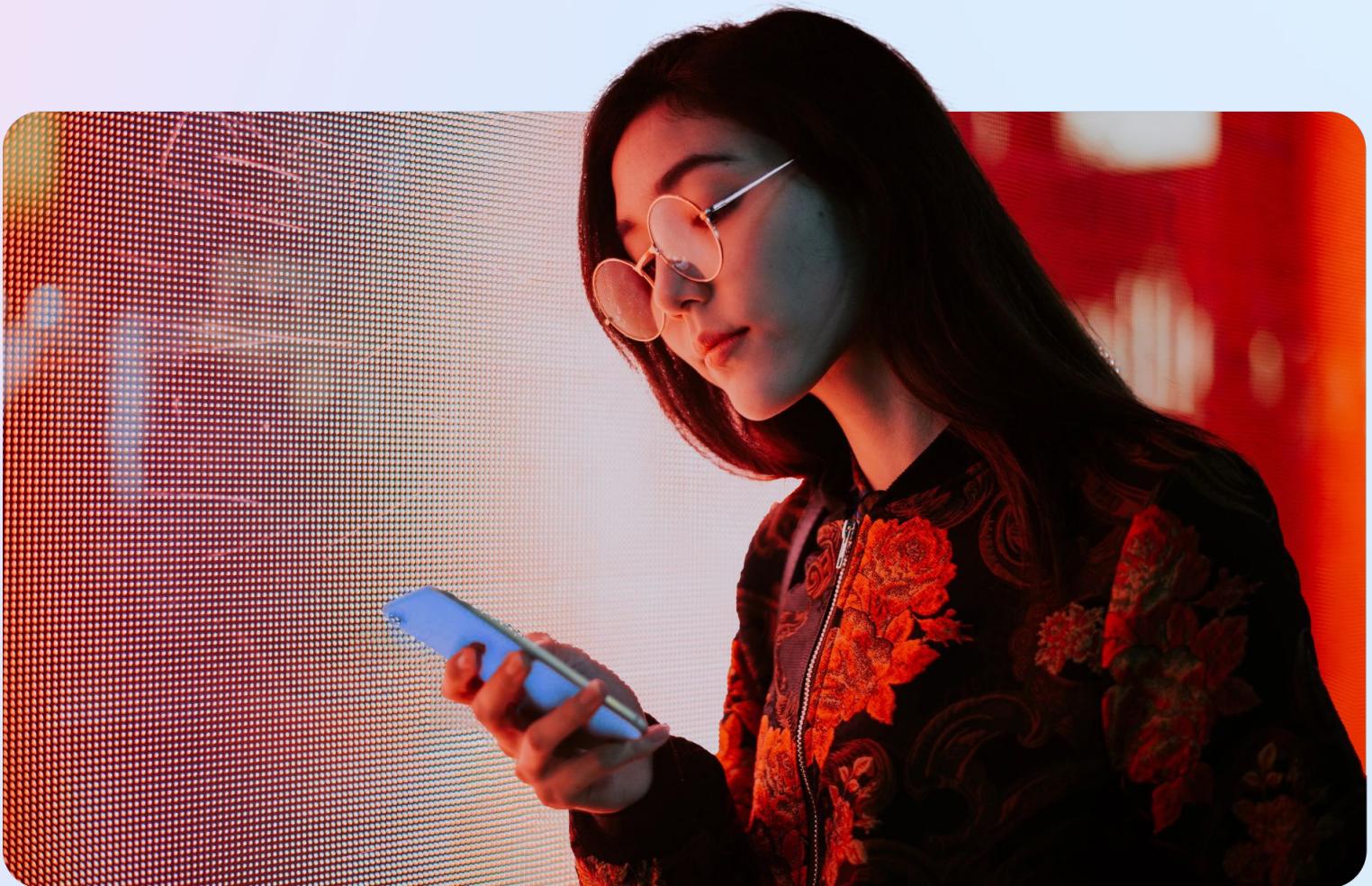


Meta Certified Meta Spark Creator Exam

Study guide



JANUARY 2023 VERSION

Table of contents

<u>Overview</u>	3
<u>Introduction to Meta Spark Studio</u>	4
<u>Design and conceiving</u>	7
<u>Features of Meta Spark Studio</u>	35
<u>Adding scene interactivity</u>	47
<u>Scene optimization</u>	59
<u>Publication and effect management</u>	68
<u>Glossary</u>	73

Overview



The goal of this study guide is to help you prepare to take the Meta Certified Meta Spark Creator Exam.



To make the most of your experience, we recommend that you use this study guide in conjunction with the latest version of [Meta Spark Studio](#) and the [AR Pro](#) course.

Learn more about [Meta Certification](#) and find answers to [frequently asked questions](#).

Meta Certification
Build on your professional skills. Grow your digital marketing career. Earn a Meta Certification.

Reach your professional goals on Facebook, Messenger, Instagram, and WhatsApp

Master digital marketing skills to level up your productivity, credibility, confidence and career.

The value of Meta Certification:

- Meta Certification improves confidence.
- Meta Certification proves expertise.
- Meta Certification promotes work efficiency.

ABOUT META CERTIFICATION

How do I become certified?

Our scaled scoring system ranges from 300-1000 w to pass. You can take the proctored certification exam VUE Professional testing center.

The certification expiration date will vary depending

What certifications are available?

Why should I consider getting Meta certified?

Was this information helpful?
 Yes No

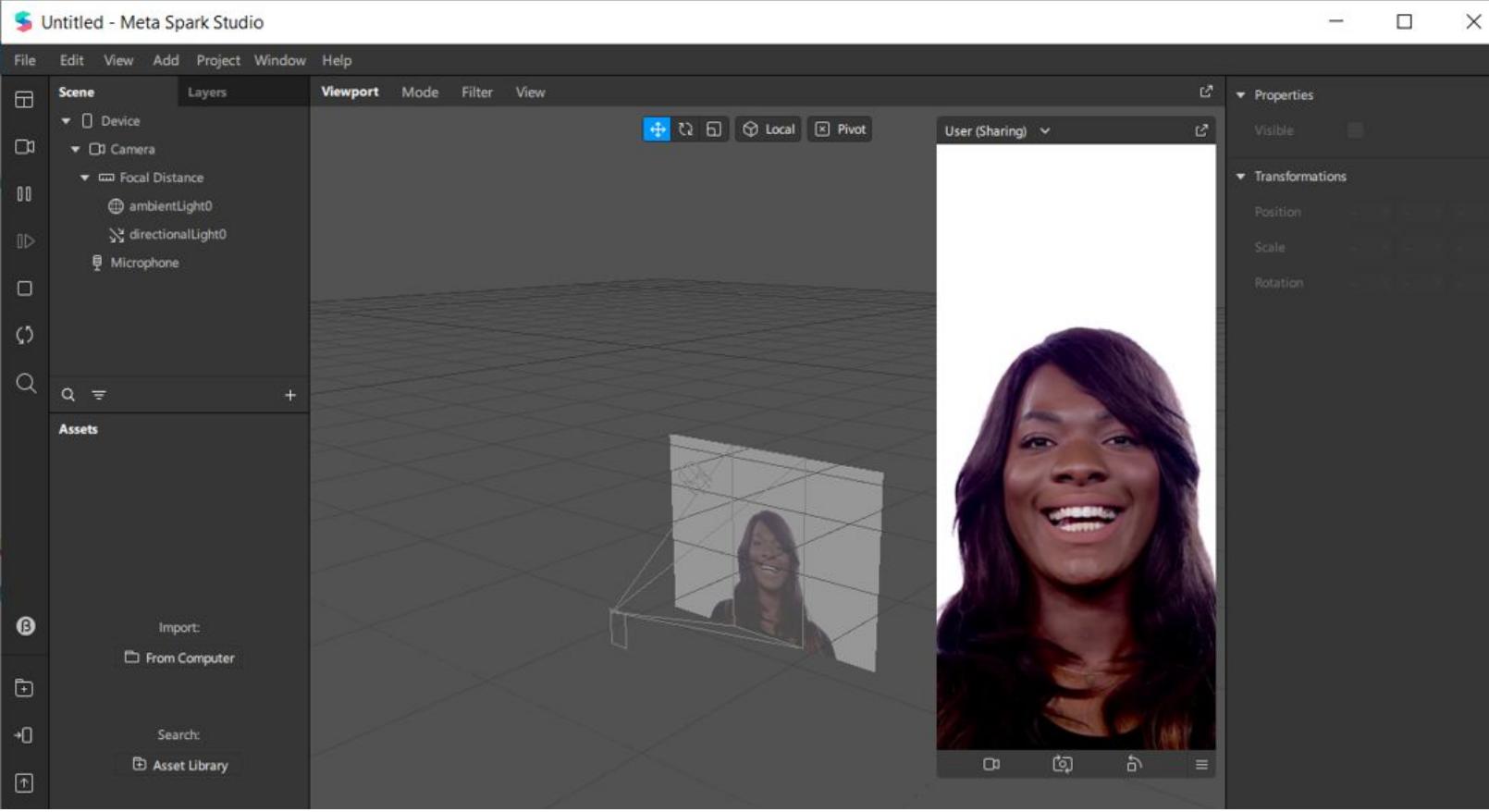
What's the difference between certification and earning badge?

What's the difference between an exam and a certification?

SCHEDULING YOUR EXAM

Introduction to Meta Spark Studio





What can you do with Meta Spark Studio?

Build with or without code.

You don't have to be an expert coder to create effects in Meta Spark Studio. You can use visual programming to easily add interactivity, animate objects and build logic into your effects. You can even make textures and materials to use in your scene. [Use Patch Editor](#) to create interactive experiences without scripting or [create effects using JavaScript](#).

Import objects and sounds.

Add your own sound files and 3D objects, browse the free library of assets or import high-fidelity models through our partnership with Sketchfab. [Add audio clips](#) or create distortion effects using the microphone in your mobile device. Import your own 3D objects, including ones with bones and joints and ones with animations.

Create and edit assets.

In Meta Spark Studio, you can create materials, textures and some 3D and 2D objects. You can use multiple texture inputs to apply a realistic look to the surface of your objects. You can also create and position 2D surfaces and apply textures to them.

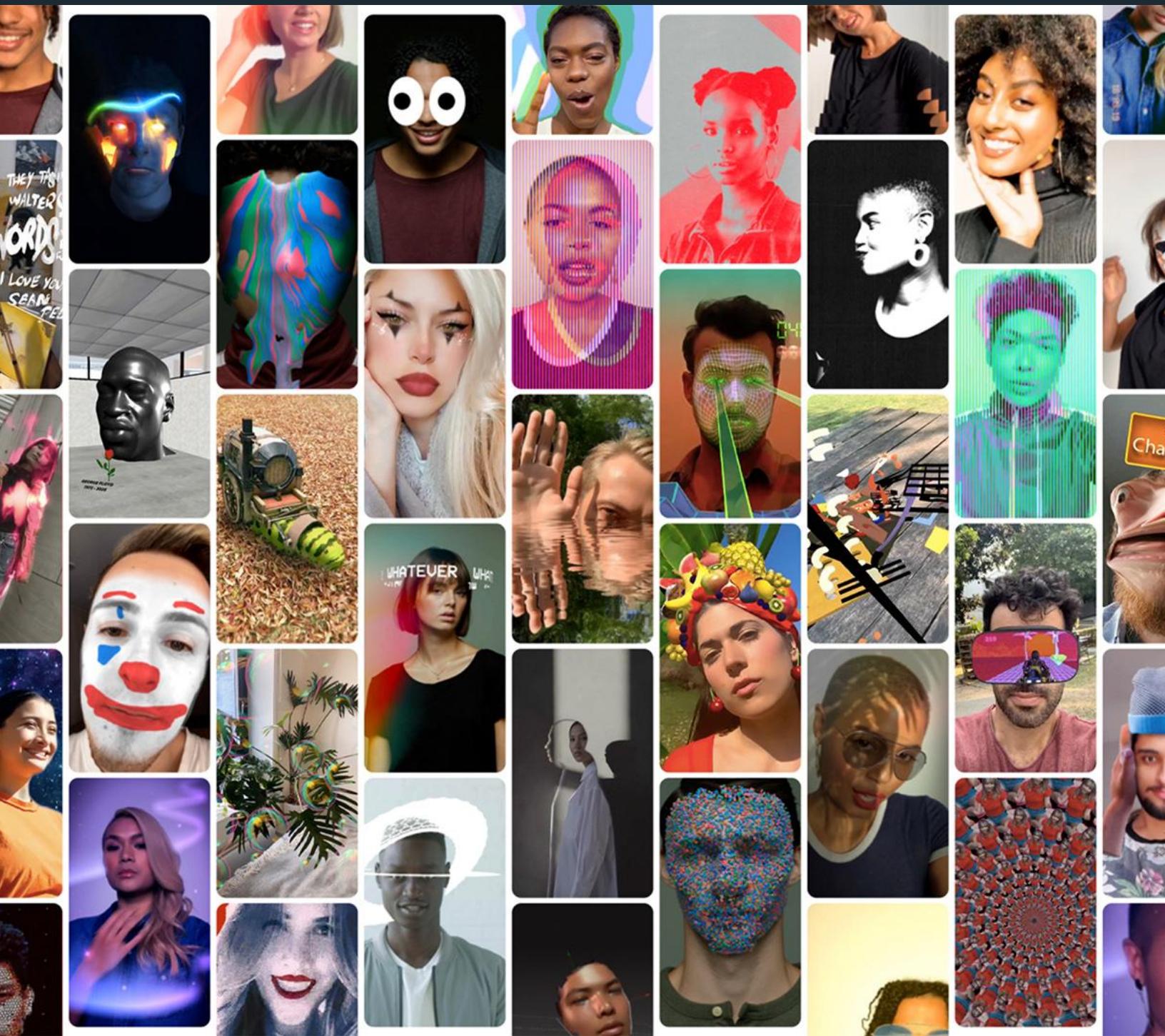
Develop unique effects.

Meta Spark Studio enables you to create a range of effects including effects that respond to a person's facial and body movements and augment a person's environment. There are many options for enhancing your effects. Add depth to objects in your scene with [five types of lighting](#) or [use particle systems](#) to generate, display and move particles in great numbers.

Test, publish and track effects.

[Use Meta Spark Player](#) to preview and test your effect. Meta Spark Hub helps you publish, manage and understand your AR effect performance across Facebook and Instagram. You can also use the [livestreaming module](#) to get information if somebody uses your effect when they go live on Facebook.

Design and concepting





Get started with Meta Spark Studio.

Whether you're a new learner or an experienced professional, Meta Spark Studio has all of the features and capabilities you need in order to create engaging effects for clients.

In this chapter you'll learn about the process of making an effect in Meta Spark Studio and the features you can use to create stunning AR experiences.

Tips for designing an effect



Focus on a solid use case.

Start by identifying a specific goal or task to achieve. For example, you may want to create an effect that expresses a feeling or addresses current news, topics or trends.

Make it social.

Consider adding support for multiple faces to your effect, so people can use it with friends.

Design for different device types.

Test your effect on multiple types and generations of mobile devices to ensure your effect works well.

Design for repeat use.

Engaging effects tend to be flexible. They can be used in different contexts and stay relevant over time.

Keep it simple.

Not all people are familiar with AR. You can take steps to manage expectations of an AR experience so people know what to do and how much effort to contribute. You can do this by:

- Adding clear instructions.
- Making interactions intuitive.
- Focusing on one or two main interactions so people can easily learn how to use the effect.

Add to the camera experience.

Avoid taking over lots of the camera view or making people completely unrecognizable. Allow people to keep recognizable elements of their environment or body to make the effect feel personal.

Avoid the edges of the screen.

If you add objects on the edges of the composition, they may be obscured by UI. Avoid placing objects too close to the edge of the camera screen if you can.

Give people something to capture with minimal effort.

People may move on from an effect in just a few seconds. Ensure that the main features of your effect happen as soon as possible.



Note that effects submitted to both Facebook and Instagram are reviewed to ensure they follow [Community Standards](#) and [Meta Spark Policies](#).

Design interactions that are obvious.

Use a common movement to prompt an obvious change, such as pinching the screen to make an object bigger or smaller.

Avoid using gestures that conflict with system-level commands.

System-level commands are interactions that people are already familiar with on their mobile devices. If you change how these commands work, it can cause confusion and frustration.

System-level commands include:

- Switch camera view: Double tap.
- Camera zoom: Pinch in or out, except when resizing stickers or 3D objects.
- Video zoom: Touch and hold camera capture button, then drag up.
- Return to Facebook Feed: Swipe left.



As you scope for your client's effect, consider accessibility concerns. For example, if your effect is going to contain flashing lights, add a warning so people know what to expect.

Objects and assets

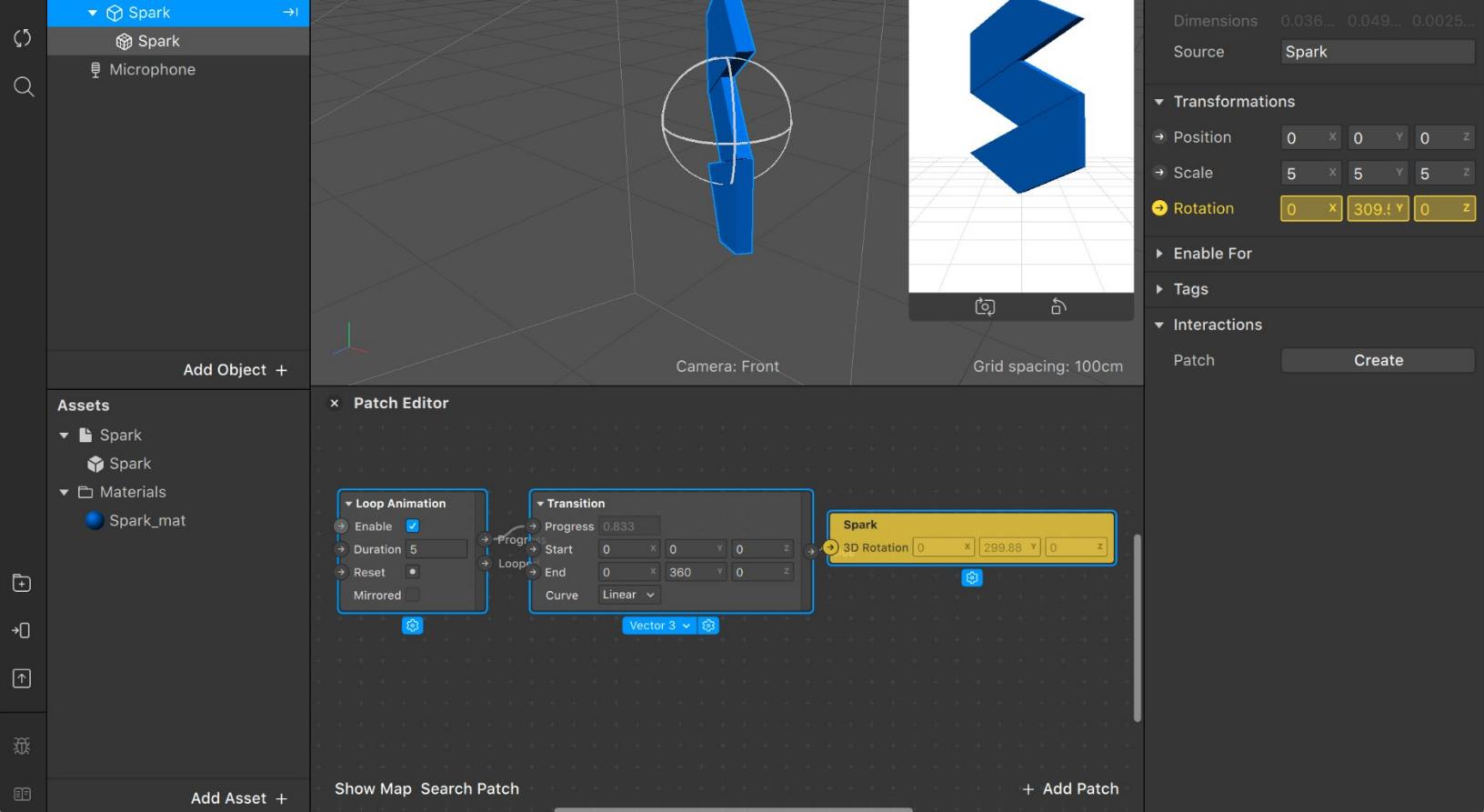
You can use the many objects available in Meta Spark Studio to add a variety of elements to your effect. Some of these include 2D and 3D objects, sounds, typography and lights.

You can make assets in Meta Spark Studio, or you can import them from your computer and add them to your project through the assets panel. You can import one or more assets at a time. Then you can see them listed in the assets panel. When you select an asset, you can edit its properties in the Inspector.

File formats

Meta Spark Studio supports various file formats depending on the type of object or asset you're importing:

2D assets	<ul style="list-style-type: none">● PNG● JPEG● SVG
3D models	<ul style="list-style-type: none">● FBX 2014/2015 (binary and ASCII versions)● glTF 2 (binary and text versions)● COLLADA● DAE● OBJ <p>The following features are supported for 3D models:</p> <ul style="list-style-type: none">● 3D scene● Materials● Textures● Animations targeting a model's position, rotation and scale
Audio	<ul style="list-style-type: none">● Mono M4A● AAC codec, with a sampling frequency of 44.1KHz
Fonts	<ul style="list-style-type: none">● TrueType● OpenType



Patch Editor

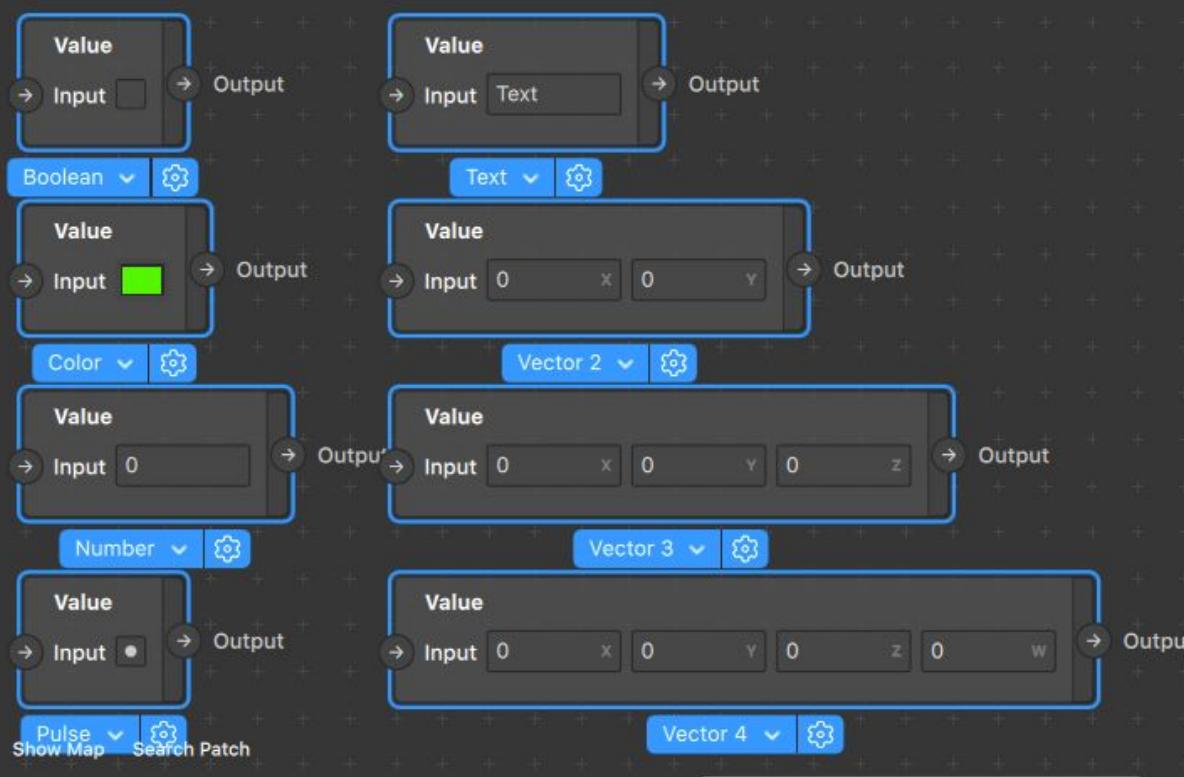
Patches in Meta Spark Studio enable you to add interaction, animation and logic to effects. You can also use patches to create materials. Patches act as visual building blocks, with each patch performing a different function. Connected together in a graph, patches pass and receive information to and from one another to make things happen in the effect. There are many capabilities available in scripting that you can add to your effect as patches, allowing you to add complexity without knowing how to code.

Most scene objects and assets can be represented as patches. You can also create patches to represent an object or an asset's properties in the Inspector. Use these patches to add logic, animation and interactivity to your project.

Patch Editor data types

Boolean signal	At any point in time, a boolean signal is either true or false. For example, the openness port in the mouth open patch can tell whether a mouth is open or closed.
Scalar	A scalar is a one-dimensional value. The openness port in the mouth open patch has a scalar value because it reflects how open the mouth is. If a mouth is wide open, the value might be 1. If it's half open it might be 0.5.

Pulse	A pulse is an event that happens at some point in time. It's not a continuous signal like a boolean, which is always true or false.
3D point	A 3D point is a three-dimensional signal that can represent a rotation, scale or position. For example, a patch representing the position of a 3D object has a 3D position port. This port has a 3D point data type, because the object is positioned in 3D space.
2D point	A 2D point is a two-dimensional signal that can represent a rotation, scale or position. Patches with a 2D point type have X and Y values.
Color	This type represents the RGBA channels, red, green, blue and alpha. It's indicated by a square with a color.
Progress	Progress is a scalar signal that's restricted to the 0–1 range. It could be used to drive an animation. Ports with progress types are labeled Progress , such as in the Animation patch.
Vector	VectorN is a multidimensional value consisting of N scalar components. <ul style="list-style-type: none"> • Vector2 for 2D points and 2D texture coordinates • Vector3 for 3D points, 3D normals and 3-channel (RGB) color • Vector4 for 4D position and 4-channel RGBA color
Matrix	MatrixN represents a matrix of N by N scalar components. <ul style="list-style-type: none"> • Matrix2 to represent a rotation transformation for 2D vectors • Matrix3 to represent a rotation transformation for 3D vectors or general affine transformation for 2D vectors with an additional homogeneous coordinate, which effectively makes them Vector3 • Matrix4 to represent a general affine transformation for 3D vectors with an additional homogeneous coordinate, which effectively makes them Vector4
Texture 2D	Texture2D typically represents an image and can have one to four channels.



Add Patch

Value patch data types

When a port has a box next to it, you can edit the value associated with that port. Some patches are able to switch their data types to accommodate the needs of a creator, as in the case of the value patch that can be used as a color or a number.

You can choose different data types for some of the patches. For example, if you're manipulating the shape of a 2D object, set the data type of certain patches to **Vector 2**. If you're applying a color to a material, change the data type of certain patches to **Color**.

In the example above, the UI of the number data type default value is 0 and it expects a number input. The UI of the boolean data type is a checkbox, which should be ticked if the input is true and unticked if the input is false.

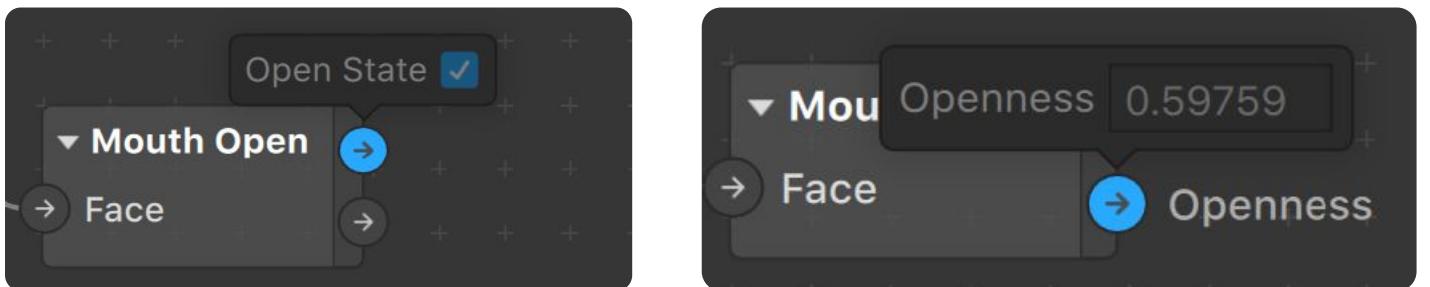


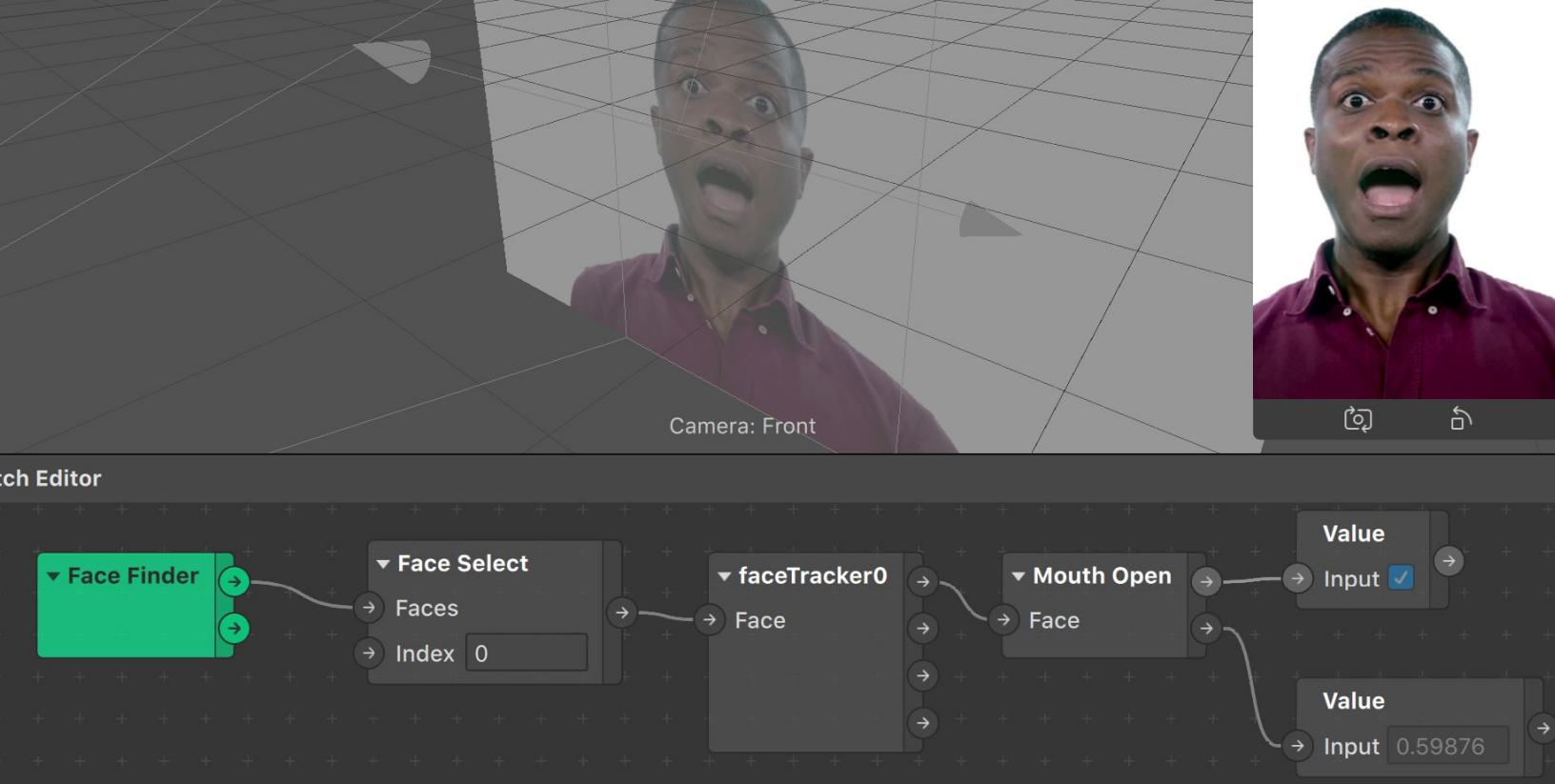
Meta Spark Studio UI changes according to the data type. In the image above, notice the different UI for each value patch. This process can be helpful when you create the patch graph and need to figure out what patches come next. It can also help you learn how to debug with patches.

In the example below, the mouth open patch has two outputs: the open state output, which is boolean and either true or false, and the openness output, which is a number from 0 to 1 that indicates how open the mouth is.



An intuitive way to create the patch graph is to look at the UI data type of the output or to use a value patch to debug and to understand the connection flow .

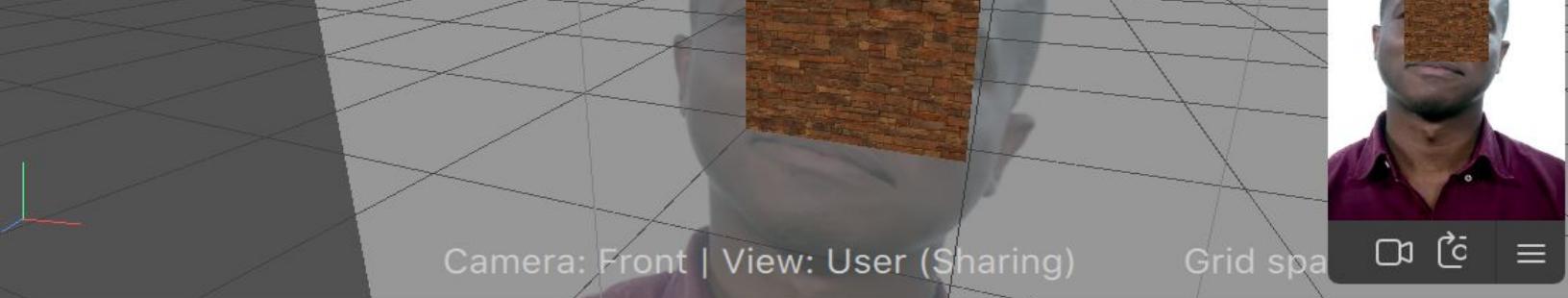




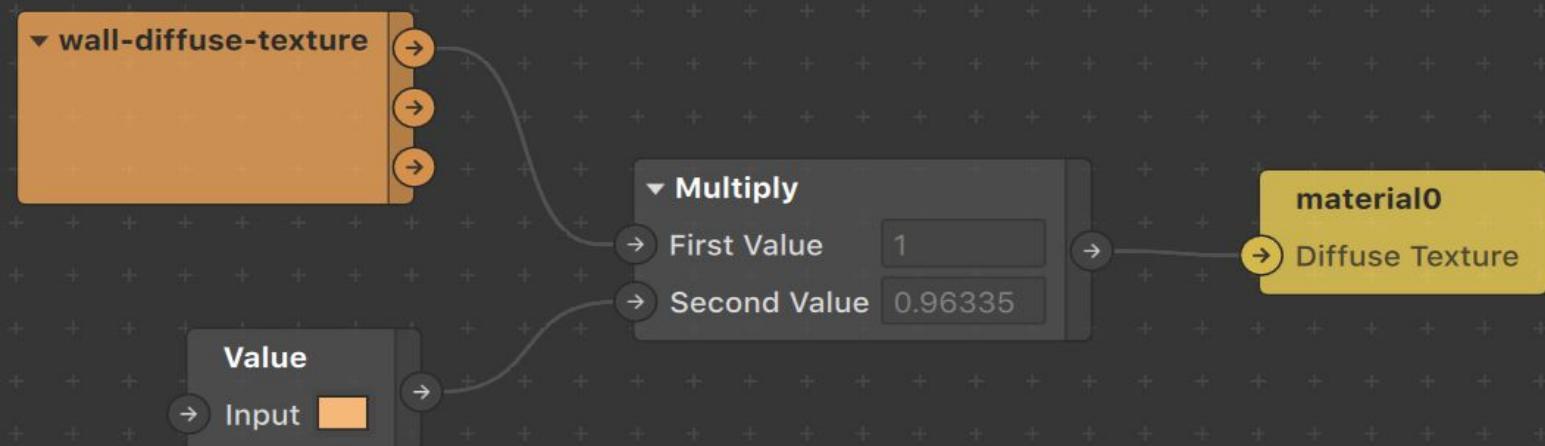
What are patches?

Patches allow you to add interaction and animation to your effects. Each patch performs a unique function and can pass and receive information to and from other patches. There are many types of patches to choose from in Meta Spark Studio:

<u>Interaction</u>	Interprets a person's movements, facial expressions and screen interactions to produce an action.
<u>Face landmark</u>	Precisely captures the position of facial features.
<u>Facial movement</u>	Detects facial expressions such as a happy, kissing or surprised face.
<u>Animation</u>	Drives an animation triggered by an interaction or logic patch.
<u>Audio player</u>	Manages and controls audio assets.
<u>Shaders</u>	Used for working with textures and materials.
<u>Math</u>	Performs mathematical functions.
<u>Logic</u>	Performs logical functions.
<u>Utility</u>	Performs a function depending on the type of patch. A range of patches, including face trackers and random number generators, are utility patches.
<u>User interface</u>	Adds UI functions to your effects.



Patch Editor



A visual shader in the Patch Editor

What is a shader?

A shader is a set of algorithms that determines the appearance of a 3D object's surface. Shaders can be useful when applying complex changes to a material. Experienced creators can use [shader code asset](#) to write custom shaders in Meta Spark Studio.

You don't need to know how to code to create a shader. Use Patch Editor to create and connect a group of patches called a visual shader.

How to use visual shaders

You can use visual shader patches with other patches in a graph to change the appearance of a material. You can apply more complex changes with visual shader patches than you can with built-in shaders found in the Inspector.

Types of visual shader patches

Consumer	Visually represents a property of a material, such as position or scale, and appears at the end of a graph to specify how the material appears in the effect.
Texture asset	A producer patch that applies a texture to a material.
Blend	Blends two colors or textures together.
Color space	Outputs a texture's RGB (red, green, blue), HSV (hue, saturation and value), and HSL (hue, saturation and lightness) color values. Useful for isolating color values to change a texture's appearance.
Composition	Outputs two combined functions as a single data stream.
Fallback	Automatically outputs a fallback value when you disconnect the main value. For example, you can use a color as a fallback value for an image.
Fragment stage	Processes each fragment in a texture individually instead of processing a texture at the vertices.
Function	Used as an identity function, always returns the same value as its argument.
Gradient	Creates a grayscale gradient, can be combined with a Mix patch to create a color gradient.
Gradient step	Defines each color you want to use in a color gradient, can be used multiple times to add multiple colors.
Render target size	Outputs the render target size and is useful for building effects using shaders requiring the exact pixel size, such as pixel blurring.
Shader derivative	Captures standard derivatives.
Texture sampler	Samples a texture at the specified coordinates.
Texture transform	Used with a 2D Transform Pack patch to scale, rotate, pivot and reposition a 2D texture on a 2D or 3D object.
Vertex attribute	Provides the value of the selected vertex attribute among a selection of parameters.
Vertex transform	Provides a vertex transformation matrix from a dropdown.

How to use data types

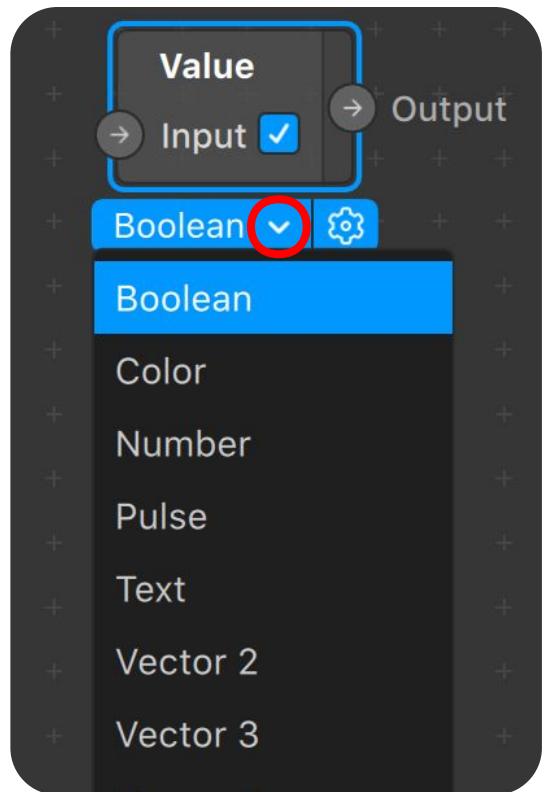
Every patch in Meta Spark Studio has one or more ports. You can send and receive data by connecting ports to other patches. You can also change the data types of some ports. To do this, select the patch and click the dropdown menu.

When you draw a connection from an output port, Meta Spark Studio suggests patches that you can connect. If you try to connect ports with incompatible data types, you'll see an error message.

Some data types are specific to Meta Spark Studio. These ports can be connected only to other ports of the same type.

For example, the face port in the face tracker patch outputs data on the position of the face. It can be connected only to another face port, such as in a face landmark patch or an interaction patch.

To learn more about how you can use patches to make your effect more interactive, see [Adding scene interactivity](#).



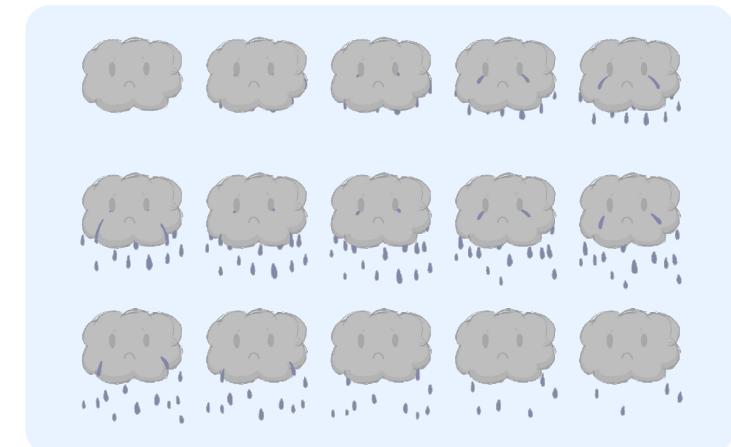
Value patch data types

Objects

2D objects

2D objects can be images, text, video or audio elements that are superimposed on the real world. For example, fonts, animations and [sprite sheets](#) are all 2D objects.

Sprite sheets are single image files that contain multiple images within them, each a frame in a 2D animation.



Sprite sheet

3D objects

When creating an effect in Meta Spark Studio, you often place 3D objects within the real world, so it's important that these objects represent a realistic and uniform scale. For example, if your effect enables people to place items from a furniture catalog into their living space, the 3D furniture should be realistically scaled.

There are two ways to enable real-world scale in Meta Spark Studio, depending on the type of user experience you want to create:

Enable real-world scale in the Inspector.	Enable real-world scale in the Inspector and add further logic via Patch Editor or scripting.
<ul style="list-style-type: none">Object appears realistically scaled only after a real-scale signal is detected.Viewers see the object abruptly change size at some point after the opening effect.	<ul style="list-style-type: none">Hides the object until a real-scale signal is detected.Viewers only see the object once it's realistically scaled.

The number of triangles in the mesh of a 3D object impacts performance, so minimize the triangle count. Meta Spark Toolkit for Blender allows creators to optimize their 3D objects. The maximum number of triangles per object should be below 50,000. Keep the total triangle count for all objects in an effect below 150,000.

How to select an effective target

Not all images make effective targets. An ineffective target image could mean that your effect won't respond to it.

To ensure that the camera will detect your target, follow a few basic principles:

Make it high contrast.	Both color and black-and-white images make for effective targets, as do all images with high tonal contrast. Avoid images with lots of pastel colors.
Make it high resolution.	An image should have a resolution of at least 300 x 300 pixels.
Make it sharp.	Avoid smooth or soft edges and textures with a lot of gradients.
Make it asymmetrical.	Avoid symmetry and repeated patterns. Images that can easily be inverted might not track as well.
Make it complex.	Simple images might not work well. You should also avoid repeating the same motifs.
Make it flat.	Images on curved surfaces, such as bottles, don't work well, as targets are meant for flat surface images.
Make it clear.	Target images can be many shapes. They don't have to be only rectangular or square. However, you should still consider how clear the target image is.
Avoid blank space.	Avoid targets that have a lot of transparency or blank space around the actual image.
Focus on placement.	Target AR effects work best when the camera is closer to the image so it fills a large portion of the device frame. Large targets also work better. Small images like stickers might not work even when in the full frame.

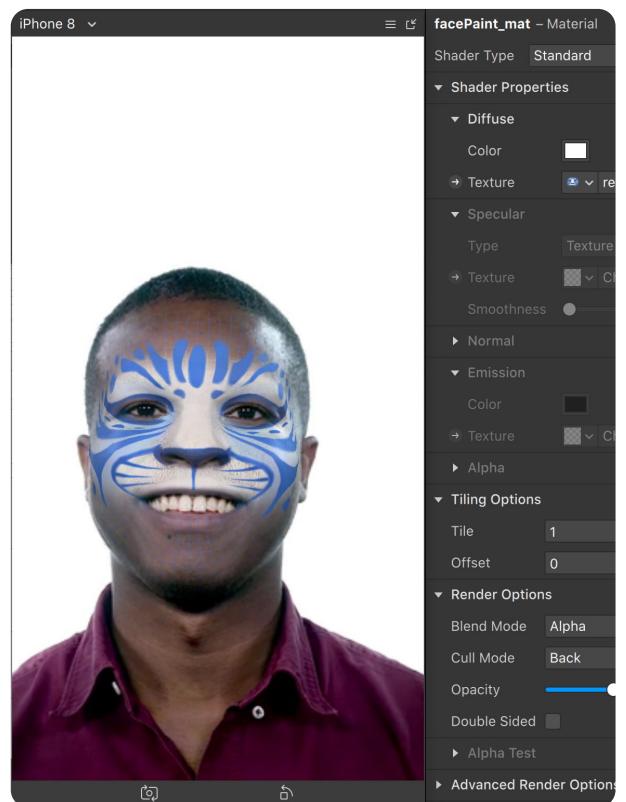
Materials

You can add materials to objects to help make them more vivid. There are four materials that you'll primarily work with in Meta Spark Studio.

Standard material

The standard material adds realistic depth to 3D objects using a lighting system called the Phong model.

When you create a material in Meta Spark Studio, it will often be a standard material unless you create it for a 2D object.



An effect that uses the standard material as the shader type

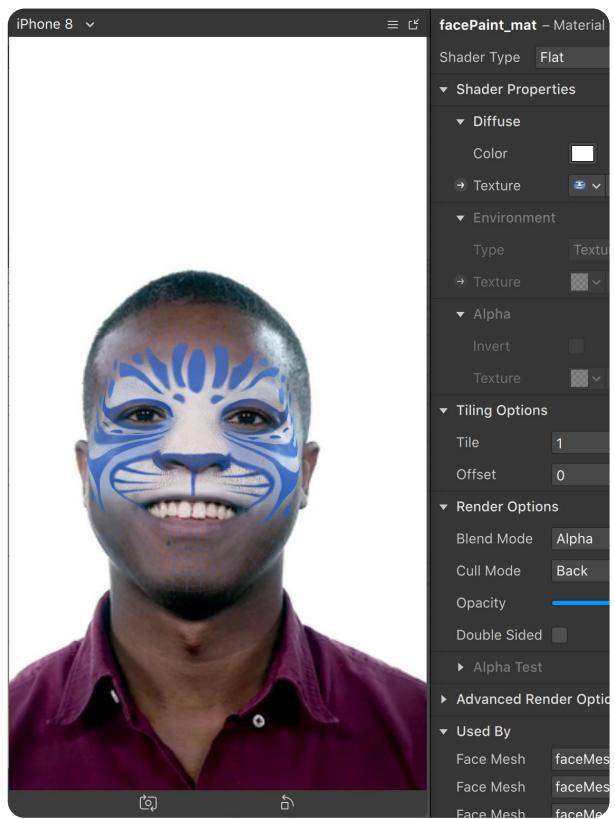
Standard material properties

Diffuse	Defines the base color and pattern of an object. Use color and texture together to create variations in the texture map.
Specular	Defines the shininess and highlight color of a surface.
Normal	Creates realistic textures such as bumps, grooves and rivets without adding extra geometry to your object.
Emission	Allows your material to illuminate itself without casting rays, so the illumination won't affect other objects in the scene.
Alpha	Masks the alpha channel. Use Invert to switch which part of the texture is masked.
Tile	Scales the textures you've applied to your material.
Offset	Shifts the origin of your textures.
Render options and advanced render options	Control how the material renders in the scene.
Used by	Checks which object this material is applied to.

Flat material

Flat materials don't respond to lighting, and display color and texture values absolutely.

You might use this material if there's lighting and shadow present in your texture file already or if you're creating a material for a 2D object.



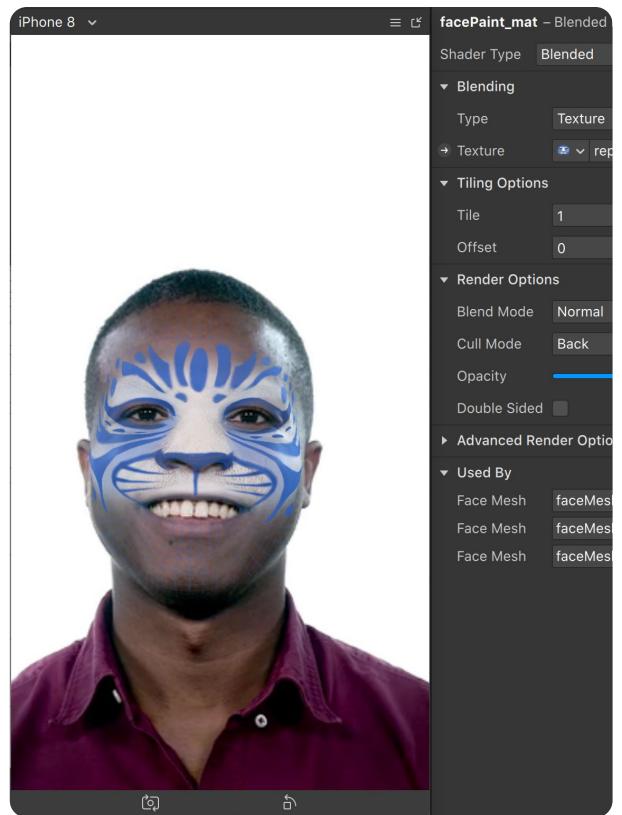
An effect that uses the flat material as the shader type

Flat material properties

Diffuse	Edit the color option to add a color from the palette to your material.
Environment	Check this box to add an environment texture to your material.
Alpha	Use this option to mask the alpha channel. Use Invert to switch which part of the texture is masked.
Tile	This property scales the textures you've applied to your material.
Offset	Offset shifts the origin of your textures.
Render options and Advanced render options	Edit these properties to control how the material renders in the scene .
Used by	Check which object this material is applied to.

Blended material

The blended material includes a blending mode to mix textures and colors together.



An effect that uses the blended material as the shader type

Blended material properties

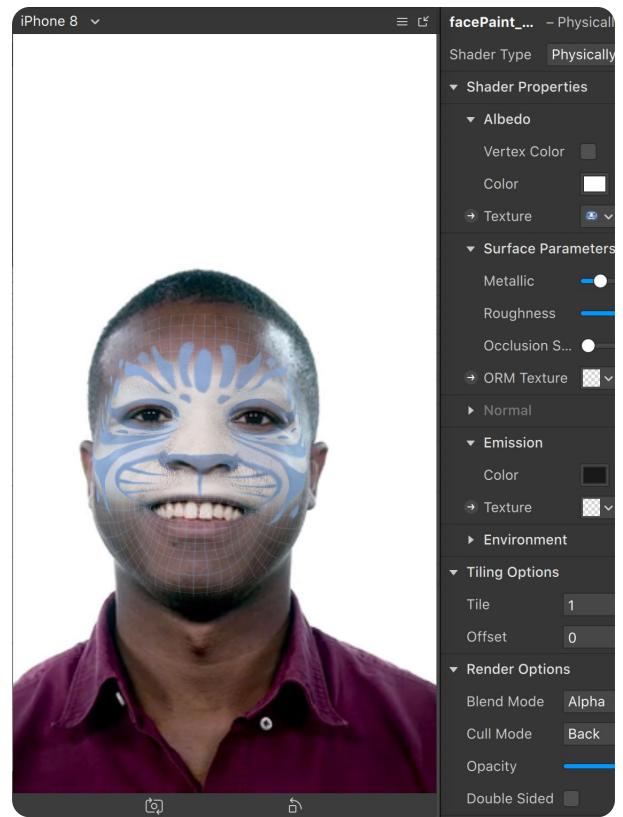
Texture	Choose the texture file you want to add from the menu or import a new one.
Color	Add a color from the palette.
Blend mode	Use this to set the blend mode of the material: normal, add or multiply.
Opacity	Sets the transparency of the material on an individual map level.
Double-sided	Displays both sides of a face normal.
Used by	Check which object this material is applied to.
Cull mode	This mode describes how material is culled. <ul style="list-style-type: none">Front: discards pieces of the mesh that are facing the viewerBack: discards pieces of the mesh that are facing away from the viewer Most effects use the back cull mode to improve performance.

Physically-based material

This material is used to create realistic objects in Meta Spark Studio. The different properties in this material allow you to add surface roughness, metallic effects and lighting that mimics real-world light.

For a complete physically-based material, you need:

- A base texture:
 - Forms the base of your material.
 - Can be used to add color and details.
- An ORM texture:
 - Allows you to define the roughness, metallic details and occlusion strength of the material.
- A normal map:
 - Creates the appearance of real-world texture such as bumps, grooves and rivets without adding extra geometry to your object.



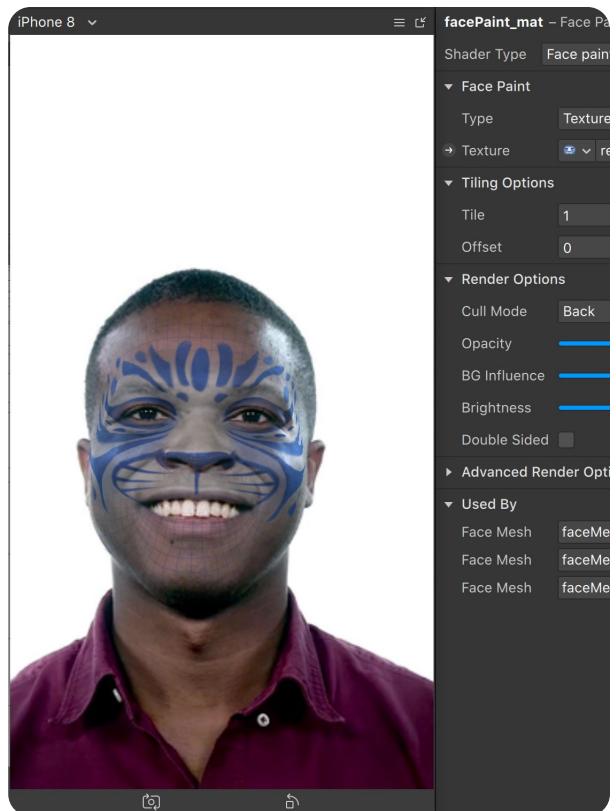
An effect that uses the physically-based material as the shader type

Physically-based material properties

Albedo	This is the base color and texture of the material. It offers two options: <ul style="list-style-type: none">• Color: Use this option to choose the base color of your material.• Texture: Apply a base texture, which will be modulated by the color you've selected.
Surface parameters	You apply an ORM texture in the dropdown menu. You can adjust the slider to control: <ul style="list-style-type: none">• Occlusion: This option approximates soft shadows baked into the creased areas of a surface.• Roughness: Use this option to add rough surface details.• Metallic: This option adds metallic surface details.

Face paint material

The face paint material is used to create a mask effect that shows the skin and features of the person using the effect, either behind a custom texture or a color. For example, you can use this material to create a tattoo or makeup effect.



An effect that uses the face paint material as the shader type

Face paint material properties

Texture	Choose the texture file you want to add from the menu or import a new one.
Color	Add a color from the palette.
Tile	Tile scales the textures you've applied to your material.
Offset	Offset shifts the origin of your textures.
Render options	Edit these properties to control how the material renders in the scene .
Used by	Check which object this material is applied to.

Tips for creating materials:



1. Use as few materials as possible.
2. Use the [face mask template](#) in Adobe® Photoshop® to create assets.
3. Use the physically-based material and environment textures.

Blend mode

The blend mode enum describes how material is blended.

Values	
Subtract	This option subtracts pixel values of the source from the destination. In the case of negative values, black is displayed.
Replace	The non-transparent pixels of the source replace corresponding pixels of the destination.
Alpha	Alpha describes the linear interpolation between the pixels of the source and destination as specified by the alpha values of the source.
Associate_Alpha	This is the same as alpha blend mode, but for a source with RGB channels of each pixel premultiplied by its alpha.
Add	This option adds pixel values of the source to the destination.
Multiply	This option multiplies the RGB channels of each pixel from the source with the values of the corresponding pixel from the destination.
Screen	Values of each pixel in the source and destination are inverted, multiplied and then inverted again.

Skin retouching

Skin retouching is an effect that can be applied when we want to give a smoother appearance to a face or scene using a material in Meta Spark Studio.

Textures

A texture is an image file used to help define the appearance of an object in your scene. You can apply one or more textures to a material, then apply the material to an object.

Texturing with realistic lighting is difficult to achieve in AR effects, and it takes a lot of processing power. You can take the following steps to reduce the impact on performance:

Bake lighting directly into 3D objects used for decoration in face effects to hint at highlights and shadows.	Use MatCap, material capture, reflection environment textures with flat shaders.
Bake lighting into static, nonmoving objects such as furniture and accessories for a more realistic effect.	Use a soft light setup without strong highlights or shadows. Baked lighting works best in environments where the light, intensity and color don't change throughout the effect.

You can make a variety of textures in Meta Spark Studio. For example, you can turn the video captured by the camera or the face tracker into a texture. You can also use visual shaders to create color gradients, and signed distance fields, SDF, to create shapes and patterns. You can even use the segmentation texture to separate the person from their background to transform the environment behind them.

File size for textures

Compressing textures can make your effects perform better. Look for opportunities to compress textures in your effects where you can.

When working with textures, consider the following:

	File type	Meta Spark Studio supports PNG and JPG texture files.
	File size	Use textures in sRGB space with a maximum resolution of 1024 x 1024 pixels.

	Compatibility	Use Meta Spark templates to make textures compatible if you've created them using Adobe® Substance Painter® texturing software. Delete unused textures from the project to save space.
	Merging materials	Merge materials as much as possible for performance benefits.
	Repeating patterns	For textures with a repeating pattern, import a lower resolution image and use Meta Spark Studio's visual shaders and texture tiling feature instead of using a larger resolution image.
	Transparency	When transparency isn't needed, use lower resolution JPG files instead of PNG files.

How to integrate assets into a scene

How you add an asset to your scene, whether by using Patch Editor or canvas, depends on the type of asset it is.

Textures	Audio files	3D objects
Textures, including animated textures, are applied to material. You then apply the material to an object in the scene using the material option in the Inspector.	Audio files should be connected to an asset called an audio playback controller, which can then be connected to an object called a speaker.	3D objects can be dragged from the assets panel to the Viewport or scene panel.

If a 3D object includes an animation, you can add the animation using the animation property of the 3D object. You can import all kinds of assets into Meta Spark Studio, such as textures you've made in external programs and audio files.

The Manipulator

The Manipulator displays three icons at the top of the Viewport.



You can use the Manipulator to quickly change your objects:

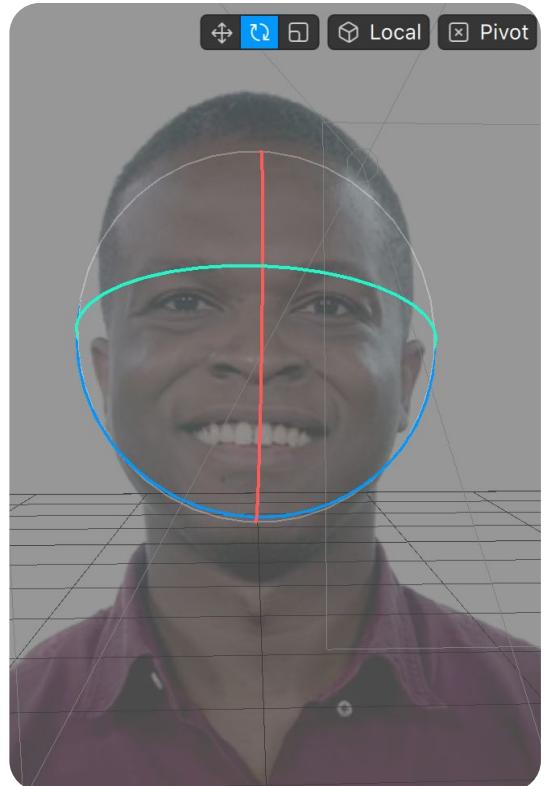
- Position determines where your object is placed within the scene.
- Scale makes your object appear larger or smaller.
- Rotation rotates your object.

Position, scale and rotation are known as transformations. You can also edit these properties in the Inspector.

When an object is a child of another object, it can be transformed in relation to the coordinates of its parent object. These coordinates might not be the same as the global coordinates. This is called local. An object can also be transformed based on the fixed X, Y and Z axis. This is called global. You have the option to make a patch for the object in the scene by clicking **Create**, which is located next to **Patch** in **Interactions**.

For large changes to an object, use the simulator as it allows you to drag entire objects across the screen.

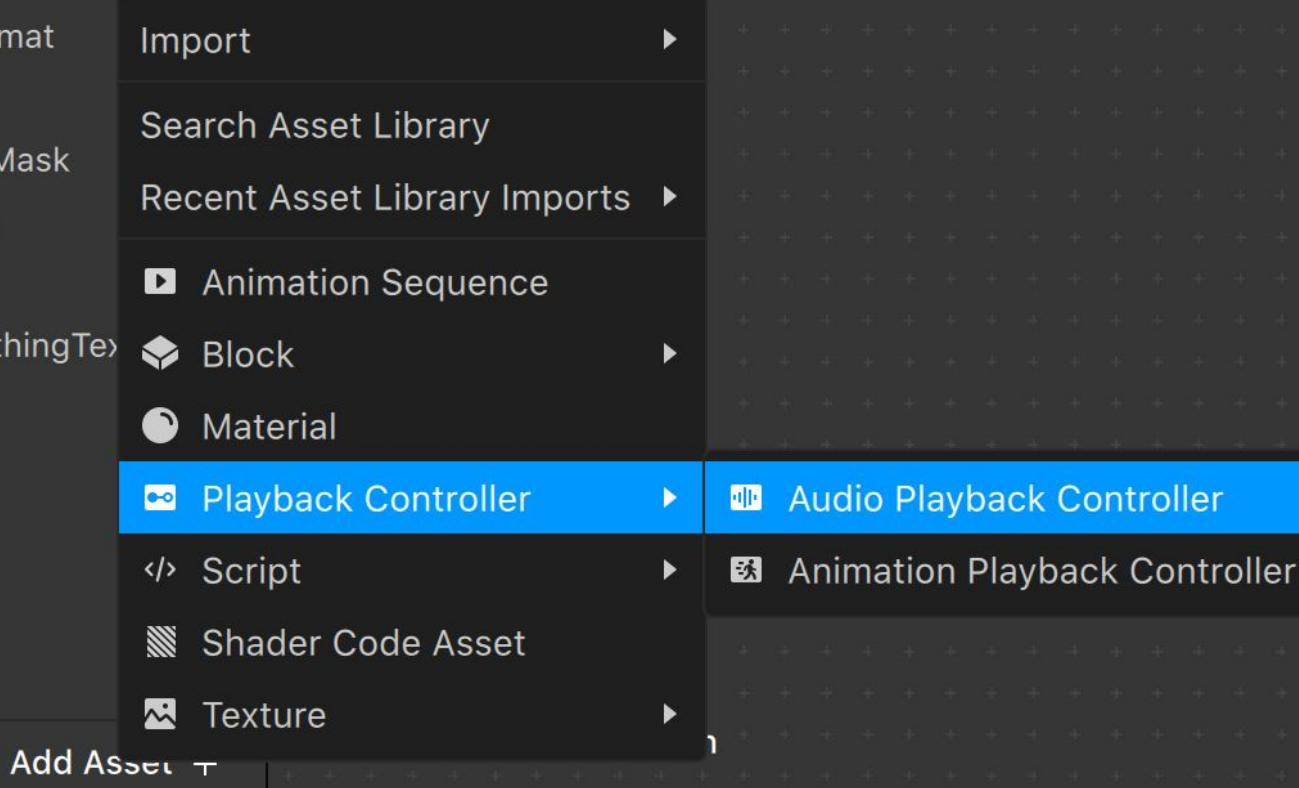
For small changes, use the properties panel. There you can make minor tweaks to your object by inputting any number you like until you land on your preferred scale, position or rotation.



The Manipulator interface in the Viewport

Assets

- ▼ Materials
 - facePaint_mat
- ▼ Textures
 - faceMeshMask
 - replaceMe
 - office0
 - SkinSmoothingTe



Audio assets

There are many options available for adding audio to your Meta Spark Studio effects.

All Meta Spark Studio effects include a microphone. This captures a person's voice and plays it back into the effect. You can disable the microphone if you want to have an audio file playing in your effect on load instead of capturing the sound from the device's microphone. An object called a speaker can be added to render ambient and one-shot audio clips in the scene.

To connect an audio clip to the speaker, you can use an audio playback controller or use the Audio Player patch and Single- or Multi-Clip Controller patches.

The audio playback controller lets you play an audio clip on a loop in the background of your effect or play a one-shot sound in response to a boolean signal. The Audio Player patch and Single- or Multi-Clip Controller patches are used for more complex playback systems, such as when you want to modify the audio live through pitch shifting or distortion.

Meta Spark Studio also includes a wide range of audio effect patches that you can use to modify audio clips or the sound picked up by the microphone.



To use the audio playback controller, search for it under **Add assets** in the assets window.

Before creating an ambient audio asset you should consider:

1. What you want people to use them for.
2. How your effect sounds on a mobile device.
3. Whether you want the file to loop seamlessly. A small, looping ambient bed is suggested.
4. Whether you want to use a seamless looping ambient bed. If not, make sure that your ambient bed runs the max capture length, which, for the camera feature in Messenger from Meta, is 20 seconds.

AR music for Instagram

To create an effect that responds to music, you can either import and use your own legally licensed music files or you can let people pick a song they like from the thousands of free tracks in Sound Collection, available through Facebook. This feature is now available for people that use Android and iOS.

The screenshot shows the AR Library interface. At the top left is a search bar labeled "Search All". On the far left is a sidebar with a "Home" button highlighted in blue, and other buttons for "3D Objects", "Music and Sound", "Patch Assets", "Textures", "Blocks", "Script Packages", "Color LUTs", and "Voice Intents". The main area is titled "AR Library" and has a sub-section "Featured". It displays several asset cards: a large "Shaders" card with a color wheel, an "Animation" card with a graph, a "Utility" card with a blue circuit board image, a "Recently Added" card with a "L" logo, and a "3D Shapes" card with a 3D cube and cone. Below these are "Categories" cards for "3D Objects", "Music and Sound", "Patch Assets", "Textures", "Blocks", and "Script Packages", each with its respective icon.

AR Library and in-app integration

[AR Library](#) is used to import complete 3D objects, audio clips, patch assets, textures and scripts into your Meta Spark Studio projects. It provides a quick and easy way to start building your own effects.

AR Library includes the following assets:

- 3D objects and patch assets licensed for use under [Meta Spark Studio terms](#) as well as the [CC BY 4.0 license](#)
- Scripts licensed under the [MIT License](#)
- Audio files licensed under [Sound collection terms](#)

Add 3D objects to create more immersive effects that take place in people's real surroundings. You can create your own 3D objects with tools such as Blender, a free and open-source 3D content creation suite.

You can also create your own 2D assets via third-party software integrations such as [Adobe® Substance Painter®](#), [Adobe® Photoshop®](#) and [Photopea](#). Learn more about [third-party asset creation tools](#).

Meta Spark Toolkit for Blender

The Meta Spark Toolkit is an add-on you can use in Blender to optimize your 3D object and check that it's ready to import to Meta Spark Studio.

Use the toolkit to:

- Make your 3D objects more lightweight.
- Set the height and the pivot point position.
- Export quickly and easily.

Features of Meta Spark Studio



Apply the concepts

Now that you're familiar with all of the elements and capabilities of Meta Spark Studio, it's time to learn how to develop an interactive scene using the effects you've created.

Script with Javascript

Meta Spark Studio supports JavaScript for adding logic and interactivity to your effects. Many capabilities available in JavaScript can be added to your effect as patches so you can add interaction and complexity without knowing how to code.

Scripting is broken down into modules, with each module implementing a particular functionality. For a complete list of modules, each with their own code example, check out [scripting API](#).

Meta Spark Studio opens your scripts with the default editor assigned to JavaScript and TypeScript files on your macOS or Windows operating system. If you don't already have one installed, you can download an editor such as Visual Studio Code and set it as the default app for opening JavaScript and TypeScript files.



Scripting in Meta Spark Studio supports JavaScript ES6 and [TypeScript](#). You can [split scripts across multiple files](#).

The Javascript patch bridge

You can pass information between scripts and the Patch Editor by adding shared variables. Shared variables can be created from the Inspector after selecting a script file in the assets panel.

There are seven variable types supported with some type differences between Patch Editor and script:

Patch Editor type	Script type
Boolean	Boolean
Number	Scalar
Pulse	EventSource
Text	String
Vector 2	PixelPointSignal
Vector 3	VectorSignal or PointSignal
Vector 4	RgbaSignal

Should you use Patch Editor or scripting?

Scripting is more complex, but it's also a more time efficient option if you already know how to code. Scripting allows you the flexibility to run loops and iterations and save variables. In contrast, patches have to be manually connected one by one. Which option you use depends on your preference and experience level.

Reactive programming in Meta Spark Studio

Meta Spark Studio's implementation of reactive programming allows you to create relationships between objects, assets and values. This means that the engine doesn't have to execute JavaScript code every frame when performing common tasks such as animating content, looking for user input or realigning a mask to a face.

Reactive programming is also compatible with visual programming, reducing the frequency of calls made into the scripting engine. The reactive programming model allows you to define relationships between objects and values so that a value bound to a signal is automatically updated when the latter changes.

If you have experience with other development tools, you may be more familiar with imperative programming. This model differs from reactive programming in that statements are acted upon in order and values are set only when an expression is evaluated. With a reactive model the updating of the value happens in native code rather than JavaScript, reducing the impact on application performance. The reactive model's compatibility with visual programming also means that the frequency with which calls are made to the scripting engine is reduced. Both this and the previous points result in performance improvements for your effects.

Logic

[Logic patches](#) define the conditions needed for something to happen in an effect. For example, you can use logic to trigger an animation to play when the person interacts with the device or after an effect runs for a certain number of seconds.

Patch	Description
And	Makes something happen when two conditions are met at the same time, such as when a person opens their mouth while raising their eyebrows.
Equals	Makes something happen when the value coming from the input patch is within a close range or around a certain number. For example, use this patch if you want the person using it to face roughly a certain point on an axis to make something happen.
Equals exactly	Makes something happen when the value coming from the input patch is an exact number.
Greater than	Makes something happen when the value coming from the input patch is greater than a preset value.
Greater or equal	Makes something happen when the value coming from the input patch is greater than or equal to a preset value.
If then else	Switches between two states in response to something depending on the data type selected.

Patch	Description
Less or equal	Makes something happen when the value coming from the input patch is less than or equal to a preset value.
Less than	Makes something happen when the value coming from the input patch is less than a preset value.
Not	Reverses the output of a connected patch.
Or	Makes something happen when one of two conditions are met, such as when a tap or a long press on the device screen occurs.

Signals

A signal is a special object containing a value that changes over time. One of the ways Meta Spark Studio makes use of a [reactive programming](#) model is by allowing you to treat values as signals.

Values

There are signal equivalents for primitive data types, including:



Number

[ScalarSignal](#)



String

[StringSignal](#)



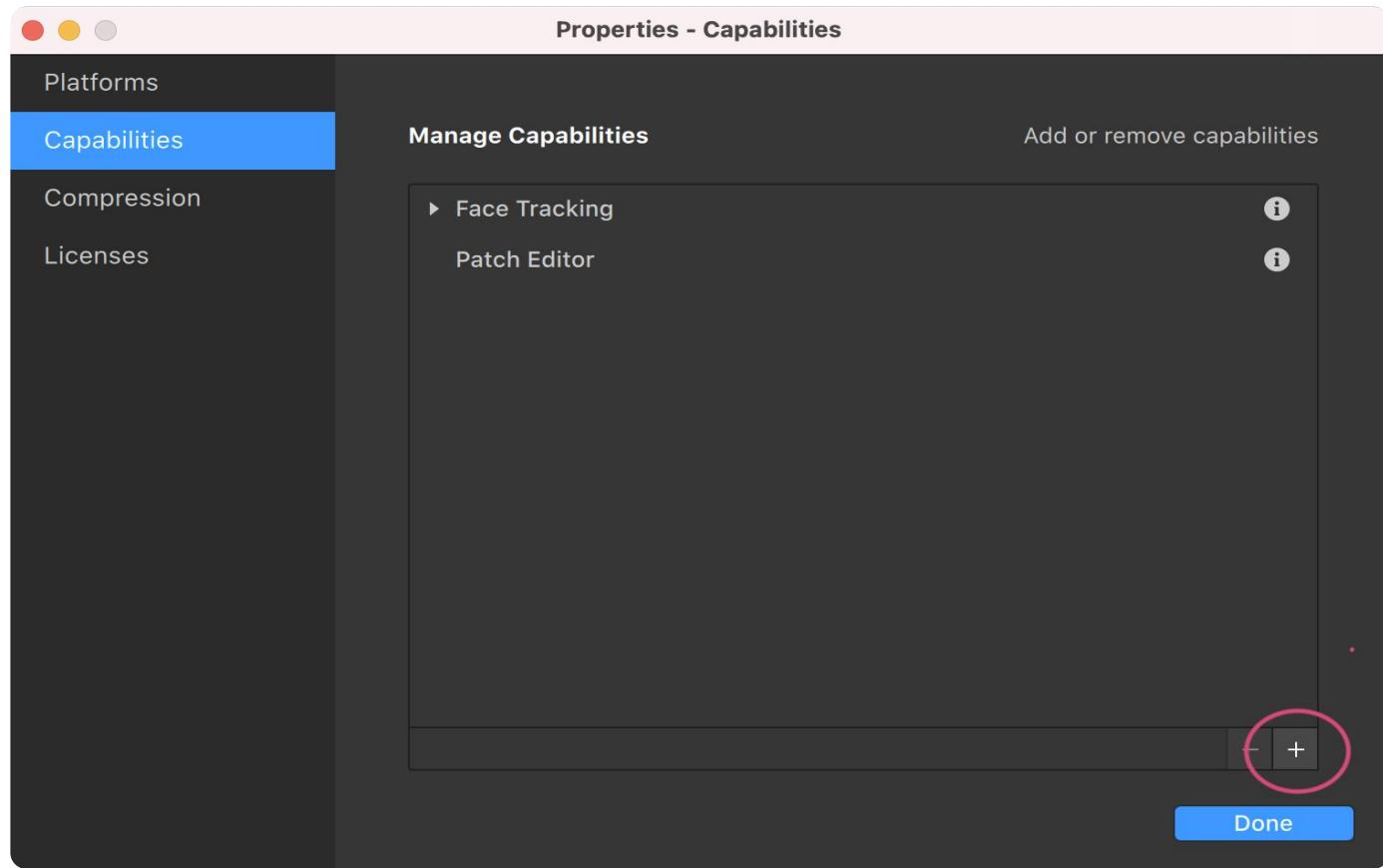
Boolean

[BoolSignal](#)

The syntax for binding a signal to an object is the same as performing a standard assignment.

When a number, string or boolean value is passed to a function or property setter that expects a signal, the value is implicitly converted to a constant signal. You can also explicitly convert a primitive data type to its signal equivalent with the `val()` method exposed by the `ReactiveModule` class.

If you need to retrieve the value that a signal contains, you can use the `pinLastValue()` method exposed by the `ScalarSignal`, `StringSignal` and `BoolSignal` classes.



As their value changes over time, standard JavaScript operators such as +, -, * and / can't be performed directly on signals. Instead, these types of calculations are performed via methods exposed by the `ReactiveModule` class and the `ScalarSignal` class.

The `watch()` method exposed by the `DiagnosticsModule` class allows you to add a signal to the watch view in Meta Spark Studio to monitor how its value changes over time.

Assign and configure asset properties.

Configure settings for a project.

Not all capabilities are added to projects automatically. If needed, you may have to add other capabilities manually. You can configure the capability to be used via the project tab at Meta Spark Studio.



You can also set the type of the experience for the effect and the compression configurations.

Note that some capabilities are platform-specific. It is important to be aware of these limitations so you can manage client expectations.

Camera configuration and visibility

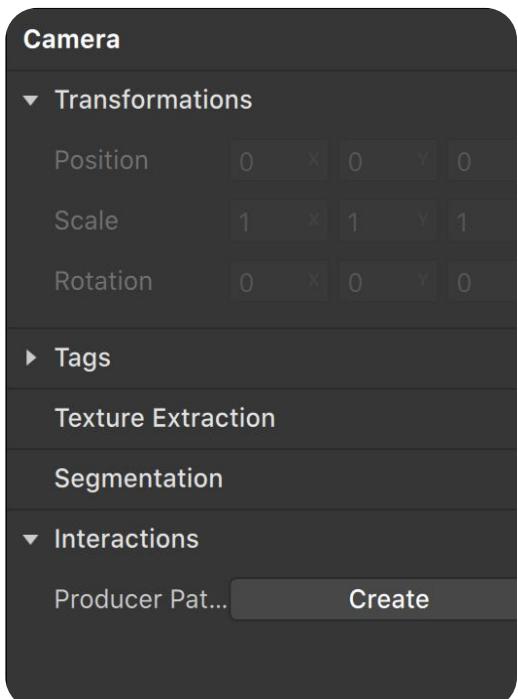
The camera is an object that's automatically listed in the scene panel in every effect. It represents the camera in the device showing your effect, which means it can't be removed from a project.

The camera is listed at the top of the scene panel. You can use the camera to add background segmentation or create a texture representing the video captured by the camera while your effect is being played.

When objects are children of the camera in the scene panel, they move with the movement of the camera. When objects are no longer children of the camera, you can position them in a fixed place in the world, creating a world effect.

When the camera is selected in the scene panel, you can see its properties in the Inspector.

The following are properties of the camera:



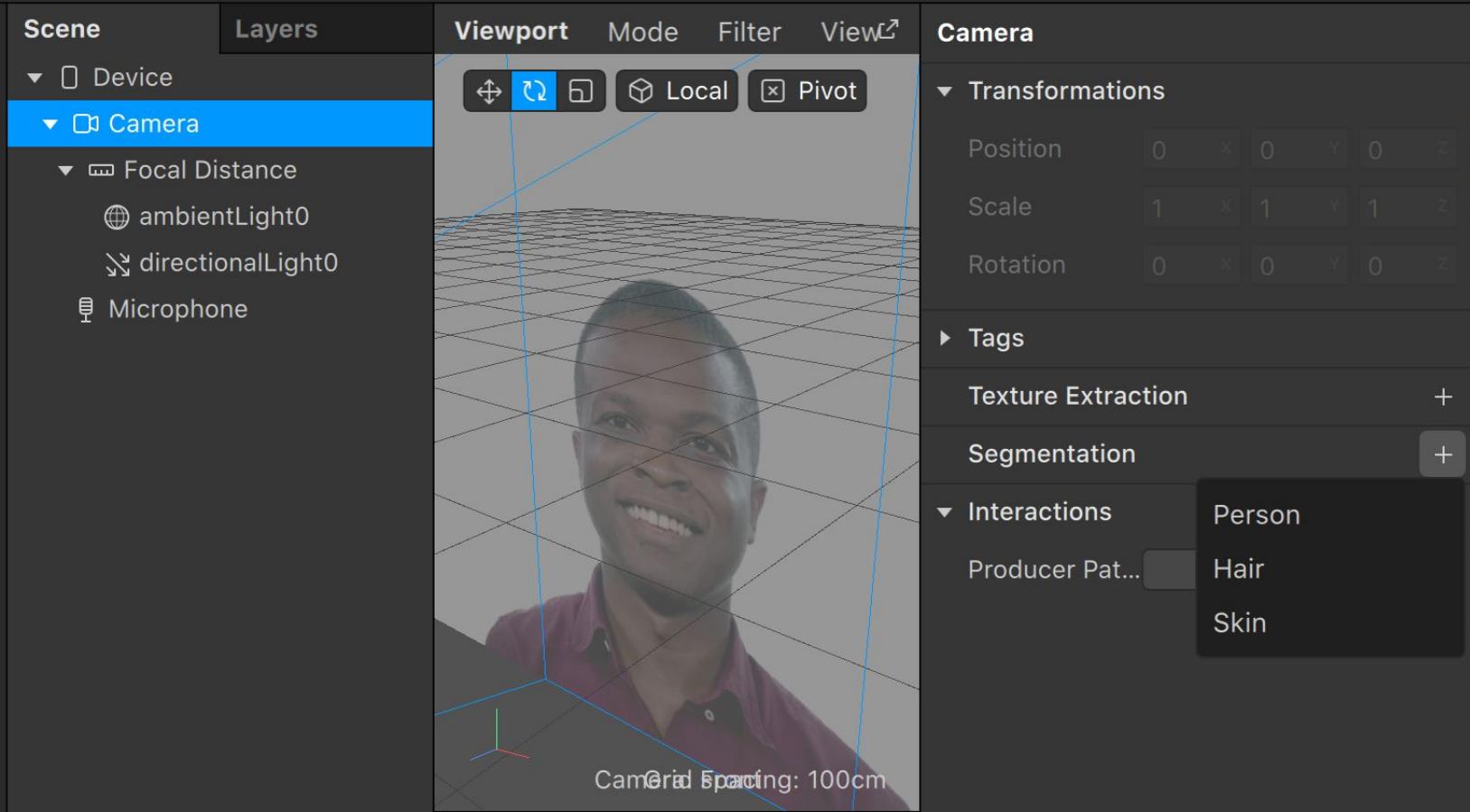
Camera properties in the Inspector panel

Transformations	The position, scale and rotation can't be edited.
Texture extraction	Click the arrow next to Texture extraction to create a texture from the video captured by the camera.
Texture	Texture is listed in the assets panel as cameraTexture0.
Segmentation	Segmentation separates part of the scene so you can make changes to it separately from the rest of the effect.

Segmentation in AR

Segmentation is used in Meta Spark Studio to detect and separate parts of the camera input. For example, if you want to separate someone's hair from the rest of the scene, you can change the color without changing anything else in the effect.

You can add segmentation to an effect by applying a segmentation texture to the texture property of a material. You would then apply this material to a rectangle that covers the full width and height of the device screen.



Types of segmentation under the camera's properties

There are three types of segmentation available and you can combine all three types of segmentation to achieve a more complex effect.

Person segmentation	To separate the person using the effect from their background
Hair segmentation	To separate someone's hair
Skin segmentation	To separate any skin detected by the camera from the rest of the scene

Assign relationships between objects.

Use the scene tree.

You can create relationships between objects in the scene panel. Creating a child-parent relationship means you can make the same change to multiple objects at the same time by making a change to the parent object. To create a child-parent relationship, drag the object that you want to be the child onto the object that you want to be the parent.

This is also where you'd insert an element that makes your effect respond to the person using it or their environment. For example, you can use a face tracker to build an effect that responds to someone's face.

Group with null objects.

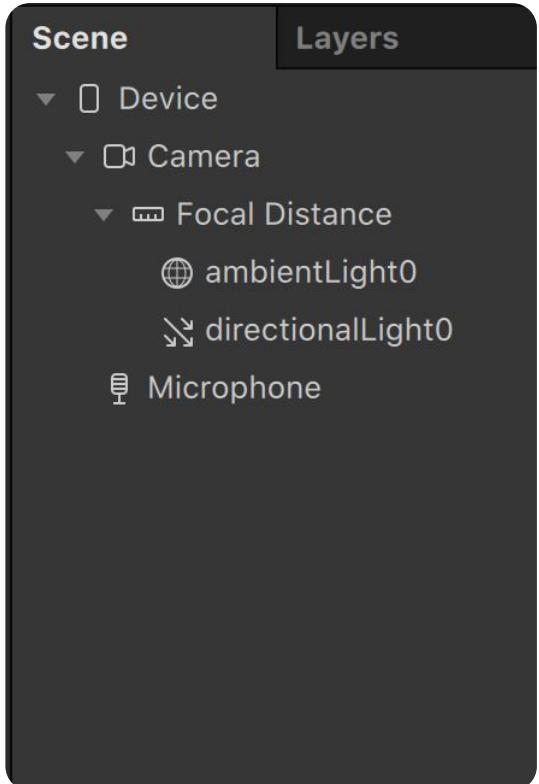
In Meta Spark Studio, a null object can be used to group other objects. It doesn't have a shape, but it can be manipulated in 3D space.

Objects grouped under a null object are known as children. The children take on the properties of the null object. For example, if you enlarge a null object, any children will also be enlarged.

Null objects are particularly useful for adding secondary animation or ensuring a group of objects have exactly the same position.

Group other objects.

To group objects using a null object, the objects must be children of the null object. You can also use patch logic to assign relationships between objects. Here's how:



The scene panel

Click the arrow next to a property in the Inspector to create a patch to represent it.

Choose between two interactions:

Object tap

Creates a patch representing the null object and its children in the scene and an object patch, which detects when someone taps the null object.

Producer patch

Creates a patch that represents the null object.

Null objects can have certain properties:

Visible	Clear this box to stop the null object and any children from being rendered in the scene.
Transformations	Change the position, scale and rotation of the null object. You also use the Manipulator to do this.
Interactions	Insert simple combinations of patches involving the instance, such as tap gestures, into Patch Editor.
Enable for	Choose the camera or cameras on a mobile device in which you want to render the null object.

Delete objects.

You can delete unused assets by checking **Used by**. This property of the asset states whether it's related to another asset in the project file. If empty, consider it safe to delete. Child-parent relationships between grouped objects can be removed by dragging and moving the child object outside the parent object.

Render and layer objects.

Objects in a Meta Spark Studio effect are always rendered in a specific order. In general, an object that's listed closer to the camera in the scene panel appears in front of one that's further away. This can mean an object intended to appear in front of another object appears behind it and vice versa. You can use layering to change this. By default, new projects have one layer to which every object you add the scene is assigned. There must always be one layer in a project.

The render order is the sequence in which the layers are rendered. If you have more than one layer in your project, the layer at the bottom of the list is rendered first and the layer at the top of the list is rendered last.

Opaque objects are affected by the render order only if their position on the z-axis overlaps. Transparent and semi-transparent objects may be affected by render order regardless of their position on the z-axis, because they can occlude objects in other layers. This means transparent objects set to render first will occlude objects rendered on a subsequent layer. You might need to adjust your layers to account for this.

Render options

You can use the render options and advanced render options in premade materials to set how the material renders in your Meta Spark effect. Render options and advanced render options can be edited in the Inspector when a material is selected in the assets panel.

Cull mode	<ul style="list-style-type: none">Back: Discards pieces of the mesh that are facing away from the viewer. Most effects use this mode to improve performance.Front: Discards pieces of the mesh that are facing the viewer.
Blend mode	Sets the blend mode of the material.
Opacity	Sets the transparency of the material on an individual map level.
BG influence	Multiplies a color or texture against a grayscale version of the video feed. Only available in the face paint material.
Brightness	Sets the brightness of a color. Only available in the face paint material.
Double-sided	Displays both sides of a face norm.
Alpha test	Uses the slider to make semi-transparent parts of a material transparent.

Advanced render options

Use depth test and write to depth buffer	These control certain order parameters for the object the material is applied to.
Both boxes selected	Enables read and write depth buffer. This means that closer objects will obscure farther objects.
Write to depth buffer selected	Disables read depth and enables write depth. These objects will always draw over any other object. They will obscure parent objects that are drawn after unless these objects have either the write to depth buffer or both boxes selected.
Use depth test selected	Enables read depth and disables write depth. These objects will be obscured by nearer objects that have already been drawn that have either the write to depth buffer or both boxes selected. These objects can be obscured by any objects drawn afterwards, whether they're nearer or farther away.

Clear both boxes	Disables read and write depth. These draw on top of any objects of any type already drawn, and in turn will be overdrawn by any later rendered objects regardless of whether they're nearer or farther away.
Color mask	Disable writing to specific channels of the color buffer. The color channels of the cleared editors will be preserved after rendering. When using this feature, the color in the simulator and the Viewport might not be the same. The simulator will most closely match the end result on a device.

Adding scene interactivity



Once you've integrated objects into your scene, the next step is to make it engaging for people so that they'll want to share it with friends and continue using it themselves.

Process signals within a scene.

How to implement logic and user input with patches

You can use utility patches for a variety of functions, such as controlling outputs and transforming data types.

<u>Animation patch</u>	<u>Frame Transition patch</u>	<u>Loop Animation patch</u>
Creates an animation triggered by an interaction patch or logic that plays once.	Controls the frames in an animation sequence.	Creates animations that play on a loop.

Interaction patches detect a person's taps, swipes and gestures on the screen of their device and add interactivity to your effects in response.

Add this type of interactivity to effects that use the plane tracker to place objects on surfaces in the real world. That way people can make objects the right size and move them around.

Screen interaction patches	
Screen Pan	Enables you to control position. Detects a swipe of the finger across the device screen.
Divide	Manipulates input from the Screen Pan patch.
Screen Pinch	Enables you to change scale. Makes an object bigger or smaller when someone pinches the screen.
Screen Rotate	Allows you to rotate an object.

Tap patches

Screen tap	Makes something happen when the screen is tapped.
Object tap	Makes something happen in your effect when someone taps a specific object.

Here are some of the interaction patches you can use to add movements and gestures to your effect:

Movement patches

<u>Blink</u>	Makes something happen when a person blinks.
<u>Eyebrows lowered</u>	Makes something happen while a person's eyebrows are lowered.
<u>Eyebrows raised</u>	Makes something happen while a person's eyebrows are raised.
<u>Head nod</u>	Makes something happen when a person nods.
<u>Head rotation</u>	Makes something happen while a person's head moves in a certain direction.
<u>Head shake</u>	Makes something happen when a person moves their head from left to right or right to left.
<u>Left eye closed</u>	Makes something happen while a person's left eye is closed.
<u>Mouth open</u>	Makes something happen while a person's mouth is open.
<u>Right eye closed</u>	Makes something happen while a person's right eye is closed.

Gesture patches

<u>Happy face</u>	Makes something happen while a person makes a happy face.
<u>Kissing face</u>	Makes something happen while a person brings the corners of their mouth together to make a kissing face.
<u>Smile</u>	Makes something happen while a person smiles.
<u>Surprised face</u>	Makes something happen while a person makes a surprised face.

The face landmark patch

Face landmark patches are used to capture the precise position of parts of the face, such as eyes and nose. This allows you to position objects like piercings to move in sync with facial and head movements.

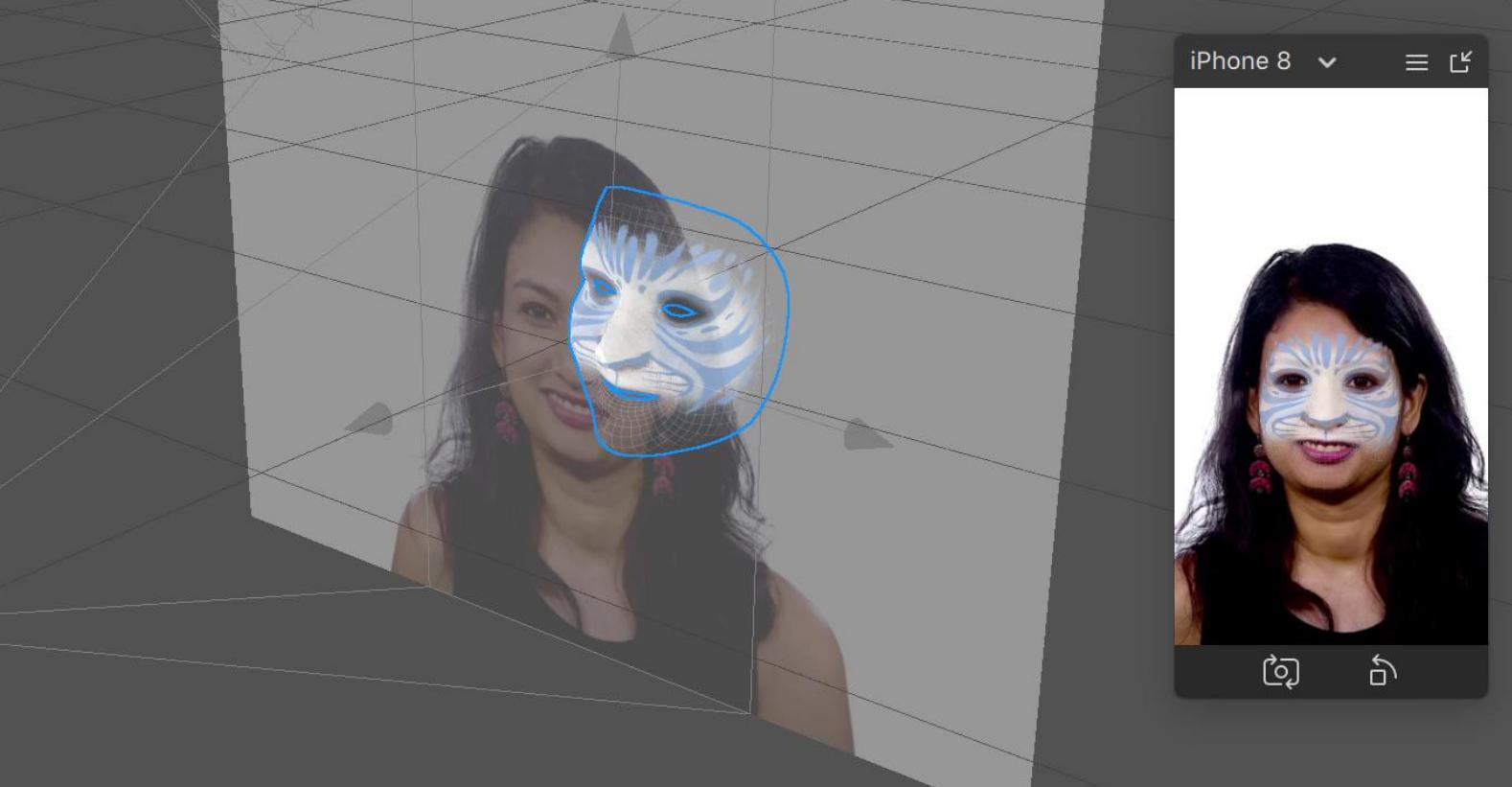
Apart from objects positioned using the Eyeball patch, objects positioned with face landmark patches don't rotate in sync with the person's facial and head movements, but still move along the X, Y and Z axes as normal.

To position objects using face landmark patches, the object must be a child of a face tracker in the scene panel.

Face landmark patches

<u>Cheek</u>	Position an object at a precise location on a person's cheeks, such as the center of a cheek or a cheekbone.
<u>Chin</u>	Position an object on the tip of a person's chin.
<u>Eyeball</u>	Position an object at a location on a person's eyes, such as the center of an eyeball. This patch is used for <u>iris tracking</u> .
<u>Eyebrow</u>	Position an object at a precise location on a person's eyebrows, such as the middle of an eyebrow.

<u>Eyelid</u>	Position an object at a precise location on a person's eyelid, such as the inside or outside corner of an eye, and track the degree the eye is open.
<u>Forehead</u>	Position an object on the center or top of a person's forehead.
<u>Mouth</u>	Used to detect a specific part of a person's mouth.
<u>Nose</u>	Position an object at a precise location on a person's nose, such as a nostril or the tip of the nose.



Face tracker and face mesh

The face tracker

When a face tracker is combined with a face mesh, it creates a surface that can detect facial movements and expressions. You can add a material to the face mesh to create a mask effect.

Objects that are children of a face tracker will move with the face detected in the scene. To make an object the child of the face tracker, simply drag the object onto the face tracker in the scene panel. You can then use the Manipulator to adjust its position.

To track objects to precise positions on the face, use face landmark patches.



Adding multiple face trackers to a scene will have an impact on performance, so test your effect as you build it using Meta Spark Player.

How to dynamically modify a scene

Use target markers.

If your effect tracks a target, show a target marker in the effect until the target image is detected. This helps people using the effect know what target to look for in the real world. When the correct target is detected, the target marker can be set to disappear, revealing the effect.

The target marker is made of two textures:

1. A semitransparent preview of the target image.
2. A texture provided by Meta Spark Studio that looks like a frame. The file is called target-marker-texture.png and you can download the image file [here](#).

A simple combination of patches in Patch Editor can make the target marker disappear when the target is found.

Each of these textures will be added to the scene using an object included in Meta Spark Studio called a rectangle. The rectangle will automatically be added as a child of a canvas which will ensure it resizes in sync with the device screen. The canvas will also be used to group the target marker texture and the preview image, so changes can be made to both of them at the same time.

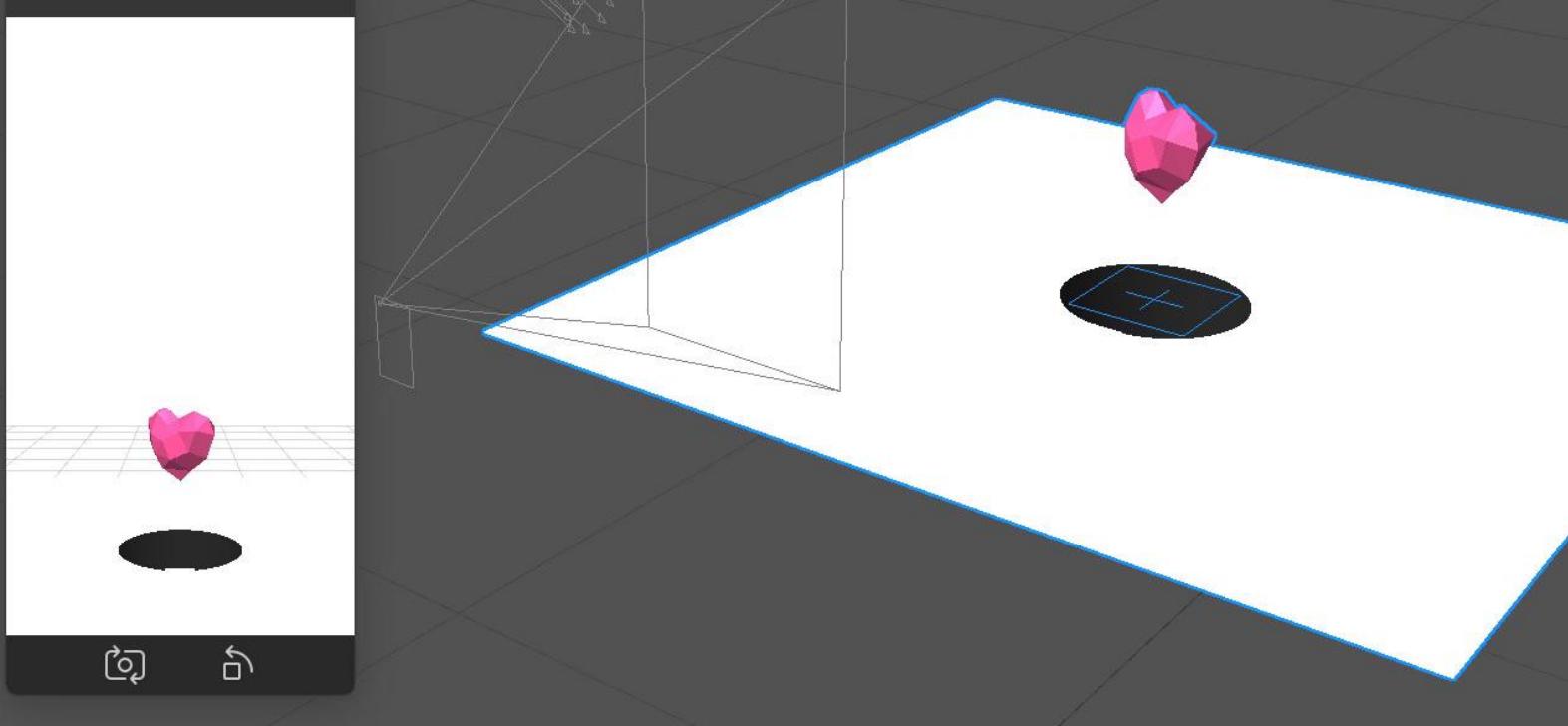
Use the plane tracker to integrate effects into the real world.

A plane tracker finds a horizontal surface in the real world, like a floor to table. Plane trackers enable you to place 3D objects in people's environments.

Plane tracking effects are only visible through the camera on the back of the device. Once you select the plane tracker in the scene panel, you're able to see its properties in the Inspector.

The following are property options:

Layer	Click the dropdown next to Layer to adjust which layer the plane tracker is assigned to.
Visible	Clear this box to stop the plane tracker and any children from being rendered in the scene.
Auto start	Select this box to find a plane in world space as soon as your effect is opened.



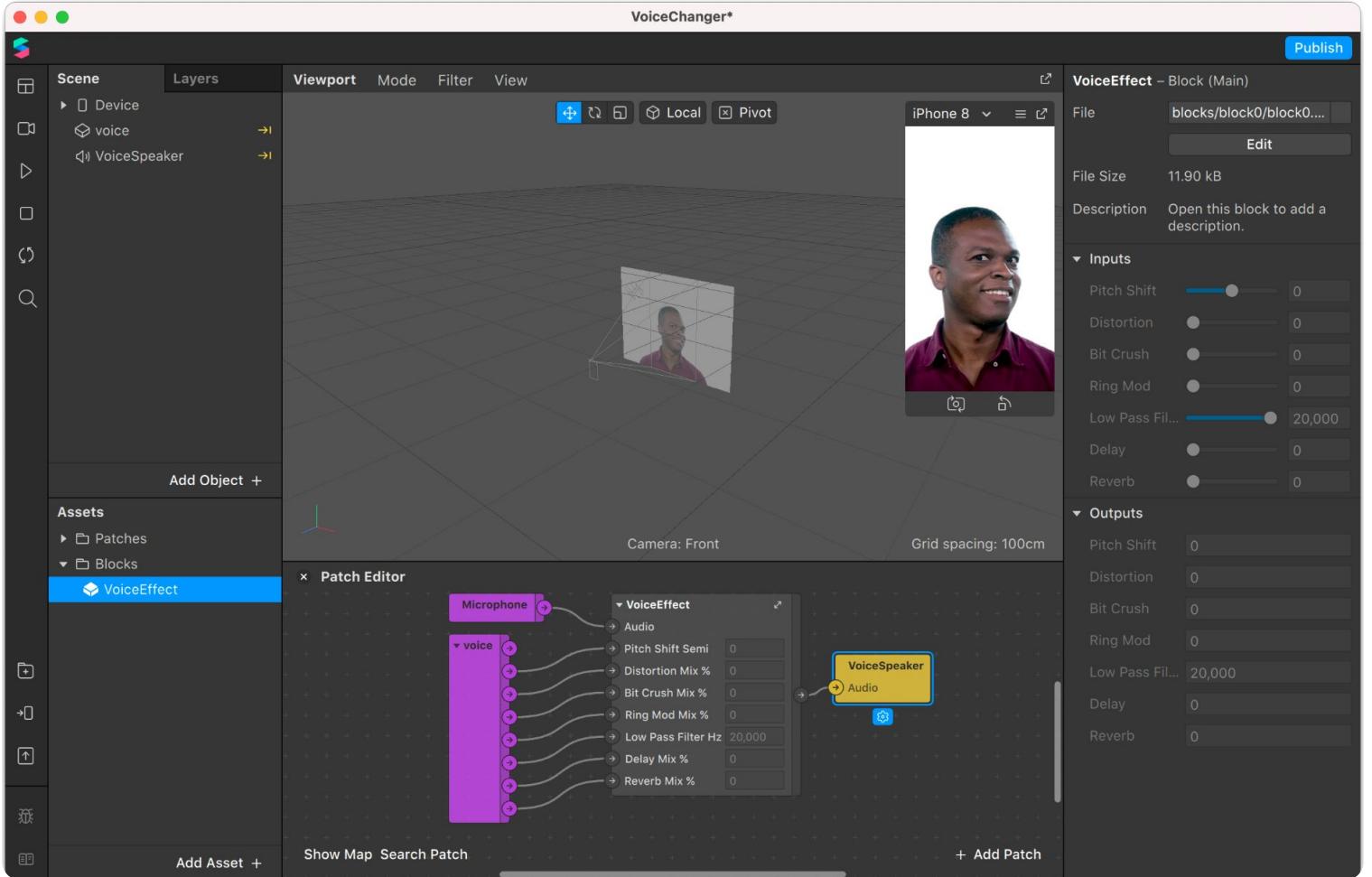
An example of a plane tracker

Transformations	You can't change the transformations for a plane tracker because it isn't an object.
Enable for	This sets camera-specific visibility. For the plane tracker, Enable for is set to back camera automatically and can't be changed. This means that the plane tracker will be visible only in the back camera of a mobile device.

When adding a plane tracker to an effect, add instructions so that people know that the effect should be viewed through the back camera. You can use Patch Editor to do this.



The target image preview should be the same image that triggers the target tracking effect. To complete the effect, use Patch Editor to make the target marker disappear when the target is found. Use Meta Spark Player to preview your target tracking effect on a device.



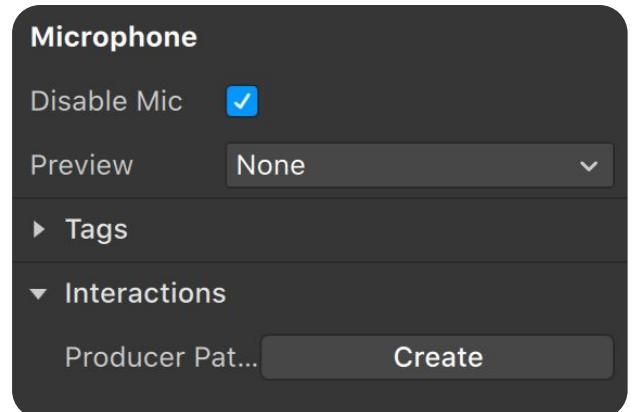
Voice changer template

The microphone

The microphone captures sound detected by the device's microphone when your effect is being used and plays it back in the scene. You can use the microphone to analyze and modify the voice of the person using your effect.

By default, the microphone records and plays back any sound detected when the effect is being used. Check **Disable mic** to stop microphone sound being recorded when capturing a video with the effect. Note that there are some limitations. For example, you can't play audio while recording with the microphone in an effect.

You can use audio effect patches to distort and modify the sound detected by the microphone in Patch Editor. You can also analyze the sound detected by the microphone via the energy meter patch.



Microphone properties in the Inspector

How to use gallery textures

The gallery picker lets people add an image from their phone's camera roll to an effect, such as a green screen. To create an effect with the gallery picker, add the gallery texture to a material in your scene. The material can be applied to any object, such as a 3D object or a rectangle, to create a segmentation effect.

By default, the gallery texture renders as a blank white color. You can add a texture or color to the gallery texture, to be replaced with the image selected by the person using your effect.

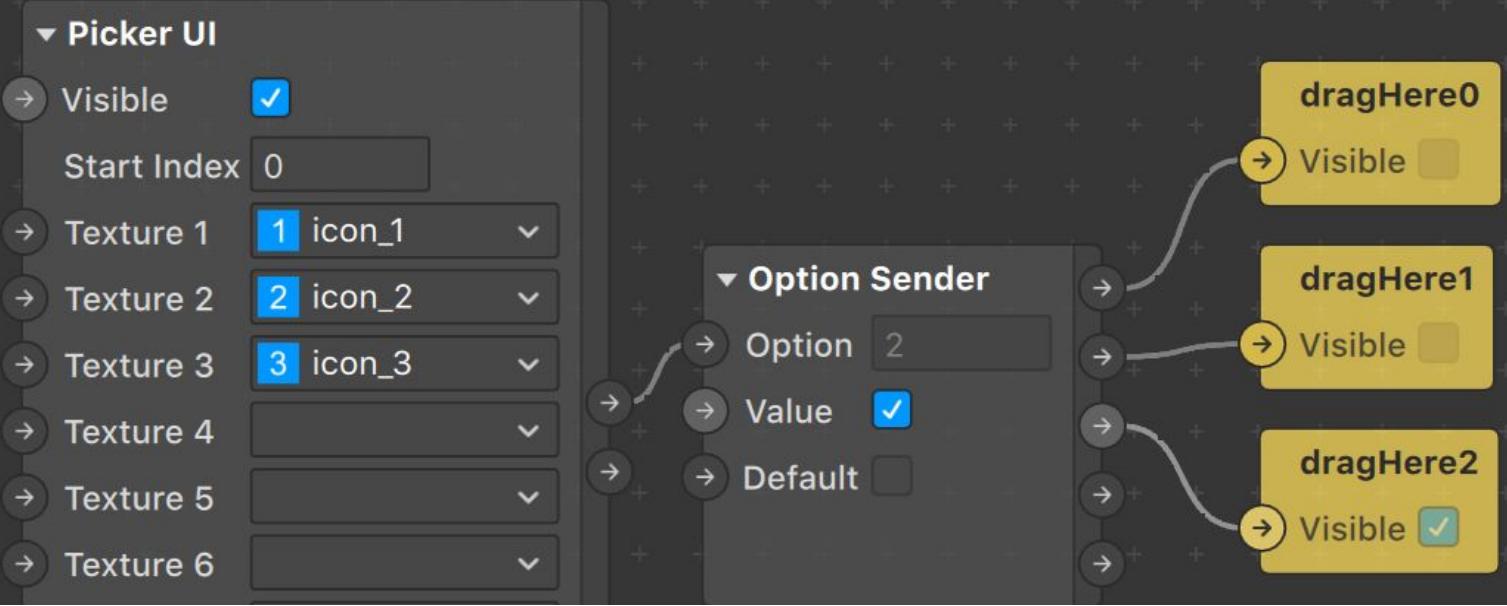
If an effect includes the gallery texture, an add media button will appear at the bottom of the effect. People can then browse and add images from their own media library.

There are a few ways you can edit the gallery texture's properties in order to change how it appears in the effect:

Pending texture	Check this box to select an initial texture to show in the effect. This texture will be replaced with the image selected by the person using your effect.
Type	<ul style="list-style-type: none">• If type is set to color, select a color from the menu.• If type is set to texture, add an image file there.
Used by	Any materials the texture is applied to will be listed here.

When planning effects that use the gallery texture, note that only one camera roll texture can be selected at a time.

Either photos or videos can be added to effects with the gallery texture, and since those images or videos can be square, horizontal or vertical, you might need to resize the image to fit the area where you want to use it.



Example of Picker UI patch

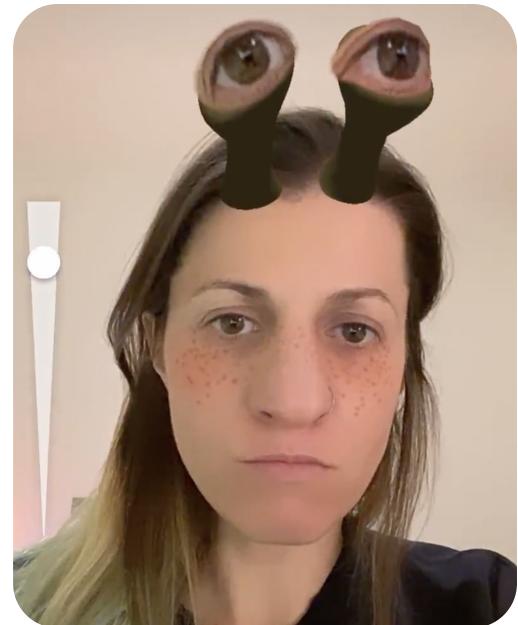
How to process user inputs

User interface patches let you add UI functions to your effect. These patches enable people to pick among different options or adjust properties in an effect by tapping on-screen icons or using an interactive slider.

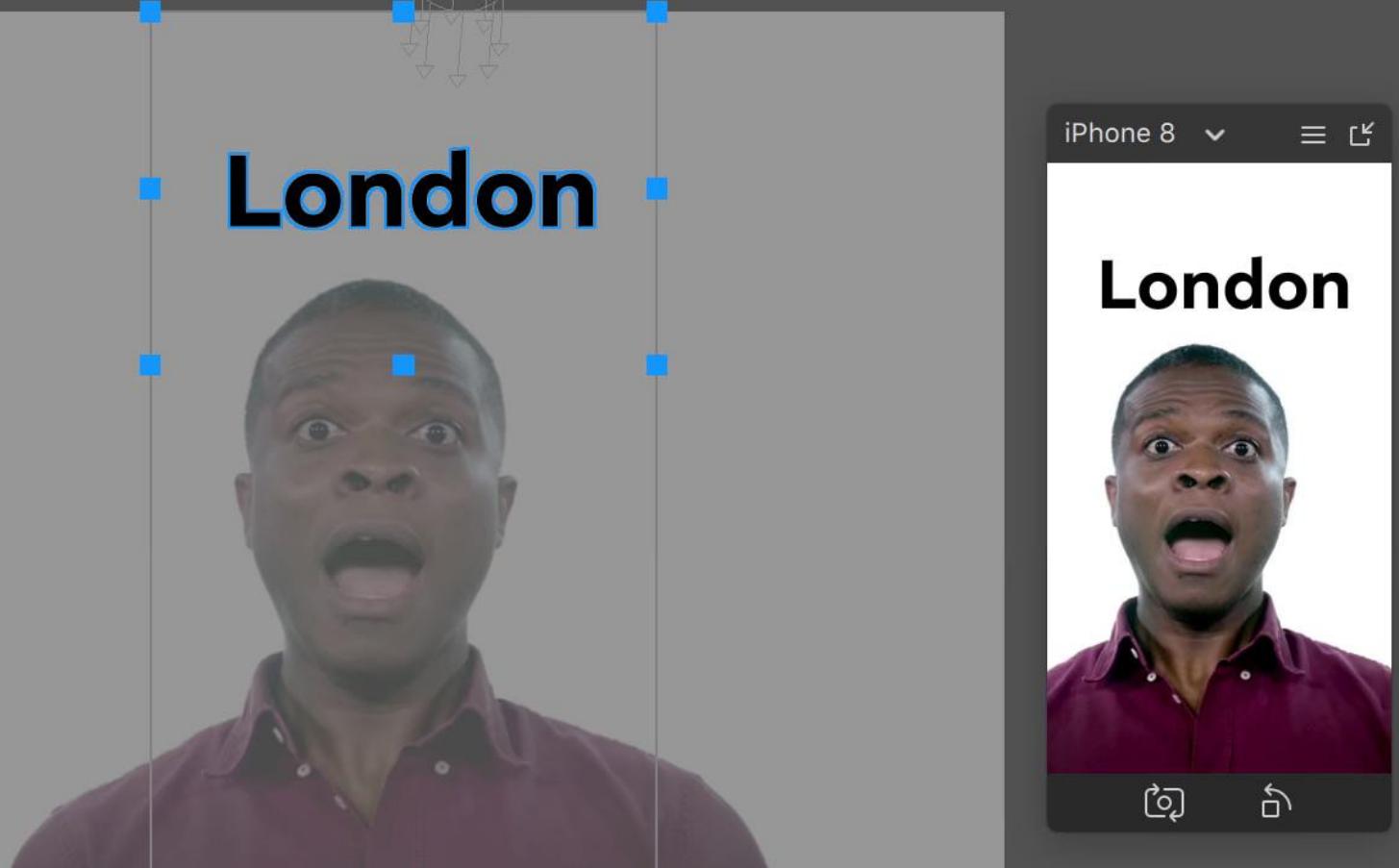
Picker UI patch and Slider UI patch

These patches can be added to Meta Spark Studio effects. The Picker UI patch object needs to be configured with a list of uncompressed textures that are displayed in the patch.

<u>Picker UI patch</u>	Use the Picker UI patch to select from up to 10 options that change something in your effect when people tap icons on the screen.
<u>Slider UI patch</u>	Enable people to increase or decrease the value of something in your effect using an interactive slider.



Example of interactive slider



Example of dynamic text

Dynamic text

Dynamic text is used to create personalized effects that show text specific to someone's situation, such as their location or the date. You can add both 2D and 3D dynamic text to your effects.

Use instructions for effects.

Provide clear instructions to help people learn how to use your effect. Your instructions will appear in the middle of your device screen, and you can set the duration time during your initial set-up.

You can add custom instructions by using the Inspector to show instructions when an effect has been open for a set amount of time. For more complex instructions involving logic and other interactions, you can use Patch Editor or scripting to relay them to the person using the effect.



Example of instructions

Scene optimization



While it's important that your effects are engaging and high quality, they should also perform well on a variety of devices. In this chapter you'll learn a number of ways to improve the performance of your effect.

Technical guidelines and performance considerations

There are maximum limits for object count, texture resolution and triangle count for 3D objects. Follow the guidelines below to strike a balance between quality and performance.

Scene	Add no more than 50 objects to a scene.
3D objects	<p>When importing 3D objects, note that:</p> <ul style="list-style-type: none">• The maximum number of vertices per object should be below 20,000.• The total triangle count per object in an effect should be below 50,000.• Keep the total triangle count for all objects in an effect below 150,000.• The height should be between 1 centimeter and 5 meters.
Exporting	Before exporting a project, make sure you've removed any capabilities that aren't used in the effect.

Methods to optimize assets

Special considerations when creating 3D objects

If you create 3D objects with Blender, use Meta Spark Toolkit to optimize your objects before exporting them to Meta Spark Studio.

To reduce the impact of 3D objects with animations on your effect, minimize the number of bones and avoid using complex blend shapes wherever possible.

Tips for optimizing textures

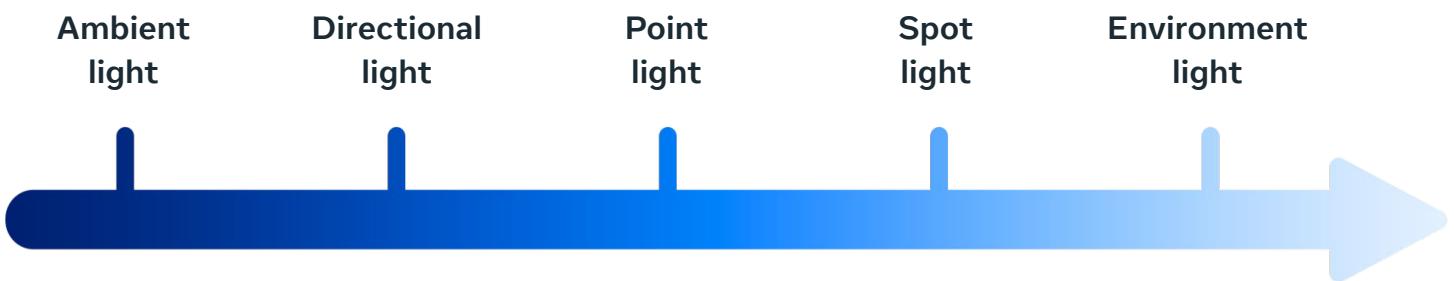
Size	Make textures as small as you can wherever possible.
Resolution	The maximum resolution for a texture is 1024 x 1024 pixels. Any textures you add to a project that are larger than this will be automatically resized.
Format	Make sure textures are square and sized to the power of two as in 2x2, 4x4, 16x16 and 32x32. Some texture compression options, such as PowerVR texture compression, require textures to be in this format.
Sequences	Avoid long 2D texture sequences. If possible, render your sequence every other frame and avoid making sequences that are 24 or 30 frames per second.

Limitations of segmentation effects

Segmentation works best:	Segmentation doesn't perform well:
<ul style="list-style-type: none">• When detecting people from the chest upwards in the immediate foreground.• In settings that have balanced lighting.• With newer devices. These include:<ul style="list-style-type: none">◦ iOS: iPhone 6S or later◦ Android: Samsung Galaxy S6, Sony Xperia Z2 or equivalent, or later	<ul style="list-style-type: none">• With effects that contain both segmentation and the face tracker.• In settings with unbalanced lighting.• With older devices.

Light sources

Some light sources have more of an impact than others on the performance of an effect. Note each light source in Meta Spark Studio, starting with the lowest impact and ending with the highest:



Remove the ambient light automatically included in a project where possible. For example, you can remove ambient light if you're creating an effect with 2D objects that don't require realistic lighting.

Methods to optimize logic

If you're using patches, grouping them can improve performance. You should also avoid duplicating logic, such as when you are calculating the same value more than once.

If you're using scripting, wherever possible, avoid using subscriptions, cast variables as constant and avoid frequent callbacks and functions for every frame.

How to use compression to build efficient effects

Before publishing an effect, experiment with different compression settings to make your overall project size as small as possible. Smaller effects are quicker to load, and this can improve the number of impressions your effect gets.

Any compression you apply in Meta Spark Studio won't affect your original files.

When to use texture compression

To help decide which textures would benefit from compression, you can see the file sizes for textures in both downloaded and unpacked bundles for each device type including iOS, Android and older Android.

There are three possible compression settings in Meta Spark Studio for each target device:

Automatic	Manual	None
This setting finds the best compression format for your texture on the targeted device, ensuring your effect works on all devices.	Use this option to manually apply additional compression and control the settings.	This setting isn't recommended, but you could use it if you already compressed your textures before uploading them.

When to use image compression

When you choose the texture format, you're choosing to prioritize either file size or in-memory size. If you want a smaller effect file size, choose uncompressed. If you want a smaller in-memory size, choose ETC.

An uncompressed effect uses either a PNG or JPG image file. Use either one of these formats to reduce the image file size when exporting the effect, but note that both of these files expand to the full raw image on loading.

ETC textures have a fixed compression ratio. This means that the image file can be larger than the equivalent PNG or JPG but will be smaller than the full raw image. On loading, ETC textures remain the same size, which is smaller than the equivalent PNG or JPG.

Tips for saving and sharing your files

Use patch groups and patch assets.

You can save and share groups of patches with other Meta Spark Studio creators, and import ready-made patch groups into your own projects. It's a useful way to efficiently build effects with logic and interactivity.

Any groups of patches you've added to a project will be saved in the patch library, under **Saved patches**. You can share these files with other creators.

Once you've created a shareable patch asset file, you can import or drag the ARP file directly into the assets panel, as you would any other asset. You can use the patch asset as many times as you want in your patch graph and quickly edit all instances of this patch asset at the same time.

Any changes you make to the patch asset, either in the assets panel or in your graph, will be applied to the patch asset wherever it's used in the project.



Find and share patch assets by joining the [Meta Spark creator groups](#). You can also import patch assets into your project directly from AR Library to quickly add logic and interactivity to your effect.

Use project bundle to share your project.

If you want to send a version of your project to someone, you can [bundle](#) the project folder and file together to create a package.

You can save your project in a shared folder to collaborate over time, but packaging projects is useful for quick and easy distribution of your work. To package your project:

- Click **File** in the menu bar.
- Select **Package**.

Use AR blocks.

With blocks, you can save and export sections of a project in Meta Spark Studio to reuse in other projects and share with other creators.

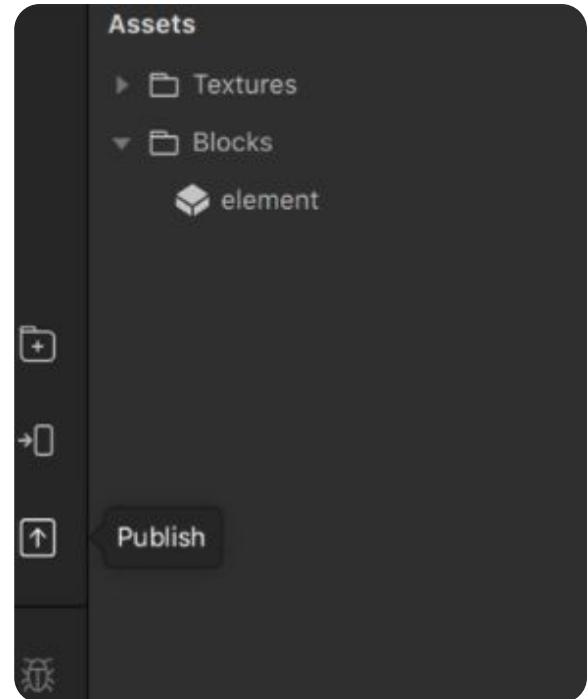
You can create a block for most things you often need to repeat or reuse. Combinations of objects, assets, patches, materials and texture settings are all supported.

When you save a block, it'll automatically update in all projects that use the block. The quickest way to add a block to your scene is to click and drag it from the assets panel to the scene panel.

Inputs and outputs pass information to and from blocks into your projects. You use them to set default values for the block's properties, allowing you to customize the block.

The following data types are supported:

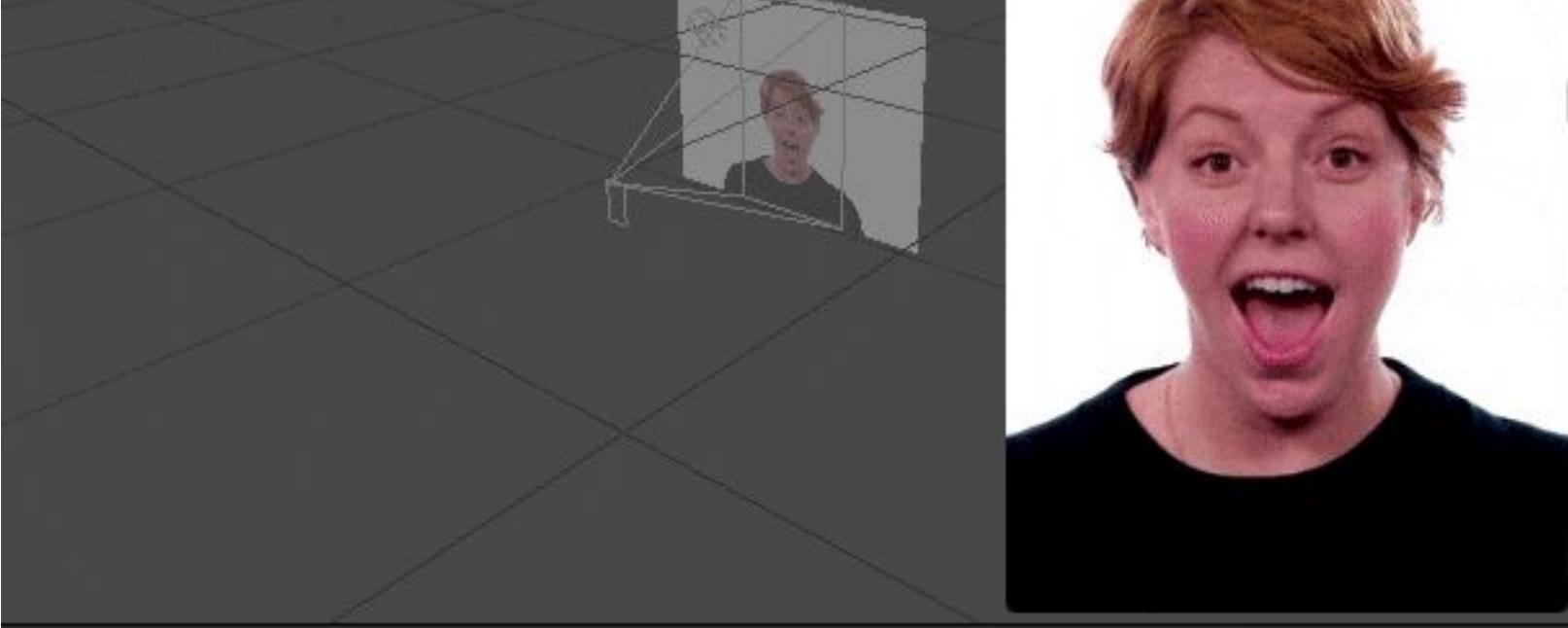
- | | |
|--------------------------------|-----------|
| • Boolean | • Number |
| • Scalar | • Text |
| • Vector2, Vector3,
Vector4 | • Texture |



Blocks in the assets panel

Make sure your effects are the correct size. Check your effect's file size across all three device types: iOS, Android and older Android. To do this, open your effect in Meta Spark Studio and click **Publish** in the toolbar.

If you see the message "file sizes meet requirements," your effect is the correct size for all device types and you can upload it to Meta Spark Hub. If you see a cross next to the technology you want to make your effect available on, you need to make the file size smaller. There are many ways to reduce the file size, from removing unused assets to optimizing objects in your scene.



Console

0 0 Local Show All Search Clear

Mouth Openness - 0.755000

An example of watch() method to monitor mouth openness value

Javascript

Test and debug scene interactions and process signals.

The Console can be used to test and debug your projects in Meta Spark Studio in two ways:

Console logging	You can use the Console in Meta Spark Studio to log messages and variables from scripts you've added to your project. Messages and variable values can be logged from within a script with the log() method of the diagnostics module.
Signal watching	Signal values can be shown in the Console from within a script with the watch() method of the diagnostics module.

Console

0 0 Local Show All Search Clear

> A console message logged from the script 16:41:48

> 5 16:41:48

Messages logged in the Console

Dynamically modify a scene.

With the dynamic instantiation feature in Meta Spark Studio, you can create, destroy and reorder objects in the scene dynamically via script.

Dynamic creation and destruction of scene objects, materials and blocks are all supported via calls to their respective APIs. Dynamic objects aren't displayed in the UI by default. Instead, Meta Spark Studio automatically detects when a dynamic API is called within a script and displays a notification prompting you to switch to dynamic mode.



Dynamic objects can't be edited from within the Meta Spark Studio UI. Their properties are exclusively set via script.

Publication and effect management

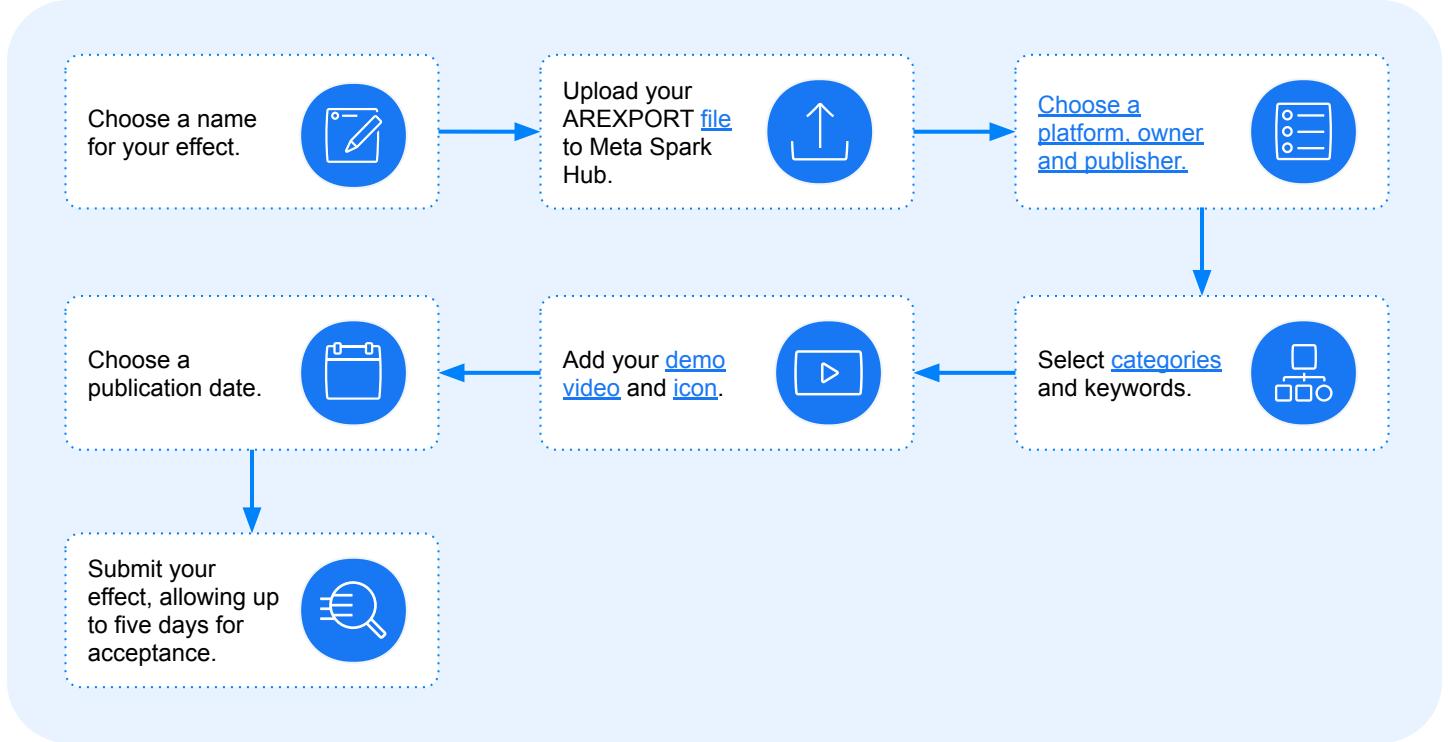


Meta Spark Studio enables a global community of creators and developers to imagine, create and share AR experiences. This community offers a space where you can make safe, respectful and high-quality effects and share them with the world.

How to publish your effects

During the publishing process, you'll need to select an owner, publishers and platform for your effect. These options control which account manages the effect, which account the effect is attributed to and where the effect is published.

To publish a public effect:



Publish upload workflow within the app.

The publishing window from Meta Spark Studio offers the option to directly record a demo video that teaches people how to use your effect.

Once you choose whether you're publishing a new effect or updating an existing effect, you can upload directly to Meta Spark Hub. After the upload is complete, Meta Spark Hub will open automatically in your browser.

Optimize and export effects.

When you export an effect from Meta Spark Studio, an AREXPORT file is saved to your computer. This is the file that you upload to Meta Spark Hub when you publish your effect. It contains all the individual files needed for your effect to work.

Export an AR export.

The size limits for the files exported from Meta Spark Studio to Meta Spark Hub ensure your effects perform well across different devices.

- Instagram should be 4MB or less on both iOS, Android and older Android.
- Facebook should be under 2MB for the best reach, but they can be up to 10MB.
- Facebook ads should be under 5MB.

Platform review

To publish an effect you need to [upload it](#) to [Meta Spark Hub](#).

When working with clients you should let them know that it could take up to five business days for an effect to be reviewed and accepted.

If an effect is accepted, you'll be notified on Facebook and in Meta Spark Hub. Depending on the platform you pick, your effect will then be publicly available through the app's camera.

Until the effect is accepted, it won't be visible to the public. However, you can share a link to the effect with your friends and followers.

Sometimes an effect is initially accepted but a further review finds that it doesn't meet Meta Spark Policies. In this instance, your effect will remain visible but you'll need to update it within 10 days.

Example of a noncompliant effect

Consider this example of a noncompliant effect: You've created an effect for a client that promotes the use or depicts the sale of a potentially dangerous cosmetic procedure.

Reasons this effect is noncompliant	Examples of compliant face altering effects
<ul style="list-style-type: none">• Promotes potentially dangerous cosmetic surgery.• May include instructional elements such as surgery lines or needles.	<ul style="list-style-type: none">• Making a person's eyes or lips bigger or smaller• Transforming people into animals or fantastical characters



How to appeal a review decision

If your effect isn't approved, you'll be notified as to which issues need to be resolved. You can review the specific policies covering effect requirements in [Meta Spark Policies](#) and either [Community Standards](#) or [Community Guidelines](#).

To appeal a rejection, the rejected effect needs to be edited in Meta Spark Studio and resubmitted for review.

Once uploaded to Meta Spark Hub, you can select whether to make your effect available at all supported platforms or specific platforms only.

Meta Spark Policies

[Meta Spark Policies](#) are here to help you understand what type of content is and isn't allowed on Meta Spark. We encourage you to become familiar with the policies before submitting your content for publishing. The following are a few key highlights:

Community Standards	All content published to Facebook, Messenger or Meta Portal must comply with the Community Standards and all applicable laws, statutes and regulations.
Community Guidelines	All content published to Instagram must comply with the Community Guidelines and all applicable laws, statutes and regulations.

Content	<p>Content created in Meta Spark Studio must not:</p> <ul style="list-style-type: none"> ● Be shocking, sensational, disrespectful or excessively violent. ● Promote illegal products, services or activities or content that is inappropriate or unsafe. ● Promote alcohol, tobacco, marijuana or pharmaceutical medical products, nor be published by brands associated with such products. ● Be of an adult or sexual nature, including content such as nudity, depictions of people in explicit or suggestive positions or activities that are sexually suggestive or provocative. ● Discriminate against, harass, provoke or disparage people. ● Promote the use or depict the sale of a potentially dangerous cosmetic procedure as per Community Standards. This includes effects that depict such procedures through surgery lines.
----------------	---

Maintain, update and improve effects

You can update your effects at any time, but you may need to update older effects to make sure they continue to function.

Effects will be supported for six months after the release of the version of Meta Spark Studio you used to create them. After six months, effects might no longer be compatible with the latest versions of Instagram and Facebook, and people might not be able to find or use them.

If an effect needs to be updated, you'll get an email or notification on Facebook. You'll also see a warning in Meta Spark Hub.

You can make changes or updates to your effect after uploading it to Meta Spark Hub as well as updating the effect AREXPORT file. You can make changes to both published and unpublished effects.

Glossary

Acronym	Term	Definition
	Ambient light	Will provide each object in the scene with a constant amount of light. You can't change the position, rotation or scale of an ambient light.
	Animation patch	Used to drive an animation triggered by an interaction patch or logic .
	AR effect	A piece of AR content created through Meta Spark Studio, uploaded in Meta Spark Hub, and used by people on Facebook, Instagram and Meta Portal devices. AR effects allow people to create new content, such as stories, videos and GIFs.
	AREXPORT	Final zipped file that is created when you export an effect in Studio. 40MB limit.
	ARP	File extension used when you make a patch asset.
	ARPROJ	Project file (zipped JSON + internal assets). Instagram should be 4MB or less on both iOS, Android and older Android. Facebook should be under 2MB for the best reach but can be up to 10MB. Facebook ads should be under 5MB.
	Audio clips	Should be mono M4A, AAC codec, with a sampling frequency of 44.1KHz. You can import your own or choose from a huge range of free, licensed sound effects in AR Library .
	Audio playback controller	Renders sound in the scene only after it's connected to a speaker object. Use it to play sound continuously on a loop in your AR effect, or connect one-shot audio playback to boolean signals in Patch Editor.
AR	Augmented reality	An interactive version of reality created by the use of technology to overlay digital information on something being viewed through a device, such as a smartphone camera.
	Authoring tools	Help you create digital content. They can be classified as programming or content design tools.

Acronym	Term	Definition
	Blend mode	Sets the blend mode of blend material: <ul style="list-style-type: none"> • Add • Associative alpha • Alpha • Subtract • Multiply • Replace • Screen
	Blended material	Includes a blending mode to mix textures and colors together.
	BG influence	Multiplies a color or texture against a grayscale version of the video feed. Only available in the face paint material.
	Body tracking	Detects and tracks bodies in the scene.
	Brightness	Sets the brightness of a color. Only available in the face paint material.
	Camera	An object that's automatically listed in the scene panel in every effect. It represents the camera in the device showing your effect, which means it can't be removed from a project.
	Camera texture	A texture that extracts live video as it's being captured by the device camera. The camera texture can be combined with segmentation to separate someone from their background. You can also manipulate the pixels in the texture to modify the image.
	Canvas	A container for laying out 2D objects and making them responsive to the size of the device screen.
	Cheek patch	Used to position an object at a precise location on a person's cheeks, such as the center of a cheek or a cheekbone.
	Chin patch	Used to position an object on the tip of the person's chin.

Acronym	Term	Definition
Color LUTs	Color lookup tables	Color LUTs (lookup tables) are tables of RGB color values. In Meta Spark, you can use color LUTs to quickly create color-graded effects across the entire scene.
	Cull mode	Describes how material is culled. <ul style="list-style-type: none"> Front: Discard pieces of the mesh that face the viewer. Back: Discard pieces of the mesh that face away from the viewer. Most effects use this mode to improve performance.
	Device	An object that's automatically listed in the scene panel in every Meta Spark Studio project. It represents the device showing the effect, which means it can't be removed from a project.
	Demo video	Shows a preview of an effect to someone, giving them an idea of what it will look like before they open it in the camera. On Instagram, they appear alongside your effect in the effect gallery and in the effect tray.
	Directional light	Will shine on each object in the scene from the same direction, like the sun. It's a good idea to have a directional light in any scene that contains 3D objects, as they give objects a sense of depth.
	Double-sided	Displays both the front facing and back-facing sides of the mesh.
	Dynamic text	Used to create personalized effects that show text specific to someone's situation, such as their location or the date.
	Effect revision	Each effect has a number of revisions. Think of this as the versions of the effect. For example, the first revision is the effect that the creator uploads for the first time. When the name, icon, or demo video changes, a second revision is created for that effect with all the updated information.
	Effect owner	The personal Facebook account or Facebook Page that manages the effect and controls permissions .
	Enumerated	A data type that consists of predefined values. Enums provide a highly structured way to store data since they can store only a single predefined value.

Acronym	Term	Definition
	Environment texture	Used to mimic the light in a specific real-world environment. This is also known as using image-based lighting.
	Eyeball patch	Used to position an object at a precise location on a person's eyes. This patch is used for iris tracking .
	Eyebrow patch	Used to position an object at a precise location on a person's eyebrows, such as the middle of an eyebrow.
	Eyelid patch	Used to position an object at a precise location on a person's eyelid, such as the inside or outside corner of an eye. This patch also allows you to track the degree the eye is open.
	Face landmark patch	Used to capture the position of parts of the face, such as eyes and nose. This allows you to position objects like piercings to move in sync with facial and head movements.
	Face mesh	A 3D model of a generic face. It works in combination with the face tracker in Meta Spark Studio to create a surface that mimics someone's expressions.
	Face paint material	Use the face paint material to create a mask effect that shows the skin and features of the person using the effect, behind either a custom texture or a color.
	Face reference assets	A collection of textures and 3D objects. Use them when you're creating face effects in Meta Spark Studio.
	Face tracker	A smart object that detects and responds to the position and orientation of a face.
	Face tracker texture	Allows you to use the video of someone's face when they're using the effect as a texture in your scene. You can apply the texture to a material and object, for example, to create a face swap effect.
	Feature tracking	Tracks the perspective, including position, rotation, viewing angle and scale, of a predefined 2D pattern, typically an image. It works by tracking the 2D translation of individual feature points and solving for the change in perspective from frame to frame.

Acronym	Term	Definition
	Flat material	This material is more performant than the standard material and doesn't respond to lighting.
	Forehead patch	Allows you to position an object on the center or top of a person's forehead.
	Gallery picker	Lets people add an image from their phone's camera roll to an effect, like a green screen.
	Hand tracker	A tool that finds and follows a single hand in your scene and controls the position of any child objects.
HSL		HSL stands for hue, saturation, lightness and represents how colors appear under light.
HSV		HSV stands for hue, saturation and value. It's also commonly referred to as HSB, or hue, saturation and brightness. These are alternative representations of the RGB color model. In these models, colors of each hue are arranged in a radial slice around a central axis of neutral colors that range from black at the bottom to white at the top.
	Icon	An image that represents your effect in the camera. It can be either an image you upload or a frame from your demo video. You can upload this image to Meta Spark Hub while publishing your effect.
	Inspector panel	The Inspector panel is part of the Meta Spark Studio interface. It's the panel at the right-hand side where you can inspect and change the properties of objects.
	Linear space	The color and luminance values distributed in a straight line from bottom left to upper right.
Spark SL	Meta Spark Shading Language	The shading language for Meta Spark that enables you to program your own shaders.
	Microphone	Captures sound detected by the device's microphone when your effect is in use and plays it back in the scene. You can use the microphone to analyze and modify the voice of the person using your effect.
	Mouth patch	Allows you to detect a specific part of someone's mouth.

Acronym	Term	Definition
	Nose patch	Enables you to position an object at a precise location on a person's nose, such as a nostril or the tip of the nose.
	Null object	A container that is used to group objects. It doesn't have a shape, but it can be manipulated in 3D space.
	Occluders	Make AR effects more realistic by hiding things that wouldn't be visible in real life. Occluders are made by using an object and a material.
	Opacity	Sets the transparency of the material.
	Patches	Enable you to add interaction, animation and logic to effects. Patches act as visual building blocks, with each patch performing a different function. Connected together in what are known as a graph, patches pass and receive information to and from one another to make things happen in the effect.
	Patch asset material	Visual shaders created in Patch Editor then set as a shader type for a material in the Inspector.
	Patch Editor	Implements visual programming. <ul style="list-style-type: none"> • Lowers the barrier to entry for creators without a technical background. • Increases speed of creation of interactive effects. • Supports bridging with JavaScript..
	Particle systems	Let you display and move large numbers of objects, called particles. You can apply force and drag particles to mimic the effects of gravity. Use particle systems to produce all kinds of visual experiences, such as rain, smoke and confetti.
	Plane	A flat 3D object that can be placed at any depth within the scene. You could use a plane to place a 2D image or texture into 3D space.
	Plane tracker	A tool that tracks an infinite, horizontal plane. Currently, Meta Spark Studio supports only one plane tracker in the scene at a time.

Acronym	Term	Definition
	Point light	A sphere-shaped light that shines in all directions evenly. It becomes less intense with distance, like a lamp.
	Point tracker	Tracks a specified point.
	Physically-based material	Used to create realistic-looking objects in Meta Spark Studio. The different properties in this material allow you to add surface roughness, metallic effects and lighting that mimic real-world light.
	Publisher	A Facebook account, Page or Instagram account linked to the effect owner. The publisher controls the account your effect is attributed to.
	Reactive programming	A declarative paradigm which is primarily structured around asynchronous data streams and the propagation of change. Meta Spark Studio uses a reactive programming model. This model enables you to define relationships between objects and values in such a way that a value bound to a signal is automatically updated when the latter changes.
	Real-world scale	<p>Enables your object to be sized to fit in the real world. For example, if your effect allows people to place items from a furniture catalog in their living space, the 3D furniture should be sized based on the real-world space.</p> <p>There are two ways to use real-world scale in Meta Spark Studio depending on the type of experience you want to create:</p> <ul style="list-style-type: none"> • Turn on real-world scale in the Inspector. Your object appears realistically scaled but only after a real-scale signal is detected. This means the person sees the object abruptly change size at some point after opening the effect. • Turn on real-world scale in the Inspector and add further logic via the Patch Editor or scripting. This method allows you to hide the object until a real-scale signal is detected. This means the person only sees the object once it's realistically scaled.

Acronym	Term	Definition
	Rectangle	A 2D shape in Meta Spark Studio. Use it to render textures and materials and to create 2D effects like frames, backgrounds and colored overlays.
RGB		RGB means red, green and blue. An RGB file consists of composite layers of red, green and blue, each being coded on 256 levels from 0 to 255.
	Retouching material	Used to add retouching effects to faces and scenes.
SVG	Scalable vector graphic	A graphic file type that is particularly useful in AR effects, because it doesn't lose precision when it's rescaled.
	Script/scripting	A program or sequence of instructions interpreted or carried out by another program.
	Scripting API	Serves as a basis for using coding language, in this case Javascript, to execute programs.
	Segmentation	Used to identify and separate parts of the camera input, such as to separate someone's hair from the rest of the scene so you can change the color without changing anything else in the effect.
	Shader asset	A material type resulted from shader code.
	Shader code asset	Enables you to write custom shaders in Meta Spark Studio.
	Signals	Special objects that contain a value that changes over time.
	Simple shadows	Adds a shadow to a 3D object to make it look more realistic.
	Simulator	Represents a screen. Use it to preview how your effect will look on a device.

Acronym	Term	Definition
	Speaker	An object in Meta Spark Studio that renders sound in your effect. You can add multiple speakers to a project to render different audio clips.
	Spot light	A cone-shaped beam of light, like a flashlight or headlight.
	sRGB Space	sRGB is the standard red, green and blue color space and one of the color-encoded options supported in Meta Spark.
	Standard material	This material uses a lighting system called the Phong model that's good for simulating realistic lighting on 3D objects.
	Target marker	Helps people using the effect know what target to look for in the real world.
	Target tracker	Meta Spark Studio allows creators to create effects that get triggered when the camera points to an image in the real world, for example a movie poster. This target image is then tracked in the world using SLAM (simultaneous localization and mapping).
	Touch gestures	Include, for example, screen tap, object tap, long press, pan, pinch and rotate.
3D	Three-dimensional	Describes an image that has the perception of depth. For example, a shape or object that has three dimensions: length, width and height.
	Three-dimensional (3D) object	A 3D model or asset created in another program that can be imported into Meta Spark.
	Tracking	Tracking answers the question, “Where is the object of interest located in the camera frame over time?”
2D	Two-dimensional	A flat-plane figure with only two dimensions, such as width and height but lacking depth.

Acronym	Term	Definition
UX		Short for user experience.
UI		Short for user interface.
	UV map	A 2D representation of the surface of your 3D object.
	Vector objects	Used to render a scalable vector graphic (SVG) image in your scene.
	Viewport	The Viewport is where you can see and build your effect.
	Visual programming	Enables people to create effects by connecting patch nodes graphically rather than scripting them in code.
	Visual shaders	A shader is a set of algorithms that determines the appearance of a 3D object's surface. Shaders are particularly useful for applying complex changes to a material.

 Meta