Metamorphosis

Site Interaction

Option 3; Process Journal

Bug or Man?

There is much cause for much debate about the nature of Gregor Samsa's Metamorphosis.

What did he exactly change into?

The original German text describes him as:
"ungeheures Ungeziefer," or "monstrous vermin" .

This interactive option asks the user to make choices about what Gregor might be. Man, or Bug?

The user choice then turns on a layer that will show a Bug body part or a Man body part based on the selection made.

The Images:

I collected a series of images from online, cut the background features away using Photoshop and created a collage in Adobe Illustrator using the resulting .png files.

By creating one image and layering everything together I could export a series of user choices and have a transparent background. The page formatting was set to A4, and each file could be aligned using the same CSS properties.
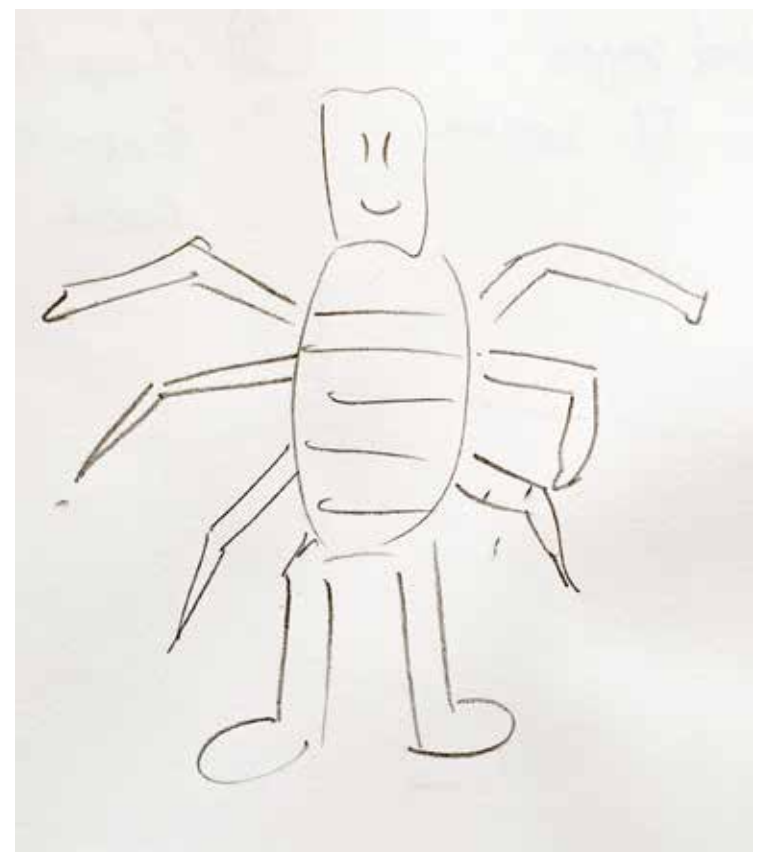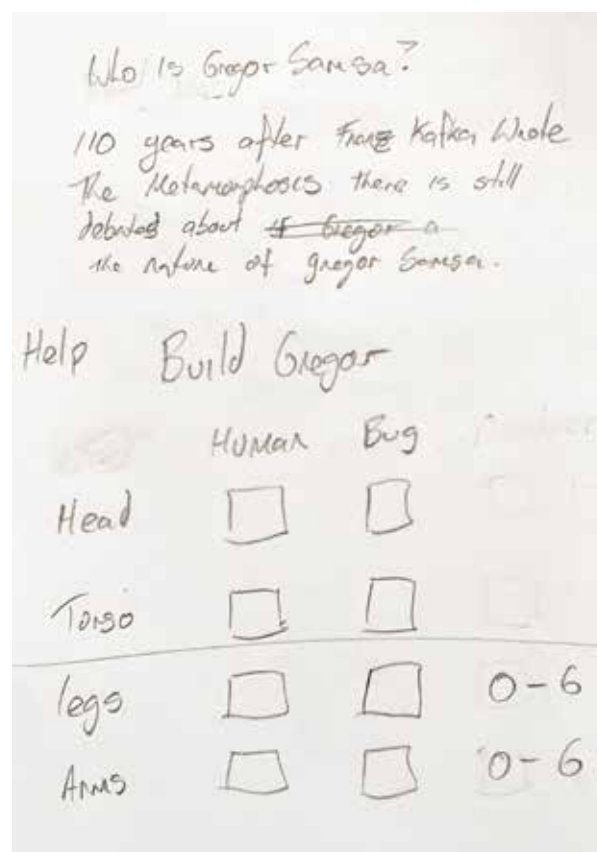
I was particular in the layer positions so that it looked appealing, although I had to change the layers when they were placed in the HTML document.

By using Illustrator I could make an image that worked when all the pieces were put together.
Before exporting I grouped all the legs and the arms to make it a bit easier to manage the CSS and JavaScript actions.

Site Interaction

Concept Ideation:

Ideation Sketches.

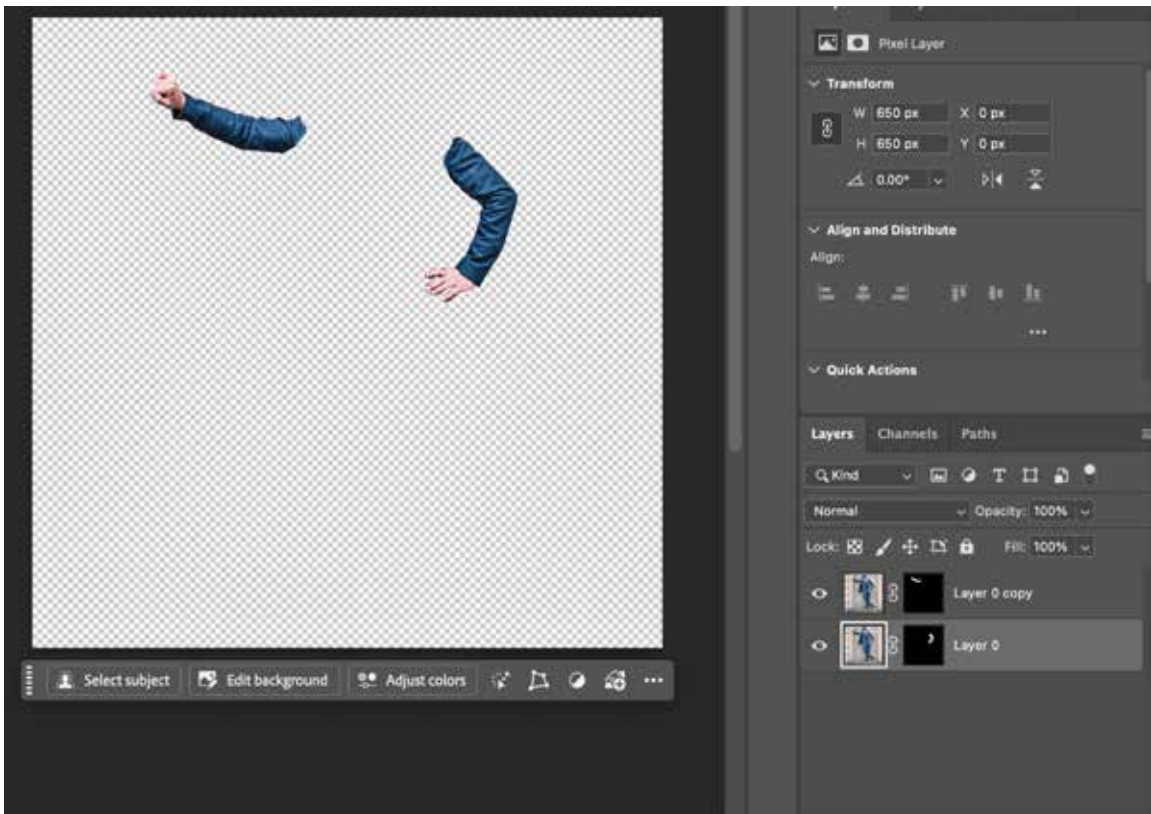When I started this concept I had a loose idea of what I wanted to achieve, but not exactly.
It was difficult to envision what would be needed to make this concept actually work.

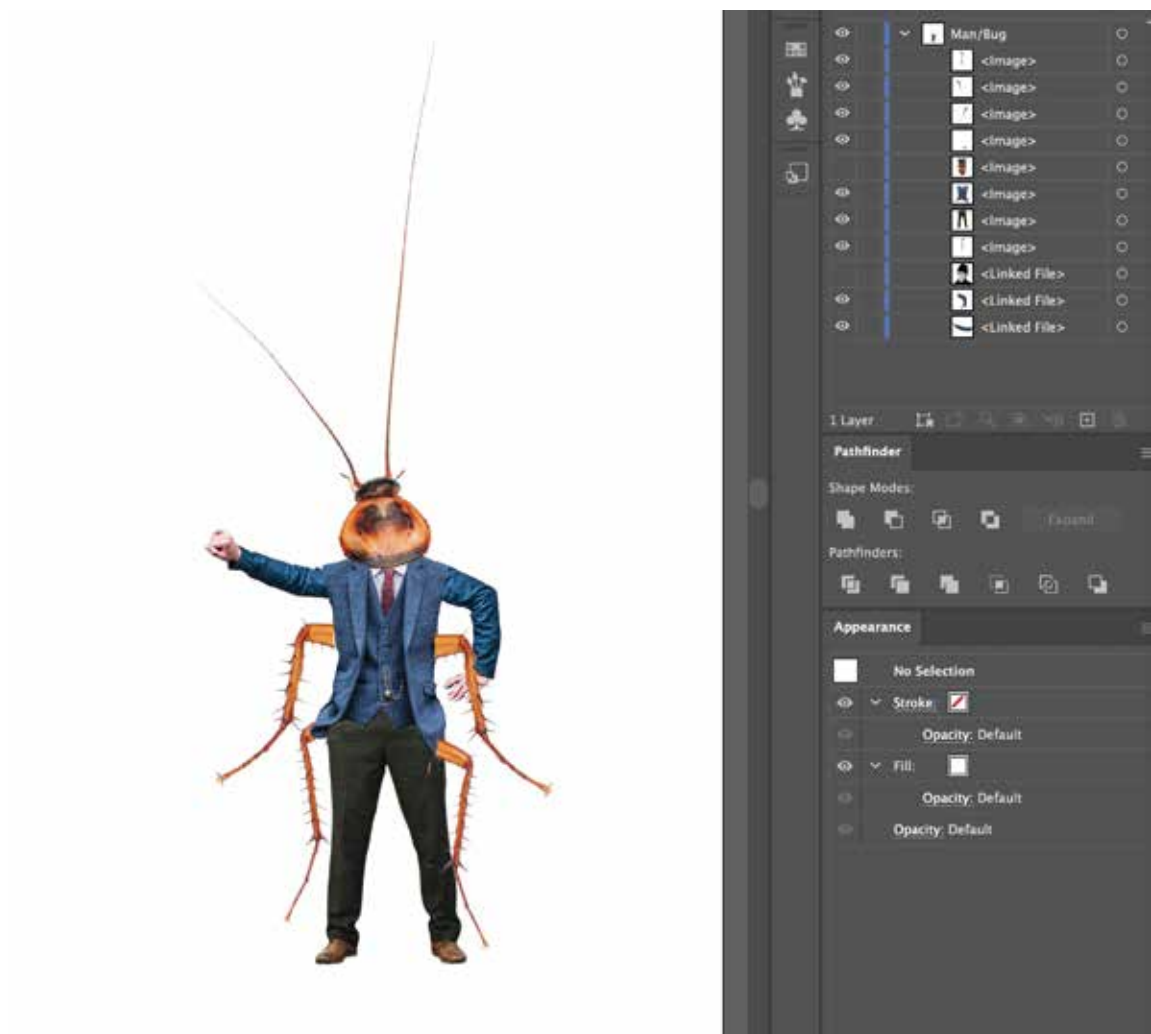I had questions to answer, and I could not do it on the computer.

- Which parameters would be animated?
- What could be manipulated?
- How to make the changes?
- How would the user interact with it?
- What if there was too many 'on' states?
- How many choices were realistic?

Site Interaction

Working Screenshots:

Photoshop View: Creating Layers.

Illustrator View: Assembling the Layers.

Site Interaction

Putting it Together: HTML.

I created a new page for the interactive work, and linked it to the CSS Style Sheet as well as the Google Fonts.
Deciding on the exact wording to get the user to make a selection took a bit of iteration. There needed to be balance on the page and not too much text.
I defined each image as a class and set a base width of 25%.

Visual Studio View: Importing and defining image properties using HTML.

Creating the input fields for the radio button.

The naming convention got a little bit confusing when I was trying to animate because I reused the class name in the radio button id.
I needed to know what I was working on, but also needed to figure out if I was manipulating the class or the id selector.

Site Interaction

Putting it Together: CSS.

The style sheet was linked to my Animation page and I used the classes to manipulate the formatting and style the type so that it was the same as the other pages.

```css
.animation label {
  font-family: "Commissioner", sans-serif;
  font-size: 0.8em;
  color: #195026;
}
.man-head, .man-body, .man-limbs, .bug-head, .bug-body, .b
  margin-left: 10%;
  position: absolute;
  opacity: 0;
  transition: opacity 0.5s;
}
.show {
  opacity: 1;
}
.animation button {
  background-color: rgba(25, 80, 38, 0.1);
  border: 0.18em solid #195026; /*Computed font-size of
  color: #195026;
  font-size: 1em;
  margin-top: 0.8em;
```

CSS Formatting and Animated features.

I chose to animate the opacity property, which can be changed from 0 (visible) to 1 (invisible). This is a common animation variable used in programs like After Effects and Premiere Pro.
A bonus of manipulating this property is that the duration can be managed in seconds or milliseconds.

Site Interaction

Making it Work: JavaScript.

I created a new page for the interactive work, and linked it to the CSS Style Sheet as well as the Google Fonts.

```
</form>
<!-- JavaScript animates Gregor -->
<button onclick="see_gregor()">
    <h2>See Gregor</h2>
</button>
<script>
    function see_gregor() {
        // Head
        if (document.getElementById("bug-head").checked) {
            document.querySelector('.bug-head').classList.add('show');
            document.querySelector('.man-head').classList.remove('show');
        }
        else if (document.getElementById("man-head").checked) {
            document.querySelector('.man-head').classList.add('show');
            document.querySelector('.bug-head').classList.remove('show');
        }

        // Body
        if (document.getElementById("bug-body").checked) {
            document.querySelector('.bug-body').classList.add('show');
            document.querySelector('.man-body').classList.remove('show');
        }
        else if (document.getElementById("man-body").checked) {
            document.querySelector('.man-body').classList.add('show');
            document.querySelector('.bug-body').classList.remove('show');
        }
```

JavaScript Animation.

The JavaScript animates the CSS opacity of the show class selector. I used query selector to manipulate class and the actions add or remove.
A series of "if" and "else if" statements allocate each of the three animated options to be either off or on.