



2. Click for Sound

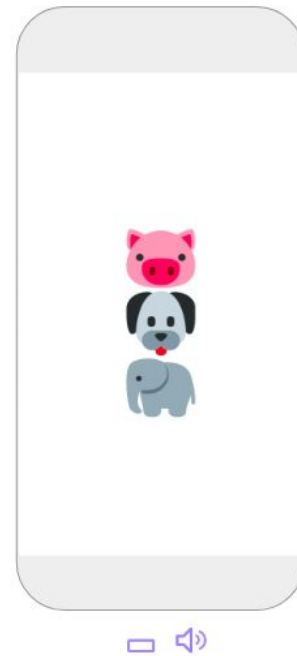
Objective:

In the sample app, students build a simple “See and Say” app that uses the `Sound` component. Since the source of the sound component can be changed students learn to reuse components rather than unnecessarily duplicating them.

Design Concepts:

Uploading Images and Sounds [[docs](#)]

When adding assets to your app these can be uploaded directly to your app or you can use assets that are stored in the cloud. There is a trade-off to consider here. While local assets may load faster, there is a limit of 50MB on eachThunkable app. If you choose to use cloud storage then your app will be “lighter” (take up less space on disk) but it will require your end user to have an active internet connection.



Absolute Sizing [[docs](#)]

All your mobile devices have physical screens that are measured in pixels, for example 1024 pixels wide and 768 pixels high. In this example absolute sizing is used to create multiple buttons that all have the same dimensions. This is a fairly easy concept to understand, so we’re introducing it here. It’s worth pointing out that there is no such thing as a standard pixel so as our UIs increase in sophistication we’ll introduce more robust ways of sizing visible components.

Sound [[docs](#)]

You can use the sound component in your app to play audio files that you have uploaded directly to your app or you can stream audio files that are stored online.

Blocks:

Component Reuse

The key programming concept introduced in this app is that it is much more efficient to reuse one component (`Sound1`) and programmatically change its source files than it is to create multiple sound components and assign each one a different source file..

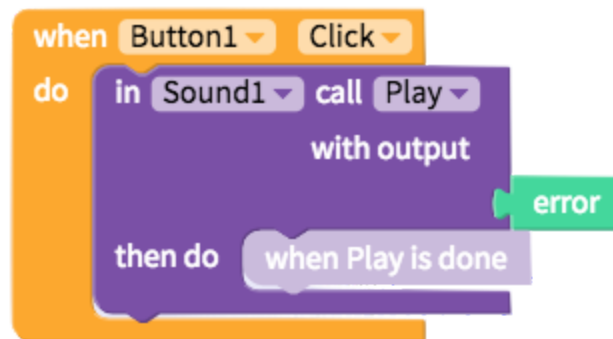
Asynchronous Events.

Any time you see a procedure with a “When N is done” placeholder block you know you’re dealing with asynchronous events. Essentially these are events that take some unknown amount of time to complete and they report back to the app when they have finished. The example that we see in this app is “in Sound1 call Play” which is asynchronous because we don’t know how long the sound will take to finish playing.

Instructions:

1. Drag and drop a Button component into the phone.
2. In the Voice section, drag and drop a Sound component into the phone. Unless you've completed the progressions above this will be a new component to you.
3. In the properties for **Sound1** on the right, click the source property to upload a sound file. You can also type in a url like this:
<https://thinkable.github.io/digital-asset/provided-assets/cow.wav>
4. Click the blocks tab and open the drawer for **Button1**. Drag and drop the "when Button1 Click" block into the blocks editor.
5. Next, open the drawer for **Sound1**. Drag and drop the "in Sound1 call Play" block into the "when Button1 Click" block.
6. Congratulations! You just built your second app. When you click the button, your app should now call a sound.

When the app is finished the blocks should look something like this:



Sample App:

Call of the Wild <http://bit.ly/thunk102>

Progression:

- Add more Animals
- Substitute in new icons and sound effects
- Create custom icons and record sound effects.