An Instructional Platform for Technology Concepts

Kevin Ashton

Department of Instructional Psychology & Technology

Brigham Young University

January 2013

**Abstract**

An Instructional Platform for Technology Concepts

Kevin Ashton

Department of Instructional Psychology and Technology

This paper summarizes a design project completed for the Instructional Psychology & Technology Department at Brigham Young University. The purpose of the project was to design and prototype a learning tool geared towards teaching foundational technology concepts. Example content was also generated as part of this project. Development was broken into the five phases of the ADDIE model, namely: analysis, design, development, implementation, and evaluation. The design was approached in a layered manner and development was broken into four sprints. The instructional platform was programmed and tested. The various stages of design and evaluation are explained and discussed in the paper.

**Acknowledgements**

I would like to thank my graduate committee for their continued encouragement in this program. Specifically I am grateful to Peter Rich who has given me the liberty to pursue the topics and ideas that interest me while offering me insight and advice. Finally I am grateful to my brother Scott Ashton who has spent many hours testing and providing feedback on this project.

Table of Contents

[Introduction 6](#_Toc348171873)

[Problem Statement 6](#_Toc348171874)

[Purpose Statement 7](#_Toc348171875)

[Development Model 8](#_Toc348171876)

[Analysis phase 8](#_Toc348171877)

[Target Population Analysis 9](#_Toc348171878)

[Current Material Analysis 10](#_Toc348171879)

[Resource Analysis/Budget 10](#_Toc348171880)

[Design Phase 10](#_Toc348171881)

[Content Layer 11](#_Toc348171882)

[Strategy Layer 12](#_Toc348171883)

[Multiple Instances 12](#_Toc348171884)

[Interactive Visual Feedback 13](#_Toc348171885)

[Concept Maps 14](#_Toc348171886)

[Activating Prior Knowledge 15](#_Toc348171887)

[Message and Control Layers 15](#_Toc348171888)

[Representation Layer 16](#_Toc348171889)

[Media Logic Layer 16](#_Toc348171890)

[Data Management Layer 17](#_Toc348171891)

[Develop Plan 17](#_Toc348171892)

[Prototyping 19](#_Toc348171893)

[Sprint 1 – Concept Map 21](#_Toc348171894)

[Purpose of Sprint 21](#_Toc348171895)

[Design Changes and Development 22](#_Toc348171896)

[Future Development 26](#_Toc348171897)

[Sprint 2 – Gadget Template Plus Actual Gadgets 27](#_Toc348171898)

[Purpose of Sprint 27](#_Toc348171899)

[Design Changes and Development 30](#_Toc348171900)

[Future Development 31](#_Toc348171901)

[Sprint 3 – Content Engine 31](#_Toc348171902)

[Purpose of Sprint 31](#_Toc348171903)

[Design Changes and Development 32](#_Toc348171904)

[Future Development 34](#_Toc348171905)

[Sprint 4 – Content 35](#_Toc348171906)

[Purpose of Sprint 35](#_Toc348171907)

[Design Changes and Development 35](#_Toc348171908)

[Future Development 39](#_Toc348171909)

[Implement Phase 39](#_Toc348171910)

[Evaluation Phase 39](#_Toc348171911)

[Reflection 41](#_Toc348171912)

[References 47](#_Toc348171913)

# Introduction

## Problem Statement

During the course of a career, most instructional designers will be required to work with computer programmers in order to create instructional materials. For this reason most instructional design education programs offer introductory programming classes. Such classes both help students understand currently available technologies used for instruction and equip students with the language and ability to interact effectively with programmers.

One common challenge frequently encountered by instructional designers is learning to adapt to the constantly evolving and changing nature of technology. Simply put, software tools, languages, and programs are often outdated in only a few years, thus leaving the skills and knowledge acquired years previously in the classroom largely irrelevant to the designer. Hence the challenge for educators becomes how to train students not just in the mere usage of current technology tools, but to give students the foundational technological background required to adapt successfully to new programs and technologies that will inevitably emerge during the course of their careers (Gibby, Quiros, Demps, & Liu, 2002). In a case study that examined how we might prepare instructional designers for different career environments Larson and Lockee list both “Knowledge of recent technologies” and “Evaluation of new and existing technologies” (2007, p. 3) as highly desirable areas of preparation. An analysis of recent instructional designer job postings show employers are interested in having their employees knowledgeable in such programs as eLearning, LCMS, MS Office, Captivate, Photoshop, Illustrator, Web authoring tools, Camtasia, Audacity, Virtual worlds, Web 2.0, Dreamweaver, Acrobat, WebEx and a host of other specific technologies (Sugar, Hoard, Brown, & Daniels, 2012). With so many technologies to master, it is difficult for an instructional designer to become highly proficient in each of these programs. Additionally, many of the tools designers will use in the future do not currently exist. Hence it begs the question what should an instructional designer learn in order be proficient with technology tools?

While discussing the complexity of teaching technology literacy, Lin argues that skills with specific applications are necessary “but not sufficient for individuals in the information age”(2000, p. 73). Rather he argues that because information technology changes so quickly “it is impossible to give a fixed, once-and-for-all definition of what an individual must know and understand that will remain current and effective” (p. 73). Therefore, in order to help people to gain “fluency” (or the ability to adapt), instructional material should include foundational concepts that will enable one to “acquire new skills independently after one’s formal education is complete” (p. 73). In essence, Lin talks about teaching the underlying foundational concepts of technology in order to help learners adapt more easily to the continually changing technology landscape. Though specific programs and tools are constantly changing the foundational concepts of technology (FTCs) are relatively constant. For example, an individual who understands the concepts of *styles* as they apply to word processors will be able to more easily adapt to using different word processing programs since the underlying concept is the same. Learning such FTCs is one way by which we can enable students to effectively adapt to the changing nature of technology.

## Purpose Statement

The purpose of this project was to create a learning platform specifically suited for effectivly teaching FTCs, and create sample content for this platform. In order to design this platform, a variety of instructional theories were drawn upon and integrated (Gibbons & Rogers, 2009; Gibbons, 2003; Grissom, McNally, & Naps, 2003; Hunter, 2007; Schooler & Anderson, 1990; Wood, Dykes, Slingsby, & Clarke, 2007). The class selected as the main audience for this tool was Instructional Pyschology and Technology 560, an introductory programming course taught at Brigham Young University. The sample material created focused on the some of the underlying concepts of technology such as HTML (as a reprepstative of markup languages), creating colors through mixing , number systems, and representing text as numbers. These concepts were selected for their widespread usage in a variety of different programming languages and applications.

# Development Model

In order to create effective instructional material, a systematic design and development methodology was employed. A variety of different development models (sometimes called instructional design processes) were considered for this project but ultimately the generic ADDIE model was selected as a process model for its simplicity and adaptability. Molenda describes ADDIE as “an umbrella term that refers to a family of models that share a common underlying structure”(2003, p. 1). ADDIE’s underlying structure is composed of five phases: analyze, design, develop, implement, and evaluate (Gustafson & Branch, 2002). Additionally, during the design phase, I incorporated elements of Gibbons’ and Rogers’ theory of design layers (2009). Each phase in the ADDIE model is explained in the following sections along with the literature and rational for various design decisions.

# Analysis phase

The purpose of this phase is to allow the designer to gather pertinent information that will inform the designer of the instruction. This can include identifying a performance problem in a work or educational settings (Gustafson & Branch, 2002). Typically the analysis done in this phase includes such items as gathering information about the target audience, evaluating currently available instructional materials, considering resource constraints, and scheduling (Peterson, 2003). Deliverables for this phase typically include: target population analysis, resource analysis, proposed schedule, and content analyses analysis.

## Target Population Analysis

The typical student in IP&T 560 is an adult learner enrolled in a graduate program. All students have access to either personal laptops or computers in the IP&T graduate lab. Although students within the class exhibit a wide range of technological and educational backgrounds, all students are familiar with basic computer programs, including spreadsheets and word processors. Since the degree of programming experience varies from little-to-no experience in programming all the way to having experience working in professional software development shops, it becomes difficult to customize the instruction to a single level that would include relevant material for each of the learners. For this reason, students are allowed to adapt their learning materials based on a contract they make with the instructor at the beginning of the semester.

About half of the students are typically masters students while the other half are Ph.D. students. Generally, the master students are training to become practitioners while the Ph.D. students are training to become educational theorists. Whether practice or theory oriented, however, most students take the class with the intent of learning a specific technology– the most popular being web-based technologies such as HTML, CSS, and Flash. Given this interest in web technologies, priority was given to FTCs that were most closely associated with HTML, CSS and other web technologies.

## Current Material Analysis

The class is currently taught in such a manner that each student can select which technology he or she is going to learn. Subsequently, they create a contract with the professors outlining their process for learning the selected technology, and tests they will use to demonstrate proficiency. Students learn primarily from textbooks, online tutorials, video tutorials such as Lynda.com, and classroom lectures. It is worth noting that even though a wide range of technologies are usually selected (i.e., HTML, CSS, PHP, Javascript, Objective C, etc.), each of these technologies share some key concepts. By focusing on the underlying concepts of these technologies, students should be able to learn similar technologies more quickly in the future. Thus, the proposed content of this project stands as a complement to the current class materials and not as a replacement thereof.

## Resource Analysis/Budget

Personal resources for this project consisted of one programmer who was in charge of all aspects of the project including analysis, design, development, implementation, and evaluation. The individual provided programming expertise, server space, bandwidth, and the necessary hours required to code the project. Originally, the individual planned to devote 30 hours a week for 2.5 months. The total estimated cost was approximated at 30 hours x 10 weeks x $16 per hours = $4,800. The final cost came out to be $3,784. The resulting difference was mainly due to hours the resource decided to donate pro bono near the end of the project. There were no reusable resource existing that would reduce the cost of this project by remixing.

# Design Phase

The design phase includes specifying learning activities, selecting instructional strategies, outlining instructional objectives (Gustafson & Branch, 2002). The design phase is informed by the analysis phase since materials generated from the analysis phase inform the selection of instructional methods. While designing I relied heavily on the “layers approach” as described by Gibbons and Rogers. As they explain, “layers represent specialized design sub-problems that result from the decomposition by functionality of whole instructional design problems” (2009, p. 315). By breaking the design into functional layers, I was able to consciously apply different theories and design principles to my actual design. In essence, breaking instructional design into layers makes it easier for the designer to think about specific sub-systems without getting caught up in the details of the overall picture.

## Content Layer

Gibbons and Rogers (2009) describe the content layer which contains an abstract representation of the material to be taught to the learner. The target population analysis revealed that the students in IP&T 560 were typically training to become theorists or designers but not programmers. For these reasons, the content selected focuses on the underlying concepts rather than the detailed implementation of programming structures. In essence it was more important that the students understand the conceptual ideas so they could effectively work with the programmers.

In order to identify FTCs I combed through a variety of programming books at the university library looking for topics that reoccurred across languages. I used introductory books in Java, PHP, JavaScript, and C# as sampling commonly used programming languages. These topics were then grouped together in categories on a large concept map with directional lines linking related topics. The final map contained 152 different topics. From this list six topics were selected. These six topics include the following: the binary number system, pixels, the RGB coloring system, bitmap images, the ASCII character code, and HyperText Markup Language (HTML). These topics were selected for their wide range of applicability in almost any programming language or software program, thus meeting the definition of being a foundational technological concept (FTC).

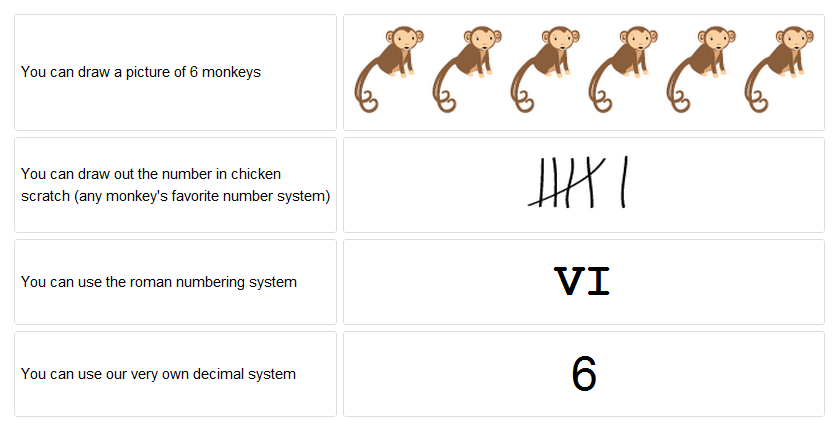
## Strategy Layer

The *strategy layer* describes the methods by which instructional goals are to be accomplished. These include details regarding physical organization of the learning space, learner roles, the sequencing of content, and other purposeful activities engaged in to achieve the instructional goals (Gibbons & Rogers, 2009). The general strategy for this project included demonstrating multiple instances of a concept, providing interactive visual models, using concept maps, and activating students’ prior knowledge. These strategies are discussed in greater detail below.

### Multiple Instances

Students are better able to understand concepts as they see various examples or implementations of a concept. As Hunter points out, “concepts should be presented in a range of forms to help learners discriminate between similar concepts” (2007, p. 14). Demonstrating concepts in various forms helps students distinguish the core essence of a concept from implementation-specific syntax. For this reason, instructional content on the topic pages were deliberately designed so as to provide students with multiple representations of the same basic idea.

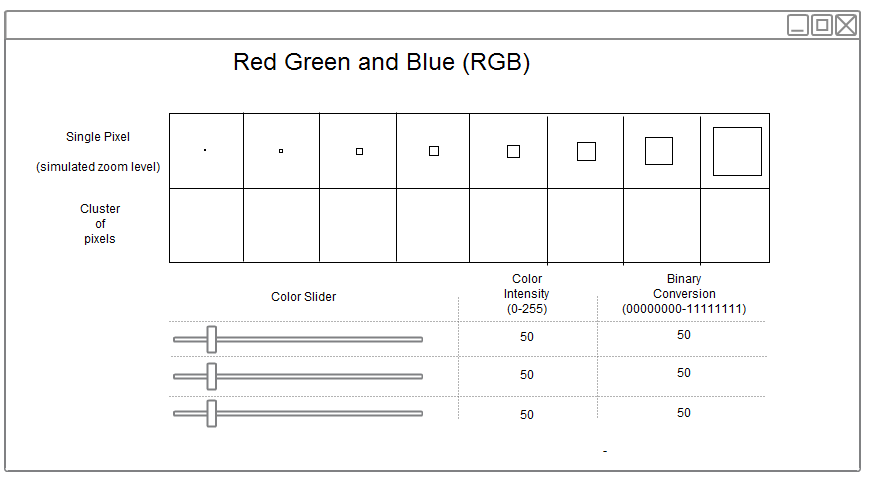
An example of using multiple representations can be seen in *Figure 1* where learners are presented with the basic problem of writing out numbers. A variety of numbering systems are shown including a picture numbering system, Roman Numerals, “chicken scratch”, and the decimal system. By showing a variety of different examples, it is intended that students will be able to abstract out the core concepts that make up a “numbering system.”



*Figure 1 – Example of multiple instances of the same concept*

### Interactive Visual Feedback

Visual aids and pictures greatly enhance the ability of students to understand new concepts. This learning can be greatly enhanced if students are able to actively engage and interact with the visualization (Grissom et al., 2003; Naps et al., 2003). Despite the instructional benefits of visualizations, often the amount of time required to learn how to use them discourages their use(Naps et al., 2003). For this reason, interactive elements were designed with simplistic controls. For example *Figure 2* shows the prototype design for an interactive gadget designed to help learners understand the concept of color on digital displays. The controls are simple sliders that can be used to easily adjust the intensity of red, green, and blue. By using a common controller such as a slider, most users will require no instruction to begin interacting with the widget. It should also be noted that design of this visualization tries to minimize feedback response time so as to minimize the amount of time exploring incorrect solution paths (Schooler & Anderson, 1990). In essence, the design attempts to react immediately to user feedback rather than wait for a user to click a “submit” button or a similar control to trigger feedback. The immediacy of the feedback significantly reduces the time needed for the feedback loop, in turn aiding student learning.

 *Figure 2 – Prototype sketch of an interactive gadget to teach color*

### Concept Maps

Teaching programming concepts is often difficult to teach and this is due at least partially to the high interrelatedness of the many concepts (Wood et al., 2007). Concept maps are one way of helping students understand the relationships between topics. A concept map visually displays various nodes of information. The nodes are represented as boxes with labels. Arrows are then placed between the nodes which specify the direction of a relationship between the connected topics. Typically a label will be placed on the connecting relationship that explicitly explains how the nodes are related (Novak & Cañas, 2008). In this project the arrow labels were purposefully excluded in order to create a cleaner view without excessive text and noise.

This concept map was used as the central navigational mechanism for the site, allowing students to seamlessly navigate between topics while helping them create accurate schemas in memory. Using a concept map as the navigation mechanism also provided a natural way of sequencing the material. Additional details on how this was implemented can be found in the develop phase section of this paper.

### Activating Prior Knowledge

According to schema theory, we learn new information as we put it in context of formerly acquired knowledge (Mcvee, Dunsmore, & Gavelek, 2005). In order to help students understand new concepts, the instruction was designed to include material that would relate it to concepts with which they would likely already be familiar. For example, a student learning Hyper Text Markup Language (HTML) for the first time could potentially be reminded of the meaning of “markup” as it relates to grading English papers. By relating new content to closely associated familiar topics, the learner can acquire knowledge more quickly. All of the content and associated examples were created with this principle in mind. For example, within the instruction, I purposely related the RGB coloring system with the more familiar RYB color system. Similarly *Figure 1* shows various examples of number systems. Each example was purposely chosen to be an example that most students would have already been familiar with in order to aid them in activating prior knowledge.

## Message and Control Layers

The *message* and *control layers* deal with learners’ ability to interact with the educational resource. Specifically the message layer controls how the system communicates the content to the learner and the control layer describes how the learner interacts with the system(Gibbons & Rogers, 2009). This resource was created using a variety of messages and various controls. Some of the main messages included the links between topics on the concept map. Each link provided a little message about how different topics were related while providing a natural sequencing mechanism. Once a user had clicked on a node (a control) they were presented with textual messages explain a topic and how to use the interactive model. Finally the interaction with the models themselves severed to convey causal messages about how concepts worked. These messages were controlled by the controls found on each gadget. For example the binary gadget has various “On/Off” button controls. As the user would interact with these controls the message would change to show updated binary numbers.

## Representation Layer

The *representation layer* specifies how the content encoded in messages is made “visible, hearable, and otherwise sense-able”(Gibbons & Rogers, 2009, p. 314). The bulk of the content was made “sense-able” using models, text, or illustrations. I tried to keep the text and content simple and clean while conveying the essential ideas of each concept. Content was arranged on the page clean and simple way following standard web page conventions. The relationships between topics were represented on the concept map, hence it was easy to see ASCII was closely related to binary numbers. The final product of the representation layer is shown in screenshots of the resource which is discussed in greater detail latter on.

## Media Logic Layer

The media logic layer specifies the mechanism by which representations are caused to occur (Gibbons & Rogers, 2009) and what technologies are used to implement the representation layer. Various web technologies were used to encode the content and make it “sense-able” to the learners. Specifically, all the textual content was encoded in HTML tags and then specific CSS styles were applied. Modern JavaScript libraries have a high degree of functionality and allow the ability to generate rich interactions with little effort. The jQuery UI library was used to create animated visualizations for richer content. This layer could then be rendered using a web browser.

In this project some programming frameworks were created to aid in the implementation of various instructional strategies. These frameworks included a system for storing and retrieving content, tracking student progress, a template for creating highly interactive gadgets, and a content visualization system that also severed as the navigation mechanism.

Though the content created for this project was geared towards IP&T 560, it is assumed that the programming concepts were generic enough that they would find use in other educational settings. Hence, to maximize the usage and benefit of this resource, the pieces should be modular enough that they can be easily included on in other instructional systems. To increase the reusability of the content, HTML embed codes were provided for each gadget. An embed code is a small slide of code that can be pasted onto almost any webpage and the gadget will appear in the page when displayed. Youtube and many large sites use this approach to encourage the distribution and reusability of their content. Hence, by lowering the time required to move and reuse the content, it is supposed this will facilitate greater reusability and adoption of the content.

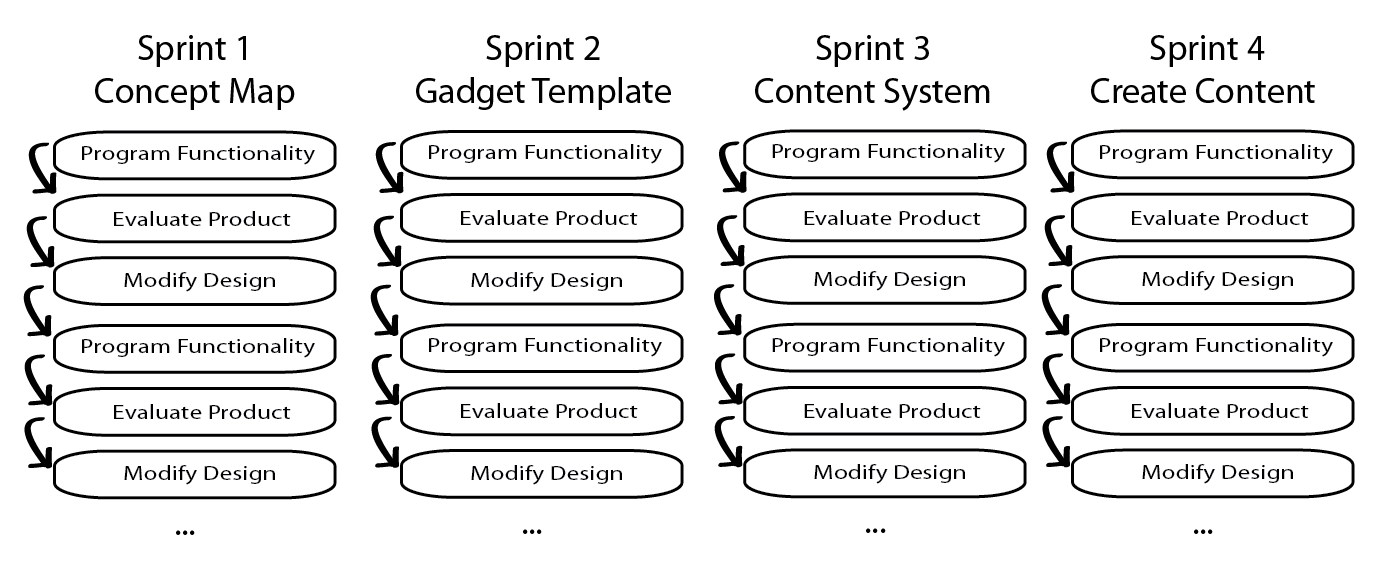
## Data Management Layer

The final layer discussed is the *data management layer*. This layer specifies what data is to be captured and archived. The data can then be used to track learner progress and create relevant metrics. The framework for this project tracked and stored learner progress using a MySQL database on the server. This data was then used to display students’ progress on the concept map, thus facilitating navigation of the six FTC concepts by visually informing learners of the concepts they have or have not mastered.

# Development Plan

The development phase shifts the work from planning and theories to production. In this phase, the designers “select materials and media and conduct formative evaluations” (Peterson, 2003, p. 231). These formative evaluations help the designer to enhance and tune the product while still in development.

The development process employed various aspects of the SCRUM software development model including: scrum meetings, sprints, and breaking the work into small functional units divided into each sprint (Schwaber & Beedle, 2001), with the ultimate goal of being better able to produce a high quality product delivered in a timely manner. Scrum meetings are daily meetings where team members report their progress and their plans each day for increased accountability. Scrum sprints are several-week periods where specific units of work are planned out and executed. This phase took several months to complete. In short, much of the time during this phase was used to program the platform. Each sprint consisted of many small design and develop phases. Each day the resource would work on the product, solicit feedback in a scrum meeting, modify the design, and incorporate the design changes. This smaller iterative processes can be somewhat visualized by *Figure 3*. Details of these small design decisions and their rationales are explained below each sprint. Four main sprints were identified for this project, including sprints for the concept map system, the gadget system, the content engine system, and the sample content. The details of each of these sprints are explained in the following sections.



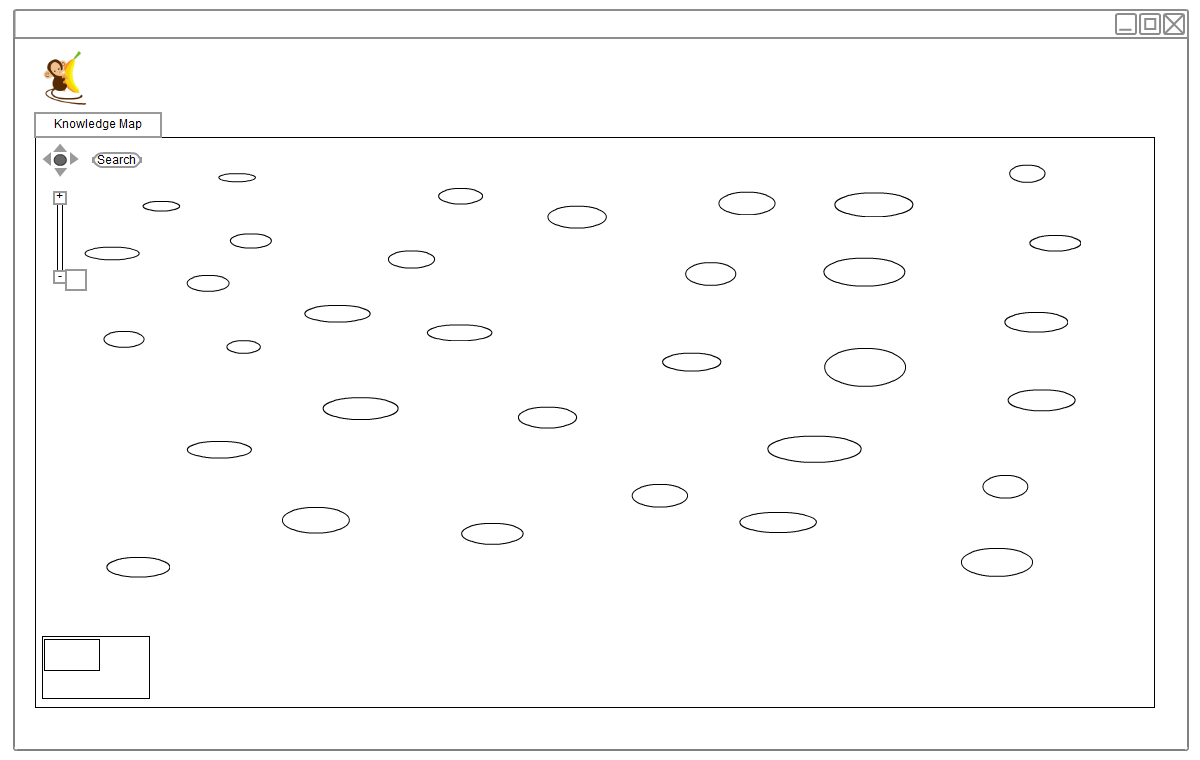
*Figure 3 – Visualization of process followed in each iteration*

## Prototyping

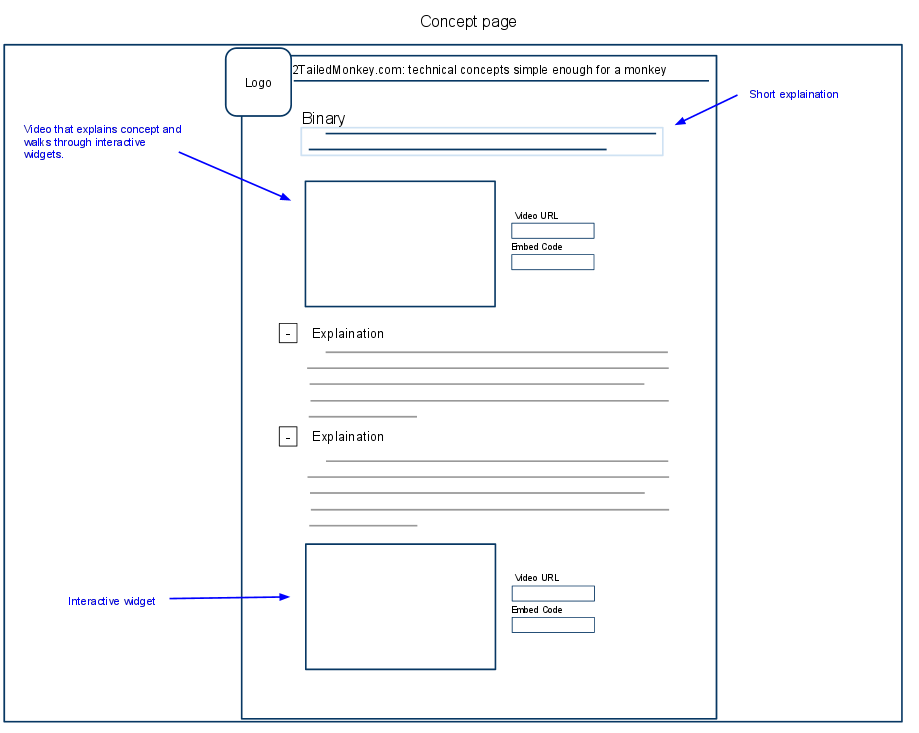
In order to synthesize the above-mentioned instructional theories, it became necessary to prototype the platform by creating a conceptual wireframe of each component. The following design sketches were made using the online vector tools of Gliffy and Google docs. All pages and sketches were designed with the overall goal of making the interface simple and easy to understand. The following sketches were pre-production and do not necessarily reflect the final product, though the differences between the original design and the final product will be discussed later.

The front page of the site was designed with ease of navigation in mind. For this reason the main navigation mechanism is an interactive concept map as seen in *Figure 4*. This is a similar approach that has been used by other successful educational platforms such as the math section of the Khan academy (http://khanacadmey.org). By linking the topics together with arrows, the users are able to see implicitly which topics provide a basis for understanding other related topics. In this way learners are able to begin forming correct cognitive schemas of the different topics with their associated relationships. Further this high-level map view allows learners to view and select only the content they are interested in.

Another key attribute of the concept map is a visual way of tracking students’ progress. As students take the assessments for each topic the corresponding topic turns green on the concept map. In this way students can quickly see their overall progress.

  
*Figure 4 – Prototype sketch of site landing page*

Each node on the concept map links to a topic page. The topic page template was designed with the idea of making the content modular and also following a common design. Each page begins with a short explanation of the topic, followed by a detailed explanation that includes an interactive gadget. The original design sketches for each topic page can be seen in *Figure 5*.



*Figure 5 – Prototype sketch of a topic page*

## Sprint 1 – Concept Map

### Purpose of Sprint

The purpose of this sprint focused on programming a concept map for the educational platform. This sprint took roughly 4-5 weeks. The following table outlines the objectives of the sprint with their corresponding educational purpose.

Table 1

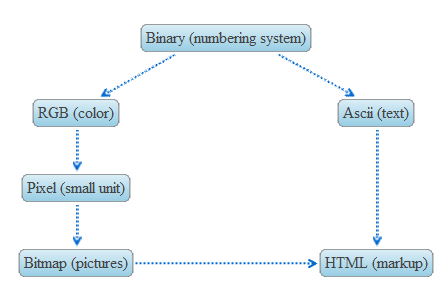
*Overview of the Concept Map Sprint*

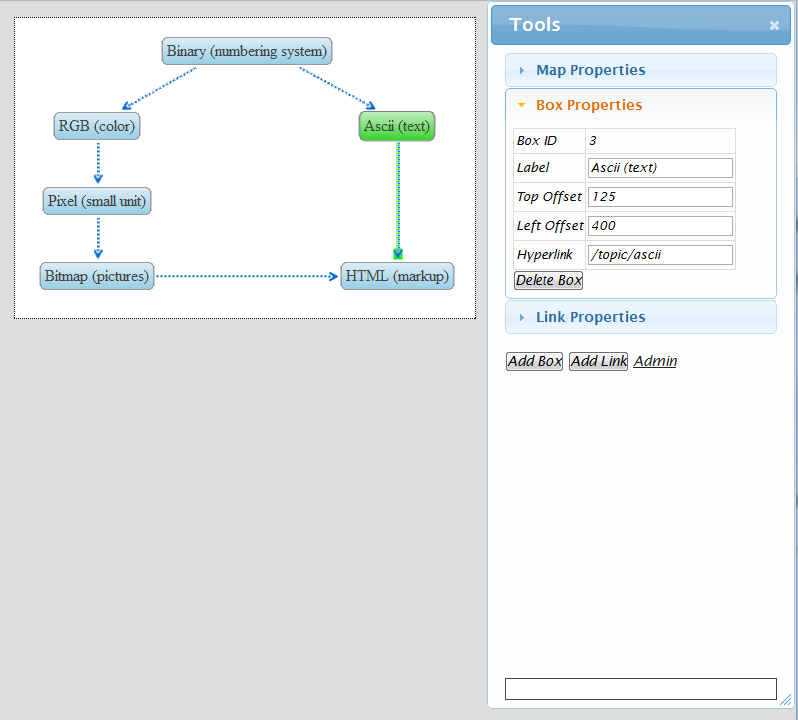
| **Development/Design Objective** | **Educational Purpose** |
| --- | --- |
| Ability to create, delete, and edit nodes. | Each node serves as a link to a particular topic page. Hence the ability to create and edit these nodes is a basic administrative function. In essence nodes are the portals to content. |
| Ability to create, delete, and edit links. | Nodes can be connected by arrows that serve to link topics. These links aid the learner in understanding the correct relationship between topics. This feature is a basic administrative feature to enable the use of links. |
| Nodes have the ability to link to other web pages. | Nodes are to serve as the primary navigation mechanism. Making them clickable allowed them to serve as portals to the topic pages. |
| Ability to drag the concept map around in a smaller window similar to Google maps. | The concept map viewer was designed under the assumption that larger concept maps would eventually be created. Larger maps would be too large to display all their nodes at once. For this reason learners will want the ability to drag and move the concept map around much the same way geographic maps can be dragged around on services such as Mapquest or Google maps. |
| Automatically generated embeddable code for easy reusability. | In order to make it simple for instructors to use the concept maps they need an easy way to embed the content on their site. Embed codes using iframes is a common solution. |

### Design Changes and Development

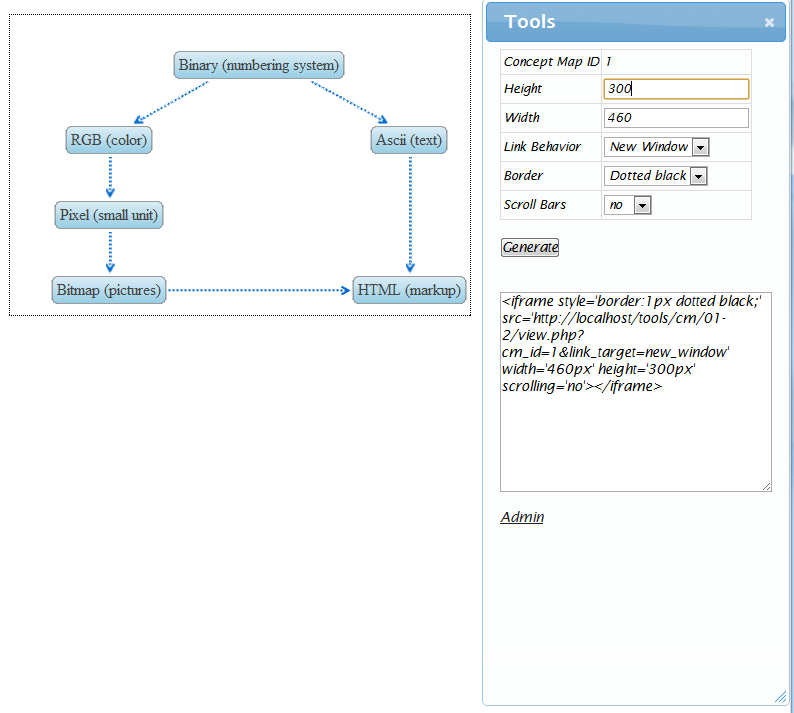
The concept map sub-system was programmed in PHP, MSQL, CSS, HTML, and Javascript. Canvas and Flash were considered for the technology platform but upon investigating each technology, it was decided that Flash lacked support on too many key platforms (namely the iPad and other iOS devices) and HTML5 canvas lacked an easily accessible event listener framework. The concept map system was programmed twice with the respective versions 1.0 and 2.0. This was necessary since the first implementation turned out to lack elegance in the programmatic structure and added needless complexity which would limit the ability to add additional features later on.

Version 1.0 included all the basic functionality described in the design document, including a way to view the concept map (see *Figure 6*), a concept map builder (see *Figure 7*), and a tool to easily create embed codes (see *Figure 8*). Unfortunately, because this was the first attempt, some components were poorly programmed. For example, initially the background used images to add color to the nodes, thus limiting my ability to easily add new node colors. Second, the concept map builder was extremely slow to use and required too many unnecessary clicks. Finally, the overall code was poorly organized, which would have made it significantly more difficult to add new features in the future. For these reasons, it became prudent to scrap that code base and begin recoding the product again, applying the lessons learned from the first round of programming. It is worth noting that it is not uncommon for developers to program something several times. They frequently can do it much better the second time as they understand the requirements and have a better idea of how to implement specific features.

  
*Figure 6 – The concept map view of version 1.0*

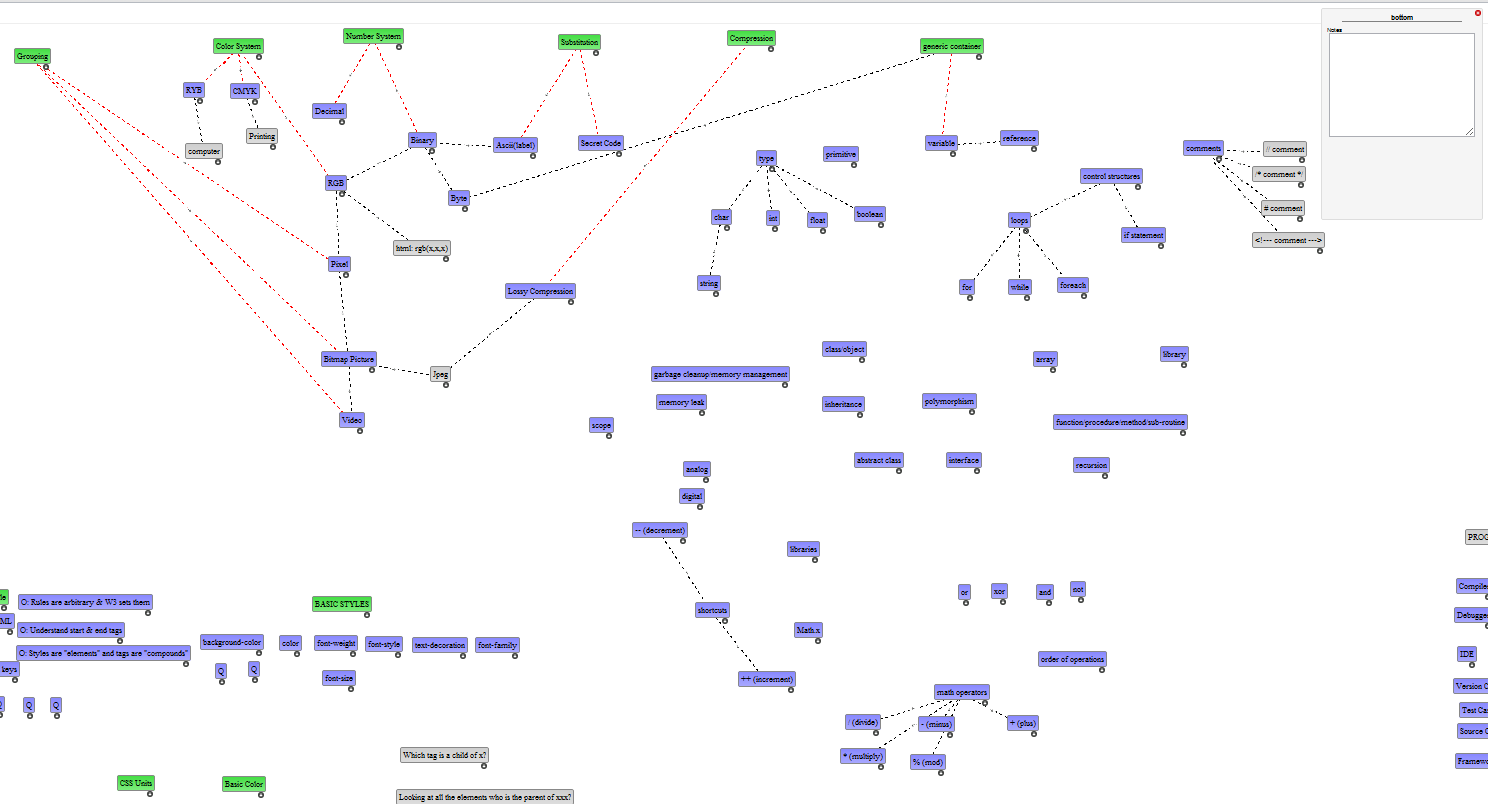


*Figure 7 – The concept map editor of version 1.0*



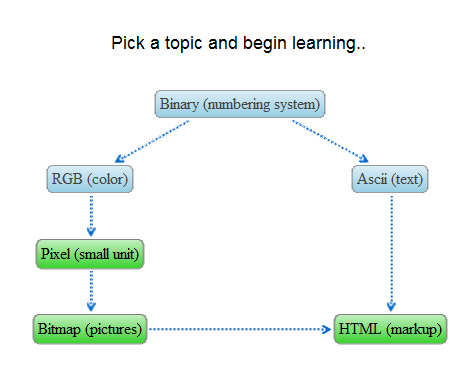
*Figure 8 – The concept map embed tool of version 1.0*

The second version of the concept map focused on improving the user interface. The clunky aspects of the 1.0 editor were overcome and simplified. For example, the process for adding and editing new nodes was greatly simplified in version 2.0. In version 1.0, a user would have to click an “Add node” button, find the node, move it to the correct location, and edit it. Version 2.0 simplified the process by allowing the user to just double click on a location and a new node would appear instantly ready to edit. Enhancements similar to this were also implemented for adding links between nodes. Other key updates included a shift to using JSON objects (a construct similar to XML) to make it easier to extend the data model. I also moved away from using background images and focused on using CSS to make it easier to add additional node types. The overall simplicity of the underlying code was significantly improved which will make it much easier to add new features in the future. A high level screen shot of the significantly enhanced version 2.0 can be seen in *Figure 9.*



*Figure 9 – Version 2.0 of the concept map tool*

Though version 2.0 added many important features, there was not sufficient time in this sprint to add all the desired functionality. Despite improved editing capabilities, the system still lacked a practical viewer module. Plans currently exist to finish the 2.0 version in the future but for the time being, the 1.0 version has been modified to work as an interim solution. One of these modifications made to the 1.0 version is the ability for users to visually see which topics they have already mastered. Topics that have been mastered appear in green while those topics not yet mastered remain a light blue as seen in *Figure 10*. Several modifications similar to this one made version 1.0 a viable short-term solution until 2.0 version can be finished.



*Figure 10 – Version 1.5 of the concept map tool*

### Future Development

The current features implemented on the concept map tool are useful and are sufficient for current instructional needs. However, there are a variety of features that have yet to be implemented. These features and their intended instructional goals are listed in the table below.

Table 2.

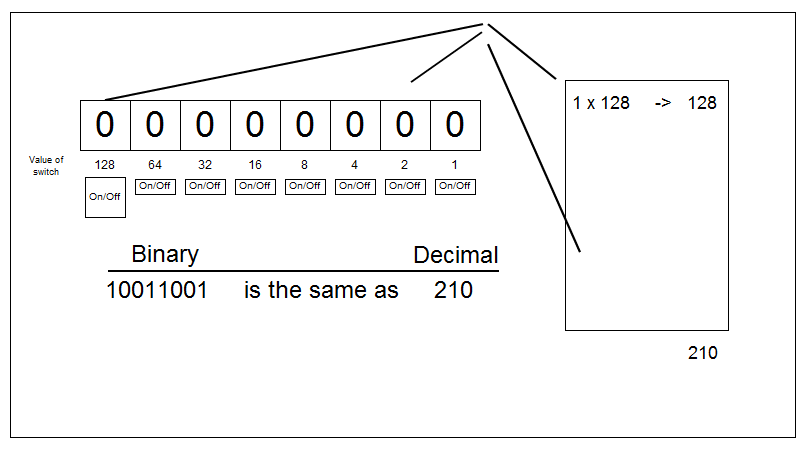
*Future Features for the Concept Map*

| **Development/Design Objective** | **Educational Purpose** |
| --- | --- |
| Persistence of the concept map data in a database rather than in flat JSON objects. | Storing the map in a database will allow for the concept map to be more tightly integrated with the other modules of the learning platform, thus resulting in an improved instructional experience. It will further provide the ability to more easily version the maps. |
| Ability to zoom in and out. | Due to the complexity of some knowledge domains it becomes desirable to “zoom out” as it were and see how all the main topics are connected. This feature allows learners to get the high level picture quickly. |
| Small navigation map at the bottom of the screen. | For large concept maps (where much of the map is hidden) it can become difficult for learners to quickly navigate. For this reason a small map showing the larger picture provides a way to anchor them in the concept maps. Proof of concepts using Javascript have been completed for this feature. |
| Different objects should appear at different zoom levels. | On a complex concept map showing all the nodes can cause the overall appearance to be noisy and so distract the learner. For this reason it is desirable for different nodes and elements to appear at different zoom levels. Depending on time constraints the ability to gradually fade out nodes would also be useful. |
| Ability to specify initial location of the concept map on page load. | When concept maps become too large to fit onto a normal screen instructors will want to zoom in to draw attention to specific nodes. The ability to specify where the map loads will help them accomplish this. |
| Visual indicator of active topic. | Each node on the concept map has a corresponding topic page. If the concept map is to be used as an effective navigation mechanism the current node (the one corresponding with the current webpage) needs to be made visually explicit. For example if I am on the “binary” topic page the “binary” node should be highlighted in some way. |
| Ability to add free floating HTML elements | Currently the tool only allows the creation of nodes and links. Future iterations would benefit from the ability to add pictures and additional content on the concept map. It is supposed that visually stimulating content such as pictures will help engage the learner. |
| Ability to link and edit the content of each topic on the concept map editor. | Currently the web pages associated with each topic are completely separate from the concept map. By linking the web page to the actual node it would greatly enhance the speed at which concept maps and their associated content could be created. |

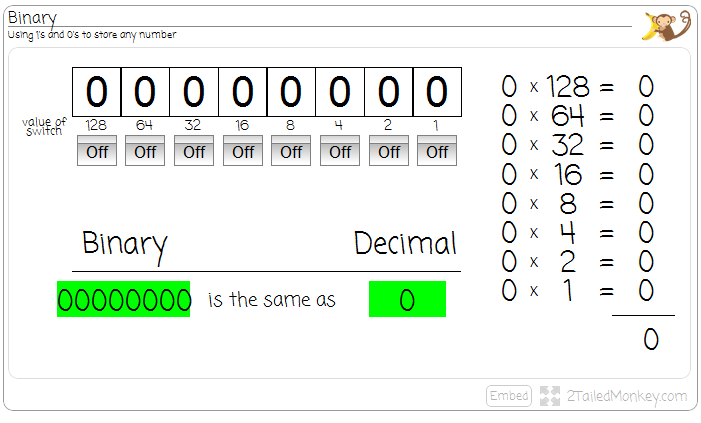
## Sprint 2 – Gadget Template Plus Actual Gadgets

### Purpose of Sprint

In order to teach the concepts more effectively, interactive visualizations or gadgets were planned. Allowing the learner to interact with the visual representation of the concept helps the learner activity engage with the material. Through a series of experimental clicks, they are able to begin understanding how certain conceptual systems work. For example, a prototype for the binary gadget can be seen on *Figure 11* which eventually was programmed as *Figure 12*. Using a series of simple controls, learners are able to experiment with the concept of binary. Each of the interactive gadgets was designed with intentionally simple controls while structuring information in such a way so as to facilitate learning.



*Figure 11 – Prototype sketch of the binary gadget*



*Figure 12 – Actual implementation of the binary gadget*

It became fairly obvious early on that most gadgets would share some common functionality. For example, each gadget would need the ability to generate embed code, maximize the screen, resize for different screen resolutions, etc. In order to speed the creation of these gadgets, a template to handle these basic features was programmed. The following table outlines the objectives and their educational purposes of this gadget template.

Table 3

*Overview of gadget sprint*

| **Development/Design Objective** | **Educational Purpose** |
| --- | --- |
| Easily accessible embed code | This feature aims to address the objective of making the gadget content modular and easy to remix. |
| Check browser version and give appropriate error if the version is not supported | Different gadgets use various new features emerging with the HTML specifications. Users using unsupported browser versions need to be prompted to update their web browsers so the content can render correctly. |
| Content resizes based on screen size | In order to minimize irrelevant stimulus it is supposed that many instructors will wish to maximize the gadgets on the screen while explaining the content. This functionality allows the instructor to enlarge the gadget so all learners can easily see it. |
| Link to an external link as needed | In some instances video supplements will be provided for these interactive gadgets. Providing external links on each gadget will allow the gadgets to serve as portals to additional content. |
| Separate the content version from the template version | In keeping with the layered approach to design, the content of each gadget should be separated from the programming logic. This allows the content of each gadget to be unaffected by new versions of the gadget template. |

### Design Changes and Development

All of the functionality above was successfully programmed into the gadget template. It is worth noting that during the course of development, it quickly became apparent that the gadget content layer (images, colors, shapes, interactivity, etc.) would need to be separate from the template code. By programmatically separating these two layers, the template allows for these layers to “slide” by one another seamlessly. Due to this design, the template and the content can be updated independently from each other.

One of the key areas that caused some serious reflecting in this sprint came in trying to decide the value of adding dynamically resizing visualization code. In essence, the question was “How valuable is it to an instructor to be able to enlarge the interactive gadget to fill the entire screen?” Ultimately I decided this feature was worth adding to the template for the following reasons: (1) by adding the feature to the template, it would require minimal programming effort to include the same feature to gadgets in the future; (2) enlarging a gadget to fill the screen reduces content noise by removing all stimuli except the gadget; (3) the resizing code would also implicitly allow me to include small versions of gadgets on assessment questions, allowing a greater degree of flexibility in creating questions.

### Future Development

There are some minor bugs and cosmetic changes that need to be fixed and added in future versions. However, this layer has no major items that need to be added for instructional reasons at this point. The only concern stems from the lack of flexibility of the template itself. Only one gadget has been developed fully with this system and the adaptability of this template system has not been fully tested. Depending on future gadget needs, this layer will likely be updated.

## Sprint 3 – Content Engine

### Purpose of Sprint

The purpose of this sprint was to create the content engine that would house the actual content for each topic. The engine needed the ability to bring together the various elements such as concept maps, the authentication system, the interactive visualizations, assessment, text, and picture content. In a sense, this engine is the glue that pulls the different sub-systems together. Some of the key features planned for in the sprint included:

Table 4

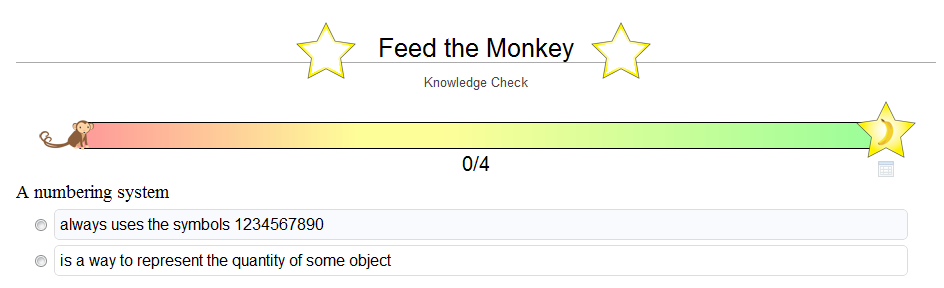
*Overview of content sprint*

| **Development/Design Objective** | **Educational Purpose** |
| --- | --- |
| Ability to create versioned topic pages | Provide the learner with pages specifically designed to deliver content. Versioning was added to help authors track their changes. |
| Ability to fold/hide of content | To prevent overloading learners with too much information. This allows instructors to include relevant, but not critical, information on each lesson without cluttering things. This took the form of content that was partially hidden until clicked on. |
| Ability to zoom in on pictures | In order to preserve screen real estate many of the visualizations are small to start out with. However, anticipating that instructors and students would want to view images at a larger size a simple way of zooming up on the picture was added. |
| Authentication system | In order to track student progress the identity of the user must be verified. |
| Automatic creation of a table of contents | Table of contents help instructors assess the material coverage of their test and help learners understand |
| Ability to include multiple choice questions at the bottom of the content to assess learning. | Formative assessment can frequently help students learn more effectively by providing immediate feedback. |
| Dynamic generation of a table of specifications | A table of specifications helps instructors evaluate the types of items and content covered in an assessment. |
| Integration of the assessment system with the concept map system | By integrating the assessment system with the content map users are able to quickly view what items they have mastered and what items they still lack mastery on. |

### Design Changes and Development

A variety of different design changes and development challenges surfaced during this sprint. One of the first issues dealt with deciding where to persist the data—in a database or on a file system. Ultimately the file system was chosen for two reasons. First, most of the development took place on a laptop and it is much easier to synchronize file systems than it is databases. Second, using the file system made it much easier to version media objects such as images. As a downside, it should be noted that using a local file system does make it more difficult for others to contribute and add their own content. However, since this is just a version 1.0 of the platform, speed of content creation was valued higher than collaboration so ultimately the decision was made to persist the data on a file system.

The second main item that surfaced dealt with creating an assessment system. Originally, assessment was not necessary planned. However, its pedagogical usefulness became obvious as I observed how assessment was being used at the Khan academy. I was particularly drawn to the fact that students could visually track their progress as nodes on the concept map turned green as they mastered topics. For these reasons, I decided to build and integrate a formative assessment system into the content engine. The assessment system was designed to only support multiple choice questions for a variety of reasons. Namely: (1) multiple choice questions can test a variety of different learning outcomes, (2) multiple choice questions can easily be machine scored, (3) it is a format that most students are already familiar with, and (4) it took less time to program. *Figure 13* shows what the final multiple choice engine looks like.



*Figure 13 – Screenshot of the multiple choice assessment engine*

Another feature added during the content phase was the ability to authenticate and track user progress. Authentication was accomplished by using the authentication services provided by Google, Yahoo, and Facebook. It is assumed most users will have an account with at least one of these online providers. One acknowledged disadvantage of using three providers, opposed to just one, stems from the fact each login creates a new account. Hence, it is possible for users to login with Google on one visit and use Facebook on another visit, and create two separate accounts. In future versions of this product, it might be prudent to restrict this service to a single third party authentication provider to prevent confusion caused from multiple accounts.

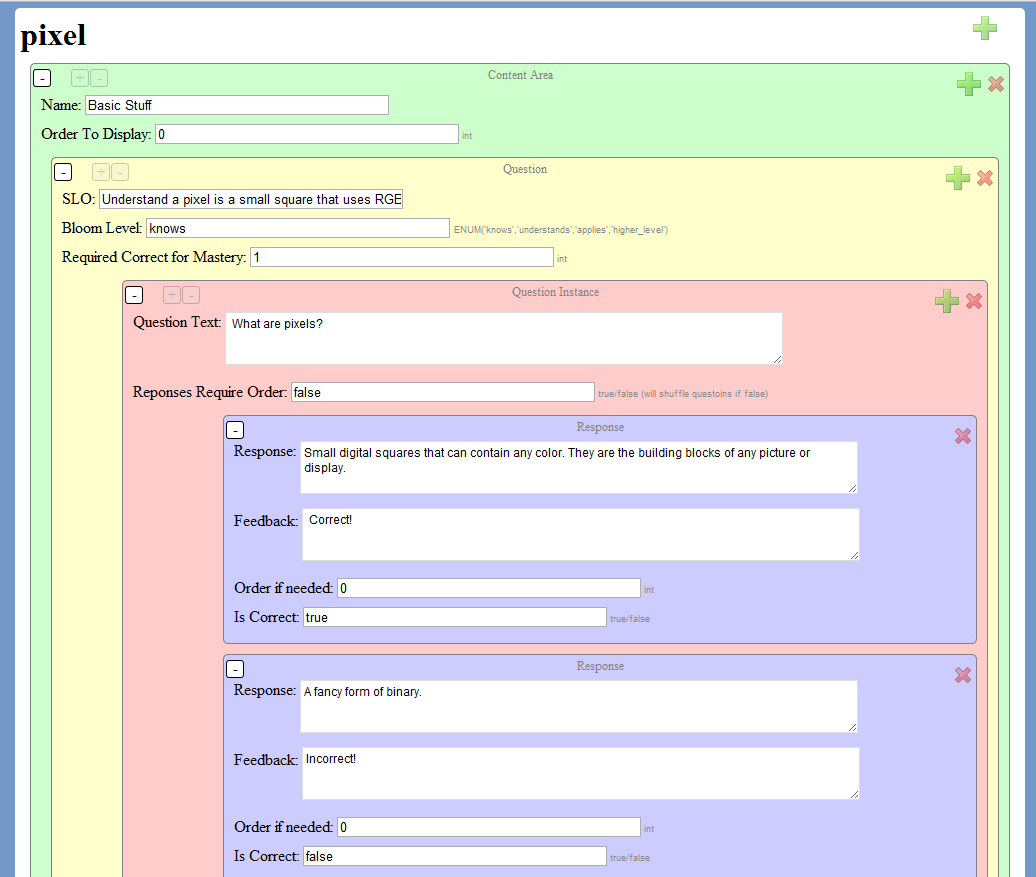
### Future Development

The current implementation of the content engine works well for the amount of content that is current created. However, various features will be developed in future iterations of the platform. Some of the key features that will be added plus their instructional purposes are as follows.

Table 6

*Overview of future features for the content engine*

| **Development/Design Objective** | **Educational Purpose** |
| --- | --- |
| Allow collaborative editing of the content | Because the current content engine does not persist the content in the database it makes it difficult for multiple users to edit content. The ability to add collaborators on a node will increase the quality of content. |
| Allow reuse of questions | Questions are currently tied to a specific node and are not easily reused. Adding the ability to reuse and copy questions should speed the content creation process for instructors. |
| Create a friendly assessment builder. | The user interface for the current assessment builder is functional but slightly cryptic. It assumes the instructor knows the nuances of how it works while providing very little instruction on how to use it. Creating a nicer assessment builder should speed the creation of quality assessments. A screenshot of the current builder can be seen in *Figure 14*. |
| Add additional assessment types | The current assessment builder only supports multiple choice questions. Additional items types are planned for future iterations such as short answer and matching. These additional assessment types will help instructors customize the formative assessment to their specific class demographics. |
| Allow students to join groups and grant the instructor access to a console for monitoring student progress | Currently the system tracks the progress of each student as they “master” various topics. However, there is no reporting mechanism for the instructor. Adding a report of some type will help the instructor identify students who are struggling and need additional help. |

 *Figure 14 – Screenshot of the multiple choice master builder.*

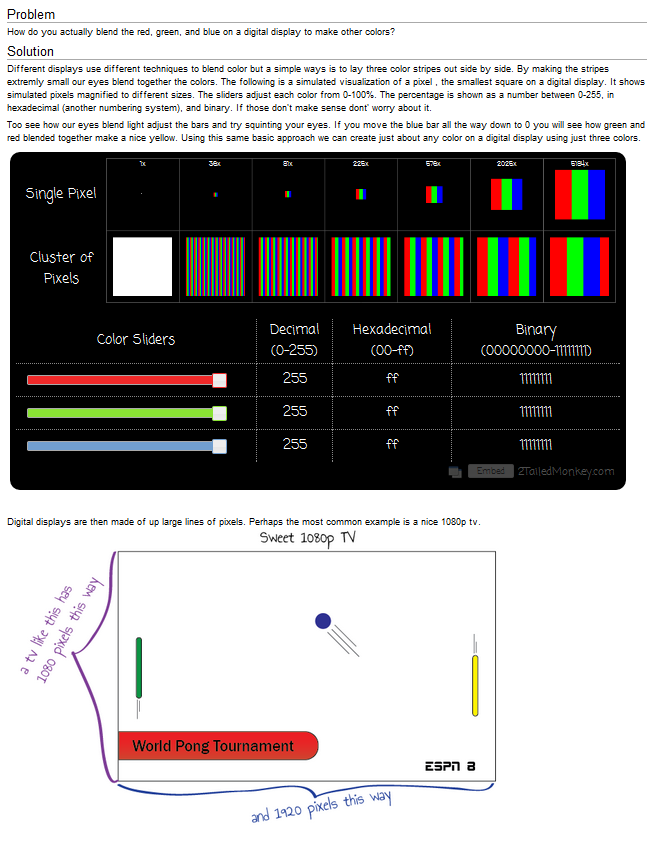
## Sprint 4 – Content

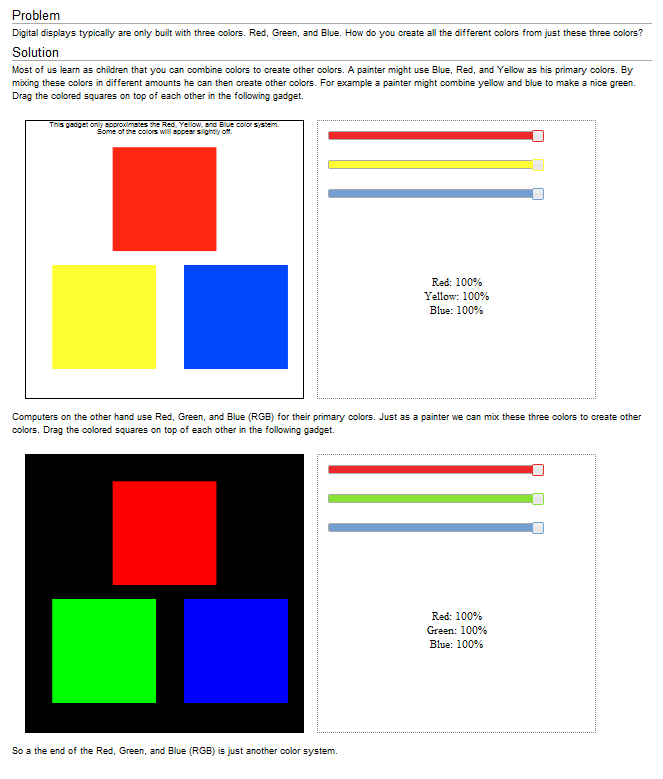
### Purpose of Sprint

The purpose of this sprint was to create sample content to test out the effectiveness of the FTC instructional platform. There was not sufficient time to create content for a large number of topics, so a handful of purposefully selected contents were selected. The chosen concepts were binary, RGB, ASCII, pixels, bitmap, and HTML. These concepts were selected for their wide use in a variety of different technology settings.

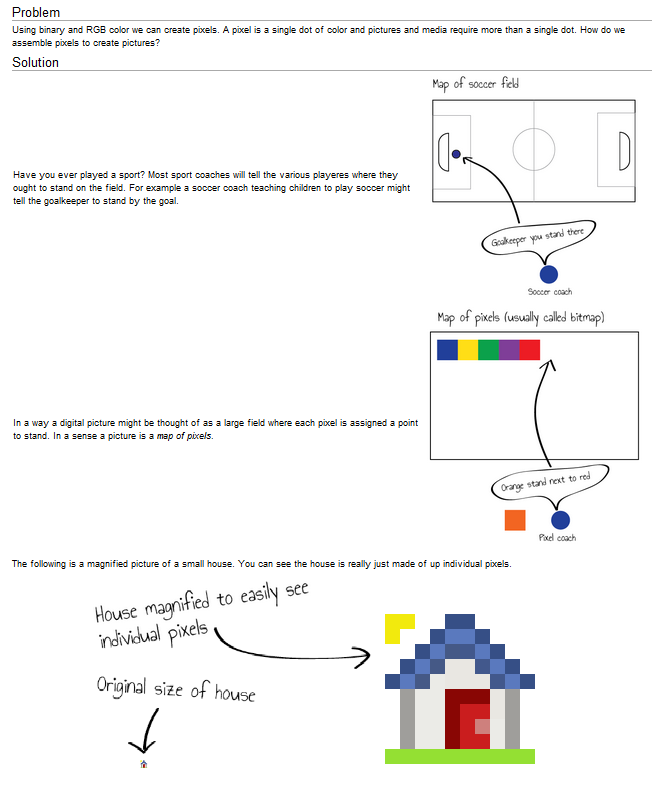
### Design Changes and Development

Development of content was a fairly smooth process. For each concept, several deliverables were generated. These deliverables included: (1) textual explanations that were brief and to the point, (2) pictures (if deemed useful), (3) an interactive gadget, and (4) assessment questions. Building the content for each of these concepts took longer than expected, with much of that time spent building the gadgets. Screenshots of the finished content can be seen in *Figures 15 through 20.*

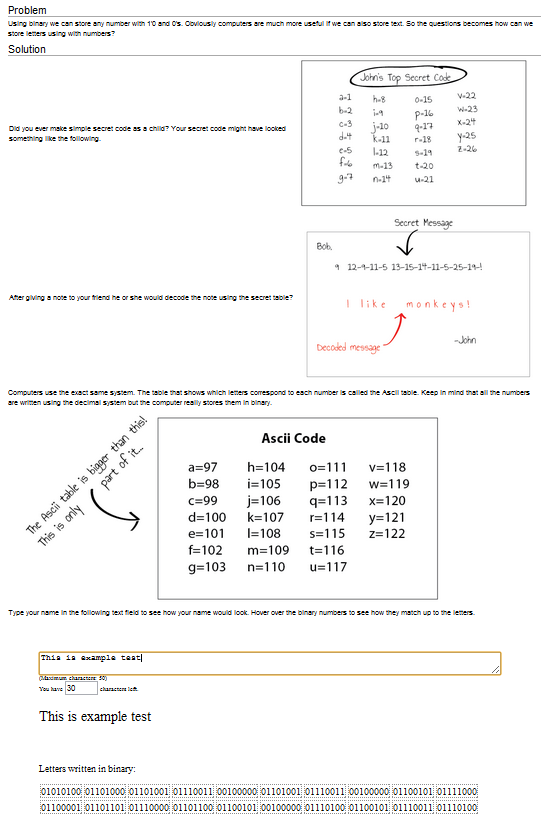
  
*Figure 15 – Content for the topic of “Pixel”*



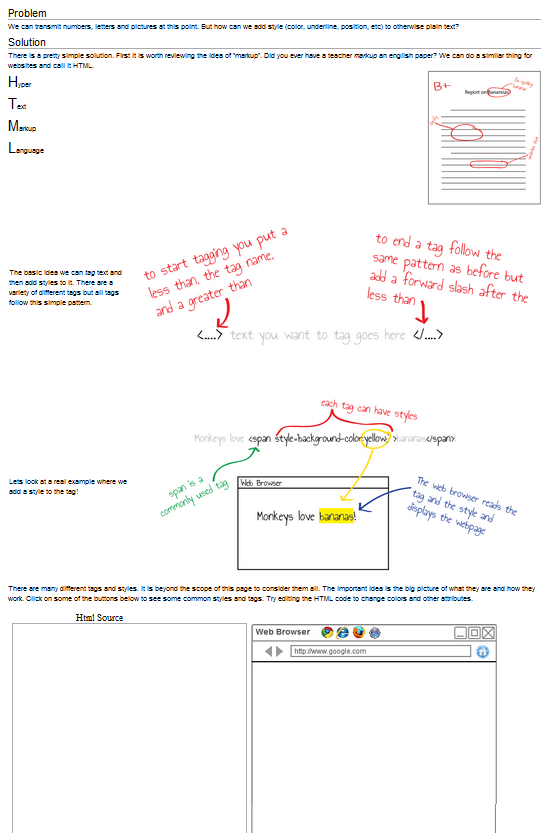
*Figure 16 – Content for the topic of “RGB”*



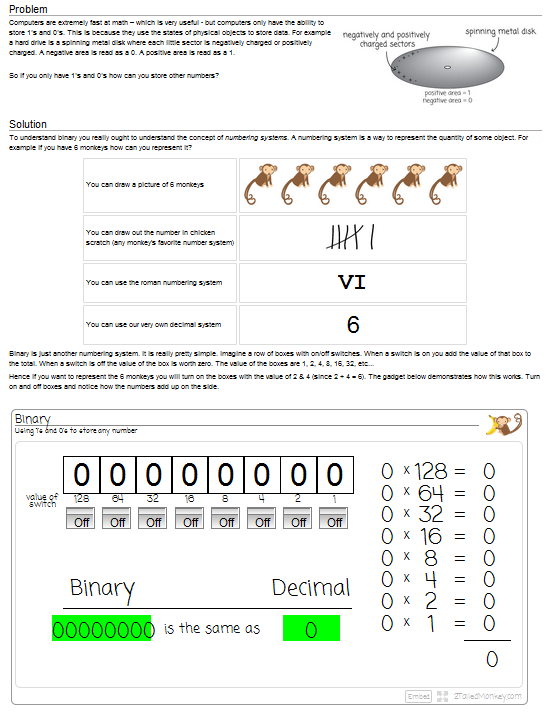
*­Figure 17 – Content for the topic of “Bitmap”*



*Figure 18 – Content for the topic of “ASCII”*



*Figure 19 – Content for the topic of “HTML”*



*Figure 20 – Content for the topic of “Binary”*

### Future Development

This project was able create content for six different FTCs. A much larger concept map was informally assembled as part of this project showing topics in the domains of databases, cryptography, programming, and other related computer areas. Each of these domains has various FTCs. Though this project mainly focused on creating the platform for teaching these FTCs, there are plans to create more content in the future. Additionally the assessment system allows for a large question bank to be used. Though sufficient questions were generated for this project, ideally a larger bank of questions will be generated in the future to lessen the likelihood of student encountering the same questions multiple times.

# Implement Phase

During the implement phase, the educational product is put into production environments (Gustafson & Branch, 2002). This allows end users to begin using the newly constructed resources. If training is needed on the usage of the product, this is also provided in this phase. In order to implement the platform, the domain of 2tailedmonkey.com was purchased. Hosting was placed on an existing account maintained by the designer. Afterwards, the code was copied to the remote server and the database was migrated. The implementation was easy and straightforward. The gadgets, concept map, and assessment engine were designed to be self-explanatory to limit the amount of instruction required for learners to use the product. Most users were able to Figure out how the various software tools worked but some informal training on the tool was conducted in the graduate lab.

# Evaluation Phase

Evaluation occurs throughout the development process and at the end of the project to see if the product meets the instructional objectives (Gustafson & Branch, 2002; Peterson, 2003). Evaluation took several different forms in project. The first type of evaluation occurred in small incremental steps each day at SCRUM meetings. At the meeting, the customer would evaluate the product and provide feedback. The evaluation feedback provided would then be integrated into the design and future versions of the product.

The second type of evaluation took the form of a formal survey prepared to distribute to the IP&T department mailing list. The mailing list was selected as a sample of convenience but it should be noted that all evaluators at least had some grounding in instructional theory. The intent of the survey was not to make statistical inferences but rather to gather general thoughts and feedback from potential users, some of whom had already taken IP&T 560 and others who had not. Participants were asked to provide feedback on the various topics as well as on the quality of the content and platform. The questions sent in this survey can be found in Appendix A. A total of eight individuals submitted feedback in this survey. All participants were offered small amounts of candy for participating in the survey.

The feedback from the participants can be grouped into several different categories. First, most of the users found the resource helpful and educational. Comments of this type included statements such as “Overall...Awesome learning tool! I actually could really use this type of tool in my classroom this next fall” and “Cool! It kept me engaged all the way through. The interactivity makes a huge difference.” In addition to the positive feedback, the users also provided insights into features of the resource that were difficult to use and which could be improved. For example one user suggested a color change to the binary gadget to make it more readable. Another user suggested moving some user controls to make certain gadgets less confusing.

Finally, several users suggested enhanced assessment items. They complained of the assessment items being “tedious” or “too simple”. One participant suggested adding more practice before taking the assessment. Despite some complaints about particular assessment questions, most found the assessment system intuitive with a mean score of 1.65, with 1 being very easy to use and 5 being difficult to use. Overall the feedback was positive with feedback suggesting improvements and fixes.

Given the user feedback and deficient features mentioned in each sprint, it is obvious that a variety of new additional features are needed in future releases of this product. The future development and research can be broken down into several broad categories including: (1) enhancing the software platform, (2) the creation of additional FTC content, and (3) the development of associated instructional theory.

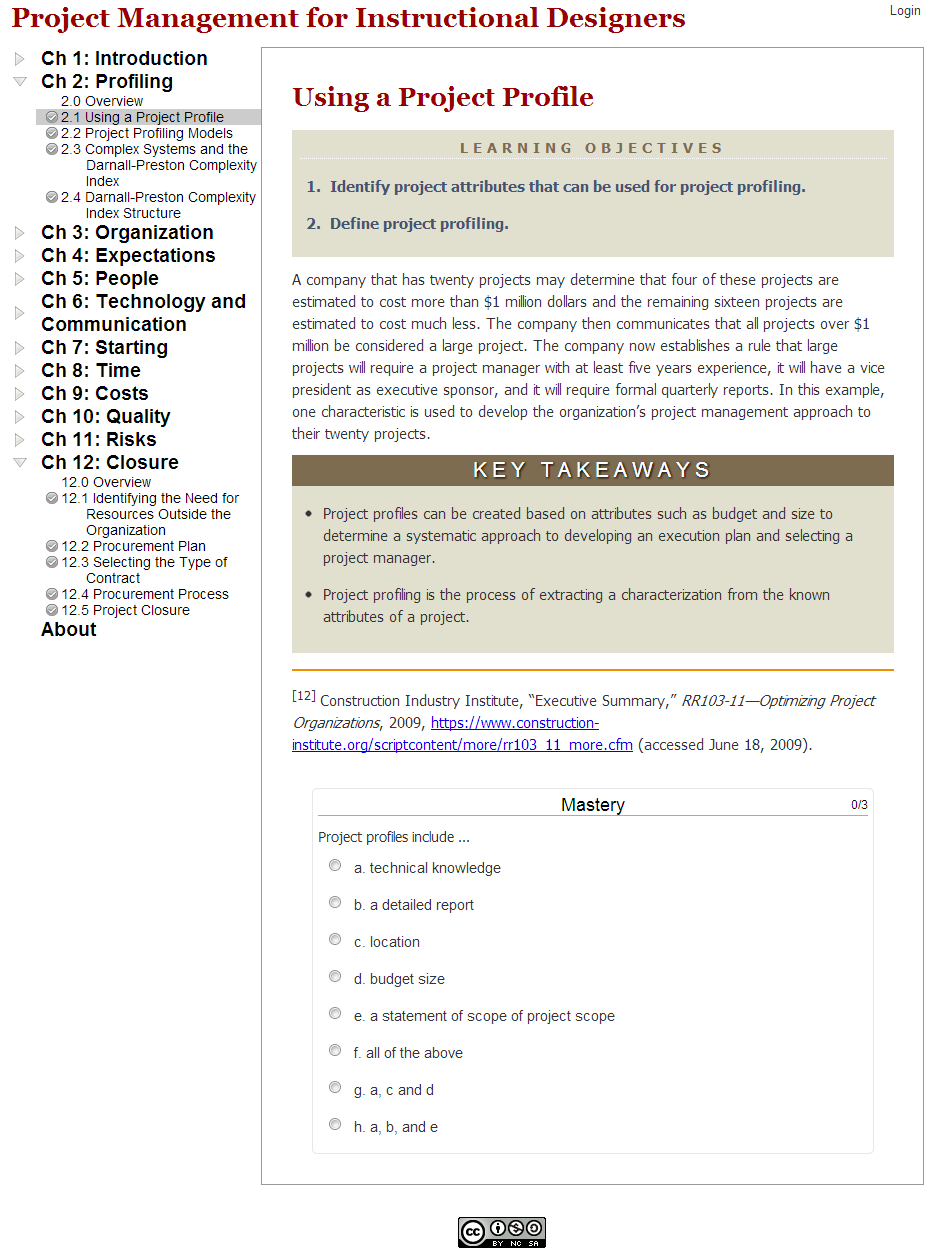
The specific features and objectives for improving the software platform have been outlined in previous sections and I will not articulate details here. Additionally, it can be assumed additional content will need to be created in order for the resource to have sufficient material to attract users. There is still a great need to articulate and develop the instructional theory regarding the best way to teach foundational concepts. Additional questions that might be probed in this include: What are foundational concepts? How might they be defined and identified? What instructional methods are best suited to teaching foundational concepts? How does focusing on foundational concepts affect things such as transfer, retention, learner self-efficacy, and other cognitive measurements?

# Reflection

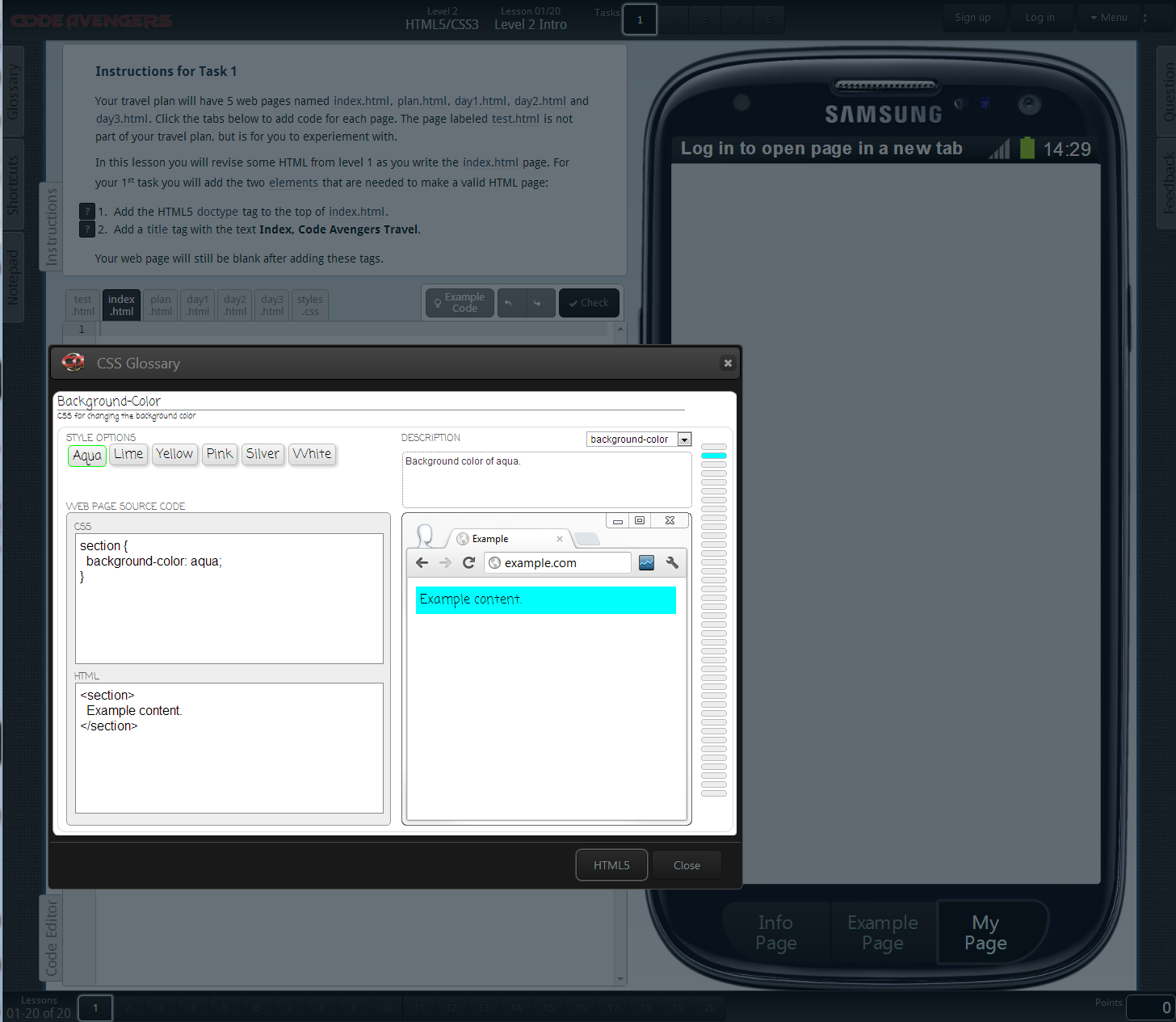
As I reflect on the project I am struck by several things. I am pleased at the final product and feel confident the lessons learned in the development process will inform and enhance my ability to design and build effective instruction in the future. There are various things I would do differently the second time around. The main lessons learned and what modifications I would make to my design and development process are discussed below.

First, for this particular project I think the concept of designing in layers was useful, though the exact layer breakdown suggested by Gibbons and Rogers (2009) was not always easy to separate programmatically. To clarify, it was not too difficult to break functionality into the different design layers on paper but translating that to programming structures was more difficult. For example, when designing the gadgets, it was not practical to separate the control and message layers since the same code controlled both of those layers. Gibbons and Rogers describe layers as constructs that can “slide” by each other. However, because the code for one layer is so directly tied up in the code of another layer it is often difficult to make layers that truly “slide” in a programmatic sense.

Even though the final product may not have strictly conformed to the suggested layers as outlined by Gibbons it was nonetheless extremely helpful to design the product in terms of layers. Also the platform, assessment system, and gadget systems were programed in a layered approach that made them far easier to adapt for other educational purposes. For example the assessment layer was able to be easily repurposed for an open textbook on project management for instructional design. *Figure 21* shows what the repurposed assessment layer looks like at idpm.us as of Jan 2013. The main difference between the original version and the repurposed version of the assessment layer was the removal of the monkey and the slider bar since they would not have been appropriate for the site’s content.

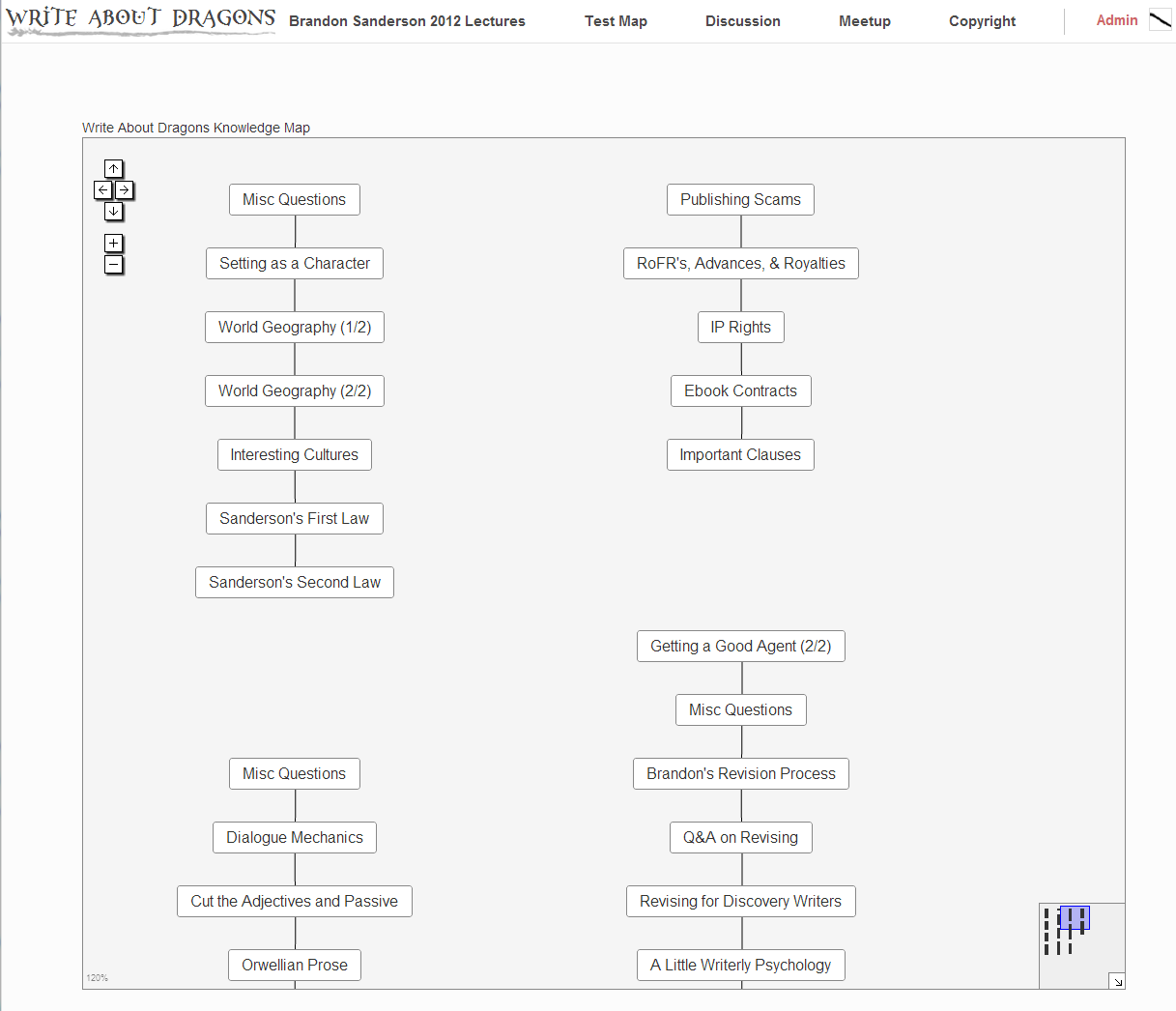
  
*Figure 21 – The repurposed assessment layer at idpm.us as of Jan 2013*

Since releasing the project I have had at least one website adopt some of my content. CodeAvengers.com helps students in New Zealand learn HTML, CSS, and Javascript. The site content is focused at an introductory level which is perfectly suited for FTCs. Upon seeing my CSS gadget the site owner asked if he could adapt some of my code and use it as his CSS reference. Due to the fact that the gadget had been layered in such a way that the content layer was completely separated from the other layers it was not difficult to customize the content for his site as seen in *Figure 22*. This is but one example of how layering things allows for the reuse of content that would not otherwise be adaptable. Currently his site provides my gadget with 300-400 unique visitors a week.



*Figure 22 – An example of the content layer being changed on the CSS Style gadget at Codeavengers.com*

Another layer that is currently being repurposed for other educational uses is the concept map. Writeaboutdragons.com is a website focused on helping students learn about fantasy writing and gets about 6000 video views a week. The next version of the website is planning to incorporate the concept map to help students get a high level view of what topics are covered on the site. Once again, because of the way the content was separated from the functionality the site is able to integrate the concept map layer with minimal effort. *Figure 23* shows the current prototype concept map being developed. By designing systems in a modular/layered way it becomes far more practical to reuse educational material.

  
*Figure 23 – An example of the concept map being repurposed at writeaboutdragons.com*

As I reflect from a holistic standpoint, I admit that using ADDIE process was more difficult than I had anticipated. I believe this mainly stemmed from the fact that ADDIE seems to be well suited for generating instructional materials, while this project involved a fair amount of system building. In retrospect, perhaps a different approach would have better suited the design and development of the platform, whereas the generic ADDIE model seems to be better suited for creating content. Additionally standard deliverables such as task-analysis were more difficult to create for the content selected. I believe this is due to the nature of FTCs which are already very foundational by nature and are not easily decomposed into smaller tasks. Granted, binary might be decomposed into the ideas of “understand symbols” and “understand numbering systems” but usefulness of breaking this content into these smaller tasks is questionable since some skills are assumed to be present in the learner. In short, graduate students are assumed to already understand the idea of “symbols” and “numbering systems”.

The ADDIE model was useful in the sense that it allowed me to think of the work to be accomplished in a step-by-step way. This linear approach was somewhat complicated because I was trying to (a) design a robust platform for content, (b) generate a gadget layer that was reusable in various settings, (c) define what curriculum would be useful to instructional designers, and (d) actually design and build that content. I think the ADDIE model is better suited to the last of those four tasks. In retrospect I was spread a little thin as a result of trying to accomplish all four tasks. Given that my purpose was to generate good content for instructional designers I think it would have been better to focus on just making good content and use an existing platform such as Wordpress. Had I done this, I suspect I would have been able to generate substantially more instructional material, which in turn would have been more useful to the IP&T 560 class.

In conclusion, this project has allowed me to build a fully working first-generation platform where I can design and build FTC content. At the time of writing, additional gadgets have already been designed and built at <http://2tailedmonkey.com>. Additionally some of the features mentioned in “future” versions are already implemented such as the “zooming” on the concept map. The instructional resource designed and developed in the project will continue to gain additional instructional content and platform features. Based upon my lessons learned I understand the important of focusing on either the platform or the content and not trying to accomplish both of them at the same time. Ultimately these features and content will help instructional designers better understand technology and create better instructional material.

# References

Gibbons, A. S. (2003). What and How Do Designers Design: A Theory of Design Structure. *TechTrends*, *47*(5), 22–27.

Gibbons, A. S., & Rogers, P. C. (2009). The Architecture of Instructional Design. In C. M. Reigeluth & A. A. Carr-Chellman (Eds.), *Instructional-design Theories and Models III: Building a Common Knowledge Base* (p. 305).

Gibby, S., Quiros, O., Demps, E., & Liu, M. (2002). Challenges of Being an Instructional Designer for New Media Development: A View from the Practitioners. *Journal of Educational Multimedia and Hypermedia*, *11*(3), 195 – 219. Retrieved from http://editlib.org/p/9266

Grissom, S., McNally, M. F., & Naps, T. (2003). Algorithm Visualization in CS Education: Comparing Levels of Student Engagement. *Proceedings of the 2003 ACM symposium on Software visualization* (pp. 87–94). San Diego, CA.

Gustafson, K. L., & Branch, R. M. (2002). What is Instructional Design? *Trends and issues in instructional design and technology*.

Hunter, E. D. (2007). *Using Interactive Diagrams in a Web-Based Tutorial to Teach Graduate Students about Statistical Power*. Brigham Young University.

Larson, M. B., & Lockee, B. B. (2007). Preparing Instructional Designers for Different Career Environments: A Case Study. *Educational Technology Research and Development*, *57*(1), 1–24. doi:10.1007/s11423-006-9031-4

Lin, H. (2000). Fluency with Information Technology. *Government Information Quarterly*, *17*(1), 69–76.

Mcvee, M. B., Dunsmore, K., & Gavelek, J. R. (2005). Schema Theory Revisited. *Review of Educational Research*, *75*(4), 531–566.

Molenda, H. (2003). In Search of the Elusive ADDIE Model. *Performance iIprovement*, *42*(5), 34–36.

Naps, T., Guido, R., Darmstadt, T. U., College, M., Dann, W., Cooper, S., Fleischer, R., et al. (2003). Evaluating the Educational Impact of Visualization. *ITiCSE-WGR  ’03 Working group reports from ITiCSE on Innovation and technology in computer science education* (pp. 124–136).

Novak, J. D., & Cañas, A. (2008). *The theory underlying concept maps and how to construct and use them*. *Florida Institute for Human and Machine Cognition Pensacola Fl, www. ihmc. us.[http://cmap. ihmc. {us/Publications/ResearchPapers/T} {heoryCmaps/TheoryUnderlyingConceptMaps.} htm]*. Pensacola, FL.

Peterson, C. (2003). Bringing ADDIE to Life : Instructional Design at Its Best California University of Pennsylvania. *Journal of Educational Multimedia and Hypermedia*, *12*(3), 227–241.

Schooler, L. J., & Anderson, J. R. (1990). The Disruptive Potential of Immediate Feedback. *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society* (pp. 702–708).

Schwaber, K., & Beedle, M. (2001). *Agile Software Development with Scrum* (1 edition., p. 158). Prentice Hall Upper Saddle River.

Sugar, W., Hoard, B., Brown, A., & Daniels, L. (2012). Identifying Multimedia Production Competencies and Skills of Instructional Design and Technology Professionals: An Analysis of Recent Job Postings. *Journal of Educational Technology Systems*, *40*(3), 227–249.

Wood, J., Dykes, J., Slingsby, A., & Clarke, K. (2007). Interactive visual exploration of a large spatio-temporal dataset: reflections on a geovisualization mashup. *IEEE Transactions on Visualization and Computer Graphics*, *13*(6), 1176–1183.

Appendix A

**2TailedMonkey Feedback #1**

Please go to http://www.2tailedmonkey.com and test out the content before answering this survey.

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Email \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Were you able to login with Facebook, Google, or Yahoo?

Yes No

Did you master all of the topics? (they turned green on the knowledge map)

Yes No

Which topics seemed difficult to understand?

Binary RGB Pixel Ascii Bitmap HTML

How intuitive was the testing system? (Feed the Monkey)

Very Intuitive 1 2 3 4 5 Not Very Intuitive

What topic was most difficult to understand and why?

What topic was most easiest understand and why?

What suggestions would you have for the interface?

What suggestions would you have for the content?

Other Feedback