Virtual Biology Lab: Mitosis
Technical Documentation

Group 11: Steve Pak, Marvin Liu, Anthony Porturas, Brandon Lum, Jeffrey Gillen, Evan Chipps

Python Functions

startPage()
Represents the start page of the simulation. The simulation can be started from here.

instructionPage(*s*, *f*, *args = ()*)
Represents the instruction pages that the student shall read before starting the lab simulation. It provides information on what to do and how to do it.

s must be a string and represents the message that will show up on the instruction page. **f** must be a function and represents the next instruction that the page will be brought to. **args** represents the arguments of the next instruction.

nextInstruction(*s*, *f*, *args = ()*)
Represents the next instruction page to be brought to.

s must be a string and represents the message that will show up on the instruction page. **f** must be a function and represents the next instruction that the page will be brought to. **args** represents the arguments of the next instruction.

onionRoot2()
Represents the second onion root page that deals with the characterizing of an onion root. Specifically it will ask you to determine a specific region. This function will not be reached unless **onionRoot1()** successfully brings you to this function.

onionRoot3()
Represents the third onion root page that deals with the characterizing of an onion root. Specifically it will ask you to determine a specific region. This function will not be reached unless **onionRoot2()** successfully brings you to this function.

onionRoot4()
Represents the page that tells you if you correctly determined the regions of the onion root. This function will not be reached unless **onionRoot3()** successfully brings you to this function. This page will be able to bring you to the next set of instructions and next section of the simulation.

prepareMic1()
Represents the page that tells you to correctly identify objects that are needed for the lab.

prepareMic2()
Represents the page that tells you to correctly identify objects that are needed for the lab. This function will not be reached unless **prepareMic1()** successfully brings you to this function.

prepareMic3()
Represents the page that tells you to correctly identify objects that are needed for the lab. This function will not be reached unless **prepareMic2()** successfully brings you to this function.

prepareMic4()
Represents the page that tells you to correctly identify objects that are needed for the lab. This function will not be reached unless **prepareMic3()** successfully brings you to this function.

prepareMic5()
Represents the page that tells you if you correctly indentified all of the objects correctly. This function will not be reached unless **prepareMic4()** successfully brings you to this function. This page will be able to bring you to the next set of instructions and next section of the simulation.

mitosis1()
Represents the page that tells you to correctly identify the stages of mitosis.

mitosis2()
Represents the page that tells you to correctly identify the stages of mitosis. This function will not be reached

unless **mitosis1()** successfully brings you to this function.

`mitosis3``()`
Represents the page that tells you to correctly identify the stages of mitosis. This function will not be reached unless **mitosis2()** successfully brings you to this function.

`mitosis4``()`
Represents the page that tells you to correctly identify the stages of mitosis. This function will not be reached unless **mitosis3()** successfully brings you to this function.

`mitosis5``()`
Represents the page that tells you to correctly identify the stages of mitosis. This function will not be reached unless **mitosis4()** successfully brings you to this function.

`mitosis6``()`
Represents the page that tells you if you correctly identified the stages of mitosis. This function will not be reached unless **mitosis5()** successfully brings you to this function. This page will be able to bring you to the end of the simulation.

`endPage``()`
Represents the end page telling you your grade and how many mistakes were made.

`getGrade``()`
Returns the grade of the lab simulation

`highlight``(`*RectObj*`)`
Simply puts a highlight of a rectangle over text.

RectObj represents a rectangle object.

`unhighlight``(`*RectObj*`)`
Simply unhighlights a rectangle over text.

RectObj represents a rectangle object.

`feedback``(`*s*, *f*, *font = 36*, *args = ()*`)`
Represents the page that provides feedback based on the answer chosen.

s must be a string and represents the feedback message that will be shown. **f** must be a function and represents the next page that you will be brought to. **font** represents the font of the text on the feedback page. **args** represents the arguments of the next function.

JavaScript and CSS Files

route.js

Takes you to the homepage.

Gruntfile.js

Injects .css and .js files into EJS and HTML files.

User.js

Creates the User model. It holds name, password, position, and email. It also checks to make sure the passwords match before creating a new user.

UserController.js

Holds most of the functions associated with the User model. It creates users, shows the user's profile, lists all the users, allows editing to the user's profile, and also deletes users.

bootstrap.js

This is a default framework made by Twitter. It helps build interface components.

jquery.js

A javascript library to simplify scripting.

jquery.validate.js

This is a default validation file that is downloadable which checks for valid inputs into data fields.

customValidate.js

It overrides the jquery.validate.js file with personal customization.

flash.js

Shows error messages.

policies.js

Makes sure that for access into the user subpage, the user has to be logged in.

local.js

Sets up the use for MongoDB as the database for the website.

SessionController.js

It handles the logging in and out of the website. It searches for the inputted email and compares the inputted password with the one in the database.

isAuthenticated.js

It makes sure the user is authenticated, or logged in.

package.json

Lists the dependencies for the code to run.

index.ejs

The home menu. It displays two buttons: log in and register.

layout.ejs

Creates the general layout of the website (header and footer) using stylesheets.

edit.ejs

It creates "user/edit" page.

user/index.ejs

It holds the layout of the "/user" page. It displays several buttons that link you to their respective functions.

show.ejs

It holds the layout of the "/user/show" page. It displays the user's profile.

user/new.ejs

It creates the registration page. It displays the layout.

session/new.ejs

It creates the sign in page. It displays the layout.

bootstrap.css

This is a default layout for websites that is downloadable.

custom.less

It overrides the bootstrap.css file with personal customization.

HTML Files

index.html

This is the first page in the simulation; contains the log-in option for both the student and professor.

student.html

This page asks the student whether they want to check their grades or move on to the simulation.

prophase.html

This is a page that asks the student to correctly identify the picture that illustrates prophase. Five options in the form of buttons are given, and the end goal is to connect selections to backend grade storage.

metaphase.html

This is a page that asks the student to correctly identify the picture that illustrates metaphase. Five options in the form of buttons are given, and the end goal is to connect selections to backend grade storage.

anaphase.html

This is a page that asks the student to correctly identify the picture that illustrates anaphase. Five options in the form of buttons are given, and the end goal is to connect selections to backend grade storage.

telaphase.html

This is a page that asks the student to correctly identify the picture that illustrates telaphase. Five options in the form of buttons are given, and the end goal is to connect selections to backend grade storage.

interphase.html

This is a page that asks the student to correctly identify the picture that illustrates interphase. Five options in the form of buttons are given, and the end goal is to connect selections to backend grade storage.

simulation.html

A big selling point was the image-mapped biology lab, to give the impression of being inside an actual biology lab. The user can either click on the microscope or the onion sample. If the microscope is clicked on initially, an error message pops up instructing the user to prepare a sample first. Once the onion is clicked on, there are a series of pages where the student has 5 different choices of what to do next.

onion.html

Each of the onion.html pages gives the user five options for which they can choose to take next. For this one the

user is supposed to take off a slice of the onion. A wrong answer will cause a feedback statement showing where the student went astray.

onion2.html

For this one the user is supposed to put the slice of the onion on the slide. A wrong answer will cause a feedback statement showing where the student went astray.

onion3.html

For this one the user is supposed to put blue dye on the sample of the onion. A wrong answer will cause a feedback statement showing where the student went astray.

onion4.html

For this one the user is supposed to put the slide cover on the sample. A wrong answer will cause a feedback statement showing where the student went astray.

onion5.html

For this one the user is supposed to take the entire sample over to the microscope. A wrong answer will cause a feedback statement showing where the student went astray.

microscope.html

There was a plan to develop the simulation highly at the microscopic level; however, at the realization of the incapability to connect to the server side, the efforts were focused on finding a new template to create a simulation with the same goals in mind as the previously outlined simulation. Currently, this page is just a page holder.