

# Entrada of the Mage

An interactive web experience

By Nat Deis

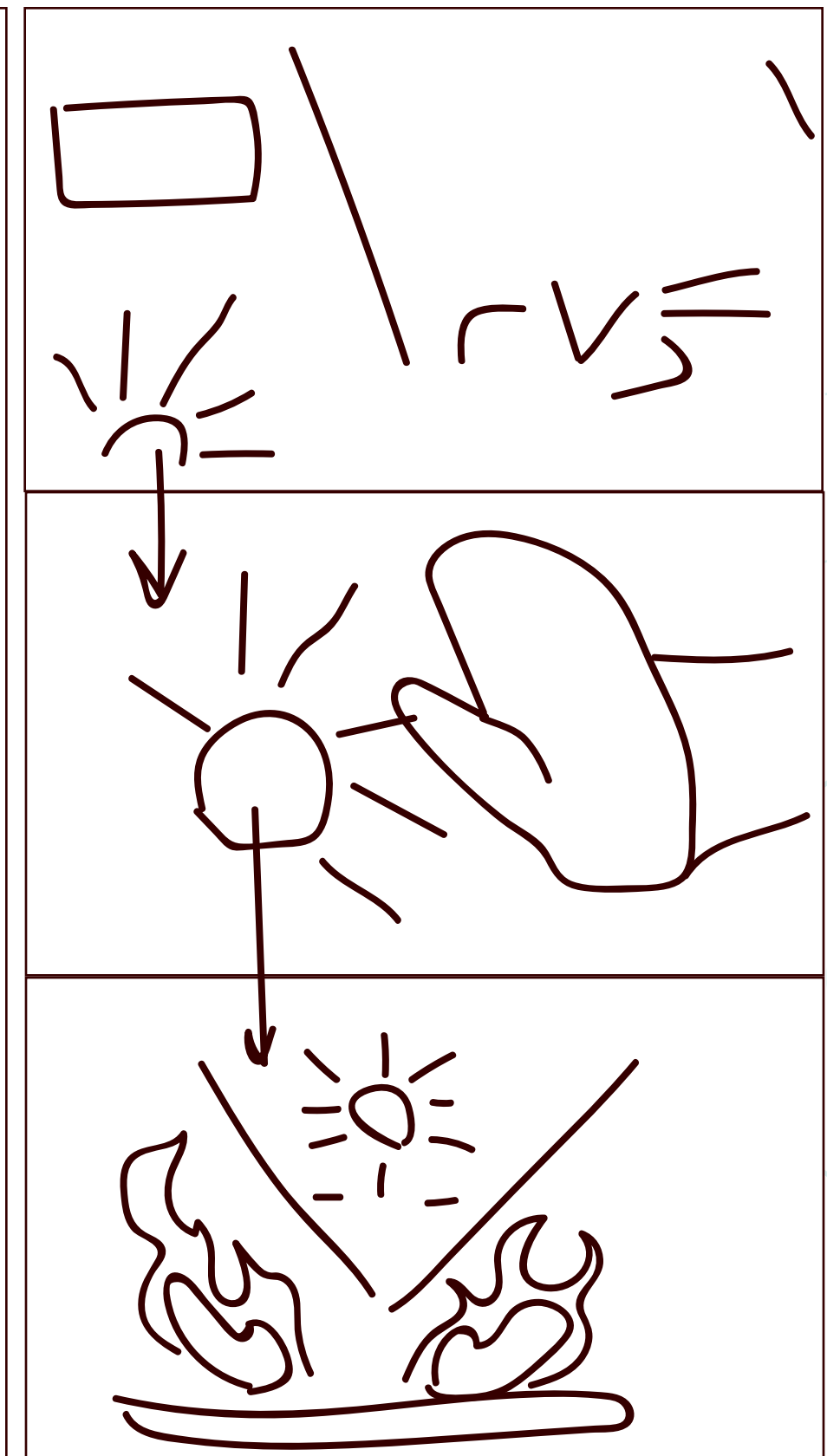
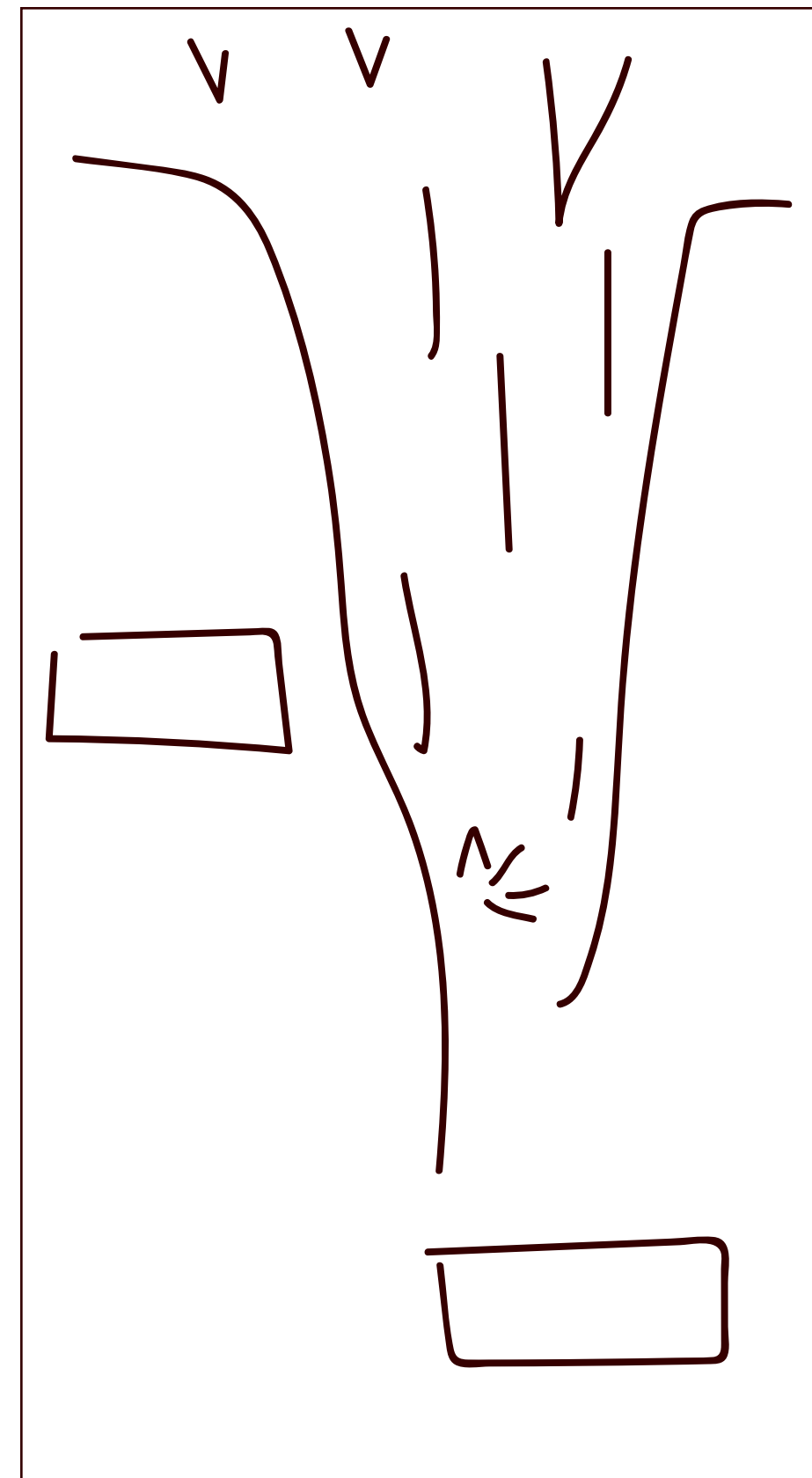
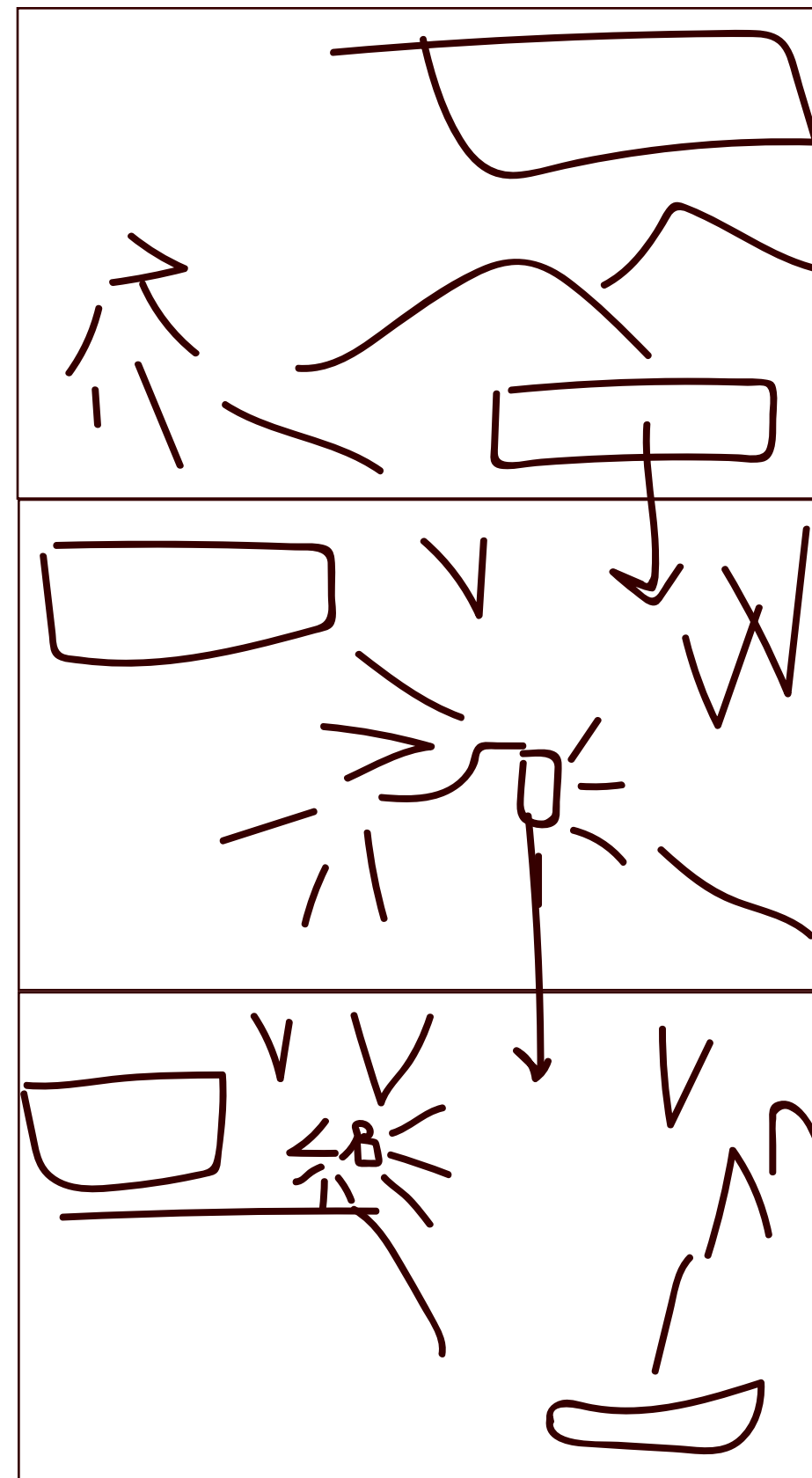
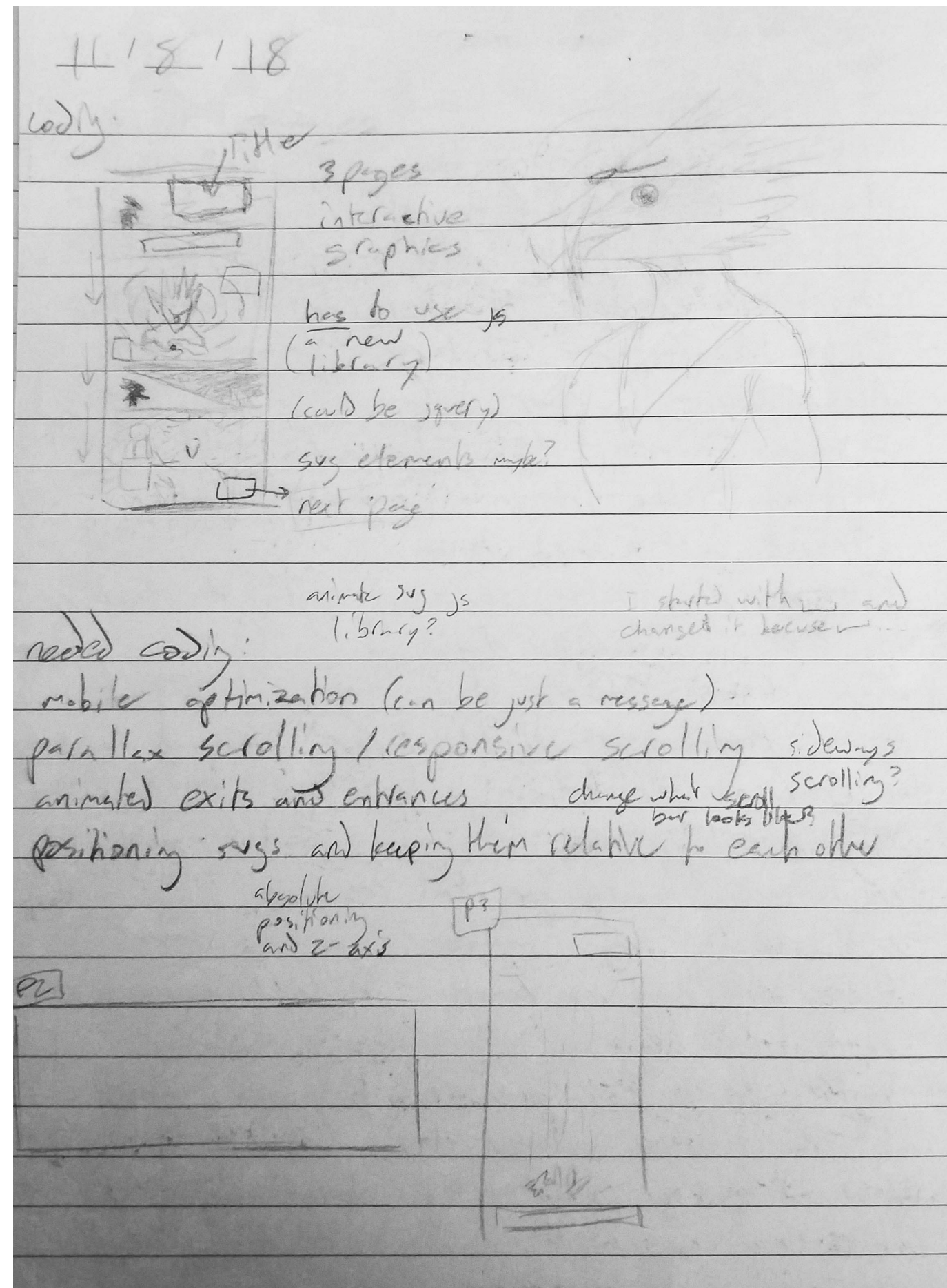
# Scope

This project is going to be an interactive narrative piece that will follow a simple story line. It will be told through the user manipulating a few digital illustrations.

It is intended simply to be an image-based interactive story viewable only on desktop. I want to focus much more on the aesthetics and narrative and make sure all of the interactions are perfect for one medium.

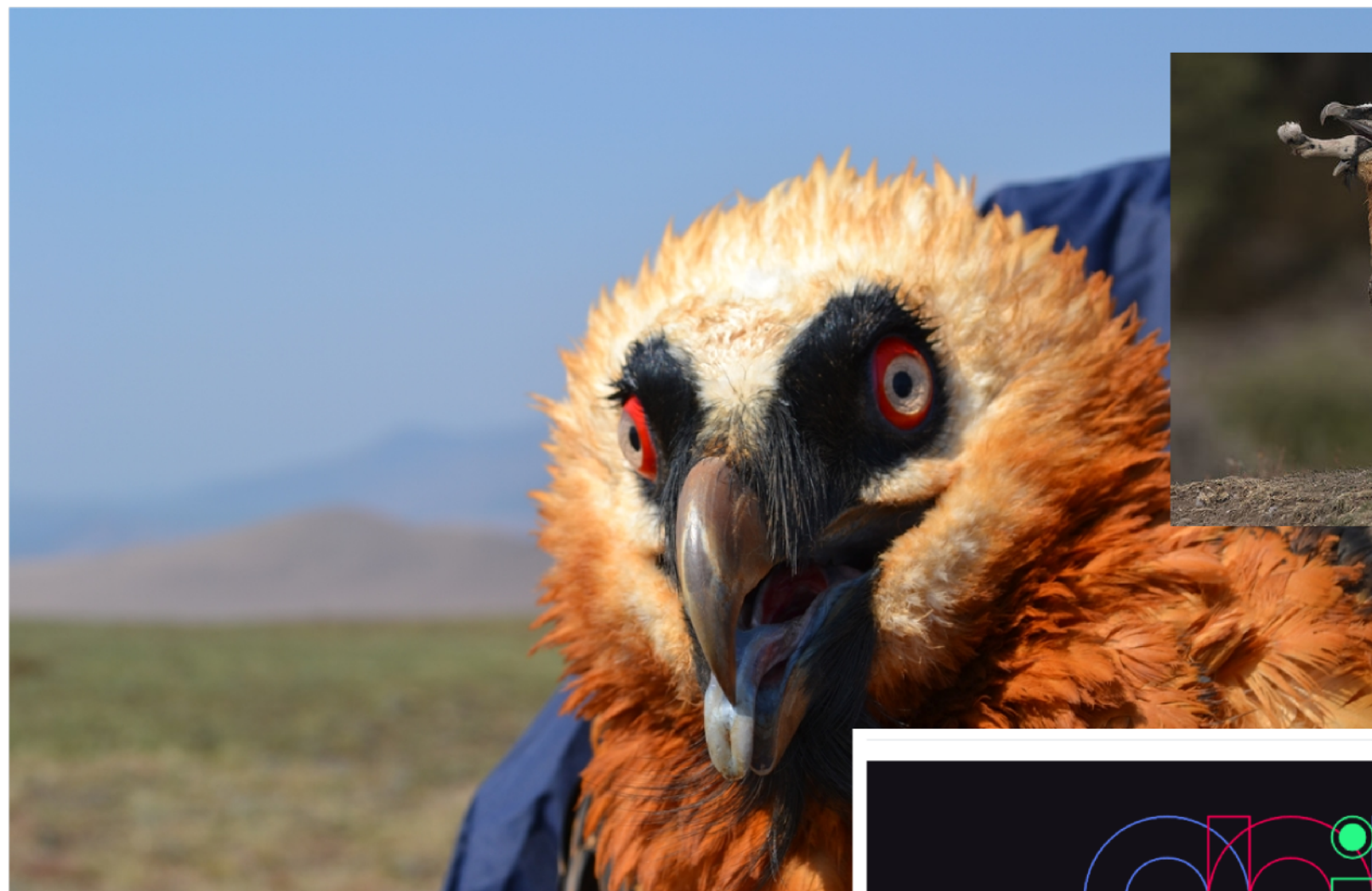
I will need to find and learn a proper java script library that will allow me to work with animating svg images. Then I will need to see how I can use that while maintaining proper image linking and placement.

# Sketches





# Research

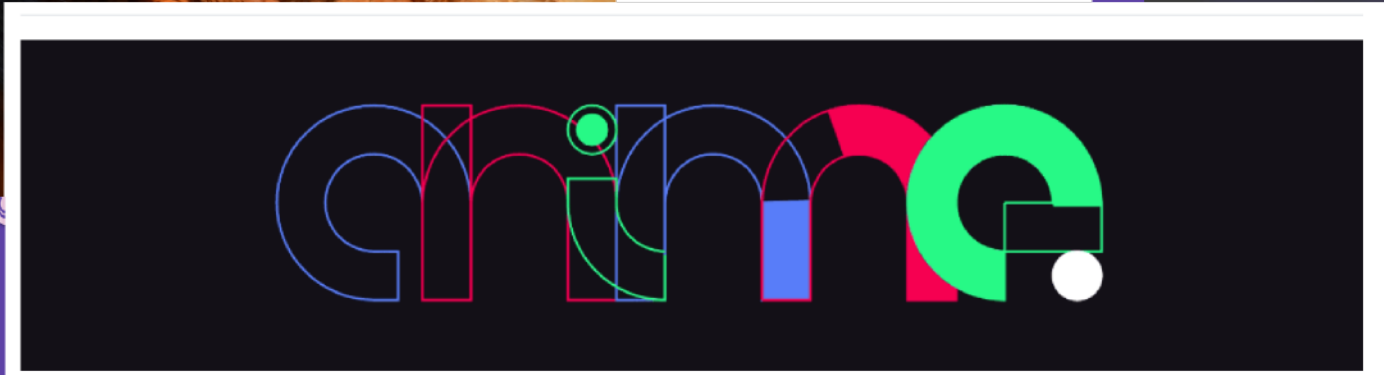


Scrolling is fun 🙌

```
html
class="jumbotron" style="background:
url('https://unsplash.it/1920/800?image=950')
no-repeat center;
background-size: cover;"
data-paroller-factor="0.5"
data-paroller-factor-xs="0.2"

js
$('.jumbotron').paroller();
```

```
js
$('.mega').paroller();
oller-
oreground"
```



Anime (/ˈæn.ə.meɪ/) is a lightweight JavaScript animation library. It works with any CSS Properties, individual CSS transforms, SVG or any DOM attributes, and JavaScript Objects.

⚠️ Migrating from v1.x? Make sure to read the [changelog](#) ⚠️

### Main features

- **Keyframes:** Chain multiple animation properties.
- **Timeline:** Synchronize multiple instances together.
- **Playback controls:** Play, pause, restart, seek animations or timelines.
- **CSS transforms:** Animate CSS transforms individually.
- **Function based values:** Multiple animated targets can have individual value.
- **SVG Animations:** Motion path, line drawing and morphing animations.
- **Easing functions:** Use the built in functions or create your own Cubic Bézier curve easing.

### Demos and examples

- [CodePen demos and examples](#)
- [julianqarnier.com](#)

To enable parallax scrolling effect you can use **data-paroller\*** attributes on selected elements or **set values** via jQuery. **\*factor** sets speed and distance of element's parallax effect on scroll.

📌 Since **factor** is multiplier it **must be set for paroller.js to have parallax effect**

```
<div class="paroller">Awesome element with parallax effect</div>

// initialize paroller.js and set attributes for selected elements
$("*paroller, [data-paroller-factor]*").paroller({
  factor: 0.3, // multiplier for scrolling speed and offset
  factorXs: 0.1, // multiplier for scrolling speed and offset
  type: 'foreground', // background, foreground
  direction: 'horizontal' // vertical, horizontal
});
```

```
<div data-paroller-factor="0.1"
  data-paroller-factor-xs="0.1"
  data-paroller-type="foreground"
  data-paroller-direction="vertical">
  Awesome element with parallax effect
</div>

// initialize paroller.js
$("[data-paroller-factor]").paroller();
```



### TARGETS

CSS SELECTOR

DOM NODE

NODE LIST

JAVASCRIPT OBJECT

```
{ "charged": "0%" }
```

### JAVASCRIPT

```
var cssSelector = anime({
  targets: '#cssSelector .el',
  translateX: 250
});
```

### HTML

```
<div id="cssSelector">
  <div class="line">
    <div class="square el"></div>
  </div>
</div>
```



# Initial Images

## Entrada of the Mage



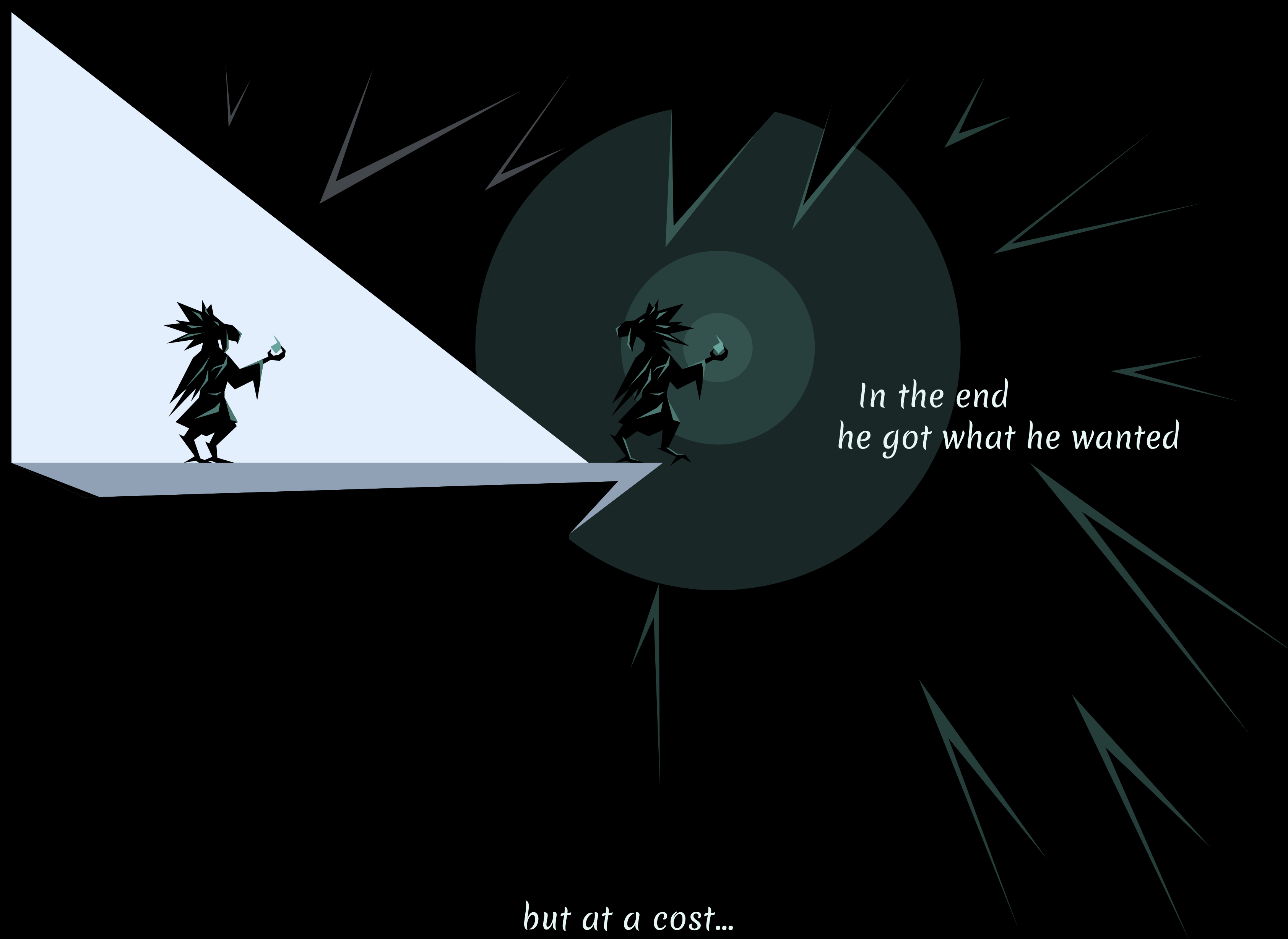
Begin



There was once a young mage  
told of a greater power by his master

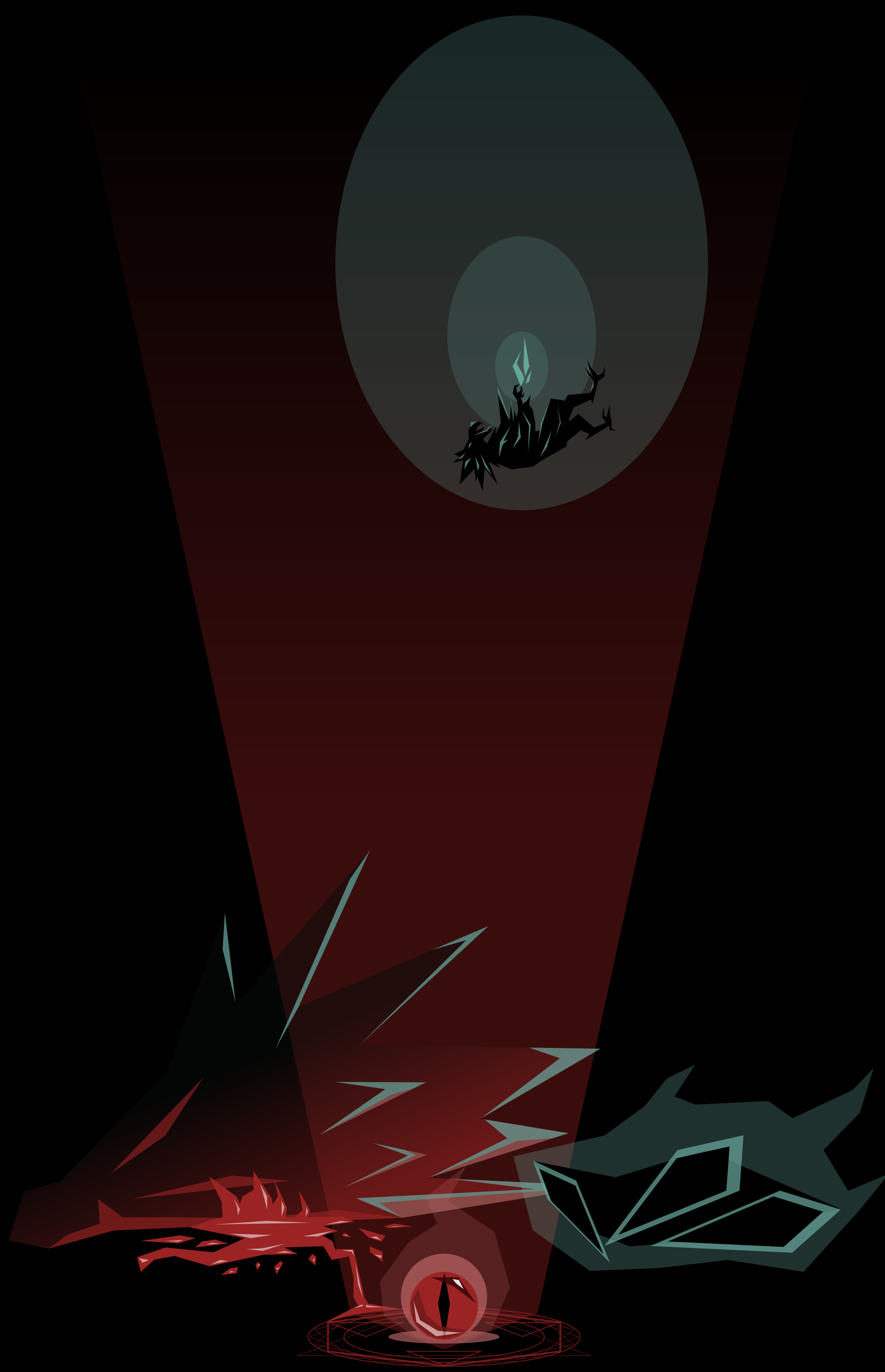
His heart burned with avarice...

Shall we send him on his way?



In the end  
he got what he wanted

but at a cost...



a blood price paid for power gained

# Fulfilling the Vision

There was a lot that I could not accomplish that I wanted to, but there was also a lot that I learned and implemented that I did not expect to before. I ended up using JQuery, Anime, and Paroller to create all the animations on the site.

The placement of the svg images was a lot more difficult than I had thought. I had to neglect the use of a flexbox or grid because they simply did not work with the placement of the images in a way that I was satisfied. The result is a bit shaky in terms of how all the graphics scale together as the window size is changed, but it was the best result I could get. I am still rather satisfied with the outcome despite its shortcomings.

The animations turned out a lot better than I thought, though. I did not know the full extent of JQuery or Anime before using them, but this project let me experiment and find amazing new things that I could do with them. I am really happy with the smooth animations and how they all look working together. Being able to smoothly animate opacity and position really helped make this project look great. I am very pleased with how it all came together at the end. I definitely had to look up a lot of resources to help figure out all of the parts of it, but in the end it was well worth the effort.