

# **Animation in Cell Biology Learning Content I**

## **Final Report**

### **GitKrakin**

#### **Product Owner**

Arjun Singh ([arjunsingh@tamu.edu](mailto:arjunsingh@tamu.edu))

#### **Scrum Master**

Kenta Nakajima ([kamesamakenta@tamu.edu](mailto:kamesamakenta@tamu.edu))

#### **Developers**

Sri Teja Chamarthy ([sritejachamarthy@tamu.edu](mailto:sritejachamarthy@tamu.edu))

Prajwal Das ([prajwaldas95@tamu.edu](mailto:prajwaldas95@tamu.edu))

Lingaraju Vasuraju ([lingaraju@tamu.edu](mailto:lingaraju@tamu.edu))

Ching-Hsien Liu ([issacliu@tamu.edu](mailto:issacliu@tamu.edu))

Devi Sandeep Endluri ([dsandeep97@tamu.edu](mailto:dsandeep97@tamu.edu))

CSCE 606: Software Engineering

Instructor: Dr. Hank Walker

Spring 2021

## Team Roles

Product Owner : Arjun Singh  
Scrum Master : Kenta Nakajima  
Developers : Sri Teja Chamarthi, Prajwal Das, Lingaraju Vasuraju  
Ching-Hsien Liu, Devi Sandeep Endluri

## Relevant Links

Pivotal Tracker	<a href="https://www.pivotaltracker.com/n/projects/2494759">https://www.pivotaltracker.com/n/projects/2494759</a>
GitHub	<a href="https://github.com/VillainJoKeR/BiologyAnimation">https://github.com/VillainJoKeR/BiologyAnimation</a>
Poster and Demo presentation	<a href="https://youtu.be/CkhaF_U_YFM">https://youtu.be/CkhaF_U_YFM</a>
StepStone Deployments	<a href="#">Slide4</a> , <a href="#">Slide8</a> , <a href="#">Slide12</a> , <a href="#">Slide31</a> , <a href="#">Slide41</a> , <a href="#">Slide44</a> , <a href="#">Slide46</a> , <a href="#">Slide53</a> , <a href="#">Slide63</a>
Web Deployments	<a href="#">Slide4</a> , <a href="#">Slide8</a> , <a href="#">Slide12</a> , <a href="#">Slide14</a> , <a href="#">Slide31</a> , <a href="#">Slide41</a> , <a href="#">Slide44</a> , <a href="#">Slide46</a> , <a href="#">Slide53</a> , <a href="#">Slide63</a>  Note: The rest of the slides which are developed by legacy teams have been deployed as well

## Implementation Summary

The customer for this project is the Veterinary Medical & Biomedical Sciences department (VetMed) of Texas A&M University. The main customer need is to develop and integrate interactive animations into the Stepstone learning environment and WordPress-based website to improve the biology learning experience of middle school students. The Cell Biology module consists of a bunch of individual animations spread throughout the content. Our primary focus for this project has been to build and polish the animations based on the prior framework and deploy the existing and polished animations into the StepStone environment and WordPress-based website.

We have initially created an inventory of the existing animations in the Cell Biology module. Then, we looked at the developed animations but yet to be deployed on the StepStone learning environment. Next iteration, we have verified these animations in the test environment and on

the WordPress-based website locally. We are a team of 7 members, and we have divided these animations between us and polished these animations before verifying. We then made the necessary modifications to these polished animations to make them compatible with mobile as well. Further, we have parameterized some of our animations so that these can be reused at multiple places. Then, we verified the animations on multiple mobile devices, Ipad and on multiple browsers, and finally on the StepStone test environment as well as WordPress. In the final iteration, i.e., iteration 2, we have deployed them to StepStone and the website with the help of StepStone developer Daniel Shuta and web-master Harshita.

## User Stories

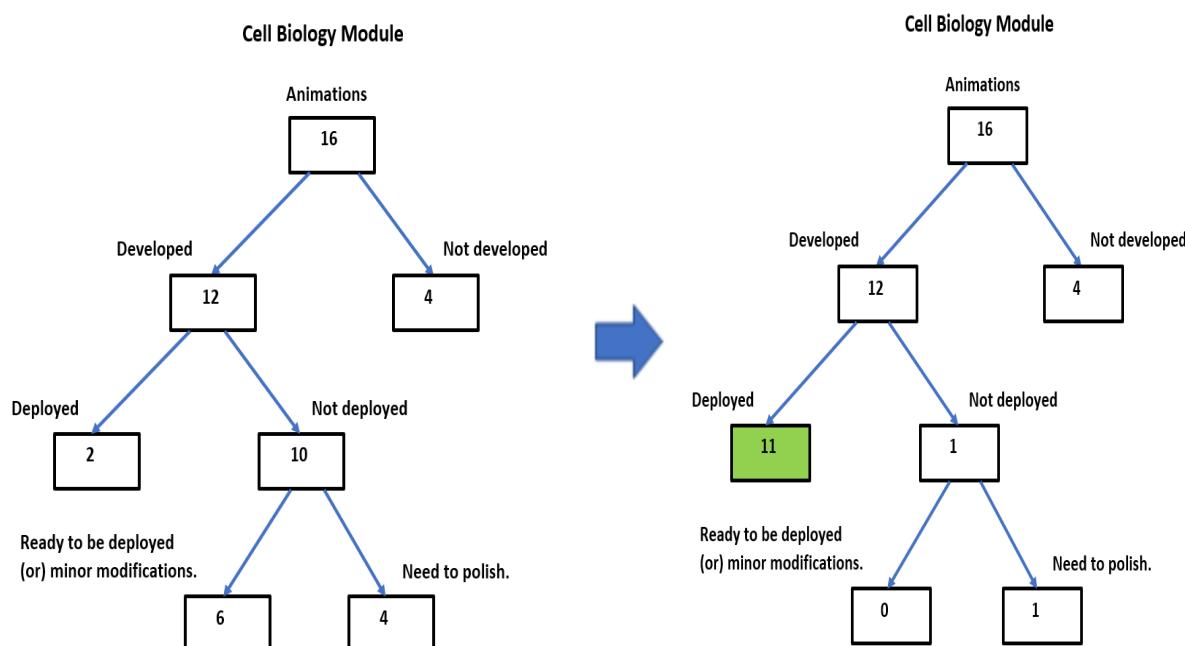
- #### **1. Feature:** Inventory and mapping of the existing animations

**Points:** 3

## *As an Instructor*

*So that I can know the list of the animations developed*

*I want to make an inventory of the existing animations*



We have listed all the animations developed and made an inventory for these. <https://docs.google.com/spreadsheets/d/1U9dx7o4WeZ8uM42JVHVhocqbuY4KdrgK4BCEkZSLTE/edit#gid=0> includes the deployment status of all these animations. Then, we were able to identify that only 2 of the developed animations are deployed, and 6 of the ones are ready to

be deployed with some minor modifications. By the end of Iteration 2, we were able to deploy 9 more animations, as seen in the deployment status above.

## 2. Feature: Enhance the interface of 'Click to reveal answer'

Points: 1

As a student

So that I can learn in an interactive way

I want improvements to the existing interface of 'Click to reveal answer'

And correct implementation of the existing ones.

We found some issues with the 'Click to reveal' slides developed by the legacy team, where the previous interface did not have any instruction in the beginning. We added the instruction 'Click to reveal the answer' so that the user is aware of the next steps. The image and the button were also moved to the centre of the screen and it stays aligned even while resizing the window on the go. This was achieved by placing three breakpoints for different resolutions. We tested it on Wordpress and StepStone. Based on the tests, the slide was deployed.



## 3. Feature: Add matching lines in a picture matching animation

Points: 2

As a Student

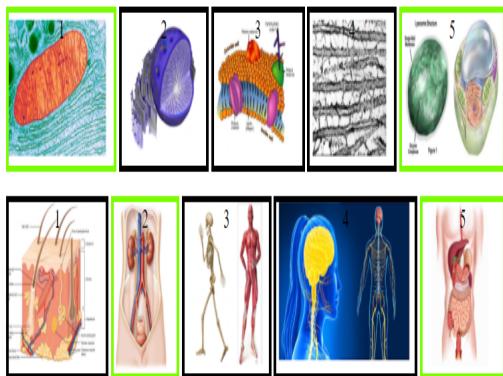
So that I can understand which two terms (or concepts) are related to each other

I want to have a picture matching animation with matching lines

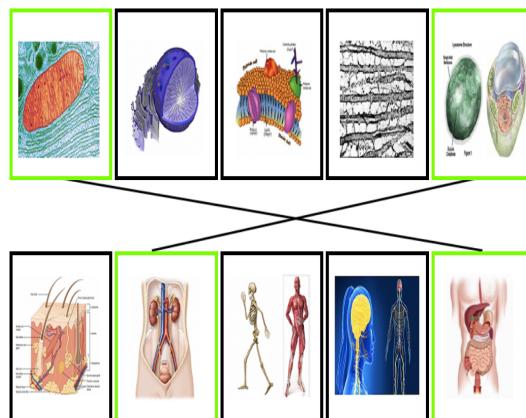
We found few issues with the "MatchMaker" animation developed by the legacy team. If an image on the top matches with the one on the bottom, the border for both images turned green. But, after multiple matches, we would not know the matches since all of the images turn

green. So, we modified this animation with matching lines between the matches to make sure the student sees the matching throughout. This is clearly shown in the diagram above. Further, we have taken steps and refactored the code to make it suitable for parameterization. In the next iteration, we will work on making this animation compatible with other devices (mobile, Ipad), and parameterize the animation so that we can reuse the code elsewhere.

Select one Image from the top and the corresponding answer image to match them



Select one Image from the top and the corresponding answer image to match



#### 4. Feature: Enhance the interface of hangman game

Points: 2

As a student

So that I can test my knowledge correctly and learn in an interactive way

I want improvements to the existing interface of hangman game

And correct implementation of the existing ones

The initial interface of the hangman game had a few issues; firstly, due to the colors used, everything was not clearly discernible; hence we implemented a color combination that makes everything visible for normal vision and people with color blindness. Secondly, Once the game was over, the animation imitating the hangman would disappear, and the game would still allow the user to input guesses. We fixed this issue by disabling the input space, so that once the user has no lives, he or she cannot enter a new guess. Another thing that we noted was that the initial UI allowed students to restart the game whenever they wanted to. So we also disabled the “Play again” button until the end of the game.

For this iteration, we further improved the animations from a mobile compatibility point of view. And we have tested the interface and how it performs on different devices such as Ipad, Iphone, Google Pixel and on different browsers. So based on the tests, the animation is ready to be deployed. This completes the user story for slide 44 (Hangman game).

**PLANT HANGMAN**

WHAT PROCESS TAKES PLACE IN THE CHLOROPLASTS OF A PLANT CELL?

Use the alphabet below to guess the word, or click hint to get a clue.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
u	v	w	x	y	z														

— — — — — Y — — — — —

Game Over

Clue -

[Hint](#)    [Play again](#)




**PLANT HANGMAN**

WHAT PROCESS TAKES PLACE IN THE CHLOROPLASTS OF A PLANT CELL?

Use the alphabet below to guess the word, or click hint to get a clue.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
U	V	W	X	Y	Z														

— — — — — — — — — — —

You have 4 lives

Clue: -

[Hint](#)    [Play again](#)



## 5. Feature: Enhance knowledge check for Cell Biology module

Points: 2

As a student

So that I can test my knowledge of the content covered in the module and learn in an interactive way.

I want improvements to the existing animations

And new animations for the Cell Biology module.

### Slide 8:

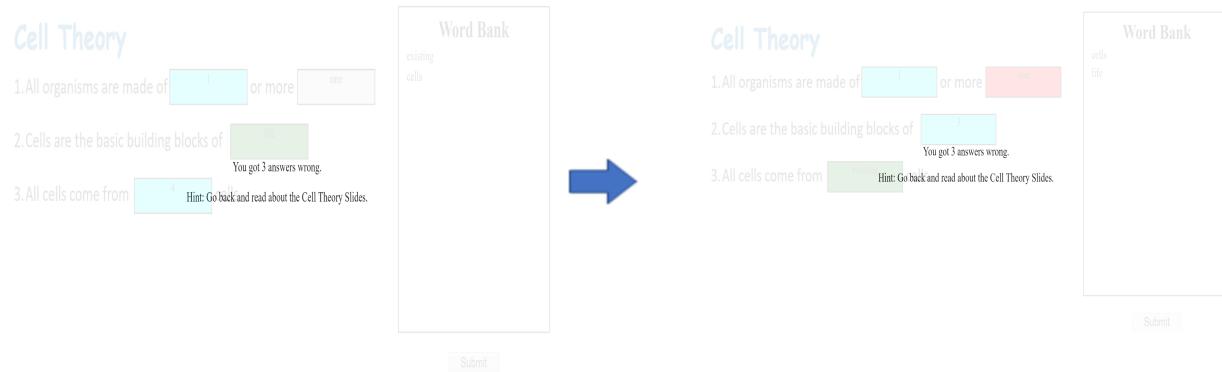
While testing the slide developed by the legacy team, we noticed the following points for improvement-

- Wrong answers were displayed in the grey color

- User was able to change the answer after submission and then resubmit to get more marks.
- The background image of another box was visible behind the Q2 box.
- Drag and drop functionality was not working properly in mobile view.

Below changes were implemented-

- Now wrong answers are being displayed in red boxes, correct in green boxes, and unattempted in cyan.
- Draggable and droppable functionality has been disabled after the submission to remove the hit and try approach to score full marks.
- Now Q2 box covers the background box completely
- The slide is now mobile friendly.



### **Slide 12:**

After testing the slide developed by the previous team, we listed the following points for improvement

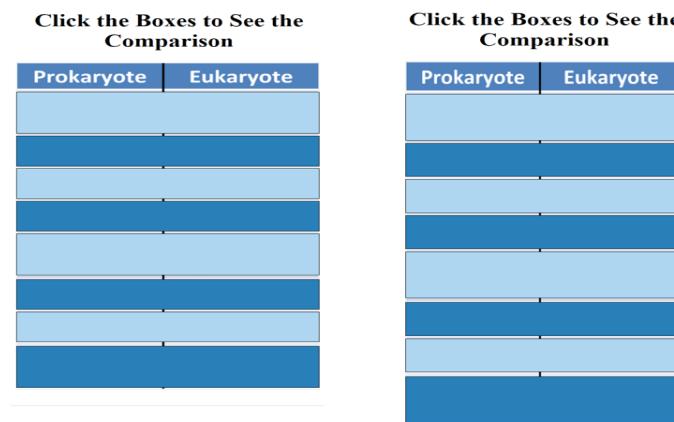
- Mobile compatibility
- Automatically optimized table size in PC
- Need instruction for this page
- Order to the table of contents to be revealed

Based on the test, we improved the animation in several ways below.

- We adjusted the view to be mobile compatible. The view was tested with multiple devices (iPhone 6/7/8/X, iPad, iPad pro, etc)
- The view is updated also for PC so that the aspect ratio is automatically optimized corresponding to the current window size (full-size, half-size, etc)

- The order of the boxes to be revealed is fixed so that students can read the statement from the top
- The simple instruction was added so it is clear for students to do on this page

The following images represent the view from each device (the left for iPhone 6/7/8 and the right for iPhone X, just as an example)



### **Slide 14:**

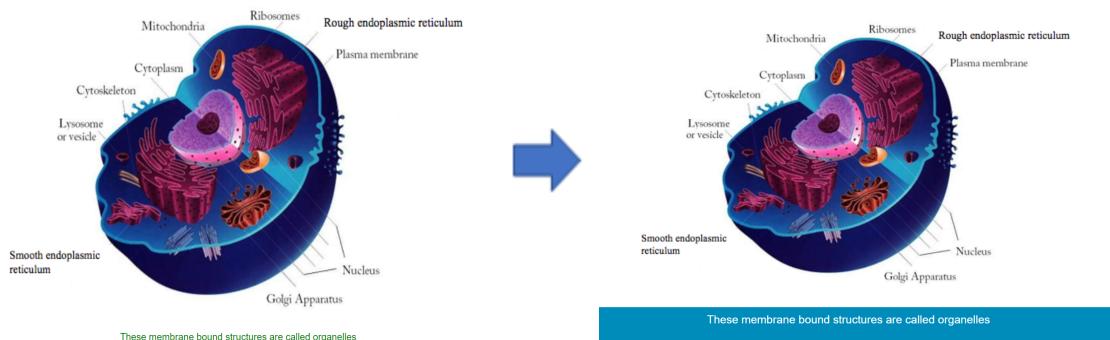
While testing the slide developed by the legacy team, we noticed the following points for improvement:

- The definition was displayed against a white background, which was inconsistent.
- The animation did not work in mobile view.
- The image and the box were not centered.

Below changes were implemented-

- The definition is displayed in the box as expected.
- The image and the button were moved to the center of the screen and it stays aligned even while resizing the window on the go. This was achieved by placing three breakpoints for different resolutions.

We tested it on WordPress and StepStone, and based on the tests, the slide was deployed.



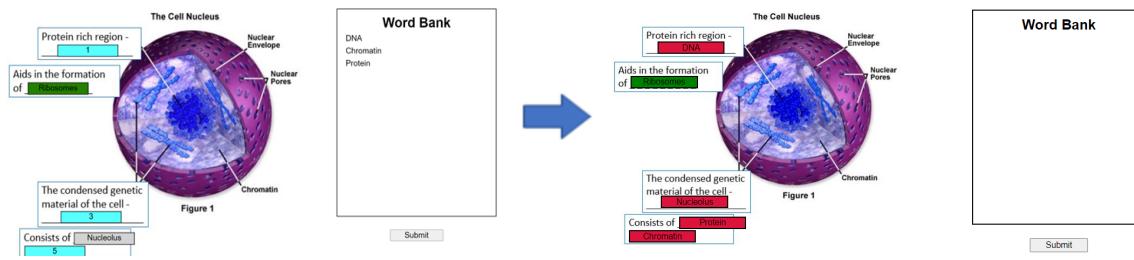
### **Slide 31:**

While testing the slide developed by the legacy team, we noticed the following points for improvement:

- Wrong answers were displayed in the grey color
- The user was able to change the answer after submission and then resubmit to get more marks.
- Drag and drop functionality was not working correctly in mobile view

Below changes were implemented-

- Now wrong answers are being displayed in red boxes, correct in green boxes, and unattempted in cyan.
- Draggable and droppable functionality has been disabled after the submission to remove the hit and try an approach to score full marks.
- The slide is now mobile friendly.
- The module has been parameterized to allow quick implementation in the future.



### **Slide 41:**

While testing the slide developed by the legacy team, we noticed the following points for improvement:

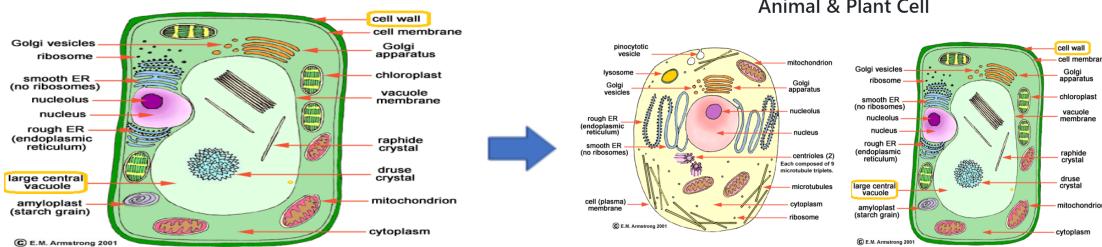
- It was not mobile friendly.
- Users could not see both the images at the same time to compare the differences.
- Users did not know when all the animations were completed.
- The images were hardcoded into the source code and it was hard to reuse the module for similar animations.

Below changes were implemented-

- The slide is now mobile friendly.
- Users can now see both the images and easily comprehend the differences between them.
- Users can now know when the animations are completed.

- The module is now parameterized so that same code can be reused for other similar animations.

### The difference between Animal & Plant Cell



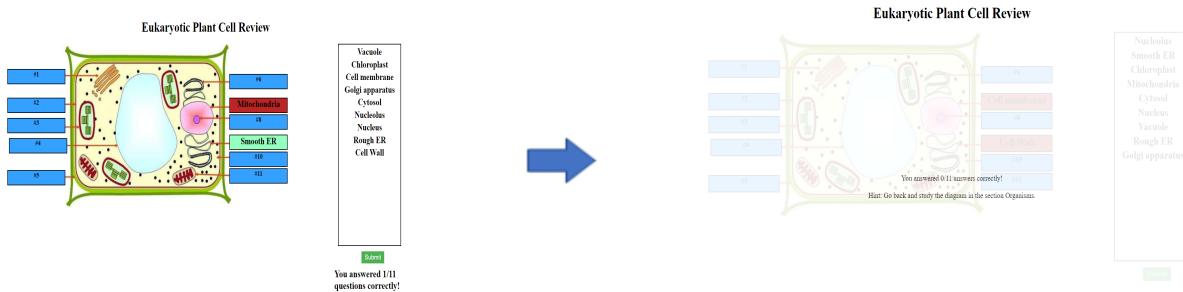
### Slide 46:

While testing the slide developed by the legacy team, we noticed the following points for improvement:

- Hint was not being displayed after clicking on Submit.
- User was able to change the answer after submission and then resubmit to get more marks.
- Drag and drop functionality was not working properly in mobile view
- The options were always displayed in the same order.

Below changes were implemented-

- Hint is displayed on submission in both desktop and mobile views.
- Draggable and droppable functionality has been disabled after the submission to remove the hit and try approach to score full marks.
- The slide is now mobile friendly.
- The options are now parameterized.



### Slide 53:

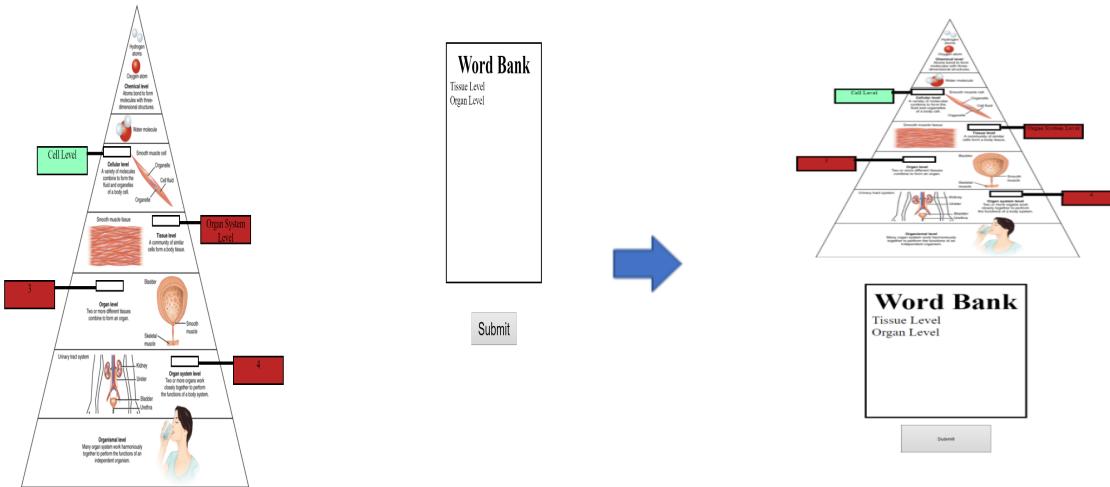
#### Points: 1

While testing we noticed the following points for improvement-

- Wrong answers were displayed in the grey color
- User was able to change the answer after submission and then resubmit to get more marks.
- Animation was getting misaligned while resizing

Below changes were implemented-

- Now wrong answers are being displayed in red boxes, correct in green boxes, and unattempted in cyan.
- Drag and drop functionality has been disabled after the submission to remove the hit and try approach to score full marks.
- Added two breakpoints to avoid misalignment issues while resizing the window and made the slide mobile friendly.



Animation adjusts accordingly after resizing

## 6. Feature: Parameterization of animations

*As a developer*

*So that the animation template can be used for multiple slides*

*I want to parameterize This or That animation*

To make the animation code reusable across other similar animations, we have introduced parameterization of the variable artifacts in the code via a json file. In order to reuse the code,

we just need to modify the json file which is easily comprehensible. As a result, the complexities of modifying the source code are avoided.

## 7. Feature: making animations mobile compatible

*As an Instructor*

*So that students can access the material on mobile*

*I want to make the animations and games mobile-compatible*

To make the animations compatible with mobile devices, a viewport meta tag is used which instructs the browser to use the actual device width with a scaling factor of 1. Then, we have used media queries to deliver different CSS styles to different mobile screen sizes. Most of the time, relative widths are used to avoid issues with resizing or different screen sizes.

## 8. Feature: Make animations StepStone-compatible

We faced some issues while deploying the animations in the StepStone environment. The following are the issues observed and are described in detail in the “Issues” section.

- Animation height problems
- Issues with JSON file for parameterization
- Issues with hyperlinks for jquery

## 9. Feature: Deployment of animations to Stepstone and WordPress based website

*As a developer*

*So that the students can access the animations in StepStone and WordPress site*

*I want to deploy the animations on the StepStone system and Vetmed WordPress web site*

Since we do not have access to the production server ourselves, In this user story, we deployed our working apps to the production environment with the help of Danial Shuta. For deploying a mini-app into a production environment, we need to provide the following information to Daniel:

- **Module ID:** SEPACellBio
- **Path ID:** “ek\_cellth” [*This is the sub-module information.*]
- **Step ID:** 45 [*Step ID (or Slide ID)*]

## Parameterized animations

### Animal vs Plant Cell (Slide 41):

We have parameterized two things regarding comparing differences between two images. Both of them can be found in the json file parameters.json. Firstly, we have removed the hardcoded of the images in the javascript source code so that any two images can be compared. To compare any two images, the json keys **images1** and **image2** should be modified. Also, the differences (which are shown by images) can also be easily modified so as to show new images for the same differences. Secondly, we have removed hardcoded of the ‘number of differences’ in the javascript source code so that showing/removing any number of additional/existing differences can be done directly in the json file without the need for touching the javascript source code. To achieve this, all we need to do is add/remove elements from the **differences** array to add/remove new/existing differences.

```
1  {
2      "image1": "animal_cell-1.png",
3      "image2": "plant_cell-original.png",
4      "differences": [
5          {
6              "image": "plant_cell-1.png"
7          },
8          {
9              "image": "plant_cell-2.png"
10         },
11         {
12             "image": "plant_cell-3.png"
13         }
14     ]
15 }
```

### Cell Theory(Slide 8):

We have parameterized the input of answers and made their order randomized. Initially the answers were hard coded in the javascript file, which made editing/adding new answers a bit difficult and required modifying the source code. Now the answers are saved in the parameters.js file where the answers can be changed/added. From there the answers are always fetched in random order so that users can't just remember the order and answer the questions. Making the answers parameterized has also removed the limit on number of answers, as now the textboxes are created according to the number entries in the js file.

## Cell Theory

1. All organisms are made of 1 or more 2
2. Cells are the basic building blocks of 3
3. All cells come from 4 cells

### Word Bank

one  
cells  
existing  
life

```
1 var data = [  
2 {  
3   "id": "a1",  
4   "value": "one"  
5 },  
6 {  
7   "id": "a2",  
8   "value": "cells"  
9 },  
10 {  
11   "id": "a3",  
12   "value": "life"  
13 },  
14 {  
15   "id": "a4",  
16   "value": "existing"  
17 }  
18 ];  
19  
20  
21 ];
```



## Cell Theory

1. All organisms are made of 1 or more 2
2. Cells are the basic building blocks of 3
3. All cells come from 4 cells

### Word Bank

cells  
one  
life  
existing

## Cell Structure (Slide 31):

We have parameterized the input of answers and made their order randomized. The legacy code hardcoded the answers in the javascript file, which made editing/adding new answers a bit difficult and required modifying the source code. Now the answers are saved in the contents.js file where the answers can be changed/added.

```

1  var questions = [
2    {
3      "question id": "1",
4      "answer id": "a1",
5      "answer": "Nucleolus"
6    },
7    {
8      "question id": "2",
9      "answer id": "a2",
10     "answer": "Ribosomes"
11   },
12   {
13     "question id": "3",
14     "answer id": "a3",
15     "answer": "Chromatin"
16   },
17   {
18     "question id": "4",
19     "answer id": "a4",
20     "answer": "DNA"
21   },
22   {
23     "question id": "5",
24     "answer id": "a5",
25     "answer": "Protein"
26   }
27 ];
28
29 ]
30 ];
31 ];

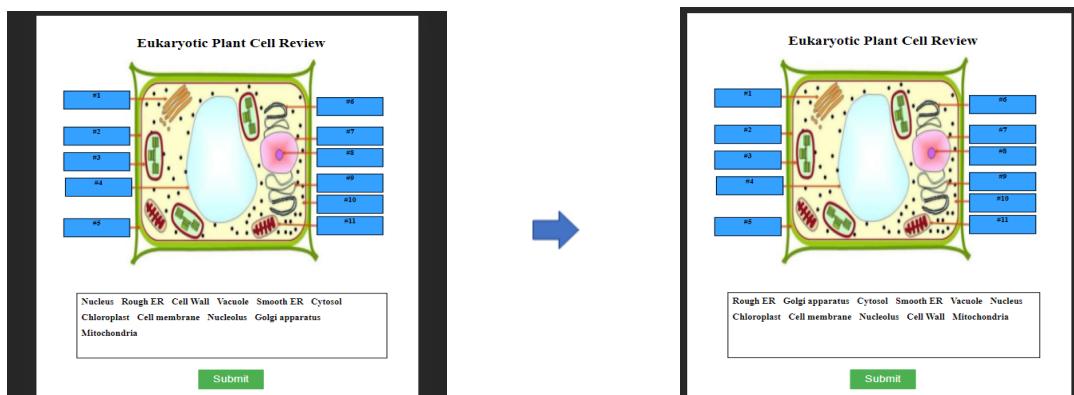
```

### Eukaryotic Plant Cell Review(Slide 46):

The previous version of the code didn't allow for code reuse (if necessary) across different modules. So it would hamper development time if a similar module were to be developed or if identical bugs were to be fixed. It would also not allow instructors to change options(or add more differences) if necessary without modifying the source code.

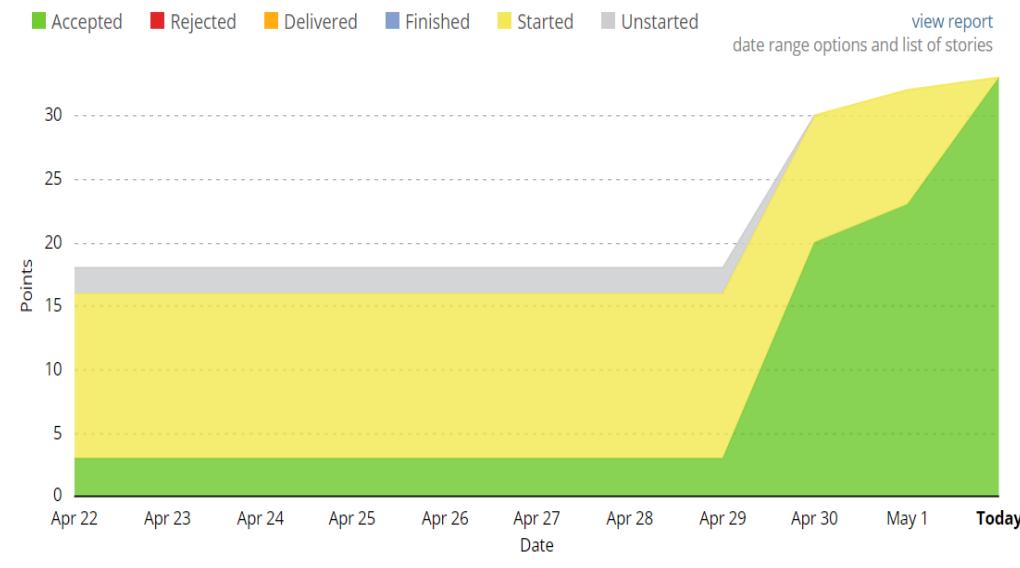
So, a **js** file was created to make the code more reusable. It will also help making updates to the options or adding additional options simple and straightforward.

Additionally, we have also randomized the display of these options for the user. Every time the user opens this slide the options are presented in a different order. This helps prevent blind memorization of the order of the choices.



## Iteration summaries

### Velocity graph



Iteration 0	Iteration 1	Iteration 2	Post Iteration 2
<ul style="list-style-type: none"><li>• Getting familiar with legacy code</li><li>• Inventory of developed animations and their deployment status</li></ul>	<ul style="list-style-type: none"><li>• Setting up WordPress locally</li><li>• Verification of animations in StepStone and WordPress</li><li>• Modifications to fix issues in existing animations</li></ul>	<ul style="list-style-type: none"><li>• Parameterizing animations</li><li>• Making animations mobile-compatible</li></ul>	<ul style="list-style-type: none"><li>• Deployment of animations in StepStone and WordPress based web site</li></ul>

### Iteration 0:

After our first meeting with our customer, we worked on clearly stating the user stories that we identified. Next, we created a record of what animations have been implemented and which ones need improvements. Next, we devised a strategy for the legacy code improvement with improvements such as - Parameterization, mobile compatibility, improvements in UI, etc. In conclusion, we laid out a clear plan to successfully complete the project and deliver based on expectations.

**Iteration 1:**

As for the second iteration we presented our plan to the customer and got the approval on different animations that we intended to do. For this iteration, we created the user stories in the Pivotal tracker to track our progress and we began implementing preliminary improvements. We created several design diagrams highlighting the changes we had made, to show the customer our progress in terms of design.

One of our user stories was to deploy the animations on WordPress and the Stepstone server, therefore we created a detailed tutorial to test our animation on WordPress to allow for proper testing of our animations before we deploy.

**Iteration 2:**

In this iteration, we completed all the user stories and made several changes based on our customer meeting. Firstly we made all animations mobile compatible and rigorously tested all of them on both stepstone and WordPress and secondly, we parameterized some of the animations to make it easier to implement changes in the future. We then asked Dan Shuta and Harshita Chaudhary to assist us in deploying the animation.

**Customer Meeting Summaries**

Iteration	Meeting Date	Customer Meeting Summary
Iteration 0	13th April, 2021	<ul style="list-style-type: none"><li>• Create inventory</li><li>• Contact Dan and Harshita to deploy in stepstone and WordPress</li><li>• Identify existing and deployed animations</li><li>• Create a strategy to implement and modify animations</li></ul>
Iteration 1	20th April, 2021	<ul style="list-style-type: none"><li>• Identified changes and improvements in animations</li><li>• Create a tutorial on testing</li><li>• How to parameterize the animations</li></ul>
Iteration 2	27th April, 2021	<ul style="list-style-type: none"><li>• Testing on both WordPress and stepstone</li><li>• Mobile compatibility and final review of designs</li></ul>

## **BDD/TDD Process**

### **Manual Testing:**

To test the apps on browsers and mobile phones, we can just open the HTML files in the source code from the browser. We have manually validated the formatting of the newly added content and modified legacy content on multiple browsers such as Chrome, Firefox, and Safari. We have also tested the same on Android and iOS-based mobile devices.

### **Mobile/Ipad testing from browser:**

To test the apps on Mobile and Ipad screens, we used the “inspect element” option on chrome. Chrome gives the option of choosing the screens of a number of devices including Android devices such as Moto G4, Pixel 2, etc as well as iPhone models, IPad, and IPad pro. We can also see how the animation would work in different orientations of these screens. Using this method, we verified the functionality of our apps on mobile devices.

### **Testing on StepStone server:**

For testing our applications, we deployed our applications to the stepstone testing environment and then launched the stepstone test server to verify the layout and functionality of the applications. We followed the steps outlined by one of the previous teams to deploy our code to the testing server. This involved copying our application folder onto the test server and then modifying the configuration file to point to our folder, after which the application can be accessed in the stepstone environment by going to the following [URL](#). The detailed steps for testing can be found in the tutorials of the previous team's report [here](#).

### **Testing on WordPress:**

Initially, we reached out to Harshitha and confirmed the compatibility of our tech stack with the WordPress website. For testing the applications in WordPress, we set up the environment locally. We installed the Desktop server application which creates a WordPress server on which we ran all of our animations after testing them locally. The steps to install and host the server has been mentioned under the tutorials section. After successfully testing all the animations we sent out the link of our GitHub Repository to Harshita to deploy the project to WordPress. All the animations were deployed.

## **Configuration Management**

### **GitHub:**

We have used GitHub as a configuration management tool for the project. The changes are not merged directly onto the master branch. Instead, every team member created a new branch to upload their changes. Then he/she raised a pull request, which is reviewed by other team members before merging the change into the master branch. After every iteration, we have created a release tag and in total we had 9 pull requests over the course of 2 releases.

### **Issues:**

#### **1. Pivotal tracker**

Pivotal Tracker is a project planning tool that helps software developers determine how long a feature will take depending on the team's current performance. It assigns different states to the task being tracked and a developer can update the state once he has made progress on it. One key issue with our project was the deployment state. We were able to finish many user stories in our earlier iterations but those stories couldn't get deployed early due to misunderstanding. Moreover, the way the deployment is set up for our project, it makes us dependent on someone outside the team as well to deploy things on production. Owing to these issues, our deployment of user stories happened in the 4th iterations and this made all our tasks pending. This is the reason why we have no points in our initial 3 iterations and 23 in the 4th (last) iteration and the plot of the velocity looks skewed at the end.

#### **2. Mobile compatibility**

Touch (i.e., drag and drop) code doesn't work for mobile devices. To make this work, we have included the touch-punch.js jquery module. To make the animations compatible with mobile devices, a viewport meta tag is used which instructs the browser to use the actual device width with a scaling factor of 1. Then, we have used media queries to deliver different CSS styles to different mobile screen sizes. Most of the time, relative widths are used to avoid issues with resizing or different screen sizes.

#### **3. Stepstone issues**

##### **➤ Animation height problems**

Mini-apps are placed inside the StepStone environment in a customized "iframe" element. Therefore, the apps had to be resized using an iframe resizer. If this isn't

triggered properly, the app is incompatible with StepStone. Also, using relative widths for the HTML page didn't work well with StepStone. Therefore, the height of the HTML, body, and/or the main container is changed to use a fixed value (600px or 800px for example.).

➤ Issues with JSON file for parameterization

There is a CORS issue when json is used for parameterization. So, we have modified the parameterization approach to use a js file instead of json.

✖ Access to XMLHttpRequest at '[file:///D:/Spring2021/Coursework/SE-pro\\_index.html:1](file:///D:/Spring2021/Coursework/SE-pro_index.html:1)' from origin 'null' has been blocked by CORS policy: Cross origin requests are only supported for protocol schemes: http, data, chrome, chrome-extension, chrome-untrusted, https.

➤ Issues with hyperlinks for jquery

We see the below error when we use HTTP (insecure) links for jQuery. StepStone expects HTTPS links instead of HTTP.

✖ Mixed Content: The page at '[https://futuredogter.com/stepstone/playerShell.php?org=CET&sys=public.Lates...www.futuredogter.com&resourceavatar=NIH-SEPA-1&resource=sample1&ppj=1\\_1\\_40](https://futuredogter.com/stepstone/playerShell.php?org=CET&sys=public.Lates...www.futuredogter.com&resourceavatar=NIH-SEPA-1&resource=sample1&ppj=1_1_40)' was loaded over HTTPS, but requested an insecure script '<http://code.jquery.com/jquery-1.9.1.min.js>'. This request has been blocked; the content must be served over HTTPS.

#### 4. WordPress deployment issues

Similar to the issue we have seen above for the StepStone environment, WordPress also expects HTTPS links instead of HTTP. So, we had to modify the HTTP links for jQuery to HTTPS.

#### 5. Deployment issues

There is no way for us to upload our apps into the production environment. StepStone developer Daniel Shuta would do this. We ensured that we tested the apps on a test framework simulating StepStone, but still we faced some height-related issues on the production server due to differences in the way the iframe resizer interacts with the mini-app. Also creating a new version of the module, with new instances of some paths does not include the steps already added and Daniel Shuta had to manually create these again to retain apps, nor existing mini-apps.

## Tutorials

### Setting up WordPress locally

#### Manual Installation of DesktopServer on Windows

1. First, download the installation package for your platform from [here](#)
2. Unzip the .zip file and locate the “libs” sub-folder containing the xampplite folder.
3. Drag the xampplite folder to your c:\ hard drive.
4. Open the Windows command prompt by pressing the Windows button on your keyboard or clicking the “Start” button and typing “Command Prompt”, followed by pressing the enter key.

Type the following (on one line) in the Command Prompt window, followed by pressing the enter key:

```
cd /D c:\xampplite\ds-plugins&..\all\bin\unzip ds-cli-win.zip
```

Type the next command, followed by pressing the enter key:

```
cd /D c:\xampplite\mysql&..\all\bin\unzip backup.zip
```

You may now close the Windows Command Prompt window or type “exit”.

5. Double-click the DesktopServer program icon from within your c:\xampplite

#### Steps for Manual Installation of DesktopServer on a Mac

1. First, download the installation package for your platform from [here](#)
2. Unzip the .zip file and locate the “libs” sub-folder containing the XAMPP folder.
3. Drag the XAMPP folder to your Applications folder.
4. Double click the /Applications/XAMPP/ds-plugins/ds-cli-mac.zip to unpack the ds-cli plugin. You may delete the ds-cli-mac.zip file.
5. Double click the /Applications/XAMPP/xamppfiles/var/backup.zip to unpack the MySQL folder.
6. Right-click the DesktopServer application icon from within your /Applications/XAMPP folder and select “Open” from the pop-up context menu.

Open the Desktop server and select the option ‘Create a new development website’.



Then you will see the below window where you have given the site name and the local address where you want to save the site root.



Click on create to start the webserver. To test any module, place your files/folder in the root directory and open the site in the browser.

For example <http://www.example.dev.cc/slides8/index.html>

File location- C:\Users\prajw\OneDrive\Sites\www.example.dev.cc\Slide4\index.html

## **Acknowledgments**

*Dr. Hank Walker*, POC/Customer for the Project, for his help and guidance throughout the project.

*Shravani Shridhar*, TA for the course CSCE 606, for her help and guidance during this project

*Daniel Shuta*, StepStone developer, for deploying the animations in the StepStone production environment.

*Harshita*, Student worker for StepStone to the website, for deploying the animations on the VetMed WordPress website.