



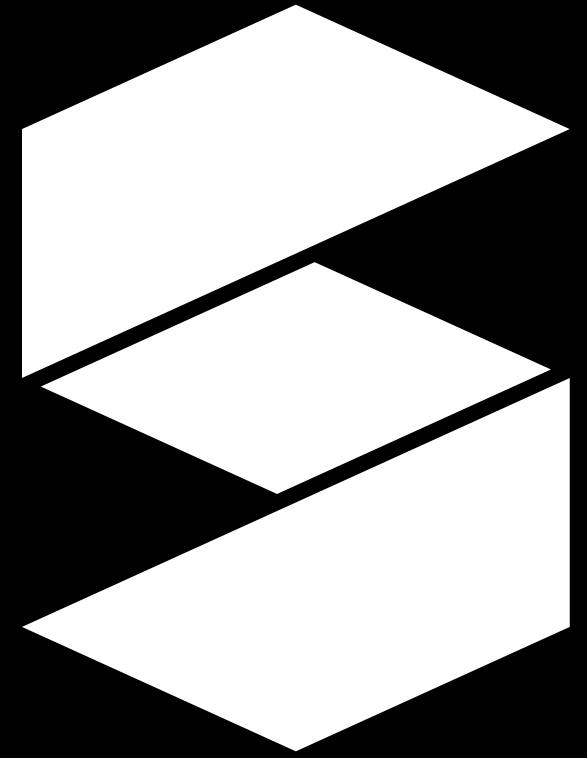
# Augmenting **Attention** through **Reality** Workshop

Kate Zaprazna

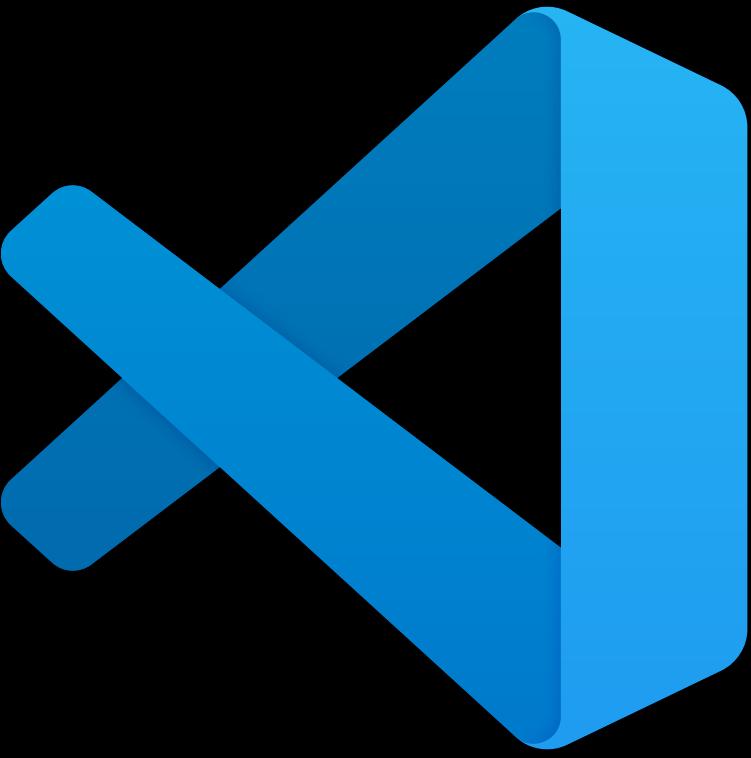
# Tooling



What would be working with?



Meta Spark Studio



Visual Studio Code

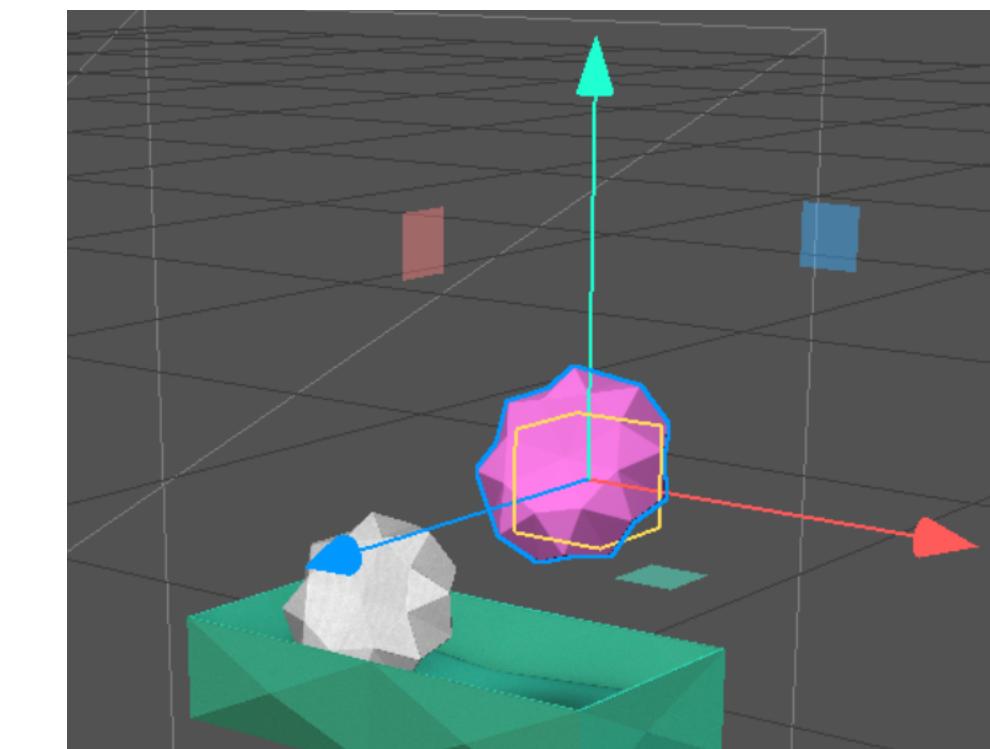
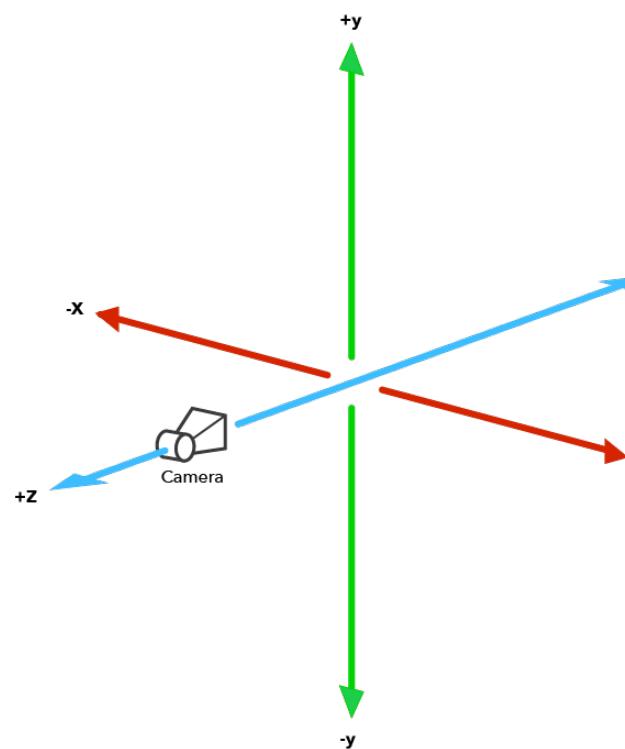
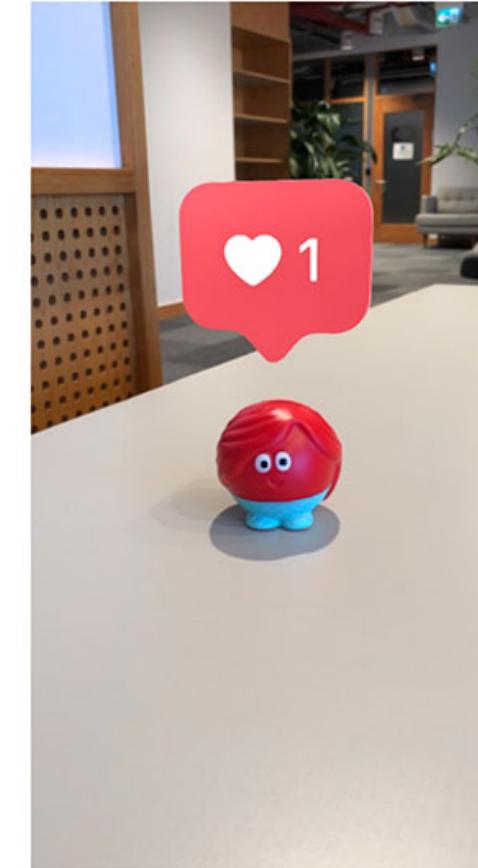
# Types of Spark experiences

1. **Facial effects** (freckles)
2. **World effects** (floating objects)
3. **Styles effects** (VCR, B&W)
4. **Light effects** (light beam, LED)

## A note on coordinate systems

Global coordinates describe an object's placement based on a fixed X, Y and Z axis. It stays in the same place in **Viewport**, even when we move it around to different parts of **Scene** panel.

An object's transform describes its placement in your scene: position, scale, rotation. Its position in **Viewport** changes according to where it is placed relative to other objects in **Scene** panel.

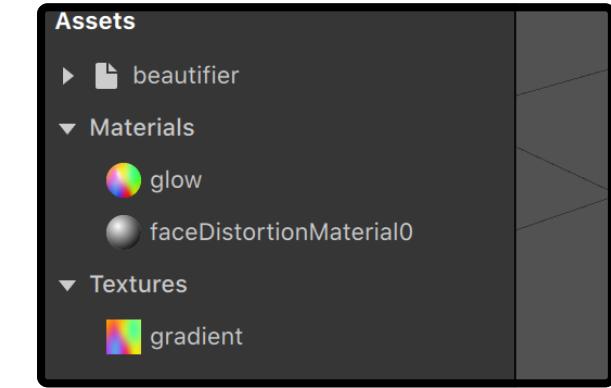
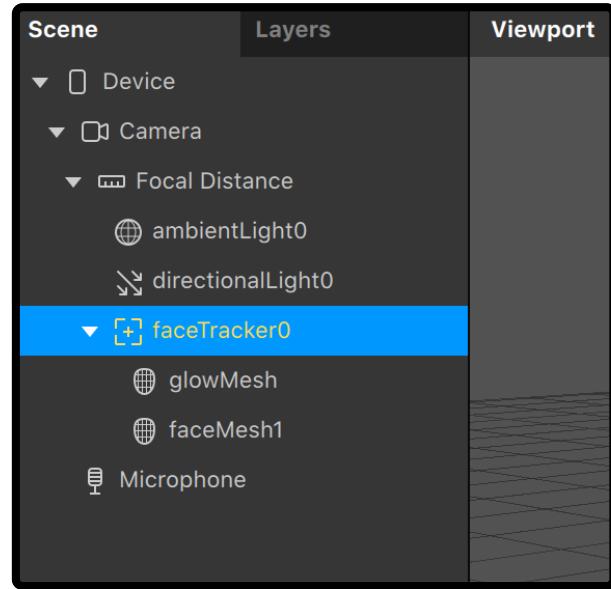


# Navigating Meta Spark Studio

Meta Spark Studio is an AR creation tool for building, testing and publishing AR experiences for effects, AR ads and multi-participant AR experiences.

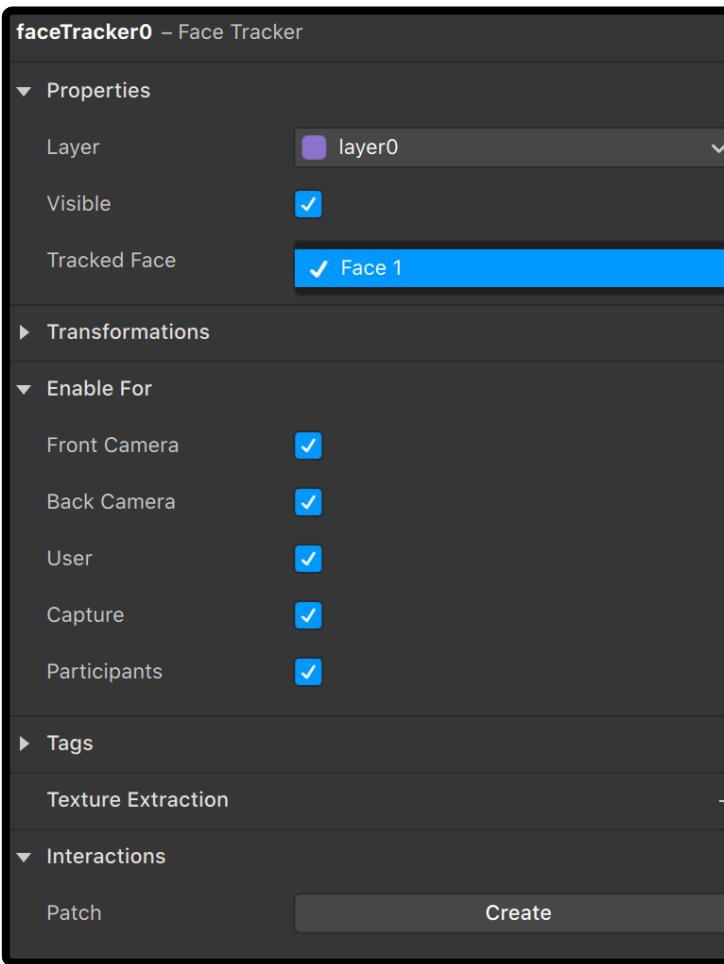
## 1 Scene Panel

Adding objects to **Scene** panel on the left of the interface will add them to your effect.



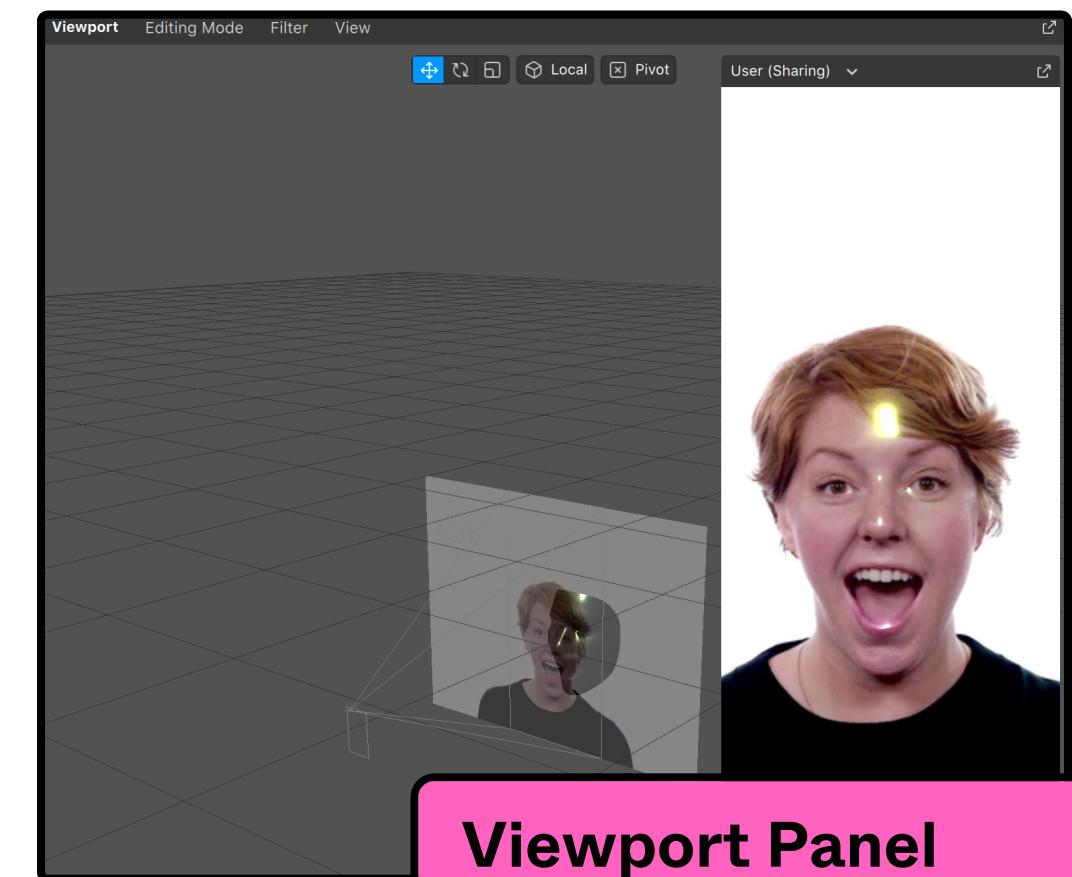
## 2 Asset Panel

This is where you can add assets from your computer to a project, or create new assets in Spark Studio.



## 3 Inspector

Editing properties of assets and objects. By selecting an asset its properties will be displayed in **Inspector**.



## Viewport Panel

You can see and work with your effect here. The Simulator represents a device's screen.

# Building a scene

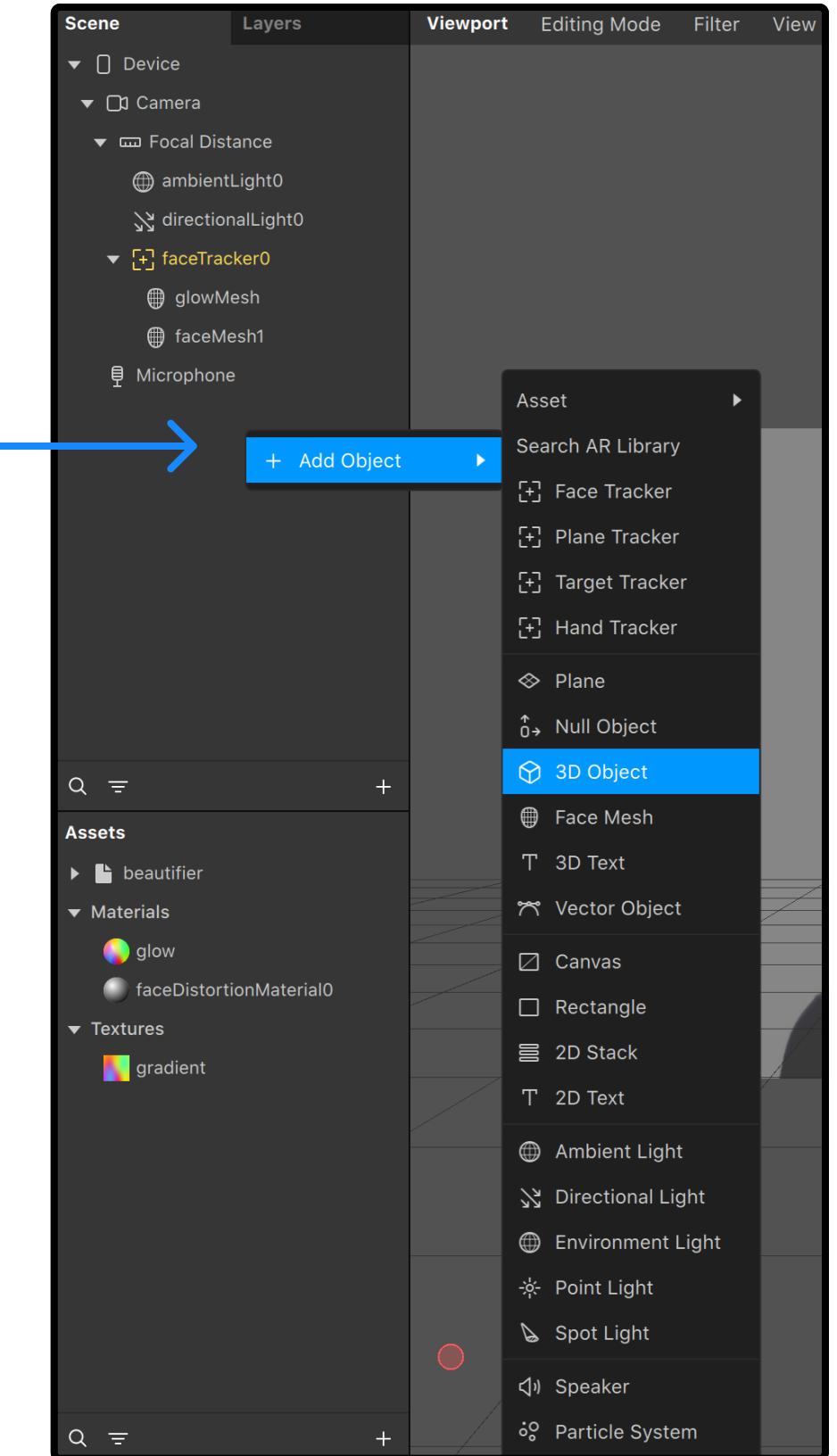
**Scene** panel is a place where your effects come to life. Whenever you add something to your **Scene**, whether it's objects or assets, they automatically become part of your effect.

The added objects can be abstract like a **Face Tracker** or tangible like a 3D model of a cat.

You can keep everything tidy by organising objects into hierarchies and grouping them together.

To add new object, right click in the Scene panel

Select one of the multiple objects from the menu.



# Previewing a scene

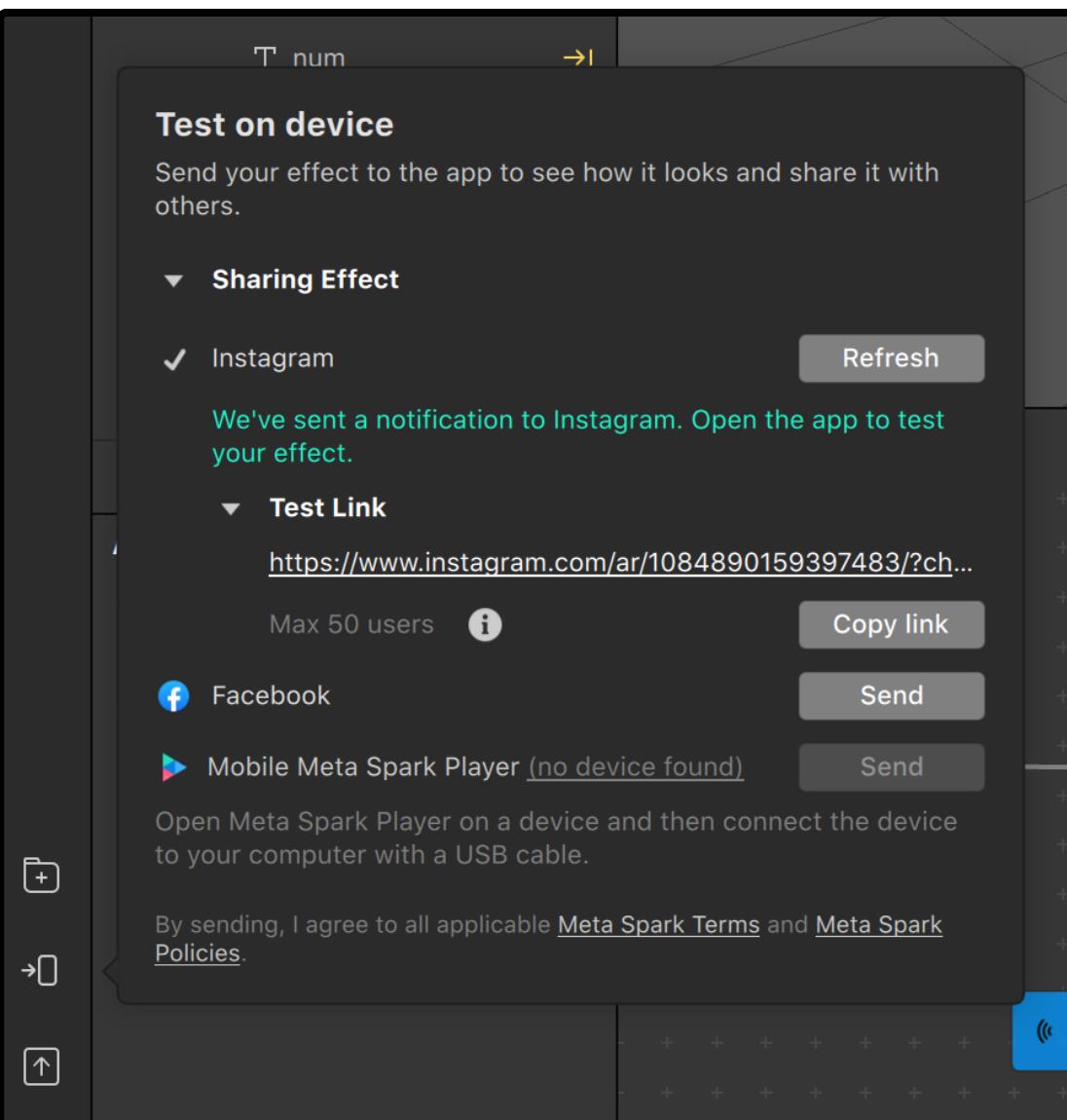
Did you know that you can preview your effects in **Simulator**? You can see them on the right side of your **Viewport**.

Your effects can also be previewed on your own mobile device.

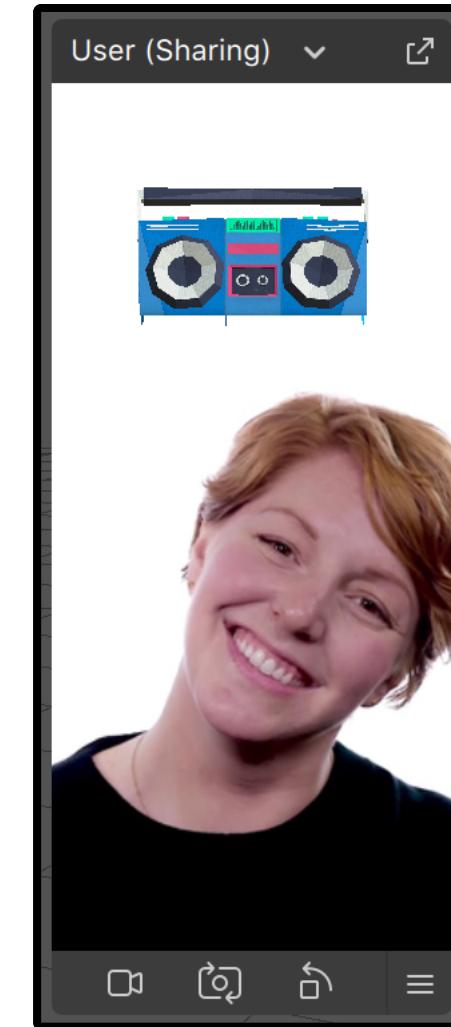
If you share them on Instagram or Facebook, you can simply send a link to your friends so they can check out your effects too! How cool is that?

## Different ways for previewing your effects

- Simulator preview
- Instagram preview
- Device preview (charging cable needed)



Test on device



Test on Simulator

# Spark Fundamentals



How to create an effect in Spark?

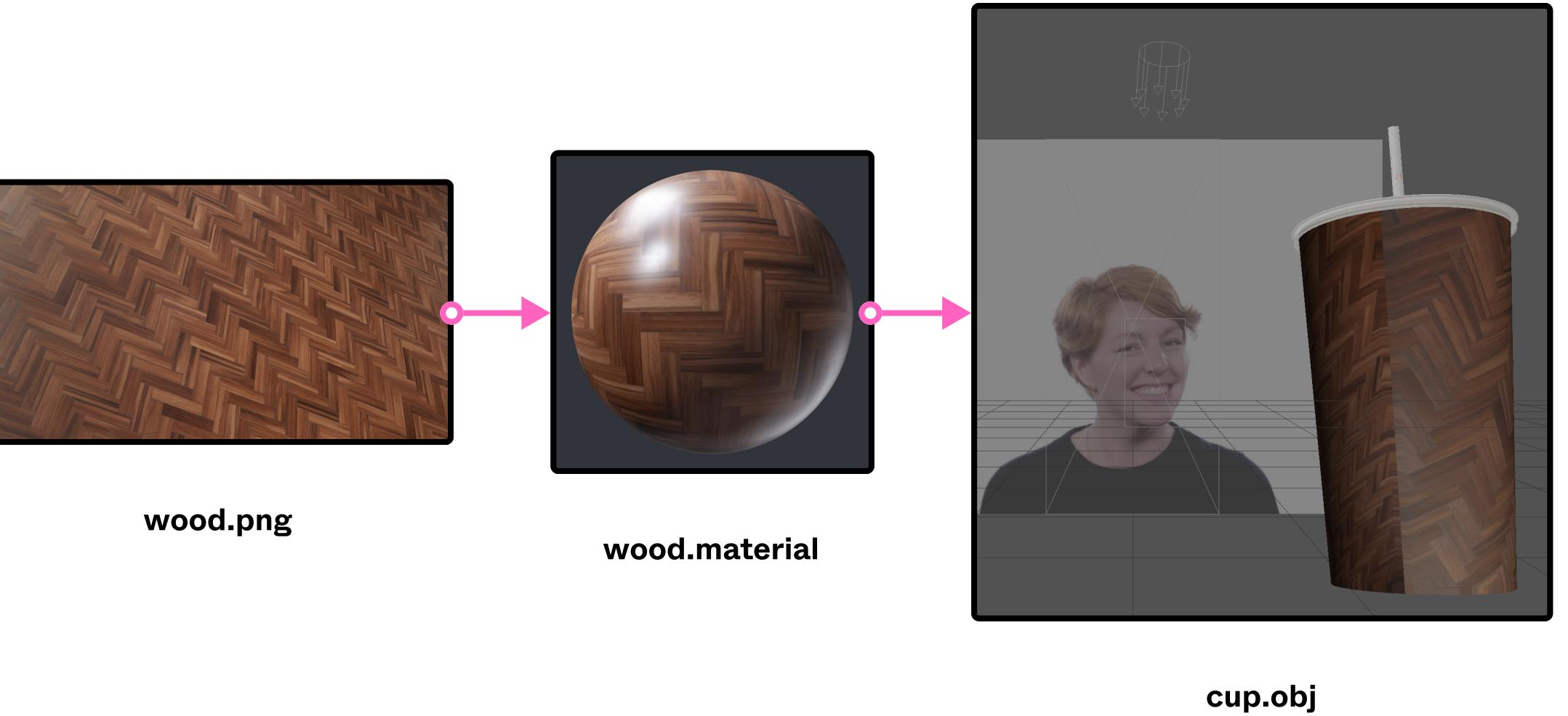
# Textures, Materials and Objects

**Textures** are assets used to add realistic lighting and colours to objects and applied to materials.

**Materials** specify how the surface of an object will appear, including light and texture. Materials are applied to objects.

**Objects** are the core building blocks of every Scene. Each object has a visibility property and can be conditionally shown.

- 2D objects (plane, canvas...)
- 3D objects (face mesh...)
- Trackers (face, hand...)
- Text (2D text...)
- Lighting (directional light...)
- Speakers (audio speaker...)



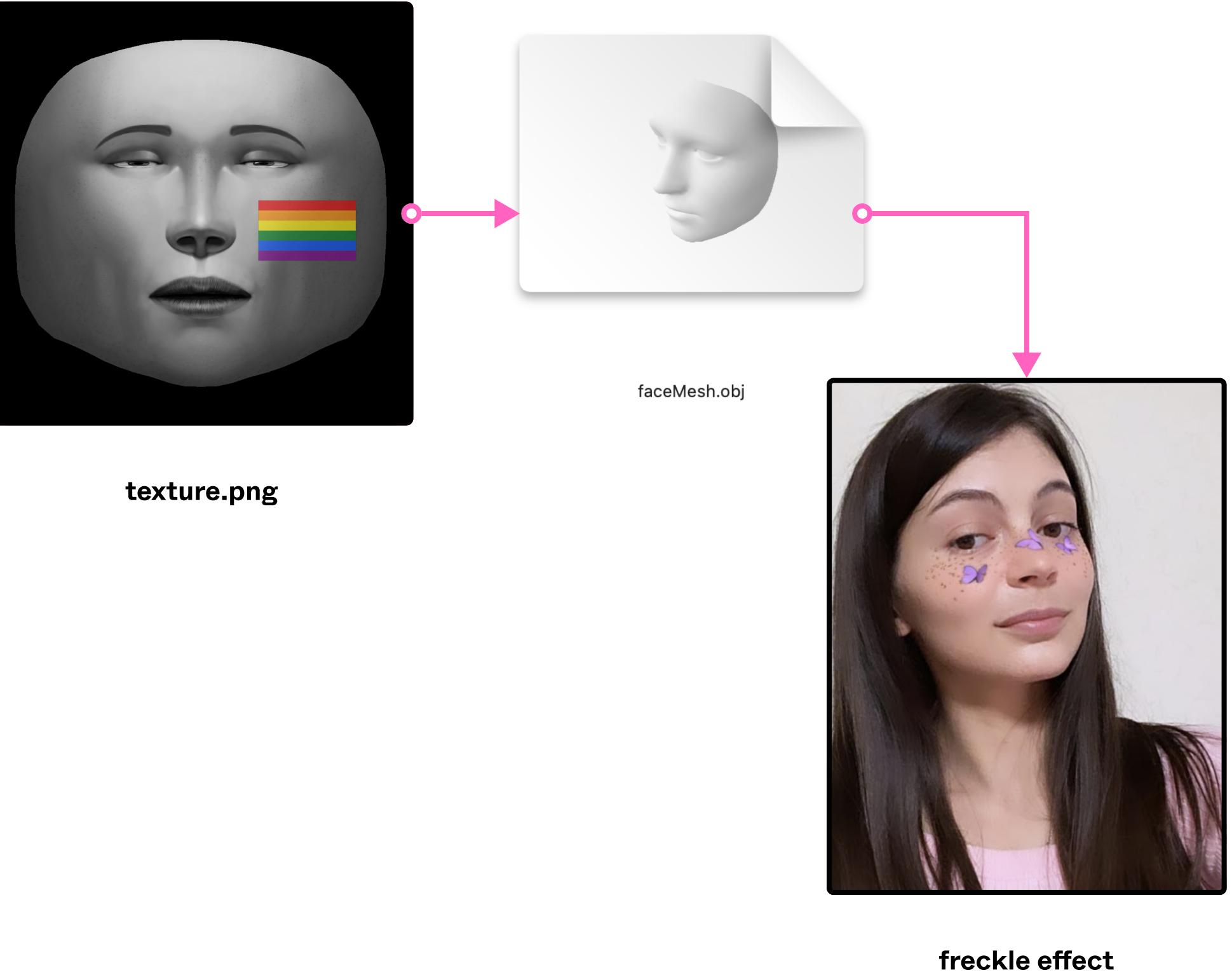
# Face Tracker

**Face Tracker** allows for facial features and expression detection (finding and following). Materials cannot be applied to a **Face Tracker**. Instead we add meshes.

**Face Mesh Mapping** allows us to create effects that wrap to fit the human face perfectly.

**Mesh** is a 3D model of human's facial anatomy. **Texture** applied to **Face Mesh** will wrap anatomically.

To ensure your **Texture** has a transparent background, save it in the PNG file format.



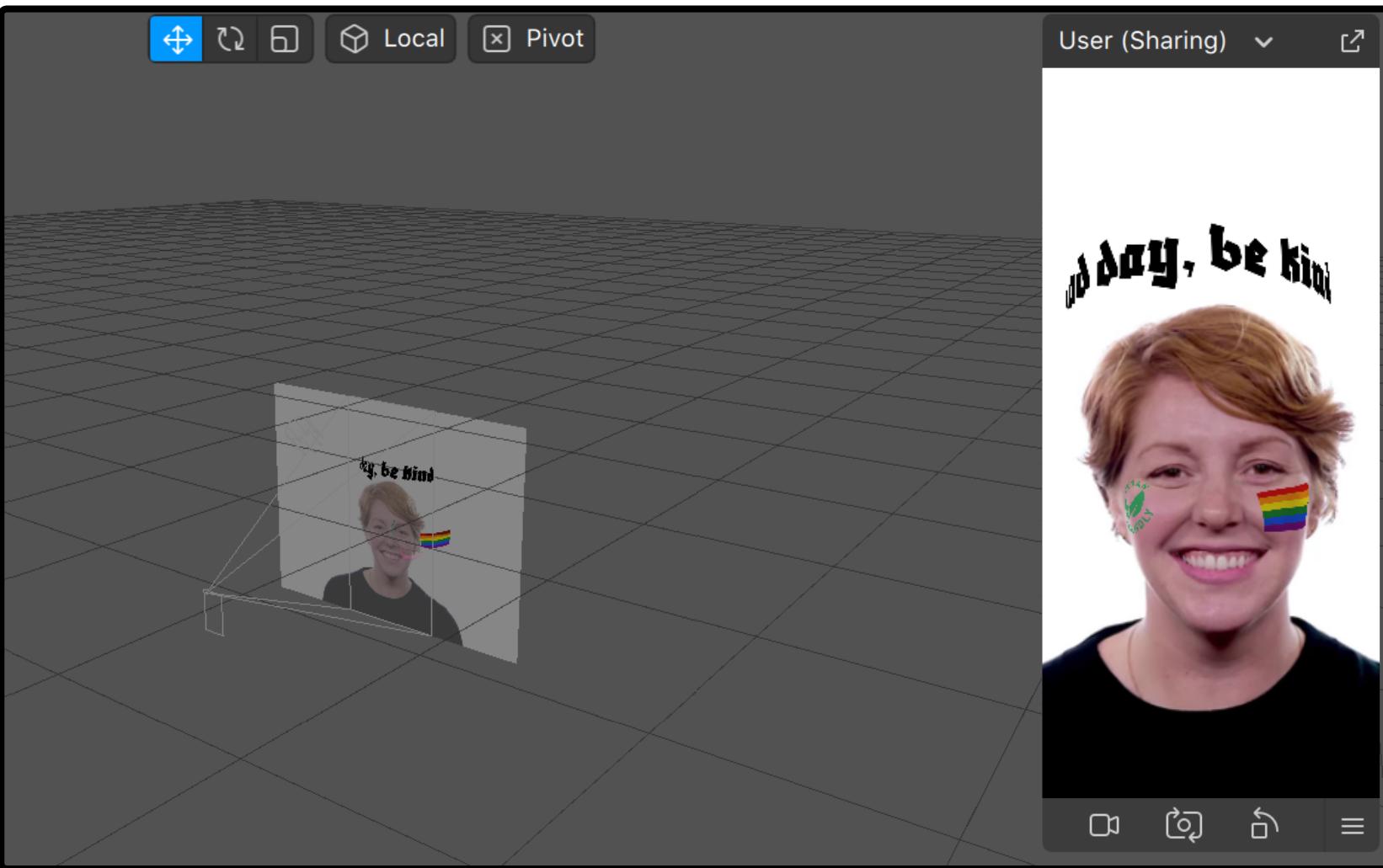
# Demo 1

How this effect could be build?

- Add **Face Tracker**
- Add **Face Mesh**
- Prepare texture for the **Mesh**
- Prepare a lip mask asset
- Prepare **Text** asset

What else we need?

- Position **Text** around user's head
- Rotate **Text**



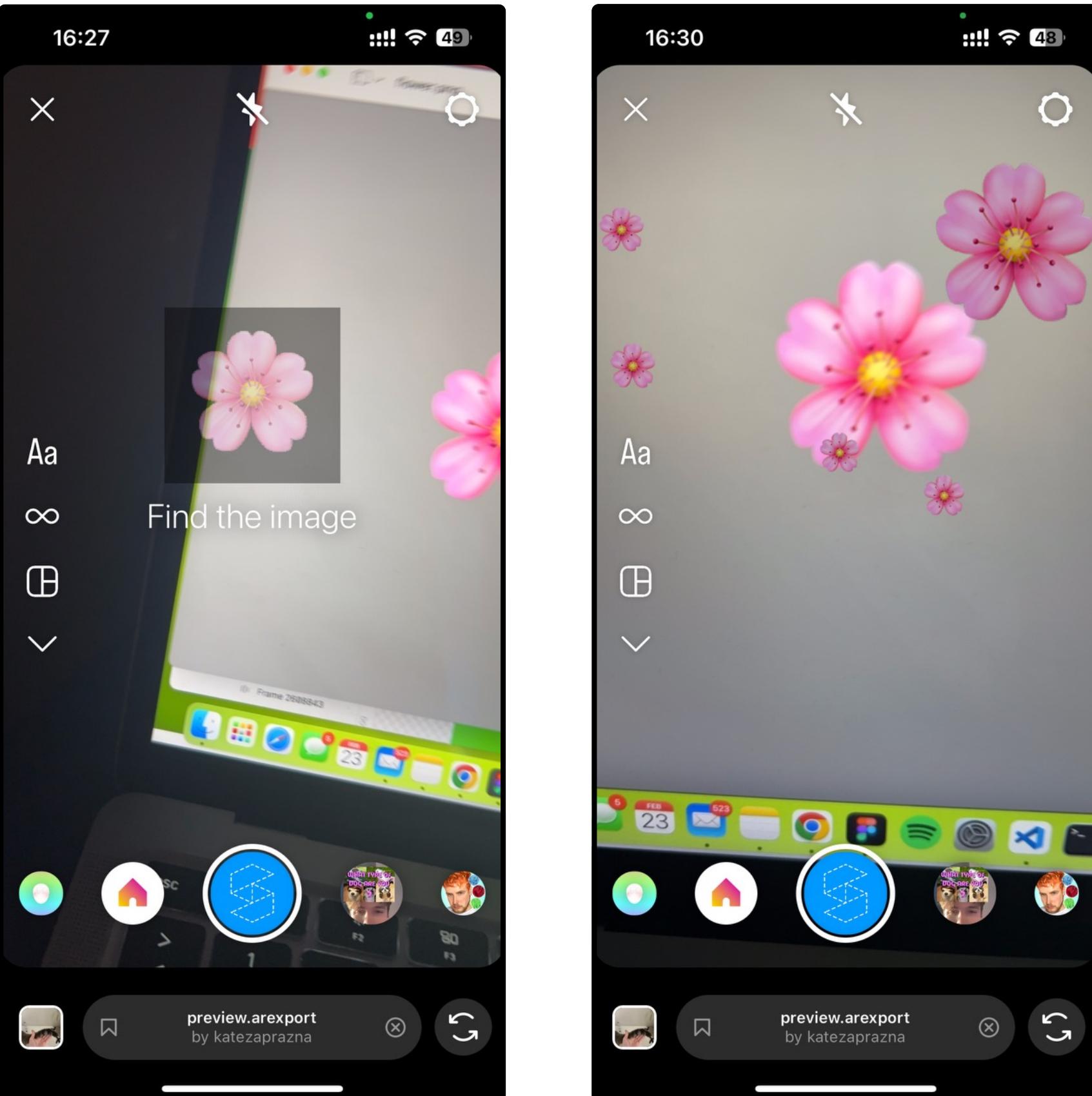
Let's try this out

# Target Tracker

**Target Tracker** in Spark is used to trigger an effect when the camera is pointed at an image in the real world, like a banner for a movie.

There are two target types: **Fixed** or **Moving**.

- Add **Target Tracker**
- Add image to trigger the effect

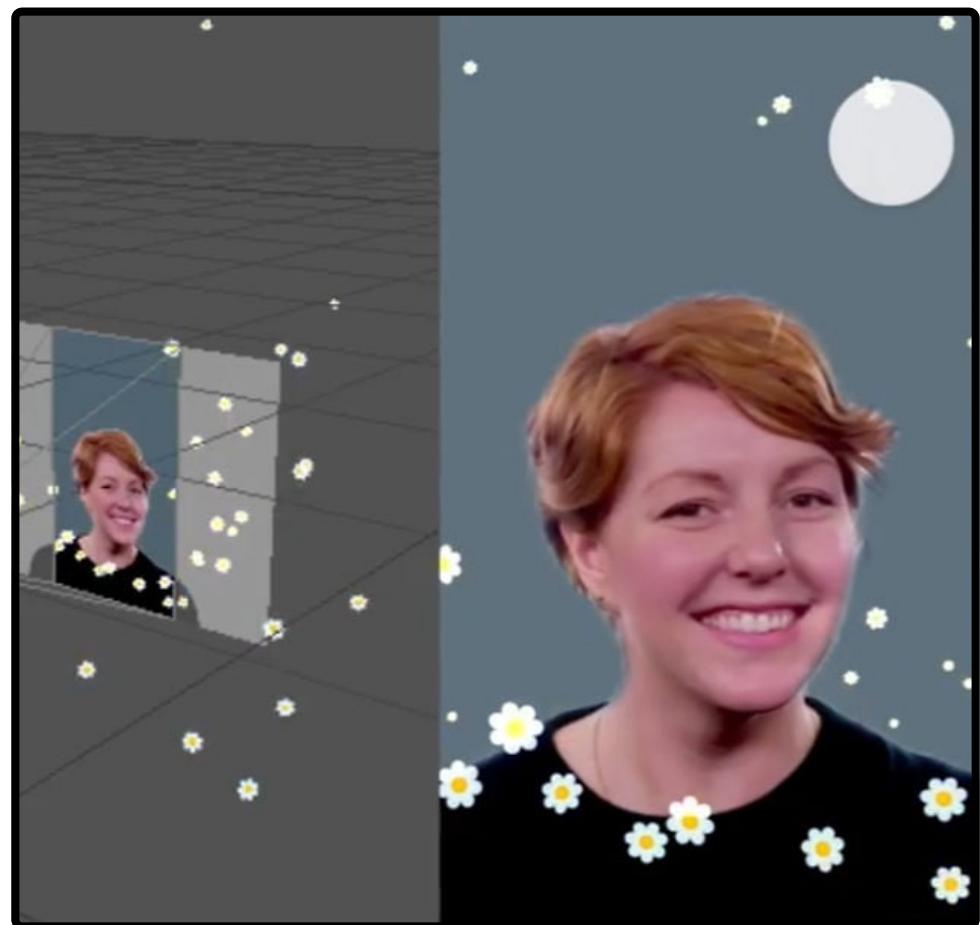
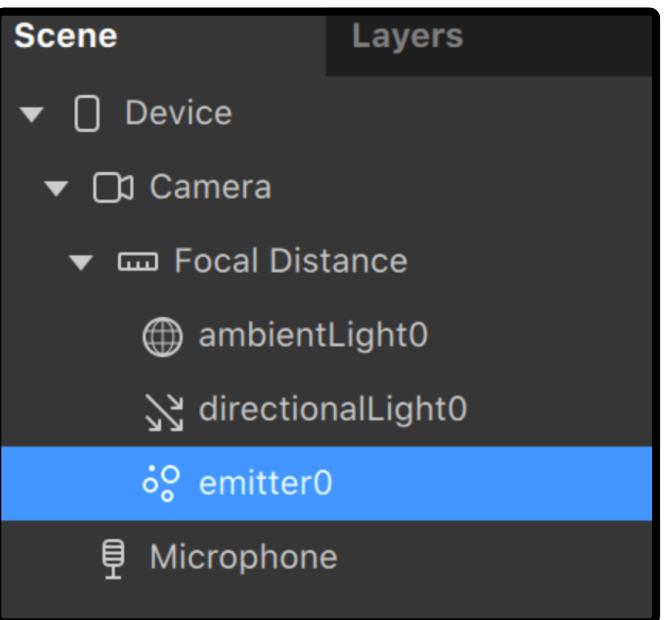


# Particle Systems

**Particle Systems** let you display and move large numbers of particle objects to which you can apply force and mimic gravity.

**Particle Systems** are useful for recreating the effect of rain, smoke or confetti.

After adding the **Particle Systems** object to the Scene, a new **Emitter** object is created. **Emitter** renders a small stream of particle objects. The appearance of particles can be customised through **Inspector**.



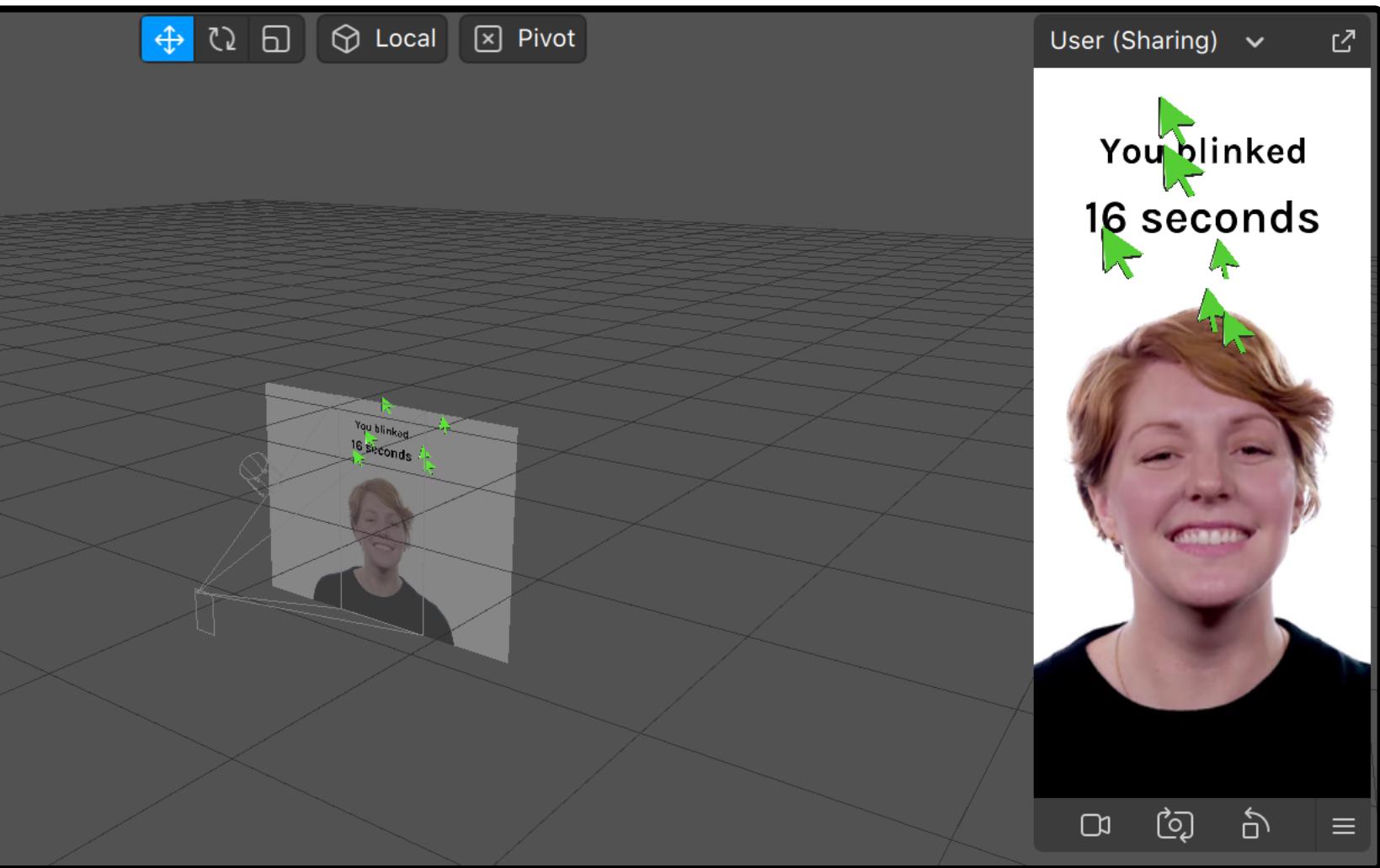
# Demo 2

How this effect could be build?

- Set up **Face Tracker**
- Add text as the title
- Add text representing the timer
- Set up **Particle System**

What else we need?

- Know when the user blinks
- When they blink, modify the interface text



Let's try this out

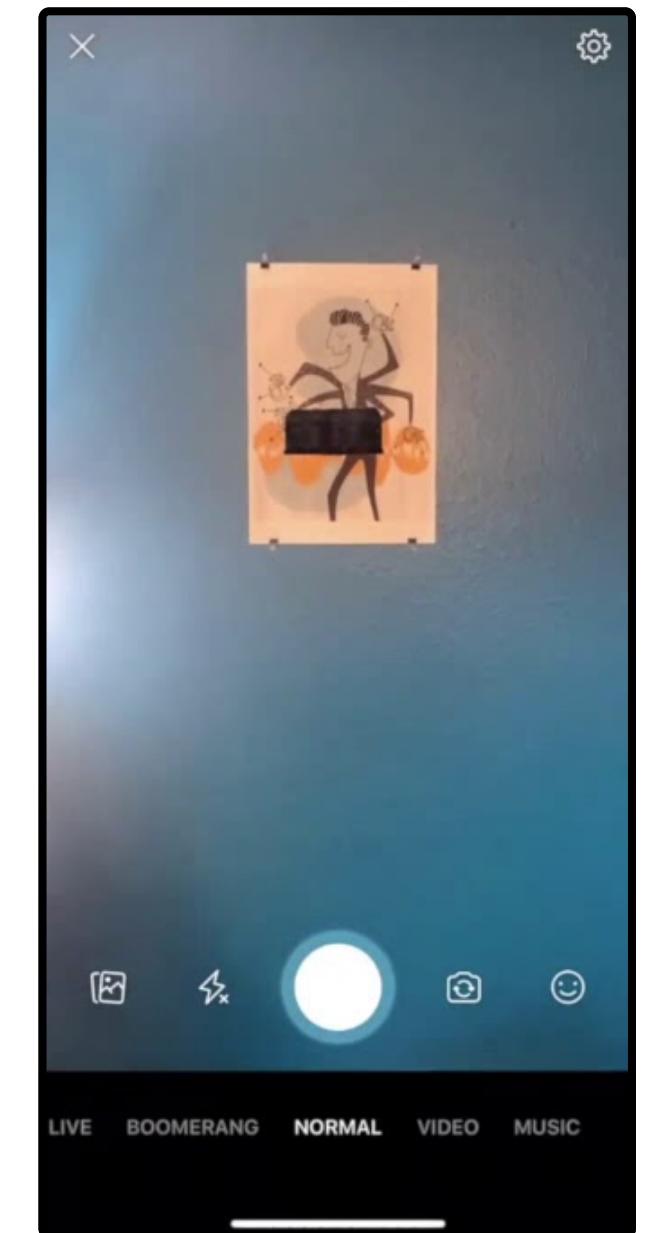
# Canvas, Plane, Audio and LUTs

**Canvas** is a layer scaling with the device's viewport and a necessary base for 2D objects such as **Rectangles** or **Text**. **Canvas** is typically used in "World Effects" to position objects.

**Planes** are flat 3D objects allowing for incorporating **2D Images** or **Textures** into a 3D space. **Planes** are typically used in "World Effects" to position objects onto horizontal surfaces like a floor or the table.

**LUTs** are pre-made colouring effects used to alter image contrast, saturation and so on.

**Audio** in Spark exists in three categories: music, sound effects (camera shutter) and ambient sounds (rain sounds).



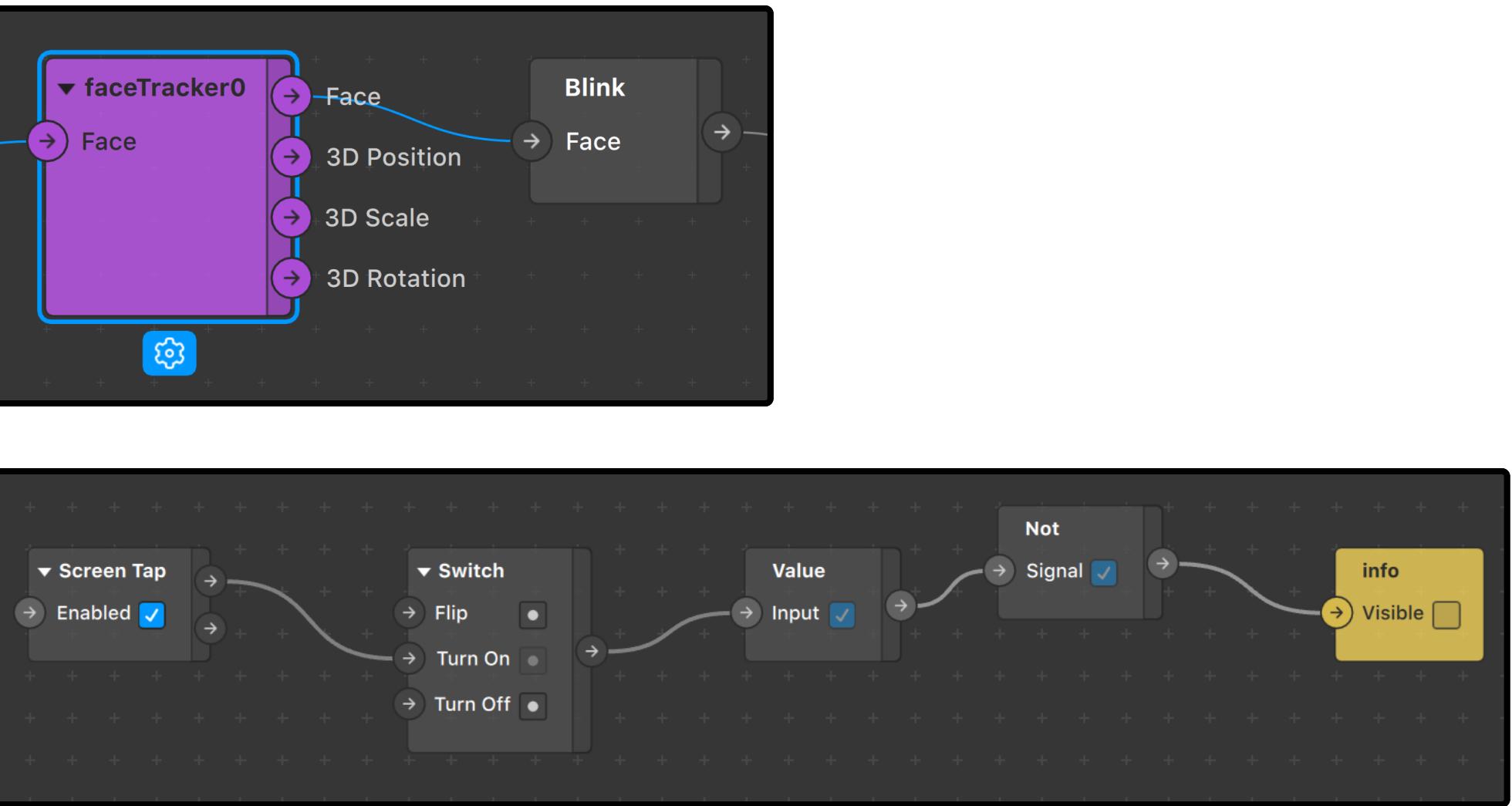
**Plane Object**

# Patch Editor

**Patches** are pre-made code blocks that together form a visual programming language. By connecting patches, you can add interaction to your effects.

When connected they pass and receive data between each other. Each **Patch** contains data and functions.

- Animations
- Audio (voice distortion)
- Face Landmarks (lips, cheeks)
- Body Landmarks (head, neck)
- Device Patches (rotation)
- Movements (blinking)
- Gestures (making happy face)
- Touch interactions (screen taps)
- Logic (AND, OR, NOT, XOR)
- Math
- Time (track time passages)
- UI (picker elements)
- Shaders



# Scripts

Besides **Patches**, Spark supports JavaScript for adding logic, which can be edited in a standard code editor. Interacting with Spark via code is always **asynchronous**.

Any functionality or object we want access has to be **imported**. Properties aren't accessed via **Inspector** panel, but via the **dot notation** `scoreText.text`.

```
● ● ●  
import Scene from "Scene"; // get access to Scene panel  
import TouchGestures from "TouchGestures";  
  
// communicate with Scene panel via async function  
(async function () {  
    // ask Scene to fetch the objects you want  
    const [scoreText, planeTracker, speakerLeft] = await  
        Promise.findAll([  
            Scene.root.findFirst("scoreText"),  
            Scene.root.findFirst("planeTracker0"),  
            Scene.root.findFirst("speaker_left_jnt"),  
        ]);  
  
    // access the objects Spark fetched from Scene  
    scoreText.text;  
    scoreText.text = 0; // set the initial value of scoreText  
    scoreText.hidden = true; // hide scoreText from your effect  
  
    // do something when user pans the screen  
    TouchGestures.onPan().subscribe((gesture) => {  
        planeTracker.trackPoint(gesture.location, gesture.state);  
        // allow users to swipe your effect's UI to see more elements  
        // change viewable area as user's finger moves across screen  
    });  
  
})(); // invoke the async function
```

# Testing vs. Publishing

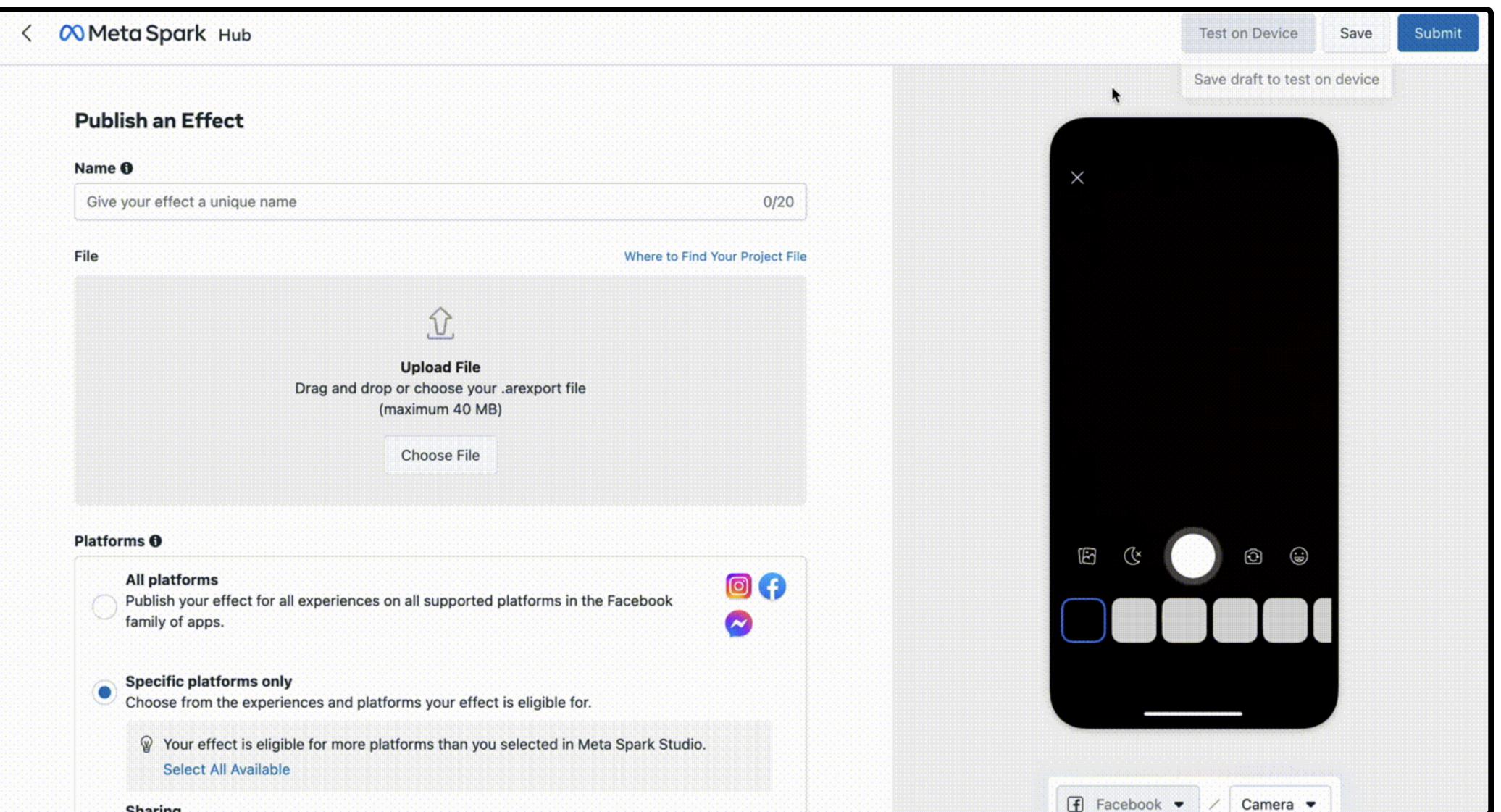
## Testing

Testing your effect only exposes the effect to you and other testers with whom the test link is shared. To make your effect available to others, you need to publish it.

## Publishing

The workshop does not cover effect publishing in great depth. In brief, to publish your effect, click on the **Publish button in Toolbar**.

Provide information about your effect, such as its name, category, icon or demo video. Once your effect is submitted, it generally takes 10 days to receive the reviewer's decision. To read more about effect publishing head [here](#).

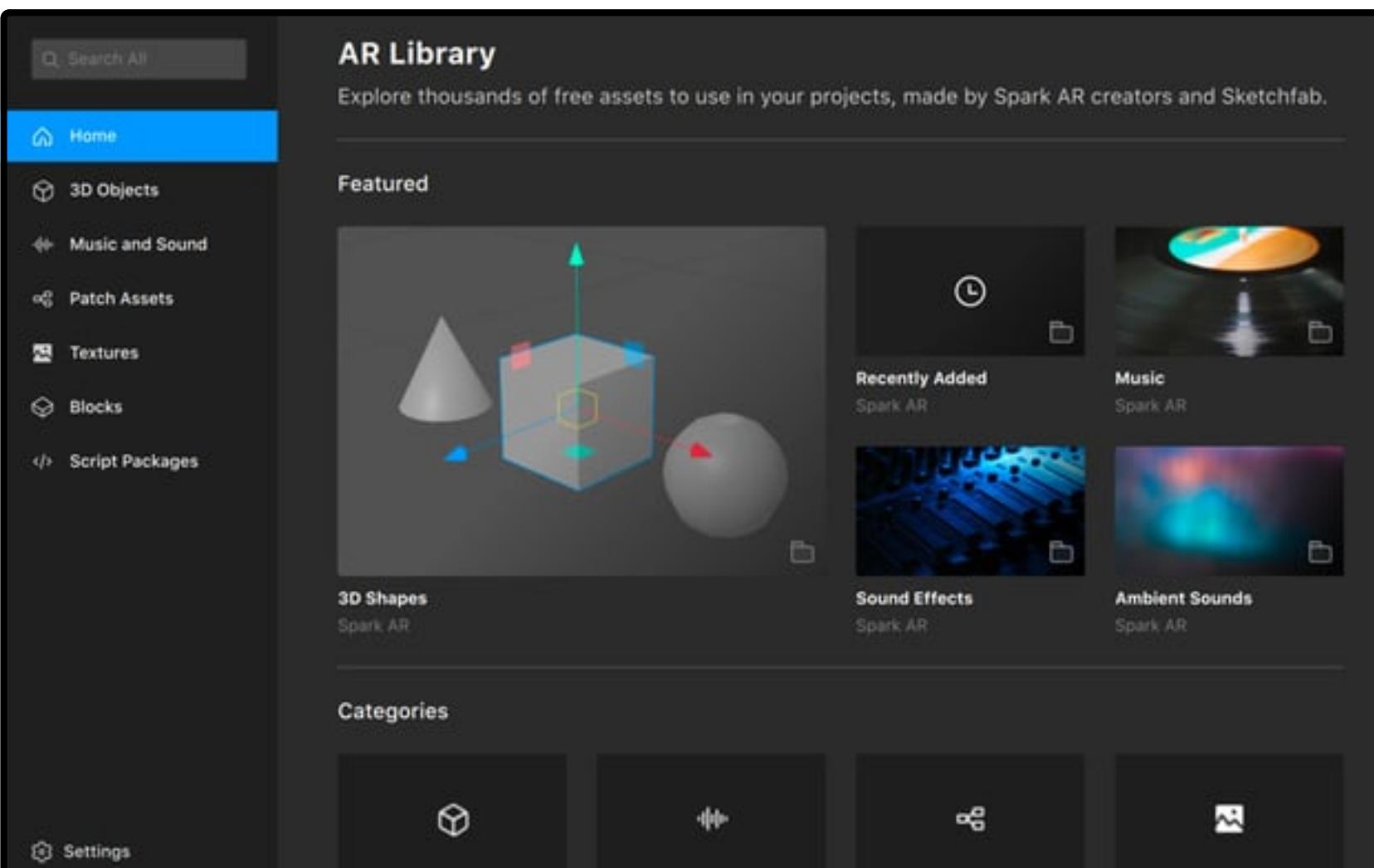


# Spark Library

**Spark AR Library** can be accessed via **Toolbar**.

Besides the assets we provided you with, you can find plenty more over there such as:

- 3D assets there such (cube)
- Music and sound
- Patches with already-implemented interactivity
- Textures
- Blocks, importable Spark projects you can build upon
- Scripts (code snippets)
- Colour LUTs



# Practice time ✨

- Brainstorm alone or with your peers
- Consider making a sketch
- Break your effect down into manageable steps
- Google and have fun

# Brainstorming 😎

- What do you like and don't like in AR effects in terms of their visuals?
- Do you want your filter concept to be more traditional or innovative?
- On which occasions do you want users to use your filter?
- What are some words, concepts or ideas that have inspired you recently?
- Do you have a favourite AR effect? If yes, why do you think you like the effect?

# Thank you



@kateHCl