Grey paper – Elazar Kagan – website project 2

1) HTML, CSS, JS

```
<div class="card" onclick="goToGame()">
```

```
  cursor: pointer;
```

```
function goToGame() {
    window.location.href = "../html/bug_smash.html";
  }
```

This is the HTML for the div that holds the card the displays the info about the bug smashing game. I added a JavaScript function using onClick to make it that when the user clicks on the box he gets redirected to the page with the game. Also, I used CSS to change the pointer type to signify that the box is clickable.

2) HTML:

```
<a class="btn" id="menuButton" href="../html/main.html">Home</a>
      <button class="btn" id="helpButton">Help</button>
      <div class='help' id="help">
        <table>
          <tr>
            <th></th>
            <th>How to play</th>
          </tr>
          <tr>
            <td>Goal of the game</td>
            <td>Smash bugs, by clicking on them</td>
          </tr>
          <tr>
            <td>How to win</td>
            <td>You can't, this game goes on forever</td>
          </tr>
          <tr>
            <td>What!?</td>
            <td>Yeah, every bug you smash will add double</td>
          </tr>
        </table>
      </div>
```

I added to the html of the bug smashing game 2 buttons. The first is a button that lets the user go back to the main page, using a link styled as a button. The second is a button that when clicked reveals a table that explains how to play the game, the JavaScript for this is in #3.

3) JS, CSS:

```
const helpButton = document.getElementById('helpButton')
const showHelp = document.getElementById('help')

helpButton.addEventListener('click', () => showHelp.classList.toggle('showHelp'))
```

```
.help {
  display: none;
  max-width: 40%;
  z-index: 10;
  position: absolute;
  top: 80px;
  left: 20px;
}

.showHelp {
  display: block;
  background-color: #fff;
  color: #516dff;
  padding: 15px 20px;
}
```

This is the JavaScript and CSS for the help button in the bug smashing game, that when clicked reveals a table explaining how to play the game. The JavaScript gets the button and the table from the html, then makes it that when the button is clicked it adds or removes the .showHelp class. The CSS has it that by default the table is hidden (with display: none) but when the table has the class of .showHelp the display changes to now visible thus revealing the table. When the button is clicked again the .showHelp class is removed making the table disappear as it does when it is in its default status.

4) CSS:

```
.blink-bg{
  animation: blinkingBackground 1s infinite, fadingBackground 1s;
  animation-delay: 3.5s, 7s;
}

@keyframes blinkingBackground{
  0%    { background-color: #10c018;}
  25%   { background-color: #1056c0;}
```

```
  50%   { background-color: #ef0a1a;}
  75%   { background-color: #254878;}
  100%  { background-color: #d504b2;}
}

@keyframes fadingBackground{
  0%    { background-color: rgba(223, 24, 190, 1);}
  25%   { background-color: rgba(223, 24, 190, 0.75);}
  50%   { background-color: rgba(223, 24, 190, 0.50);}
  75%   { background-color: rgba(223, 24, 190, 0.25);}
  100%  { background-color: rgba(223, 24, 190, 0);}
}
```

This is some of the cool CSS that adds some animation to the site. I have it that right after the countdown there's an exciting "lets go!" message and to add to that I have a flashing background with many bright colors. I did this using the animation declaration and created 2 animations (separated by a comma) and defined the animations a few lines later with @keyframes. The second animation has the last background fading out to match the text (I used background colors with the opacity less and less to accomplish this)

5) CSS:

```
h1 {
  letter-spacing: 3px;
  animation: startAnimation 3s ease-in-out;
  z-index: 20;
  border-radius: 25px;
  padding: 20px;
}

@keyframes startAnimation{
  0%    { background-color: rgba(50, 24, 190, 1);
          font-size: 180px;}
  25%   { background-color: rgba(50, 24, 190, 1);
          font-size: 190px;}
  50%   { background-color: rgba(50, 24, 190, 1);}
  75%   { background-color: rgba(50, 24, 190, 0.25);}
  100%  { background-color: rgba(50, 24, 190, 0);}
}
```

I did another CSS animation that when you enter the main page the main logo text starts very big in the center with a pink background and then, in the next 3 seconds, moves and fades to its correct position.

6) CSS, JS:

```
.slide {
  transform: translateX(13%);
  transition: transform 0.4s ease-in-out;
}
```

```
const open_btn = document.querySelector('.open-btn')
const close_btn = document.querySelector('.close-btn')
const nav = document.querySelectorAll('.nav')
const body = document.querySelector('.card-container')

open_btn.addEventListener('click', () => {
    nav.forEach(nav_el => nav_el.classList.add('visible'))
    body.classList.add('slide')
})

close_btn.addEventListener('click', () => {
    nav.forEach(nav_el => nav_el.classList.remove('visible'))
    setTimeout(() => {
        body.classList.remove('slide');
      }, 500)
})
```

I used a sliding nav that comes out of the left. I have this CSS to make the content in the middle also slide with the nav making the whole page smoothly move instead of having the nav cover the main part of the site which didn't look so good. For the JS I just added my class for the content in the site (.slide) into the code that moved the nav so both movements are in sync.

7) HTML, JS:

```
<input type="number" class="code" placeholder="0" min="0" max="9" required oninput = pageRedirect()>
```

```
function pageRedirect() {
    window.location.href = "../html/start.html";
  }
```

For the verifying part of the site (where you add a code that would be emailed to you) I wanted that right after you finish typing the code it will redirect you to the next part of the website (this is it works on most sites that you don't have to push a button). Since each number has its own box to input I added oninput = pageRedirect() which when something was inputted into the last box (which means the code was fully written out) it calls the function from the JS that redirects to the next stage of the site.

8) JavaScript:

```
setTimeout(()=>{
    window.location.replace("main.html");
    }, 7000);
```

In order to transition from the intro animation to the main part of the site I added a setTimeout function because I know that it takes 7 seconds for the animation to run so after 7 seconds you get redirected to the html of the main part of the site.

9) CSS:

```
.help table, th, td {
  border: 2px solid #516dff;
  border-collapse: collapse;
}

.help th, td {
  padding: 10px;
}

.help th {
  border: none;
  text-align: left;
}
```

Here is the CSS for the table that is inside the bug smashing game, I put it at the end so it can inherit the same style as the game and then added my own style for the rest of the table, to make it fit in with the game as best as possible.

10) CSS:

```
.background {
background: url('../img/security-privacy-image.jpg');
background-size: cover;
position: absolute;
top: -150px;
bottom: -20px;
left: -20px;
right: -20px;
z-index: -1;
filter: blur(20px);
opacity: 0.8;
}
```

For the password strength that is showed based on how clear the background is, I used CSS to add a different background which is a lock and a bunch of code around it which I think was a more appropriate image to get to see by creating a strong password. The rest of the CSS is fitting it on the page to make it look as best as I could, including lowering the opacity a little as it is a background.