Step 1a: Create a Github.com account. (Note: If you already have a Github account you can proceed to Step 1b)

1. Go to https://github.com

2. Follow the sign-up workflow to create an account. You'll need a working email address to confirm your new account.
    a. You want "Unlimited public repositories for free". Leave the checkboxes unchecked, then click Continue.
    b. Answer the questions on the next page as appropriate.
    c. You can stop at the "Learn Git and Github without any code" page
    d. Now check your email, find the email from Github and verify your email address

Step 1b: Log into Glitch and download the starter code

1. Go to https://glitch.com

2. Click Sign in (top right corner)

3. Click Sign in with Github and use the Github credentials you used to create your Github account, or, since you're probably already logged in to Github, just click the green Authorize FogCreek button
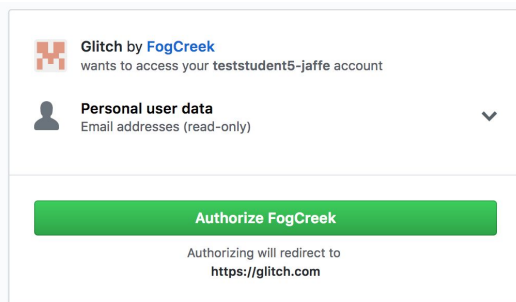
4. Click the New Project button (top right corner)

5. Click the Hello Webpage choice

6. Now we have to create a new project from a code repository. Click the name of your project in the top left corner which will be two randomly-generated words like "periodic-panda" then click New Project.

7. Click clone-from-repo

8. Enter https://github.com/JaffeAPCSP/aframe-tutorial.git in the import box and click OK

9. Let's test your installation: After the files are imported, click Show Live 👓 Show Live . Your browser should open a new tab and you should see the scene view below:
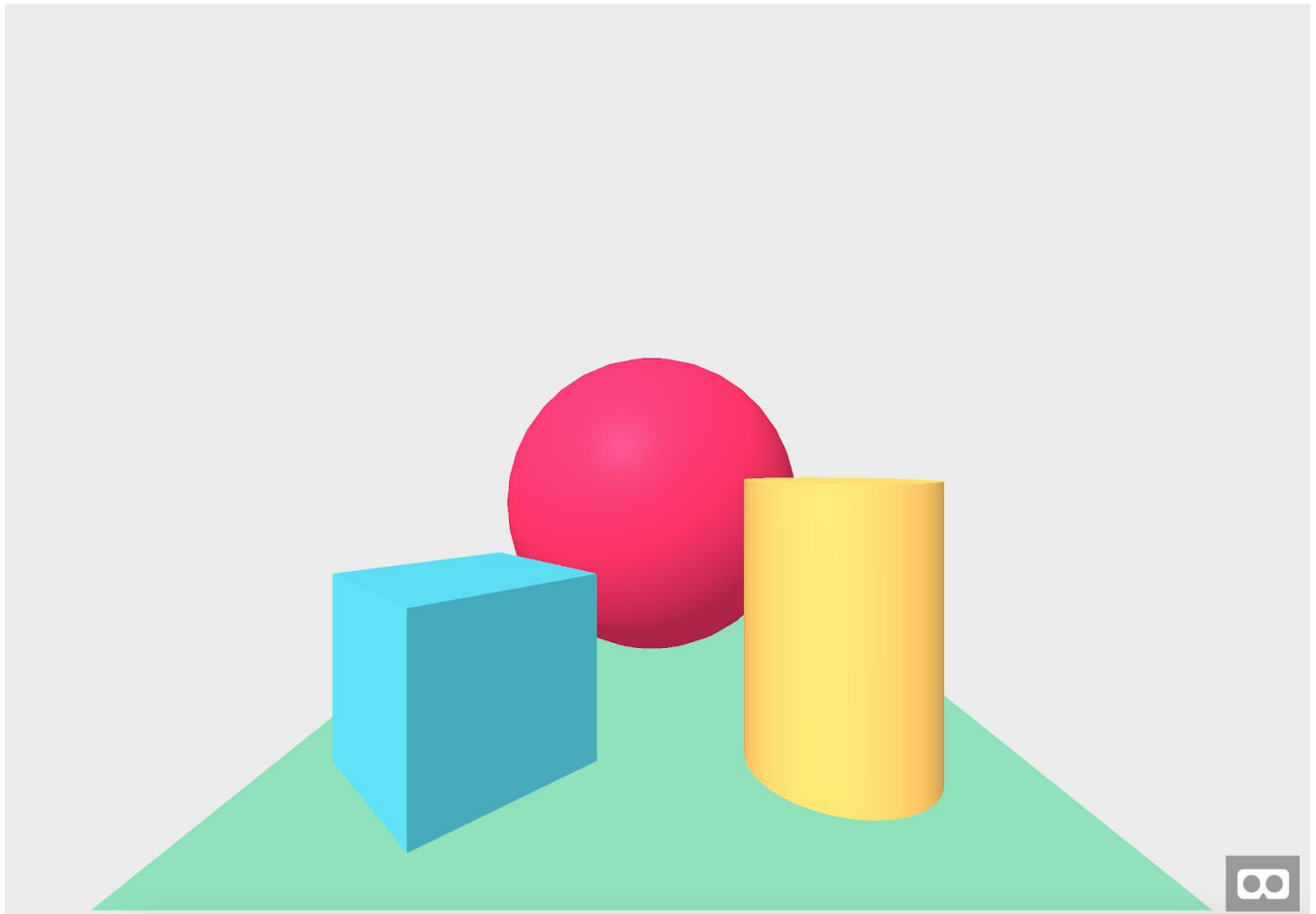

10. If you click and hold the mouse button, then drag it around you can move the view as if you were sitting in the middle of a room looking at the shapes

11. You can load this image onto a phone or other portable device by opening a Chrome web browser on the phone (you'll have to download the Chrome app either in the Google Play Store for Android devices or the iTunes App Store for iOS devices) then going to the same URL that your computer's web browser is on. The URL will look like `https://<your-project-name>-<random-number>.glitch.me`. Clicking on the Google Cardboard icon in the bottom right corner should switch your portable device into VR mode. Safari (on iOS devices) is also supported and will work, but Chrome works better.
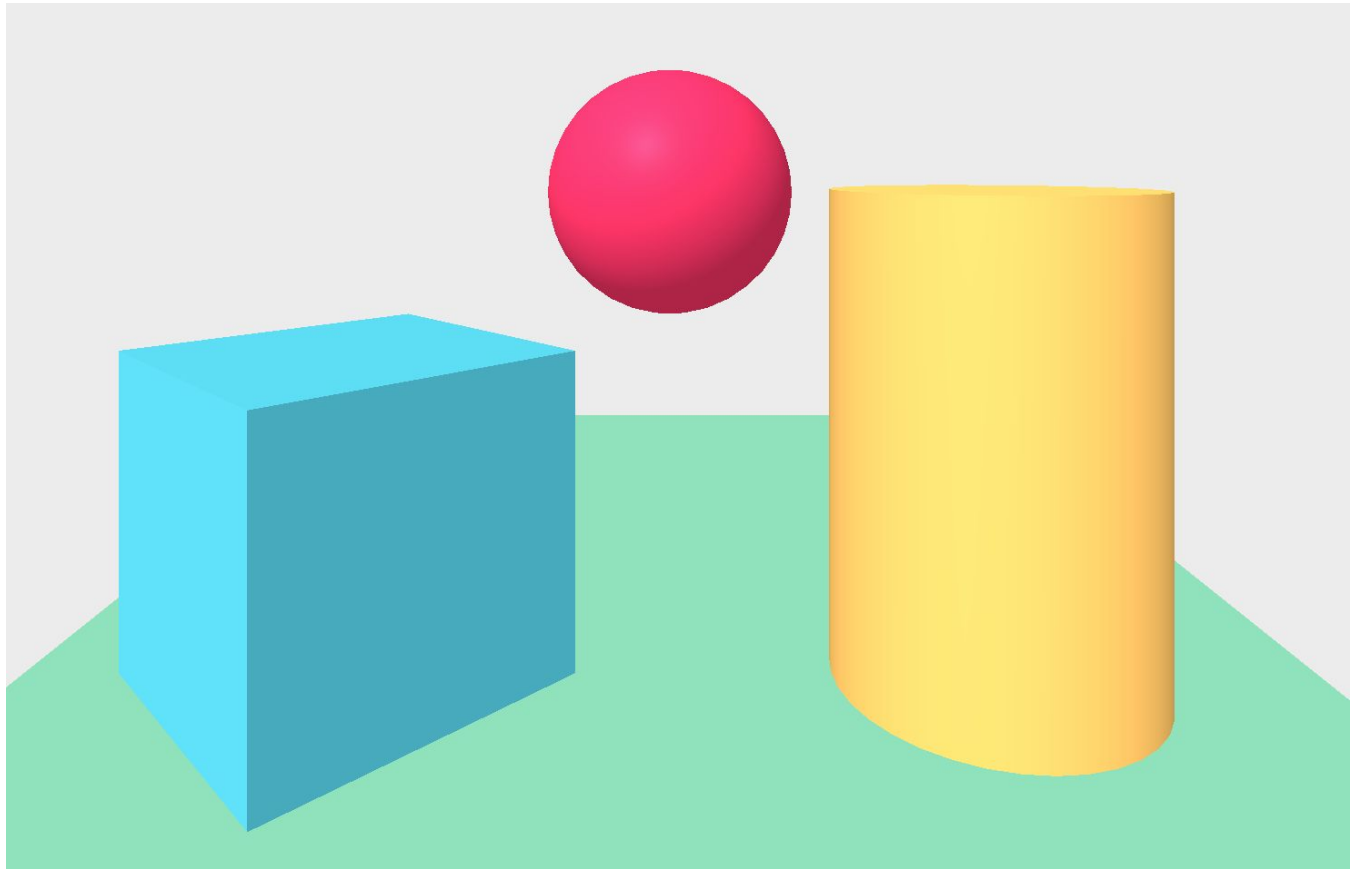
Step 2: Positioning in 3D

We describe positions in 3D space with a number triplet which corresponds to the *x*-, *y*-, and *z*-coordinates. This means that (1, 2, 3) corresponds to a position x=1, y=2, z=3

- *x*-axis: Moves the object left (negative) and right (positive).
- *y*-axis: Moves the object up (positive) and down (negative).
- *z*-axis: Moves the object away from the viewer (negative) and behind the viewer (positive)

In the view below, the red sphere is located at position (0, 1.25, -5) -- the middle of the window left-to-right, a little bit above the green 'floor', and in front of the user view.
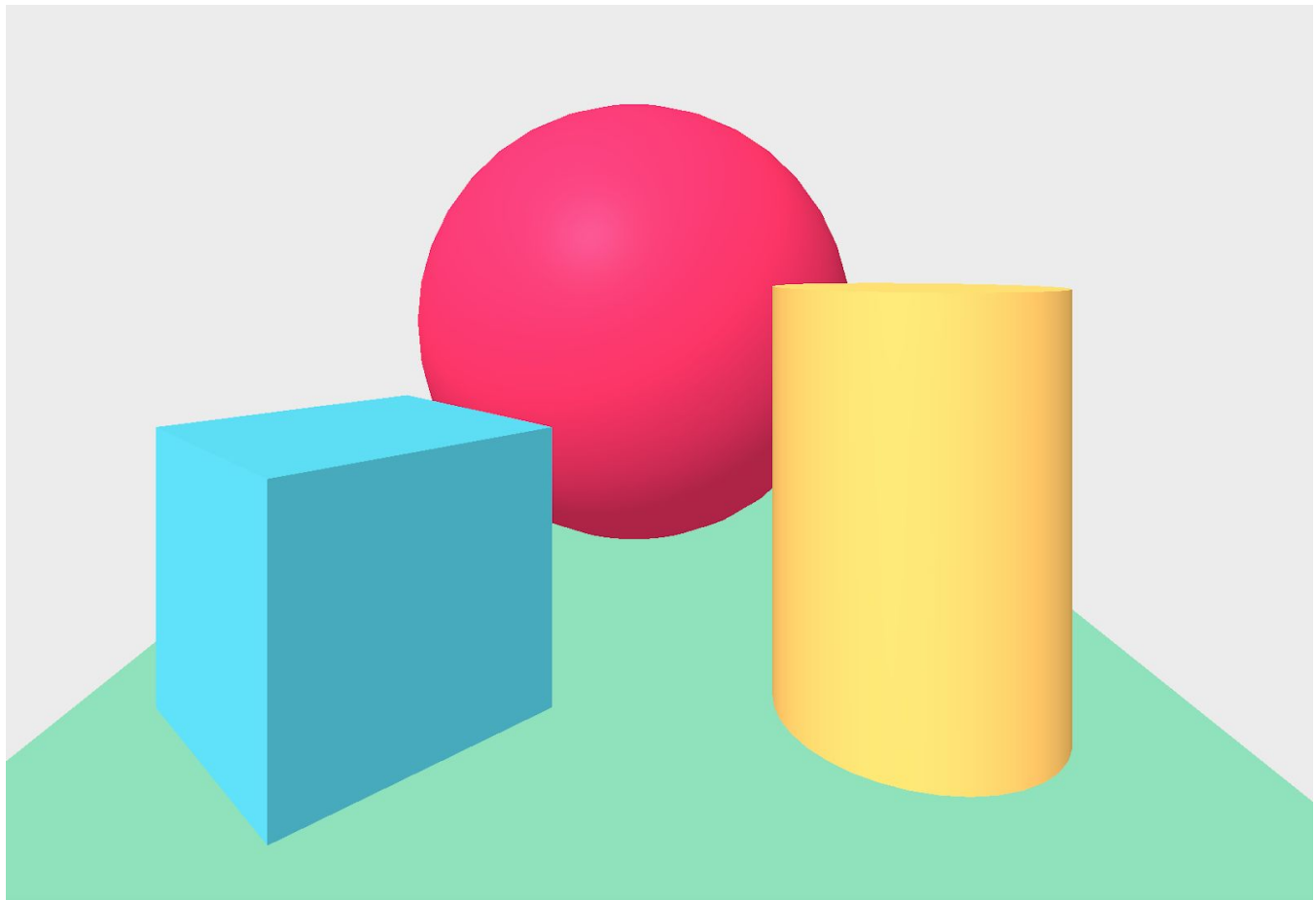
In this new view below the red sphere's z-coordinate value has been changed from -5 to -10, thus illustrating what happens when you change the *z*-axis value

We can move the blue box to the right by changing its x-coordinate from -1 to 1.
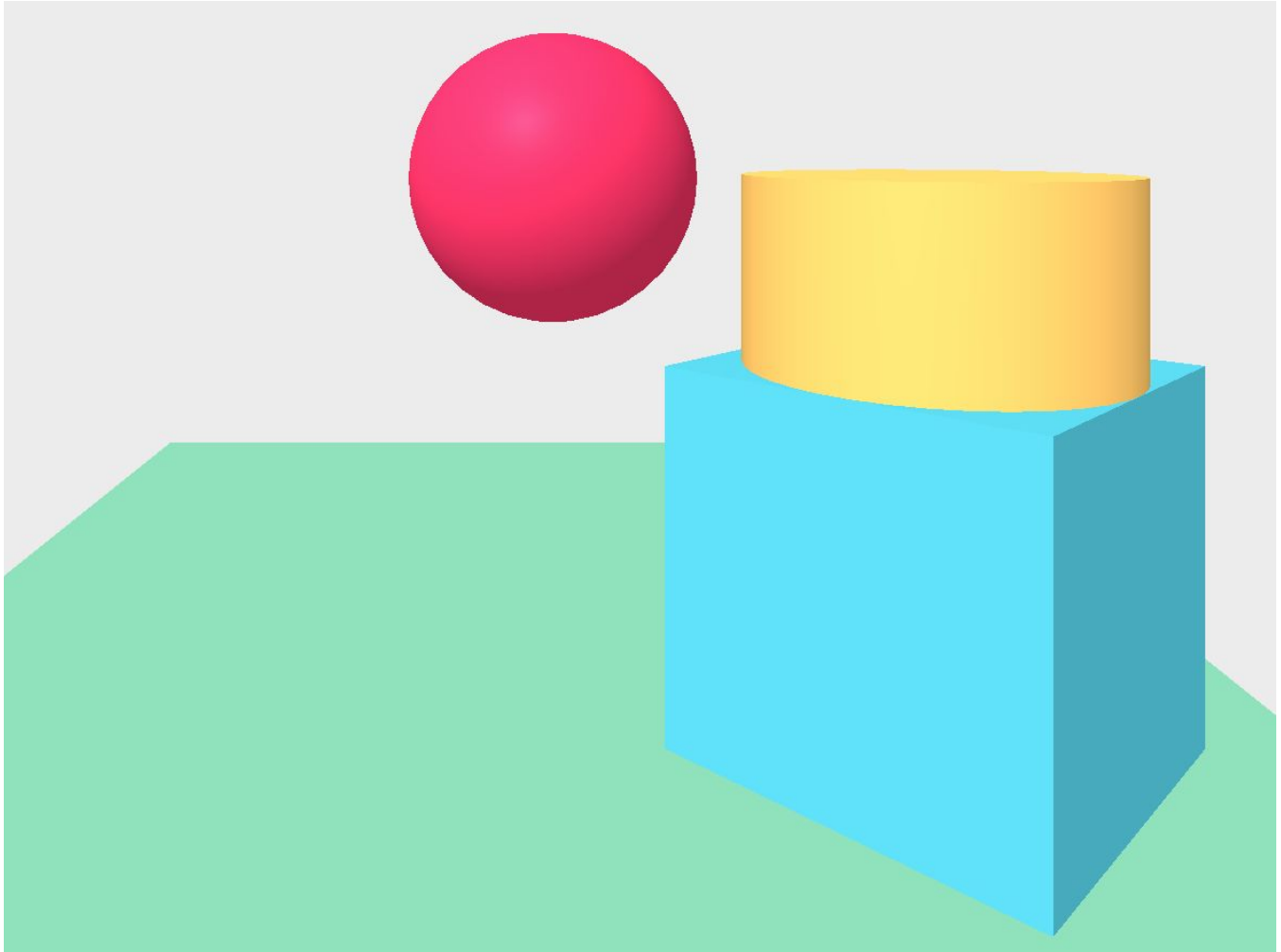
Not only did the blue box move to the right, it also partially covered the yellow cylinder!

You can find out more about positioning on the A-Frame School webpage

Step 3: Explaining the *index.html*

Look at the new Glitch web page you created in Step 1b and click on the index.html file.

```
1    <!DOCTYPE html>
2    <html>
3      <head>
4        <title>Hello, WebVR! - A-Frame Now is the time</title>
5        <meta name="description" content="Hello, WebVR! - A-Frame">
6        <script src="https://aframe.io/releases/0.5.0/aframe.min.js"></script>
7      </head>
8      <body>
9        <a-scene>
10         <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9"></a-box>
11         <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"></a-sphere>
12         <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#FFC65D"></a-cylinder>
13         <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" color="#7BC8A4"></a-plane>
14         <a-sky color="#ECECEC"></a-sky>
15       </a-scene>
16     </body>
17   </html>
```

`<head>` section

The `<head>` section defines some meta information about the web page like the title and the type of content. The `<script>` tag loads the javascript code that enables the A-Frame engine. All of your A-Frame programs must include this `<script>` tag

`<body>` section

The `<body>` section is where the game assets are defined and placed on the webpage. Notice that the tags are arranged in a hierarchical way as indicated by the indenting. This means that `<a-scene>` is a child of `<body>`, and `<a-box>`, `<a-sphere>`, `<a-cylinder>`, `<a-plane>`, and `<a-sky>` are all children of `<a-scene>`. Also notice that each tag is closed by the same tag name preceeded with a "/", i.e. `<a-box>` is closed with `</a-box>`.

The `<a-scene>` tag defines the scene; assets that should be displayed in the scene are included as children of the `<a-scene>` tag.

Step 4: A-Frame School for positioning and rotation

The box, sphere, and cylinder shapes are created with the respective HTML tags. All of the shapes have attributes that define position, and color attributes. The box has a rotation attribute and the sphere and cylinder have a radius attribute. These attributes control the look of the objects. The position and rotation attributes reflect the *x*-, *y*-, *z*-axis as we saw in the last section. Use the `index.html` page during this tutorial

Now open https://aframe.io/aframe-school/#/4 and read the information on this page (the page number is in the bottom right corner)

Click the down arrow to go to the next page (5.2). This page will show you how to position objects in the scene. Be sure to click the links in the tutorial to learn about the various aspects of positioning assets.   Click the down arrow to go on to the page 5.3 when you're finished.

Page 5.3 shows you how the rotation component works. Click next when you're finished.

Page 5.4 will show you how to add primitive shapes of your own.