

Roboterra Desktop App

Mikaela Reyes
Summer 2016

Table of Contents

- 1 Context & Overview
- 2 Goals
- 3 Empathy & Research
- 4 Wireframes
- 5 Mockups
- 6 Moving Forward

Context & Overview

In the summer of 2016, I interned at a robotics edtech startup in Silicon Valley called Roboterra, which teaches kids how to code through robotics. I was the first full-time UX/UI designer on the team. Eventually, I ended up doing PM work as well: I wrote code and created educational content for the platform.

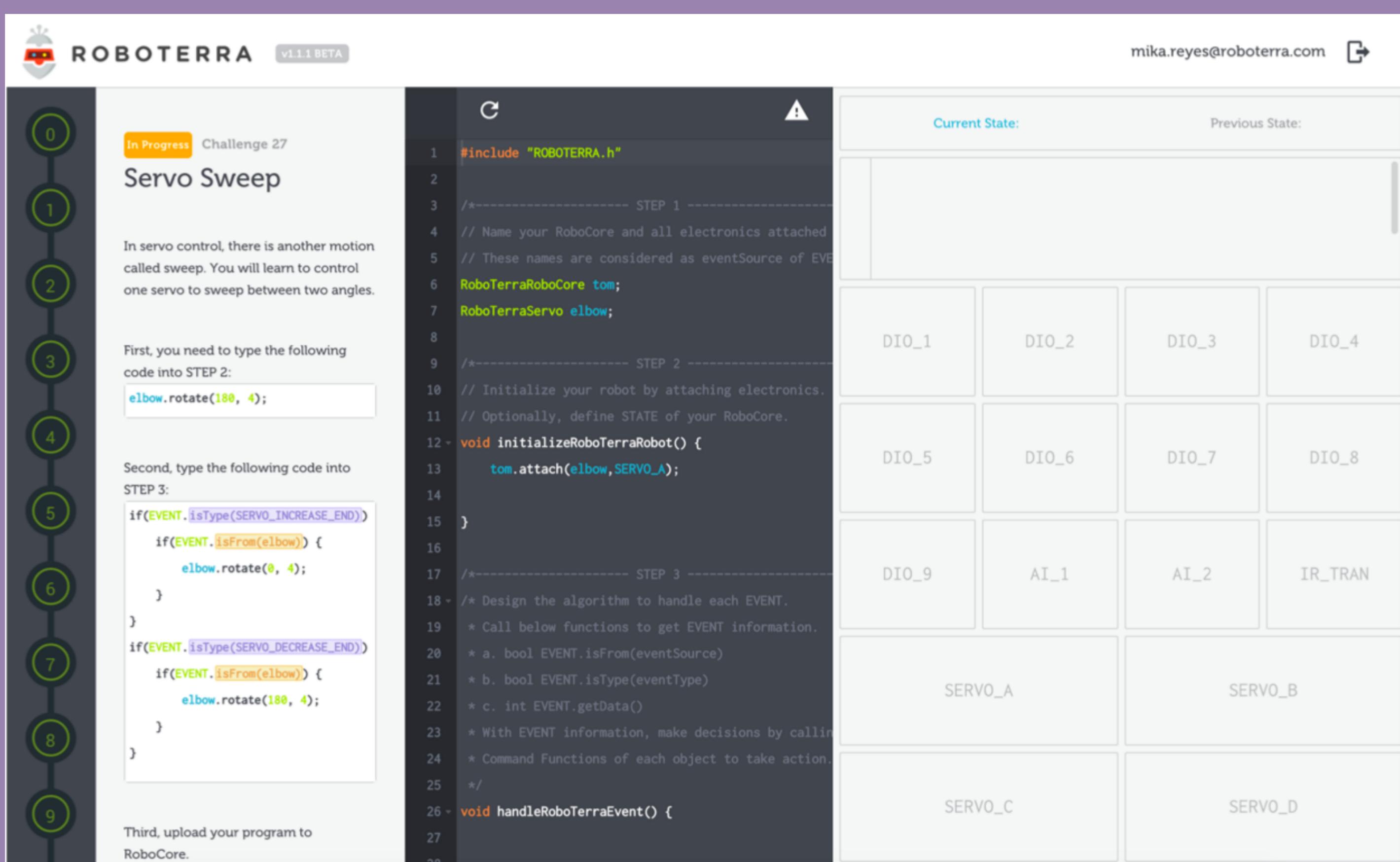
I was tasked to redesign their main product, Castlerock. They had their first version of the product but it was ripe with several bugs. In addition, they were going to change the way students learned on the platform, so an entirely new product, with new ways of managing the content was required.

This is possibly the most difficult design feat I've encountered to this day. I was the sole designer and was only given the first version of the product, a few kids to test on and a big picturel idea of where the company was going. Besides designing the interface, I was brainstorming ways we could move the product and the company forward. Phew! Lots of work for an intern, but work I welcomed with open arms.

Empathy & Research

Method #1: Analyze Version 1.0

I did and redid the first CastleRock version noting any bugs and important UX fixes. I also looked at other people's reviews of the product.



Origin Kit Hardware						
Hardware No.	Hardware Name	No.	Category	Description	Suggested Solution	Location
1	LED	1	Stability	LED is always on a little bit		Challenge 18:
		2	Specs/design/quality control?	In C3 LED turned on for a second, turned off, and on, and then solid on. Not sure of the reason and there is no explanation. (Tried again for C3 with a red LED, didn't happen. Could be the quality issue of Taran's LED)		NA
		3	Green LED	Green LED blinked twice when nothing is programmed.	Beibei Comment: No explanation? Signal initialization?	Challenge 11
2	Button	1	Design	Button has LED which will turn on. It confuses the users that the LED is working instead of the button	Different LED color used for buttons	Challenge 6
		2	Design	Two buttons can't distinguish from each other		All
3	Origin Kit	1	Variety	Compared with similar price level product, the kit is missing: more servos, an enclosed PCB or a remote control,a screen, wheels, gears or axiles.		All
		2	Component Numbers	Pieces are smaller and fewer. Not enough to make a robot with one kit, such as welcome robot		All
		3	Price	Higher than products with more components.		NA
		4	Packaging	Nice looking but impractical since it's difficult to take out a box when all 5 are inserted.		NA
		5	Packaging	Lid of each box is difficult to remove from the base	A small hole for air to move through	NA
		6	Packaging	Build box, once unpacked, is impossible to fit all pieces back inside in a fairly short time. Nobody wants to stack and align exactly as packaged.	two boxes for aluminum parts, where each box has dividers, like sense box./Refer to Vex IQ Super Kit.	NA
Software						
4	Castle Rock	Bug		had to restart Castle Rock to recognize robocore		NA
		Error Message		No instructions on what to do if something goes wrong		NA

Option to enlarge the font

Great things:

1. Storyline relating coding to nature
2. Virtual components responsive to actual physical components
3. Window panes are all on the app
4. Coding platform mimics text editors really well
5. I can't just copy-paste code – an important feature for learning.

General Comments

1. No onboarding system.
2. No free mode.
3. If I don't have the exact, exact, exact code, it doesn't work, even if I completed the challenge checklist. Mapping is very different.
4. Don't auto-next once I complete a challenge. Give me an indication that I complete the challenge, but the option of moving onto the next challenge when I'm ready.
5. Logout button is not intuitive. Fix: change the button to the words "logout."
6. Window panes should not cut out text – I'm sometimes unsure if the text is cut or if it's meant to be that short.
7. Auto-scroll up every time I finish a challenge.
8. Give me clearer indications that I'm on the next level. I don't always pay attention to the number changes to the left.
9. Give a clearer indication of what challenge level I'm in—the "unlocked" button doesn't seem to do that well
10. Find a way to differentiate between hardware vs. software instructions.
11. Label the steps as actual steps or find a way to differentiate the steps from each other (by steps, I don't mean the one in the text editor, but the instructions on the instruction pane.) I missed an important set of instructions before I could complete the challenge, because I thought I was done with the entire challenge. The clutter of text at the bottom, I usually skip, and because there was no barrier, I skipped a lot of the important text at the bottom anyway.
12. Challenge list should be above, so I know what goals I'm completing at the very onset.
13. Give syntax + example, so I know what the variables in the instructions mean.
14. There are lots of copy mistakes that can be fixed.
15. When I exit the program or restart my text editor, I restart to the default "tom" name, not the customized one I made.
16. Define terms, after all, the kids are learning something new. Ex. API.
17. I think the person making the instructions page always started from the default mode, not from the previous block of code – stay consistent. I ended up just defaulting to the template code to go through the challenges.
18. When I recomplete old challenges, I should still get some indication that I completed it, even if it isn't as elaborate as completing a new challenge.
19. I caught a quick block of code in the "Congratulatory" message – sometimes loads too slowly.
20. Why didn't I get a big congratulatory message at the end of the entire tutorial?

Challenge-Specific Comments

1. #7 – Copy mistake: "first entered in STEP 3"
2. #10 – It didn't tell me to erase else if { }. Again goes in line with theory that the person making the instructions always started from the default template configuration.
3. #11 – Copy mistake: "sometimes they **are** fast"
4. #12 – Differentiate between telling me to take out all the code or base it off my old code
5. #15 – In instructions, it tells me to use "button." In the default template, I initialized "button1" instead of "button." This takes me a while longer to figure out that the two are different.
6. #15 – The default for toggle is on, not off. The instructions don't match the functions.
7. #24 – Tell me that what I am declaring with int var is a variable. Someone new to coding will not get that initially.
8. #25 – The default template puts button in DIO_1, not DIO_2, but my button socket has always been in DIO_2. I'm forced to switch the socket, (I can't change the code only because my code has to be exact so I can pass the challenge). Easy fix: change DIO_1 to DIO_2.
9. #25 – What does SERVO_INCREASE_END mean and do?
10. #26 – Servos launch before I'm even able to press the button because the button is on ACTIVATED mode immediately. I only got to move onto the next challenge when I pressed the button *while* the servos were still turning, but that shouldn't be the case. Fix: initialize button so that it is deactivated, not activated at the start.

These were my comments. Here were some big changes:

1. Adding an onboarding system.
2. Users aim to finish a "project" or actually create a robot, instead of just going through the exercises without a bigger goal. Within each project are smaller "challenges."
3. Integration of hardware instructions via a "build" mode and animations.
4. Free mode to allow users to build whatever they want, separate from the learning interface.

Method #2: Usability Testing

I did usability testing with about 7 high school students. These were some samples of my questions and tasks.

User Research Questions

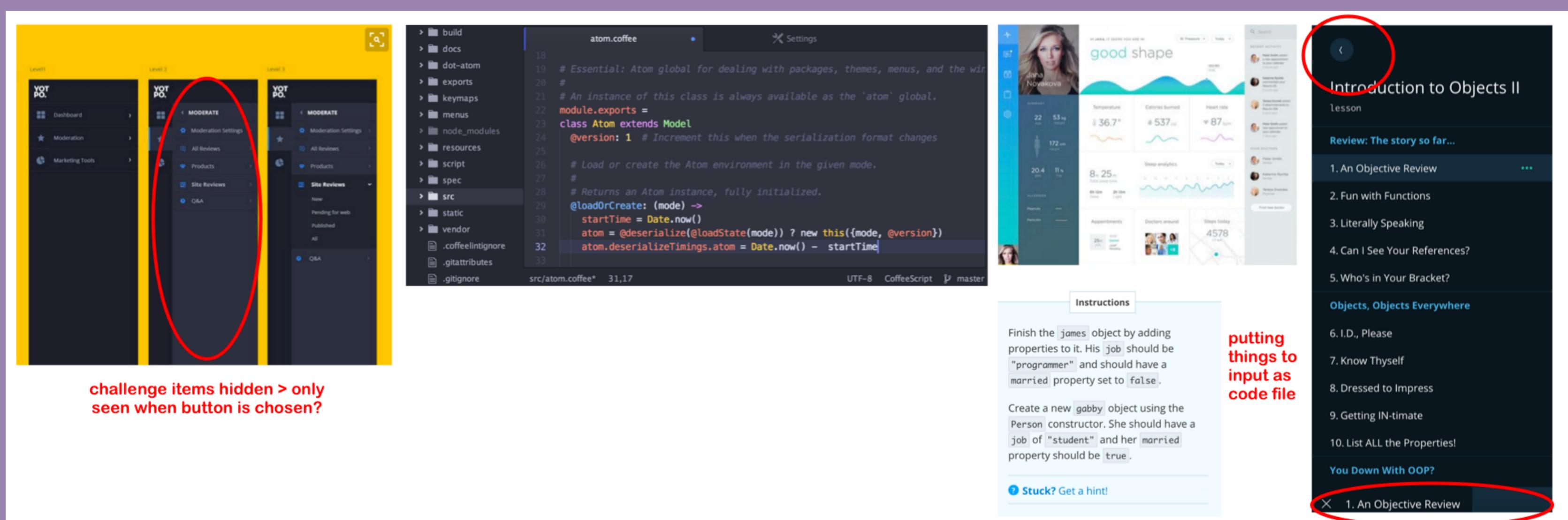
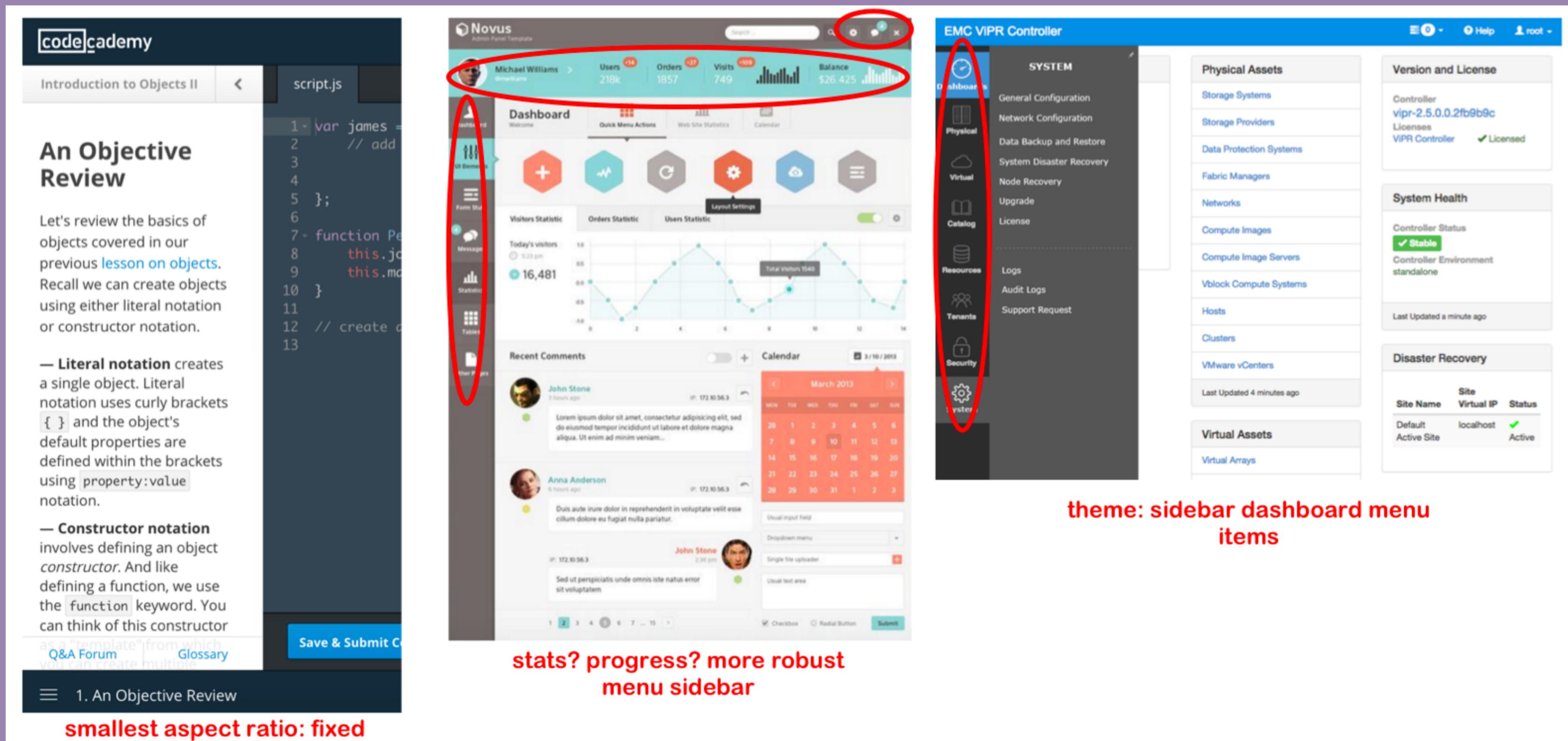
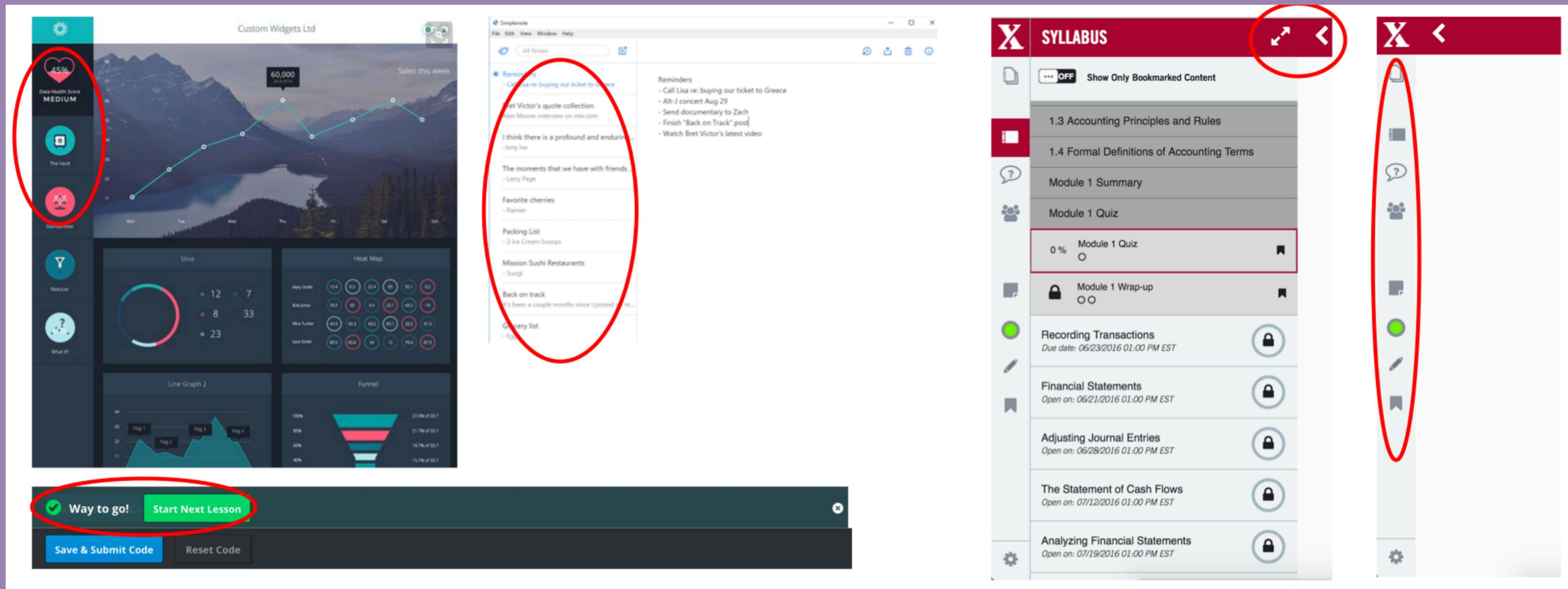
1. What do you do in your free time?
2. How long have you been coding? Is this your first time coding?
3. How did you find out about Roboterra?
4. Why are you interested in robots/Roboterra?
5. What do you want to be doing in this internship program?
6. How do you like to learn best?
7. Tell me about your favorite subject in school. Why do you like it?
8. Tell me about your typical day in school.

Usability Testing Tasks and Follow-Up Questions

1. *Collect demographics (age, background info, etc.)
2. Explore the 5 boxes.
 - a. Can you tell me what you think about these parts?
 - b. How do you feel about making your own robot?
 - c. Have you made your own robot before? Why or why not?
3. Let's open CastleRock. Type Roboterra.com on your browser. Can you download CastleRock on your laptop?
 - a. If you're having trouble downloading, where are you going to look first?
4. Please sign into CastleRock and explore it for a bit. Can you logout of CastleRock and then sign in again?
 - a. What do you think these panels mean?
 - b. What do you think these numbers indicate?
 - c. Where did you first look to when you signed out?
 - d. Have you seen this screen before? What do you think it is?
 - e. How do you feel about building your new robot?
5. Can you access the framework documentations for Roboterra? Can you report a bug to Roboterra?
 - a. How did you know to look there?
 - b. What would be helpful to help you find the buttons better?
6. Please do Challenge #1.
 - a. How did you know how to connect all the parts?
 - b. How did you know you were connected?
7. (Intentionally disconnect RoboCore). Please press the hazard button. How are you going to fix this problem?
 - a. Where did you look to first?
 - b. How did you know to look there?
8. Please do Challenges #2 - #4.
 - a. What was the hardest part about the challenges?
 - b. Have you done something similar to this before? When and how was it different from this experience?
 - c. When you were having trouble, how did you want to get help?
 - d. Have you got any questions for me?
 - e. Where do you initialize your robot?
 - f. Where do you find the eventSource of the event?
 - g. When you finish a challenge, how do you check if it works?
 - h. Can you go back to an old challenge and trying doing it again?
 - i. How did you know you were finished?
9. What do you think of the virtual robot now?

Method #3: Moodboards

I created a moodboard to scope best practices in web app designs.



Wireframes

ROBOTERRA

G PointLight

```

1 // Container that passes loginStatus and functions
2 import { connect } from 'react-redux';
3 import LoginButton from '../components/LoginButton';
4 LoginButton'
5 import { push } from 'react-router-redux';
6 import { setLoginStatus, setUsername } from '../actions/actionCreator';
7
8 // Gives LoginButton a loginStatus bool
9 const mapStateToPropsToProps = (state)=>{
10   return {
11     loginStatus : state.loginStatus,
12     username : state.username
13   };
14
15 // Two functions to help determine whether login
16 // was sucessful/legal etc.
17 const mapDispatchToPropsToProps = (dispatch) => {
18   return {
19     setError: () => {
20       dispatch(setLoginStatus('error'));
21     },
22     goToHome: (name) => {
23       dispatch(setLoginStatus('true'));
24       dispatch(setUsername(name));
25       dispatch(push('/home'));
26     }
27   };
28
29 // Connects with loginButton
30 const LoginContainer = connect(
31   mapStateToPropsToProps,
32   mapDispatchToPropsToProps
33 )(LoginButton);
34
35 export default LoginContainer;

```

INSTRUCTIONS

We're going to first initialize the base servo with a final angle so that it knows what angle to stop in. Write this code below your initialization for the names of your component parts:

```

int base_angle = 140;

```

Next, we rotate the base servo. Type this code:

```

if (baseStatus == 1) {
  if (base_angle == 140)
}

```

NEED A HINT?

ROBOTERRA

G Dark Spot Detector

3. Operate the first servo, part 1

spot. Now we're getting to the fun stuff!

INSTRUCTIONS

We're going to first initialize the base servo with a final angle so that it knows what angle to stop in. Write this code below your initialization for the names of your component parts:

```

int base_angle = 140;

```

Next, we rotate the base servo. Type this code:

```

if (baseStatus == 1) {
  if (base_angle == 140)
}

```

NEED A HINT?

ROBOTERRA

G Dark Spot Detector

PROJECT PREVIEW

- 1 Prepare components ✓ from your box.
- 2 Make the detector ✓ base.
- 3 Operate the first servo, step 1. ✓
- 4 Understand what a servo is. ○
- 5 Two servos work together. ○
- 6 Build a two-servo structure. ○
- 7 Detect the dark spot and stop. ○
- 8 Install the light detector. ○
- 9 Test the dark spot detector. ○

ROBOTERRA

2. Make the detector base.

With anything we build, we need a solid foundation to stand on. That goes with life and with robots! In this challenge, we are going to make the detector base. But first, let's introduce the concept of an actuator. We're going to build the base structure for our first servo, which we will now call the base servo.

INSTRUCTIONS

Build your robot! We've prepared an animated tutorial to help you out. Click the button below when you're finished building.

DONE BUILDING

ROBOTERRA

ALL PROJECTS





Challenge #1: Dark Spot Detector

90%

CONTINUE >

Challenge #2: Candle

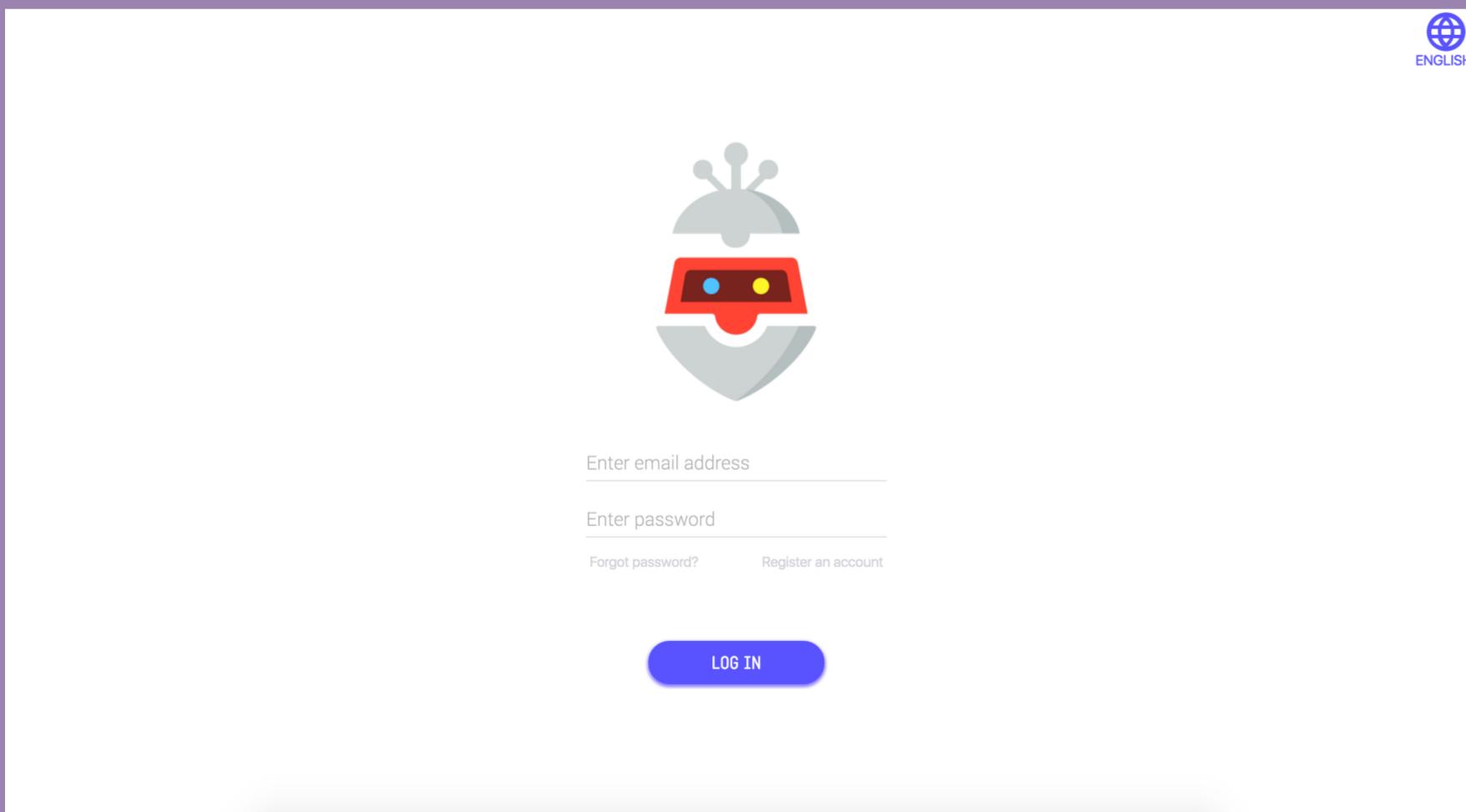
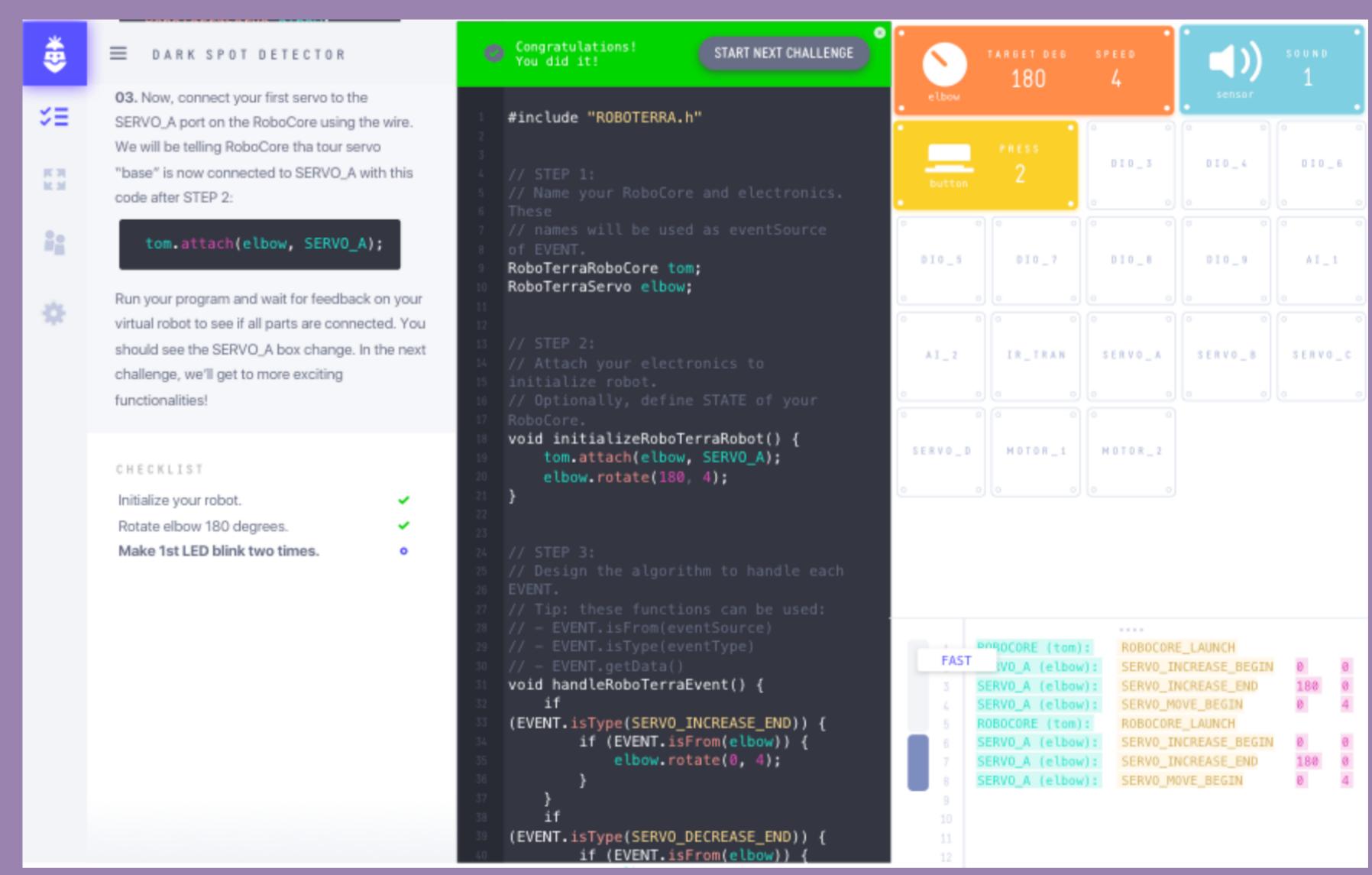
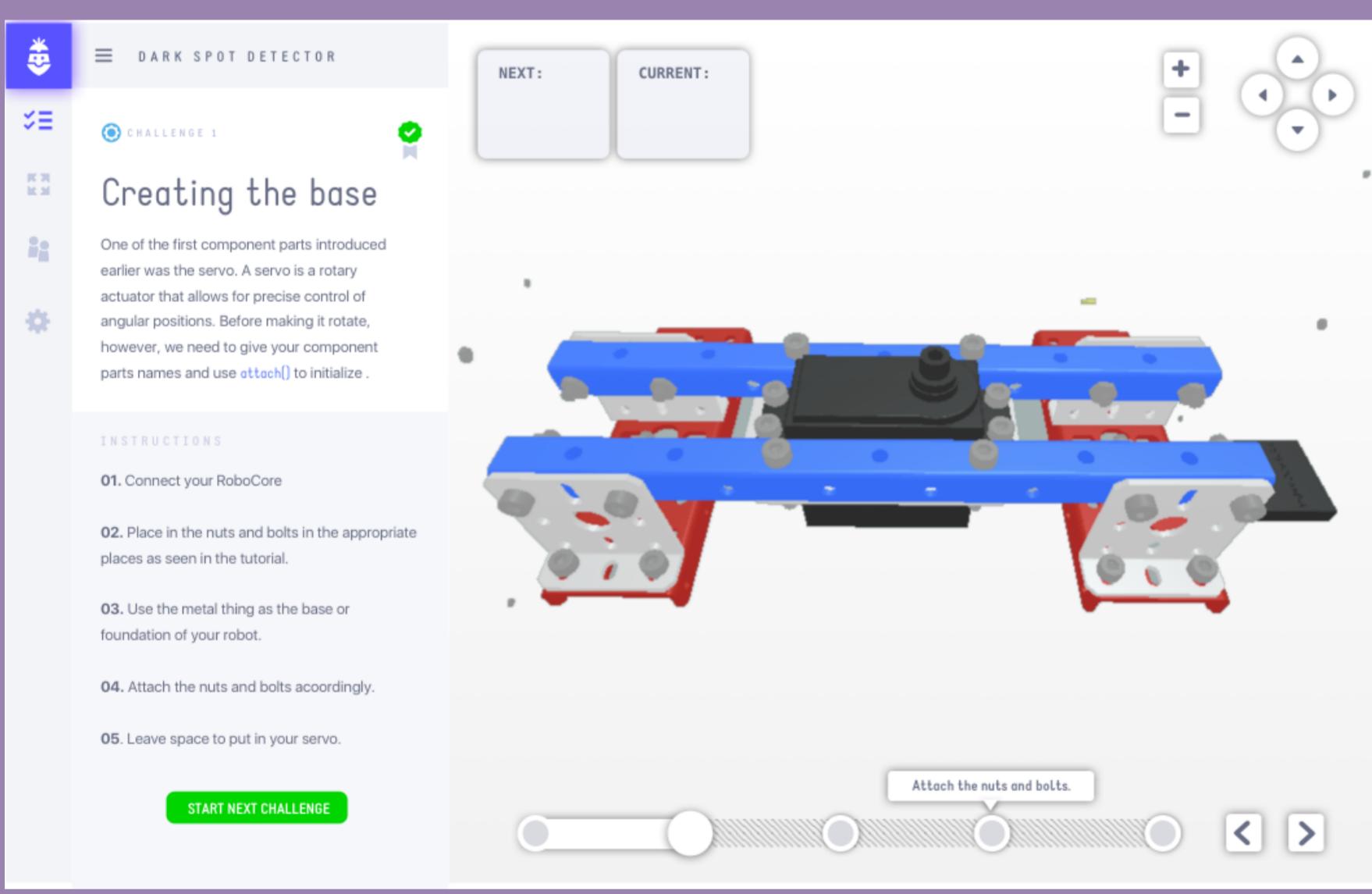
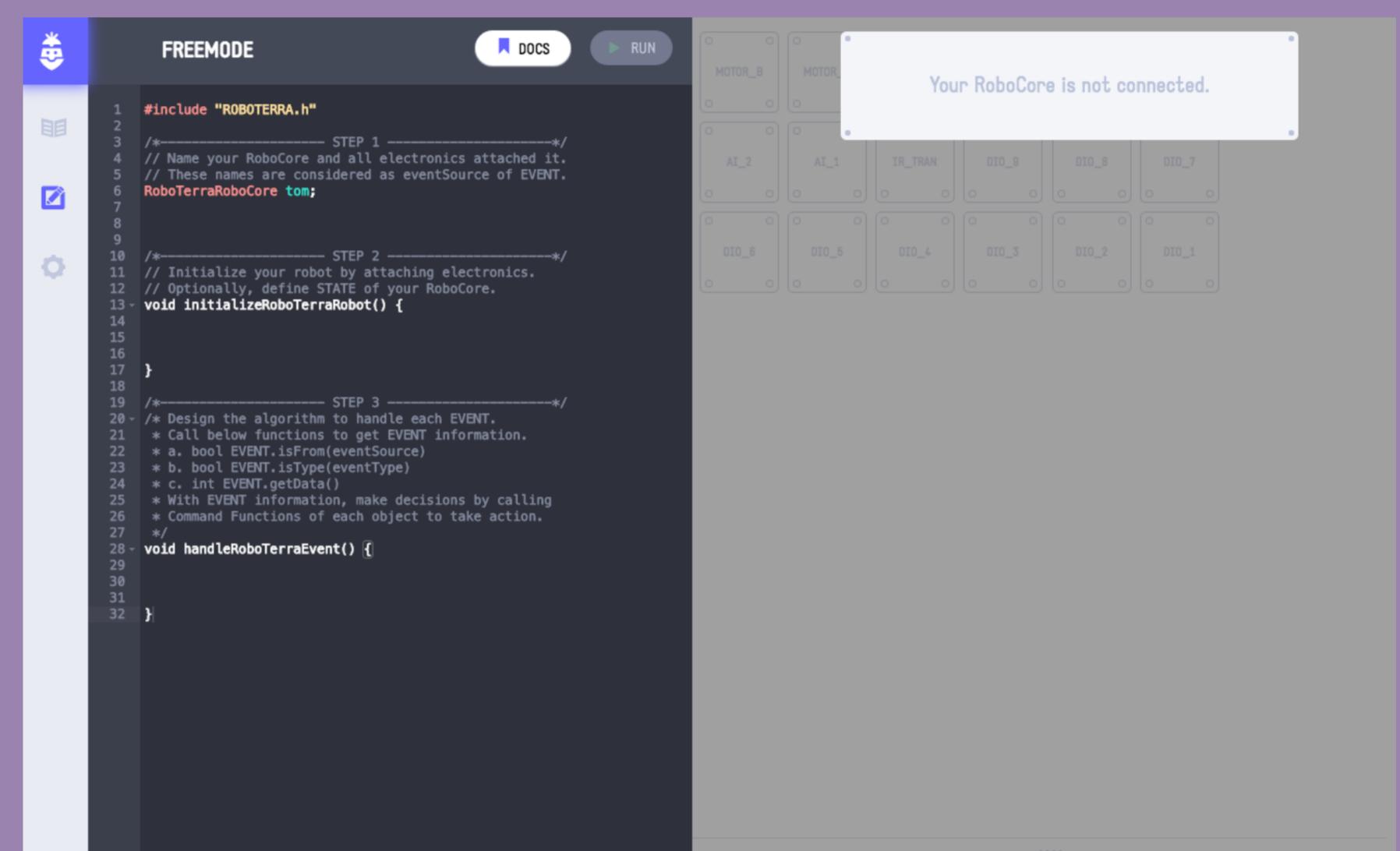
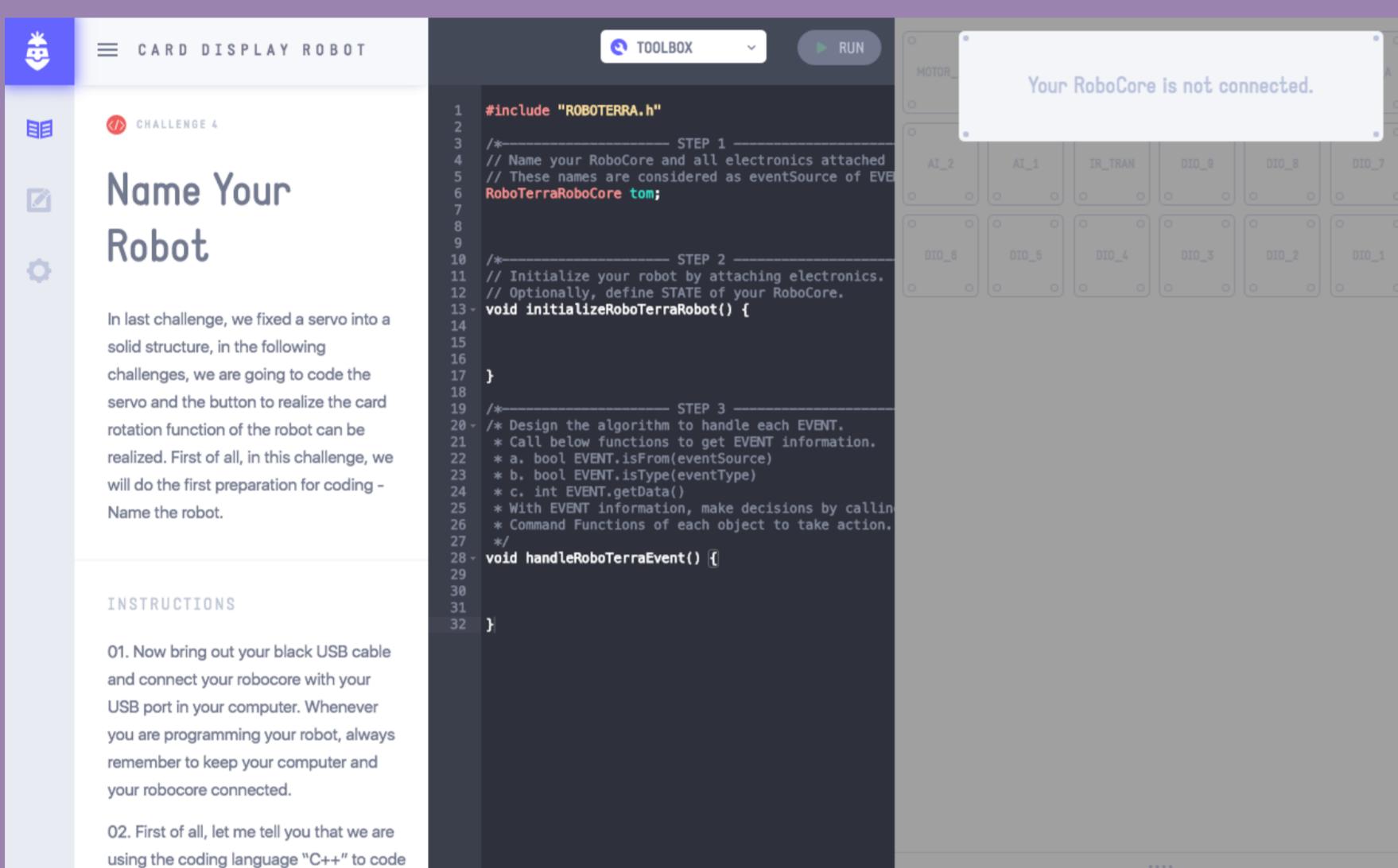
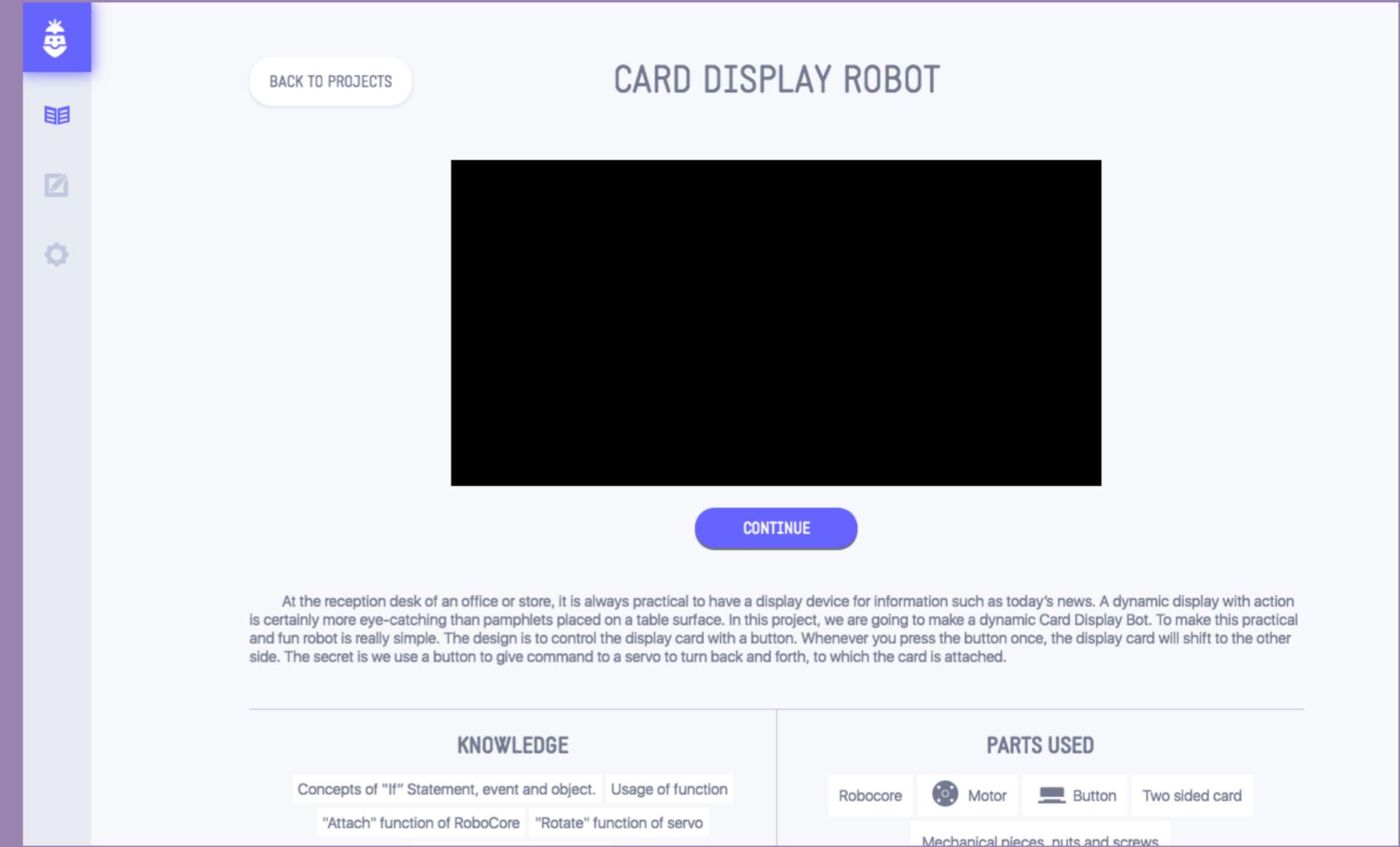
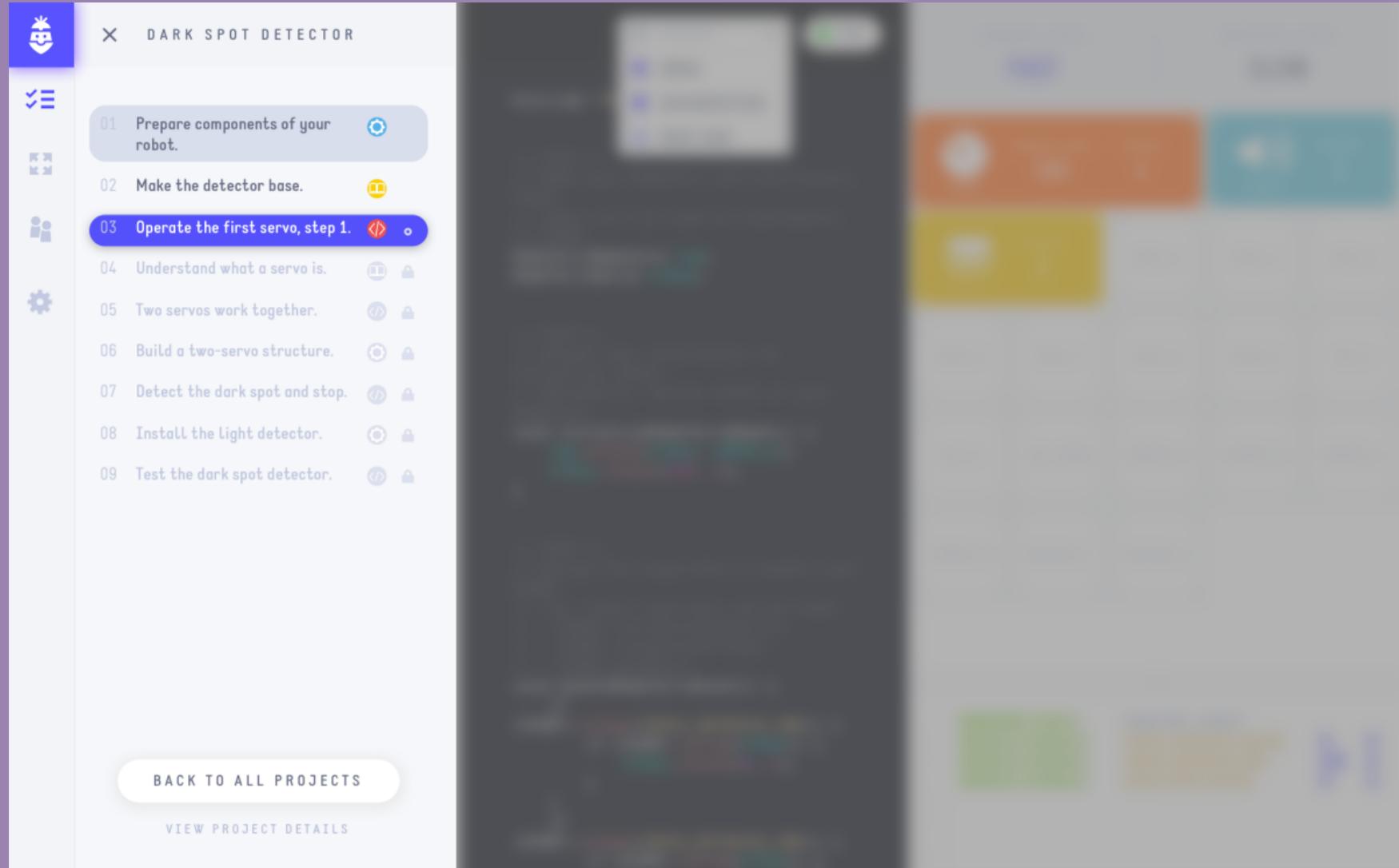
0%

START >

This level is unlocked. Finish Challenge #2: Candle to unlock this level.

I made 3 versions of wireframe sets with a total of 54 wireframes. I also created an InVision version, however, for confidentiality, I am unable to post the link.

Mockups



I made 9 versions of mockups with a total of 100+ boards. I also have an InVision version but am unable to post that here.

Moving Forward

We built a pretty sick product within 2 months -- great work for 4 interns! Before leaving Roboterra, however, I left the next designer a list of things we were unable to add (and a short and sweet letter!)

Dear next Roboterra designer,

Hey! I'm Mika Reyes. As far as I know, I'm the first full-time designer on the team and it's been a great learning experience. I hope you learn as much from it as I did.

Anyway, since you're hired, I'm sure the team puts a lot of trust in your capabilities, not just as a designer, but also as a possible "mini-software engineer." In the JD, it highlights that you do need to be an HTML/CSS guru and be willing/capable to learn other frameworks and languages (i.e Javascript, React, Redux, etc.). I should know, I wrote the JD! If you don't currently know, no worries. There's lots of room to learn, with an atmosphere of really helpful people.

A lot of your work will have you take a lot of initiative and proposing projects and implementing ideas. Don't be afraid to share them. People will listen!

Besides this, here is a list of ideas I had for the future of CastleRock that we weren't able to implement during my time here, but I'm sure can be implemented (or innovated on) in the future! A lot of it, I wrote without proofreading, but I'm hoping it makes enough sense.

Leave that amazing design legacy of yours! Good luck and I wish you the best!

Best,
Mika

olor to cater to the branding guidelines of Roboterra. t up to me late, so I didn't have time to change up stuff, rrently out of place logo on the eternal top left corner. g maybe change things up to a blue palette to match the logo? ation functionalities! sponsive hover and flawless animations. Sooo important. en add more interactive components, this one I'm not an , but maybe you are! This is important, I believe, since we cater to kids.

at with whatever new feature or part you add, is to change up as well!

LOG IN
[Forgot Password](#) or [Register an Account](#) according to the

s are already setup, but I imagine there may be changes e sales team wants to be setting up registration. Listen to ls and follow through with the designs. ter an Account has two "pages." The second one asks ' information but this extra information is pertinent data for us to collect. Make sure we collect enough data that it doesn't overwhelm the user, but gives us enough information to use for the future.

- When more languages are implemented, change the top right logo (as needed) to make it cater to the different types of languages CastleRock can carry.

CODE CHALLENGE

- The chunk of text for the instructions gets very daunting to read for the student. I tried breaking it down with code text (both inline and with the codepad editor), but it still looks a bit intimidating. Maybe there are other ways (including asking the content team to shorten their content) to improve how it looks so the kids don't get scared. Maybe even not showing the instructions all at once? Just a thought

QUIZ CHALLENGE

- There are better ways to give feedback for getting wrong or right on the quiz. It was difficult to implement, but if there's a better way to give feedback (or a great way to test this quiz to the kids), explore it.

CASTLEROCK

designs for these, but we realized we didn't actually need it. d be an all-in-one dashboard that encompasses they can do and that we allow them to do. It might also important announcements we want to show our users, to freemode or to a project they're currently working

is implemented, perhaps a run down of the progress and the global progress as well. ssible ideas to innovate here!

s able to cater my designs based on data that was to me about how people were currently behaving on the pitched this more data-centric atmosphere, but I feel erra's needs are focused on building the platform first, king about the backend further.

n, though, since Roboterra moves quickly, I imagine e ready enough to engage in data. Try to establish a -driven approach, first by collecting current data from nd (no matter how small they say it is!) and basing your f A/B tests around that.

setup an interface that helps users see the data and compare their progress globally, so we can get them platform more often.

- o There should be a way for us to collect relevant data (clicks per challenge, time length in each challenge or project, etc.) so we can eventually sell it to education professionals. This data will help with our business model, but also will help the education professionals understand their students. We can categorize these according to each school or by age bracket, etc.

Saving Feature

- o I already discussed this in the Freemode section, but we want to enable the students to save the code they wrote so they can access it earlier.
- o Perhaps this actually gets saved on their computer as a file, perhaps it's internal on CastleRock. Either way, the point is, we