Eileen Lee
12/11/2020

Link to Github Repo
Link to Bug Visualizer Website

Part 1

My website is a bug visualizer in which the user can click on and interact with bugs, rocks, and bushes. The user can "collect" bugs by clicking on them and adding their names to a glossary, which updates as they discover more bugs. I want the user to learn about different types of common bugs that can be found in their backyard or in their garden, so I included a couple of fun facts on each bug. (For example, did you know ants can carry something 5000 times their own weight?) My website includes interactive game elements to engage the audience. For example, the user can drag around and rearrange rocks and bushes, click on bugs to pull up a unique animation and blurb, and win a badge for discovering more than half the bugs in the game. Since the website is randomly generated, they can also refresh and play again to discover new insects, which gives the game replay value. My target audience can be an elementary school student, or a young adult, since it's fairly simple and straightforward, but not too difficult, to navigate and play. I think my game will be especially entertaining for people who want to learn about bugs they have seen before in a relaxed and casual way.

Part 2

- Click and drag -- Click on a *bush* (green cube) or *rock* (gray cube). *Drag and drop* to another location. This is especially useful when a bug is hiding under a bush or rock, and you want to click on that bug.
- Hover -- *Hover* over a *bug*. This pops up an orange arrow that points to the specific bug that your mouse is over. This is useful for indicating what bug you are pointing at in case you want to click on it.
- Click -- *Click* on a *bug*. This pulls up the animation, name, and description for the bug.
- Click -- *Click* on the *magnifying glass* in the top right corner. This toggles the menu on and off so that you can also explore the area under the information box.
- Click -- *Click* on the *book* below the magnifying glass. This pulls up the glossary, which includes all of the names of the unique bugs you found. Bug names will be added to your list as you click on and learn about more bugs.
- Click -- *Click* on *"Explore More Projects"* in the very top right corner of the page. This will take you to my portfolio website where I have my other art projects listed.
- Refresh -- If you're getting bored of the current bugs on the screen, *refresh the page* to start the game anew, with a new collection of random bugs and randomly generated rocks and bushes.
- Click -- *Click* on *6 unique bugs*. This will require you to explore the bugs on the page and hopefully find 6 unique bugs. If you do, this will pop up a bug badge below the book icon, along with a description congratulating you for discovering lots of bugs.

Part 3

- P5.js

- I chose p5.js for my external tool because it was optimal for implementing a game that is based heavily on visuals. I have played around with p5.js before to make images and drawings, so I thought that it would be the perfect tool for creating an interactive art game.
- My code was heavily based on p5.js and javascript. I made most of my visuals using the p5.js draw function, and I was easily able to organize an MVP structure for my art assets to have game elements. I also had to do some object-oriented programming for the three objects in my game: bugs, rocks, and bushes.
- P5.js was honestly a life-saver for my website. I think it really helped with my goals of achieving a pleasing and fun visual for my game and also making the game functional and fairly easy to implement. Also, it was fairly easy to make a responsive page with p5.js through a resizeCanvas function.

Part 4

My website went through a few iterations since my first HW 7 mockup. Although the characters and assets are mostly similar, I did make a few color changes to make my game more bright and bouncy, and I also recreated some of the art assets instead of exporting the images themselves. As I added more to my game, I felt confident to add in a couple more features that I thought would make it more fun. For example, I didn't add in the glossary and bug badge until the very end, when I got the idea for it after talking with Professor Carrington for our project show and tell. Besides the visual changes, I also added in a reference to my website, so that people can learn about who made the game (me!) and my other projects.

Part 5

One challenge I had was figuring out how to work with objects in javascript and p5.js. I had to do a lot of research looking at how other people created different interactions, and I also went to office hours to get some help on how to loop through and call certain functions for objects. For example, when I click on a specific bug, how can I loop through my list of bug objects, pinpoint what bug the user clicked on, and call the function only for that bug? By getting help at office hours and also looking at advice online, I was able to get my code to work the way I want it to, and also be more efficient!