Animations In Stress Learning Content: Final Report

Product Owner:

Jincheng Li: lijincheng0903@tamu.edu

Scrum Master:

Xiaomu Dong: xid2019@tamu.edu

Team Members:

Ruxin Zhang: ruxinzhang@tamu.edu

Ruichao Wang: chaochaoddln@tamu.edu

Wenlin Li: wenlin_li@tamu.edu

Jia Fu: fujia_tamu_2020@tamu.edu

Course Information:

CSCE 606: Software Engineering Instructor: Dr. Hank Walker Spring 2021

May 1, 2021

Table of Contents

1. Summary

The project cost our team over one month to set up a variety of quiz animations that have user interaction to enhance students' learning experience. Each animation and mini-app are deployed under the stepstone learning environment, which can be used in different browsers and devices. They are mobile-friendly and debug free. Our team pay attention to the Stress Module, one of part of the One Health curriculum, providing basic biology knowledge through quiz. Torri Whitaker, the content specialist for the per program, provides the slides and animations, and Dr. Walker is our customer.

Each quiz animation have developed based on HTML5, JavaScript, and CSS. Each team member is assigned their own user story to finish the corresponded quiz animation. The product owner is responsible for the test in Stepstone environment and give feedback and suggestions. As a result, the corresponded mini-app is created in local development. Then, we test them as FTP server in Stepstone environment. Finally, the Stepstone developer Daniel helped us embed our apps into production.

Ultimately, our team completed all tasks that requires user interaction animation. All completed user stories are compatible with most browsers and touch screen compatible. In addition, automated test cases are finished for all completed user stories to help further work. More details about our project could be found on Youtube video: https://www.youtube.com/watch?v=lxtMF4Zcnwo.

2. Development and Management

a. BDD/TDD

We followed the Agile methodology with 3 iterations. Google drive is used to modify the user stories, Wechat is for the constant communication, Zoom is for team and customer meetings, and Github is for the version control. Although based on this method some issues are induced between the production and testing, most of tasks are completed smoothly and quickly. The StepStone environment is sandboxed in an "iframe" element which caused various sizing and loading problems. In addition, the staging environment (FTP server and staging URL) was shared across multiple developers, so only one developer could test their app at a time.

b. Configuration Management

Because the each user story are completely independent from each other. It is easy for us to develop them one by one to create their own animation in the StepStone environment. With no branches and two releases, one for each iteration after iteration 0, it is easy to manage our project. Each includes the process what we done for the user stories. We only have one issue. The touch screen compatible is not available for Drag and Drop (Slides 24, 54, and 66). we conferred with our customer and were able to close this extraneous story.

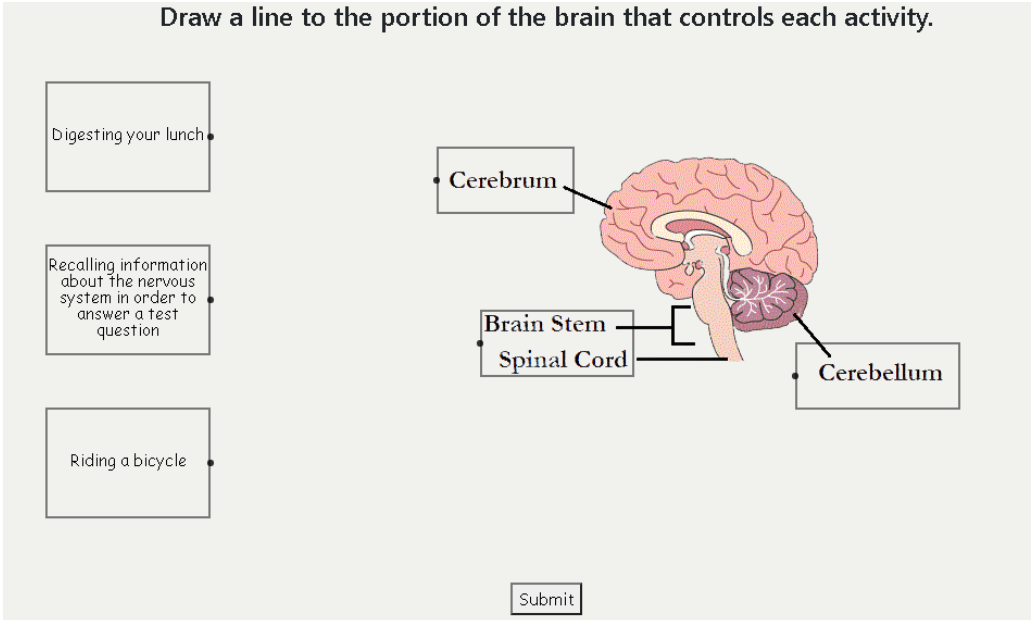
3. User Stories

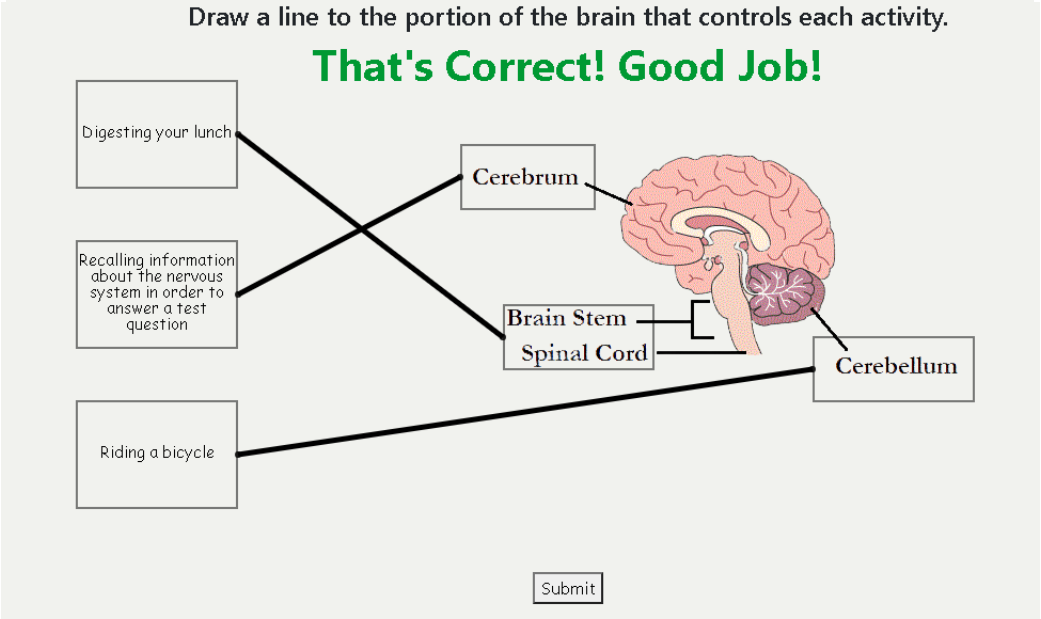
a. User story 1 - slide 12&42 (Completed)

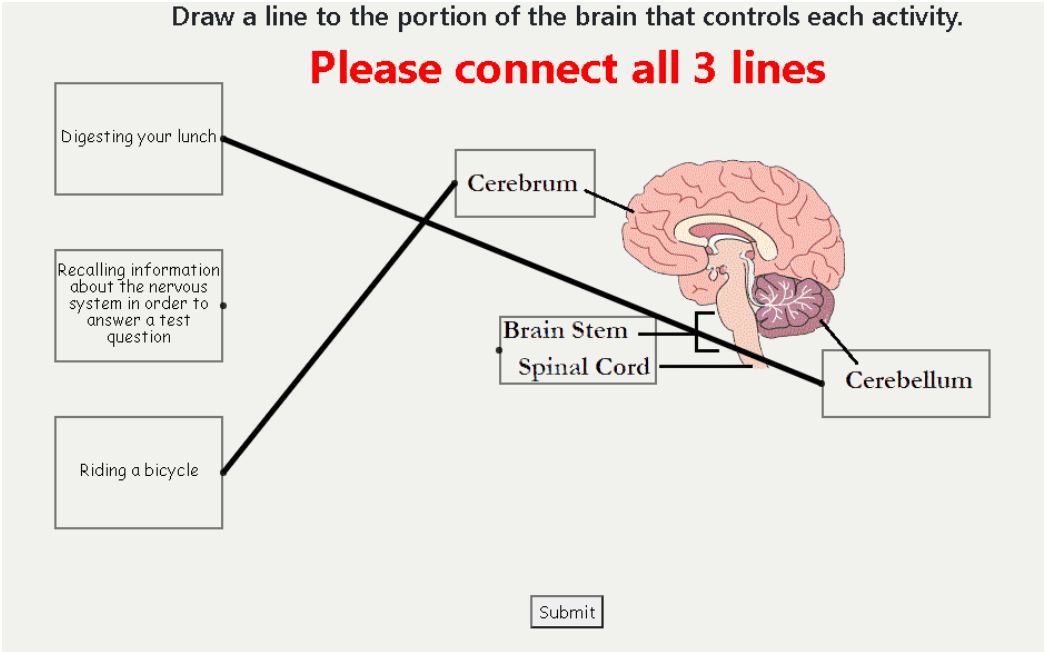
Feature: Feedback animation for drawing a correct line to the right portion of the brain.

As a user, when I draw a line that connects the function to the brain portion, I want to see the correct or incorrect feedback. We improved this slide and added a submit button to control the data flowing. As shown in the following figure, once I connect the right lines and click submit, we can see feedback from the slides.

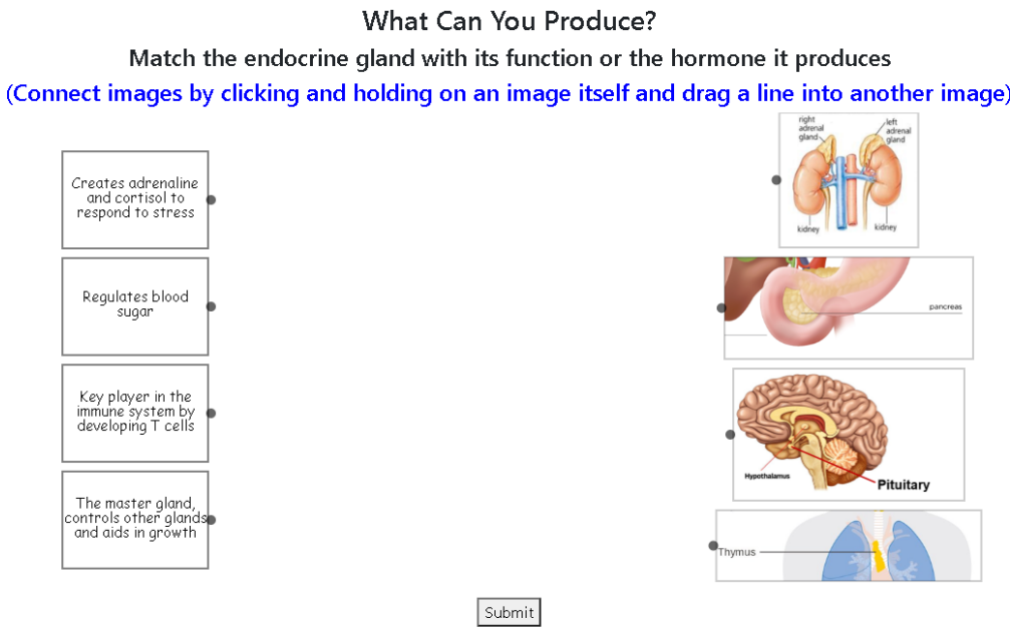
UI sketch for user story 1 - slide 12:

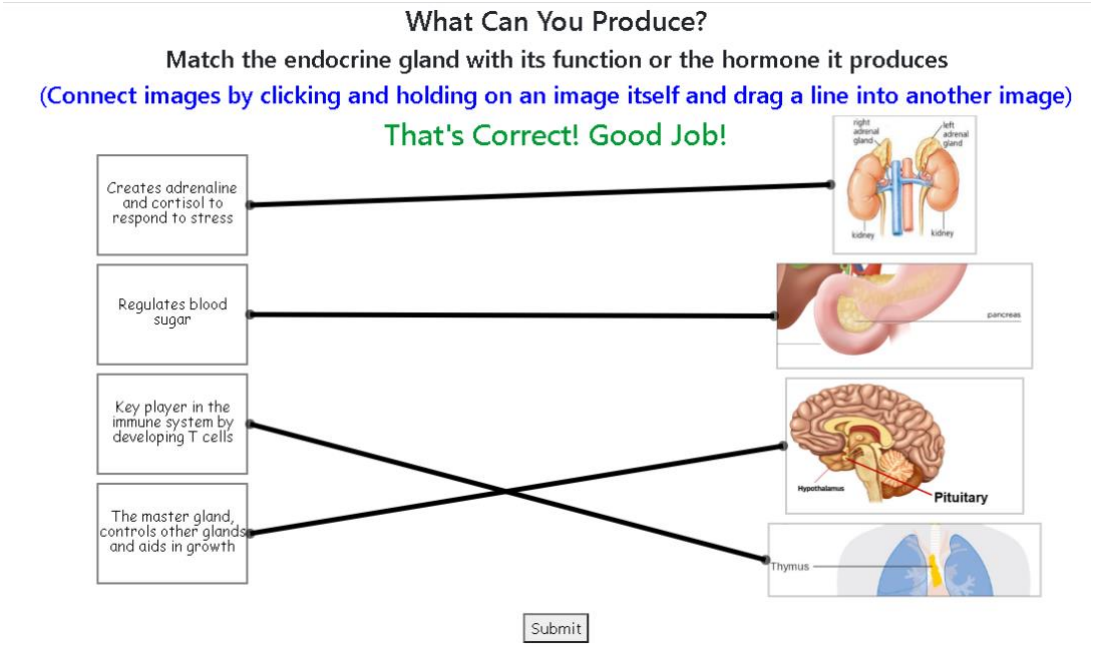


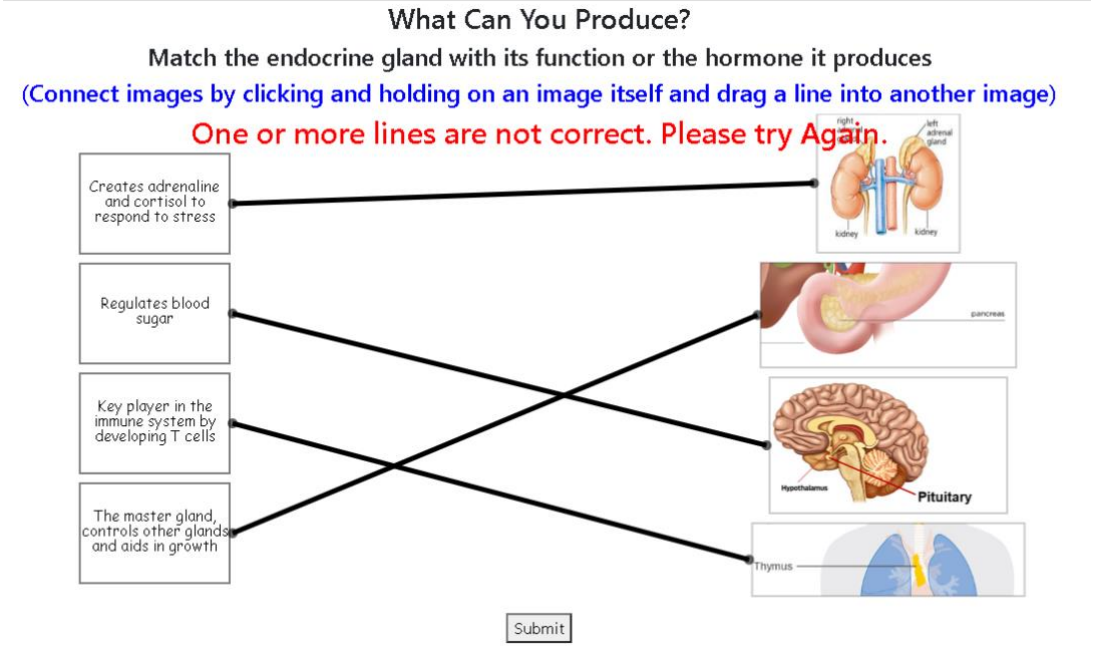




UI sketch for user story 1 - slide 42:





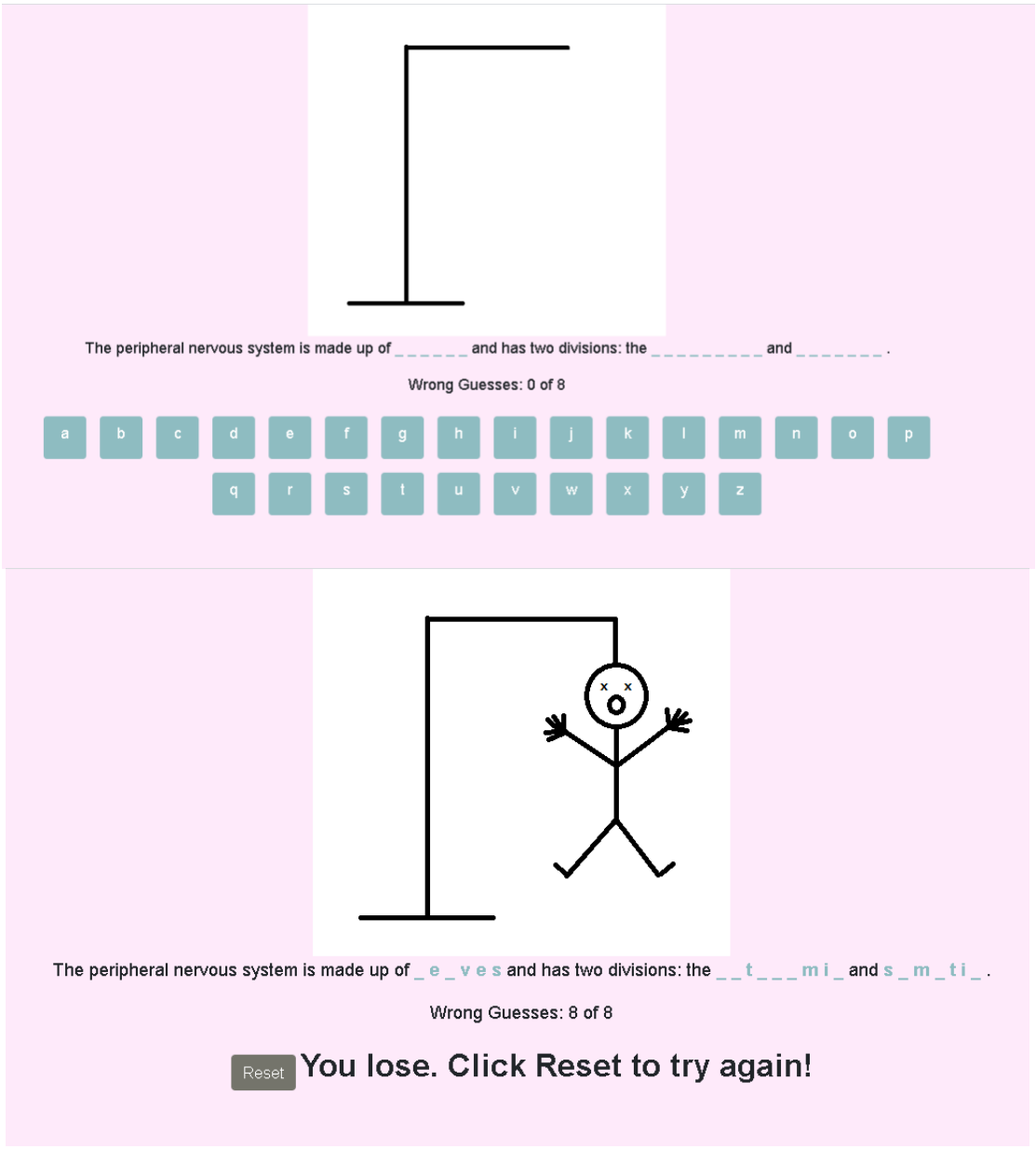


b. User story 2 - slide 15 (Completed)

Feature: Feedback animation for showing whether the word spelling is correct or not.

As a user, when I type in a correct letter, I want to see the letter is blue. When I type in a wrong letter, I want to see one more line for the hangman. When I have eight wrong letters, I want to a complete hangman and lose the game. Then, I can restart the game by clicking the reset button.

Storyboard for user story 2 - slide 20:

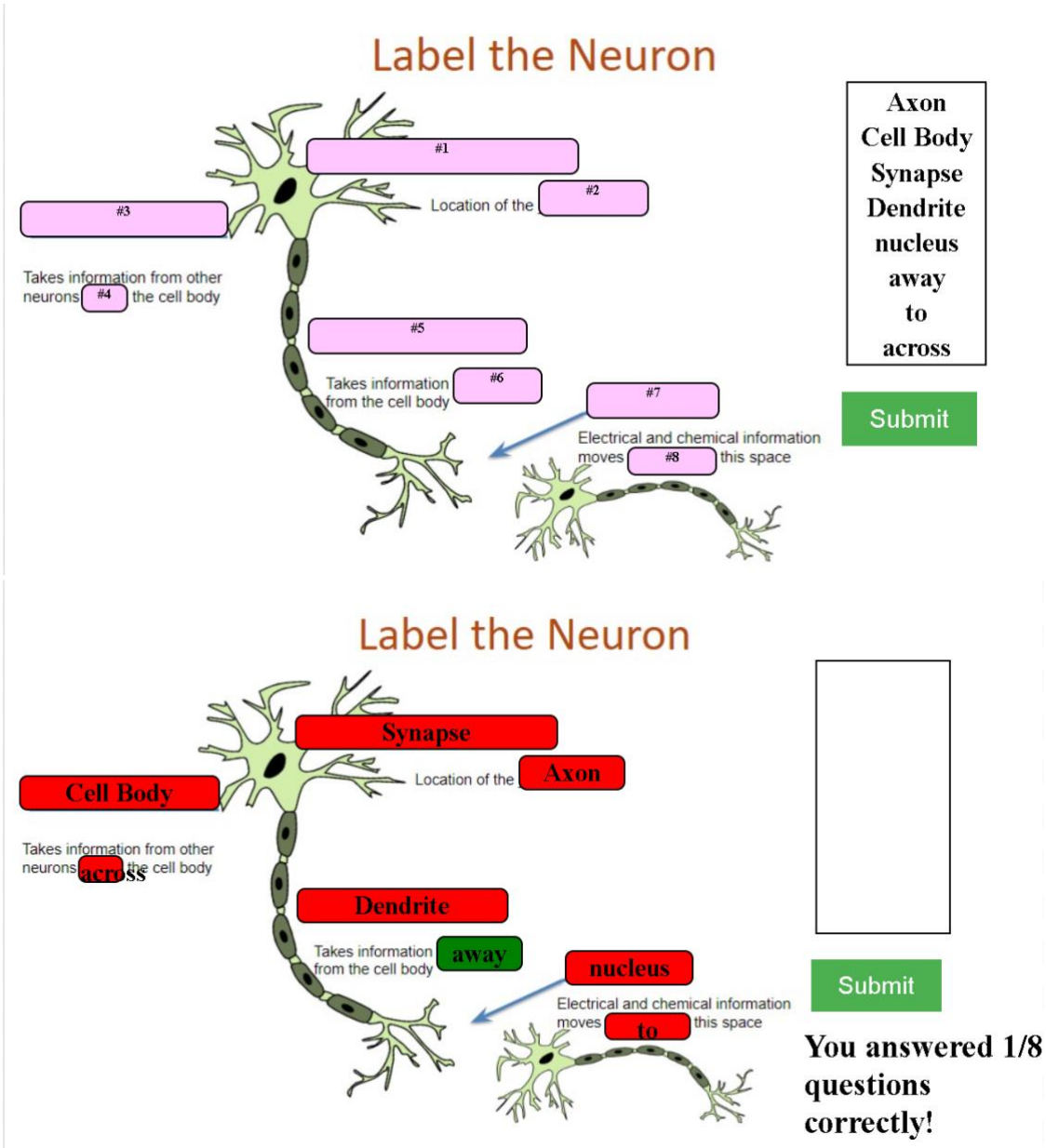


c. User story 3 - slide 24 & 54 &66 (Completed)

Feature: Feedback animation for dragging the correct word to the blank.

UI sketch for user story 3 - slide 24:

As a user, when I drag the correct word to the blank, I want to see the word stay on the blank. After pulling all words to the blanks, click submit button. I expect that the slide tells me how many answers are correct. The right answers become green, while the wrong answers get red.



UI sketch for user story 3 - slide 54:

After completing all the blanks and clicking the submit button, I expect to see feedback from the slide. If I am wrong, the slide displays “Incorrect”. When I get the correct answer, I will get feedback that says “Correct”.

Please place the correct answers in the blank below questions

What is the term which means maintaining a constant internal environment despite changes in the external environment?

Answer: _____

Which two body systems help regulate the internal environment?

Answer: _____ _____

Balance
Muscular system
Homeostasis
Endocrine system
Stimulus
Nervous system
Reflex arc
Cardiovascular system

Submit

Please place the correct answers in the blank below questions

What is the term which means maintaining a constant internal environment despite changes in the external environment?

Answer: Cardiovascular system

Which two body systems help regulate the internal environment?

Answer: Balance _____ Homeostasis _____

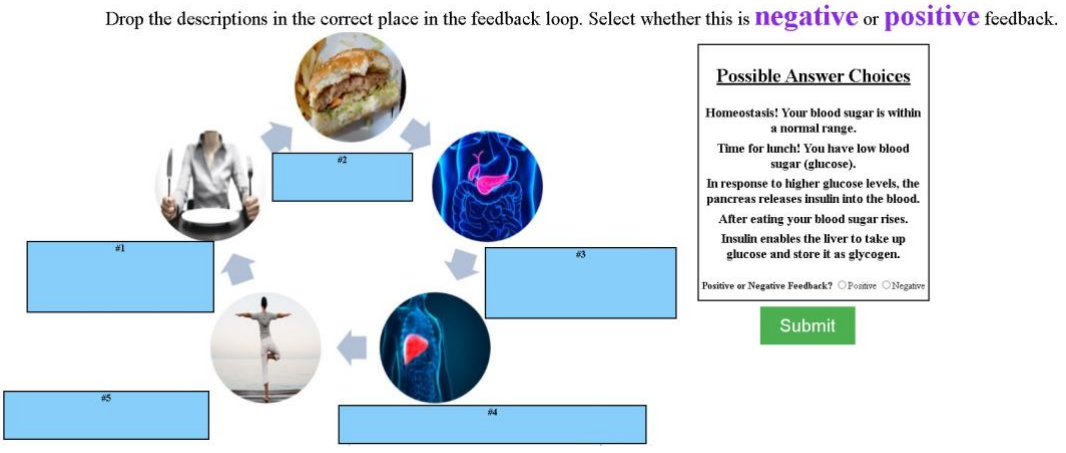
Muscular system
Endocrine system
Stimulus
Nervous system
Reflex arc

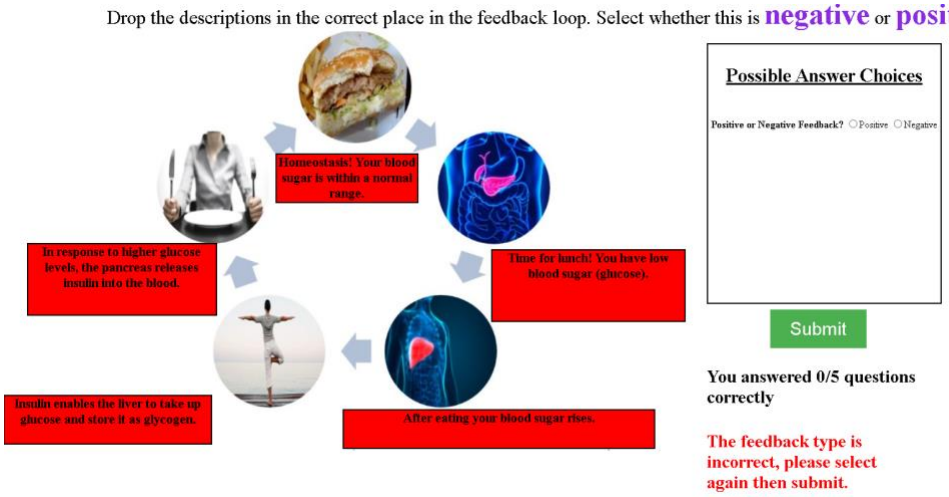
Submit

Incorrect!

UI sketch for user story 3 - slide 66:

As a user, when I drag the correct word to the blank, I want to see the word stay on the blank. After dragging all words to the blanks, click submit button. I expect that the slide tells me how many answers are correct. The right answers become green, while the wrong answers get red.



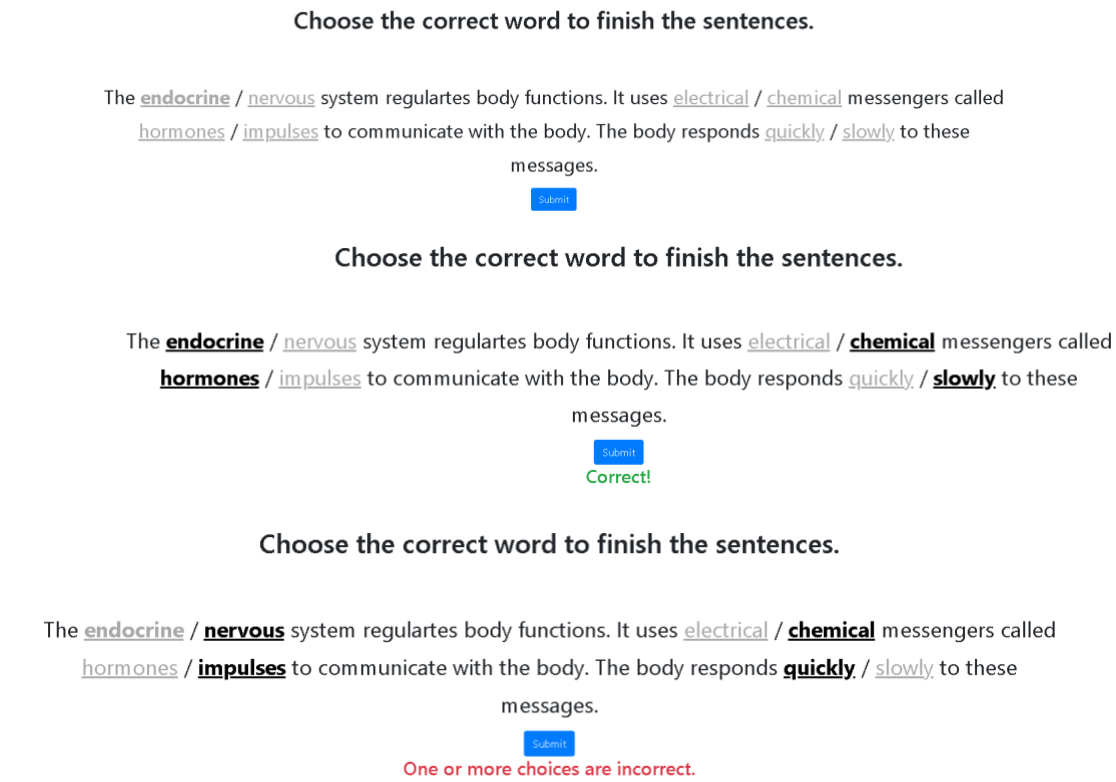


d. User story 4 - slide 33 (Completed)

Feature: Choose the correct word to finish the sentence of the fill-in-blank question. Formatted the layout of the words.

As a user, when I click on the words in the sentence and submit my answer, I want to see if I'm correct or not. If I am right, the slide shows "Correct". If I am wrong, the slide will give feedback and display "One or more choices are incorrect."

UI sketch for user story 4 - slide 33:



4. Iterations

a. Iteration 0

The primary aim of this project is to create animations to be used in StepStone, a learning environment currently used by the PEER Program in Texas A&M's College of Veterinary Medicine to aid in providing TEKS curriculum content for use by 6th - 8th-grade teachers and students. The Peer homepage further links to the One Health curriculum, named after the biological reality humans and animals share the same biology and many similar diseases. The curriculum currently offers multiple modules, and past projects involved the Cell Biology Module, such as creating games and animations so that the learning experience would be more interactive and engaging for students. All the animations will ultimately be integrated into the already existing StepStone module to improve the current slides.

b. Iteration 1

In this iteration, we met our customers by Zoom and discussed the detailed goals of our project. In addition to the slides in stress learning content, we received more optional work related to infectious diseases. Also, we checked our previous work and the legacy code. We are planning to try to deploy our project to StepStone and see if it works.

c. Iteration 2

In this iteration, we met our customer by Zoom, and we went over our progress since the last time we had talked with him, specifically including demonstrations of things we've done with our animations and with the StepStone testing environment.

5. Customer meetings

a. April 8th, 2021, 10:00 A.M.:

This was our Iteration 0 meeting with our customer, Dr. Walker, who served to introduce our team to the goal of the project in his eyes, get clarification on the expectations of the animations we needed to produce and ensure that we had the necessary materials and contacts we would need to be successful throughout the project.

b. April 15th, 2021, 10:00 A.M.:

We met our customer by Zoom and discussed the detailed goals of our project. In addition to the slides in stress learning content, we received more optional work related to infectious diseases and during this meeting we got extra clarification on how to get started using StepStone, who we should contact and make sure that everything was in order for us to make progress on our user stories.

c. April 22nd, 2021, 10:00 A.M.:

We went over our progress since the last time we met, specifically including demonstrations of things we've done with our animations and with the StepStone testing environment. Earlier this week, we sent a detailed inventory of what has been deployed and

developed at the request of our “real” customer Torri Whitaker and got the response that she was satisfied.

d. April 29th, 2021, 10:00 A.M.:

We went over our progress: fixed all the bugs in the legacy project and successfully deployed all the developed animations on StepStone with the help of Daniel. Also, Dr. Walker specified the expectations and requirements for the final report/poster/demo.

6. Issues

a. Pivotal Tracker

Organizing user stories originally proved problematic, as each customer item was underspecified. Because of this, we had to reorganize and iterate on our stories multiple times over the course of the project as we better understood what was desired of us. Each story was atomically very large, with each animation being individual but intensive work. Because of this, our Agile process was a bit different from usual, so we had to adjust to this change. We had one story that was already complete, but it took a bit of investigation to uncover that no further work was needed.

b. Integration with StepStone

Because mini apps are placed inside the StepStone environment in a customized “iframe” element, the apps had to be resized using an iframe resizer. If this isn’t triggered properly, the app is incompatible with StepStone; you can refer to Tutorial 5 for more information. Another obstacle we overcame was touch screen compatibility: some mini-apps used mouse events in the canvas element. Since mouse events are not the same as touch screen events, we added event triggers for “touchBegin”, “touchDrag”, and “touchEnd” to certain mini-app scripts.

The browser compatibility issue was complicated because some apps worked flawlessly in all browsers except that one where loading or interaction just doesn’t work properly. Some mini-apps used canvas and the “2D context”, which had some issues on Firefox. Wrapping the render function in a “try, catch” statement solved the issue, as the render function would error on load up for firefox. Making sure to test apps on different browsers is important because there might be inconsistencies or discrepancies that would cause your app to fail for users. We also experienced numerous loading issues that would not happen locally. Daniel Shuta added an interval function to the beginning of our document.ready function in the JS file and told us to clear the cache when testing, as the iframe resizer works oddly in the StepStone environment.

7. Legacy

a. Prior Code

The legacy project we are referring to is the Animations in Stress Learning Content developed by the team HalfGrads in the previous semester. Their animation repository contains animations of different modules, including drag and drop, drag and connect, Hangman, ordering images, weighted scale and word selections. In each module, they include different slides that utilize those animations.

b. Our Code and Configuration

If you would like to reuse any of our mini-apps for your own slides:

1. Connect Lines:

To reuse the Connect Lines animation for other similar slides, you will need to configure the JSON object called "config" in "/src/main.js" defined at the top of the "init" function. This JSON object has a structure that determines the answer boxes (left side) size, position, and text content, and the image boxes (right side). To configure the left boxes, just add box objects in the "boxesLeft" array pertaining to the answer you want to represent. For example, this represents a box on the left side that has text content "Digesting your lunch" and answer at the index 1 in the "boxesRight" array:

boxesLeft: [{
content: "Digesting your lunch", side: "left",
answer: 1,
}]

The "boxesRight" array is similar but will require extra configuration, and is best explained in the code. Slide 12 is an example where there is only one image and multiple boxes per image, while slide 42 has multiple images each with only one box.

2. Hangman

Reusing Hangman is easy. "hangman.js" is solely contained in a window.onload function. At the top of this function, replace the sentence variable with whatever sentence you would like filled such that each word guess is replaced by '1?', then '2?', and so on for however many words must be guessed. Then, place each word as a string into the words array immediately below '*sentence'* . If you've followed these instructions, as well as the more detailed instructions in the comments within "hangman.js ", you'll be able to adapt it however you like.

3. Select Words

Reusing the Select Words is straightforward and modular. It requires a few small changes such as the answers array in the wordSelect.js and studentAnswers array to be the correct length. Next, all that is required is the clickable options (labels were used in this scenario) in the HTML are configured to call the question function with the question number it is in reference to on click. Finally, a button connected to the submit function and an element with the id of "result" will be edited to provide feedback.

4. Drag and Drop

First, swap the background image accordingly to the one required for the slide; this can be done in the "style.css" file.

background-image: url("img/neuron.jpg");

Second, change the location and size of the drop boxes by changing the absolute pixel percentages, found in "index.html" under the div with class "question-container". In the code below, I have bolded the aspects that need to be modified.

<div id = "#1" class = "dropBox" style = " top:16.8%;
*left:28%; width:25%; height:4%" >#1 *

One thing to note is that this user story only works on mouse events. It will not work on mobile touch events, so a few lines of code will need to be added to the "js/drag_and_drop.js" file in order to account for this.

5. Ordering Images

Simply reuse the "sorting.js" file and refer to "index.html" to coordinate element IDs, as each image element will have a corresponding ID indicating the correct order. The code for the image ordering slide expects five images to be ordered, and each image will have an ID such as "first", or "second", indicating their correct order in the entire lineup of images. The code for animating how an image snaps to the cursor, gets relocated to a new place in line, and bumps all images to the right one space over, can all be replicated exactly to simulate the animation effects.

8. Logistics

a. Team Roles

Product Owner: Jincheng Li

Scrum Master: Xiaomu Dong

b. Our Project's Important Links

Github Repository:

https://github.com/jinchengli97/Animations-in-Biology-Learning-Content-2

Pivotal Tracker:

https://www.pivotaltracker.com/n/projects/2495412

Project Video Demo:

https://youtu.be/lxtMF4Zcnwo

c. Other Important Links

One Health Modules:

https://vetmed.tamu.edu/peer/one-health/

StepStone Editor:

https://stepstonelearning.net/

d. Contacts

Daniel Shuta, StepStone developer: dshuta@cvm.tamu.edu

9. Tutorials

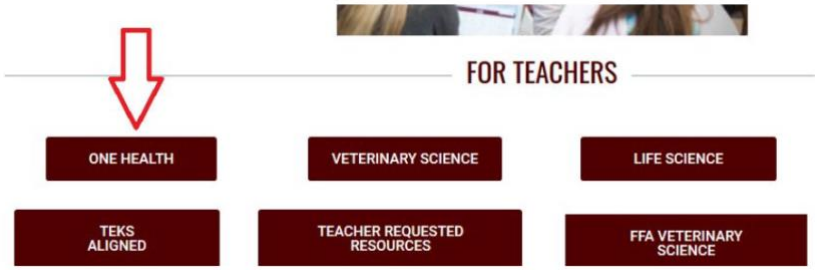
a. Tutorial 1: What is StepStone?

StepStone is the learning environment system created by the Peer Program. The relevant web pages and steps for reaching the StepStone modules are as follows:

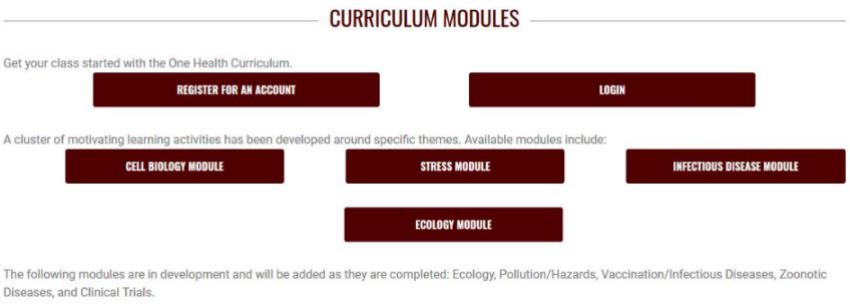
1. Visit the Peer Program home page: https://vetmed.tamu.edu/peer/

2. Near the middle, click the button that says *One Health*.

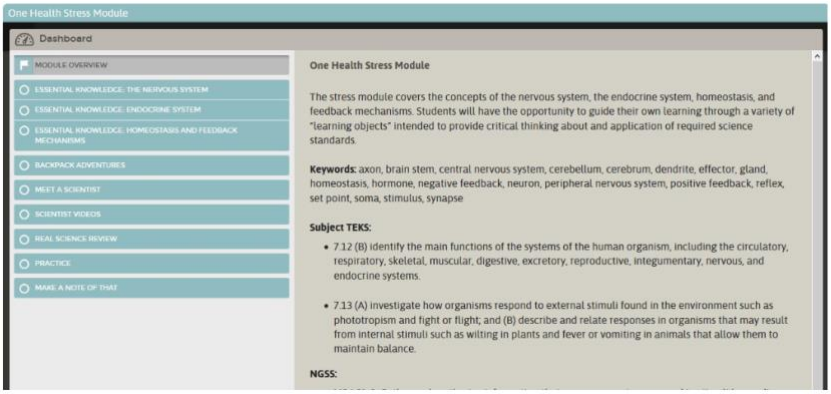
The direct link is: https://vetmed.tamu.edu/peer/one-health/



3. On the page, there is a section titled Curriculum Modules. This is the production environment where, depending on the module you work on, your mini-apps will be placed. No development is going to occur on this site, but you will go here to view your apps in action or determine where they are going to be placed.



4. When you click on a module, you will be redirected to the StepStone learning environment and can then navigate it using the Dashboard and internal paths.



b. Tutorial 2: How to Test Your Apps in StepStone

As of 05/02/2021, there is no way for you to stick our apps into the production environment; Daniel Shuta must do this for you. We are aware that Daniel is hard at work creating a method so that you CAN upload your apps yourself. Regardless, this tutorial explains how you can put your apps into a testing environment that Daniel provides.

1. Download an app capable of establishing an FTP connection such as WinSCP or FileZilla. I used WinSCP, so this tutorial may have steps that won't match exactly with other clients.

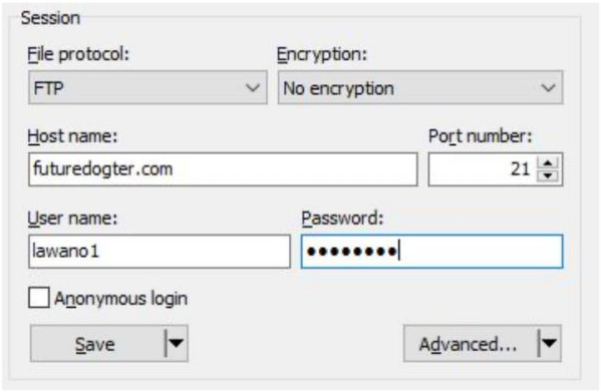
2. Use the following credentials:

Server/Host: futuredogter.com

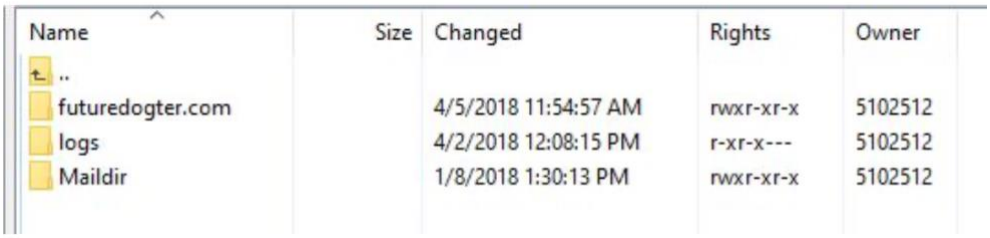
Username: lawano1

Password : H-8Uqp5Z

In WinSCP, establishing a new session should look something like this:



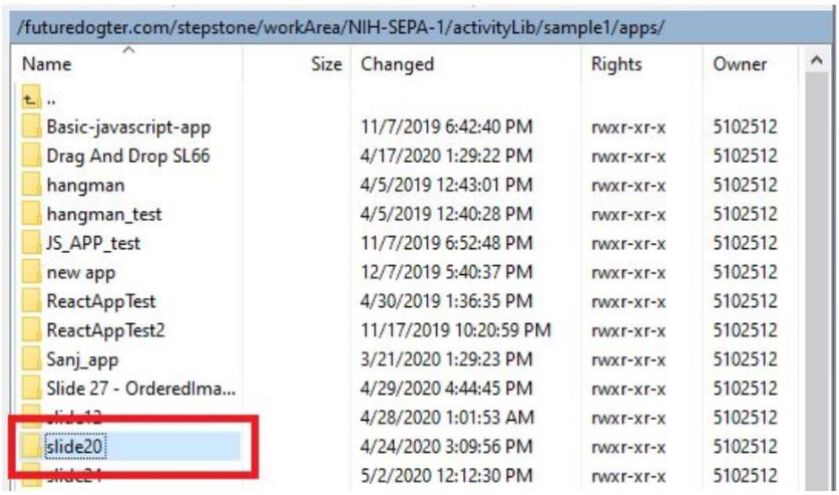
3. Once you are connected, you should see a directory that looks something similar to this but it should always contain a *futuredogter.com* directory.



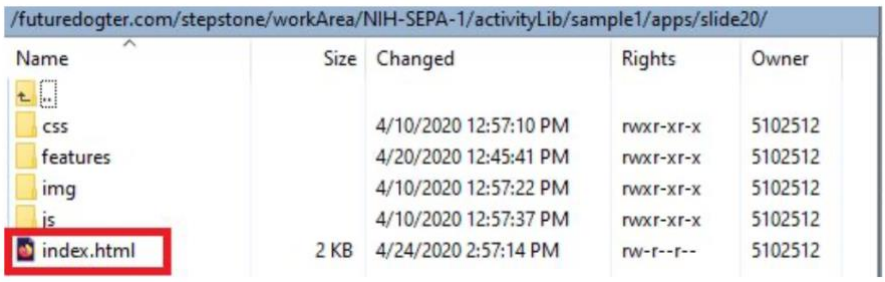
Now, navigate to the following path:

futuredogter.com/stepstone/workArea/NIH-SEPA-1/activityLib/sample1

4. This directory should contain three subdirectories titled “apps”, “json”, and “media”. Whenever you want to test out your standalone mini-app, upload the entire app contents into the /apps directory as a subdirectory.



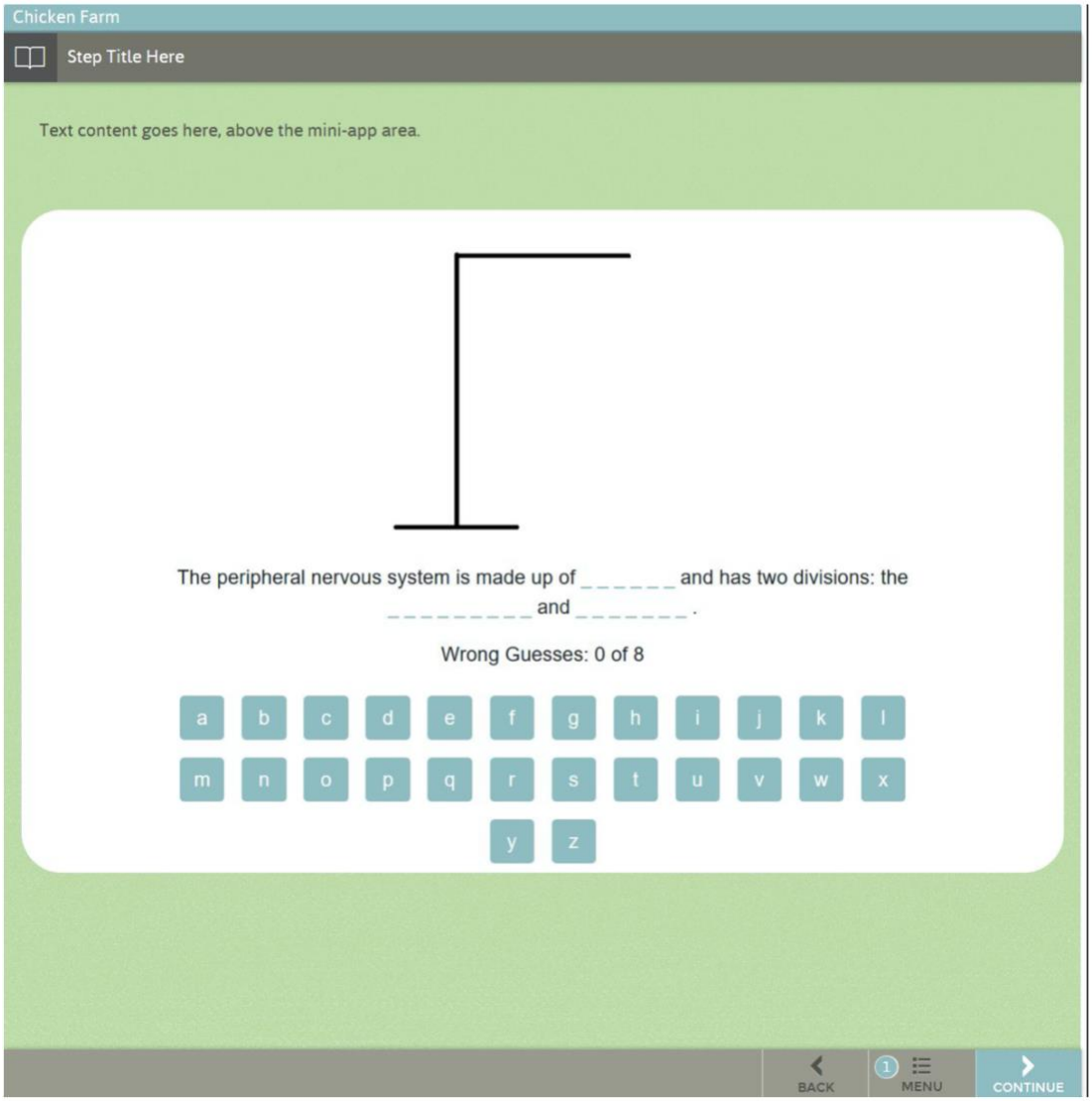
Be sure your specific app folder contains an index.html file at the root as well as any necessary JavaScript, CSS, and even other HTML files. It has been advised by Daniel that you shouldn’t try to pull scripts externally, meaning any scripts you need to include for your app should be downloaded and included in your app’s folder.



5. Once you’ve uploaded your app’s folder into the */apps* directory, go into the */json* directory. You should see multiple JSON files, one of which is titled *draggybox40.json*. To view your app in a StepStone testing environment, open *draggybox40.json*, and edit the launched app to be the title of your app’s folder. Remember, this folder must have an *index.html* file at the root.

6. Now you're ready to view your app in the testing environment. Visit the following link: http://www.futuredogter.com/stepstone/playerShell.php?org=CET&sys=public.Latest&pool=TAMU-CET-1&resourceloc=www.futuredogter.com&resourceavatar=NIH-S EPA-1&resource=sample1&ppj=1_1_40

7. At that link, you should see your app embedded into the frame. You can disregard the titles and whatnot, as your app has no control over the page and is only responsible for itself. You can use this testing environment to repeatedly launch your app and make small adjustments so that Daniel doesn't have to reapply your app to production numerous times. Note that we had some loading and responsiveness issues in this environment when switching launched apps, so be sure to clear your browser's cache if you experience these types of issues.



c. Tutorial 3: Editing The StepStone Production Path

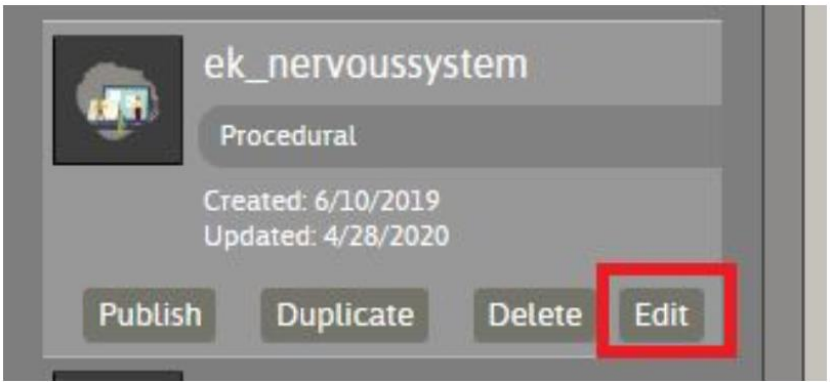
Before you email Daniel to add your apps into the production environment, you first need to edit the live StepStone paths that dictacte where your app is going to go on the slide.

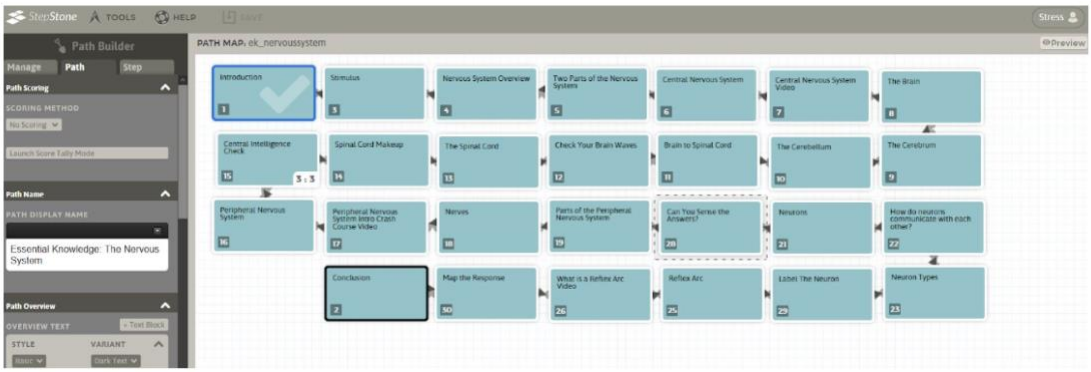
1. Go to the StepStone learning path editing website: https://stepstonelearning.net/

2. Log in with whatever credentials match your project. The account portal for all modules is *TAMU-NIH-1*

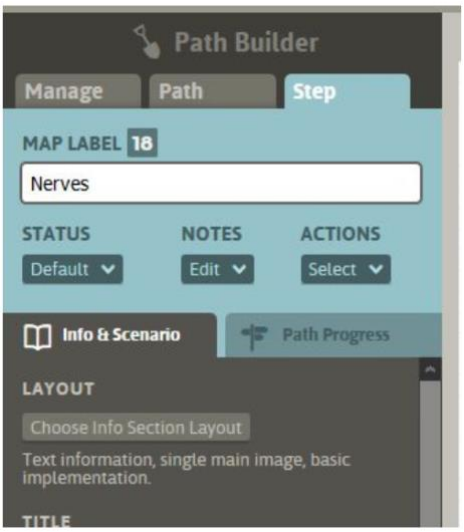
Module Name	Username	Password
Cell Biology	Cells	PEER1
Stress	Stress	PEER2
Ecology	Ecology	PEER3
Clinical Trials	ClinicalTrials	PEER4
Infectious Diseases	InfectiousDiseases	PEER5

3. Once logged in, you should by default be at the Path Builder Tool. For your module, this will contain all of the relevant paths that are used by the module in production. If you choose a path and click ‘Edit’, you will be shown all of the internal steps that the path contains.

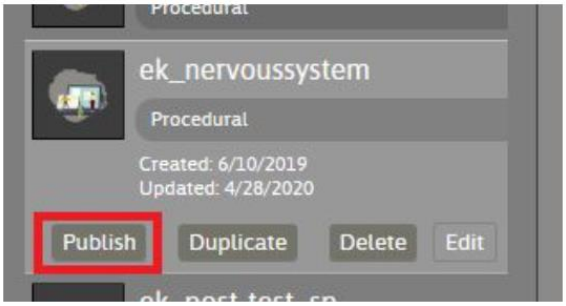




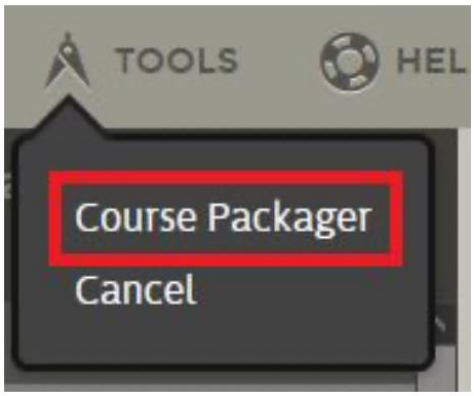
4. Each step, displayed as the blue boxes, is linked to its previous and next steps. To edit a specific step, click the step tab on the left of the page. This will display all of the relevant information (title, text contents, media, etc.) for that specific step. In StepStone, this will be one single slide in a path of the module. Use this feature of the tool to change anything you want about the step your app will go on, from text contents to background. The most likely occurrence will be that you want the slide to be completely empty because your mini-app SHOULD be completely independent and contain all information and interactions necessary on the slide. In that case, you might find it easier to simply delete the step from the path altogether and insert a new, completely empty step in its place.



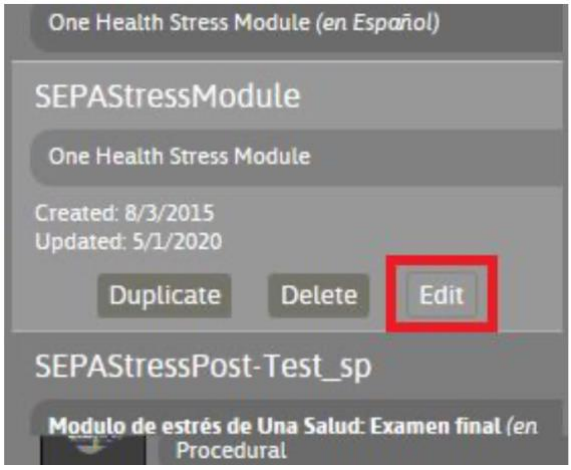
5. Once you have all of your steps edited for the current path and your changes saved (which should be required by the system whenever you switch steps), you need to publish the new path. Go back to the ‘Manage’ tab and click ‘Publish’ for the path you’ve just edited. You will be asked for an ID for your new publication. Give it a unique ID, different from any of the IDs published for this path previously. If you try to name it a non-unique ID, the section will flash orange.



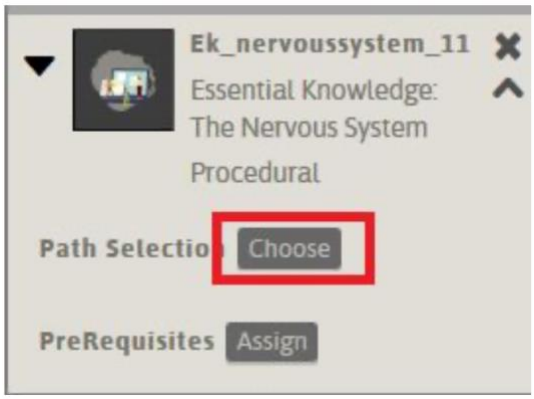
6. With your path published, go to the Course Packager tool.

7. In the list of courses, you've edited paths for, 

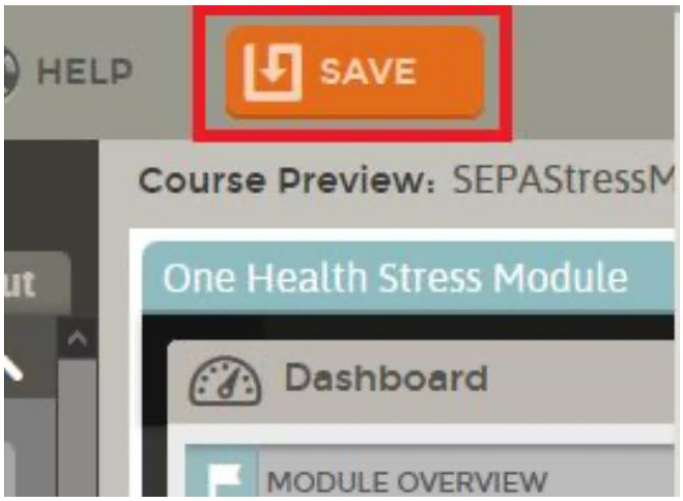
click on the module and click edit.



8. In this course's 'Paths' tab, find the path you made changes to, select its dropdown, and click 'Choose' to select a new path for the course.



9. After selecting the ID of the path you published in Step 5, click the save button at the top of the page to apply this selection to production.



10. Now, you should see whatever changes you've made appear in the live corresponding StepStone module. At this point, you can email Daniel Shuta and tell him that your app is ready to be applied. The general procedure for doing that is:

a. Have your most recent app version uploaded to the apps folder on the FTP server.
b. Tell Daniel where the app should go; we followed the format:
(app directory name) : (path name in StepStone) : (step/slide title)

d. Tutorial 4: Local Automated Testing for JS Apps

Since it is purely low-level HTML, CSS, and JavaScript-based apps, we used development-dependent node modules. If you'd like to set up your own automated testing environment like ours, you can follow the simple steps:

1. Install NodeJS
2. In the root directory for your app, run
   npm init
3. Install the necessary development modules:
   npm install --save-dev cucumber selenium-webdriver chai
   Depending on the browsers you want to test with, you can install the following modules:

Google Chrome:	*chromedriver*
Microsoft Edge:	*edgedriver*
Firefox:	*geckodriver*
Internet Explorer:	*iedriver*

4. It's important to include the */node_modulesin* the gitignore file so that these dependencies are not tracked by GitHub.
5. It is also important to note that the "production" version of your mini app should not include any of the testing files and dependencies
6. Be sure you have the latest version of the given browsers or else Selenium will throw errors.
7. Write your Cucumber features in */features/my_feature.feature*
8. Write your step definitions in */features/support/steps.js*
9. Follow the selenium tutorial: Selenium WebDriver JS API

10. Remember to use *asynca* and *await* in your step definitions/world functions so that your code can finish execution during steps/document manipulation. Here is more documentation for JavaScript Promises and Async/Await
11. In *package.json*, you can write a test script to run with *npm*:

    "scripts": {
    "test": "./node_modules/.bin/cucumber-js
    }

    Then, on your command line at root directory:
    npm run test

12. Valuable References

    Cucumber-js: https://github.com/cucumber/cucumber-js
    Selenium WebDriver: https://www.npmjs.com/package/selenium-webdriver
    https://www.selenium.dev/selenium/docs/api/javascript/index.html
    Chai: https://github.com/chaijs/chai
    Our GitHub (referencing our file structures and code would likely be the most helpful): https://github.com/jinchengli97/Animations-in-Biology-Learning-Content-2

e. Tutorial 5: What is the iframe Resizer?

The iframe resizer is a JavaScript library that is used by StepStone to automatically maximize the height of iframe shells within each slide. The parent StepStone structure already implements one of the crucial library files. Your job when creating your mini-app is to include the second resizer responsible for resizing iframe contents. The JS file you must include is titled "iframeResizer.contentWindow.min.js". This file can be found in all of our apps, generally under the "js" directory. You can also find it at the library's GitHub: https://github.com/davidjbradshaw/iframe-resizer

As per Daniel Shuta: "You can place it anywhere in your mini-app's structure, as long as it is invoked at run time to communicate with the paired parent file (I've been calling it right after the jQuery inclusion in the index.html head code of my mini-apps).
Most importantly, the way you choose to design your app and implement CSS can affect the interaction of the app with the resizer. Be sure to thoroughly test your app using the FTP server and ensure this script is included properly within your app. Again, you can refer to our GitHub for examples, but it's pretty straightforward. Daniel was also incredibly helpful and responsive throughout the semester we worked on our apps.

10. Correspondence

This section represents a compilation of all significant correspondence between our team, Daniel Shuta and Dr. Walker. If you still have unanswered questions, hopefully some of this information will prove useful.

From Dr. Walker, April 7th, 2021

I am listing everyone on GitKrakin and RoughneckCoders here, since you are both working on animations in biology games. This is the top priority of the "real" customer Torri Whitaker. Before our meeting this week, please very thoroughly go through what is there in the system. The first thing she wants is an inventory of what has been deployed and developed. Then we need to develop a lot of new animations across many learning modules, which is where your teams will split up working on different things. This semester I would like to deploy them to the web as well as the StepStone learning system, as they are developed. I am cc'ing Harshita, who is the webmaster for the project.

From Torri Whitaker, April 15th, 2021

I've created/shared a folder in Google Drive that has documentation we created about interactive pieces requested to be added to the Essential Knowledge portion of the One Health curriculum in StepStone. It includes documentation for cells, stress, infectious diseases, and ecology. We will work on adding documentation for clinical trials, genetics, and zoonotic diseases soon.

https://drive.google.com/drive/folders/1Q73anlOae1R_cD3JidCvC26Pa9Zvng82?usp=sharing

Let me know if I can answer any questions or assist in any other way.

From Dr. Walker

Can you open this shared drive and see the animations for the other modules? If not, let me know and I will pull them separately for your teams. Torri calls the animations "interactions".

From Dr. Walker, April 19th, 2021

Please send me a report like this one that GitKrakin did for animations in cell biology. I need to send to Torri Whitaker ASAP.

From Daniel Shuta, April 22nd, 2021

Hello.
I will need to know the module id, path id, and step id (number) where you want this moved in our main hosting area. I will also need to know the folder name of the mini-app you want pushed over.

Examples of what I'll need :

Module - SEPAStressModule

Path - Ek_nervoussystem_13
Step - 13

Mini-app - Scramble

From Daniel Shuta, April 27th, 2021
Hello, yes thank you. You did a great job collecting the info. Thank you for the screenshot, and you were correct -- that number 29 is the step id that I needed.

I would have had this for you yesterday but I found a bug in the StepStone player that was preventing the direct preview from jumping directly to the pertinent Step. I was able to resolve that today and so here is the link to the course itself:

https://stepstonelearning.net/directPortal/playerShell.php?org=CET&sys=public.Latest&pool=TAMU-NIH-1&resource=SEPAStressModule

And here is a direct link to the step with your application on it:

https://stepstonelearning.net/directPortal/playerShell.php?org=CET&sys=public.Latest&pool=TAMU-NIH-1&resource=SEPAStressModule&ppj=1_3_29

If you need to me to reapply an updated version of your application, let me know. it should be easy now that I've repaired the little glitch.

From Daniel Shuta, April 29th, 2021
Direct preview links, corresponding to your ordered list :

#1
https://stepstonelearning.net/directPortal/playerShell.php?org=CET&sys=public.Latest&pool=TAMU-NIH-1&resource=SEPAStressModule&ppj=1_2_17

#2
https://stepstonelearning.net/directPortal/playerShell.php?org=CET&sys=public.Latest&pool=TAMU-NIH-1&resource=SEPAStressModule&ppj=1_1_29

#3
https://stepstonelearning.net/directPortal/playerShell.php?org=CET&sys=public.Latest&pool=TAMU-NIH-1&resource=SEPAStressModule&ppj=1_3_28

#4
https://stepstonelearning.net/directPortal/playerShell.php?org=CET&sys=public.Latest&pool=TAMU-NIH-1&resource=SEPAStressModule&ppj=1_2_16

#5

https://stepstonelearning.net/directPortal/playerShell.php?org=CET&sys=public.Latest&pool=TAMU-NIH-1&resource=SEPAStressModule&ppj=1_1_12

#6
https://stepstonelearning.net/directPortal/playerShell.php?org=CET&sys=public.Latest&pool=TAMU-NIH-1&resource=SEPAStressModule&ppj=1_1_28