already cover. Colour and opacity are in **Settings ► Timeline**.

Mirror Facing (**Animation ► View ► Mirror Facing...**) writes one loop's frames into another, reflected. Three of the eight facings are somebody else's mirror — left and right, and the two pairs of

diagonals — so an eight-way walk is five drawings rather than eight. It reflects about the **character pivot** where the sprite has one, not the middle of the canvas, so a walk mirrored does not shift sideways every time he turns. Every art layer comes across; masks stay put. The sheet says how many frames it will write and warns when the target already has art on it, or when the reflection would push art off the canvas. It is one undo entry.

A STAMP, NOT A LINK

You mirror and then fix — the sword hand, the satchel, the parting in the hair. A live link would break at exactly the moment it mattered.

Getting the animation out

`⌘G` writes an animated GIF with each frame's own delay (and the current tag's range, if one is active). `⌘E` writes every frame as a numbered PNG. **File ▶ Export ▶ Sprite Sheet...** packs every frame into one image — see chapter ten.

Painting a room

This is the part of Minium that is not in other pixel editors. A point-and-click room is a background plus a set of masks written over it: where a character may walk, what he passes behind, what the player can click, and which areas change what happens.

Mask layers

Layer ▶ New ▶ New Mask Layer adds one, in four kinds — **walkable**, **walk-behind**, **hotspot** and **region**. A mask is not part of the picture and the app treats it that way: mask layers are excluded inside the compositor itself, so they cannot reach an export, the preview, a flatten or a trim by being forgotten somewhere. The canvas draws them separately, over the art, at an opacity set in **Settings ▶ Background**.

WHAT A MASK STORES IS A NUMBER, NOT A COLOUR

Painting on a mask layer lays down the selected *area's* colour whatever colour is in hand — an engine reads area 3, not a shade of green — and the options bar swaps its usual controls for an area picker while a mask layer is active. Erasing still erases.

Naming areas

Every area on any mask can be named — Door, Window, behind the counter — in the options bar beside the area picker. The number is what an engine reads, but a room of areas called 1 to 7 tells the person painting it nothing, and the names go out in room.json so the engine can read them too.

Walk-behinds and their baselines

A walk-behind carries one more thing: a **baseline per area**, deciding where a character stops being hidden by it. Feet below the line put him in front, above it behind — without one he walks through the pillar. Each area's baseline defaults to the bottom of what is painted, which is usually right, so nothing has to be set at all. Drag the dashed line to move it, or use the stepper in the options bar; *Reset* drops it back to the default. Dragging is offered only on the layer being worked on, so a baseline never intercepts a stroke meant for another layer.

Walkable areas and depth

A character is smaller at the back of a room than at the front, and the walkable area is where that is decided. Set *far* and *near* percentages in the options bar and everything between is interpolated. An area with nothing set reports *no scaling* rather than 100% — a flat room and a room scaled to full size are different answers.

View ▶ Depth Preview () puts a stand-in figure in the room at the pointer, drawn at the scale that row gives and labelled with the height it comes out at, plus faint figures at the back and front of each scaled area. A silhouette rather than real art: the question is whether the scaling reads, and a silhouette answers it without needing anything loaded. Its height is in **Settings ▶ Background**.

The screen marker and the pivot

Image ▶ Room Viewport ▶ Show One Screen draws one screen's worth of a scrolling room — 16:9 of the room's height by default, or half or a third of the room — and dims everything outside it. Painting a 960 × 270 room means painting three screens, and nothing on the canvas said where one ended. Drag either edge to slide it along; the middle is still canvas to paint on.

Image ▶ Character Pivot marks where the character stands: the point an engine positions the sprite by. It starts at centre-bottom and drags anywhere, drawn as a ring and a cross so the pixel being aimed at stays visible. Both the viewport and the pivot are part of the document.

Perspective

A room is painted to an eye level, and everything in it runs to the same places. One-point for a corridor running away from the camera, two-point for a room seen into its corner, three-point for the one room looked up or down at.

The control is in the options bar, at the right-hand end beside symmetry, because that bar is on screen whatever tool is in hand. The icon fills and the label says what is set up — *Two-point, One-point* — so the bar answers "does this room have an eye level?" without being opened. The same set is in **Image ► Perspective**, and  shows and hides it.

The **horizon** is the eye level, drawn across the whole window, starting a third of the way down. Three ways to move it, because there are three different things you might be doing:

- **Drag the line** anywhere along it, for an eye level found by eye.
- **Type a row** into the stepper in the options bar, for one you already know.
- **Press *Horizon from this*** with a walkable area in hand, which puts it where that area's scaling says it is.

Vanishing points are drawn as a ring and a cross and dragged. Points that sit off the canvas have nothing on screen to take hold of, so **Image ► Perspective ► Perspective...** is where those are typed: the horizon row, the line count, snapping, and every point's name and position, with X free to be well outside the picture. The sheet applies the lot as a single undo entry. Positions are deliberately not clamped to the canvas — a two-point room's points belong well outside it, and points placed just off the edge give the bulged wide-angle look that reads as a mistake.

The line tool snaps to the rays. Drag roughly along one and the line lands on it exactly, keeping the length the drag gave it. A drag well off any ray is left alone, and holding  turns the snap off for one stroke — a room drawn to a perspective still has a picture hanging straight on the wall. The rectangle and ellipse do not snap: they are their own two axes.

WHERE PERSPECTIVE MEETS THE WALKABLE AREAS

A character of fixed height projects to a screen height proportional to his distance below the eye level, and an area's depth scaling is interpolated linearly down its rows — so a straight far-to-near

ramp is exactly right, and only has to start and end in the right places. The two halves of the room can therefore be made to agree:

- **Fit to horizon** sets an area's far scale from the eye level and keeps the near one. The size you drew the character at is the thing you chose; the far end is the thing that is fiddly to work out.
- **Horizon from this** goes the other way, for a room painted before its eye level was decided — laying a one-point perspective down if the room had none.
- Beside them, how far the area's scaling currently is from what the horizon implies, as a percentage.

With the Transform tool, moving lifted art up or down the floor and pressing **Fit** scales it by exactly the ratio the horizon already knows — and the fit keeps the feet planted, since a character resized about his middle sinks through the floor.

Pixel aspect ratio

Pixel art was not always drawn on square pixels: a 320 × 200 VGA screen was displayed at 4:3, which makes every pixel a fifth taller than it is wide. **Image ► Pixel Aspect Ratio** sets it — square, VGA 320 × 200 (5:6), double-wide (2:1) as in C64 multicolour and EGA/Amiga low-res modes, double-tall (1:2), or a ratio typed in. The canvas, preview, grid, guides and rulers all stretch together, and the status bar shows the ratio whenever it is not 1:1. Image files cannot carry a ratio, so **Settings ► Files** bakes the stretch into exports by default: a 320 × 200 sprite at 5:6 leaves as a 320 × 240 PNG.

Getting the room out

File ► Export ► Room Masks... writes one PNG per mask layer, named after the document and the mask, which is how an engine expects to find them — plus a small JSON file of walk-behind baselines and walkable-area scaling, since those are numbers and a PNG cannot carry them. Masks survive what happens to the art: a flatten keeps them, a merge refuses to mix a mask with art or two masks of different kinds, and duplicating one keeps it a mask.

Colour and palettes

Minium ships with three palettes drawn for it and a grayscale ramp. They are built to a rule, and the rule is why they are worth using.

Ramps 32	The one to work in. Seven ramps of four and a set of greys.
Room 24	For backgrounds. Stone, sky and water, foliage, wood, brick, lamplight — held back on purpose, since a room is painted to be stood in front of.
Night 16	A night exterior. Twelve cold swatches and four warm ones.
Grayscale	Sixteen greys, generated.

The shading tool steps a colour to its neighbour along the ramp it belongs to — and a ramp here *is* a hue family. So a palette is only as useful as its families are coherent. In these, every family holds exactly one ramp, every step is clearly brighter than the last, and every ramp can be walked end to end a step at a time.

The Swatches panel

Laid out as a document about the palette: its name and origin at the top, a filter field, a card showing the current colour with its index and ramp, the swatches themselves, named actions, recently used colours, and the mixer at the foot — a saturation/brightness square with hue and alpha strips, the foreground and background wells with a swap button, and the Idx-N and hex read-outs. Everything in the panel's options menu is also in the **Swatches** menu in the menu bar.

Ramps and filtering

Palette files carry no grouping, so the swatches are grouped into hue families (Neutrals first, then Reds through Pinks) worked out from the colours themselves. Grouping is only a way of looking at the list — every swatch keeps its real palette index, so indexed sprites and the Idx-N read-out are unaffected. Turn it off to see the palette in index order. The filter field matches a hex or a ramp name.

Editing and creating

Click a swatch to set the foreground colour; right-click one to make it the background, overwrite it, or delete it. Sort by hue, saturation, brightness or luminance, reverse the order, remove duplicates, or pull in every colour used by the current frame with *Add Colors from Sprite*. **Palette edits go through the same undo stack as pixel edits**, so **⌘Z** takes back an accidental delete or sort. New Empty Palette, Duplicate Palette, New Palette from Sprite and Rename are all there; renaming a built-in saves a copy under the new name rather than altering the original.

Sampling an image

Import from Image... and *New Palette from Sprite* open a sheet before adding anything: an anti-aliased or photographic source holds hundreds of barely-different shades, and taking the first 256 in scan order produces a palette of one gradient. The sheet offers a colour cap and a merge distance, with the resulting swatches previewed live as you move them. Reduction is median cut weighted by how often each colour actually appears, so the swatches land where the pixels are. When there is nothing to reduce — a palette strip, or a sprite with fewer colours than the cap — the image's own order is preserved, because for a palette strip that order *is* the palette.

Importing and exporting

Load Palette... reads .gpl (GIMP), .pal (JASC), .act (Adobe Color Table), .aco (Adobe Color Swatch), .clr (macOS colour list), .hex/.txt, and image files. A file with an unhelpful extension is sniffed by content, so a mislabelled .gpl still loads; imports are capped at 256 colours. *Export As* writes any of the same formats, plus a palette strip PNG — one pixel per swatch, one row — that reads back cleanly.

The library

Saved Palettes ► *Save to Library* writes the palette as a .gpl into the app's Palettes folder in Application Support. Everything in that folder loads at launch

and appears in the switcher, in every document. The switcher marks each palette with where it came from — a lock for the built-ins, a folder for the library, a page for one that only exists here.

TWO COMMANDS THAT READ ALIKE AND ARE NOT

Remove from This List takes a palette out of the switcher for the session and leaves the disk alone. *Delete from Library...* removes the file, with a confirmation naming it. They sit apart for that reason — one inside the Saved Palettes menu and one outside it.

Indexed mode

Image ► Mode offers RGB, Grayscale and Indexed. Indexed is a constraint rather than a storage format: converting quantizes every pixel to the palette, and later painting snaps to the nearest palette entry.

The canvas

Moving around

- Scroll to pan; **⌘**-scroll or pinch to zoom about the pointer.
- **⌘0** actual size, **⌘9** fit in window — fitting snaps to a whole-number zoom so pixels stay square.
- **⌘+** and **⌘-** zoom in and out.
- **⌘H** hides and shows the extras drawn over the art.

Guides and rulers

Rulers (**⌘R**) run along the top and left edges, marked in sprite pixels. Drag out of a ruler to pull a guide onto the canvas — the top edge gives a horizontal guide, the left a vertical one — and drag a guide to move it, its position showing beside it. A guide sits on the boundary *between* two pixels, so a vertical guide at 8 runs down the left edge of column 8.

Throw one away by dragging it back into its ruler or well past the edge of the canvas; both are undoable, as are moves. A press within a few points of a guide grabs the guide rather than painting — **View ▶ Guides ▶ Lock Guides** (**⌘;**) turns that off when drawing tight against one. **⌘;** hides and shows them. Guides belong to the document and ride the undo history.

Grid, checkerboard, tiled mode

View ▶ Show is what is drawn over the canvas — the pixel grid, the transparency checkerboard, the perspective rays. **View ▶ Grid** takes a cell size and offset. **Tiled mode** repeats the sprite around the

original, on both axes or just one, so a seamless tile can be checked while it is being drawn.

Reference image

View ▶ Reference Image ▶ Place... puts an image behind the canvas to work against — the room a character is being animated over, a photograph being traced, a screenshot being matched. It is never part of the sprite: it does not composite, does not export, and belongs to the workspace rather than the document, since the same character might be drawn against three rooms in an afternoon. **⌘⌘R** shows and hides it; the menu also sets its opacity, fits it to the canvas, centres it and removes it. Move it with an **⌘;**-drag.

Resizing

Image ▶ Canvas Size... (**C**) adds or removes space around the artwork *without scaling it*: type a size, pick a common one, double or halve it, or *Fit to artwork* to shrink the canvas to exactly what has been drawn. A 9-way anchor decides where the existing artwork sits in the new canvas, and the dialog says what the change will do, including a warning when it would crop something away.

Image ▶ Image Size... (**⌘⌘I**) scales the artwork itself, with nearest-neighbour sampling and an aspect lock. **Crop to Selection** and **Trim** are in the same menu, as are **Image Rotation** — 90° either way, 180°, flip horizontal or vertical.

Undo

Snapshot-based, one entry per stroke or command, bounded by a memory budget rather than a fixed count. Depth and budget are in **Settings ▶ Undo**.

Saving, exporting, importing

Documents

⌘S writes a .minium document — the sprite itself, with its layers, frames, tags, blend modes, opacities, guides, perspective and the palette it was painted with. Reopening one returns exactly what was saved. **Export** writes a flattened picture instead and **does not count as saving**, so the document stays marked as edited.

Double-clicking a document opens it, as does dropping one on the app's icon. **Open Recent** lists the last ten documents and images; one that has been moved or renamed is followed to wherever it went, and one that has been deleted is dropped from the list rather than failing quietly.

The four exports

Frame as PNG	⌘E	The current frame, flattened. A save panel asks where.
Frames as PNGs	⇧⌘E	A run of numbered files, with options.
Animated GIF	⇧⌘G	Each frame's own delay; the current tag's range if one is active.
Sprite Sheet	—	Every frame packed into one image, plus the JSON an engine needs.

FRAMES AS PNGS

The dialog remembers its settings. Choose **which frames** (everything, this one, this loop, this tag, or a typed range — 1-based and inclusive, and it reads either way round, so 8 to 2 is the same eight frames as 2 to 8); **what they are called** (a name, the separator before the number, how many digits, and what the first number is — starting at zero is there because an engine that counts frames from zero wants the files to agree); and a **scale** of $\times 1-8$. *Named by loop* makes each facing count from the start again — walk-down-01, walk-left-01 — since a view of eight facings exported as 01 to 32 is a folder nobody can read. The dialog shows the first and last file it is about to write, which answers the separator, the padding and the numbering at once.

SPRITE SHEET

Frames: everything, this loop, or this tag. Arranged *one row per facing* by default, which is how a character sheet is read — a short loop leaves a gap rather than shifting the next facing left. One row, one column and a grid are there for everything else. Then scale $\times 1-8$, **padding** between cells, and **extrude**, which repeats each cell's edge outwards so the cell's own edge does not fade when a GPU samples half a texel past it. **The JSON beside it** is on by default: cell size, the grid, every frame's position and duration, the loops with their facings and which row each occupies, the pivot, and the pixel aspect ratio when it is not square. A sheet without it is a picture rather than an asset.

LAYERS

File ► Export ► Layers..., or *Export Layer...* on a layer's own row, writes a picture of some of the stack rather than all of it — a character lifted off his room, or the room without him. Choose the layers picked in the panel, every visible one, or every one; and whether they come out as **one composited image** or **a file each**. A group is one entry, exported at the group's own opacity and blend mode. A hidden layer that was picked is drawn — picking it is asking for it. Nothing is trimmed: every file is the size of the canvas, so the layers drop straight back on top of one another.

Importing

File ► Import Image... (**⇧⌘O**) brings a picture — or several — into the sprite already open. (*Open* does something different: it replaces the document with the image.) It takes .minium files as well, which can land **as layers**, keeping each layer's name, opacity, blend mode, visibility and any room masks.

A still image lands either **as a layer**, centred on the current frame and named after the file, or **as frames**, cut into a grid the way a sprite sheet is laid out. The sheet says how many frames the grid gives and how big each cell is, and warns when the numbers do not divide evenly.

A set of files is a sequence. Choose several images and they land along the timeline in order, on the layer being worked on, ordered by number rather than by text so frame 10 follows frame 9. You are

asked where they land — at the start, at the frame being looked at, or after the last — and from there the sequence *fills* the frames already on the timeline rather than making new ones, adding frames only where it runs off the end. **An animated GIF brings its own frames** and its own timing, so it is not cut into a grid.

How they fit is asked only when it makes a difference: *resize the canvas* (offered only for a sprite with nothing on it), *scale to fit* (never scaling up, nearest-neighbour), or *centre, clipping*.

Recovery

While there are unsaved changes the open document is copied to the app's Recovery folder every couple of minutes. Saving clears it, so a copy only survives a session that ended badly — and the next launch *offers* to bring it back rather than restoring it silently, since you may have moved on. Both the interval and whether it happens at all are in **Settings ► Files**.

Timelapse

File ► Record Timelapse () starts capturing the canvas as you draw it; the same key stops it, and the

status bar shows a red **REC** and a frame count while it runs.

- **A frame is taken per finished edit** — a stroke ended, a fill, a layer merged — rather than on a clock, so a session left open over lunch costs nothing and the film has no dead stretches. Undo and redo *are* recorded: backing a stroke out is part of how the picture came to be.
- **A recording belongs to the sprite, not to the sitting.** Quitting and reopening carries on with the same recording, so a piece made over a week is one timelapse. **Discard Timelapse Recording** throws it away, and asks first.
- **The export fits the session into a length you choose** rather than the other way round, so a three-stroke sketch and a twelve-hour piece both come out watchable. The dialog says how many frames, whether they are being sampled or held, and how big the video will be before you commit. The film always ends on the finished piece.
- **The background is transparent by default**, which needs HEVC in a .mov — most editors take that, most web players do not. Turn *Transparent background* off for an H.264 .mp4 composited over black, which plays anywhere.
- The encode runs in the background with its progress in the status bar, and can be cancelled.

Making it yours

Themes

Four, chosen in **Settings ▶ General** or from **View ▶ Theme**, applied as they are picked.

Ember (default)	Warm — dull red panels, amber where it counts. Both colours are held back on purpose so they sit under artwork all day.
Dark	Cool grey — the chrome as it was before there were themes.
Light	Pale chrome for a bright room.
Midnight	Near-black and blue, for working at night.

The grid over the art flips with the theme, light over dark and dark over light, as do the markers drawn on the canvas. **The backdrop behind the sprite** follows the theme until you pick a colour in **Settings ▶ Background** — after that your choice stands. Guide, grid and perspective colours stay separate: they are explicit choices rather than chrome.

The workspace

The app comes back to how it was left: which panels were showing and how big they were, the foreground and background colours, the palette in use (remembered by name), and the tool with its brush size, freehand algorithm and symmetry. A workspace is not the document — it outlives whatever sprite happens to be open. **Window ▶ Workspace** also keeps one deliberately: *Save Workspace*, *Restore Workspace*, and *Reset Workspace* to go back to the defaults in Settings.

Settings

 opens them: a list of sections on the left, the chosen one on the right. Changes apply as they are

made rather than waiting behind an OK button, and each pane says whether a setting takes effect straight away or seeds the next sprite or session.

SECTION	WHAT IT HOLDS
General	Theme, default size for new sprites, which panels start open, which side the tools sit on, the gutter between palette swatches
Editor	Default freehand algorithm, brush size and tolerance, bucket sampling, what the right button does, the hover outline
Selection	Whether pasting selects what it placed, whether the selection moves with its contents, whether cropping deselects
Timeline	Default frame duration, playback loop and rewind, onion skin range and opacity, the facing reference's silhouette colour and opacity
Background	Checkerboard on/off, cell size and colours, the backdrop behind the sprite and whether it follows the theme
Grid & Guides	Default grid cell size, grid and pixel-grid line colours, guide and locked-guide colours, the horizon and its vanishing-point lines
Undo	History depth and memory budget
Files	Recovery copies and how often, sprite sheet layout, whether exports bake in the pixel aspect ratio, whether to come back to the last workspace, recent files, default palette export format, the presets folder
Reset	Restore every setting to its original value

Keyboard shortcuts

Single-key shortcuts are suspended while a text field has focus, so typing a layer or area name never triggers one.

TOOLS

M	Marquee
A	Layer Selector
V	Move
T	Transform
W	Magic wand
B	Pencil
E	Eraser
L	Line
R	Rectangle
O	Ellipse
S	Shading
G	Paint bucket
Y	Text
I	Eyedropper
H	Hand

PAINTING

X	Swap colours
[]	Narrower / wider stroke
⇧P	Toggle Pixel Perfect
⇧	Pick a colour, any tool
⌘	Pick a colour (pencil, bucket)
⌘X	Clear
⇧⇧X	Fill with foreground
⌘⇧X	Fill with background

FILE

⌘N	New Sprite
⌘O	Open
⇧⌘O	Import Image
⌘S	Save
⇧⌘S	Save As
⇧⌘W	Close Document
⌘E	Export frame as PNG
⇧⌘E	Export frames as PNGs
⇧⌘G	Export animated GIF
⇧⌘R	Record / stop timelapse
⌘,	Settings

EDIT

⌘Z	Undo
⇧⌘Z	Redo
⌘X ⌘C ⌘V	Cut, copy, paste
⇧⌘T	Rotate and Scale
⇧⌘[	Turn Left
⇧⌘]	Turn Right

IMAGE

⌘P	Image Properties
C	Canvas Size
⇧⌘I	Image Size

LAYER

⇧P	Layer Properties
⇧X	Toggle visibility
⇧N	New Empty Layer
⌘J	Duplicate Layer
⇧⌘C	Copy Layer
⇧⌘V	Paste Layer
⇧⌘E	Merge Layers / Merge Down
⌘G	Group Layers
⇧⌘G	Ungroup
⌘] ⌘[	Bring forward / send backward

ANIMATION

⇧F	Frame Properties
⇧N	New Frame
⇧B	New Empty Frame
⇧D	Duplicate Cel
⇧C	Delete Frame
⇧I	Reverse Frames
⏮	Play / stop
⏮ ⏭	Previous / next frame
⇧⌘[⇧⌘]	Previous / next loop

SELECT

⌘A	All
⌘D	Deselect
⇧⌘D	Reselect
⇧⌘I	Inverse

VIEW

^H	Extras
⇧⌘P	Show perspective
⇧⌘R	Show reference image
⇧⌘D	Depth Preview
⌘;	Show guides
⇧⌘;	Lock guides
⌘R	Show rulers
F3	Onion skin
⇧F3	Facing reference
⌘+ ⌘-	Zoom in / out
⌘0	Actual size
⌘9	Fit in window
^⌘F	Full screen

What it does not do

Worth knowing up front, so you are not hunting for something that is not there.

- **One document at a time.** Multiple open documents are not implemented, so **Image ▶ Duplicate** reports that it is unavailable.
- **No tilemap layers,** no MSK files, and .ase/.aseprite documents cannot be opened. Palettes, sprite sheets and PNGs are the way in and out.
- **The bucket and wand sample the flattened image by default,** which differs from Aseprite — switch it in the options bar.
- **Reference images are not layers.** One image sits behind the canvas, moved with an -drag and remembered by path in the workspace. It never composites or exports, but there is only one of it and it cannot be stacked or reordered.
- **Masks are per-document, not per-frame.** A layer's mask applies to the frame being worked

on; a room that animates would need its masks to hold still, which nothing enforces.

- **The pivot is one per sprite,** not one per loop or frame. A walk that shifts its weight sideways wants a pivot per loop.
- **Palette presets are plain .gpl files,** so they interoperate with GIMP, Aseprite and Krita — but they do not round-trip swatch names.

IF SOMETHING GOES WRONG

Unsaved work is copied to the app's Recovery folder every couple of minutes, and the next launch offers to bring it back. A timelapse recording lives on disk rather than in memory, so a crash costs at most its last frame.