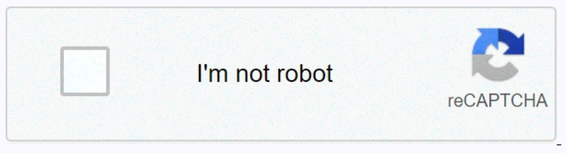
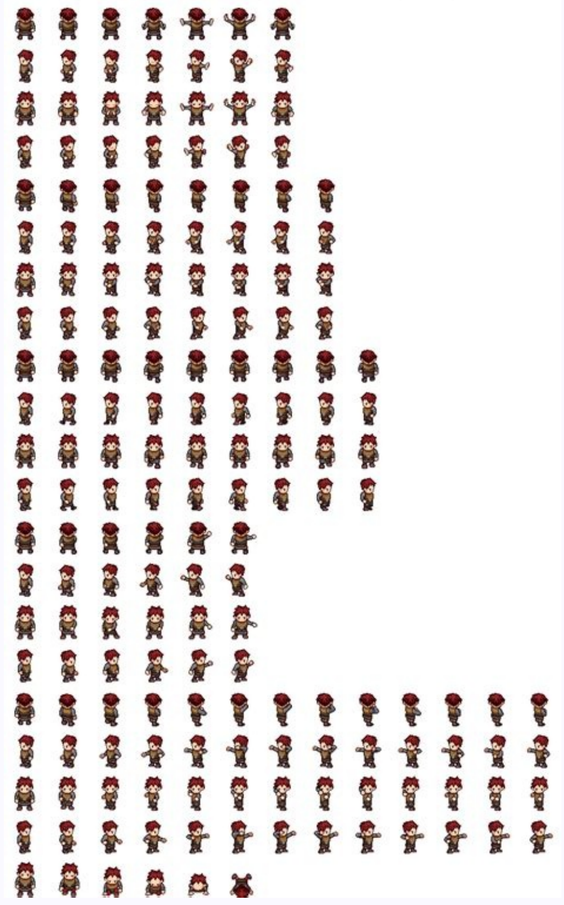


I'm not robot!

What is a sprite sheet

How does a sprite sheet work. How do sprite sheets work. Sprite sheet example. Why use sprite sheets. What is a sprite sheet in scratch.

Sprite sheets have already existed since the first days of computer games. The idea was to create one image that contains all images instead of many single files. Sprites can be used to create sprite animations, or to display individual image sprites separately while using only a single image source file. This is important for web development as it helps reduce page loading times and image file sizes. It's easy for a game to retrieve the sprites since they all have the same width and height.

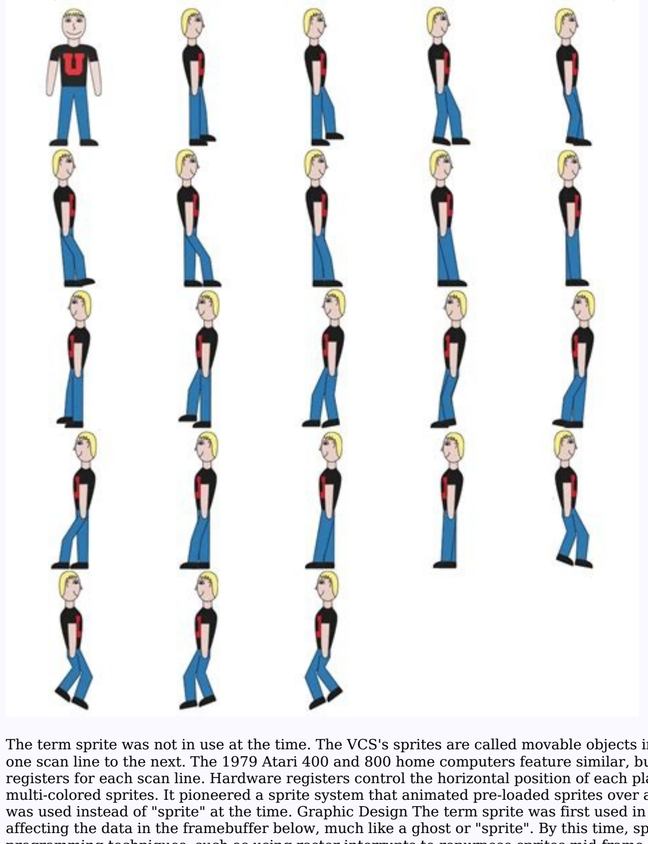


The disadvantage is that the sprites waste a lot of memory because of all the additional transparency. Types of Sprite Sheets [1] Sprite strips and animation strips An animation strip is the simplest form of a sprite sheet: It's just placing each animation frame next to each other. All frames have the same size, and the animation is aligned in each frame. A simple sprite strip or animation strip for a game character [2] Tile sets It might also contain multiple lines if the animation is longer - or if several different animations are combined. A tile set is not different from a sprite sheet: It contains building blocks for game levels. This is an example for a tile map: The system now needs CSS data to find the sprite. This information can be stored in a data file that is shipped with the sprite sheet. It contains the coordinates and names for the sprites. With these data files it's easier to tell the system that you want to draw a specific sprite — instead of paint the sprite at position (299,192). a) Creating a sprite sheet The easiest way to create optimized sprite sheets is using Sprite Sheet.

Sprite Sheet is a tool that specializes in creating sprite sheets. The free version allows you to create sprite strips and tile maps.

b) How to create a sprite strip Creating a sprite sheet requires nothing more than dragging your sprites onto Sprite Sheet: c) Upload and deploy Sprite Sheet takes all image files in the folder and packs the sheet for you. It supports a range of image formats.
You can then download the sprite sheet and the accompanying CSS file with the data to access the images on your website.
Sprite Sheet Optimization Sprite strips are a good start — but also a waste of memory in many cases. If you plan to create a game, you should optimize the sprite sheet. The developers of game engines are aware of the wasted memory in the simple sprite sheets and started to optimize the space. The easiest way is to remove the transparency surrounding the sprite and shrink it to the bounding box.
Originating in computer graphics, a sprite is a two-dimensional bitmap that is integrated into a larger scene, most often in a 2D video game. The term was first used by Danny Hillis at Texas Instruments in the late 1970s. Originally sprites referred to fixed-sized objects composited together, by hardware, with a background. This composition occurs as each scan line is prepared for the video output device, such as a CRT, without involvement of the main CPU and without the need for a full-screen frame buffer. Sprites can be positioned or altered by setting attributes used during the hardware composition process. The use of sprites originated with arcade games.
The first video game to represent player characters as human player images was Taito's Basketball, which was licensed in February 1974 to Midway, releasing it as TV Basketball in North America.
Other systems with hardware sprites include the TI-99/4A (1979), Atari 8-bit family (1979), Commodore 64 (1982), Nintendo Famicom / NES (1983), Amiga (1985), Sega Mega-Drive / Genesis (1988), and many coin-operated arcade machines of the 1980s. Sprite hardware varies in how many sprites are supported, how many can be displayed on a single scan line (which is often a lower number), the dimensions and colors of each sprite, and special effects such as scaling or reporting pixel-precise overlap. Use of the term sprite has expanded to refer to any two-dimensional bitmap used as part of a graphics display, even if drawn into a frame buffer (by either software or a GPU) instead of being composited on-the-fly at display time. Signetics devised the first chips capable of generating sprite graphics (referred to as objects by Signetics) for home systems.

The Signetics 2636 video processors were first used in the 1978 1292 Advanced Programmable Video System and later in the 1979 Elektor TV Games Computer. The Atari VCS, released in 1977, features a hardware sprite implementation where five graphical objects can be moved independently of the game playfield.



The term sprite was not in use at the time. The VCS's sprites are called movable objects in the programming manual, further identified as two players, two missiles, and one ball. These each consist of a single row of pixels that are displayed on a scan line. To produce a two-dimensional shape, the sprite's single-row bitmap is altered by software from one scan line to the next. The 1979 Atari 400 and 800 home computers feature similar, but more elaborate, circuitry capable of moving eight single-color objects per scan line: four 8-bit wide players and four 2-bit wide missiles. Each is the full height of the display—a long, thin strip. DMA from a table in memory automatically sets the graphics pattern registers for each scan line. Hardware registers control the horizontal position of each player and missile. Vertical motion is achieved by moving the bitmap data within a player or missile's strip. The feature was called player/missile graphics by Atari. The Namco Galaxian arcade system board, for the 1979 arcade game Galaxian, featured animated, multi-colored sprites. It pioneered a sprite system that animated pre-loaded sprites over a scrolling background, which became the basis for Nintendo's Radar Scope and Donkey Kong arcade hardware and home consoles such as the Nintendo Entertainment System. According to Steve Golson from General Computer Corporation, the term "stamp" was used instead of "sprite" at the time. Graphic Design The term sprite was first used in the graphic sense by one of the definers of the Texas Instruments 9918(A) video display processor (VDP). The term was derived from the fact that sprites, rather than being part of the bitmap data in the framebuffer, instead "floated" around on top without affecting the data in the framebuffer below, much like a ghost or "sprite". By this time, sprites had advanced to the point where complete two-dimensional shapes could be moved around the screen horizontally and vertically with minimal software overhead. Systems with Hardware Sprites These are base hardware specs and do not include additional programming techniques, such as using raster interrupts to repurpose sprites mid-frame. Computer system Sprite hardware Year Sprites on screen Sprites on line Max. texels on line Texture width Texture height Colors Hardware zoom Rotation Background Collision detection Transparency Amstrad Plus 1990 16 16 ? 16 16 15 1, 2, 4× vertical, 1, 2, 4× horizontal No 1 bitmap layer No Color key Atari 2600 TIA 1977 5 5 19 1, 8 262 1 1, 2, 4, 8× horizontal Horizontal mirroring 1 bitmap layer Yes Color key Atari 8-bit family GTIA/ANTIC 1979 8 8 40 2, 8 128, 256 1 1, 2× vertical, 1, 2, 4× horizontal No 1 tile or bitmap layer Yes Color key Commodore 64 VIC-II 1982 8 8 96, 192 12, 24 21 1, 3 1, 2× integer No 1 tile or bitmap layer Yes Color key Amiga (OCS) Denise 1985 Arbitrary 8 128 16 Arbitrary 3, 15 Vertical by display list No 2 bitmap layers Yes Color key Amiga (AGA) Lisa 1992 Arbitrary 8 512 16, 32, 64 Arbitrary 3, 15 Vertical by display list No 2 bitmap layers Yes Color key Colecovision Texas Instruments TMS9918A 1983 32 4 64 8, 16 8, 16 1 1, 2× integer No 1 tile layer Partial Color key Texas Instruments TI-99/4A Texas Instruments TMS9918A 1981 32 4 64 8, 16 8, 16 1 1, 2× integer No 1 tile layer Partial Color key Gameduino 2011 256 96 1,536 16 16 255 No Yes 1 tile layer Yes Color key Intellivision STIC AY-3-8900 1979 8 8 64 8 8,16 1 1, 2, 4, 8× vertical, 1, 2× horizontal Horizontal and vertical mirroring 1 tile layer Yes Color key MSX Texas Instruments TMS9918A 1983 32 4 64 8, 16 8, 16 1 1, 2× integer No 1 tile layer Partial Color key MSX2 Yamaha V9938 1986 32 8 128 8, 16 8,16 1, 3, 7, 15 per line 1, 2× integer No 1 tile or bitmap layer Partial Color key MSX2+ / MSX turbo R Yamaha V9958 1988 32 8 128 8,16 8,16 1, 3, 7, 15 per line 1, 2× integer No 1 tile or bitmap layer Partial Color key Namco Pac-Man(arcade) TTL 1980 6 6 96 16 16 3 No Horizontal and vertical mirroring 1 tile layer No Color key TurboGrafx-16 HuC6270A 1987 64 16 256 16, 32 16, 32, 64 15 No No 1 tile layer Yes Color key Namco Galaxian(arcade) TTL 1979 7 7 112 16 16 3 No Horizontal and vertical mirroring 1 tile layer No Color key Nintendo Donkey Kong, Radar Scope(arcade) 1979 128 16 256 16 16 3 Integer No 1 tile layer Yes Color key Nintendo DS Integrated PPU 2004 128 128 1,210 8, 16, 32, 64 8, 16, 32, 64 65,536 Yes, affine Yes, affine 4 layers per screen; each layer is independent No Color key, blending NES/Famicom Ricoh RP2C0x PPU 1983 64 8 64 8 8, 16 3 No Horizontal and vertical mirroring 1 tile layer Partial Color key Game Boy Integrated PPU 1989 40 10 80 8 8, 16 3 No Horizontal and vertical mirroring 1 tile layer No Color key Game Boy Advance Integrated PPU 2001 128 128 1210 8, 16, 32, 64 8, 16, 32, 64 15, 255 Yes, affine Yes, affine 4 layers, 2 layers, and 1 affine layer, 2 affine layers No Color key, blending Master System,Game Gear VDP(TMS9918-derived) 1985 64 8 128 8, 16 8, 16 15 1, 2× integer, 1, 2× vertical Background tile mirroring 1 tile layer Yes Color key Sega Genesis YM7101 VDP(SMS VDP-derived) 1988 80 20 320 8, 16, 24, 32 8, 16, 24, 32 15 No Horizontal and vertical mirroring 2 tile layers Yes Color key Sega OutRun (arcade) 1986 128 128 1600 8 to 512 8 to 256 15 Yes, anisotropic Horizontal and vertical mirroring 2 tile layers and 1 bitmap layer Yes Alpha Sharp X68000 Cynthia jr. (original), Cynthia (later models) 1987 128 32 512 16 16 15 1, 2× integer Horizontal and vertical mirroring 1-2 tile layers and 1-4 bitmap layers Partial Color key Neo Geo LSPC2-A2 1990 384 96 1536 16 16 to 512 15 Sprite shrinking Horizontal and vertical mirroring 1 tile layer Partial Color key Super NES/Super Famicom S-PPU1, S-PPU2 1990 128 34 272 8, 16, 32, 64 8, 16, 32, 64 15 Background only Horizontal and vertical mirroring 3 tile layers or 1 affine mapped tile layer Yes Color key, averaging In computer graphics, a sprite sheet is defined as a bitmap image file made up of numerous smaller graphics that are integrated into a tiled grid formation. This combination of several graphics into one file allows you to utilize them in Anime and various other applications by simply loading a single file. This efficiency can really come in handy in situations where performance is particularly vital, e.g. game development. To create a sprite sheet, you need to choose between any combination of bitmaps, graphic symbols, button symbols, and movie clips. Items can be selected either on the Stage or in the Library panel, but not both.

Each frame and bitmap of the symbols you have selected will appear in the sprite sheet as an isolated graphic. Exporting from the Stage will preserve any transformations you have made to the symbol instance (skewing, scaling, and so on) at the image output.

Steps for creating a sprite sheet Choose one or more symbol instances on the Stage or symbols in the Library panel. You can also include bitmaps in the selection. Right click on the selection, and hit Generate Sprite Sheet. A dialog box will appear – click options, then select Export Available export options Image dimensions This is the overall size of the sprite sheet in pixels. The default image dimension is usually set as autosize, which allows the sheet to work with any sprite you include. Border padding Padding, in pixels, around the sprite sheet's edges Image format This describes the file format of the sprite sheet you've exported. PNG 32 bit and PNG 8-bit both allow the use of a transparent background, while JPG and PNG 24-bit are not compatible with transparent backgrounds. The difference between PNG 32-bit and PNG 8-bit in terms of visual graphics is relatively insignificant. But when it comes to size, PNG 32-bit files tend to be four times as big as PNG 8-bit files. Algorithm This is the technique used to pack graphics into the sprite sheet, and comes in two alternatives: Data format This describes the internal format used to create image data, and is usually set as Starling as the default format.

Choose the most suitable format for your desired workflow after exporting into the sprit Rotate This option rotates the sprites ninety degrees, and is only available in certain data formats. Stack frames This option prevents replica frames within the symbols you have selected from being reproduced in the subsequent sprite sheet. Trim This function cuts off unused pixels from every symbol frame included in the sprite sheet in order to save space. Sprites have been occasionally used in movies for special effects. More specifically, in The Lord of the Rings: The Fellowship of the Ring, sprites were used to simulate fire coming out from the surface of the fire breathing Balrog demon. The term "sprite" may also be used to loosely describe any 2D image drawn on a computer, otherwise referred to as Pixel Art.