

Gene Expression Flash Development Documentation

Overview

In order to create the Flash project for gene expression, several different procedures were necessary to get the program to run effectively. So that others can track what scripts were used, and why they were used, this document was created to facilitate understanding of the scripts used and how they function within this project.

This document will cover the roles played by various developers on this project, as well as the adherence to the design plan. As this is a work in-progress, further steps for development will be covered. This document will then delve into the technical details, scripts, naming conventions, variables, codes, and bugs which are within the Flash project.

Credits will first be given for various helps for this project.

Section I: Project Development Credit and Roles

Credit

The subject matter experts for this project were Greg Podgorski and Debra Lowe. Greg is a Biology Professor here at USU. Debra has a bachelor's degree in biology and taught middle school science for 7 years. The actionscripting was done by Rebecca Clark and Debra Lowe.

Several resources were used for coding:

1. Andrew Walker, Ph.D. was an invaluable resource, as he taught the course and got us going down this road.
2. "Edge" Newsletter, found at:
<http://www.adobe.com/newsletters/edge/april2009/articles/article4/index.html?trackingid=EOCQY> – an excellent Actionsript tutorial assisted in the concept design of the multiple choice quiz section.
3. Adobe ActionScript 3.0 manual, found at:
http://help.adobe.com/en_US/ActionScript/3.0_ProgrammingAS3/WS5b3ccc516d4fbf351e63e3d118a9b8cbfe-7ff7.html .
4. John Jeon, Instructional Designer at FACT and a colleague. He helped as a consultant on some of the touchier parts of the script. Is there anything he doesn't know? John Jeon and Joel Gardner created the concept map using CMap. We re-created it in Flash.
5. Neal Legler, FACT colleague. Acknowledgement should be given for help with formatting this document, and for the work of his partner Erik Hjorten, who created the document we used as a template.

Team Roles

The project was developed by Debra Lowe and Rebecca Clark. Both worked together on many of the parts, although some division of labor occurred as each specialized in different fields. Intense cross-checking and sharing of ideas took place. The project breaks down as follows:

- Debra
 - Introduction
 - Work Plan
 - Project Documentation
 - Web Page
- Rebecca
 - Overall project look and feel
 - Drag and Drop Concept Map Development
 - Multiple Choice Quiz Development

Section II: Adherence to Work Plan and Next Steps

Adherence to Design Plan

This project has followed the specific guidelines given through the instructor of the Biology 1010 course. Because of this, it is nearly identical to the original plan. This project has three main parts:

- Large Draggable Concept Map
- Concept Map Drag and Drop
 - Transcription
 - Initiation
 - Elongation and Termination
- Multiple Choice Quiz Challenge

As was intended, we have completed the first portion of the gene expression process, and will be working on the final portions over the summer. Efforts have been made to streamline the product for ease of use and understanding for the user.

Next Steps

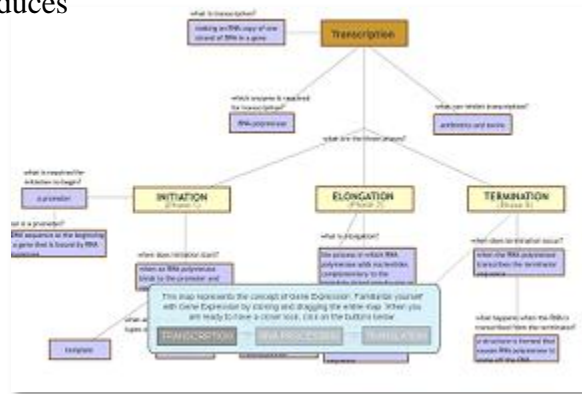
It is our intention to finish building out this learning aid so that it will address the entire subject of gene expression, and not just transcription. Also, in the multiple choice section it would be useful to develop a completion action that occurs when users finish the last question. Right now, when a student responds to the question by clicking on an answer, the correct answer is indicated, whether the chosen answer was correct or incorrect. Also, there is no scoring. It simply asks the user to look at the concept map, and give a good guess. It doesn't keep track or even concern itself with how the student guessed. This may be a good tool for formative evaluation, but does not yet allow the user to gauge improvement in subsequent uses.

Section III: Technical Details

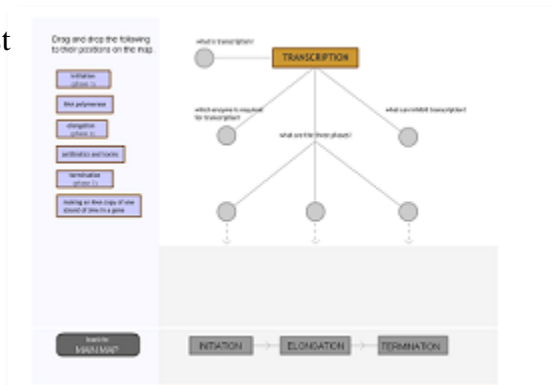
Timeline Structure

The main timeline is organized so that each page is on one frame. It is possible to return to the main frame from each of the others. There are nine keyframes total on the main timeline and they contain the following:

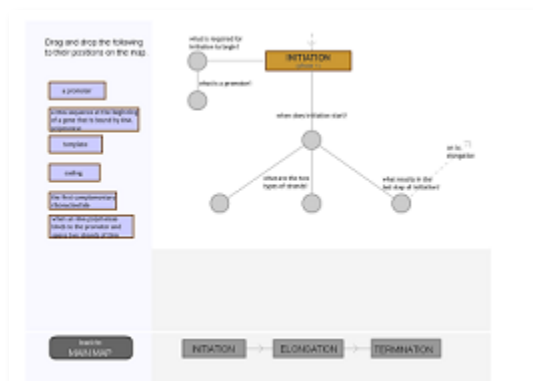
1. A large draggable map that introduces the concept.



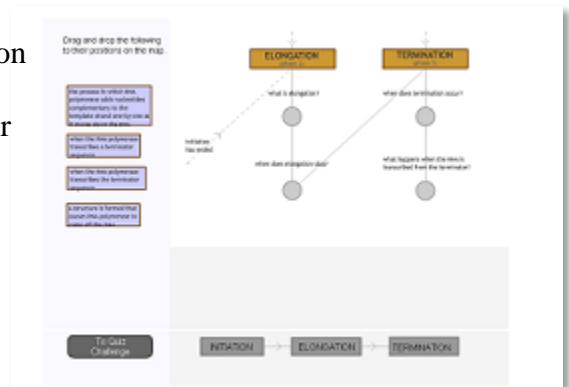
2. A drag and drop build-up of the first tier of Transcription.



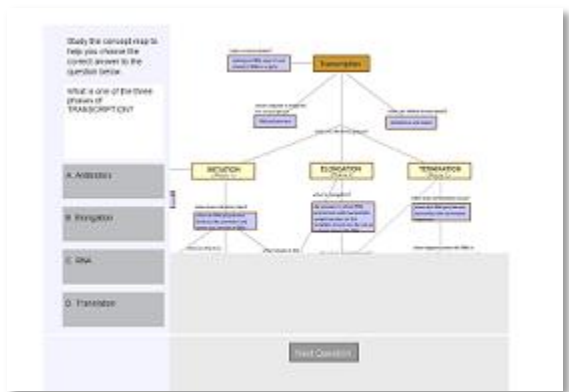
3. A drag and drop build-up of the initiation section of the second tier of Transcription.



4. A drag and drop build-up of the Elongation and Termination section of the second tier of Transcription.



5. Page 1 of the Quiz Challenge.
6. Page 2 of the Quiz Challenge.
7. Page 3 of the Quiz Challenge.
8. Page 4 of the Quiz Challenge.
9. Page 5 of the Quiz Challenge.

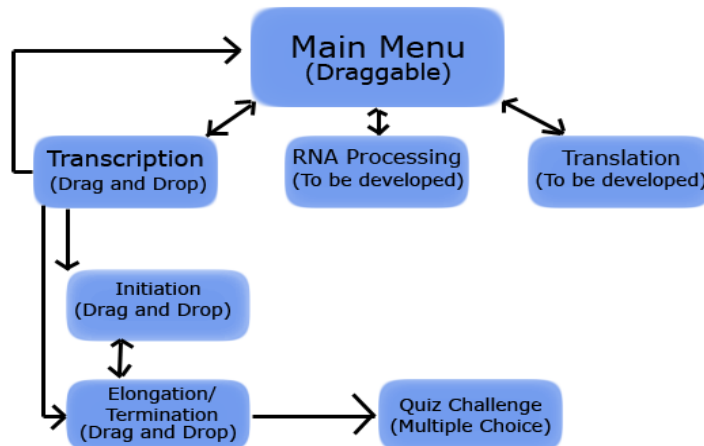


It was our strategy in organizing our module was to have as few key frames on the time line as possible. To do that, we used movie clips where we could.

The various navigation options throughout the module are movie clips. And most of the items that were going to need to be used more than once were converted to movie clips. (There are some exceptions.)

In the timeline, we used layers in a way that would allow us to categorize similar elements together. This was useful when we needed to replicate the background graphics across each section. We selected the graphics, copied and then pasted, as needed.

Navigation through the sections is easily achieved through the use of buttons and prompts. Here is the overall structure of the project:



Naming Conventions

We used the camel hump naming convention, beginning with a lower case word, and then any subsequent words with a capital initial letter.

We named the layers to reflect the items they contained. Movie clips, buttons, and graphics were named to reflect their function. Instances of movie clips were given the name of the movie clip plus “Inst” for instance.

Important Variables

```
var transCounter: Number = 0;
```

```
var transCounter: Number = 0;
```

```
var transCounter: Number = 0;
```

The counter variables were used in the drag and drop section. They were used to count the number of items that had been successfully dropped, so that after the last item had been

dropped, the user received a message that let her know she was done, to move on to the next item.

```
var dBounds:Rectangle = new Rectangle(100, 25, 650, 550);
```

dBounds defined the area that was draggable. The item could not be dragged outside of this rectangle.

```
var startX:Number;
```

```
var startY:Number;
```

These variables kept track of the original positions of the draggable items, so that if they were not dropped in the right place, they would pop back.

```
var activeAlpha:Number = 1;
```

```
var inactiveAlpha:Number = .6;
```

These variables controlled transparency. We used transparency to indicate if something was active or inactive.

```
var correctAnswer = optionB;
```

In the quiz challenge section, this variable was used to indicate which answer was the correct answer. For the first question we declared the variable, and the questions thereafter we assigned it new values.

```
var whatFrame:Number = 1;
```

We used this variable in the quiz section as a reference point for our buttons.

Critical Code Segments

The drag and drop section was difficult to code until I (Rebecca) discovered that the rectangle boundary was for the entire draggable area, and not the drop target. Once I discovered that, the coding became much easier.

Draggable items: the code associated with the draggable items basically used startDrag and stopDrag. We added listeners, then referred to the functions. We defined the rectangle drag boundary and we used a counter to keep track of how many items had been successfully dropped. Navigation is at the bottom of the page so the user can move forward or redo.

Quiz Challenge: Here is the basic flow (this is taken mostly verbatim from the website,

<http://www.adobe.com/newsletters/edge/april2009/articles/article4/index.html?trackingid=E>

OCQY):

1. Display the question in the Question Text Field object
2. Display the four option buttons
3. Display the possible answers in the Answer Text Fields in each button
4. Create the Correct Answer object and set its value based on the correct answer, A, B, C, or D.
5. When the mouse is clicked on one of the four answer buttons, check to see if the clicked button is the correct answer.
6. If the selected button is the correct answer, congratulate the user.
7. If the selected button is not the correct answer, display the correct answer.

(None of the code was provided for this concept so while the logic was not ours, the code came straight from the heart.)

To display the questions in the question text field objects, we created buttons with dynamic text fields so we could change the questions. We did the same with the four answer option buttons. Each has a dynamic text field.

For each new question, the values had to be re-assigned. And, for each question, the coder will pick which option will be correct, and then take special care to make sure the correct answer is located in the position he has chosen.

The code is heavily commented.

Known Bugs

There are only two known bug in this system, and one is fairly minor. In the drag and drop section, it is possible for the user to begin drop the object near the target, but not exactly on it, and it will generate an error.

The other bug occurs when the user is in the initial stages of the drag and drop section. The navigation menu should allow the user to choose to go to Initiation or Elongation or Termination. If the user clicks on Elongation or Termination before Initiation, it will take the user to Initiation anyway.

There are other matters that we would like to resolve. For instance, even though the buttons were created as “non-selectable” the buttons still don’t seem right. They aren’t as readily apparently buttons as we would like them to be.