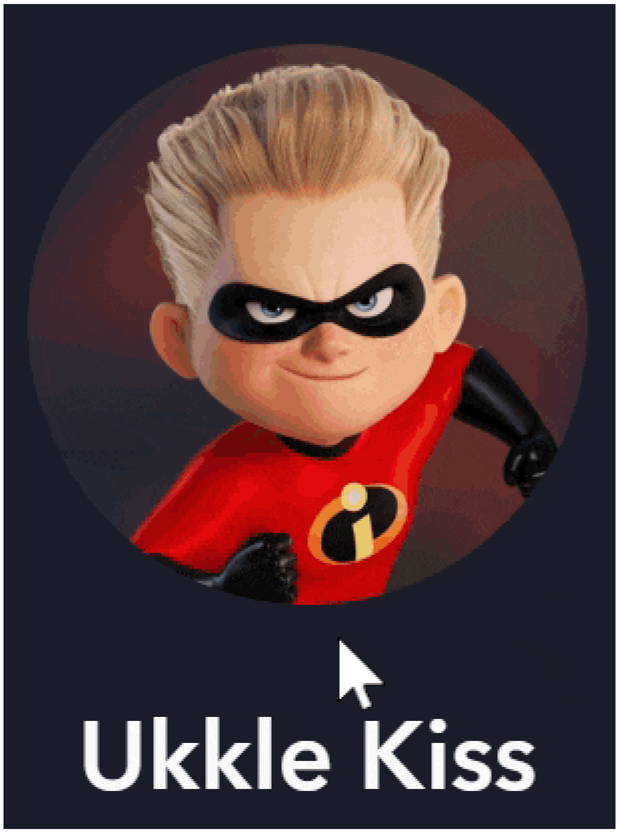
8. Animating User Selection



Let's break it down...

UserSelection.js contains a simple ***img*** & ***span*** setup wrapped up in a ***div*** tag. The primary div is just a wrapper that put the elements in a vertical flex box (see CSS line 1) these elements are wrapped in a map to draw for each user.

*Note there is an additional empty div inside, I would usually frown upon this but the div is required to style the image correctly.

The animation that we care about here is all in CSS and requires us to:

- Shape the image into a circle
- Animating the circle on border colour, scale and shadow.

Turning a square into a circle:

This is a fairly easy technique that often gets overlooked and replaced with actually creating a circular image. There are a number of ways to achieve this purely in CSS and if we were looking at more complicated shapes CSS masking would be perfect. But for us it's even easer, we simply reduce the ***border-radius*** of the ***div*** by 50% (CSS line 12). This creates a rounded corner, on each corner, which collectively forms a circle. Simple.

Animating the values:

UserSelection.js

```
{users.map((item) => (
  <div key={item.displayName}
className={styles.avatar}
onClick={() => signIn(item)}>
    <div>
      <img src={item.avatar} />
    </div>
    <span className={styles.
name}>{item.displayName} </span>
  </div>
))}
```

We are going to use the CSS transition attribute here (discussed in more detail in the Menu breakdown) and simply transition from our defaults to the new values. You'll notice that unlike the menu animation we don't define the starting values in the div, in this case we want to animate from nothing, to our goal so we can leave the starting values to their default "off" state.

Instead of ***scaleX*** we use the ***scale*** attribute which scales in all directions, scale is 1 by default so we simply bump it up a little, just enough to give our users feedback. The ***border*** was empty before, now we are setting a ***5px, solid white border***. Easy.

Finally the ***box-shadow*** attribute, easy to forget about but it adds a nice bit of detail as the box grows and the shadow beneath expands.

UserSelection.css

```
.avatar{
    display: flex;
    flex-direction: column;
    cursor: pointer;
    align-items: center;
    justify-content: center;
    position: relative;
    width: 20%; /*1/5 per item equals 5 per line*/
    margin-bottom: 20px;
}
.avatar > div {
    border-radius: 50%;
    height: 150px;
    width: 150px;
    z-index: 1;
    overflow: hidden;
    transition: all 250ms cubic-bezier(0.25, 0.46, 0.45, 0.94) 0s;
}
.avatar > span {
    margin-top: 5px;
}
.avatar > div > img {
   width: 100%;
   height: 100%;
}
.avatar > div:hover {
    box-shadow: 0px 0px 20px 1px rgba(0,0,0,0.75);
    transform: scale(1.05);
    border: 5px solid rgb(255, 255, 255);
  }
```